

| | |
|--------------------------------|------------|
| ④ ブランクの読み飛ばし | 221 |
| ⑤ \$00で終了する文字列の管理テーブルを作成 | 223 |
| ⑥ 文字列の内容を交換する | 226 |
| ⑦ 文字列の比較を行う | 229 |
| 16 FIFO(Queue) | 232 |
| ① メモリ上にFIFO(Queue)を実現する | 233 |
| 17 コード変換 | 241 |
| ① ビット列を意味する文字列を内部表現(バイナリ)に変換する | 242 |
| ② 16進文字列をバイナリ表現に変換する | 245 |
| ③ 10進文字列をバイナリ表現に変換する | 248 |
| ④ 10進文字列をBCD表現へ変換する(1) | 250 |
| ⑤ 10進文字列をBCD表現へ変換する(2) | 252 |
| ⑥ バイナリ表現をビット列の文字列へ変換する | 254 |
| ⑦ バイナリ表現を16進文字列へ変換する | 257 |
| ⑧ バイナリ表現を10進文字列へ変換する | 260 |
| 18 モジュール別開発と市販ツール | 263 |

| | |
|------------------------|------------|
| 第3部 命令の詳細 | 265 |
| 1●MOVE | 266 |
| 2●MOVEA | 268 |
| 3●MOVE to CCR | 269 |
| 4●MOVE to SR [特権命令] | 270 |
| 5●MOVE from SR | 271 |
| 6●MOVE from USP [特権命令] | 272 |
| 7●MOVE to USP [特権命令] | 273 |
| 8●MOVEM from reg | 274 |
| 9●MOVEM to reg | 276 |
| 10●MOVEP to Dn | 278 |
| 11●MOVEP from Dn | 280 |
| 12●MOVEQ | 282 |
| 13●EXG | 283 |
| 14●SWAP | 284 |
| 15●LEA | 285 |
| 16●PEA | 286 |
| 17●LINK | 287 |
| 18●UNLK | 288 |
| 19●ADD | 289 |
| 20●ADD | 290 |
| 21●ADDA | 291 |
| 22●ADDI | 292 |
| 23●ADDO | 294 |
| 24●ADDX | 295 |
| 25●ADDX | 296 |
| 26●SUB | 297 |
| 27●SUB | 298 |
| 28●SUBA | 299 |
| 29●SUB I | 300 |
| 30●SUBQ | 301 |
| 31●SUBX | 302 |
| 32●SUBX | 303 |
| 33●MULS | 304 |
| 34●MULU | 305 |
| 35●DI VS | 306 |
| 36●DI VU | 308 |
| 37●CMP | 310 |
| 38●CMPA | 311 |

目次

| | |
|----------------------|-----|
| 39●CMP I | 312 |
| 40●CMPPM | 313 |
| 41●CLR | 314 |
| 42●EXT | 315 |
| 43●NEG | 316 |
| 44●NEGX | 317 |
| 45●TST | 318 |
| 46●TAS | 319 |
| 47●AND | 320 |
| 48●AND I | 321 |
| 49●AND II | 322 |
| 50●ANDI to CCR | 323 |
| 51●ANDI to SR [特権命令] | 324 |
| 52●EOR | 325 |
| 53●EOR I | 326 |
| 54●EORI to CCR | 327 |
| 55●EORI to SR [特権命令] | 328 |
| 56●OR | 329 |
| 57●OR | 330 |
| 58●OR I | 331 |
| 59●ORI to CCR | 332 |
| 60●ORI to SR [特権命令] | 333 |
| 61●NOT | 334 |
| 62●ASL | 335 |
| 63●ASL | 335 |
| 64●ASL | 336 |
| 65●ASR | 337 |
| 66●ASR | 338 |
| 67●ASR | 339 |
| 68●LSL | 340 |
| 69●LSL | 341 |
| 70●LSL | 342 |
| 71●LSR | 343 |
| 72●LSR | 344 |
| 73●LSR | 345 |
| 74●ROL | 346 |
| 75●ROL | 347 |
| 76●ROL | 348 |
| 77●ROR | 349 |
| 78●ROR | 350 |
| 79●ROR | 351 |
| 80●ROXL | 352 |
| | 353 |

| | |
|------------------|-----|
| 81●ROXL | 354 |
| 82●ROXL | 355 |
| 83●ROXR | 356 |
| 84●ROXR | 357 |
| 85●ROXR | 358 |
| 86●BTST | 359 |
| 87●BTST | 360 |
| 88●BSET | 361 |
| 89●BSET | 362 |
| 90●BCLR | 363 |
| 91●BCLR | 364 |
| 92●BCHG | 365 |
| 93●BCHG | 366 |
| 94●ABCD | 367 |
| 95●ABCD | 368 |
| 96●SBOD | 369 |
| 97●SBOD | 370 |
| 98●NBOD | 371 |
| 99●Bcc | 372 |
| 100●DBcc | 374 |
| 101●Scc | 376 |
| 102●BRA | 378 |
| 103●BSR | 379 |
| 104●JMP | 380 |
| 105●JSR | 381 |
| 106●RTR | 382 |
| 107●RTS | 383 |
| 108●TRAP | 384 |
| 109●TRAPV | 385 |
| 110●CHK | 386 |
| 111●RTE [特権命令] | 387 |
| 112●RESET [特権命令] | 388 |
| 113●STOP [特権命令] | 389 |
| 114●NOP | 390 |

MOVE

[MOVE data 転送]

MOVE {.B/.W/.L} <ea>, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説

汎用のデータ転送命令で、左側（ソース・オペランド）の内容が右側（ディスティネーション・オペランド）へ転送されますが、コンピュータの転送命令はコピー命令であり、命令実行後、ソース側の内容は元の値を保持し、空（カラ）になるわけではありません。なおデータの移動方向は左から右であり、我々の習慣と同様です。

サイズはバイト、ワード、ロングワードを指定できますが、指定したサイズ以外のビットは変化せず、たとえばバイトを指定すれば、オペランドのビット7～0の1バイトが操作対象となり（上位のすべてのビットは操作の対象外となる）、本命令によって影響を受けることはありません。

データ転送後、フラグは“N”と“Z”が変化し、その値がゼロであったとか、負であったとか、結果としてどのようなデータが転送されたのかを判別できます。

一方、“X”フラグは保持されるので、本命令に先行して実行された演算命令の桁上がり結果が（本命令によって）失われる事がないように配慮されています。

転送先がアドレスレジスタである場合はMOVEAという専用命令があり、汎用のデータ転送とは区別されます。MOVEAはバイトオペランドがサポートされないこと、フラグ変化しない点などでMOVEと異なります（モトローラではアドレスと単なるデータとを区別するために専用のアドレス転送命令を用意したのでしょうか）。

●機械語フォーマット



CCR

X : 変化せず

N : 転送結果が負ならセット(1)、それ以外はリセット(0)

Z : 転送結果がゼロならセット(1)、それ以外はリセット(0)

V : 常にリセット(0)

C : 常にリセット(0)

●アドレッシング・モード

| sou | size | dest | | | | | | | | | | | | | | | |
|------------|------|------|------|--------|--------|---------|------------|-------|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| | | Dn | (An) | (An) + | - (An) | d16(An) | d8(An, IX) | Abs.W | Abs.L | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ |
| Dn | B | 2 | 4 | 2 | 8 | 2 | 8 | 2 | 8 | 4 | 12 | 4 | 14 | 4 | 12 | 6 | 16 |
| | W | 2 | 4 | 2 | 8 | 2 | 8 | 2 | 8 | 4 | 12 | 4 | 14 | 4 | 12 | 6 | 16 |
| | L | 2 | 4 | 2 | 12 | 2 | 12 | 2 | 12 | 4 | 16 | 4 | 18 | 4 | 16 | 6 | 20 |
| An | B | | | | | | | | | | | | | | | | |
| | W | 2 | 4 | 2 | 8 | 2 | 8 | 2 | 8 | 4 | 12 | 4 | 14 | 4 | 12 | 6 | 16 |
| | L | 2 | 4 | 2 | 12 | 2 | 12 | 2 | 12 | 4 | 16 | 4 | 18 | 4 | 16 | 6 | 20 |
| (An) | B | 2 | 8 | 2 | 12 | 2 | 12 | 2 | 12 | 4 | 16 | 4 | 18 | 4 | 16 | 6 | 20 |
| | W | 2 | 8 | 2 | 12 | 2 | 12 | 2 | 12 | 4 | 16 | 4 | 18 | 4 | 16 | 6 | 20 |
| | L | 2 | 12 | 2 | 20 | 2 | 20 | 2 | 20 | 4 | 24 | 4 | 26 | 4 | 24 | 6 | 28 |
| (An) + | B | 2 | 8 | 2 | 12 | 2 | 12 | 2 | 12 | 4 | 16 | 4 | 18 | 4 | 16 | 6 | 20 |
| | W | 2 | 8 | 2 | 12 | 2 | 12 | 2 | 12 | 4 | 16 | 4 | 18 | 4 | 16 | 6 | 20 |
| | L | 2 | 12 | 2 | 20 | 2 | 20 | 2 | 20 | 4 | 24 | 4 | 26 | 4 | 24 | 6 | 28 |
| - (An) | B | 2 | 10 | 2 | 14 | 2 | 14 | 2 | 14 | 4 | 18 | 4 | 20 | 4 | 18 | 6 | 22 |
| | W | 2 | 10 | 2 | 14 | 2 | 14 | 2 | 14 | 4 | 18 | 4 | 20 | 4 | 18 | 6 | 22 |
| | L | 2 | 14 | 2 | 22 | 2 | 22 | 2 | 22 | 4 | 26 | 4 | 28 | 4 | 26 | 6 | 30 |
| d16(An) | B | 4 | 12 | 4 | 16 | 4 | 16 | 4 | 16 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 |
| | W | 4 | 12 | 4 | 16 | 4 | 16 | 4 | 16 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 |
| | L | 4 | 16 | 4 | 24 | 4 | 24 | 4 | 24 | 6 | 28 | 6 | 30 | 6 | 28 | 8 | 32 |
| d8(An, IX) | B | 4 | 14 | 4 | 18 | 4 | 18 | 4 | 18 | 6 | 22 | 6 | 24 | 6 | 22 | 8 | 26 |
| | W | 4 | 14 | 4 | 18 | 4 | 18 | 4 | 18 | 6 | 22 | 6 | 24 | 6 | 22 | 8 | 26 |
| | L | 4 | 18 | 4 | 26 | 4 | 26 | 4 | 26 | 6 | 30 | 6 | 32 | 6 | 30 | 8 | 34 |
| Abs.W | B | 4 | 12 | 4 | 16 | 4 | 16 | 4 | 16 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 |
| | W | 4 | 12 | 4 | 16 | 4 | 16 | 4 | 16 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 |
| | L | 4 | 16 | 4 | 24 | 4 | 24 | 4 | 24 | 6 | 28 | 6 | 30 | 6 | 28 | 8 | 32 |
| Abs.L | B | 6 | 16 | 6 | 20 | 4 | 20 | 6 | 20 | 8 | 24 | 8 | 26 | 8 | 24 | 10 | 28 |
| | W | 6 | 15 | 6 | 20 | 4 | 20 | 6 | 20 | 8 | 24 | 8 | 26 | 8 | 24 | 10 | 28 |
| | L | 6 | 20 | 6 | 28 | 4 | 28 | 6 | 28 | 8 | 34 | 8 | 32 | 10 | 36 | | |
| d16(PC) | B | 4 | 12 | 4 | 16 | 4 | 16 | 4 | 16 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 |
| | W | 4 | 12 | 4 | 16 | 4 | 16 | 4 | 16 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 |
| | L | 4 | 16 | 4 | 24 | 4 | 24 | 4 | 24 | 6 | 28 | 6 | 30 | 6 | 28 | 8 | 32 |
| d8(PC, IX) | B | 4 | 14 | 4 | 18 | 4 | 18 | 4 | 18 | 6 | 22 | 6 | 24 | 6 | 22 | 8 | 26 |
| | W | 4 | 14 | 4 | 18 | 4 | 18 | 4 | 18 | 6 | 22 | 6 | 24 | 6 | 22 | 8 | 26 |
| | L | 4 | 18 | 4 | 26 | 4 | 26 | 4 | 26 | 6 | 30 | 6 | 32 | 6 | 30 | 8 | 34 |
| # Imm | B | 4 | 8 | 4 | 12 | 4 | 12 | 4 | 12 | 6 | 16 | 6 | 18 | 6 | 16 | 8 | 20 |
| | W | 4 | 8 | 4 | 12 | 4 | 12 | 4 | 12 | 6 | 16 | 6 | 18 | 6 | 16 | 8 | 20 |
| | L | 6 | 12 | 6 | 20 | 6 | 20 | 6 | 20 | 8 | 24 | 8 | 26 | 8 | 24 | 10 | 28 |

●サンプル・リスト

MOVE.L D0, (A0)+
 MOVE.B D2, \$4(A2)
 MOVE.L #\\$4888FFAA, \$2000

2 ●MOVEA [MOVE Address アドレスデータの転送]

MOVEA {.W/.L} <ea>, An

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

<ea>の内容を指定したアドレスレジスタへ転送しますが、ワード転送を指定した場合は32ビットに符号拡張され、指定されたアドレスレジスタへ転送されます。

ほとんどのアセンブリの文法ではMOVEの転送先にアドレスレジスタを指定すると、MOVEAと解釈しますが、“アドレス”という概念は大変重要であり、他のデータ転送とは区別した方がバグの発生を抑制できるので、

MOVE.L D0, A0

と記述できても、

MOVEA .L D0, A0

と記述した方がよいと（経験上）思われます。

CCR

X : 変化せず

N : 変化せず

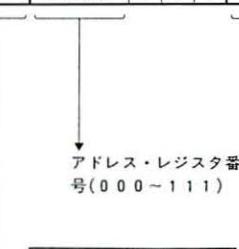
Z : 変化せず

V : 変化せず

C : 変化せず

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|-----|------------|-----------|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | サイズ | ディスティネーション | ソース実効アドレス | | | | | | | | | | | |



| アドレス・レジスタ番号(000~111) | ソース側(すべてのモード) | | | | |
|----------------------|---------------|--------|------------|-------|--------|
| | アドレス | 対応ビット | モード | 対応ビット | モード |
| Dn | 0 0 0 | レジスタ番号 | An | 0 0 1 | レジスタ番号 |
| (An) | 0 1 0 | レジスタ番号 | (An) + | 0 1 1 | レジスタ番号 |
| - (An) | 1 0 0 | レジスタ番号 | d16(An) | 1 0 1 | レジスタ番号 |
| d16(An) | 1 1 0 | レジスタ番号 | d8(An, IX) | 1 1 0 | レジスタ番号 |
| d8(An, IX) | 1 1 1 | レジスタ番号 | Abs.W | 1 1 1 | レジスタ番号 |
| Abs.W | 1 1 1 | レジスタ番号 | Abs.L | 1 1 1 | レジスタ番号 |
| Abs.L | 1 1 1 | レジスタ番号 | d16(PC) | 1 1 1 | レジスタ番号 |
| d16(PC) | 1 1 1 | レジスタ番号 | d8(PC, IX) | 1 1 1 | レジスタ番号 |
| d8(PC, IX) | 1 1 1 | レジスタ番号 | # Imm | 1 1 1 | レジスタ番号 |

●サンプル・リスト

MOVEA.L (A0), A2

MOVEA.L #2000, A4

3 ●MOVE to CCR [CCRへの転送]

MOVE {.W} <ea>, CCR

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

<ea>で指定した内容をCCR (SR : ステータスレジスタの下位8ビット) へ転送するもので、通常は退避してあったフラグの復帰を行うのに使用されますが、5つのフラグを直接操作することも可能です。

転送サイズはワードですがCCRはビット4~0に意味があるため、ソース・オペランドの上位バイトはユーザ状態では意味を持ちませんし、本命令による影響も受けません。

CCR

X : ソース・オペランドの対応ビット (ビット4) の値が反映される

N : ソース・オペランドの対応ビット (ビット3) の値が反映される

Z : ソース・オペランドの対応ビット (ビット2) の値が反映される

V : ソース・オペランドの対応ビット (ビット1) の値が反映される

C : ソース・オペランドの対応ビット (ビット0) の値が反映される

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|--------|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 実効アドレス | | | | | |

| アドレス | 実効アドレス(データ・モード) | | | | |
|------------|-----------------|--------|------------|-------|--------|
| | アドレス | 対応ビット | モード | 対応ビット | モード |
| Dn | 0 0 0 | レジスタ番号 | An | 0 1 0 | レジスタ番号 |
| (An) | 0 1 0 | レジスタ番号 | (An) + | 0 1 1 | レジスタ番号 |
| - (An) | 1 0 0 | レジスタ番号 | d16(An) | 1 0 1 | レジスタ番号 |
| d16(An) | 1 1 0 | レジスタ番号 | d8(An, IX) | 1 1 0 | レジスタ番号 |
| d8(An, IX) | 1 1 1 | レジスタ番号 | Abs.W | 1 1 1 | レジスタ番号 |
| Abs.W | 1 1 1 | レジスタ番号 | Abs.L | 1 1 1 | レジスタ番号 |
| Abs.L | 1 1 1 | レジスタ番号 | d16(PC) | 1 1 1 | レジスタ番号 |
| d16(PC) | 1 1 1 | レジスタ番号 | d8(PC, IX) | 1 1 1 | レジスタ番号 |
| d8(PC, IX) | 1 1 1 | レジスタ番号 | # Imm | 1 1 1 | レジスタ番号 |

●サンプル・リスト

MOVE #0, CCR

●アドレスシング・モード

| sou | size | dest |
|------------|------|------|
| Dn | W | CCR |
| (An) | W | CCR |
| - (An) | W | CCR |
| d16(An) | W | CCR |
| d8(An, IX) | W | CCR |
| Abs.W | W | CCR |
| Abs.L | W | CCR |
| d16(PC) | W | CCR |
| d8(PC, IX) | W | CCR |
| # Imm | W | CCR |

4 ●MOVE to SR [特権命令]

[SRへの転送]

MOVE {W} <ea>, SR

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説 <ea>で指定した内容をステータスレジスタ(SR)へ転送します。サイズはワードであり、ステータスレジスタのすべてのビットが影響を受けます。

68000の動作を決定する重要な命令であり、以下のステータス・ビットを操作することができますが、その意味に関しては必要な説明を参照してください。

- SR**
- T : ソース・オペランドの対応ビット(ピット15、トレース)の値が反映される
 - S : ソース・オペランドの対応ビット(ピット13、スーパーバイザ状態)の値が反映される
 - I₂ : ソース・オペランドの対応ビット(ピット10、割り込みマスク)の値が反映される
 - I₁ : ソース・オペランドの対応ビット(ピット9、割り込みマスク)の値が反映される
 - I₀ : ソース・オペランドの対応ビット(ピット8、割り込みマスク)の値が反映される
 - X : ソース・オペランドの対応ビット(ピット4)の値が反映される
 - N : ソース・オペランドの対応ビット(ピット3)の値が反映される
 - Z : ソース・オペランドの対応ビット(ピット2)の値が反映される
 - V : ソース・オペランドの対応ビット(ピット1)の値が反映される
 - C : ソース・オペランドの対応ビット(ピット0)の値が反映される

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|--------|----|----|----|----|----|---|---|---|---|-----|---|------|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 実効アドレス | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | モード | | レジスタ | | | |

↓
実効アドレス(データ・モード)

| アドレッシング モード | 対応ビット | | | | |
|----------------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| Dn | 0 | 0 | 0 | レジスタ番号 | |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| (An) + | 0 | 1 | 1 | レジスタ番号 | |
| - (An) | 1 | 0 | 0 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 | |
| Abs.W | 1 | 1 | 1 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 1 |
| d16(PC) | 1 | 1 | 1 | 0 | 1 |
| d8(PC,IX) | 1 | 1 | 1 | 0 | 1 |
| # Imm | 1 | 1 | 1 | 1 | 0 |

●サンプル・リスト

MOVE (A0), SR
MOVE \$1000, SR

5 ●MOVE from SR [SRからの転送]

[SRからの転送]

MOVE {W} SR , <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説 ステータスレジスタ(SR)の内容を<ea>で指定した場所へ転送(退避)します。オペラントサイズはワードであり、フラグは変化しません。

68000では特権命令ではありませんが、68010では特権命令となっています。

SR

T : 変化せず
S : 変化せず
I₂ : 変化せず
I₁ : 変化せず
I₀ : 変化せず

X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●機械語フォーマット

| | | | | | | | | | | | | | | | | |
|--------|----|----|----|----|----|---|---|---|---|---|-----|---|------|---|---|--|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 実効アドレス | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | モード | | レジスタ | | | |

実効アドレス(データ・可変モード)

| アドレッシング モード | 対応ビット | | | |
|----------------|-------|---|---|--------|
| | 5 | 4 | 3 | 2 |
| Dn | 0 | 0 | 0 | レジスタ番号 |
| (An) | 0 | 1 | 0 | レジスタ番号 |
| (An) + | 0 | 1 | 1 | レジスタ番号 |
| - (An) | 1 | 0 | 0 | レジスタ番号 |
| d16(An) | 1 | 0 | 1 | レジスタ番号 |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 |
| Abs.W | 1 | 1 | 1 | 0 |
| Abs.L | 1 | 1 | 1 | 1 |

●アドレッシング・モード

| sou | size | dest | | | | | | | | | | | | | | | |
|-----|------|------|------|--------|--------|---------|-----------|-------|-------|----|------|--------|--------|---------|-----------|-------|-------|
| | | Dn | (An) | (An) + | - (An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | Dn | (An) | (An) + | - (An) | d16(An) | d8(An,IX) | Abs.W | Abs.L |
| SR | | B | | | | | | | | B | | | | | | | |
| | | W | 2 | 6 | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 4 | 18 | 4 | 16 | 6 |
| | | L | | | | | | | | | | | | | | | 20 |

●サンプル・リスト

MOVE SR,D0
MOVE SR,\$1000

6 ●MOVE from USP[特権命令] [USPからの転送]

MOVE {.L} USP, An

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

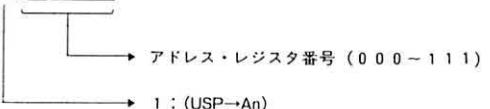
解説

ユーザ・スタック・ポインタ(USP, A7)の内容を指定したアドレスレジスタへ転送します。USPは32ビットですが、扱うサイズはロングワードです。

CCR
X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|----|------|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | dr | レジスタ | | |



●サンプル・リスト

MOVE USP, A5

APPENDIX ●MOVE from CCR

MOVE {.W} CCR, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

CCRの内容を<ea>で指定した場所へ転送しますが、オペレーションはワードであり、上位バイトは\$00、下位バイトがCCRの内容となります。

●この命令は、68000ではサポートされません。

CCR
X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

7 ●MOVE to USP[特権命令] [USPへの転送]

MOVE {.L} An, USP

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

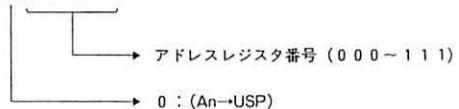
解説

指定したアドレスレジスタの内容をユーザ・スタック・ポインタ(USP, A7)へ転送(設定)します。USPは32ビットですから扱うサイズはロングワードです。

CCR
X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|----|------|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | dr | レジスタ | | |



●サンプル・リスト

MOVE A0, USP

8 ●MOVEM from reg [Move Multiple registers] 複数レジスタの同時転送]

MOVEM {.W/.L} <レジスタ・リスト>, <ea> X N Z V C

解説

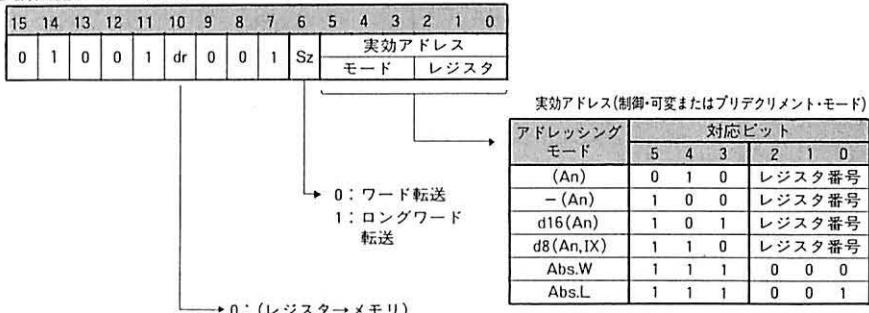
本命令は複数のレジスタ群をメモリ領域へ退避する命令です。

▶制御モードに含まれるアドレス形式のいずれかを用いた場合、転送は指定したメモリロケーションから始まって、アドレスの増加する方向にレジスタ群の内容を転送しますが、転送順は D0～D7, A0～A7 の順序であり、このうちの指定されたレジスタ(マスクフィールドが有効)だけが対象になります。

▶プリデクリメント・アドレスレジスタ間接モード “-(An)” の場合は、“PUSH” ですから、「レジスタ群→メモリ」への一方通行となり、サブルーチンの入り口で必要なレジスタを退避する場合に便利です。

この場合は、指定したメモリロケーション -2 (ワードサイズ) または -4 (ロングワードサイズ) から始まり、アドレスの減少する方向に、アドレスレジスタ A7～A0, データレジスタ D7～D0 の順で転送されます。デクリメントされたアドレスレジスタは更新され、最後に格納したレジスタのアドレスをポイントします。

●機械語フォーマット



<第2ワード: レジスタ・リスト・マスク・フィールド>

オブジェクト・コードの次のワードはレジスタ・マスク・フィールドが続き、1ビットがレジスタ1個を表し、転送順位は、下位ビットに対応するレジスタから割り当てられます(対応するレジスタ番号ビットは“1”)。マスクフィールドの対応するビットがセット(1)されているレジスタが転送対象であり、さらにこのフィールドは転送順も管理します。すなわち、下位ビットから最初に転送されるレジスタがマップされ、上位ビットが最後に転送されるレジスタに対応します。

●制御・可変モードのレジスタ・リスト・マスク・フィールド

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

●プリデクリメント・モードのレジスタ・リスト・マスク・フィールド

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 |

CCR

X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●アドレスシングル・モード

| sou | size | dest | | | | | | | | | | | | | |
|-----|------|-------|-------|---------|-----------|--------|--------|---|---|---|---|---|---|---|--|
| | | (An) | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ | # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ | # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ | # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ | # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ | # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ | # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ # ~ | |
| Dn | B | | | | | | | | | | | | | | |
| | W | 4 4 n | 4 4 n | 6 12 n | 6 14 n | 6 12 n | 8 16 n | | | | | | | | |
| | L | 4 8 n | 4 8 n | 6 8 n | 6 8 n | 6 8 n | 8 16 n | | | | | | | | |
| An | B | | | | | | | | | | | | | | |
| | W | 4 8 n | 4 4 n | 6 12 n | 6 14 n | 6 12 n | 8 16 n | | | | | | | | |
| | L | 4 8 n | 4 8 n | 6 8 n | 6 8 n | 6 8 n | 8 16 n | | | | | | | | |

●サンプル・リスト

MOVEM.L D0-D7/A0-A7,-(SP)

9 ● MOVEM to reg

[Move Multiple registers 複数レジスタとの同時転送]

MOVEM {.W/.L} <ea>, <レジスタ・リスト>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説 メモリ領域に退避しておいたレジスタ群を復帰する命令です。

▶ワードサイズの場合、アドレスレジスタもデータレジスタも、復帰される値は符号拡張され、結果的に32ビットの値がレジスタへロード（転送）されます。

したがって、上位ワードは\$0000または\$FFFFで満たされることになることから、レジスタ群のメモリへの退避／復帰はロングワードで行い、ワードサイズを指定しない方が賢明でしょう。

▶制御モードに含まれるアドレス形式のいずれかを用いた場合、転送は指定したメモリロケーションから始まって、アドレスの増加する方向にレジスタ群の内容を転送しますが、転送順はD0～D7, A0～A7の順序であり、このうちの指定されたレジスタ（マスクフィールドが有効）だけが対象になります。

▶ポストインクリメント・アドレッシング “(An)+” は “POP” であり、「メモリ→レジスタ群」への一方通行です。

この場合は、指定したメモリロケーションから始まって、アドレスの増加する方向に向かってD0～D7, A0～A7の順で転送します。それゆえ、アドレッシングに使用したアドレスレジスタは、命令の実行完了時には最後に転送したアドレス+2（ワードサイズ）または+4（ロングワードサイズ）をポイントします。

