Wissenswertes über binäre Felder

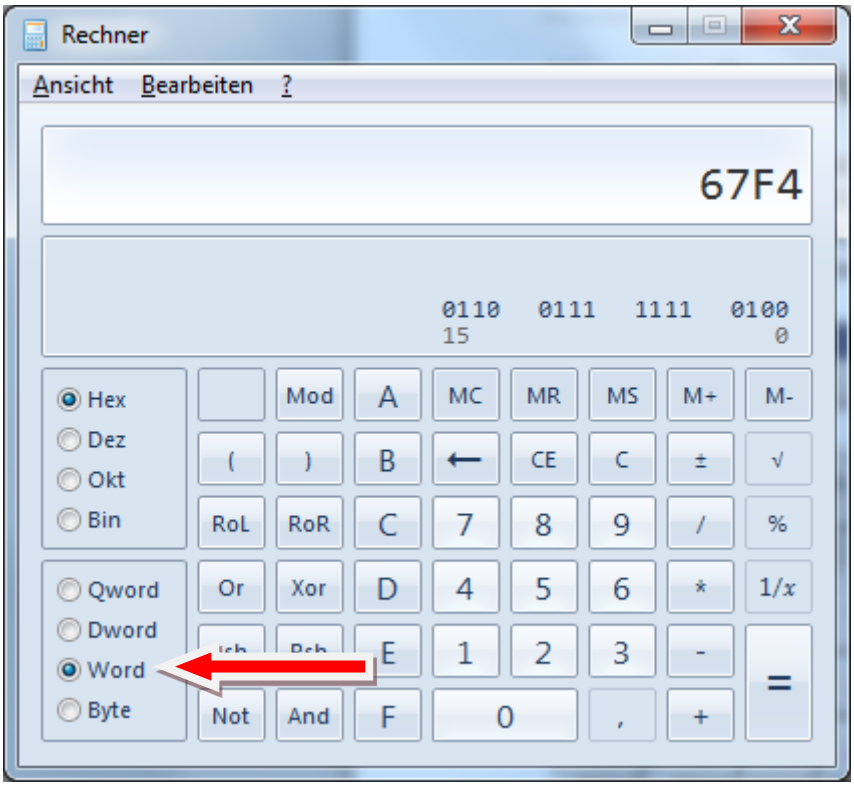
Inhaltsverzeichnis

Genauigkeit des PC-Taschenrechners 2
Ausgangspunkt 2
Binäres Feld ohne Vorzeichen-Definition 2
Binäres Feld mit Vorzeichen-Definition 3
Der binäre Kreis – ohne Vorzeichen – 2 Byte Feld (PIC 9(4) binary) 4
Der binäre Kreis – mit Vorzeichen – 2 Byte Feld (PIC S9(4) binary) 4
Umrechnen von (positiven) binären Feldern in dezimale Werte 5
Umrechnen von (negativen) binären Feldern in dezimale Werte 6
Umrechnen von positiven binären Feldern > 32767 in dezimale Werte 7

Eine Ausarbeitung von:

Ralf Seidler • Stromberger Straße 36A • 55411 Bingen
Fon: +49-6721-992611 • Fax: +49-6721-992613 • Mail: ralf.seidler@cps4it.de
Internet : http://www.cps4it.de
Steuernummer: 08/220/2497/3, Finanzamt Bingen, Ust-ID : DE214792185

Genauigkeit des PC-Taschenrechners



Ausgangspunkt

```
F-BIN     PIC  9(04) BINARY.
```
Das Feld belegt 2 Bytes.
```
F-BIN-99  PIC  9(02) BINARY
```
Das Feld belegt 2 Bytes.

Bei `PIC  9(02) BINARY` hat der Compiler zusätzliche Instruktionen, um das Feld künstlich zu „kürzen“. Sehr inperformant.

Binäres Feld ohne Vorzeichen-Definition

```
F-BIN     PIC  9(04) BINARY.
```
Zahlenbereich bei 2 Bytes: 0 bis 65535
Darstellung: 0000 bis FFFF.

