

For the dest arg (Xn then M, bits 11-6), the bits aren’t completely reversed compared to M then Xn. It’s only the order of M and Xn that changes. For example, an arg that’s an address register always has M as 001, regardless of whether it’s the source or destination.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Bit number** | | | | | | | | | | | | | | | |
| **Name** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| NOP | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| MOVE | 0 | 0 | S | | Xn | | | M | | | M | | | Xn | | |
| MOVEM | 0 | 1 | 0 | 0 | 1 | D | 0 | 0 | 1 | S | M | | | Xn | | |
| ADD | 1 | 1 | 0 | 1 | Dn | | | D | S | | M | | | Xn | | |
| SUB | 1 | 0 | 0 | 1 | Dn | | | D | S | | M | | | Xn | | |
| AND | 1 | 1 | 0 | 0 | Dn | | | D | S | | M | | | Xn | | |
| MULS | 1 | 1 | 0 | 0 | Dn | | | 1 | 1 | 1 | M | | | Xn | | |
| DIVU | 1 | 0 | 0 | 0 | Dn | | | 0 | 1 | 1 | M | | | Xn | | |
| LEA | 0 | 1 | 0 | 0 | An | | | 1 | 1 | 1 | M | | | Xn | | |
| NOT | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | S | | M | | | Xn | | |
| LSL | 1 | 1 | 1 | 0 | 0 | 0 | 0 | D | 1 | 1 | M | | | Xn | | |
| ASL | 1 | 1 | 1 | 0 | 0 | 0 | 0 | D | 1 | 1 | M | | | Xn | | |
| LSL (alt) | 1 | 1 | 1 | 0 | Rotation | | | D | S | | M | 0 | 1 | Dn | | |
| ASL (alt) | 1 | 1 | 1 | 0 | Rotation | | | D | S | | M | 0 | 0 | Dn | | |
| BRA | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | Displacement | | | | | | | |
| BGT | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | Displacement | | | | | | | |
| BLE | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | Displacement | | | | | | | |
| BEQ | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | Displacement | | | | | | | |
| JSR | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | M | | | Xn | | |
| RTS | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |

Illegal modes: Dest Source

MOVE: An, #imm, PC (both) All modes are valid

MOVEM: Dn, An, (An)+, #imm, PC Dn, An, -(An), #imm

ADD(Dn,<ea>): Dn, An, #imm, PC An,(An),(An)+,-(An),(xxx).W,  
 (xxx).L,#imm,PC  
ADD(<ea>,Dn): An,(An),(An)+,-(An),(xxx).W, An, #imm  
 (xxx).L,#imm,PC

SUB: Same invalids as ADD (both forms)

AND(Dn,<ea>): Dn,An,#imm, PC An,(An),(An)+,-(An),(xxx).W,(xxx).L,#imm,PC

AND(<ea>,Dn): An,(An),(An)+,-(An),(xxx).W, An, #imm  
 (xxx).L,#imm, PC

MULS: An,(An),(An)+,-(An),(xxx).W, An  
 (xxx).L,#imm,PC

DIVU: An,(An),(An)+,-(An),(xxx).W, An  
 (xxx).L,#imm,PC

LEA: Dn, (An), (An)+, -(An), PC, #imm,  
 (xxx).W, (xxx).L Dn, An, (An)+, -(An), #imm

NOT: An, #imm, PC No source

LSL: Dn, An, #imm, PC (only in single arg) No source

ASL: Dn, An, #imm, PC (only in single arg) No source

JSR: N/A Dn, An, (An)+, -(An), #imm

**Edge Cases:**

ASd/LSd - ASL.W (A1) [1110 0001 1101 0001] {E1D1}

Template binary   
(ASd): [1110 000(d) 11 (mode) (reg)]  
(LSd): [1110 001(d) 11 (mode) (reg)]