▶ MOVE <レジスタ・リスト>, <ea>

と

MOVE <ea>, <レジスタ・リスト>

とはペアで使用するのが前提ですから、このことを理解すれば、どのような転送モードを選択すべきかを理解できるでしょう。たとえばサブルーチンの入り口では “-(An)” でレジスタ群をメモリ領域へ退避し、出口では必然的に “(An)+” で復帰する、というのが通常の選択です。

| CCR | X : 変化せず |
|-----|----------|
| | N : 変化せず |
| | Z : 変化せず |
| | V : 変化せず |
| | C : 変化せず |

●アドレッシング・モード

| sou | size | dest | |
|-----------|------|------------------------------------|------------------------------------|
| | | Dn | An |
| | # | - | - |
| (An) | B | | |
| | W | 4 ₁₂₊ 4 ₈ | 4 ₁₂₊ 4 ₈ |
| | L | 4 ₁₂₊ 4 ₈ | 4 ₁₂₊ 4 ₈ |
| (An) + | B | | |
| | W | 4 ₁₂₊ 4 ₈ | 4 ₁₂₊ 4 ₈ |
| | L | 4 ₁₂₊ 4 ₈ | 4 ₁₂₊ 4 ₈ |
| d16(An) | B | | |
| | W | 6 ₁₆₊ 6 ₈ | 6 ₁₆₊ 6 ₈ |
| | L | 6 ₁₆₊ 6 ₈ | 6 ₁₆₊ 6 ₈ |
| d8(An,IX) | B | | |
| | W | 6 ₁₆₊ 6 ₈ | 6 ₁₆₊ 6 ₈ |
| | L | 6 ₁₆₊ 6 ₈ | 6 ₁₆₊ 6 ₈ |
| Abs.W | B | | |
| | W | 6 ₁₆₊ 6 ₈ | 6 ₁₆₊ 6 ₈ |
| | L | 6 ₁₆₊ 6 ₈ | 6 ₁₆₊ 6 ₈ |
| Abs.L | B | | |
| | W | 8 ₂₀₊ 8 ₈ | 8 ₂₀₊ 8 ₈ |
| | L | 8 ₂₀₊ 8 ₈ | 8 ₂₀₊ 8 ₈ |
| d16(PC) | B | | |
| | W | 6 ₁₆₊ 6 ₈ | 6 ₁₆₊ 6 ₈ |
| | L | 6 ₁₆₊ 6 ₈ | 6 ₁₆₊ 6 ₈ |
| d8(PC,IX) | B | | |
| | W | 6 ₁₆₊ 6 ₈ | 6 ₁₆₊ 6 ₈ |
| | L | 6 ₁₆₊ 6 ₈ | 6 ₁₆₊ 6 ₈ |

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|----|--------|---|-----|---|------|---|
| 0 | 1 | 0 | 0 | 1 | dr | 0 | 0 | 1 | Sz | 実効アドレス | | モード | | レジスタ | |

実効アドレス(制御またはポストインクリメント・モード)

| アドレッシング モード | 対応ビット | | | | |
|----------------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| (An) + | 0 | 1 | 1 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 | |
| AbS.W | 1 | 1 | 1 | 0 | 0 |
| AbS.L | 1 | 1 | 1 | 0 | 0 |
| d16(PC) | 1 | 1 | 1 | 0 | 1 |
| d8(PC,IX) | 1 | 1 | 1 | 0 | 1 |

→ 0: ワード転送
1: ロングワー
ド転送

→ 1: (メモリ→レジスタ)

第2ワード：レジスタ・リスト・マスク・フィールド

オブジェクト・コードの次のワードは、レジスタ・リスト・マスク・フィールドが続きます。1ビットがレジスタ1個を表し、転送順は下位ビットに対応するレジスタから割り当てられます。(対応するビット番号は*1)。

マスクフィールドの対応するビットがセット（1）されているレジスタが転送対象であり、さらにこのフィールドは転送順も管理します。すなわち、下位ビットから最初に転送されるレジスタがマップされ、上位ビットが最後に転送されるレジスタに対応します。

●制御またはポストインクリメント・モードのレジスタ・リスト・マスク・フィールド

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

●サンプル・リスト

MOVEM.L (SP)+, D0-D7/A0-A7

10 ● MOVEP to Dn [MOVE Peripheral data周辺データの転送]

MOVEP {.W/.L} d16 (An), Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

"d16(An)"で指定されるメモリロケーションから始まり、2つずつ増加するような1番地おきのバイト配列との間でデータ転送するのですが、このフィールドで指定するアドレスは、多くの場合I/Oアドレスであることを想定した命令です。

I/Oデバイスとの入/出力は本質的にバイトの属性があり、このため、上位バイトまたは下位バイトに対応するロケーションだけが使用される点で、I/Oデバイスへの書き込みに便利な命令となっています。

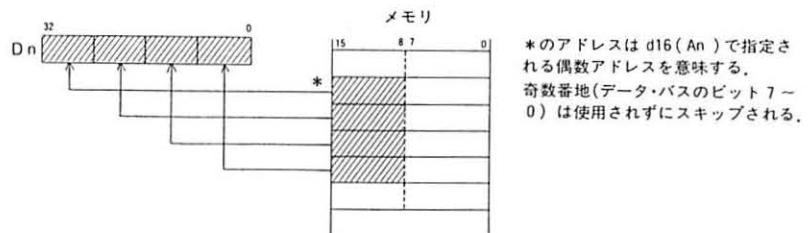
▶データレジスタへは2バイト（ワード）または4バイト（ロングワード）が読み込まれるわけですが、ゼロ番地に近い方がデータレジスタの上位に対応します。

▶"d16(An)"に設定されるロケーションが偶数であれば、そこから偶数アドレスの番地だけをアドレスの増加する方向へ2バイトまたは4バイト読み出し、指定されたデータレジスタへ転送します。したがって、下位データバスは使用されず、奇数アドレスはスキップされます。

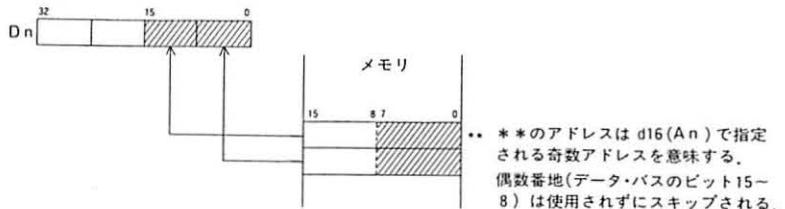
▶"d16(An)"へ設定されるロケーションが奇数であれば、そこから奇数アドレスの番地だけをアドレスの増加する方向へ2バイトまたは4バイト読み出し、指定されたデータレジスタへ転送します。したがって、上位データバスは使用されず、偶数アドレスはスキップされます。

実行の様子

例1：偶数ロケーションとデータ・レジスタ（サイズはロングワード）



例2：奇数ロケーションとデータ・レジスタ（サイズはワード）



CCR

X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●アドレッシング・モード

| sou | size | dest |
|---------|-------------|-------------------------|
| d16(An) | B W L | Dn # 4 16 4 24 |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|---------|-------|---|---|---|----------|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | データレジスタ | OPモード | 0 | 0 | 1 | アドレスレジスタ | | | | | | |

→ d16(An)で指定したアドレスレジスタの番号 (0 0 0 ~ 1 1 1)

→ 1 0 0 : メモリ→レジスタ (ワード転送)
1 0 1 : メモリ→レジスタ (ロングワード転送)

→ Dnで指定したデータレジスタの番号 (0 0 0 ~ 1 1 1)

〈第2ワード：ディスプレースメント・フィールド〉

オブジェクト・コードの第2ワードにはオペランドのロケーションを計算する際に使用されるディスプレースメントが格納されます。

●サンプル・リスト

MOVEP.W \$18(A5), D0

11 ● MOVEP from Dn

[Move Peripheral data
周辺データの転送]

MOVEP {.W/.L} Dn, d16 (An)

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

データレジスタと指定したメモリロケーションから始まり、2つずつ増加するような1番地おきのバイト配列との間でデータ転送するのですが、d16 (An) で指定するアドレスは、多くの場合I/Oアドレスであることを想定した命令です。

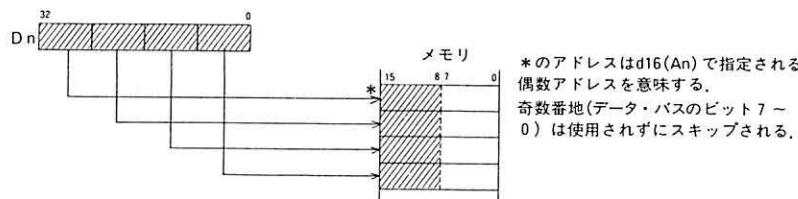
I/Oディバイスとの入/出力は本質的にバイトの属性があり、このため、上位バイトまたは下位バイトに対応するロケーションだけが使用される点で、I/Oディバイスへの書き込みに便利な命令となっています。

▶データレジスタのサイズはワードとロングワードであり、2バイトまたは4バイトがメモリ(I/O)のどこかへ転送されますが、データレジスタの上位バイトはゼロ番地に近いアドレスへ、下位バイトはゼロ番地から遠いアドレスへ、それぞれ転送されます。

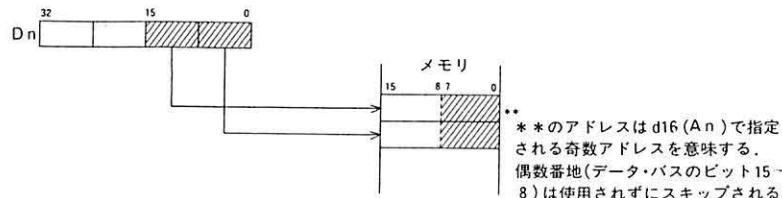
▶ロケーションが偶数であれば偶数アドレスだけが使用され、データバスの上位だけが使用されます。奇数であれば奇数アドレスだけが使用され、データバスの下位半分だけで転送が行われます。

実行の様子

例1：偶数ロケーションとデータレジスタ(サイズはロングワード)



例2：奇数ロケーションとデータレジスタ(サイズはワード)



CCR

X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●アドレスシングル・モード

| sou | size | dest |
|-----|------|---------|
| | | d16(An) |
| Dn | B | # ~ |
| | W | 4 16 |
| | L | 4 24 |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|---------|-------|---|---|---|----------|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | データレジスタ | OPモード | 0 | 0 | 1 | アドレスレジスタ | | | | | | |

→ d16(An) で使用するアドレスレジスタの番号
(0 0 0 ~ 1 1 1)
→ 1 1 0 : レジスタ → メモリ (ワード転送)
1 1 1 : レジスタ → メモリ (ロングワード転送)
→ Dn で指定したデータレジスタの番号 (0 0 0 ~ 1 1 1)

〈第2ワード：ディスプレースメント・フィールド〉

オブジェクト・コードの第2ワードには、オペランドのロケーションを計算する際に使用するディスプレースメントが格納されます。

●サンプル・リスト

MOVEP.W D0, \$18(A5)

12 ●MOVEQ

[Move Quick クイック転送]

MOVEQ {.L} # <data>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説

1バイトのイミディエイトデータ（即値）をデータレジスタへ転送する命令ですが、データレジスタは32ビットが使用され、指定した1バイトのデータは32ビットに符号拡張されてデータレジスタへ転送されます。
通常の転送命令よりはオブジェクトコードが短く高速に実行できる命令です。

CCR

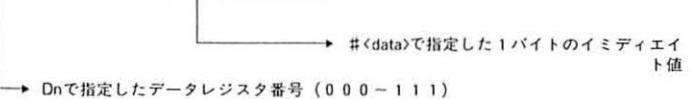
X : 変化せず
N : 転送結果が負ならセット(1), それ以外はリセット(0)
Z : 転送結果がゼロならセット(1), それ以外はリセット(0)
V : 常にリセット(0)
C : 常にリセット(0)

●アドレスシング・モード

| sou | size | dest |
|-----|------|------|
| Dn | # ~ | |
| B | | |
| W | | |
| L | 2 4 | |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|----|-----|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 1 | 1 | レジスタ | 0 | データ | | | | | | | | | |



●サンプル・リスト

MOVEQ #\$4A,D0

13 ●EXG

[Exchang registers レジスタ交換]

EXG {.L} <reg>, <reg>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

2つのレジスタの内容を交換しますが、32ビット全部が操作されます。
次の3通りの方法を指定できます。

- データレジスタ同士の交換。
- アドレスレジスタ同士の交換。
- データレジスタとアドレスレジスタとの交換。

CCR

X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●アドレスシング・モード

| sou | size | dest |
|-----|---------|---------|
| Dn | Dn An | # ~ # ~ |
| B | | |
| W | | |
| L | 2 6 2 6 | |
| An | | |
| B | | |
| W | | |
| L | 2 6 2 6 | |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|---------|----|-------|---------|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 0 | 0 | ソースレジスタ | 1 | OPモード | データレジスタ | | | | | | | | |

| 交換の内容 | 対応ビット |
|------------------|---------------------|
| データレジスタ同士 | 7 6 5 4 3 0 1 0 0 0 |
| アドレスレジスタ同士 | 0 1 0 0 1 |
| データレジスタとアドレスレジスタ | 1 0 0 0 1 |

ソースオペランド (第1オペランド) で指定したデータレジスタあるいはアドレスレジスタ番号 (000~111)。
データレジスタとアドレスレジスタを交換する場合は、常にデータレジスタの番号 (000~111) がマップされる。

●サンプル・リスト

EXG D0,D1

14 SWAP

[Swap register halves レジスタ半分交換]

SWAP [.W] Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説 指定したデータレジスタの上位ワードと下位ワードを交換します。

CCR X : 変化せず

N : 演算の結果、データの最上位ビット (MSB) が "1" ならセット(1), それ以外はリセット(0)

Z : 演算の結果がゼロならセット(1), それ以外はリセット(0)

V : 常にリセット(0)

C : 常にリセット(0)

●アドレスシング・モード

| sou | size | dest |
|-----|------|------|
| Dn | # | - |
| B | | |
| W | 2 | 4 |
| L | | |

* ソースオペランド
は存在しない。

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | レジスタ | |

→ Dnで指定したデータレジスタ番号
(000~111)

●サンプル・リスト

SWAP D0

15 LEA

[Load Effective Address 実効アドレスのロード]

LEA [.L] <ea>, An

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説 <ea> で指定した実効アドレスをAnへ転送します。

実効アドレスはロングワードであり、Anの32ビットすべてが影響を受けます。

CCR X : 変化せず

N : 変化せず

Z : 変化せず

V : 変化せず

C : 変化せず

●アドレスシング・モード

| SOU | SIZE | dest |
|-----------|------|------|
| (An) | # | - |
| B | | |
| W | | |
| L | 2 | 4 |
| d16(An) | | |
| B | | |
| W | | |
| L | 4 | 8 |
| d8(An,IX) | | |
| B | | |
| W | | |
| L | 4 | 12 |
| Abs.W | | |
| B | | |
| W | | |
| L | 4 | 8 |
| Abs.L | | |
| B | | |
| W | | |
| L | 6 | 12 |
| d16(PC) | | |
| B | | |
| W | | |
| L | 4 | 8 |
| d8(PC,IX) | | |
| B | | |
| W | | |
| L | 4 | 12 |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|----|---|---|--------|---|-----|------|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 0 | 0 | レジスタ | 1 | 1 | 1 | 実効アドレス | | モード | レジスタ | | | | |

↓
Anで指定したアドレスレジスタ
の番号(000~111)

↓
実効アドレス (制御モード)

| アドレスシング モード | 対応ビット | | | | |
|----------------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 | |
| Abs.W | 1 | 1 | 1 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 |
| d16(PC) | 1 | 1 | 1 | 0 | 1 |
| d8(PC,IX) | 1 | 1 | 1 | 0 | 1 |

●サンプル・リスト

LEA 20(A6), A0

LEA (A2), A5

LEA LABEL, A3

↑LABELは記号番地を意味する

16 ● PEA

[Push Effective Address 実効アドレスのスタック]

PEA {.L} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説 <ea>で指定した実効アドレスをシステムスタックへプッシュ(転送)します。

システム・スタック・ポインタ(SP)が暗黙のうちに使用されます。

CCR

- X : 変化せず
- N : 変化せず
- Z : 変化せず
- V : 変化せず
- C : 変化せず

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|-----|--------|------|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | モード | 実効アドレス | レジスタ | | | |

↓

実効アドレス(制御モード)

| アドレッシングモード | 対応ビット | | | | |
|------------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 | |
| Abs.W | 1 | 1 | 1 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 |
| d16(PC) | 1 | 1 | 1 | 0 | 1 |
| d8(PC,IX) | 1 | 1 | 1 | 0 | 1 |

| sou | size | dest |
|-----------|------|------|
| # | - | - |
| (An) | B | |
| | W | |
| | L 2 | 14 |
| d16(An) | B | |
| | W | |
| | L 4 | 18 |
| d8(An,IX) | B | |
| | W | |
| | L 4 | 22 |
| Abs.W | B | |
| | W | |
| | L 4 | 18 |
| Abs.L | B | |
| | W | |
| | L 6 | 22 |
| d16(PC) | B | |
| | W | |
| | L 4 | 18 |
| d8(PC,IX) | B | |
| | W | |
| | L 4 | 22 |

destは存在しない

●サンプル・リスト

PEA 20(A6)
PEA LABEL

↑LABELは記号番地を意味する

17 ● LINK

[Link and allocate スタック・リンクと割付け]

LINK An, # <エリア長>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説 ネストされたサブルーチン用にローカルエリアを確保します。

メインからの引数をサブルーチンへ渡し、サブルーチン内ではメイン側へ影響を与えることなく作業エリアを確保できるなど、きわめて有用な命令です。

<エリア長>の値はシステムスタックを破壊しないためにマイナスを指定する必要があり、-32768~0の範囲を指定することになります。

- ① Anで指定したアドレスレジスタをシステムスタックへプッシュする(この時点でSPは4だけ減少している)。
- ② 更新されたSPの内容をAnへ転送する。
- ③ SPに<エリア長>を加算する。

CCR

- X : 変化せず
- N : 変化せず
- Z : 変化せず
- V : 変化せず
- C : 変化せず

●アドレッシング・モード

| sou | size | dest |
|-----|--------|------|
| # | (エリア長) | - |
| An | | |

sizeは存在しない

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|------|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | レジスタ |

→ Anで指定したアドレスレジスタ番号
(0 0 0 ~ 1 1 1)

<第2ワード：エリア長>

エリア長フィールドで指定した値(2バイトの符号付整数)は、オブジェクトコードの第2ワードに格納されます。

| | |
|-------|---|
| 15 | 0 |
| エリニア長 | |

●サンプル・リスト

LINK A6, #-80

20 ● ADD

[Add binary 2進加算]

ADD {.B/.W/.L} <ea>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドとソース・オペランドを加算し、結果をディスティネーションへ格納します。ソースオペランドの<ea>にAn (アドレスレジスタ直接形式)を指定した場合、オペレーションサイズはワードまたはロングワードであり、バイトサイズは許されません。

CCR

X : Cビットと同じ値
N : 演算結果が負ならセット(1), それ以外はリセット(0)
Z : 演算結果がゼロならセット(1), それ以外はリセット(0)
V : オーバフローが発生すればセット(1), それ以外はリセット(0)
C : 柄上がり(キャリ)が発生すればセット(1), それ以外はリセット(0)

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|-------|--------|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 0 | 1 | レジスタ | OPモード | 実効アドレス | | | | | | | | | |

| サイズ | 対応ビット |
|--------|-------|
| 8 | 8 7 6 |
| バイト | 0 0 0 |
| ワード | 0 0 1 |
| ロングワード | 0 1 0 |

指定したデータレジスタ番号

ソース実効アドレス(すべてのモード)

* バイトサイズ不可
(注) ADDIを使用する

●サンプル・リスト

ADD (A2)+, D0

21 ● ADDA

[Add Address アドレス・データの加算]

ADDA {.W/.L} <ea>, An

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

Anとソース・オペランドを加算し、結果をディスティネーションへ格納します。サイズがワードの場合、<ea>の内容は32ビットに符号拡張され、アドレスレジスタの全32ビットが演算の対象になります。

CCR

X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|-------|--------|---|---|---|-----|---|------|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 0 | 1 | レジスタ | OPモード | 実効アドレス | | | | モード | | レジスタ | | | |

| サイズ | 対応ビット |
|--------|-------|
| 8 | 8 7 6 |
| ワード | 0 1 1 |
| ロングワード | 1 1 1 |

指定したアドレスレジスタの番号
(000~111)

実効アドレス(すべてのモード)

| アドレッシングモード | 対応ビット |
|------------|---------------|
| Dn | 5 4 3 2 1 0 |
| An | 0 0 0 0 0 1 |
| (An) | 0 1 0 0 1 0 |
| (An) + | 0 1 1 0 1 1 |
| - (An) | 1 0 0 1 0 0 |
| d16(An) | 1 0 1 1 0 1 |
| d8(An,IX) | 1 1 0 0 0 0 |
| Abs.W | 1 1 1 0 0 0 |
| Abs.L | 1 1 1 0 0 1 |
| d16(PC) | 1 1 1 0 1 0 |
| d8(PC,IX) | 1 1 1 0 1 1 |
| #Imm | 1 1 1 1 0 0 |

●サンプル・リスト

ADDA.L A0, A6
ADDA.L (A2), A5

| sou | size | dest |
|-----------|------|------|
| Dn | | # ~ |
| An | | |
| (An) | | |
| - (An) | | |
| d16(An) | | |
| d8(An,IX) | | |
| Abs.W | | |
| Abs.L | | |
| d16(PC) | | |
| d8(PC,IX) | | |
| #Imm | | |

22 ● ADDI

[Add Immediate イミディエイト・データの加算]

| | | | | | |
|--------------------------------|---|---|---|---|---|
| ADDI {.B/.W/.L} # <data>, <ea> | X | N | Z | V | C |
| | * | * | * | * | * |

解説 ディスティネーション・オペランドとイミディエイト値を加算し、結果をディスティネーションへ格納します。

イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。

CCR X : Cビットと同じ値

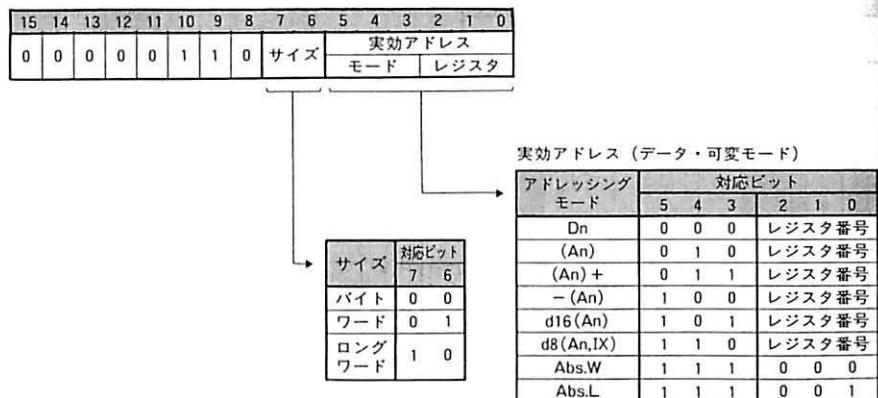
N : 演算結果が負ならセット(1), それ以外はリセット(0)

Z : 演算結果がゼロならセット(1), それ以外はリセット(0)

V : オーバフローが発生すればセット(1), それ以外はリセット(0)

C : 柄上がり (キャリ) が発生すればセット(1), それ以外はリセット(0)

● 機械語フォーマット



<第2ワード：サイズがバイトおよびワードの時>

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | | | | | | | | |

→ サイズがバイトの時に使用され、イミディエイト値の8ビットが格納される。

<第2および第3ワード：ロングワードサイズを指定した時>

| | | | | | | | | | | | | | | | |
|--------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 指定したイミディエイト値の上位ワード | | | | | | | | | | | | | | | |
| 指定したイミディエイト値の下位ワード | | | | | | | | | | | | | | | |

● アドレッシング・モード

| sou | size | dest | | | | | | | | | |
|-------|------|------|------|--------|--------|---------|-----------|-------|-------|-----|-----|
| | | Dn | (An) | (An) + | - (An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # ~ | # ~ |
| # Imm | B | 4 | 8 | 4 | 16 | 4 | 16 | 4 | 18 | 6 | 20 |
| | W | 4 | 8 | 4 | 16 | 4 | 16 | 4 | 18 | 6 | 20 |
| | L | 6 | 16 | 6 | 28 | 6 | 28 | 6 | 30 | 8 | 32 |
| | | | | | | | | | | | |

● サンプル・リスト

ADDI.L #\$EA041,D7

23 ● ADDQ

[Add Quick クイック加算]

ADDQ {.B/.W/.L} # <data>, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドとイミディエイト値を加算し、結果をディスティネーションのロケーションである <ea> へ格納します。
イミディエイト値として指定できる範囲は 1 ~ 8 までに限定されます。
<ea> に An (アドレスレジスタ直接形式) を指定した場合、以下の点に注意が必要です。

- オペレーションサイズはワードまたはロングワードであり、バイトサイズは許されない。
- オペレーションサイズに関係なく、ディスティネーションに指定したアドレスレジスタの全32ビットが使用される。
- CCRは本命令の影響を受けない。

CCR

X : Cビットと同じ値

N : 演算結果が負なら
セット(1), それ以外はリセット(0)

Z : 演算結果がゼロなら
セット(1), それ以外はリセット(0)

V : オーバフローが発生すればセット(1), それ以外はリセット(0)

C : 桁上がり (キャリ) が発生すればセット(1), それ以外はリセット(0)

○アドレッシング・モード

| sou | size | dest | | | | | | | | | | | | |
|-------|------|------|----|------|-------|-------|---------|-----------|-------|-------|----|----|----|--------|
| | | Dn | An | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # | ~ | | |
| # Imm | | # | ~ | # | ~ | # | ~ | # | ~ | # | ~ | # | ~ | |
| B | 2 | 4 | | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 4 | 18 | |
| W | 2 | 4 | 2 | 4 | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 4 | 18 |
| L | 2 | 8 | 2 | 8 | 2 | 20 | 2 | 20 | 2 | 22 | 4 | 24 | 4 | 26 |
| | | | | | | | | | | | | | | 6 20 |

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-----|----|-----|---|---|---|--------|---|---|---|---|---|
| 0 | 1 | 0 | 1 | データ | 0 | サイズ | | | | 実効アドレス | | | | | |

実効アドレス (可変モード)

| アドレッシングモード | 対応ビット |
|------------|-------------|
| Dn | 5 4 3 2 1 0 |
| An* | 0 0 0 |
| (An) | 0 0 1 |
| (An) + | 0 1 0 |
| -(An) | 0 1 1 |
| d16(An) | 1 0 0 |
| d8(An,IX) | 1 0 1 |
| Abs.W | 1 1 0 |
| Abs.L | 1 1 1 |

*バイトサイズ不可

○サンプル・リスト

ADDQ.L #8,A2
ADDQ.W #4,D0

24 ● ADDX

[Add Extended 拡張加算]

ADDX {.B/.W/.L} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドにソース・オペランドとXビットを加算し、結果をディスティネーションで指定したDnへ格納します。
CCRのZビットは、通常、演算を実行する前にプログラムでセットされるため、このビットを使用して、多倍精度演算を終了したときの演算結果がゼロであるかをテストできます。

CCR

X : Cビットと同じ値

N : 演算結果が負ならセット(1), それ以外はリセット(0)

Z : 演算結果がゼロでなければリセット(0), それ以外は変化せず

V : オーバフローが発生すればセット(1), それ以外はリセット(0)

C : 桁上がり (キャリ) が発生すればセット(1), それ以外はリセット(0)

○アドレッシング・モード

| sou | size | dest |
|-----|-------|------|
| Dn | # ~ | Dn |
| B | 2 4 | |
| W | 2 4 | |
| L | 2 8 | |

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----------------|----|-----|---|---|-----|---------|---|---|---|---|---|
| 1 | 1 | 0 | 1 | ディスティネーションレジスタ | 1 | サイズ | 0 | 0 | R/M | ソースレジスタ | | | | | |

ソース・オペランドで指定したデータレジスタ番号 (0 0 0 ~ 1 1 1)

0 : データレジスタとデータレジスタとの操作

| サイズ | 対応ビット |
|--------|-------|
| 7 | 6 |
| バイト | 0 0 |
| ワード | 0 1 |
| ロングワード | 1 0 |

ディスティネーション・オペランドで指定したデータレジスタ番号 (0 0 0 ~ 1 1 1)

○サンプル・リスト

ADDX.L D4,D7

25 ADDX

[Add Extended 拡張加算]

ADDX {.B/.W/.L} -(An), -(An)

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドにソース・オペランドとXビットを加算し、結果をディスティネーションへ格納します。

