

# Adressierungstechniken 68000

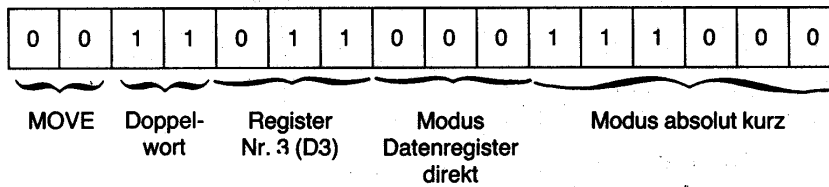
## Beispiel

### Direkte Adressierung

Bei dieser Adressierungsart ist der Operand in einem angegebenen Register als effektive Adresse (EA) enthalten. Dieses Register ist eines der acht 32-Bit Register  $D_n$

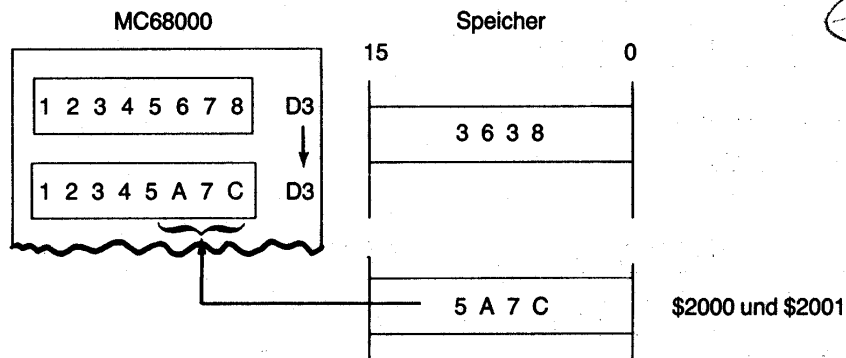
Beispiel, bei dem das **Datenregister der Zielooperand** der Übertragung ist:

MOVE        \$2000, D3                    ;übertrage das an Adresse \$2000 stehende Wort  
   ; ins Datenregister D3



Hexadezimalcode:

3638



**Beispiel** bei dem das **Adressregister der Zielperand** der Übertragung ist:

Es gibt ein 32 Bit langes Adressregister  $A_n$  das Zielperand ist

MOVE        \$305040, A<sub>2</sub>

Befehlscode: 0 0 1 1 0 1 0 0 0 1 1 1 1 0 0 1

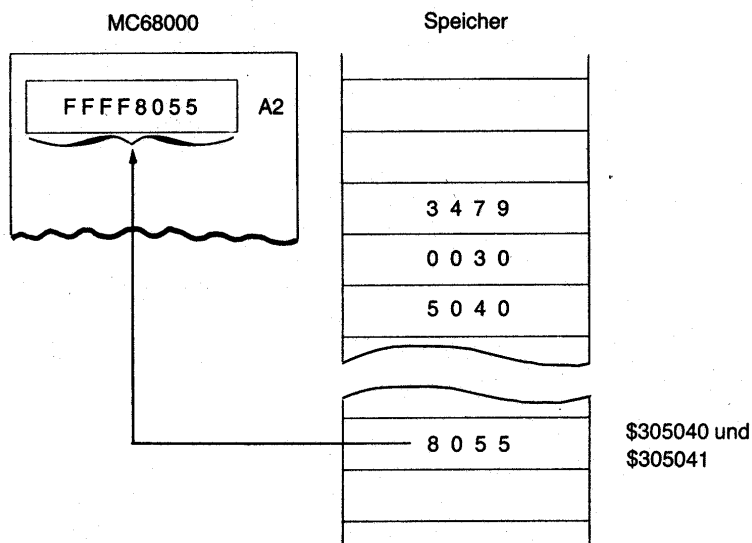
Hierbei gilt wie im obigen Beispiel von links nach rechts gelesen:

- 2 Bit: MOVE
- 2 Bit: Wortlänge .L
- 3 Bit: Register 2 (also A<sub>2</sub>)
- 3 Bit: Modus: Adressregister direkt
- 6 Bit: Moduls: absolut lang (Adressierung)

und gelangen damit zu folgender Darstellung:

Hexadezimalcode:

3479  
0030  
5040



**Anmerkung:**

Wenn eins der sieben 32-Bit-Adressregister das Ziel ist,

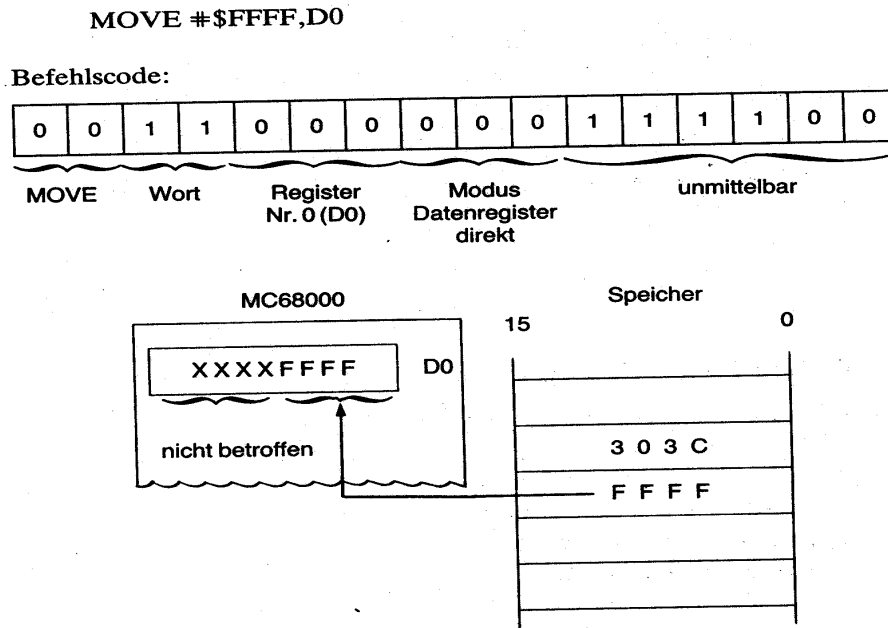
würde das oben bedeuten:        MOVE \$305040, A<sub>2</sub>

**Beispiel: Unmittelbare (imidiате) Adressierung**

Kodieren Sie den Transfer des Wertes \$FFFF ins Register D0

> Lösg. MOVE #\$FFFF, D0

Somit sehen wir:



**Beispiel-Aufgabe:**

Wie würde das Ergebnis aussehen, wenn statt des Registers D0 das Register A0 verwendet wird?

> Lösung:

MOVE #\$FFFF, A0

Würde wegen der vorzeichenbehafteten Erweiterung der führenden Zeichen

Im Register A0 nun FFFFFFFF ergeben!

### Beispiel Indirekte Adressierung

Kodieren Sie den Transfer des Inhaltes einer beliebigen Speicheradresse in ein beliebiges Datenregister.

➤ Lösung:

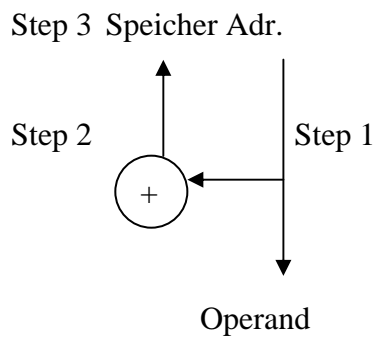
MOVE (A0), D0

Eine Adresse steht in A0; der Inhalt dieser Speicherezelle soll nun nach D0 gebracht werden.

### Beispiel Indirekte Adressierung mit Post-increment

Das Adressregister wird indirekt genutzt zB. bei Stapelverarbeitung / Stack

Adressregister: sei  $A_n+$



Inkrementwerte wären: N: 1,2,4 weil  
1 für Byte  
2 für Wort  
3 für Doppelwort

### Beispiel-Aufgabe:

Leiten Sie daraus Überlegungen ab, wie ein Pre-Decrement aussehen könnte ?

Mit: NOT.W      -(A0)      sowie mit      MOVE      -(A1), \$3000

Hinweis: Diese Technik wird zukünftig für den Umgang mit Unterprogrammen verwendet / STACK Technik

**Beispiel-Aufgabe:**

Register      D1=\$17832428

CCR=\$17(X=1, N=0, Z=V=C=1)

Welchen Inhalt hat D1 nach der Operation

MOVE.B    #\$80, D1

Lösung.              D1=\$178324**80**

**Beispiel-Aufgabe:**

Analysieren Sie das Programm und testen Sie im 68K Simulator

```
ORG $1000
MOVE.W    D1, $3115
END
```

Lösung.      Das Programm zeigt einen Fehler, denn es überträgt ein Wort an eine ungerade Adresse.

**Beispiel-Aufgabe:**

Analysieren Sie:      ORG            \$1000  
                         MOVE.B        A2, \$2000  
                         END

Lösung:      Bei dieser Operation, die ein Adressregister verwendet ist .B nicht erlaubt !!

### Beispiel-Aufgabe

Vor Ausführung: D0=\$00218013  
D1=\$00038065  
CCR=\$00

Somit:       8013  
              8065

10078        hexa !   folglich: N=0, Z=0, C=1, X=V=1

Nach Ausführung: D0=\$00218013  
D1=\$00030078  
CCR=\$13

Der Befehl ADD setzt den Bedingungscode gemäß dem Resultat der Operation.