Beispiele:

```
0000(16) =     0(10)

0001(16) =     1(10)

0011(16) =    17(10)

324A(16) = 12874(10)

FFF0(16) = 65520(10)
```

Binäres Feld mit Vorzeichen-Definition

```
F-BIN     PIC S9(04) BINARY.
```

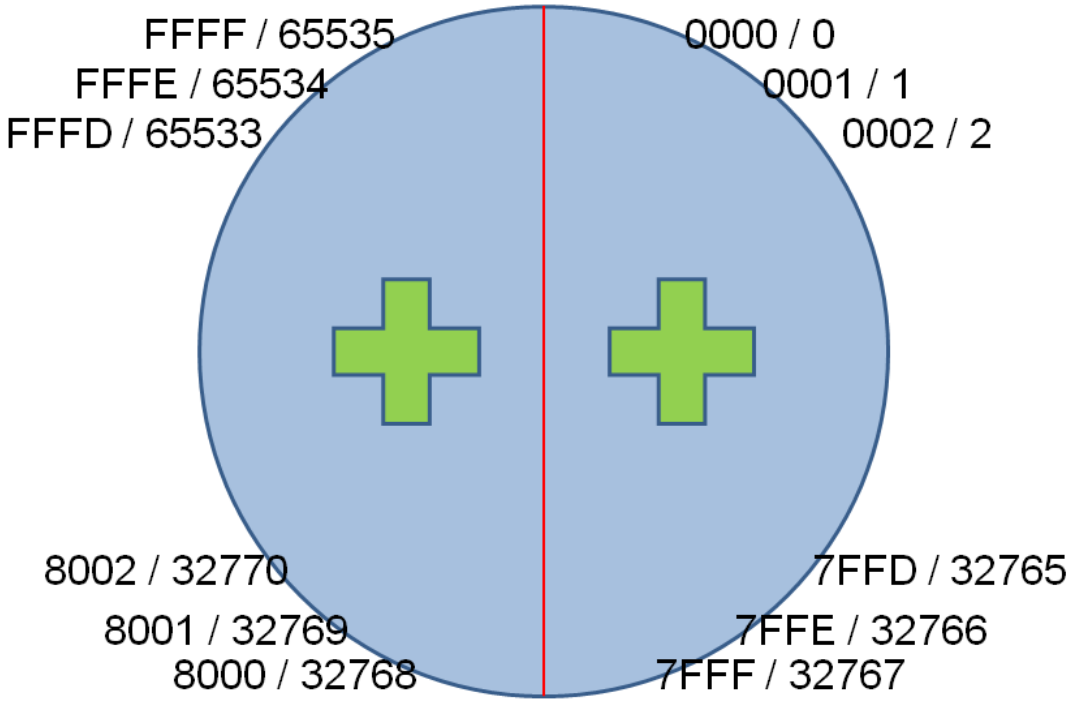
Zahlenbereich bei 2 Bytes: $-32768_{(10)}$ bis $-1_{(10)}$ / $+0_{(10)}$ bis $+32767_{(10)}$
Darstellung: $8000_{(16)}$ bis $FFFF_{(16)}$ / $0000_{(16)}$ bis $7FFF_{(16)}$.
Also: das erste Bit wird zur Kennzeichnung des Vorzeichens benutzt.

```
8001(16)   + 1(16) =   8002(16) / -32767(10) + 1(10) = -32766(10)
8002(16)   + 1(16) =   8003(16) / -32766(10) + 1(10) = -32765(10)
etc. bis …
8FFF(16)   + 1(16) =   9000(16) / -28672(10) + 1(10) = -28671(10)
etc. bis …
FFFD(16)   + 1(16) =   FFFE(16) /     -3(10) + 1(10) =     -2(10)
FFFE(16)   + 1(16) =   FFFF(16) /     -2(10) + 1(10) =     -1(10)
FFFF(16)   + 1(16) =   0000(16) /     -1(10) + 1(10) =      0(10)
```

Analoges gilt für 4 Byte lange binäre Felder mit der Definition PIC S9(08) BINARY.