CCRのZビットは、通常、演算を実行する前にプログラムでセットされるため、このビットを使用して、多倍精度演算を終了したときの演算結果がゼロであるかをテストできます。

CCR

X : Cビットと同じ値

N : 演算結果が負ならセット(1), それ以外はリセット(0)

Z : 演算結果がゼロでなければセット(0), それ以外は変化せず

V : オーバフローが発生すればセット(1), それ以外はリセット(0)

C : 柄上がり(キャリ)が発生すればセット(1), それ以外はリセット(0)

○アドレスシング・モード

| sou | size | dest |
|--------|--------|--------|
| - (An) | # - | - (An) |
| - (An) | B 2 18 | |
| - (An) | W 2 18 | |
| - (An) | L 2 30 | |

○機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----------------|----|-----|---|---|-----|---------|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 0 | 1 | ディスティネーションレジスタ | 1 | サイズ | 0 | 0 | R/M | ソースレジスタ | | | | | |

→ ソース・オペランドで指定したアドレスレジスタ番号 (000~111)
→ 1: メモリとメモリとの操作

| サイズ | 対応ビット |
|--------|-------|
| 7 | 6 |
| バイト | 0 0 |
| ワード | 0 1 |
| ロングワード | 1 0 |

ディスティネーション・オペランドで指定したアドレスレジスタ番号 (000~111)

○サンプル・リスト

ADDX.W -(A1), -(A2)

26 SUB

[Subtract binary 2進減算]

SUB {.B/.W/.L} Dn, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドからソース・オペランドを減算し、結果をディスティネーションへ格納します。

CCR

X : Cビットと同じ値

N : 演算結果が負ならセット(1), それ以外はリセット(0)

Z : 演算結果がゼロならセット(1), それ以外はリセット(0)

V : オーバフローが発生すればセット(1), それ以外はリセット(0)

C : 柄上がり(キャリ)が発生すればセット(1), それ以外はリセット(0)

○アドレスシング・モード

| sou | size | dest | | | | | | | | | | |
|-----|------|------|--------|-------|---------|-------|-------|-------|------|-----|-----|-----|
| | | (An) | (An) + | -(An) | d16(An) | 吸(An) | Abs.W | Abs.L | # - | # ~ | # - | # ~ |
| Dn | B | 2 12 | 2 12 | 2 12 | 2 14 | 4 16 | 4 18 | 4 16 | 6 20 | | | |
| | W | 2 12 | 2 12 | 2 12 | 2 14 | 4 16 | 4 18 | 4 16 | 6 20 | | | |
| | L | 2 20 | 2 20 | 2 22 | 4 24 | 4 26 | 4 24 | 4 24 | 6 28 | | | |

○機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|-------|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 1 | レジスタ | OPモード | | | | | | | | | | |

| 実効アドレス(メモリ・可変モード) | | | | | | | | | | | | | | | |
|-------------------|---|-------|---|---|---|---|---|--------|--|--|--|--|--|--|--|
| アドレスシングモード | | 対応ビット | | | | | | | | | | | | | |
| | | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | |
| (An) | 0 | 1 | 0 | | | | | レジスタ番号 | | | | | | | |
| (An) + | 0 | 1 | 1 | | | | | レジスタ番号 | | | | | | | |
| -(An) | 1 | 0 | 0 | | | | | レジスタ番号 | | | | | | | |
| d16(An) | 1 | 0 | 1 | | | | | レジスタ番号 | | | | | | | |
| d8(An,IX) | 1 | 1 | 0 | | | | | レジスタ番号 | | | | | | | |
| Abs.W | 1 | 1 | 1 | 0 | | | | 0 0 0 | | | | | | | |
| Abs.L | 1 | 1 | 1 | 1 | 0 | | | 0 0 1 | | | | | | | |

→ 指定したデータレジスタ番号 (000~111)

○サンプル・リスト

SUB.W D2, (A1)
SUB.L D3, (A0)+

27 ●SUB

[Subtract binary 2進減算]

SUB {.B/.W/.L} <ea>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドからソース・オペランドを減算し、結果をディスティネーションへ格納します。
ソース・オペランドの<ea>にAn(アドレスレジスタ直接)を指定する場合、ワードまたはロングワードのオペレーションサイズがサポートされ、バイトサイズを指定できません。

CCR

X : Cビットと同じ値
N : 演算結果が負ならセット(1), それ以外はリセット(0)
Z : 演算結果がゼロならセット(1), それ以外はリセット(0)
V : オーバフローが発生すればセット(1), それ以外はリセット(0)
C : 桁下がり(ボロー)が発生すればセット(1), それ以外はリセット(0)

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|-------|--------|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 1 | レジスタ | OPモード | 実効アドレス | | | | | | | | | |

実効アドレス(すべてのモード)

| アドレッシングモード | 対応ビット |
|------------|--------------|
| サイズ | 5 4 3 2 1 0 |
| Dn | 0 0 0 レジスタ番号 |
| An* | 0 0 1 レジスタ番号 |
| (An) | 0 1 0 レジスタ番号 |
| (An) + | 0 1 1 レジスタ番号 |
| - (An) | 1 0 0 レジスタ番号 |
| d16(An) | 1 0 1 レジスタ番号 |
| d8(An,IX) | 1 1 1 レジスタ番号 |
| Abs.W | 1 1 1 0 0 0 |
| Abs.L | 1 1 1 0 0 1 |
| d16(PC) | 1 1 1 0 1 0 |
| d8(PC,IX) | 1 1 1 0 1 1 |
| #Imm* | 1 1 1 1 0 0 |

*バイトサイズ不可
(注) SUBIを使用する

指定したデータレジスタ番号
(000~111)

●アドレッシング・モード

| sou | size | dest |
|-----------|--------|--------|
| Dn | B 2 4 | Dn # - |
| | W 2 4 | |
| | L 2 8 | |
| An | B | |
| | W 2 4 | |
| | L 2 8 | |
| (An) | B 2 8 | |
| | W 2 8 | |
| | L 2 14 | |
| (An) + | B 2 8 | |
| | W 2 8 | |
| | L 2 14 | |
| - (An) | B 2 10 | |
| | W 2 10 | |
| | L 2 16 | |
| d16(An) | B 4 12 | |
| | W 4 12 | |
| | L 4 18 | |
| d8(An,IX) | B 4 14 | |
| | W 4 14 | |
| | L 4 20 | |
| Abs.W | B 4 12 | |
| | W 4 12 | |
| | L 4 18 | |
| Abs.L | B 6 16 | |
| | W 6 16 | |
| | L 6 22 | |
| d16(PC) | B 4 12 | |
| | W 4 12 | |
| | L 4 18 | |
| d8(PC,IX) | B 4 14 | |
| | W 4 14 | |
| | L 4 20 | |
| #Imm | B 4 8 | |
| | W 4 8 | |
| | L 6 14 | |

28 ●SUBA

[Subtract Address アドレス・データの減算]

SUBA {.W/.L} <ea>, An

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

指定したアドレスレジスタからソース・オペランドを減算し、結果をアドレスレジスタへ格納します。
オペレーションサイズは、ワードまたはロングワードがサポートされ、バイトサイズは許されません。さらに、ワードのオペレーションでは、ソース・オペランドは32ビットに符号拡張され、アドレスレジスタの全32ビットが演算の対象となります。

CCR

X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|-------|--------|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 1 | レジスタ | OPモード | 実効アドレス | | | | | | | | | |

実効アドレス(すべてのモード)

| アドレッシングモード | 対応ビット |
|------------|--------------|
| サイズ | 5 4 3 2 1 0 |
| Dn | 0 0 0 レジスタ番号 |
| An | 0 0 1 レジスタ番号 |
| (An) | 0 1 0 レジスタ番号 |
| (An) + | 0 1 1 レジスタ番号 |
| - (An) | 1 0 0 レジスタ番号 |
| d16(An) | 1 0 1 レジスタ番号 |
| d8(An,IX) | 1 1 1 レジスタ番号 |
| Abs.W | 1 1 1 0 0 0 |
| Abs.L | 1 1 1 0 0 1 |
| d16(PC) | 1 1 1 0 1 0 |
| d8(PC,IX) | 1 1 1 0 1 1 |
| #Imm | 1 1 1 1 0 0 |

ディスティネーション・オペランドで指定したアドレスレジスタ番号(000~111)

●サンプル・リスト

SUBA.L A2,A1
SUBA.L \$2000,A4
SUBA.L #\$FAD00,A5

| sou | size | dest |
|-----------|------|--------|
| Dn | B | An # - |
| | W | 2 8 |
| | L | 2 8 |
| An | B | |
| | W | 2 8 |
| | L | 2 8 |
| (An) | B | |
| | W | 2 12 |
| | L | 2 14 |
| (An) + | B | |
| | W | 2 12 |
| | L | 2 14 |
| - (An) | B | |
| | W | 2 14 |
| | L | 2 16 |
| d16(An) | B | |
| | W | 4 16 |
| | L | 4 18 |
| d8(An,IX) | B | |
| | W | 4 18 |
| | L | 4 20 |
| Abs.W | B | |
| | W | 4 16 |
| | L | 4 18 |
| Abs.L | B | |
| | W | 6 20 |
| | L | 6 22 |
| d16(PC) | B | |
| | W | 4 16 |
| | L | 4 18 |
| d8(PC,IX) | B | |
| | W | 4 18 |
| | L | 4 20 |
| #Imm | B | |
| | W | 4 12 |
| | L | 6 14 |

29 SUBI

[Subtract Immediate イミディエイト・データとの減算]

SUBI {.B/.W/.L} # <data>, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドからイミディエイト値を減算し、結果をディスティネーションへ格納します。

イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。

CCR X : Cビットと同じ値

○アドレスシング・モード

N : 演算結果が負ならセット(1),

それ以外はリセット(0)

Z : 演算結果がゼロならセット

(1), それ以外はリセット(0)

V : オーバフローが発生すれば

セット(1), それ以外はリセット(0)

C : 桁下がり（ボロー）が発生すればセット(1), それ以外はリセット(0)

○機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|-----|--------|-----|------|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | サイズ | 実効アドレス | モード | レジスタ | | | | |

実効アドレス（データ・可変モード）

| サイズ | 対応ビット |
|--------|-------|
| 7 | 6 |
| バイト | 0 0 |
| ワード | 0 1 |
| ロングワード | 1 0 |

| アドレスシングモード | 対応ビット | 5 | 4 | 3 | 2 | 1 | 0 |
|------------|-------|--------|---|---|---|---|---|
| Dn | 0 0 0 | レジスタ番号 | | | | | |
| (An) | 0 1 0 | レジスタ番号 | | | | | |
| (An) + | 0 1 1 | レジスタ番号 | | | | | |
| - (An) | 1 0 0 | レジスタ番号 | | | | | |
| d16(An) | 1 0 1 | レジスタ番号 | | | | | |
| d8(An,IX) | 1 1 0 | レジスタ番号 | | | | | |
| Abs.W | 1 1 1 | 0 0 0 | | | | | |
| Abs.L | 1 1 1 | 0 0 1 | | | | | |

（第2ワード、バイトまたはワードサイズ時）

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | | | | | | | | | |

→ イミディエイト値の下位バイト
(サイズがバイトの時)

→ イミディエイト値のワード
(サイズがワードの時)

（第2、第3ワード、ロングワード時）

| | |
|--------------------|---|
| 15 | 0 |
| 指定したイミディエイト値の上位ワード | |
| 指定したイミディエイト値の下位ワード | |

○サンプル・リスト

SUBI.W #10,D0

SUBI.L #\$FF00AF,(A2)

30 SUBQ

[Subtract Quick クイック減算]

SUBQ {.B/.W/.L} # <data>, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドからイミディエイト値を減算し、結果をディスティネーションへ格納します。

指定できるイミディエイト値は、1～8までの範囲に限定されます。

ディスティネーションの<ea>にAn（アドレスレジスタ直接形式）を指定した場合、次の点に注意する必要があります。

- オペレーションサイズは、ワードまたはロングワードがサポートされ、バイトサイズを指定できない。
- オペレーションサイズがワードサイズである場合、演算実行前に32ビットに符号拡張され、アドレスレジスタの全32ビットが使用される。
- CCRは影響を受けない。

CCR

X : Cビットと同じ値

○アドレスシング・モード

N : 演算結果が負ならセット(1), それ以外はリセット(0)

Z : 演算結果がゼロならセット(1), それ以外はリセット(0)

V : オーバフローが発生すればセット(1), それ以外はリセット(0)

C : 桁下がり（ボロー）が発生すればセット(1), それ以外はリセット(0)

○機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|-----|----|-----|--------|-----|------|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 0 | 1 | データ | 1 | サイズ | 実効アドレス | モード | レジスタ | | | | | | |

実効アドレス（可変モード）

| アドレスシングモード | 対応ビット | 5 | 4 | 3 | 2 | 1 | 0 |
|------------|-------|--------|---|---|---|---|---|
| Dn | 0 0 0 | レジスタ番号 | | | | | |
| An* | 0 0 1 | レジスタ番号 | | | | | |
| (An) | 0 1 0 | レジスタ番号 | | | | | |
| (An) + | 0 1 1 | レジスタ番号 | | | | | |
| - (An) | 1 0 0 | レジスタ番号 | | | | | |
| d16(An) | 1 0 1 | レジスタ番号 | | | | | |
| d8(An,IX) | 1 1 0 | レジスタ番号 | | | | | |
| Abs.W | 1 1 1 | 0 0 0 | | | | | |
| Abs.L | 1 1 1 | 0 0 1 | | | | | |

イミディエイトデータ 0 0 0 は 8 を意味し,
0 0 1 ～ 1 1 1 は 1 ～ 7 に対応する。

* バイトサイズ不可

○サンプル・リスト

SUBQ.W #8,4(A0,D2,L)

SUBQ.L #5,D0

31 ●SUBX

[Subtract with Extend 拡張付き減算]

SUBX {.B/.W/.L} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドからソース・オペランドとXビットを減算し、結果をディスティネーションへ格納します。

CCRのZビットは、通常、演算を実行する前にプログラムでセットされるため、このビットを使用して、多倍精度演算を終了したときの演算結果がゼロであるかをテストできます。

CCR X : Cビットと同じ値

N : 演算結果が負ならセット(1), それ以外はリセット(0)

Z : 演算結果がゼロでなければリセット(0), それ以外は変化せず

V : オーバフローが発生すればセット(1), それ以外はリセット(0)

C : 衝下がり（ボロー）が発生すればセット(1), それ以外はリセット(0)

●アドレスシング・モード

| sou | size | dest |
|-----|------|------|
| Dn | # | Dn |
| | - | - |
| B | 2 | 4 |
| W | 2 | 4 |
| L | 2 | 8 |

32 ●SUBX

[Subtract with Extend 拡張付き減算]

SUBX {.B/.W/.L} -(An), -(An)

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドからソース・オペランドとXビットを減算し、結果をディスティネーションへ格納します。

CCRのZビットは、通常、演算を実行する前にプログラムでセットされるため、このビットを使用して、多倍精度演算を終了したときの演算結果がゼロであるかをテストできます。

CCR X : Cビットと同じ値

N : 演算結果が負ならセット(1), それ以外はリセット(0)

Z : 演算結果がゼロでなければリセット(0), それ以外は変化せず

V : オーバフローが発生すればセット(1), それ以外はリセット(0)

C : 衝下がり（ボロー）が発生すればセット(1), それ以外はリセット(0)

●アドレスシング・モード

| sou | size | dest |
|-------|------|------|
| -(An) | # | - |
| B | 2 | 8 |
| W | 2 | 8 |
| L | 2 | 30 |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----------------|----|-----|---|---|-----|---------|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 1 | ディスティネーションレジスタ | 1 | サイズ | 0 | 0 | R/M | ソースレジスタ | | | | | |

ソース・オペランドで指定したデータレジスタ番号 (000~111)

0 : データレジスタとデータレジスタの操作

| サイズ | 対応ビット |
|--------|-------|
| バイト | 0 0 |
| ワード | 0 1 |
| ロングワード | 1 0 |

→ ディスティネーション・オペランドで指定したデータレジスタ番号 (000~111)

●サンプル・リスト

SUBX.B D1, D2

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----------------|----|-----|---|---|-----|---------|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 1 | ディスティネーションレジスタ | 1 | サイズ | 0 | 0 | R/M | ソースレジスタ | | | | | |

ソース・オペランドで指定したアドレスレジスタ番号 (000~111)
1 : メモリとメモリとの操作

| サイズ | 対応ビット |
|--------|-------|
| バイト | 0 0 |
| ワード | 0 1 |
| ロングワード | 1 0 |

→ ディスティネーション・オペランドで指定したアドレスレジスタ番号 (000~111)

●サンプル・リスト

SUBX.L -(A1), -(A5)

33 MULS

[Multiply as Signed 符号付き乗算]

MULS {.W} <ea>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説

2つの符号付き16ビット整数（-32768～32767）のオペランドを乗算し、結果は、32ビットの符号付き整数としてディスティネーションへ格納します。

ディスティネーション・オペランドとしてのDnは、演算時に下位16ビットが取り出され上位ワードは無視されますが、結果は32ビットとして格納されることから、元の内容の32ビットすべてが失われます。

CCR

X：変化せず

N：演算結果が負ならセット(1)，それ以外はリセット(0)

Z：演算結果がゼロならセット(1)，それ以外はリセット(0)

V：常にリセット(0)

C：常にリセット(0)

機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|----|---|---|---|---|--------|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 0 | 0 | レジスタ | 1 | 1 | 1 | | | 実効アドレス | | | | | |

モード レジスタ

実効アドレス（データモード）

| アドレッシングモード | 対応ビット | レジスタ番号 |
|------------|-------|--------|
| Dn | 0 0 0 | レジスタ番号 |
| (An) | 0 1 0 | レジスタ番号 |
| (An)+ | 0 1 1 | レジスタ番号 |
| - (An) | 1 0 0 | レジスタ番号 |
| d16(An) | 1 0 1 | レジスタ番号 |
| d8(An,IX) | 1 1 0 | レジスタ番号 |
| Abs.W | 1 1 1 | 0 0 0 |
| Abs.L | 1 1 1 | 0 0 1 |
| d16(PC) | 1 1 1 | 0 1 0 |
| d8(PC,IX) | 1 1 1 | 0 1 1 |
| # Imm | 1 1 1 | 1 0 0 |

指定したデータレジスタ番号
(000～111)

サンプル・リスト

MULS (A2), D0
MULS 4(A4,A5.L), D3

34 MULU

[Multiply as Unsigned 符号なし乗算]

MULU {.W} <ea>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説

2つの符号なし16ビット整数（0～65535）のオペランドを乗算し、結果は、32ビットの符号なし整数としてディスティネーションへ格納します。

ディスティネーション・オペランドとしてのDnは、演算時には下位16ビットが取り出され上位ワードは無視されますが、結果は32ビットとして格納されることから、元の内容の32ビットすべてが失われます。

CCR

X：変化せず

N：演算の結果、データの最上位ビット（MSB）が“1”

ならセット(1)，それ以外はリセット(0)

Z：演算結果がゼロならセット(1)，それ以外はリセット(0)

V：常にリセット(0)

C：常にリセット(0)

機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|----|---|---|---|---|--------|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 0 | 0 | レジスタ | 0 | 1 | 1 | | | 実効アドレス | | | | | |

モード レジスタ

実効アドレス（データモード）

| アドレッシングモード | 対応ビット | レジスタ番号 |
|------------|-------|--------|
| Dn | 0 0 0 | レジスタ番号 |
| (An) | 0 1 0 | レジスタ番号 |
| (An)+ | 0 1 1 | レジスタ番号 |
| - (An) | 1 0 0 | レジスタ番号 |
| d16(An) | 1 0 1 | レジスタ番号 |
| d8(An,IX) | 1 1 0 | レジスタ番号 |
| Abs.W | 1 1 1 | 0 0 0 |
| Abs.L | 1 1 1 | 0 0 1 |
| d16(PC) | 1 1 1 | 0 1 0 |
| d8(PC,IX) | 1 1 1 | 0 1 1 |
| # Imm | 1 1 1 | 1 0 0 |

指定したデータレジスタ番号
(000～111)

サンプル・リスト

MULU D2, D3
MULU (A3), D0

| sou | size | dest |
|-----------|------|-------|
| Dn | B | |
| | W | 2 <10 |
| | L | |
| (An) | B | |
| | W | 2 <14 |
| | L | |
| (An) + | B | |
| | W | 2 <14 |
| | L | |
| - (An) | B | |
| | W | 2 <16 |
| | L | |
| d16(An) | B | |
| | W | 4 <18 |
| | L | |
| d8(An,IX) | B | |
| | W | 4 <20 |
| | L | |
| Abs.W | B | |
| | W | 4 <18 |
| | L | |
| Abs.L | B | |
| | W | 6 <32 |
| | L | |
| d16(PC) | B | |
| | W | 4 <18 |
| | L | |
| d8(PC,IX) | B | |
| | W | 4 <20 |
| | L | |
| # Imm | B | |
| | W | 4 <14 |
| | L | |

35 DIVS

[Divide as Signed 符号付き除算]

DIVS {.W} <ea>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | * | 0 |

解説

ディスティネーション・オペランドをソース・オペランドで除算し、結果をディスティネーションへ格納します。

演算は符号付き算術演算で行われ、オペランドに関する要点は次の通りです。

- ソース・オペランドである <ea> の内容はワードであり、下位16ビットが演算に使用される。
- ディスティネーション・オペランドのDnはロングワードであり、32ビットのすべてが演算に使用される。

演算結果に関する要点は次の通りです。

- 商はディスティネーション・オペランドDnの下位ワードへ格納される。
- 余りはディスティネーション・オペランドDnの上位ワードへ格納される。
- 余りがゼロでない限り、その符号は被除数と同じ。

本命令では、次の2つの特別な実行結果があります。

- ゼロで割るとTRAP(トラップ)が発生し、プロセッサは例外処理を開始する。
- 命令完了前にオーバーフローを検出すると、Vビットがセットされ、除算は実行されず、ディスティネーション・オペランドDnは更新されない。つまり、商も余りも格納されない。

機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|----|---|---|--------------------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | レジスタ | 1 | 1 | 1 | 実効アドレス モード レジスタ | | | | | | | |

↓
指定したデータレジスタ番号
(000~111)

| アドレッシング モード | 対応ビット | | | | |
|----------------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| Dn | 0 | 0 | 0 | レジスタ番号 | |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| (An) + | 0 | 1 | 1 | レジスタ番号 | |
| - (An) | 1 | 0 | 0 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An,1X) | 1 | 1 | 0 | レジスタ番号 | |
| Abs.W | 1 | 1 | 1 | 0 0 0 | |
| Abs.L | 1 | 1 | 1 | 0 0 1 | |
| d16(PC) | 1 | 1 | 1 | 0 1 0 | |
| d8(PC,1X) | 1 | 1 | 1 | 0 1 1 | |
| # Imm | 1 | 1 | 1 | 1 0 0 | |

アドレッシング・モード

| sou | size | dest |
|-----------|------|--------|
| | | Dn |
| | | # - |
| Dn | B | |
| | W | 2 <158 |
| | L | |
| (An) | B | |
| | W | 2 <158 |
| | L | |
| (An) + | B | |
| | W | 2 <158 |
| | L | |
| - (An) | B | |
| | W | 2 <164 |
| | L | |
| d16(An) | B | |
| | W | 4 <168 |
| | L | |
| d8(An,1X) | B | |
| | W | 4 <168 |
| | L | |
| Abs.W | B | |
| | W | 4 <168 |
| | L | |
| Abs.L | B | |
| | W | 6 <178 |
| | L | |
| d16(PC) | B | |
| | W | 4 <168 |
| | L | |
| d8(PC,1X) | B | |
| | W | 4 <168 |
| | L | |
| # Imm | B | |
| | W | 4 <168 |
| | L | |

COR

X : 変化せず

N : 商が負ならセット(1), それ以外はリセット(0)

Z : 商がゼロならセット(1), それ以外はリセット(0)

V : オーバーフローが発生すればセット(1), それ以外はリセット(0)

C : 常にリセット(0)

(注) オーバーフローが発生した場合, NビットおよびZビットは意味を持たず, 除算結果を正しく反映しない。

サンプル・リスト

DIVS D0,D1
DIVS (A0)+,D3

36 DIVU

[Divide as Unsigned 符号なし除算]

DIVU {.W} <ea>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | * | 0 |

解説

ディスティネーション・オペランドをソース・オペランドで除算し、結果をディスティネーションへ格納します。

演算は符号なし算術演算で行われ、オペランドに関する要点は次の通りです。

- ソース・オペランドである <ea> の内容はワードであり、下位16ビットが演算に使用される。
- ディスティネーション・オペランドのDnはロングワードであり、32ビットのすべてが演算に使用される。

演算結果に関する要点は次の通りです。

- 商はディスティネーション・オペランドDnの下位ワードへ格納される。
- 余りはディスティネーション・オペランドDnの上位ワードへ格納される。
- 余りがゼロでない限り、その符号は被除数と同じである。

本命令では、次の2つの特別な実行結果があります。

- ゼロで割るとTRAP（トラップ）が発生し、プロセッサは例外処理を開始する。
- 命令完了前にオーバフローを検出すると、Vビットがセットされ、除算は実行されず、ディスティネーション・オペランド Dnは更新されない。つまり、商も余りも格納されない。

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|------|----|---|---|--------|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | レジスタ | 0 | 1 | 1 | 実効アドレス | | | | | | | |

実効アドレス(データモード)

| アドレッシング モード | 対応ビット | | | | |
|----------------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| Dn | 0 | 0 | 0 | レジスタ番号 | |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| (An)+ | 0 | 1 | 1 | レジスタ番号 | |
| - (An) | 1 | 0 | 0 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An, IX) | 1 | 1 | 0 | レジスタ番号 | |
| Abs.W | 1 | 1 | 1 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 |
| d16(PC) | 1 | 1 | 1 | 0 | 1 |
| d8(PC, IX) | 1 | 1 | 1 | 0 | 1 |
| # Imm | 1 | 1 | 1 | 1 | 0 |

→ 指定したデータレジスタ番号
(000~111)

CCR

X : 変化せず

N : 商の最上位ビットが "1" ならセット(1), それ以外はリセット(0)

Z : 商がゼロならセット(1), それ以外はリセット(0)

V : オーバフローが発生すればセット(1), それ以外はリセット(0)

C : 常にリセット(0)

(注) オーバフローが発生した場合、NビットおよびZビットは意味を持たず、除算結果を正しく反映しない。

●アドレッシング・モード

| sou | size | dest |
|-----|--------|------|
| | | D n |
| # ~ | | |
| B | | |
| W | 2 <14 | L |
| B | | |
| W | 2 <14 | L |
| B | | |
| W | 2 <14 | L |
| B | | |
| W | 2 <14 | L |
| B | | |
| W | 2 <14 | L |
| B | | |
| W | 4 <14 | L |
| B | | |
| W | 4 <150 | L |
| B | | |
| W | 4 <148 | L |
| B | | |
| W | 6 <152 | L |
| B | | |
| W | 4 <148 | L |
| B | | |
| W | 4 <150 | L |
| B | | |
| W | 4 <144 | L |

●サンプル・リスト

DIVU D2,D3
DIVU (A0),D4
DIVU \$2000,D4

37 ● CMP

[Compare 比較]

CMP {.B/.W/.L} <ea>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | * | * |

解説

ディスティネーション・オペランドからソース・オペラントを減算し、その結果がCCRへ反映されますが、ディスティネーション・オペランドDnの内容は変更されません（減算結果は格納されない）。

CCR

X : 変化せず
N : 演算結果が負ならセット(1), それ以外はリセット(0)
Z : 演算結果がゼロならセット(1), それ以外はリセット(0)
V : オーバフローが発生すればセット(1), それ以外はリセット(0)
C : 柄下がり（ボロー）が発生すればセット(1), それ以外はリセット(0)

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|----|---|---|--------|---|---|---|-----|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 1 | 1 | レジスタ | | | | 実効アドレス | | | | モード | | | |
| | | | | モード | | | | レジスタ | | | | | | | |

| サイズ | 対応ビット |
|--------|-------|
| 8 | 7 6 |
| バイト | 0 0 0 |
| ワード | 0 0 1 |
| ロングワード | 0 1 0 |

実効アドレス（すべてのモード）

| アドレッシングモード | 対応ビット |
|------------|-------------|
| Dn | 5 4 3 2 1 0 |
| An* | 0 0 0 |
| (An) | レジスタ番号 |
| (An) + | 0 0 1 |
| -(An) | レジスタ番号 |
| d16(An) | 1 0 0 |
| d8(An,IX) | 1 1 0 |
| Abs.W | 1 1 1 |
| Abs.L | 0 0 0 |
| d16(PC) | 1 1 1 |
| d8(PC,IX) | 0 0 0 |
| # Imm | 1 1 1 |
| | レジスタ番号 |

指定したレジスタ番号 (0 0 0 ~ 1 1 1)

●サンプル・リスト

CMP D2, D3
CMP \$2000, D4
CMP.L \$3A(A2, D4.L), D2

* バイトサイズ不可
(注) CMP1を使用する

38 ● CMPA

[Compare Address アドレス・データ比較]

CMPA {.W/.L} <ea>, An

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | * | * |

解説

ディスティネーション・オペランドからソース・オペラントを減算し、その結果がCCRへ反映されますが、ディスティネーション・オペランドの内容は変更されません（減算結果は格納されない）。

オペランドサイズがワードである場合、ソースオペランドである<ea>の内容は、演算実行前に32ビットに符号拡張され、アドレスレジスタの全32ビットが演算の対象となります。

CCR

X : 変化せず
N : 演算結果が負ならセット(1), それ以外はリセット(0)
Z : 演算結果がゼロならセット(1), それ以外はリセット(0)
V : オーバフローが発生すればセット(1), それ以外はリセット(0)
C : 柄下がり（ボロー）が発生すればセット(1), それ以外はリセット(0)

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|----|---|---|--------|---|---|---|-----|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 1 | 1 | レジスタ | | | | 実効アドレス | | | | モード | | | |
| | | | | モード | | | | レジスタ | | | | | | | |

| サイズ | 対応ビット |
|--------|-------|
| 8 | 7 6 |
| ワード | 0 1 1 |
| ロングワード | 1 1 1 |

実効アドレス（すべてのモード）

| アドレッシングモード | 対応ビット |
|------------|-------------|
| Dn | 5 4 3 2 1 0 |
| An | レジスタ番号 |
| (An) | 0 0 1 |
| (An) + | レジスタ番号 |
| -(An) | レジスタ番号 |
| d16(An) | 1 0 1 |
| d8(An,IX) | 1 1 0 |
| Abs.W | 1 1 1 |
| Abs.L | 0 0 0 |
| d16(PC) | 1 1 1 |
| d8(PC,IX) | 0 0 0 |
| # Imm | 1 1 1 |
| | レジスタ番号 |

指定したアドレスレジスタ番号 (0 0 0 ~ 1 1 1)

●サンプル・リスト

CMPA.L A1, A2
CMPA.L \$2000, A4
CMPA.L (A2), A6

●アドレッシング・モード

| sou | size | dest |
|-----------|------|------|
| | | An |
| Dn | B | |
| Dn | W | 2 6 |
| Dn | L | 2 6 |
| An | B | |
| An | W | 2 6 |
| An | L | 2 6 |
| (An) | B | |
| (An) | W | 2 8 |
| (An) | L | 2 14 |
| (An) + | B | |
| (An) + | W | 2 8 |
| (An) + | L | 2 14 |
| -(An) | B | |
| -(An) | W | 2 10 |
| -(An) | L | 2 16 |
| d16(An) | B | |
| d16(An) | W | 4 12 |
| d16(An) | L | 4 18 |
| d8(An,IX) | B | |
| d8(An,IX) | W | 4 14 |
| d8(An,IX) | L | 4 20 |
| Abs.W | B | |
| Abs.W | W | 4 12 |
| Abs.W | L | 4 18 |
| Abs.L | B | |
| Abs.L | W | 6 15 |
| Abs.L | L | 6 22 |
| d16(PC) | B | |
| d16(PC) | W | 4 12 |
| d16(PC) | L | 4 18 |
| d8(PC,IX) | B | |
| d8(PC,IX) | W | 4 14 |
| d8(PC,IX) | L | 4 20 |
| # Imm | B | |
| # Imm | W | 4 16 |
| # Imm | L | 4 20 |

41 CLR

[Clear an Operand オペランドのクリア]

CLR {.B/.W/.L} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | 0 | 1 | 0 | 0 |

解説

ディスティネーション・オペランドのすべてのビットをクリア(0)します。
メモリへの操作は、一度読み出された後で書き込まれます。

CCR

X: 変化せず
N: 常にリセット(0)
Z: 常にセット(1)
V: 常にリセット(0)
C: 常にリセット(0)

●アドレスシング・モード

| sou | size | dest | | | | | | | |
|-----|------|------|------|-------|-------|---------|-----------|-------|-------|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L |
| # | - | # | # | # | # | # | # | # | # |
| B | 2 | 4 | 2 | 12 | 2 | 12 | 2 | 16 | 4 |
| W | 2 | 4 | 2 | 12 | 2 | 12 | 2 | 16 | 4 |
| L | 2 | 6 | 2 | 20 | 2 | 20 | 2 | 22 | 4 |
| | | | | | | | | | 28 |

*ソース・オペランドは存在しない

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|-----|--------|-----|------|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | サイズ | 実効アドレス | モード | レジスタ | | | | |



●サンプル・リスト

CLR.L D0
CLR.W (A1)+
CLR.L \$FF00

42 EXT

[Sign Extend 符号拡張]

EXT {.W/.L} Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説

データレジスタの符号ビット(バイトならビット7, ワードならビット15)を上位方向へコピーし、バイトをワードに、あるいはワードをロングワードに符号拡張します。

- オペレーションサイズがワード……Dnのビット7をビット15~8にコピーしてワードとする。
- オペレーションサイズがロングワード……Dnのビット15をビット31~16にコピーしてロングワードとする。

CCR

X: 変化せず
N: 演算結果が負ならセット(1), それ以外はリセット(0)
Z: 演算結果がゼロならセット(1), それ以外はリセット(0)
V: 常にリセット(0)
C: 常にリセット(0)

●アドレスシング・モード

| sou | size | dest |
|-----|------|------|
| # | ~ | Dn |
| B | 2 4 | |
| W | 2 4 | |
| L | 2 4 | |

*ソース・オペランド
は存在しない

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|-------|---|---|---|------|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | OPモード | 0 | 0 | 0 | レジスタ | | | | |

| 対応ビット | 意　味 |
|-------|---------------------------|
| 8 7 6 | データレジスタの下位バイトをワードに符号拡張 |
| 0 1 0 | データレジスタの下位ワードをロングワードに符号拡張 |
| 0 1 1 | データレジスタの下位ワードをロングワードに符号拡張 |

→指定したデータレジスタ番号(000～111)

●サンプル・リスト

EXT.W D0
EXT.L D2

43 NEG

[Negate 2進符号反転]

NEG {.B/.W/.L} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説 ゼロ(0)からディスティネーション・オペランドを減算し、結果をディスティネーションへ格納します。

CCR X:Cビットと同じ値

N:演算結果が負ならセット(1), それ以外はリセット(0)

Z:演算結果がゼロならセット(1), それ以外はリセット(0)

V:オーバフローが発生すればセット(1), それ以外はリセット(0)

C:桁下がり(ボロー)が発生すればセット(1), それ以外はリセット(0)

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|-----|--------|-----|------|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | サイズ | 実効アドレス | モード | レジスタ | | | | |



●アドレッシング・モード

| sou | size | dest | | | | | | | | | | | | | |
|-----|------|------|------|--------|--------|---------|-----------|-------|-------|---|---|---|---|---|---|
| | | Dn | (An) | (An) + | - (An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # | - | # | - | # | - |
| | B | 2 4 | 2 12 | 2 12 | 2 14 | 4 16 | 4 18 | 4 16 | 6 20 | | | | | | |
| | W | 2 4 | 2 12 | 2 12 | 2 14 | 4 16 | 4 18 | 4 16 | 6 20 | | | | | | |
| | L | 2 6 | 2 20 | 2 20 | 2 22 | 4 24 | 4 26 | 4 24 | 6 28 | | | | | | |

* ソース・オペランドは存在しない

●サンプル・リスト

NEG.B -(A0)
NEG.L D3

44 NEGX

[Negate with Extend 拡張付き2進符号反転]

NEGX {.B/.W/.L} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説 ゼロ(0)からディスティネーション・オペランドとXビットを減算し、結果をディスティネーションのロケーションへ格納します。

CCR X:Cビットと同じ値

N:演算結果が負ならセット(1), それ以外はリセット(0)

Z:演算結果がゼロでなければリセット(0), それ以外は変化せず

V:オーバフローが発生すればセット(1), それ以外はリセット(0)

C:桁下がり(ボロー)が発生すればセット(1), それ以外はリセット(0)

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|-----|--------|-----|------|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | サイズ | 実効アドレス | モード | レジスタ | | | | |



●アドレッシング・モード

| sou | size | dest | | | | | | | | | | | | | |
|-----|------|------|------|--------|--------|---------|-----------|-------|-------|---|---|---|---|---|---|
| | | Dn | (An) | (An) + | - (An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # | - | # | - | # | - |
| | B | 2 4 | 2 12 | 2 12 | 2 14 | 4 16 | 4 18 | 4 16 | 6 20 | | | | | | |
| | W | 2 4 | 2 12 | 2 12 | 2 14 | 4 16 | 4 18 | 4 16 | 6 20 | | | | | | |
| | L | 2 6 | 2 20 | 2 20 | 2 22 | 4 24 | 4 26 | 4 24 | 6 28 | | | | | | |

* ソース・オペランドは存在しない

●サンプル・リスト

NEGX.B D0
NEGX.W (A0)
NEGX.L (A2)+

45 ● TST

[Test an operand オペランドのテスト]

TST {.B/.W/.L} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説 ディスティネーション・オペランドをゼロ(0)と比較し、結果をCCRへ反映しますが、オペランドは変更されません。

フラグの変化にみられるように、ある値の符号(正負)やゼロか否かを知りたい場合に使います。

CCR X: 変化せず

N: 演算結果が負ならセット(1), それ以外はリセット(0)

Z: 演算結果がゼロならセット(1), それ以外はリセット(0)

V: 常にリセット(0)

C: 常にリセット(0)

● 機械語フォーマット

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|-----|--------|---|---|---|---|---|---|--|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | サイズ | 実効アドレス | | | | | | | |

実効アドレス(データ・可変モード)

| サイズ | 対応ビット |
|--------|-------|
| 7 | 6 |
| バイト | 0 0 |
| ワード | 0 1 |
| ロングワード | 1 0 |

● アドレスリング・モード

| sou | size | dest | | | | | | | | | | | | | | | |
|-----|------|------|------|-------|-------|---------|-----------|-------|-------|---|----|---|----|---|----|---|----|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # | - | # | - | # | - | | |
| | B | 2 | 4 | 2 | 8 | 2 | 8 | 2 | 10 | 4 | 12 | 4 | 14 | 4 | 12 | 6 | 16 |
| | W | 2 | 4 | 2 | 8 | 2 | 8 | 2 | 10 | 4 | 12 | 4 | 14 | 4 | 12 | 6 | 16 |
| | L | 2 | 4 | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 4 | 18 | 4 | 16 | 6 | 20 |

* ソース・オペランドは存在しない

● サンプル・リスト

TST D0
TST (A0)

46 ● TAS

[Test and Set an operand テスト・アンド・セット]

TAS {.B} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説

マルチプロセッサ・システム(複数のMPUからなるシステム)間の同期をとるための命令であり、オペランドは共有RAMに対するものです。

操作はリード-モディファイ-ライトサイクルで行われ、ハードウェアでこのようなサイクルがサポートされていなければ意味をもたない命令です。

順に次のような操作を行います。

- ① 指定したバイトオペランドをテスト(ゼロと比較)し、その結果をNとZビットに反映する。
- ② オペランドの最上位ビット(ビット7)をセット(1)する。

CCR

X: 変化せず

N: 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1), それ以外はリセット(0)

Z: 演算結果がゼロならセット(1), それ以外はリセット(0)

V: 常にリセット(0)

C: 常にリセット(0)

● アドレスリング・モード

| sou | size | dest | | | | | | | | | | | | | | | |
|-----|------|------|------|-------|-------|---------|-----------|-------|-------|---|----|---|----|---|----|---|----|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # | - | # | - | # | - | | |
| | B | 2 | 4 | 2 | 14 | 2 | 14 | 2 | 16 | 4 | 18 | 4 | 20 | 4 | 18 | 6 | 22 |
| | W | | | | | | | | | | | | | | | | |
| | L | | | | | | | | | | | | | | | | |

* ソース・オペランドは存在しない

● 機械語フォーマット

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| アドレッシングモード | 対応ビット | | | | | | | | | | | | | |
|------------|-------|---|---|---|---|---|--------|------|-------|-------|---------|-----------|-------|-------|
| | 5 | 4 | 3 | 2 | 1 | 0 | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L |
| Dn | 0 | 0 | 0 | 0 | 0 | 0 | レジスタ番号 | | | | | | | |
| (An) | 0 | 1 | 0 | 0 | 0 | 0 | レジスタ番号 | | | | | | | |
| (An)+ | 0 | 1 | 1 | 0 | 0 | 0 | レジスタ番号 | | | | | | | |
| -(An) | 1 | 0 | 0 | 0 | 0 | 0 | レジスタ番号 | | | | | | | |
| d16(An) | 1 | 0 | 1 | 0 | 0 | 0 | レジスタ番号 | | | | | | | |
| d8(An,IX) | 1 | 1 | 0 | 0 | 0 | 0 | レジスタ番号 | | | | | | | |
| Abs.W | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

● サンプル・リスト

TAS (A0)
TST \$12F00

47 ● AND [AND logical 論理積]

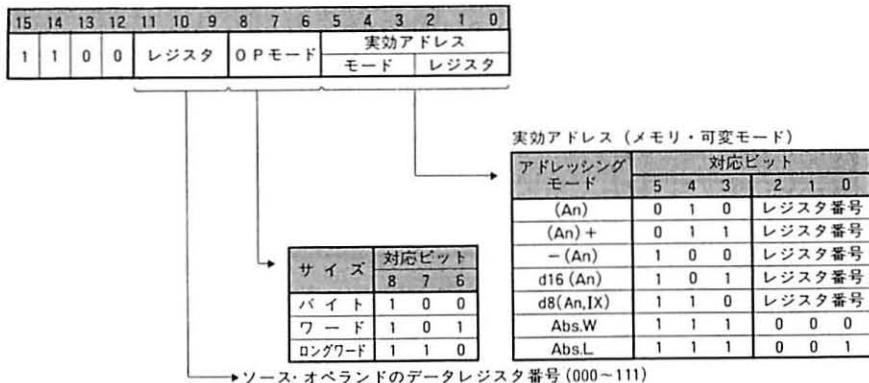
AND {.B/.W/.L} Dn, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説 ディスティネーション・オペランドとソース・オペランドの論理積をとり、結果をディスティネーションへ格納します。
(注) オペランドにアドレスレジスタを指定できません。

CCR
X : 変化せず
N : 演算の結果、データの最上位ビット (MSB) が "1" ならセット(1), それ以外はリセット(0)
Z : 演算結果がゼロならセット(1), それ以外はリセット(0)
V : 常にリセット(0)
C : 常にリセット(0)

●機械語フォーマット



●アドレッシング・モード

| sou | size | dest | | | | | | | |
|-----|------|------|-------|--------|---------|------------|-------|-------|-------|
| | | (An) | (An)+ | - (An) | d16(An) | d8(An, IX) | Abs.W | Abs.L | # Imm |
| # | - | # | - | # | - | # | - | # | - |
| Dn | B | 2 12 | 2 12 | 2 14 | 4 16 | 4 18 | 4 16 | 6 20 | |
| | W | 2 12 | 2 12 | 2 14 | 4 16 | 4 18 | 4 16 | 6 20 | |
| | L | 2 20 | 2 20 | 2 22 | 4 24 | 4 26 | 4 24 | 6 28 | |

●サンプル・リスト

AND D0, (A0)
AND D0, 10(A1, A4.L), D0

48 ● AND [AND logical 論理積]

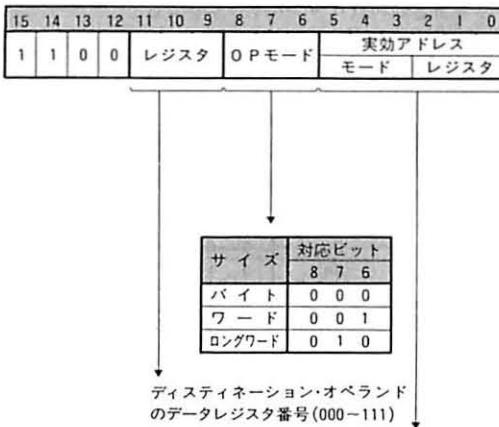
AND {.B/.W/.L} <ea>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説 ディスティネーション・オペランドとソース・オペランドの論理積をとり、結果をディスティネーションへ格納します。
(注) オペランドにアドレスレジスタを指定できません。

CCR
X : 変化せず
N : 演算の結果、データの最上位ビット (MSB) が "1" ならセット(1), それ以外はリセット(0)
Z : 演算結果がゼロならセット(1), それ以外はリセット(0)
C : 常にリセット(0)
C : 常にリセット(0)

●機械語フォーマット



●サンプル・リスト

AND D1, D2
AND 10(A1, A4.L), D0

(注) ANDIを使用する

| sou | size | dest |
|-------------|------|------|
| # | - | |
| Dn | B | 2 4 |
| | W | 2 4 |
| | L | 2 8 |
| | B | 2 8 |
| | W | 2 8 |
| | L | 2 14 |
| (An) | B | 2 8 |
| | W | 2 8 |
| | L | 2 14 |
| (An) + | B | 2 8 |
| | W | 2 8 |
| | L | 2 14 |
| - (An) | B | 2 10 |
| | W | 2 10 |
| | L | 2 16 |
| d16 (An) | B | 4 12 |
| | W | 4 12 |
| | L | 4 18 |
| d8 (An, IX) | B | 4 14 |
| | W | 4 14 |
| | L | 4 20 |
| Abs.W | B | 4 12 |
| | W | 4 12 |
| | L | 4 18 |
| Abs.L | B | 6 16 |
| | W | 6 16 |
| | L | 6 22 |
| d16 (PC) | B | 4 12 |
| | W | 4 12 |
| | L | 4 18 |
| d8 (PC, IX) | B | 4 14 |
| | W | 4 14 |
| | L | 4 20 |
| # Imm | B | 4 8 |
| | W | 4 8 |
| | L | 5 14 |

49 ● ANDI

[AND Immediate イミディエイト論理積]

ANDI {.B/.W/.L} # <data>, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説

ディスティネーション・オペランドとイミディエイト値との論理積をとり、結果をディスティネーションへ格納します。

イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。

CCR X: 変化せず

N: 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1), それ以外はリセット(0)

Z: 演算結果がゼロならセット(1), それ以外はリセット(0)

V: 常にリセット(0)

C: 常にリセット(0)

●アドレスシング・モード

| sou | size | dest | | | | | | | | | | | | | | | |
|-------|------|------|------|-------|-------|---------|-----------|-------|-------|----|----|----|----|----|----|----|----|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | | | | | | | | |
| # | - | # | - | # | - | # | - | # | - | | | | | | | | |
| B | 4 | 8 | 4 | 16 | 4 | 16 | 4 | 18 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 | |
| # Imm | W | 4 | 8 | 4 | 16 | 4 | 16 | 4 | 18 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 |
| L | 6 | 16 | 6 | 28 | 6 | 28 | 6 | 30 | 8 | 32 | 8 | 24 | 8 | 32 | 10 | 36 | |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|-----|--------|-----|------|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | サイズ | 実効アドレス | モード | レジスタ | | | | |

実効アドレス(データ・可変モード)

| サイズ | 対応ビット |
|--------|-------|
| | 7 6 |
| バイト | 0 0 |
| ワード | 0 1 |
| ロングワード | 1 0 |

<第2ワード: サイズがバイトおよびワードの時>

| | | | |
|----|---|---|---|
| 15 | 8 | 7 | 0 |
| | | | |

→ サイズがバイトの時に使用され、指定されたイミディエイト値の8ビットが格納される

<第2, 第3ワード: サイズがロングワードの時>

| | |
|--------------------|---|
| 15 | 0 |
| 指定したイミディエイト値の上位ワード | |
| 指定したイミディエイト値の下位ワード | |

●サンプル・リスト

ANDI.W #\$FF00,D0
ANDI.L #\$FF00FF00,(A0)

50 ● ANDI to CCR

[CCRとのイミディエイト論理積]

ANDI {.B} # <data>, CCR

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

CCRとイミディエイト値との論理積をとり、結果をCCRへ格納します。

イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。

(本命令はCCRの個別ビットをリセットするのに便利である)

CCR

X: ソース・オペランドの対応ビット(ビット4)が“0” ●アドレスシング・モード

ならリセット(0), それ以外は変化せず

N: ソース・オペランドの対応ビット(ビット3)が“0”

ならリセット(0), それ以外は変化せず

Z: ソース・オペランドの対応ビット(ビット2)が“0”

ならリセット(0), それ以外は変化せず

V: ソース・オペランドの対応ビット(ビット1)が“0”

ならリセット(0), それ以外は変化せず

C: ソース・オペランドの対応ビット(ビット0)が“0”ならリセット(0), それ以外は変化せず

| sou | size | dest |
|-------|------|------|
| | | CCR |
| # | - | |
| B | 4 | 20 |
| # Imm | W | |
| L | | |

●機械語フォーマット

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

<第2ワード>

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

→ 指定したイミディエイト値(8ビット)が格納される

●サンプル・リスト

ANDI #0,CCR

51 ●ANDI to SR [特權命令] [SRとの イミディエイト論理積]

ANDI {.W} # <data>, SR

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

SRとイミディエイト値との論理積をとり、結果をSRへ格納します。
イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。
(本命令はSRの個別ビットをリセットするのに便利である)

SR

- T : ソース・オペランドの対応ビット（ビット15, トレス）の値が“0”ならリセット(0), それ以外は変化せず
- S : ソース・オペランドの対応ビット（ビット13, スーパバイザ状態）の値が“0”ならリセット(0), それ以外は変化せず
- I₂ : ソース・オペランドの対応ビット（ビット10, 割り込みマスク）の値が“0”ならリセット(0), それ以外は変化せず
- I₁ : ソース・オペランドの対応ビット（ビット9, 割り込みマスク）の値が“0”ならリセット(0), それ以外は変化せず
- I₀ : ソース・オペランドの対応ビット（ビット8, 割り込みマスク）の値が“0”ならリセット(0), それ以外は変化せず
- X : ソース・オペランドの対応ビット（ビット4）が“0”ならリセット(0), それ以外は変化せず
- N : ソース・オペランドの対応ビット（ビット3）が“0”ならリセット(0), それ以外は変化せず
- Z : ソース・オペランドの対応ビット（ビット2）が“0”ならリセット(0), それ以外は変化せず
- V : ソース・オペランドの対応ビット（ビット1）が“0”ならリセット(0), それ以外は変化せず
- C : ソース・オペランドの対応ビット（ビット0）が“0”ならリセット(0), それ以外は変化せず

●アドレスシング・モード

| sou | size | dest |
|-------|------|------|
| # | ~ | SR |
| # Imm | B | |
| | W | 4 20 |
| | L | |

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

<第2ワード>

| | | |
|----|--------|---|
| 15 | ワードデータ | 0 |
|----|--------|---|

→ 指定したイミディエイト値(16ビット)が格納される

●サンプル・リスト

ANDI #\$FF00, SR

52 ●EOR [Exclusive OR logical 排他的論理和]

EOR {.B/.W/.L} Dn, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説

ディスティネーション・オペランドとソース・オペランドとの排他的論理和をとり、結果をディスティネーションへ格納しますが、ソース・オペランドにはデータレジスタのみ指定できます。

(注)●EOR <ea>, Dn という形式は存在しません。

●オペランドにアドレスレジスタを指定できません。

CCR

X : 変化せず

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1), それ以外はリセット(0)

Z : 演算結果がゼロならセット(1), それ以外はリセット(0)

V : 常にリセット(0)

C : 常にリセット(0)

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|------|-------|--------|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | レジスタ | OPモード | 実効アドレス | | | | | | | | | |

| アドレスシングモード | 対応ビット |
|-------------|---------------|
| 5 4 3 2 1 0 | |
| Dn | 0 0 0 レジスタ番号 |
| (An) | 0 1 0 レジスタ番号 |
| (An)+ | 0 1 1 レジスタ番号 |
| -(An) | 1 0 0 レジスタ番号 |
| d16(An) | 1 0 1 レジスタ番号 |
| d8(An,JX) | 1 1 0 レジスタ番号 |
| Abs.W | 1 1 1 0 0 0 0 |
| Abs.L | 1 1 1 0 0 0 1 |

ソース・オペランドで指定したデータレジスタ番号(000~111)

●アドレスシング・モード

| sou | size | dest | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An) | Abs.W | Abs.L |
|-------|------|------|-----|------|-------|-------|---------|--------|-------|-------|
| # ~ | | | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ |
| # Imm | B | | 2 | 4 | 2 | 12 | 2 | 12 | 2 | 14 |
| | W | | 2 | 4 | 2 | 12 | 2 | 12 | 2 | 14 |
| | L | | 2 | 8 | 2 | 20 | 2 | 20 | 2 | 22 |

●サンプル・リスト

EOR D2, D3
EOR.L D0, (A0)
EOR D3, \$1000

53 ● EORI

[Exclusive OR Immediate イミディエイト・データとの排他的論理和]

EORI {.B/.W/.L} # <data>, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説

ディスティネーション・オペランドとイミディエイト値との排他的論理和をとり、結果をディスティネーションへ格納します。

イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。

CCR

- X : 変化せず
- N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1), それ以外はリセット(0)
- Z : 演算結果がゼロならセット(1), それ以外はリセット(0)
- V : 常にリセット(0)
- C : 常にリセット(0)

●アドレスシング・モード

| sou | size | dest | | | | | | | | | |
|-------|------|------|------|-------|-------|---------|----------|-------|-------|-----|-----|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,X) | Abs.W | Abs.L | # ~ | # ~ |
| # Imm | B | 4 | 8 | 4 | 16 | 4 | 16 | 4 | 18 | 6 | 20 |
| # Imm | W | 4 | 8 | 4 | 16 | 4 | 16 | 4 | 18 | 6 | 20 |
| # Imm | L | 6 | 16 | 6 | 28 | 6 | 28 | 6 | 30 | 8 | 32 |
| | | | | | | | | | | 8 | 34 |
| | | | | | | | | | | 8 | 32 |
| | | | | | | | | | | 17 | 36 |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|-----|--------|-----|------|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | サイズ | 実効アドレス | モード | レジスタ | | | | |

| サイズ | 対応ビット |
|--------|-------|
| 7 | 6 |
| バイト | 0 0 |
| ワード | 0 1 |
| ロングワード | 1 0 |

実効アドレス(データ・可変モード)

| アドレスシング モード | 対応ビット | | | |
|----------------|-------|---|---|--------|
| | 5 | 4 | 3 | 2 1 0 |
| Dn | 0 | 0 | 0 | レジスタ番号 |
| (An) | 0 | 1 | 0 | レジスタ番号 |
| (An)+ | 0 | 1 | 1 | レジスタ番号 |
| -(An) | 1 | 0 | 0 | レジスタ番号 |
| d16(An) | 1 | 0 | 1 | レジスタ番号 |
| d8(An,X) | 1 | 1 | 0 | レジスタ番号 |
| Abs.W | 1 | 1 | 1 | 0 0 0 |
| Abs.L | 1 | 1 | 1 | 0 0 1 |

<第2ワード：サイズがバイトおよびワードの時>

| | | | |
|----|---|---|---|
| 15 | 8 | 7 | 0 |
| | | | |

→ サイズがバイトの時に使用され、指定されたイミディエイト値の8ビットが格納される

<第2, 第3ワード：サイズがロングワードの時>

| | |
|--------------------|---|
| 15 | 0 |
| | |
| 指定したイミディエイト値の上位ワード | |

↓

指定したイミディエイト値の下位ワード

●サンプル・リスト

EORI #FFFF,D0
EORI #\$FF00,(A2)

54 ● EORI to CCR

[CCRとのイミディエイト排他的論理和]

EORI {.B} # <data>, CCR

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドとイミディエイト値との排他的論理和をとり、結果をCCRへ格納します。

イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。
(本命令はCCRの個別ビットを反転するのに便利である)

CCR

X : ソース・オペランドの対応ビット(ビット4)が“1”なら反転、それ以外は変化せず

N : ソース・オペランドの対応ビット(ビット3)が“1”なら反転、それ以外は変化せず

Z : ソース・オペランドの対応ビット(ビット2)が“1”なら反転、それ以外は変化せず

V : ソース・オペランドの対応ビット(ビット1)が“1”なら反転、それ以外は変化せず

C : ソース・オペランドの対応ビット(ビット0)が“1”なら反転、それ以外は変化せず

●アドレスシング・モード

| sou | size | dest |
|-------|------|------|
| # ~ | | CCR |
| # Imm | B | 4 20 |
| # Imm | W | |
| # Imm | L | |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