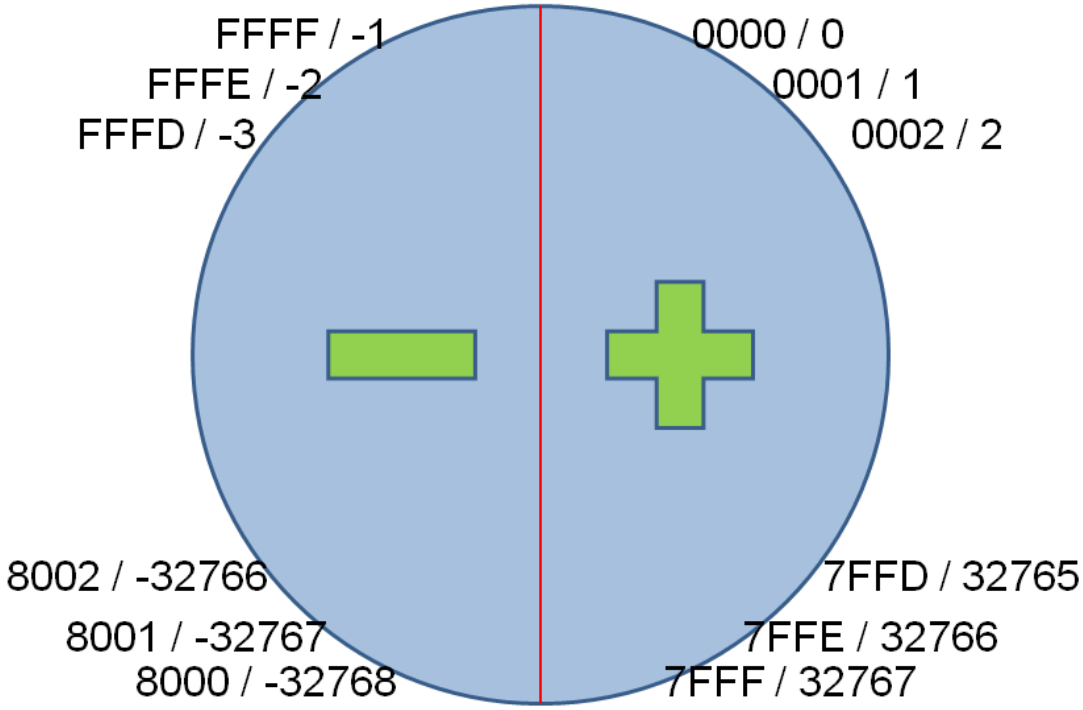
Der binäre Kreis – ohne Vorzeichen – 2 Byte Feld (PIC 9(4) binary)

Hinweis: Zu dieser Definition in COBOL gibt es keine Entsprechung in PL/1 und DB2.



Der binäre Kreis – mit Vorzeichen – 2 Byte Feld (PIC S9(4) binary)