<第2ワード>

| | | | | | | | | | |
|----|----|----|----|----|----|---|---|--------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | バイトデータ | |

→ 指定したイミディエイト値(8ビット)が格納される

●サンプル・リスト

EORI #\$FF,CCR

55 ● EORI to SR [特權命令]

[SRとのイミディエイト
排他的論理和]

EORI {.W} # <data>, SR

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説 ディスティネーション・オペランドとイミディエイト値との排他的論理和をとり、結果をSRへ格納します。

イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。
(本命令はSRの個別ビットを反転するのに便利である)

SR T : ソース・オペランドの対応ビット（ビット15, トレー
ス）の値が“1”なら反転、それ以外は変化せず

S : ソース・オペランドの対応ビット（ビット13, スーパ
バイザ状態）の値が“1”なら反転、それ以外は変
化せず

I₂ : ソース・オペランドの対応ビット（ビット10, 割り込
みマスク）の値が“1”なら反転、それ以外は変化せず

I₁ : ソース・オペランドの対応ビット（ビット9, 割り込みマスク）の値が“1”なら
反転、それ以外は変化せず

I₀ : ソース・オペランドの対応ビット（ビット8, 割り込みマスク）の値が“1”なら
反転、それ以外は変化せず

X : ソース・オペランドの対応ビット（ビット4）が“1”なら反転、それ以外は変化せず

N : ソース・オペランドの対応ビット（ビット3）が“1”なら反転、それ以外は変化
せず

Z : ソース・オペランドの対応ビット（ビット2）が“1”なら反転、それ以外は変化
せず

V : ソース・オペランドの対応ビット（ビット1）が“1”なら反転、それ以外は変化
せず

C : ソース・オペランドの対応ビット（ビット0）が“1”なら反転、それ以外変化せず

●アドレッシング・モード

| sou | size | dest |
|-----|------|------|
| # | - | SR |
| B | | |
| W | 4 20 | |
| L | | |

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

<第2ワード>

| | | |
|-----------------------------|---|--------|
| 15 | 0 | ワードデータ |
| → 指定したイミディエイト値(16ビット)が格納される | | |

●サンプル・リスト

EORI #\$4, SR

56 ● OR [Inclusive OR logical 論理和]

[Inclusive OR logical 論理和]

OR {.B/.W/.L} Dn, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説 ディスティネーション・オペランドとソース・オペランドとの論理和をとり、結果をディスティネーションへ格納します。

(注) オペランドにアドレスレジスタを指定できません。

CCR X : 変化せず

N : 演算の結果、データの最上位ビット（MSB）が“1”ならセット(1), それ以外はリセット(0)

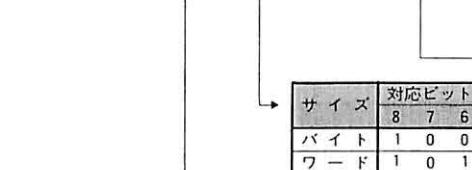
Z : 演算結果がゼロならセット(1), それ以外はリセット(0)

V : 常にリセット(0)

C : 常にリセット(0)

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|------|-------|--------|-----|------|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | レジスタ | OPモード | 実効アドレス | モード | レジスタ | | | | | | |



ソース・オペランドで指定したデータレジスタ番号(000~111)

●アドレッシング・モード

| sou | size | dest |
|------|------------------------------------|------------------------------------|
| (An) | (An) + | - (An) |
| # | - | d16(An) |
| B | 2 12 | 2 12 |
| W | 2 12 | 2 12 |
| L | 2 20 | 2 20 |
| Dn | 2 12 2 12 2 14 4 16 4 18 4 16 6 20 | 2 12 2 12 2 14 4 16 4 18 4 16 6 20 |
| | 2 20 2 20 2 22 4 24 4 26 4 24 6 28 | 2 20 2 20 2 22 4 24 4 26 4 24 6 28 |

●サンプル・リスト

OR D2, (A1)

57 OR

[Inclusive OR logical 論理和]

OR {.B/.W/.L} <ea>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説 ディスティネーション・オペランドとソース・オペランドとの論理和をとり、結果をディスティネーションへ格納します。
(注) オペランドにアドレスレジスタを指定できません。

CCR X: 変化せず

N: 演算の結果、データの最上位ビット (MSB) が "1"

ならセット(1), それ以外はリセット(0)

Z: 演算結果がゼロならセット(1), それ以外はリセット(0)

V: 常にリセット(0)

C: 常にリセット(0)

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|-------|--------|---|---|---|---|---|---|---|-----|------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | レジスタ | OPモード | 実効アドレス | | | | | | | | モード | レジスタ |

| | |
|--------|-------|
| サイズ | 対応ビット |
| 8 | 7 6 |
| バイト | 0 0 0 |
| ワード | 0 0 1 |
| ロングワード | 0 1 0 |

実効アドレス(データモード)

| アドレッシングモード | 対応ビット |
|------------|--------------|
| | 5 4 3 2 1 0 |
| Dn | 0 0 0 レジスタ番号 |
| (An) | 0 1 0 レジスタ番号 |
| (An)+ | 0 1 1 レジスタ番号 |
| -(An) | 1 0 0 レジスタ番号 |
| d16(An) | 1 0 1 レジスタ番号 |
| d8(An, IX) | 1 1 0 レジスタ番号 |
| Abs.W | 1 1 1 0 0 0 |
| Abs.L | 1 1 1 0 0 1 |
| d16(PC) | 1 1 1 0 1 0 |
| d8(PC, IX) | 1 1 1 0 1 1 |
| #Imm | 1 1 1 1 0 0 |

(注) ORIを使用する

●サンプル・リスト

OR D2, D4

58 ORI

[Inclusive OR Immediate イミディエイト論理和]

ORI {.B/.W/.L} # <data>, <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説 ディスティネーション・オペランドとソース・オペランドとの論理和をとり、結果をディスティネーションへ格納します。

イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。

CCR X: 変化せず

N: 演算の結果、データの最上位ビット (MSB) が "1"

ならセット(1), それ以外はリセット(0)

Z: 演算結果がゼロならセット(1), それ以外はリセット(0)

V: 常にリセット(0)

C: 常にリセット(0)

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|-----|-----|------|-----|------|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | サイズ | モード | レジスタ | モード | レジスタ | | | |

実効アドレス(データ・可変モード)

| アドレッシングモード | 対応ビット |
|------------|--------------|
| | 5 4 3 2 1 0 |
| Dn | 0 0 0 レジスタ番号 |
| (An) | 0 1 0 レジスタ番号 |
| (An)+ | 0 1 1 レジスタ番号 |
| -(An) | 1 0 0 レジスタ番号 |
| d16(An) | 1 0 1 レジスタ番号 |
| d8(An, IX) | 1 1 0 レジスタ番号 |
| Abs.W | 1 1 1 0 0 0 |
| Abs.L | 1 1 1 0 0 1 |

<第2ワード: サイズがバイトおよびワードの時>

| | | | |
|----|---|---|---|
| 15 | 8 | 7 | 0 |
| | | | |

→ サイズがバイトの時に使用され、指定されたイミディエイト値の8ビットが格納される

<第2, 第3ワード: サイズがロングワードの時>

| | |
|----|---|
| 15 | 0 |
| | |

→ サイズがワードの時に使用され、指定されたイミディエイト値の16ビットが格納される

●アドレッシング・モード

| sou | size | dest | | | | | | | | | | | | | | | |
|-----|------|------|------|-------|-------|---------|------------|-------|-------|------|----|---|----|---|----|----|----|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An, IX) | Abs.W | Abs.L | #Imm | # | # | # | # | | | |
| | B | 4 | 8 | 4 | 16 | 4 | 16 | 4 | 18 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 |
| | W | 4 | 8 | 4 | 16 | 4 | 16 | 4 | 18 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 |
| | L | 6 | 16 | 6 | 30 | 6 | 30 | 6 | 32 | 8 | 34 | 8 | 36 | 8 | 34 | 10 | 38 |

●サンプル・リスト

ORI #\$FFFF, D0
ORI #\$FF00, 4(A1, A2, L)

59 ORI to CCR

[CCRとのイミディエイト論理和]

ORI {.B} # <data>, CCR

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランド（CCR）とソース・オペランドとの論理和をとり、結果をCCRへ格納します。

イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。
(本命令はCCRの個別ビットをセットするのに便利である)

CCR

- X: ソース・オペランドの対応ビット（ビット4）が“1”ならセット(1), それ以外は変化せず
- N: ソース・オペランドの対応ビット（ビット3）が“1”ならセット(1), それ以外は変化せず
- Z: ソース・オペランドの対応ビット（ビット2）が“1”ならセット(1), それ以外は変化せず
- V: ソース・オペランドの対応ビット（ビット1）が“1”ならセット(1), それ以外は変化せず
- C: ソース・オペランドの対応ビット（ビット0）が“1”ならセット(1), それ以外は変化せず

アドレスシング・モード

| | | |
|-------|------|------|
| sou | size | dest |
| # | - | CCR |
| | B | 4 20 |
| # Imm | W | |
| | L | |

機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

第2ワード

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

バイトデータ

→ 指定したイミディエイト値(8ビット)が格納される

サンプル・リスト

ORI #\$FF, CCR

60 ORI to SR[特権命令]

[SRとのイミディエイト論理和]

ORI {.W} # <data>, SR

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドとソース・オペランドとの論理和をとり、結果をSRへ格納します。

イミディエイト値のサイズは、オペレーションサイズと同じでなければならず、指定したサイズに入りきらない大きさのイミディエイト値は指定できません。
(本命令はSRの個別ビットをセットするのに便利である)

SR

- T: ソース・オペランドの対応ビット（ビット15, トレース）の値が“1”ならセット(1), それ以外は変化せず
- S: ソース・オペランドの対応ビット（ビット13, スーパバイザ状態）の値が“1”ならセット(1), それ以外は変化せず
- I₂: ソース・オペランドの対応ビット（ビット10, 割り込みマスク）の値が“1”ならセット(1), それ以外は変化せず
- I₁: ソース・オペランドの対応ビット（ビット9, 割り込みマスク）の値が“1”ならセット(1), それ以外は変化せず
- I₀: ソース・オペランドの対応ビット（ビット8, 割り込みマスク）の値が“1”ならセット(1), それ以外は変化せず
- X: ソース・オペランドの対応ビット（ビット4）が“1”ならセット(1), それ以外は変化せず
- N: ソース・オペランドの対応ビット（ビット3）が“1”ならセット(1), それ以外は変化せず
- Z: ソース・オペランドの対応ビット（ビット2）が“1”ならセット(1), それ以外は変化せず
- V: ソース・オペランドの対応ビット（ビット1）が“1”ならセット(1), それ以外は変化せず
- C: ソース・オペランドの対応ビット（ビット0）が“1”ならセット(1), それ以外は変化せず

機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

第2ワード

| | | |
|----|---|--------|
| 15 | 0 | ワードデータ |
| . | . | . |

→ 指定したイミディエイト値(16ビット)が格納される

サンプル・リスト

ORI #\$FF, SR

アドレスシング・モード

| | | |
|-------|------|------|
| sou | size | dest |
| # | - | SR |
| | B | |
| # Imm | W | 4 20 |
| | L | |

61 NOT

[Logical Complement 論理否定]

NOT {.B/.W/.L} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | 0 |

解説 ディスティネーション・オペランドの1の補数をとり、結果をディスティネーションへ格納します。

CCR X: 変化せず

N: 演算結果が負ならセット(1)、それ以外はリセット(0)

Z: 演算結果がゼロならセット(1)、それ以外はリセット(0)

V: 常にリセット(0)

C: 常にリセット(0)

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|-----|--------|-----|------|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | サイズ | 実効アドレス | モード | レジスタ | | | | |



●アドレスシング・モード

| sou | size | dest | | | | | | | | | | | | | | |
|-----|------|------|------|--------|-------|---------|-----------|-------|-------|----|---|----|---|----|---|----|
| | | Dn | (An) | (An) + | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # | # | # | # | # | # | |
| B | 2 | 4 | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 4 | 18 | 4 | 16 | 6 | 20 |
| W | 2 | 4 | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 4 | 18 | 4 | 16 | 6 | 20 |
| L | 2 | 6 | 2 | 20 | 2 | 20 | 2 | 22 | 4 | 24 | 4 | 26 | 4 | 24 | 6 | 28 |

* ソース・オペランドは存在しない

●サンプル・リスト

NOT D0
NOT (A0)

62 ASL

[Arithmetic Shift Left 左へ算術シフト]

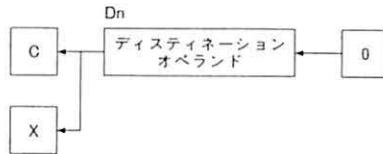
ASL {.B/.W/.L} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのデータレジスタの内容だけ、左へ算術シフトしますが、シフトカウンタに使用されるソース・オペランドのデータレジスタ値は、64の余りが用いられ、0～63までの範囲となります。

シフトの様子を次に示します。



- オペランドを左へシフト。
- 最上位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 下位ビットには0が入る。
- シフト操作によって符号変化が生じた場合、Vビットがセットされる。

CCR

X: Cビットと同じ値であるが、シフトカウント値がゼロ ● アドレスシング・モード

なら変化せず

N: 演算の結果、データの最上位ビット(MSB)が“1”

ならセット(1)、それ以外はリセット(0)

Z: 演算結果がゼロならセット(1)、それ以外はリセット(0)

V: シフト・オペレーション中一度でもデータの最上位ビ

ット(MSB)が変化すればセット(1)、それ以外はリ

セット(0)

C: 最後にシフトされて押し出されたビット値を保持、た

だし、シフトカウント値がゼロなら常にリセット(0)

| sou | size | dest |
|-----|------|----------------------------------|
| | | Dn |
| sou | size | # ~ |
| Dn | | B 2 6+2n W 2 6+2n L 2 8+2n |

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|-------------|----|-----|-----|---|---|------|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | カウントまたはレジスタ | dr | サイズ | i/r | 0 | 0 | レジスタ | | | | |

- ディスティネーション・レジスタ番号(000～111)(操作されるデータレジスタの番号)
- 1: シフト回数はソース・オペランドのデータレジスタで指定する
- 00: バイト操作
- 01: ワード操作
- 10: ロングワード操作
- 1: 左シフト
- ソースレジスタ番号(000～111)(シフト回数が入っているデータレジスタ番号)

●サンプル・リスト

ASL D2, D7

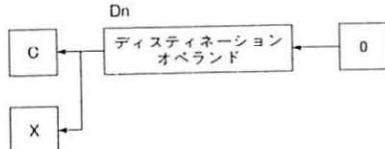
63 ASL [Arithmetic Shift Left 左へ算術シフト]

ASL {B/.W/.L} #<data>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのイミディエイト値だけ、左へ算術シフトしますが、シフトカウントとして指定できる範囲は1～8です。
シフトの様子を次に示します。



- オペランドを左へシフト。
- シフトカウントは1～8。
- 最上位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 下位ビットには0が入る。
- シフト操作によって符号変化が生じた場合、Vビットがセットされる。

CCR

X : Cビットと同じ値

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)，それ以外はリセット(0)

V : シフト・オペレーション中一度でもデータの最上位ビット(MSB)が変化すればセット(1)，それ以外はリセット(0)

C : 最後にシフトされて押し出されたビット値を保持

○アドレスリング・モード

| sou | size | dest |
|-------|-----------------------------|-----------|
| # | - | Dn |
| # | - | (An) |
| B | 2 ^{6+_n} | d16(An) |
| W | 2 ^{6+_n} | d8(An,IX) |
| L | 2 ^{6+_n} | Abs.W |
| # Imm | - | Abs.L |

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたはレジスタ | dr | サイズ | i/r | 0 | 0 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号(000～111)
(操作されるデータレジスタの番号)
- 0 : シフト回数は、ノース・オペランドのイミディエイト値で指定する
- 00 : バイト操作
01 : ワード操作
10 : ロングワード操作
- 1 : 左シフト
- シフト回数(1～8, ただし0は8を表す)

○サンプル・リスト

ASL #4,D0

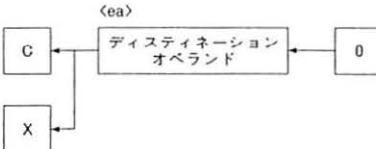
64 ASL [Arithmetic Shift Left 左へ算術シフト]

ASL {W} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

ディスティネーションで指定したメモリ・オペランドを、1ビットだけ、左へ算術シフトしますが、シフト回数(1回)やオペレーションサイズ(ワード)が固定されています。
シフトの様子を次に示します(シフト回数は固定されている)。



- オペランドを左へシフト。
- シフトカウントは1に固定されている。
- 最上位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 下位ビットには0が入る。
- シフト操作によって符号変化が生じた場合、Vビットがセットされる。

CCR

X : Cビットと同じ値

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)，それ以外はリセット(0)

V : シフト・オペレーション中一度でもデータの最上位ビット(MSB)が変化すればセット(1)，それ以外はリセット(0)

C : 最後にシフトされて押し出されたビット値を保持

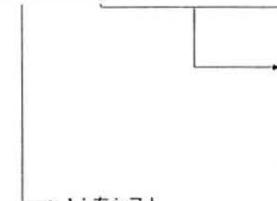
○アドレスリング・モード

| sou | size | dest |
|------|--------|-----------|
| (An) | (An) + | -(An) |
| # | - | d16(An) |
| B | | d8(An,IX) |
| W | 2 12 | Abs.W |
| L | 2 14 | Abs.L |

* ソースオペランドは存在しない

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|----|---|---|--------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | dr | 1 | 1 | 実効アドレス | | | | | |



| アドレスリングモード | 対応ビット |
|------------|-------|
| 5 | 4 |
| 4 | 3 |
| 3 | 2 |
| 2 | 1 |
| 0 | 0 |
| (An) | 0 1 0 |
| (An) + | 0 1 1 |
| -(An) | 1 0 0 |
| d16(An) | 1 0 1 |
| d8(An,IX) | 1 1 0 |
| Abs.W | 1 1 1 |
| Abs.L | 1 1 1 |

○サンプル・リスト

ASL (A0)+

65 ASR

[Arithmetic Shift Right 右へ算術シフト]

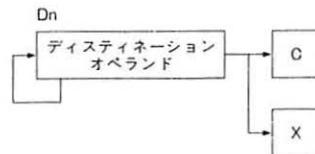
ASR {.B/.W/.L} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのデータレジスタの内容だけ、右へ算術シフトしますが、シフトカウントに使用されるソース・オペランドのデータレジスタ値は、64の余りが用いられ、0～63までの範囲となります。

シフトの様子を次に示します。



- オペランドを右へシフト。
- 最下位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 符号ビットは最上位ビットへ再度入る。

CCR

X : Cビットと同じ値であるが、シフトカウント値がゼロ
●アドレスシング・モード
なら変化せず

N : 演算の結果、データの最上位ビット (MSB) が "1"
ならセット (1), それ以外はリセット (0)

Z : 演算結果がゼロならセット (1), それ以外はリセット (0)

V : 常にリセット (0) ……最上位ビットは変化しない

C : 最後にシフトされて押し出されたビット値を保持、ただし、シフトカウント値がゼロなら常にリセット (0)

| sou | size | dest |
|-----|-------------------|------|
| Dn | # | |
| B | 2 ⁶⁺²ⁿ | |
| W | 2 ⁵⁺²ⁿ | |
| L | 2 ⁴⁺²ⁿ | |

機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-----------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたは レジスタ | dr | サイズ | i/r | 0 | 0 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号(000～111)
(操作されるデータレジスタの番号)
- 1 : シフト回数はソース・オペランドのデータレジスタで指定する
- 00 : バイト操作
- 01 : ワード操作
- 10 : ロングワード操作
- 0 : 右シフト
- ソースレジスタ番号(000～111)
(シフト回数が入っているデータレジスタ番号)

サンプル・リスト

ASR D2, D4

66 ASR

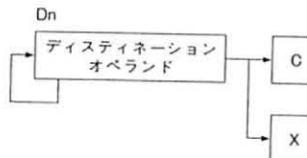
[Arithmetic Shift Right 右へ算術シフト]

ASR {.B/.W/.L} #<data>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのイミディエイト値だけ、右へ算術シフトしますが、シフトカウントとして指定できる範囲は1～8です。
シフトの様子を次に示します。



- オペランドを右へシフト。
- シフトカウントは1～8。
- 最下位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 符号ビットは最上位ビットへ再度入る。

CCR

X : Cビットと同じ値

N : 演算の結果、データの最上位ビット (MSB) が "1"
ならセット (1), それ以外はリセット (0)

Z : 演算結果がゼロならセット (1), それ以外はリセット (0)

V : 常にリセット (0) ……最上位ビットは変化しない

C : 最後にシフトされて押し出されたビット値を保持

アドレスシング・モード

| sou | size | dest |
|-----|-------------------|------|
| Dn | # | - |
| B | 2 ⁶⁺²ⁿ | |
| W | 2 ⁵⁺²ⁿ | |
| L | 2 ⁴⁺²ⁿ | |

機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-----------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたは レジスタ | dr | サイズ | i/r | 0 | 0 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号
(操作されるデータレジスタの番号)
- 0 : シフト回数はソース・オペランドのイミディエイト値で指定する
- 00 : バイト操作
- 01 : ワード操作
- 10 : ロングワード操作
- 0 : 右シフト
- シフト回数(1～8, ただし0は8を表す)

サンプル・リスト

ASR #4, D0

67 ASR

[Arithmetic Shift Right 右へ算術シフト]

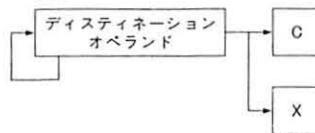
ASR {.W| <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーションで指定したメモリ・オペランドを、1ビットだけ、右へ算術シフトしますが、シフト回数(1回)やオペレーションサイズ(ワード)が固定されています。シフトの様子を次に示します。

<ea>



- オペランドを右へシフト。
- シフトカウントは1に固定されている。
- 最下位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 符号ビットは最下位ビットへ再度入る。

CCR X : Cビットと同じ値

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)，それ以外はリセット(0)

V : 常にリセット(0)……最上位ビットは変化しない

C : 最後にシフトされて押し出されたビット値を保持

●アドレスシング・モード

| sou | size | dest | | | | | | | |
|-----|-----------|-----------|-----------|-----------|---------|-----------|-------|-------|--|
| | | (An) | (An) + | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | |
| B | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | |
| W | 2 12 2 12 | 2 14 4 16 | 4 18 4 16 | 4 16 6 20 | | | | | |
| L | | | | | | | | | |

*ソース・オペランドは存在しない

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|---|---|---|--------|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | dr | 1 | 1 | | 実効アドレス | | | | | |

| アドレスシング モード | 対応ビット | | | | |
|----------------|-------|---|---|---|--------|
| | 5 | 4 | 3 | 2 | 1 |
| (An) | 0 | 1 | 0 | | レジスタ番号 |
| (An) + | 0 | 1 | 1 | | レジスタ番号 |
| - (An) | 1 | 0 | 0 | | レジスタ番号 |
| d16(An) | 1 | 0 | 1 | | レジスタ番号 |
| d8(An,IX) | 1 | 1 | 0 | | レジスタ番号 |
| Abs.W | 1 | 1 | 1 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 |



●サンプル・リスト

ASR (A0)
ASR 20(A2,D0,L)

68 LSL

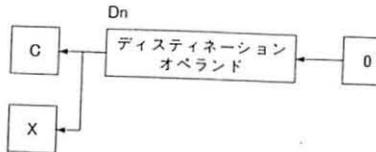
[Logical Shift Left 左へ論理シフト]

LSL {.B/.W/.L} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのデータレジスタの内容だけ、左へ論理シフトしますが、シフトカウンタに使用されるソース・オペランドのデータレジスタ値は、64の余りが用いられ、0～63までの範囲となります。シフトの様子を次に示します。



- オペランドを左へシフト。
- 最上位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 下位ビットには0が入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値であるが、シフトカウント値がゼロなら変化せず

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)，それ以外はリセット(0)

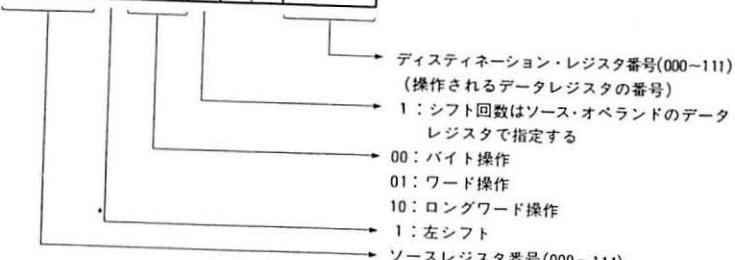
V : 常にリセット(0)

C : 最後にシフトされて押し出されたビット値を保持，ただし、シフトカウント値がゼロなら常にリセット(0)

| sou | size | dest |
|-----|------|------|
| Dn | # ~ | Dn |
| B | 2 6 | 2 6 |
| W | 2 6+ | 2 6+ |
| L | 2 8+ | 2 8+ |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|-----------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 1 | 0 | カウントまたは レジスタ | dr | サイズ | i/r | 0 | 1 | レジスタ | | | | | |



●サンプル・リスト

LSL D1,D2

69 LSL

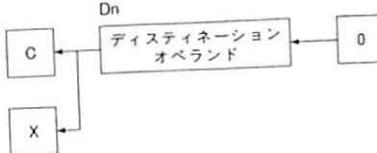
[Logical Shift Left 左へ論理シフト]

LSL {B/.W/.L} #<data>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのイミディエイト値だけ、左へ論理シフトしますが、シフトカウントとして指定できる範囲は1~8です。
シフトの様子を次に示します。



- オペランドを左へシフト。
- シフトカウントは1~8。
- 最上位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 下位ビットには0が入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1), それ以外はリセット(0)

Z : 演算結果がゼロならセット(1), それ以外はリセット(0)

V : 常にリセット(0)

C : 最後にシフトされて押し出されたビット値を保持

アドレスリング・モード

| sou | size | dest |
|-------|------------------|------|
| # | - | Dn |
| B | 2 ⁶⁺ⁿ | |
| W | 2 ⁶⁺ⁿ | |
| L | 2 ⁶⁺ⁿ | |
| # Imm | | |

機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | ガウントまたはレジスタ | dr | サイズ | i/r | 0 | 1 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号(000~111)
(操作されるデータレジスタの番号)
- 0 : シフト回数はソース・オペランドのイミディエイト値で指定する
- 00 : バイト操作
- 01 : ワード操作
- 10 : ロングワード操作
- 1 : 左シフト
- シフト回数(1~8, ただし0は8を表す)

サンプル・リスト

LSL #4,D0

70 LSL

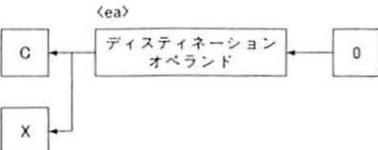
[Logical Shift Left 左へ論理シフト]

LSL {.W} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、1ビットだけ左へ論理シフトしますが、シフト回数(1回)やオペレーションサイズ(ワード)が固定されています。
シフトの様子を次に示します。



- オペランドを左へシフト。
- シフトカウントは1に固定されている。
- 最上位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 下位ビットには0が入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1), それ以外はリセット(0)

Z : 演算結果がゼロならセット(1), それ以外はリセット(0)

V : 常にリセット(0)

C : 最後にシフトされて押し出されたビット値を保持

アドレスリング・モード

| sou | size | dest | | | | | | | |
|-----|------|------|--------|-------|---------|-----------|-------|-------|--|
| | | (An) | (An) + | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | |
| # | - | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | |
| B | | | | | | | | | |
| W | 2 12 | 2 12 | 2 14 | 4 16 | 4 18 | 4 16 | 6 20 | | |
| L | | | | | | | | | |

* ソース・オペランドは存在しない

機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|----|---|---|--------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | dr | 1 | 1 | 実効アドレス | | | | | |

| アドレスリングモード | 対応ビット |
|------------|-------------|
| (An) | 5 4 3 2 1 0 |
| (An) + | 0 1 1 |
| -(An) | 1 0 0 |
| d16(An) | 1 0 1 |
| d8(An,IX) | 1 1 0 |
| Abs.W | 1 1 1 0 0 0 |
| Abs.L | 1 1 1 0 0 1 |

1 : 左シフト

サンプル・リスト

LSL (A0)
LSL 12(A1,A2.L)

71 LSR

[Logical Shift Right 右へ論理シフト]

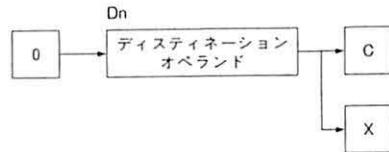
LSR {B/W/L} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのデータレジスタの内容だけ、右へ論理シフトしますが、シフトカウントに使用されるソース・オペランドのデータレジスタ値は、64の余りが用いられ、0～63までの範囲となります。

シフトの様子を次に示します。



- オペランドを右へシフト。
- 最下位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 上位ビットには0が入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値であるが、シフトカウント値がゼロなら変化せず

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)、それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)、それ以外はリセット(0)

V : 常にリセット(0)

C : 最後にシフトされて押し出されたビット値を保持、ただし、シフトカウント値がゼロなら常にリセット(0)

○アドレスリング・モード

| sou | size | dest |
|-----|------------------------------------|------|
| Dn | # ~ | |
| B | 2 ⁶⁺ 2 ²ⁿ | |
| W | 2 ⁶⁺ 2 ²ⁿ | |
| L | 2 ⁸⁺ 2 ²ⁿ | |

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-----------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたは レジスタ | dr | サイズ | i/r | 0 | 1 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号(000～111)
(操作されるデータレジスタの番号)
- 1 : シフト回数はソース・オペランドのデータレジスタで指定する
- 00 : バイト操作
01 : ワード操作
10 : ロングワード操作
- 0 : 右シフト
- ソースレジスタ番号(000～111)
(シフト回数が入っているデータレジスタ番号)

○サンプル・リスト

LSR D2, D4

72 LSR

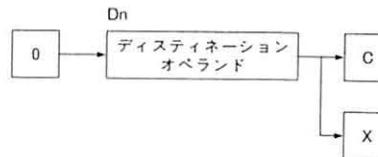
[Logical Shift Right 右へ論理シフト]

LSR {B/W/L} #<data>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのイミディエイト値だけ、右へ論理シフトしますが、シフトカウントとして指定できる範囲は1～8です。
シフトの様子を次に示します。



- オペランドを右へシフト。
- シフトカウントは1～8。
- 最下位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 上位ビットには0が入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)、それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)、それ以外はリセット(0)

V : 常にリセット(0)

C : 最後にシフトされて押し出されたビット値を保持

○アドレスリング・モード

| sou | size | dest |
|-------|------------------------------------|------|
| # Imm | | |
| B | 2 ⁶⁺ 2 ²ⁿ | |
| W | 2 ⁶⁺ 2 ²ⁿ | |
| L | 2 ⁸⁺ 2 ²ⁿ | |

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-----------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたは レジスタ | dr | サイズ | i/r | 0 | 1 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号(000～111)
(操作されるデータレジスタの番号)
- 0 : シフト回数はソース・オペランドのイミディエイト値で指定する
- 00 : バイト操作
01 : ワード操作
10 : ロングワード操作
- 0 : 右シフト
- シフト回数(1～8、ただし0は8を表す)

○サンプル・リスト

LSR #4, D0

73 LSR

[Logical Shift Right 右へ論理シフト]

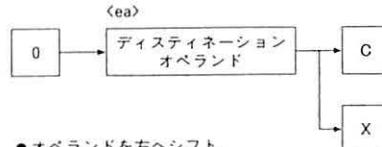
LSR {.W} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、1ビットだけ右へ論理シフトしますが、シフト回転(1回)やオペレーションサイズ(ワード)が固定されています。

シフトの様子を次に示します。



- オペランドを右へシフト。
- シフトカウントは1に固定されている。
- 最下位ビットからシフトされ押し出されたビットは、CおよびXビットへコピーされる。
- 上位ビットには0が入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)，それ以外はリセット(0)

V : 常にリセット(0)

C : 最後にシフトされて押し出されたビット値を保持

○アドレスリング・モード

| sou | size | dest | | | | | | | |
|-----|------|------|-------|-------|---------|-----------|-------|-------|---|
| | | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | |
| # | - | # | - | # | - | # | - | # | - |
| B | | | | | | | | | |
| W | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 4 |
| L | | | | | | | | | |

* ソース・オペランドは存在しない

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|----|---|---|--------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | dr | 1 | 1 | 実効アドレス | | | | | |

| アドレスリングモード | 対応ビット | | | |
|------------|-------|---|---|--------|
| | 5 | 4 | 3 | 2 1 0 |
| (An) | 0 | 1 | 0 | レジスタ番号 |
| (An)+ | 0 | 1 | 1 | レジスタ番号 |
| -(An) | 1 | 0 | 0 | レジスタ番号 |
| d16(An) | 1 | 0 | 1 | レジスタ番号 |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 |
| Abs.W | 1 | 1 | 1 | 0 0 0 |
| Abs.L | 1 | 1 | 1 | 0 0 1 |



○サンプル・リスト

LSR (A0)+

74 ROL

[Rotate Left 左へローテート]

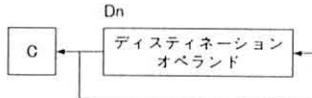
ROL {.B/.W/.L} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのデータレジスタの内容だけ、左へローテート(回転)しますが、ローテートカウンタに使用されるソース・オペランドのデータレジスタ値は、64の余りが用いられ、0～63までの範囲となります。

ローテートの様子を次に示します。



- オペランドを左へローテート。
- 最上位ビットからローテートされ押し出されたビットは、Cビットだけでなく最下位ビットにも入る。
- Vビットは常にクリア(0)される。

CCR

X : 変化せず

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)，それ以外はリセット(0)

V : 常にリセット(0)

C : 最後にローテートされて押し出されたビット値を保持，ただし、ローテート・カウント値がゼロなら常にリセット(0)

○アドレスリング・モード

| sou | size | dest |
|-----|------|------|
| # | - | Dn |
| B | 2 | 6+n |
| W | 2 | 2+n |
| L | 2 | 2+n |

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたはレジスタ | dr | サイズ | i/r | 1 | 1 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号(000～111)(操作されるデータレジスタの番号)
- 1: ローテート回数はソース・オペランドのデータレジスタで指定する
- 00: バイト操作
01: ワード操作
10: ロングワード操作
1: 左ローテート
- ソースレジスタ番号(000～111)(ローテート回数が入っているデータレジスタ番号)

○サンプル・リスト

ROL D2, D3

75 ROL

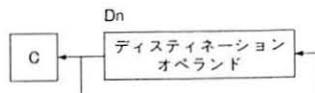
[Rotate Left 左ヘローテート]

ROL {.B/.W/.L} #<data>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのイミディエイト値だけ、左ヘローテート(回転)しますが、ローテートカウントとして指定できる範囲は1～8です。ローテートの様子を次に示します。



- オペランドを左ヘローテート。
- ローテートカウントは1～8。
- 最上位ビットからローテートされ押し出されたビットは、Cビットだけでなく最下位ビットにも入る。
- Vビットは常にクリア(0)される。

CCR

X: 変化せず

N: 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z: 演算結果がゼロならセット(1)，それ以外はリセット(0)

V: 常にリセット(0)

C: 最後にローテートされて押し出されたビット値を保持

◎アドレッシング・モード

| sou | size | dest |
|-------|-----------------------------|------|
| Dn | | |
| # | - | |
| # Imm | | |
| B | 2 ⁶ ₄ | |
| W | 2 ⁶ ₄ | |
| L | 2 ⁶ ₄ | |

◎機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-----------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたは レジスタ | dr | サイズ | i/r | 1 | 1 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号(000～111)(操作されるデータレジスタの番号)
- 0: ローテート回数はソース・オペランドのイミディエイト値で指定する
- 00: バイト操作
- 01: ワード操作
- 10: ロングワード操作
- 1: 左ローテート
- シフト回数(1～8, ただし0は8を表す)

◎サンプル・リスト

ROL #5,D0

76 ROL

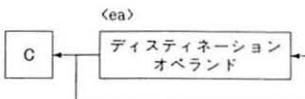
[Rotate Left 左ヘローテート]

ROL {.W} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | * |

解説

ディスティネーション・オペランドを、1ビットだけ、左ヘローテート(回転)しますが、ローテート回数(1回)やオペレーションサイズ(ワード)が固定されています。ローテートの様子を次に示します。



- オペランドを左ヘローテート。
- ローテートカウントは1に固定されている。
- 最上位ビットからローテートされ押し出されたビットは、Cビットだけでなく最下位ビットにも入る。
- Vビットは常にクリア(0)される。

CCR

X: 変化せず

N: 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z: 演算結果がゼロならセット(1)，それ以外はリセット(0)

V: 常にリセット(0)

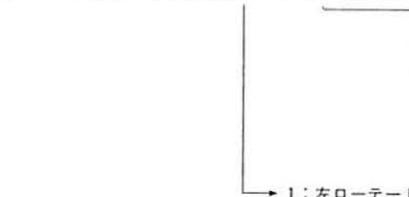
C: 最後にローテートされて押し出されたビット値を保持

◎アドレッシング・モード

| sou | size | dest | | | | | | | |
|-----|------|------|--------|-------|---------|-----------|-------|-------|----|
| | | (An) | (An) + | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | |
| # | - | # | - | # | # | - | # | - | # |
| B | | | | | | | | | |
| W | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 4 |
| L | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 6 |
| | | | | | | | | | 20 |

◎機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|----|---|---|--------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | dr | 1 | 1 | 実効アドレス | | | | | |



| アドレッシングモード | 対応ビット | | | |
|------------|-------|---|---|---|
| | 5 | 4 | 3 | 2 |
| (An) | 0 | 1 | 0 | |
| (An) + | 0 | 1 | 1 | |
| -(An) | 1 | 0 | 0 | |
| d16(An) | 1 | 0 | 1 | |
| d8(An,IX) | 1 | 1 | 0 | |
| Abs.W | 1 | 1 | 1 | 0 |
| Abs.L | 1 | 1 | 1 | 0 |

◎サンプル・リスト

ROL (A0)+

77 ROR

[Rotate Right 右ヘローテート]

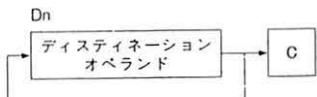
ROR {.B/.W/.L} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのデータレジスタの内容だけ、右ヘローテート(回転)しますが、ローテートカウントに使用されるソース・オペランドのデータレジスタ値は、64の余りが用いられ、0～63までの範囲となります。

ローテートの様子を次に示します。



- オペランドを右ヘローテート。
- 最下位ビットからローテートされ押し出されたビットは、Cビットだけでなく最上位ビットにも入る。
- Vビットは常にクリア(0)される。

CCR

X : 変化せず

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)、それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)、それ以外はリセット(0)

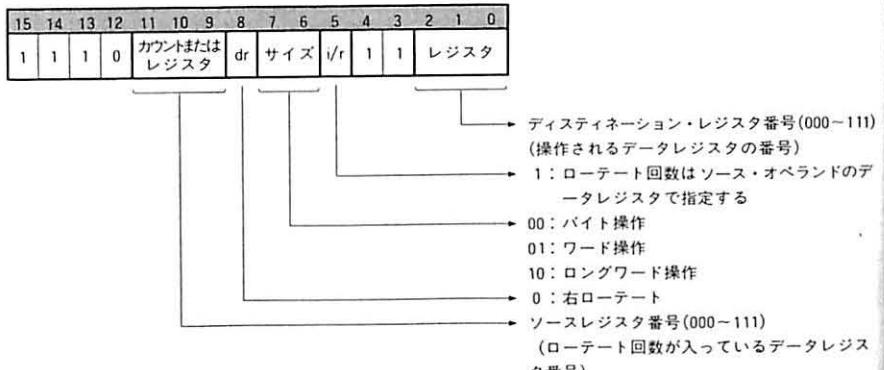
V : 常にリセット(0)

C : 最後にローテートされて押し出されたビット値を保持
ただし、ローテート・カウント値がゼロなら常にリセット(0)

●アドレスシング・モード

| sou | size | dest |
|-----|------------------------------------|------|
| Dn | # | - |
| B | 2 ⁶⁺ 2 ²ⁿ | |
| W | 2 ⁶⁺ 2 ²ⁿ | |
| L | 2 ⁸⁺ 2 ²ⁿ | |

●機械語フォーマット



●サンプル・リスト

ROR D2, D3

78 ROR

[Rotate Right 右ヘローテート]

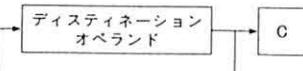
ROR {.B/.W/.L} #<data>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのイミディエイト値だけ、右ヘローテート(回転)しますが、ローテートカウントとして指定できる範囲は1～8です。ローテートの様子を次に示します。

Dn



- オペランドを右ヘローテート。
- ローテートカウントは1～8。
- 最下位ビットからローテートされ押し出されたビットは、Cビットだけでなく最上位ビットにも入る。
- Vビットは常にクリア(0)される。

CCR

X : 変化せず

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)、それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)、それ以外はリセット(0)

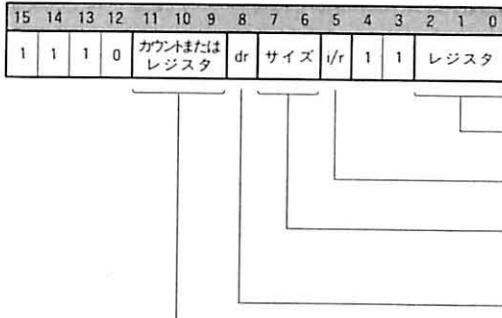
V : 常にリセット(0)

C : 最後にローテートされて押し出されたビット値を保持

●アドレスシング・モード

| sou | size | dest |
|-----|------------------------------------|------|
| Dn | # | - |
| B | 2 ⁶⁺ 2 ²ⁿ | |
| W | 2 ⁶⁺ 2 ²ⁿ | |
| L | 2 ⁸⁺ 2 ²ⁿ | |

●機械語フォーマット



●サンプル・リスト

ROR #2, D1

79 ROR

[Rotate Right 右ヘローテート]

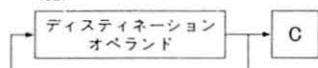
ROR {.W} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | * | 0 | * |

解説

ディスティネーション・オペランドを、1ビットだけ、右ヘローテート(回転)しますが、ローテート回数(1回)やオペレーションサイズ(ワード)が固定されています。ローテートの様子を次に示します。

<ea>



- オペランドを右ヘローテート。
- ローテートカウントは1に固定されている。
- 最下位ビットからローテートされ押し出されたビットは、Cビットだけでなく最上位ビットにも入る。
- Vビットは常にクリア(0)される。

CCR

X : 変化せず

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)，それ以外はリセット(0)

V : 常にリセット(0)

C : 最後にローテートされて押し出されたビット値を保持

機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|----|---|---|--------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | dr | 1 | 1 | 実効アドレス | | | | | |

実効アドレス(メモリ・可変モード)

| アドレス | 対応ビット | | | | |
|-----------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| (An) + | 0 | 1 | 1 | レジスタ番号 | |
| - (An) | 1 | 0 | 0 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 | |
| Abs.W | 1 | 1 | 1 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 |



→ 0 : 右ローテート

サンプル・リスト

ROR (A2)+

80 ROXL

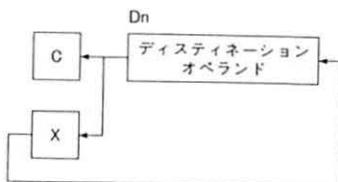
[Rotate with extend Left 拡張付き左ローテート]

ROXL {.B/.W/.L} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのデータレジスタの内容だけ、Xビットも含めて、左ヘローテートしますが、ローテートカウントに使用されるソースオペランドのデータレジスタ値は、64の余りが用いられ、0~63までの範囲となります。ローテートの様子を次に示します。



- オペランドをXビットも含めて左ヘローテート。
- 最上位ビットからローテートされ押し出されたビットは、CビットとXビットへ入る。
- 最下位ビットにはXビットの前の値が押し出されて入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値であるが、ローテート・カウント値がゼロなら変化せず

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)，それ以外はリセット(0)

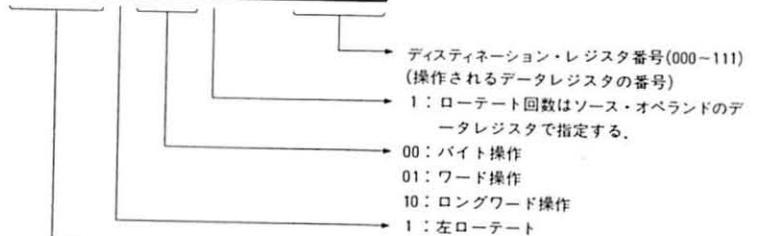
V : 常にリセット(0)

C : 最後にローテートされて押し出されたビット値を保持。ただし、ローテート・カウント値がゼロなら、演算前のXビットの値を保持

| sou | size | dest |
|-----|------------------------------------|------|
| B | # ~ | |
| W | 2 12 2 12 2 14 4 16 4 18 4 16 6 20 | |
| L | 2 12 2 12 2 14 4 16 4 18 4 16 6 20 | |

機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたはレジスタ | dr | サイズ | i/r | 1 | 0 | レジスタ | | | | | |



サンプル・リスト

ROXL D2, D7

81 ROXL

[Rotate with extend Left 拡張付き左ローテート]

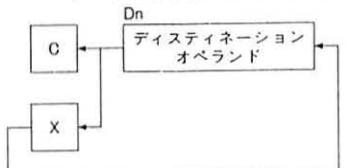
ROXL {.B/.W/.L} #<data>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのイミディエイト値だけ、Xビットも含めて、左ヘローテートしますが、ローテートカウントとして指定できる範囲は1～8です。

ローテートの様子を次に示します。



- オペランドをXビットも含めて左ヘローテート。
- ローテートカウントは1～8。
- 最上位ビットからローテートされ押し出されたビットは、CビットとXビットに入る。
- 最下位ビットにはXビットの前の値が押し出されて入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)、それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)、それ以外はリセット(0)

V : 常にリセット(0)

C : 最後にローテートされて押し出されたビット値を保持

●アドレスシング・モード

| sou | size | dest |
|-------|------|------------------------------------|
| # | - | Dn |
| # | B | 2 ⁶⁺ 2 ⁷⁺ |
| # Imm | W | 2 ⁶⁺ 2 ⁷⁺ |
| | L | 2 ⁸⁺ 2 ⁹⁺ |

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたはレジスタ | dr | サイズ | i/r | 1 | 0 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号(000～111)
(操作されるデータレジスクの番号)
- 0 : ローテート回数はソース・オペランドのイミディエイト値で指定する。
- 00 : バイト操作
01 : ワード操作
10 : ロングワード操作
1 : 左ローテート
ローテート回数(1～8、ただし0は8を表す)

●サンプル・リスト

ROXL #4,D7

82 ROXL

[Rotate with extend Left 拡張付き左ローテート]

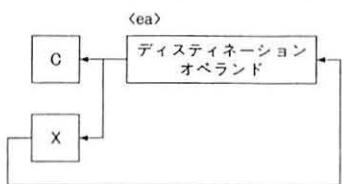
ROXL {.W} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、1ビットだけ、Xビットも含めて、左ヘローテートしますが、ローテート回数(1回)やオペレーションサイズ(ワード)が固定されています。

ローテートの様子を次に示します。



- オペランドをXビットも含めて左ヘローテート。
- ローテートカウントは1に固定される。
- 最上位ビットからローテートされ押し出されたビットは、0ビットとXビットに入る。
- 最下位ビットにはXビットの前の値が押し出されて入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)、それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)、それ以外はリセット(0)

V : 常にリセット(0)

C : 最後にローテートされて押し出されたビット値を保持

●アドレスシング・モード

| sou | size | dest | | | | | | | | | |
|-----|------------------------------------|------|-------|-------|---------|-----------|-------|-------|---|---|---|
| | | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # | # | # |
| # | - | # | - | # | - | # | - | # | - | # | - |
| B | | | | | | | | | | | |
| W | 2 12 2 12 2 14 4 16 4 18 4 16 6 20 | | | | | | | | | | |
| L | | | | | | | | | | | |

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|----|---|---|--------|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | dr | 1 | 1 | 実効アドレス | | | | |

| アドレッシングモード | 対応ビット | | | |
|------------|-------|---|---|--------|
| | 5 | 4 | 3 | 2 |
| (An) | 0 | 1 | 0 | レジスタ番号 |
| (An) + | 0 | 1 | 1 | レジスタ番号 |
| -(An) | 1 | 0 | 0 | レジスタ番号 |
| d16(An) | 1 | 0 | 1 | レジスタ番号 |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 |
| Abs.W | 1 | 1 | 1 | 0 0 0 |
| Abs.L | 1 | 1 | 1 | 0 0 1 |

●サンプル・リスト

ROXL (A0)+

83 ROXR

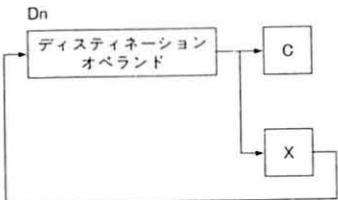
[Rotate with extend Right 拡張付き右ローテート]

ROXR {.B/.W/.L} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのデータレジスタの内容だけ、Xビットも含めて、右ヘローテートしますが、ローテートカウントに使用されるソースオペランドのデータレジスタ値は、64の余りが用いられ、0~63までの範囲となります。ローテートの様子を次に示します。



- オペランドをXビットも含めて右ヘローテート。
- 最下位ビットからローテートされ押し出されたビットは、GビットとXビットに入る。
- 最上位ビットにはXビットの前の値が押し出されて入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値であるが、ローテート・カウント値がゼロなら変化せず
N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)、それ以外はリセット(0)
Z : 演算結果がゼロならセット(1)、それ以外はリセット(0)
V : 常にリセット(0)
C : 最後にローテートされて押し出されたビット値を保持、ただし、ローテート・カウント値がゼロなら、演算前のXビットの値を保持

| sou | size | dest |
|-----|-------------------------------------|------|
| Dn | # ~ | |
| B | 2 ⁶⁺ 2 ⁷⁻ⁿ | |
| W | 2 ⁶⁺ 2 ⁷⁻ⁿ | |
| L | 2 ⁸⁺ 2 ⁷⁻ⁿ | |

機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-----------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたは レジスタ | dr | サイズ | i/r | 1 | 0 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号(000~111)
(操作されるデータレジスタの番号)
- 1 : ローテート回数はソース・オペランドのデータレジスタで指定する。
- 00 : バイト操作
01 : ワード操作
10 : ロングワード操作
0 : 右ローテート
ソースレジスタ番号(000~111)
(ローテート回数が入っているデータレジスタ番号)

サンプル・リスト

ROXR D2, D7

84 ROXR

[Rotate with extend Right 拡張付き右ローテート]

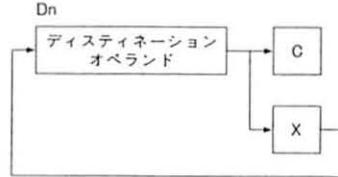
ROXR {.B/.W/.L} #<data>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、ソース・オペランドのイミディエイト値だけ、Xビットも含めて、右ヘローテートしますが、ローテートカウントとして指定できる範囲は1~8です。

ローテートの様子を次に示します。



- オペランドをXビットも含めて右ヘローテート。
- ローテートカウントは1~8。
- 最下位ビットからローテートされ押し出されたビットは、GビットとXビットに入る。
- 最上位ビットにはXビットの前の値が押し出されて入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値
N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)、それ以外はリセット(0)
Z : 演算結果がゼロならセット(1)、それ以外はリセット(0)
V : 常にリセット(0)
C : 最後にローテートされて押し出されたビット値を保持

アドレッシング・モード

| sou | size | dest |
|-------|------|-------------------------------------|
| # Imm | B | 2 ⁶⁺ 2 ⁷⁻ⁿ |
| | W | 2 ⁶⁺ 2 ⁷⁻ⁿ |
| | L | 2 ⁸⁺ 2 ⁷⁻ⁿ |

機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|-----------------|----|-----|-----|---|---|------|---|---|---|---|---|
| 1 | 1 | 1 | 0 | カウントまたは レジスタ | dr | サイズ | i/r | 1 | 0 | レジスタ | | | | | |

- ディスティネーション・レジスタ番号(000~111)
(操作されるデータレジスタの番号)
- 0 : ローテート回数はソースオペランドのイミディエイト値で指定する。
- 00 : バイト操作
01 : ワード操作
10 : ロングワード操作
0 : 右ローテート
ローテート回数(1~8、ただし0は8を表す)

サンプル・リスト

ROXR #4, D7

85 ROXR

[Rotate with extend Right 拡張付き右ローテート]

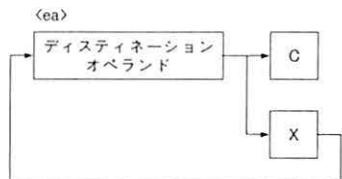
ROXR {.W} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | 0 | * |

解説

ディスティネーション・オペランドを、1ビットだけ、Xビットも含めて、右ヘローテートしますが、ローテート回数(1回)やオペレーションサイズ(ワード)が固定されています。

ローテートの様子を次に示します。



- オペランドをXビットも含めて右ヘローテート。
- ローテートカウントは1に固定される。
- 最下位ビットからローテートされ押し出されたビットは、CビットとXビットへ入る。
- 最上位ビットにはXビットの前の値が押し出されて入る。
- Vビットは常にクリア(0)される。

CCR

X : Cビットと同じ値

N : 演算の結果、データの最上位ビット(MSB)が“1”ならセット(1)，それ以外はリセット(0)

Z : 演算結果がゼロならセット(1)，それ以外はリセット(0)

V : 常にリセット(0)

C : 最後にローテートされて押し出されたビット値を保持

○アドレスシング・モード

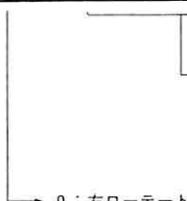
| sou | size | dest | | | | | | | |
|-----|------|------|-------|-------|---------|-----------|-------|-------|--|
| | | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | |
| B | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | # ~ | |
| W | 2 12 | 2 12 | 2 14 | 4 16 | 4 18 | 4 16 | 6 20 | | |
| L | | | | | | | | | |

○機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|----|---|---|--------------------|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | dr | 1 | 1 | 実効アドレス モード レジスタ | | | | | |

実効アドレス(メモリ・寄変モード)

| アドレスシング モード | 対応ビット | | | |
|----------------|-------|---|---|--------|
| | 5 | 4 | 3 | 2 1 0 |
| (An) | 0 | 1 | 0 | レジスタ番号 |
| (An) + | 0 | 1 | 1 | レジスタ番号 |
| -(An) | 1 | 0 | 0 | レジスタ番号 |
| d16(An) | 1 | 0 | 1 | レジスタ番号 |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 |
| Abs.W | 1 | 1 | 1 | 0 0 0 |
| Abs.L | 1 | 1 | 1 | 0 0 1 |



→ 0 : 右ローテート

○サンプル・リスト

ROXR (A0)+

86 BTST

[Test a Bit ビット・テスト]

BTST {.B/.L} Dn,<ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | * | - | - |

解説

ディスティネーション・オペランドの指定ビットがゼロ(0)であるか否かをテストし、結果をZビットへ反映します。テストしたいビットは、ソース・オペランドのデータレジスタで指定します。

オペレーションサイズは、ディスティネーション・オペランドに依存し、ハードウェアが決定しますので、意味のあるビット番号を、ソースレジスタにセットしなければなりません。

▶ディスティネーション・オペランドをデータレジスタに指定した場合、オペレーションサイズはロングワードとなり、ビット番号には、ソース・オペランドの値の32の余り(0 ~ 31)が用いられます。

▶ディスティネーション・オペランドをメモリ上のデータに指定した場合、オペレーションサイズはバイトとなり、ビット番号には、ソース・オペランドの値の8の余り(0 ~ 7)が用いられます。

CCR

X : 変化せず

N : 変化せず

Z : テストしたビット番号の値がゼロならセット(1)，それ以外はリセット(0)

V : 変化せず

C : 変化せず

○アドレスシング・モード

| sou | size | dest | | | | | | | | | | | |
|-----|------|------|------|--------|-------|---------|-----------|-------|-------|---------|-----------|-----|--|
| | | Dn | (An) | (An) + | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | d16(PC) | d8(PC,IX) | CCR | |
| B | # ~ | 2 | 8 | 2 | 8 | 2 | 10 | 4 | 12 | 4 | 14 | 4 | |
| W | | | | | | | | | | | | | |
| L | 2 | 6 | | | | | | | | | | | |

注) MPUのデータシートには CCR形式のマシンコードが存在するようだが、一般には使用する意味はない。

○機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|------|----|---|---|--------------------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | レジスタ | 1 | 0 | 0 | 実効アドレス モード レジスタ | | | | | | | |