Hinweis: PL/1: fixed bin(15), DB2: smallint

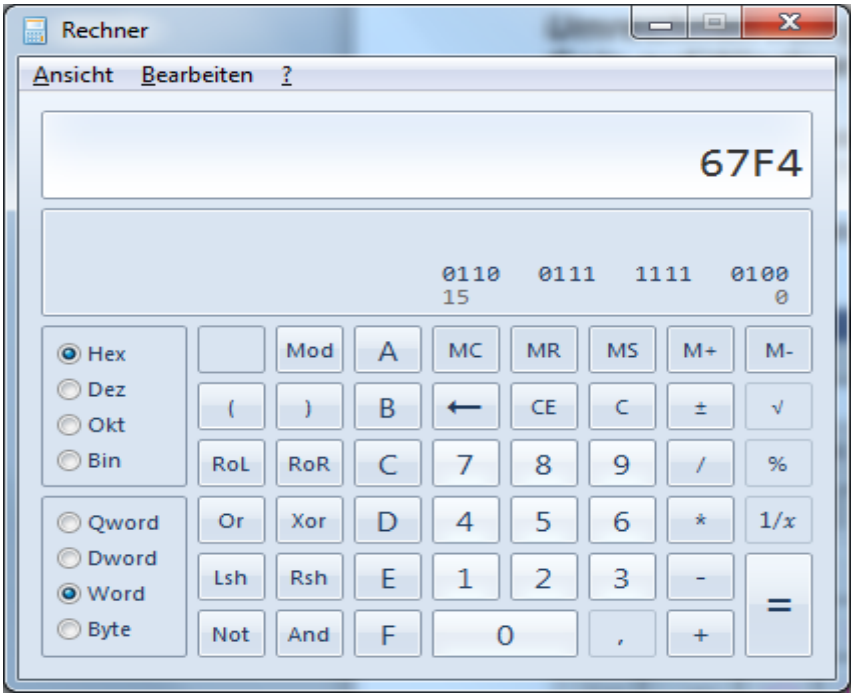


Umrechnen von (positiven) binären Feldern in dezimale Werte

unsere Ausgangsdefinition:

```
F-BIN      PIC S9(04) BINARY.
```

Nehmen wir an, wir haben im Dump für dieses Feld den (positiven) Wert $67F4_{(16)}$ gefunden.
Eingabe in Rechner, der auf Ansicht „Programmierer“ eingestellt ist:



Umschalten auf dezimal:

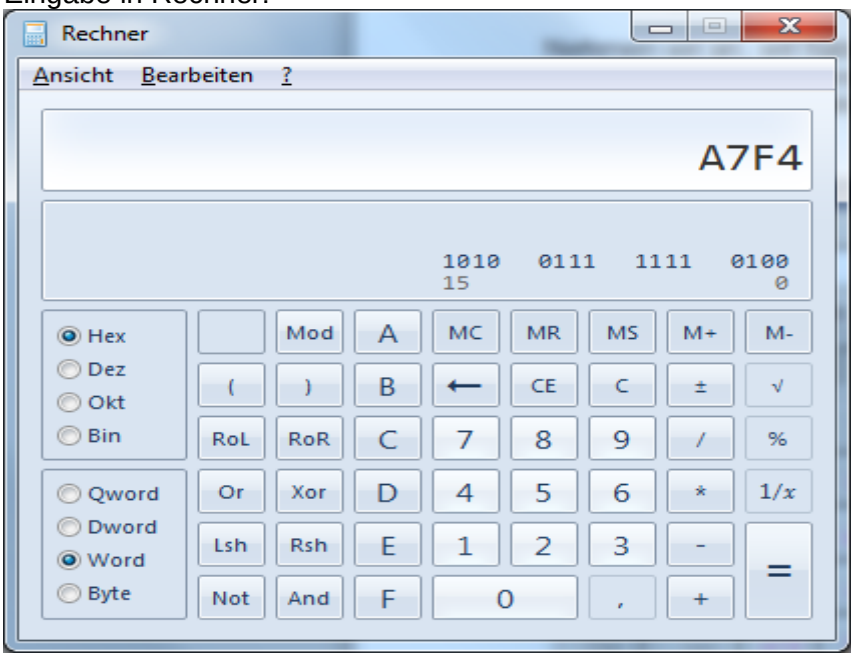
Wir haben also den Wert $26612_{(10)}$.

Umrechnen von (negativen) binären Feldern in dezimale Werte

Mit der neuen Windowsversion wurde durch die Ansicht „Programmierer“ in Calc eine Einstellung aufgeliefert, die eine trickreiche Umrechnung in negative binäre Werte überflüssig macht.

Nehmen wir an, wir haben im Dump für dieses Feld den (negativen) Wert $A7F4_{(16)}$ gefunden. Zur Erinnerung: Das erste Bit zeigt an, ob es eine negative Zahl ist. Ist die erste Hexa-Zahl 8 oder größer, ist die Zahl also negativ.