実効アドレス(データモード)

| アドレスシング モード | 対応ビット | | | |
|----------------|-------|---|---|--------|
| | 5 | 4 | 3 | 2 1 0 |
| Dn* | 0 | 0 | 0 | レジスタ番号 |
| (An) | 0 | 1 | 0 | レジスタ番号 |
| (An) + | 0 | 1 | 1 | レジスタ番号 |
| -(An) | 1 | 0 | 0 | レジスタ番号 |
| d16(An) | 1 | 0 | 1 | レジスタ番号 |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 |
| Abs.W | 1 | 1 | 1 | 0 0 0 |
| Abs.L | 1 | 1 | 1 | 0 0 1 |
| d16(PC) | 1 | 1 | 1 | 0 1 0 |
| d8(PC,IX) | 1 | 1 | 1 | 0 1 1 |
| CCR | 1 | 1 | 1 | 1 0 0 |

→ ソース・オペランドで指定したデータレジスタの番号で、テストするビット番号が格納されるデータレジスタの番号(000 ~ 111)
* サイズはロングワードのみをサポート。
その他のアドレスシングはすべてバイトサイズであることに注意。

○サンプル・リスト

BTST D0,D1

BTST D1,(A0)

87 ● BTST

[Test a Bit ピット・テスト]

BTST {.B/.L} #<data>,<ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | * | - | - |

解説

ディスティネーション・オペランドの指定ビットがゼロ(0)であるか否かをテストし、結果をZビットへ反映します。テストしたいビットは、ソース・オペランドのイミディエイト値で指定します。

オペレーションサイズは、ディスティネーション・オペランドに依存し、ハードウェアが決定しますので、意味のあるビット番号を、イミディエイト値で指定しなければなりません。

▶ディスティネーション・オペランドをデータレジスタに指定した場合、オペレーションサイズはロングワードとなり、ビット番号には、ソース・オペランドの値の32の余り(0～31)が用いられます。

▶ディスティネーション・オペランドをメモリ上のデータに指定した場合、オペレーションサイズはバイトとなり、ビット番号には、ソース・オペランドの値の8の余り(0～7)が用いられます。

CCR

X : 変化せず

N : 変化せず

Z : テストしたビット番号の値がゼロならセット(1),それ以外はリセット(0)

V : 変化せず

C : 変化せず

●アドレスリング・モード

| SOU | Size | dest | | | | | | | | | |
|------|------|------|------|-------|-------|---------|-----------|-------|-------|---------|-----------|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | d16(PC) | d8(PC,IX) |
| #Imm | B | | 4 | 12 | 4 | 12 | 4 | 14 | 6 | 16 | 6 |
| | W | | | | | | | | | | |
| | L | 4 | 10 | | | | | | | | |

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

実効アドレス
(イミディエイト・データ形式を除くデータモード)

| アドレスリングモード | 対応ビット |
|-------------|-------|
| 5 4 3 2 1 0 | |
| Dn* | 0 0 0 |
| (An) | 0 1 0 |
| (An)+ | 0 1 1 |
| -(An) | 1 0 0 |
| d16(An) | 1 0 1 |
| d8(An,IX) | 1 1 0 |
| Abs.W | 1 1 1 |
| Abs.L | 1 1 1 |
| d16(PC) | 0 1 0 |
| d8(PC,IX) | 1 1 1 |

〈第2ワード〉

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

#<data>

ソース・オペランドで指定したビット番号

* サイズはロングワードのみをサポート。
その他のアドレスリングはすべてバイトサイズであることに注意。

●サンプル・リスト

BTST #4,D1
BTST #2,(A0)

88 ● BSET

[Test a Bit and Set ピット・テストとセット]

BSET {.B/.L} Dn,<ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | * | - | - |

解説

ディスティネーション・オペランドの指定ビットがゼロ(0)であるか否かをテストし、結果をZビットへ反映します。その後、指定ビットをセット(1)します。

セットしたいビットは、ソース・オペランドのデータレジスタで指定します。

オペレーションサイズは、ディスティネーション・オペランドに依存し、ハードウェアが決定しますので、意味のあるビット番号を、ソースレジスタにセットしなければなりません。

▶ディスティネーション・オペランドをデータレジスタに指定した場合、オペレーションサイズはロングワードとなり、ビット番号には、ソース・オペランドの値の32の余り(0～31)が用いられます。

▶ディスティネーション・オペランドをメモリ上のデータに指定した場合、オペレーションサイズはバイトとなり、ビット番号には、ソース・オペランドの値の8の余り(0～7)が用いられます。

CCR

X : 変化せず

N : 変化せず

Z : テストしたビット番号の値がゼロならセット(1),それ以外はリセット(0)

V : 変化せず

C : 変化せず

●アドレスリング・モード

●アドレスリング・モード

| SOU | size | dest | | | | | | | | | |
|-----|------|------|------|-------|-------|---------|-----------|-------|-------|---|----|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | | |
| # | - | # | - | # | - | # | - | # | - | # | - |
| B | | 4 | 12 | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 |
| W | | | | | | | | | | | |
| L | 2 | <8 | | | | | | | | | 6 |

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

レジスタ 1 1 1 実効アドレス

モード レジスタ

→

実効アドレス(データ・可変モード)

| アドレスリングモード | 対応ビット |
|-------------|-------|
| 5 4 3 2 1 0 | |
| Dn* | 0 0 0 |
| (An) | 0 1 0 |
| (An)+ | 0 1 1 |
| -(An) | 1 0 0 |
| d16(An) | 1 0 1 |
| d8(An,IX) | 1 1 0 |
| Abs.W | 1 1 1 |
| Abs.L | 1 1 1 |
| d16(PC) | 0 0 0 |
| d8(PC,IX) | 0 0 0 |

ソース・オペランドで指定したデータレジスタ番号で、テストするビット番号が格納されるデータレジスタの番号(000～111)

●サンプル・リスト

BSET D1,D2
BSET D7,(A2)

* サイズはロングワードのみをサポート。
その他のアドレスリングはすべてバイトサイズであることに注意。

89 ● BSET

[Test a Bit and Set ピット・テストとセット]

BSET {.B/.L} #<data>,<ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | * | - | - |

解説

- ディスティネーション・オペランドの指定ビットがゼロ(0)であるか否かをテストし、結果をZビットへ反映します。その後、指定ビットをセット(1)します。
- セットしたいビットは、ソース・オペランドのイミディエイト値で指定します。
- オペレーションサイズは、ディスティネーション・オペランドに依存し、ハードウェアが決定しますので、意味のあるビット番号を、イミディエイト値で指定しなければなりません。
- ▶ディスティネーション・オペランドをデータレジスタに指定した場合、オペレーションサイズはロングワードとなり、ビット番号には、ソース・オペランドの値の32の余り(0～31)が用いられます。
- ▶ディスティネーション・オペランドをメモリ上のデータに指定した場合、オペレーションサイズはバイトとなり、ビット番号には、ソース・オペランドの値の8の余り(0～7)が用いられます。

CCR

X : 変化せず

N : 変化せず

Z : テストしたビット番号
の値がゼロならセット
(1), それ以外はリセ
ット(0)

V : 変化せず

C : 変化せず

●アドレスリング・モード

| sou | size | dest | | | | | | | |
|------|------|-------|------|-------|-------|---------|-----------|-------|-------|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L |
| #Imm | B | | 4 | 16 | 4 | 16 | 4 | 18 | 6 |
| | W | | | | | | | 20 | 6 |
| | L | 4 <12 | | | | | | 20 | 8 |
| | | | | | | | | 24 | |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

実効アドレス

モード レジスタ

| アドレスリング モード | 対応ビット | | | | | |
|----------------|-------|---|---|--------|---|---|
| | 5 | 4 | 3 | 2 | 1 | 0 |
| Dn* | 0 | 0 | 0 | レジスタ番号 | | |
| (An) | 0 | 1 | 0 | レジスタ番号 | | |
| (An) + | 0 | 1 | 1 | レジスタ番号 | | |
| -(An) | 1 | 0 | 0 | レジスタ番号 | | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | | |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 | | |
| Abs.W | 1 | 1 | 1 | 0 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 | 1 |

<第2ワード>

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | #<data> |

ソース・オペランドで指定したビット番号

* サイズはロングワードのみをサポート。
その他のアドレスリングは、すべてバイトサイ
ズであることに注意。

●サンプル・リスト

BSET #4,D0
BSET #2,(A0)

90 ● BCLR

[Test a Bit and Clear ピット・テストとクリア]

BCLR {.B/.L} Dn,<ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | * | - | - |

解説

- ディスティネーション・オペランドの指定ビットがゼロ(0)であるか否かをテストし、結果をZビットへ反映します。その後、指定ビットをクリア(0)します。

クリアしたいビットは、ソース・オペランドのデータレジスタで指定します。

オペレーションサイズは、ディスティネーション・オペランドに依存し、ハードウェアが決定しますので、意味のあるビット番号を、ソースレジスタにセットしなければなりません。

▶ディスティネーション・オペランドをデータレジスタに指定した場合、オペレーションサイズはロングワードとなり、ビット番号には、ソース・オペランドの値の32の余り(0～31)が用いられます。

▶ディスティネーション・オペランドをメモリ上のデータに指定した場合、オペレーションサイズはバイトとなり、ビット番号には、ソース・オペランドの値の8の余り(0～7)が用いられます。

CCR

X : 変化せず

N : 変化せず

Z : テストしたビット番号
の値がゼロならセット
(1), それ以外はリセ
ット(0)

V : 変化せず

C : 変化せず

●アドレスリング・モード

| sou | size | dest | | | | | | | |
|------|------|-------|------|-------|-------|---------|-----------|-------|-------|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L |
| #Imm | B | | 2 | 12 | 2 | 12 | 2 | 14 | 4 |
| | W | | | | | | | 16 | 4 |
| | L | 2 <10 | | | | | | 18 | 4 |
| | | | | | | | | 17 | 6 |
| | | | | | | | | 20 | |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|------|---|---|---|-------|---|---|---|---|---|----------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | レジスタ | 1 | 1 | 0 | 対応ビット | | | | | | モード レジスタ |

実効アドレス(データ・可変モード)

| アドレスリング モード | 対応ビット | | | | | |
|----------------|-------|---|---|--------|---|---|
| | 5 | 4 | 3 | 2 | 1 | 0 |
| Dn* | 0 | 0 | 0 | レジスタ番号 | | |
| (An) | 0 | 1 | 0 | レジスタ番号 | | |
| (An) + | 0 | 1 | 1 | レジスタ番号 | | |
| -(An) | 1 | 0 | 0 | レジスタ番号 | | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | | |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 | | |
| Abs.W | 1 | 1 | 1 | 0 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 | 1 |

ソース・オペランドで指定したデータレジスタ番号で、テストするビット番号が格納されるデータレジスタの番号(000～111)

●サンプル・リスト

BCLR D0,D2
BCLR D3,(A0)

* サイズはロングワードのみをサポート。
その他のアドレスリングは、すべてバイトサイ
ズであることに注意。

91 BCLR

[Test a Bit and Clear ピット・テストとクリア]

BCLR {.B/.L} #<data>,<ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | * | - | - |

解説

ディスティネーション・オペランドの指定ビットがゼロ(0)であるか否かをテストし、結果をZビットへ反映します。その後、指定ビットをクリア(0)します。

クリアしたいビットは、ソース・オペランドのイミディエイト値で指定します。

オペレーションサイズは、ディスティネーション・オペランドに依存し、ハードウェアが決定しますので、意味のあるビット番号を、イミディエイト値で指定しなければなりません。

▶ディスティネーション・オペランドをデータレジスタに指定した場合、オペレーションサイズはロングワードとなり、ビット番号には、ソース・オペランドの値の32の余り(0～31)が用いられます。

▶ディスティネーション・オペランドをメモリ上のデータに指定した場合、オペレーションサイズはバイトとなり、ビット番号には、ソース・オペランドの値の8の余り(0～7)が用いられます。

CCR

X : 変化せず

N : 変化せず

Z : テストしたビット番号
の値がゼロならセット
(1), それ以外はリセ
ット(0)

V : 変化せず

C : 変化せず

機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|--------|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 実効アドレス | | | | | |

○アドレッシング・モード

| sou | size | dest | | | | | | | | | | | | | | | | |
|-------|------|------|------|-------|-------|---------|-----------|-------|-------|---|----|---|----|---|----|---|---|--|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # | ~ | # | ~ | # | ~ | # | ~ | |
| # imm | | # | ~ | # | ~ | # | ~ | # | ~ | # | ~ | # | ~ | # | ~ | # | ~ | |
| B | | 4 | 16 | 4 | 16 | 4 | 18 | 6 | 20 | 6 | 22 | 6 | 20 | 8 | 24 | | | |
| W | | | | | | | | | | | | | | | | | | |
| L | | 4 | <16 | | | | | | | | | | | | | | | |

○サンプル・リスト

BCLR #2,D0
BCLR #1,(A0)

実効アドレス(データ・可変モード)

| アドレッシング モード | 対応ビット | | | | |
|----------------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| Dn* | 0 | 0 | 0 | レジスタ番号 | |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| (An)+ | 0 | 1 | 1 | レジスタ番号 | |
| -(An) | 1 | 0 | 0 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 | |
| Abs.W | 1 | 1 | 1 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 |

* サイズはロングワードのみをサポート。
その他のアドレッシングはすべてバイトサイ
ズであることに注意。

92 BCHG

[Test a Bit and Change ピット・テストと反転]

BCHG {.B/.L} Dn,<ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | * | - | - |

解説

ディスティネーション・オペランドの指定ビットがゼロ(0)であるか否かをテストし、結果をZビットへ反映します。その後、指定ビットを反転します。

反転したいビットは、ソース・オペランドのデータレジスタで指定します。

オペレーションサイズは、ディスティネーション・オペランドに依存し、ハードウェアが決定しますので、意味のあるビット番号を、ソースレジスタにセットしなければなりません。

▶ディスティネーション・オペランドをデータレジスタに指定した場合、オペレーションサイズはロングワードとなり、ビット番号には、ソース・オペランドの値の32の余り(0～31)が用いられます。

▶ディスティネーション・オペランドをメモリ上のデータに指定した場合、オペレーションサイズはバイトとなり、ビット番号には、ソース・オペランドの値の8の余り(0～7)が用いられます。

CCR

X : 変化せず

N : 変化せず

Z : テストしたビット番号
の値がゼロならセット
(1), それ以外はリセ
ット(0)

V : 変化せず

C : 変化せず

機械語フォーマット

○アドレッシング・モード

| sou | size | dest | | | | | | | | | | | | | | | | |
|-------|------|------|------|-------|-------|---------|-----------|-------|-------|---|----|---|----|---|----|---|---|--|
| | | Dn | (An) | (An)+ | -(An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | # | ~ | # | ~ | # | ~ | # | ~ | |
| # imm | | # | ~ | # | ~ | # | ~ | # | ~ | # | ~ | # | ~ | # | ~ | # | ~ | |
| B | | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 4 | 18 | 4 | 16 | 6 | 20 | | | |
| W | | | | | | | | | | | | | | | | | | |
| L | | 2 | <8 | | | | | | | | | | | | | | | |

○サンプル・リスト

BCHG D2,D1
BCHG D3,8(A0)

実効アドレス(データ・可変モード)

| アドレッシング モード | 対応ビット | | | | |
|----------------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| Dn* | 0 | 0 | 0 | レジスタ番号 | |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| (An)+ | 0 | 1 | 1 | レジスタ番号 | |
| -(An) | 1 | 0 | 0 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 | |
| Abs.W | 1 | 1 | 1 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 |

* サイズはロングワードのみをサポート。
その他のアドレッシングはすべてバイトサイ
ズであることに注意。

93 ● BCHG

[Test a Bit and Change ピット・テストと反転]

BCHG {.B/.L} #<data>,<ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | * | - | - |

解説

ディスティネーション・オペランドの指定ビットがゼロ(0)であるか否かをテストし、結果をZビットへ反映します。その後、指定ビットを反転します。
反転したいビットは、ソース・オペランドのイミディエイト値で指定します。
オペレーションサイズは、ディスティネーション・オペランドに依存し、ハードウェアが決定しますので、意味のあるビット番号を、イミディエイト値で指定しなければなりません。

- ▶ ディスティネーション・オペランドをデータレジスタに指定した場合、オペレーションサイズはロングワードとなり、ビット番号には、ソース・オペランドの値の32の余り(0 ~ 31)が用いられます。
- ▶ ディスティネーション・オペランドをメモリ上のデータに指定した場合、オペレーションサイズはバイトとなり、ビット番号には、ソース・オペランドの値の8の余り(0 ~ 7)が用いられます。

CCR

X : 変化せず

N : 変化せず

Z : テストしたビット番号
の値がゼロならセット
(1), それ以外はリセ
ット(0)

V : 変化せず

C : 変化せず

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | |
|----|----|----|----|----|----|---|---|---|---|--------------------|---|---|---|---|---|--|--|--|--|--|--|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 実効アドレス モード レジスタ | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |

<第2ワード>

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---------|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | #<data> | 0 | 0 | 0 | 0 | 0 |

ソース・オペランドで指定したビット番号

●サンプル・リスト

BCHG #1,D2
BCHG #4,(A0)

* サイズはロングワードのみをサポート。
その他のアドレッシングはすべてバイトサイズであることに注意。

94 ● ABCD

[Add Decimal with extend 拡張付き10進加算]

ABCD {.B} Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | U | * | U | * |

解説

データレジスター間の2進加算命令で、オペランドは指定したデータレジスタです。
ディスティネーション・オペランドに、ソース・オペランドとXビットを加算し、結果をディスティネーションのロケーションへ格納します。

CCR

X : Cビットと同じ値

N : 未定義 (結果は保証されない)

Z : 演算結果がゼロでなければリセット(0), それ以外は変化せず

V : 未定義 (結果は保証されない)

C : 柄上がり(10進キャリ)が発生すればセット(1), それ以外はリセット(0)

●アドレッシング・モード

| sou | size | dest |
|-------|------|------|
| | | Dn |
| | # ~ | |
| # Imm | B | 2 6 |
| | W | |
| | L | |

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|--------------------|----|---|---|---|---|-----|-------------|---|---|---|---|
| 1 | 1 | 0 | 0 | ティスティネーション レジスタ | 1 | 0 | 0 | 0 | 0 | R/M | ソース レジスタ | 0 | 0 | 0 | 0 |

ソース側のデータレジスタ番号(000~111)
0 : データレジスタ間の演算
ディスティネーション側のデータレジスタ番号(000~111)

●サンプル・リスト

ABCD D2,D4

95 ABCD

[Add Decimal with extend 拡張付き10進加算]

ABCD |.B| -(An), -(An)

| X | N | Z | V | C |
|---|---|---|---|---|
| * | U | * | U | * |

解説

メモリ～メモリ間の2進化10進加算命令で、オペランドは、プリデクリメント・アドレスレジスタ間接形式でポイントされるメモリ上に存在します。
ディスティネーション・オペランドに、ソース・オペランドとXビットを加算し、結果をディスティネーションのロケーションへ格納します。

CCR X : Cビットと同じ値

N : 未定義（結果は保証されない）

Z : 演算結果がゼロでなければリセット(0), それ以外は変化せず

V : 未定義（結果は保証されない）

C : 柄上がり(10進キャリ)が発生すればセット(1), それ以外はリセット(0)

●アドレスシング・モード

| sou | size | dest |
|--------|--------|------|
| - (An) | B 2 18 | |
| - (An) | W | |
| - (An) | L | |

●機械語フォーマット



- プリデクリメント・アドレス形式で使用する、ソース側のアドレスレジスタ番号(000～111)
- 1 : メモリ間の演算
- プリデクリメント・アドレス形式で使用する、ディスティネーション側のアドレスレジスタ番号(000～111)

●サンプル・リスト

ABCD -(A2), -(A6)

96 SBCD

[Subtract Decimal with extend 拡張付き10進減算]

SBCD |.B| Dn, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| * | U | * | U | * |

解説

データレジスター～データレジスター間の2進化10進減算命令で、オペランドは指定したデータレジスターです。
ディスティネーション・オペランドから、ソース・オペランドとXビットを減算し、結果をディスティネーションのロケーションへ格納します。

CCR

X : Cビットと同じ値

N : 未定義（結果は保証されない）

Z : 演算結果がゼロでなければリセット(0), それ以外は変化せず

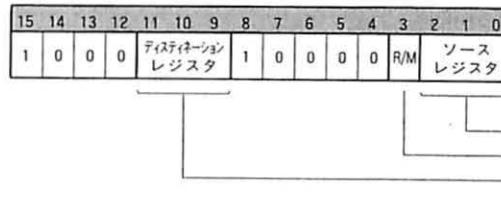
V : 未定義（結果は保証されない）

C : 柄下がり(10進ボロー)が発生すればセット(1), それ以外はリセット(0)

●アドレスシング・モード

| sou | size | dest |
|-----|-------|------|
| Dn | B 2 6 | |
| Dn | W | |
| Dn | L | |

●機械語フォーマット



- ソース側のデータレジスタ番号(000～111)
- 0 : データレジスター間の演算
- ディスティネーション側のデータレジスタ番号(000～111)

●サンプル・リスト

SBCD D2, D3

97 SBCD

[Subtract Decimal with extend 拡張付き10進減算]

SBCD {.B} -(An), -(An)

| X | N | Z | V | C |
|---|---|---|---|---|
| * | U | * | U | * |

解説

メモリ～メモリ間の2進化10進減算命令で、オペランドは、プリデクリメント・アドレスレジスタ間接形式でポイントされるメモリ上に存在します。
ディスティネーション・オペランドから、ソース・オペランドとXビットを減算し、結果をディスティネーションのロケーションへ格納します。

CCR X : Cビットと同じ値

N : 未定義 (結果は保証されない)

Z : 演算結果がゼロでなければリセット(0), それ以外は変化せず

V : 未定義 (結果は保証されない)

C : 桁下がり (10進ボロー) が発生すればセット(1), それ以外はリセット(0)

○アドレッシング・モード

| sou | size | dest |
|--------|------|------|
| - (An) | | |
| | B | 2 18 |
| | W | |
| | L | |

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|--------------------|---|---|---|---|---|-----|-------------|---|---|---|
| 1 | 0 | 0 | 0 | 0 | ディスティネーション レジスタ | 1 | 0 | 0 | 0 | 0 | R/M | ソース レジスタ | | | |

- プリデクリメント・アドレス形式で使用する、ソース側のアドレスレジスタ番号(000~111)
- 1: メモリ間の演算
- プリデクリメント・アドレス形式で使用する、ディスティネーション側のアドレスレジスタ番号(000~111)

○サンプル・リスト

SBCD -(A2),-(A6)

98 NBCD

[Negate Decimal with extend 拡張付き10進符号反転]

NBCD {.B} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | U | * | U | * |

解説

オペランドの補数をとる命令で、演算は2進化10演算で行われます。

ゼロ(0)から、ディスティネーション・オペランドとXビットを減算し、結果をディスティネーションのロケーションへ格納しますが、Xビットがリセット(0)されていれば10の補数、セット(1)されていれば9の補数が求まります。

CCR

X : Cビットと同じ値

N : 未定義 (結果は保証されない)

Z : 演算結果がゼロでなければリセット(0), それ以外は変化せず

V : 未定義 (結果は保証されない)

C : 桁下がり (10進ボロー) が発生すればセット(1), それ以外はリセット(0)

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

実効アドレス モード レジスタ

| アドレッシング モード | 対応ビット | | | | | |
|----------------|-------|---|---|--------|---|---|
| | 5 | 4 | 3 | 2 | 1 | 0 |
| Dn | 0 | 0 | 0 | レジスタ番号 | | |
| (An) | 0 | 1 | 0 | レジスタ番号 | | |
| (An) + | 0 | 1 | 1 | レジスタ番号 | | |
| - (An) | 1 | 0 | 0 | レジスタ番号 | | |
| d16 (An) | 1 | 0 | 1 | レジスタ番号 | | |
| d8 (An, IX) | 1 | 1 | 0 | レジスタ番号 | | |
| Abs.W | 1 | 1 | 1 | 0 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 | 1 |

○アドレッシング・モード

| sou | size | dest | | | | | | | | | |
|-----|------|------|------|--------|--------|----------|---------|-------|-------|---|----|
| | | Dn | (An) | (An) + | - (An) | d16 (An) | d8 (An) | Abs.W | Abs.L | # | ~ |
| | B | 2 | 6 | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 |
| | W | | | | | | | | | 4 | 18 |
| | L | | | | | | | | | 4 | 16 |
| | | | | | | | | | | 6 | 20 |

* ソース・オペランドは存在しない

○サンプル・リスト

NBCD D2
(A0)

Bcc |.B/.W| <label>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

条件付き分岐命令です。

命令のcc部で指定された条件が満たされた場合、指定したラベルへ分岐し、それ以外は分岐せず、本命令の次に位置する命令へ制御が移行します。

本命令の直前で比較命令を実行し、それが等しい場合にLABELという記号で表現されるロケーションへ分岐させるには、

BEQ LABEL

と記述します。このように、条件分岐命令は、CCRに影響をおよぼす命令の後を受け、必要なロケーションへ制御を移行させるために使用します。

条件が満足された場合、(PC)+ディスプレースメント、に分岐しますが、分岐先までの相対距離がどのように生成されるか、に関しては、アセンブラーが計算しますから、プログラマが関知する必要はありません。ただし、本命令と分岐先までの相対距離は2バイトのオフセットで表現され、それ以上離れたロケーションである場合、アセンブラーからエラーメッセージが出力されます。

ccとして以下の条件を指定することができますが、プログラミング・レベルで頻繁に使用される大小表現と、その他の表現に区分けしています。

| CMP x,y (y-xを実行している) | | |
|----------------------|-------------------|----------------------|
| 条件式 | 表現 | |
| | 符号なし | 符号付き |
| y > x | Hj(High) | GT(Greater Than) |
| y ≥ x | CC(Carry Clear) | GE(Greater or Equal) |
| y = x | EO(Equal) | EQ(Equal) |
| y ≠ x | NE(Not Equal) | NE(Not Equal) |
| y < x | CS(Carry Set) | LT(Less Than) |
| y ≤ x | LS(Lower or Same) | LE(Less or Equal) |

CCR

X:変化せず

N:変化せず

Z:変化せず

V:変化せず

C:変化せず

●アドレッシング・モード

| sou | size | dest |
|-----|------|------|
| | disp | #_ |
| | B | 2 10 |
| | W | 2 8 |
| | L | |

(注)アドレッシング形式はプログラムカウンタ・リラティブに分類されるが、書式の関係から disp という分類をした。

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|---------|----|-----------------|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | コンディション | | 8ビット・ディスプレースメント | | | | | | | | | |

→ 2の補数の整数で、この命令と分岐先の命令との相対距離を表し、バイト単位である。

| ニーモニック | 対応ビット |
|--------|-------------------|
| CC | 11 10 9 8 0 1 0 0 |
| CS | 0 1 0 1 |
| EQ | 0 1 1 1 |
| GE | 1 1 0 0 |
| GT | 1 1 1 0 |
| HI | 0 0 1 0 |
| LE | 1 1 1 1 |
| LS | 0 0 1 1 |
| LT | 1 1 0 1 |
| MI | 1 0 1 1 |
| NE | 0 1 1 0 |
| PL | 1 0 1 0 |
| VC | 1 0 0 0 |
| VS | 1 0 0 1 |

<第2ワード>

第2ワードが作成される時、第1ワードのディスプレースメント・フィールドには、ゼロが代入されます。

| | |
|------------------|---|
| 15 | 0 |
| 16ビット・ディスプレースメント | |

→ 2の補数の整数で、この命令と分岐先の命令との相対距離を表す。

(注1) 第1ワードの8ビット・ディスプレースメントが有効であれば、すなわち、ゼロでない場合、第2ワードは作成されません。これは8ビットのディスプレースメントで分岐可能であるため、作成する必要がないことを意味します。

(注2) PCの値は、この命令の位置+2に更新されているので、ハンドアセンブルで相対距離の計算をする時には、注意すること。

(注3) 本命令の直後に分岐するような分岐そのものは無意味であるわけですが、このような命令フォーマットは存在しません。8ビットディスプレースメントがゼロであれば、それはワード分岐となるからです。

●サンプル・リスト

| | |
|-----|---------|
| BH1 | EXE_JOB |
| BPL | EXE_JOB |
| BEQ | EXE_JOB |

EXE_JOBは記号番地を意味する

100 ●DBcc

[Test Condition, Decrement, and Branch
条件テスト, デクリメントおよび分岐]

DBcc {.W| Dn,<label>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

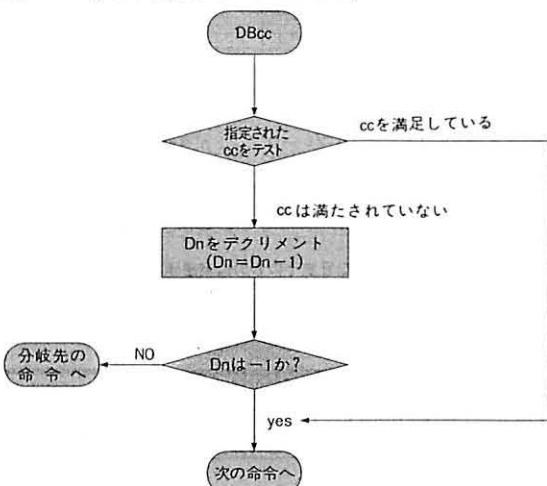
ループの制御命令です。

命令のcc部, データレジスタ, ラベル(ディスプレースメント), の3つのパラメータによって, ある命令群を繰り返し実行するもので, 68000は本命令に出会うと, 次のように行動します(分岐先のシンボルをLABELとする)。

- ① cc部で指定されているループの終了条件をテストする;
 - 条件を満足していれば, ループを抜け出す(本命令の次に位置する命令を実行).
 - それ以外は ②へ.
- ② ループカウンタであるDnの内容を1つだけ減じる(デクリメント);
 - Dnが-1(マイナス1)ならループを抜け出す(本命令の次に位置する命令を実行).
 - それ以外は LABEL という記号で表現されるロケーションへ分岐する.

分岐条件が満足された場合, (PC)+ディスプレースメント, に分岐しますが, 分岐先までの相対距離がどのように生成されるか, に関しては, アセンブラーが計算しますから, プログラマが関知する必要はありません. ただし, 本命令と分岐先までの相対距離は2バイトのオフセットで表現され, それ以上離れたロケーションである場合, アセンブラーからエラーメッセージが出力されます.

本命令は, 「Dnで指定した文字列長の読み込みを実行するが, c/r(キャリッジ・リターン・コード)を見つけたら処理を終了する」というように, 終了条件が複数の場合に威力を発揮します. いうまでもなく, このときのLABELで表現されるロケーションには, 1文字読み込み処理のエントリが位置しているはずです.



ccとして以下の条件を指定することができますが, プログラミング・レベルで頻繁に使用される大小表現と, その他の表現に区分けしています。

| CMP x,y (y=xを実行している) | | |
|----------------------|-------------------|----------------------|
| 条件式 | 表現 | |
| | 符号なし | 符号付き |
| y > x | HI(High) | GT(Greater Than) |
| y ≥ x | CC(Carry Clear) | GE(Greater or Equal) |
| y = x | EQ(Equal) | EQ(Equal) |
| y ≠ x | NE(Not Equal) | NE(Not Equal) |
| y < x | CS(carry Set) | LT(Less Than) |
| y ≤ x | LS(Lower or Same) | LE(Less or Equal) |

*DBFという表現はスマートではないので, DBRAというニーモニックが許され, これは, 単にDnにカウント値をセットしてループ制御するような, 単純ループに使用される(この用途も頻繁に存在する).

CCR

X : 変化せず

N : 変化せず

Z : 変化せず

V : 変化せず

C : 変化せず

○アドレスシング・モード

| (注) アドレスシング形式はプログラムカウンタ・リラティブに分類されるが, 書式の関係からdispという分類をした. | | | |
|--|------|------|------|
| sou | size | dest | disp |
| Dn | B | | # - |
| | W | 4 | • |
| | L | | |

○機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|---------|----|---|---|---|---|------|---|---|---|---|---|
| 0 | 1 | 0 | 1 | コンディション | 1 | 1 | 0 | 0 | 1 | レジスタ | | | | | |

→ カウンタとなるデータレジスタ番号(000~111)

<第2ワード>

| | | |
|----|---|------------------|
| 15 | 0 | 16ビット・ディスプレースメント |
|----|---|------------------|

↓
2の補数の整数で, この命令と分岐先との相対距離を表す.

(注) PCの値は, この命令の位置+2に更新されているので, ハンドアセンブルで相対距離の計算をする時には, 注意すること.

| ニーモニック | 対応ビット |
|--------|---------|
| CC | 0 1 0 0 |
| CS | 0 1 0 1 |
| EQ | 0 1 1 1 |
| F | 0 0 0 1 |
| GE | 1 1 0 0 |
| GT | 1 1 1 0 |
| HI | 0 0 1 0 |
| LE | 1 1 1 1 |
| LS | 0 0 1 1 |
| LT | 1 1 0 1 |
| MI | 1 0 1 1 |
| NE | 0 1 1 0 |
| PL | 1 0 1 0 |
| T | 0 0 0 0 |
| VC | 1 0 0 0 |
| VS | 1 0 0 1 |

○サンプル・リスト

| | |
|------|------------|
| DBCC | D2,EXE_JOB |
| DBCS | D4,EXE_JOB |
| DBVC | D5,EXE_JOB |
| DBRA | D1,EXE_JOB |

EXE_JOBは記号番地を意味する.

Seq {.B} <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

条件セット命令です。

本命令は、cc部で指定した条件が満足されると、オペランドの全ビットをセット（\$FF）にし、そうでなければ、リセット（\$00）するというものです。なお、オペランドのサイズはバイトです。

たとえば、ある値の比較結果が等しいかどうかを、D0に記憶しておきたいければ、

SEQ D0

のような記述をします。

ccとして以下の条件を指定することができますが、プログラミング・レベルで頻繁に使用される大小表現と、その他の表現に区分けしています。

| CMP x,y (y-xを実行している) | | |
|----------------------|-------------------|----------------------|
| 条件式 | 表現 | |
| | 符号なし | 符号付き |
| y > x | HI(High) | GT(Greater Than) |
| y ≥ x | CC(Carry Clear) | GE(Greater or Equal) |
| y = x | EQ(Equal) | EO(Equal) |
| y ≠ x | NE(Not Equal) | NE(Not Equal) |
| y < x | CS(Carry Set) | LT(Less Than) |
| y ≤ x | LS(Lower or Same) | LE(Less or Equal) |

その他の表現

| |
|--------------------|
| PL(Plus) |
| MI(Minus) |
| VC(Overflow Clear) |
| VS(Overflow Set) |
| F(always False) |
| T(always True) |

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|---------|----|---|---|--------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 0 | 1 | コンディション | 1 | 1 | | 実効アドレス | | | | | | | |

| アドレッシングモード | 対応ビット | | | | |
|------------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| Dn | 0 | 0 | 0 | レジスタ番号 | |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| (An) + | 0 | 1 | 1 | レジスタ番号 | |
| - (An) | 1 | 0 | 0 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An,Ix) | 1 | 1 | 0 | レジスタ番号 | |
| Abs.W | 1 | 1 | 1 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 |

| ニーモニック | 対応ビット |
|--------|-----------|
| CC | 11 10 9 8 |
| CS | 0 1 0 1 |
| EQ | 0 1 1 1 |
| F | 0 0 0 1 |
| GE | 1 1 0 0 |
| GT | 1 1 1 0 |
| HI | 0 0 1 0 |
| LE | 1 1 1 1 |
| LS | 0 0 1 1 |
| LT | 1 1 0 1 |
| MI | 1 0 1 1 |
| NE | 0 1 1 0 |
| PL | 1 0 1 0 |
| T | 0 0 0 0 |
| VC | 1 0 0 0 |
| VS | 1 0 0 1 |

CCR X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●アドレッシング・モード

| SOU | size | dest | | | | | | | | | | | |
|-----|------|------|---|------|--------|--------|---------|-----------|-------|-------|---|----|---|
| | | # | - | (An) | (An) + | - (An) | d16(An) | d8(An,Ix) | Abs.W | Abs.L | # | - | |
| B | 2 | ● | 2 | 12 | 2 | 12 | 2 | 14 | 4 | 16 | 4 | 18 | 4 |
| W | | | | | | | | | | | | | |
| L | | | | | | | | | | | | | |

→ 条件

| | |
|---|-------|
| ~ | 条件 |
| 4 | false |
| 6 | true |

* ソース・オペランドは存在しない

●サンプル・リスト

SGT D2
SEQ (A1)
SLS FLG_AREA FLG_AREAは記号番地を意味する

102 ● BRA

[Branch Always 無条件分岐]

BRA {.B/.W} <label>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

無条件ブランチ命令であり、ラベルで指定したロケーションへ無条件に分岐します。
(PC)+ディスプレースメント、に分岐しますが、分岐先までの相対距離がどのように生成されるか、に関しては、アセンブラーが計算しますから、プログラマが関知する必要はありません。ただし、本命令と分岐先までの相対距離は2バイトのオフセットで表現され、それ以上離れたロケーションである場合、アセンブラーからエラーメッセージが出力されます。

CCR X:変化せず
N:変化せず
Z:変化せず
V:変化せず
C:変化せず

●アドレスリング・モード

| sou | size | dest | disp |
|-----|------|------|------|
| # | - | | |
| B | 2 | 10 | |
| W | 4 | 10 | |
| L | | | |

(注) アドレスリング形式はプログラムカウンタ・リラティブに分類されるが、書式の関係からdispという分類をした。

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|-----------------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 8ビット・ディスプレースメント | | | | | | | |

→ 2の補数の整数で、この命令と分岐先の命令との相対距離を表し、バイト単位である。

〈第2ワード〉

第2ワードが作成される時、第1ワードのディスプレースメント・フィールドには、ゼロが代入されます。

| | | |
|----|---|------------------|
| 15 | 0 | 16ビット・ディスプレースメント |
| | | |

→ 2の補数の整数で、この命令と分岐先の命令との相対距離を表す。

- (注1) 第1ワードの8ビット・ディスプレースメントが有効であれば、すなわち、ゼロでない場合、第2ワードは作成されません。
これは8ビットのディスプレースメントで分岐可能であるため、作成する必要がないことを意味します。
- (注2) PCの値は、この命令の位置+2に変更されているので、ハンドアセンブルで相対距離の計算をする時には、注意すること。
- (注3) 本命令の直後に分岐するような分岐そのものは無意味であるわけですが、このような命令フォーマットは存在しません。8ビットディスプレースメントがゼロであれば、それはワード分岐となるからです。

●サンプル・リスト

BRA EXE_JOB ←EXE_JOBは記号番地を意味する

103 ● BSR

[Branch to Subroutine サブルーチン分岐]

BSR {.B/.W} <label>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

サブルーチン分岐命令です。BSR命令直後の命令のアドレスをシステム・スタックへプッシュし、ラベルで指定したロケーションへ分岐します。

(PC)+ディスプレースメント、に分岐しますが、分岐先までの相対距離がどのように生成されるか、に関しては、アセンブラーが計算しますから、プログラマが関知する必要はありません。ただし、本命令と分岐先までの相対距離は2バイトのオフセットで表現され、それ以上離れたロケーションである場合、アセンブラーからエラーメッセージが出力されます。

本命令は、もどり番地をシステムスタックにプッシュしており、RTSやRTRなどのリターン命令と組み合わせ、サブルーチン・コールを行うためのものです。

CCR X:変化せず
N:変化せず
Z:変化せず
V:変化せず
C:変化せず

●アドレスリング・モード

| sou | size | dest | disp |
|-----|------|------|------|
| # | - | | |
| B | 2 | 20 | |
| W | 4 | 20 | |
| L | | | |

(注) アドレスリング形式はプログラムカウンタ・リラティブに分類されるが、書式の関係からdispという分類をした。

●機械語フォーマット

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|-----------------|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 8ビット・ディスプレースメント | | | | | | | |

→ 2の補数の整数で、この命令と分岐先の命令との相対距離を表し、バイト単位である。

〈第2ワード〉

第2ワードが作成される時、第1ワードのディスプレースメント・フィールドには、ゼロが代入されます。

| | | |
|----|---|------------------|
| 15 | 0 | 16ビット・ディスプレースメント |
| | | |

→ 2の補数の整数で、この命令と分岐先の命令との相対距離を表す。

- (注1) 第1ワードの8ビット・ディスプレースメントが有効であれば、すなわち、ゼロでない場合、第2ワードは作成されません。
これは8ビットのディスプレースメントで分岐可能であるため、作成する必要がないことを意味します。
- (注2) PCの値は、この命令の位置+2に更新されているので、ハンドアセンブルで相対距離の計算をする時には、注意すること。
- (注3) 本命令の直後に分岐するような分岐そのものは無意味であるわけですが、このような命令フォーマットは存在しません。8ビットディスプレースメントがゼロであれば、それはワード分岐となるからです。

●サンプル・リスト

BSR EXE_JOB ←EXE_JOBは記号番地を意味する

104 ● JMP

[Jump ジャンプ]

JMP <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

無条件分岐命令ですが、ブランチ命令と異なり、68000のサポートする全域へ制御を移行できます。

今、A0に\$2000というアドレスデータが格納されているとき、

JMP (A0)

によって、プログラムの制御は無条件にアドレス\$2000へ移行します。

CCR

- X : 変化せず
- N : 変化せず
- Z : 変化せず
- V : 変化せず
- C : 変化せず

●アドレスシング・モード

| sou | size | dest | | | | | | | |
|-----|------|------|---------|-----------|-------|-------|---------|-----------|----|
| | | (An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | d16(PC) | d8(PC,IX) | |
| # | - | # | - | # | - | # | - | # | - |
| 2 | 8 | 4 | 10 | 4 | 14 | 4 | 10 | 6 | 12 |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

*サイズは存在しない。

●機械語フォーマット

| | | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|----------|---|---|---|---|---|--|--|--|--|--|--|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 実効アドレス | | | | | | | | | | | |
| | | | | | | | | | | モード レジスタ | | | | | | | | | | | |

実効アドレス(制御モード)

| アドレスシング モード | 対応ビット | | | |
|----------------|-------|---|---|--------|
| | 5 | 4 | 3 | 2 1 0 |
| (An) | 0 | 1 | 0 | レジスタ番号 |
| d16(An) | 1 | 0 | 1 | レジスタ番号 |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 |
| Abs.W | 1 | 1 | 1 | 0 0 0 |
| Abs.L | 1 | 1 | 1 | 0 0 1 |
| d16(PC) | 1 | 1 | 1 | 0 1 0 |
| d8(PC,IX) | 1 | 1 | 1 | 0 1 1 |

●サンプル・リスト

JMP (A0)
JMP EXE_JOB ← EXE_JOBは記号番地を意味する

105 ● JSR

[Jump to Subroutine サブルーチン・ジャンプ]

JSR <ea>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

サブルーチン・コール命令ですが、ブランチ命令と異なり、68000のサポートする全域へ制御を移行できます。

今、A0に\$2000というアドレスデータが格納されているとき、

JSR (A0)

によって、68000はJSR直後の命令が置かれているアドレスをシステム・スタックに PUSHし、アドレス\$2000へ分岐します。

本命令は、もどり番地をシステム・スタックにPUSHしており、RTSやRTRなどのリターン命令と組み合わせ、サブルーチン・コールを行うためのものです。

CCR

- X : 変化せず
- N : 変化せず
- Z : 変化せず
- V : 変化せず
- C : 変化せず

●アドレスシング・モード

| sou | size | dest | | | | | | | |
|-----|------|------|---------|-----------|-------|-------|---------|-----------|----|
| | | (An) | d16(An) | d8(An,IX) | Abs.W | Abs.L | d16(PC) | d8(PC,IX) | |
| # | - | # | - | # | - | # | - | # | - |
| 2 | 16 | 4 | 18 | 4 | 22 | 4 | 18 | 6 | 20 |
| | | | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

*サイズは存在しない。

●機械語フォーマット

| | | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|----------|---|---|---|---|---|--|--|--|--|--|--|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 実効アドレス | | | | | | | | | | | |
| | | | | | | | | | | モード レジスタ | | | | | | | | | | | |

実効アドレス(制御モード)

| アドレスシング モード | 対応ビット | | | |
|----------------|-------|---|---|--------|
| | 5 | 4 | 3 | 2 1 0 |
| (An) | 0 | 1 | 0 | レジスタ番号 |
| d16(An) | 1 | 0 | 1 | レジスタ番号 |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 |
| Abs.W | 1 | 1 | 1 | 0 0 0 |
| Abs.L | 1 | 1 | 1 | 0 0 1 |
| d16(PC) | 1 | 1 | 1 | 0 1 0 |
| d8(PC,IX) | 1 | 1 | 1 | 0 1 1 |

●サンプル・リスト

JSR (A0)
JSR EXE_JOB ← EXE_JOBは記号番地を意味する

106 ● RTR

[Return and Restore Condition Codes CCR回復とリターン]

RTR

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

サブルーチンからの復帰命令で、CCRと戻り番地を意味するプログラムカウンタ(PC)を、システムスタックから取り出します。

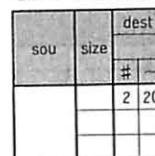
命令実行前のCCRとPCの内容は失われますが、ステータスレジスタのスーパバイザ部(上位8ビット)は影響を受けません。

note: PCの内容が書き換わるということは、PCでポイントされるアドレスへ制御が移行するのだから、分岐が起こることを意味する。

CCR

X: スタック上のワードに対応するビット4の値
N: スタック上のワードに対応するビット3の値
Z: スタック上のワードに対応するビット2の値
V: スタック上のワードに対応するビット1の値
C: スタック上のワードに対応するビット0の値

●アドレスシング・モード



* ソース・オペランド、ディスティネーション・オペランドという概念は存在しない。

* サイズは存在しない。

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

●サンプル・リスト

RTR

107 ● RTS

[Return from Subroutine サブルーチンからのリターン]

RTS

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

サブルーチンからの復帰命令で、戻り番地を意味するプログラムカウンタ(PC)を、システムスタックから取り出します。命令実行前のPCの内容は失われます。

note: PCの内容が書き換わるということは、PCでポイントされるアドレスへ制御が移行するのだから、分岐が起こることを意味する。

CCR

X: 変化せず
N: 変化せず
Z: 変化せず
V: 変化せず
C: 変化せず

●アドレスシング・モード

| SOU | size | dest |
|-----|------|------|
| | | # ~ |
| | | 2 16 |
| | | |
| | | |

* ソース・オペランド、ディスティネーション・オペランドという概念は存在しない。

* サイズは存在しない。

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |

●サンプル・リスト

RTS

108 ● TRAP

[Trap ト ラッ プ]

TRAP #<ベクタ番号>

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

プロセッサは例外処理を開始します。

本命令については、プロセッサの例外処理に関する的確な知識が要求されます。

指定ベクタ番号(0～15を指定する)と例外処理のエントリアドレスを保持する先頭アドレス(例外ベクタと呼ぶ)との対応表は、次に示す通りです。

| #<ベクタ> | アドレス(HEX) |
|--------|-----------|
| 0 | \$80 |
| 1 | \$84 |
| 2 | \$88 |
| 3 | \$8C |
| 4 | \$90 |
| 5 | \$94 |
| 6 | \$98 |
| 7 | \$9C |
| 8 | \$A0 |
| 9 | \$A4 |
| 10 | \$A8 |
| 11 | \$AC |
| 12 | \$B0 |
| 13 | \$B4 |
| 14 | \$B8 |
| 15 | \$BC |

CCR

X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●アドレスシング・モード

| squ | size | dest |
|----------|------|------|
| # | - | |
| #<ベクタ番号> | 2 34 | |

*サイズ、ソース／ディスティネーションという概念はまったく存在しない。

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |

→ ベクタ番号としては、0～15までの16通りを指定できる。

●サンプル・リスト

TRAP #15

109 ● TRAPV

[Trap on Overflow オーバフロー・トラップ]

TRAPV

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

Vビット(オーバフロー・ビット)がセットされている状態で本命令を実行すると、プロセッサは例外処理を開始しますが、そうでない場合は何もせず、単に次の命令に制御が移ります。

本命令による例外処理のエントリアドレスを保持する先頭アドレス(例外ベクタと呼ぶ)は、\$1Cです。

CCR

X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●アドレスシング・モード

| sou | size | dest |
|-----|------|-------------|
| # | - | |
| 2 | 34 | →トラップが発生した時 |
| 2 | 4 | →発生しなかった時 |

*サイズ、ソース／ディスティネーションという概念は、まったく存在しない。

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |

●サンプル・リスト

TRAPV

110 ●CHK

[Check Register Against Bounds レジスタ境界チェック]

CHK **l.W** <ea>, Dn

| X | N | Z | V | C |
|---|---|---|---|---|
| - | * | U | U | U |

解説

ディスティネーション・オペランドとして指定したデータレジスタの下位ワードとゼロ(0)を比較し、さらにソース・オペランドの上限値(下位ワード)を比較します。ただし、上限値は2の補数表現です。

比較結果によって例外処理を開始しますが、該当ケースは次の2つに限定され、3番目のケースではトラップは発生せず、単に次の命令に制御が移ります。

- ① Dn < 0 ……Nビットをセットして例外処理を開始
- ② Dn > <ea> ……Nビットをクリアして例外処理を開始
- ③ 0 ≤ Dn ≤ <ea> ……例外処理は開始されない

本命令による例外処理のエントリアドレスを保持する先頭アドレス(例外ベクタと呼ぶ)は、\$18です。

CCR

X : 変化せず

N : Dn < 0ならセット(1)、Dn > <ea>ならリセット(0)、それ以外は未定義(結果は保証されない)

Z : 未定義(結果は保証されない)

V : 未定義(結果は保証されない)

C : 未定義(結果は保証されない)

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|------|----|---|---|--------|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | レジスタ | 1 | 1 | 0 | 実効アドレス | | | | | | | |

実効アドレス(モード)

| アドレッシング モード | 対応ビット | | | | |
|----------------|-------|---|---|--------|---|
| | 5 | 4 | 3 | 2 | 1 |
| Dn | 0 | 0 | 0 | レジスタ番号 | |
| (An) | 0 | 1 | 0 | レジスタ番号 | |
| (An) + | 0 | 1 | 1 | レジスタ番号 | |
| -(An) | 1 | 0 | 0 | レジスタ番号 | |
| d16(An) | 1 | 0 | 1 | レジスタ番号 | |
| d8(An,IX) | 1 | 1 | 0 | レジスタ番号 | |
| Abs.W | 1 | 1 | 1 | 0 | 0 |
| Abs.L | 1 | 1 | 1 | 0 | 0 |
| d16(PC) | 1 | 1 | 1 | 0 | 1 |
| d8(PC,IX) | 1 | 1 | 1 | 0 | 1 |
| #Imm | 1 | 1 | 1 | 1 | 0 |

→ ディスティネーション・オペランドとして指定したデータレジスタ番号(000~111)

●アドレッシング・モード

| sou | size | dest |
|-----------|------|------------|
| | | # ~ |
| Dn | B | |
| | W | 2 <48 2 10 |
| | L | |
| (An) | B | |
| | W | 2 <48 2 14 |
| | L | |
| (An) + | B | |
| | W | 2 <48 2 14 |
| | L | |
| -(An) | B | |
| | W | 2 <48 2 16 |
| | L | |
| d16(An) | B | |
| | W | 4 <48 4 18 |
| | L | |
| d8(An,IX) | B | |
| | W | 4 <50 4 20 |
| | L | |
| Abs.W | B | |
| | W | 4 <48 4 18 |
| | L | |
| Abs.L | B | |
| | W | 6 <52 6 22 |
| | L | |
| d16(PC) | B | |
| | W | 4 <48 4 18 |
| | L | |
| d8(PC,IX) | B | |
| | W | 4 <50 4 20 |
| | L | |
| #Imm | B | |
| | W | 4 <48 4 14 |
| | L | |

トラップ発生

トラップが発生しなかったとき

●サンプル・リスト
CHK D0, D3
CHK (A2), D7

111 ●RTE [特権命令]

[Return from Exception
例外処理からのリターン]

RTE

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

例外処理からの復帰命令で、例外処理の入り口でスタックへ退避してあったシステム情報であるSRとPCを取り出し、例外発生前の状態へ戻します。つまり、取り出したPCの示すアドレスから処理を続行します。

命令実行前のSRとPCは失われ、SRの全ビットが影響を受けます。

CCR

T : スタック上のワードに対応するビット15(トレース)
の値

S : スタック上のワードに対応するビット13(スーパバイザ状態)の値

I₂ : スタック上のワードに対応するビット10(割り込みマスク)の値

I₁ : スタック上のワードに対応するビット9(割り込みマスク)の値

I₀ : スタック上のワードに対応するビット8(割り込みマスク)の値

X : スタック上のワードに対応するビット4の値

N : スタック上のワードに対応するビット3の値

Z : スタック上のワードに対応するビット2の値

V : スタック上のワードに対応するビット1の値

C : スタック上のワードに対応するビット0の値

| sou | size | dest |
|-----|------|------|
| | | # ~ |
| | | 2 20 |
| | | |

サイズ、ソース／ディスティネーションという概念は、まったく存在しない。

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |

●サンプル・リスト

RTE

112 ●RESET [特権命令]

[Reset external devices
外部デバイス・リセット]

RESET

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

MPUのリセット・ラインをアサート(アクティブ)します。

本命令により、プロセッサは124クロック・パルスの間リセット端子を駆動しますが、プロセッサの内部状態に影響を与えることはなく、内部レジスタ、SR、も変更されませんが、PCは本命令の次の命令から継続して命令を実行するため、通常の命令を実行するときのように変更されます。

CCR

X : 変化せず
N : 変化せず
Z : 変化せず
V : 変化せず
C : 変化せず

●アドレスシング・モード

| sou | size | dest |
|-----|------|------|
| # | ~ | |
| | 2 | 田 |
| | | |
| | | |

サイズ、ソース／ディスティネーションという概念は、まったく存在しない。

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

●サンプル・リスト

RESET

113 ●STOP [特権命令]

[Load Status Register and Stop
ステータスレジスタのロードとストップ]

STOP #<data>

| X | N | Z | V | C |
|---|---|---|---|---|
| * | * | * | * | * |

解説

プロセッサの実行を停止するための命令で、次のように動作します。

- ① 指定したイミディエイト値をSR全体に転送する。
ここでは、割り込みの優先度やトレース状態の設定をするが、Sビット（スーパーバイザ状態）が0なら、特権違反となる。
- ② PCは更新されて次の命令をポイントするが、プロセッサは命令の取り込み（フェッチ）及び実行を停止する。

ただし、STOP命令を実行する際にSRのT(トレース)ビットがセットされていれば、トレース例外処理を実行する。

再起動されるケースは次の2つです。

- ① 現在のプロセッサより高い優先度の割り込み要求が発生すると、割り込み例外が発生し、割り込み処理ルーチンを起動できるが、割り込みの優先度がプロセッサと同じか、低い優先度の割り込み要求は無視され、停止状態を保持する。
- ② 外部リセット

CCR

T : イミディエイト・オペランドに対応するビット15(トレース)の値
S : イミディエイト・オペランドに対応するビット13(スーパーバイザ状態)の値
I₂ : イミディエイト・オペランドに対応するビット10(割り込みマスク)の値
I₁ : イミディエイト・オペランドに対応するビット9(割り込みマスク)の値
I₀ : イミディエイト・オペランドに対応するビット8(割り込みマスク)の値
X : イミディエイト・オペランドに対応するビット4の値
N : イミディエイト・オペランドに対応するビット3の値
Z : イミディエイト・オペランドに対応するビット2の値
V : イミディエイト・オペランドに対応するビット1の値
C : イミディエイト・オペランドに対応するビット0の値

●アドレスシング・モード

| sou | size | dest |
|------|------|------|
| #Imm | # | ~ |
| | 4 | 4 |
| | | |
| | | |

サイズは存在せず、オペランドはディスティネーションだけが存在する。

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |

●第2ワード

| | |
|-------------|---|
| 15 | 0 |
| イミディエイト・データ | |



→ 指定したイミディエイト値が格納されるフィールドで、この値は、そのままSRへ代入される内容である。

●サンプル・リスト

STOP #MASK — MASKとはプログラマが定義した定数である

114 NOP

[No Operation ノー・オペレーション]

NOP

| X | N | Z | V | C |
|---|---|---|---|---|
| - | - | - | - | - |

解説

何も実行しません。

PCは本命令の次に位置する命令から継続して実行するために更新されますが、その他は、プロセッサの状態に何の影響も与えません。

CCR

- X : 変化せず
- N : 変化せず
- Z : 変化せず
- V : 変化せず
- C : 変化せず

●アドレスシング・モード

| sou | size | dest |
|-----|------|------|
| # | - | |
| | | 2 |
| | | 4 |
| | | |
| | | |

サイズ、ソース、ディスティネーションという概念は存在しない。

●機械語フォーマット

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |

●サンプル・リスト

NOP