Eingabe in Rechner:



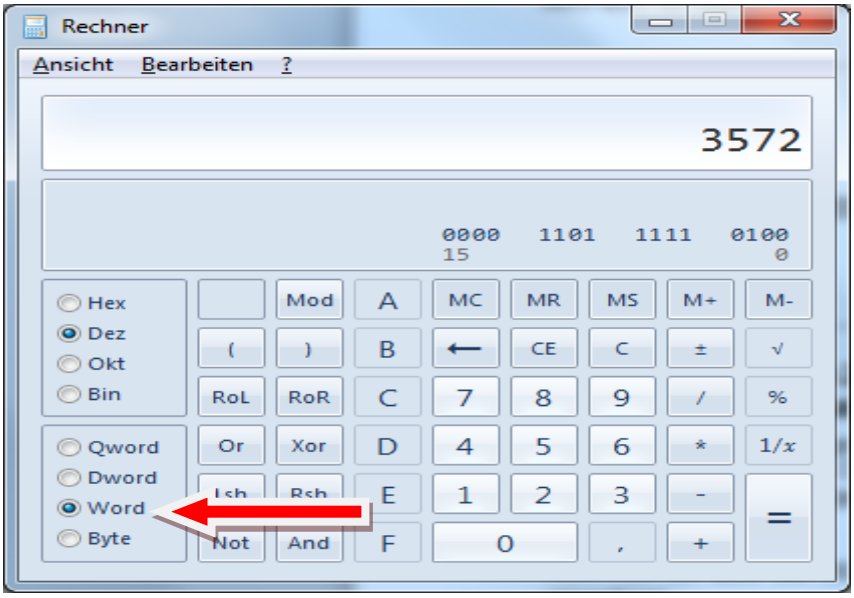
Nach dezimal wechseln:

$A7F4_{(16)}$ ist dann $-22540_{(10)}$

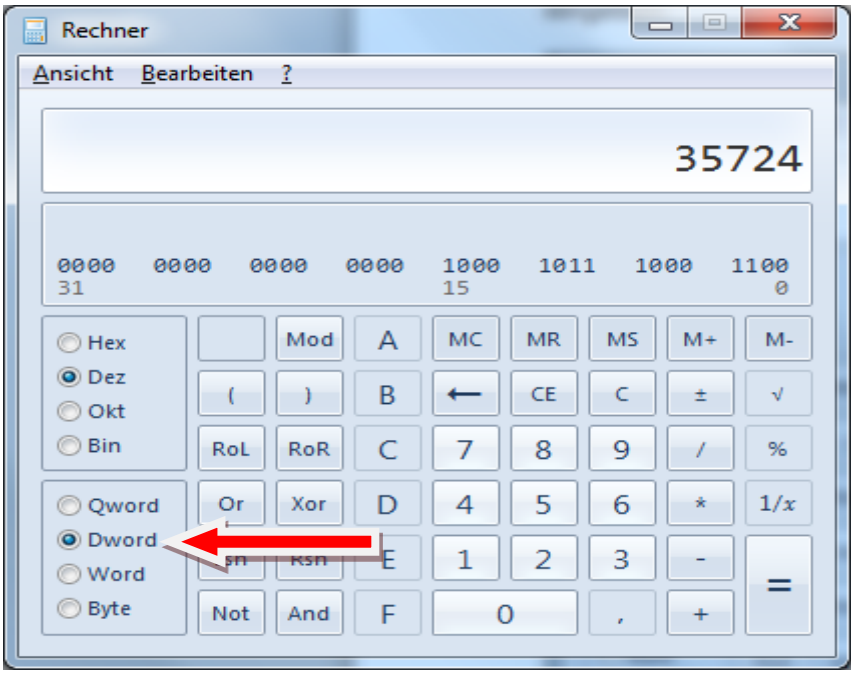
Umrechnen von positiven binären Feldern > 32767 in dezimale Werte

Dass die Ansicht „Programmierer“ die automatische Erkennung von Vorzeichen beinhaltet heißt aber auch, dass COBOL-Felder, die größer als 32767 sind, mit der Definition PIC 9(04) binary nicht richtig dargestellt werden können. Zur Erinnerung: Zahlen > 32767 haben mindestens eine 8 auf dem ersten Halbbyte.

Nehmen wir also die $35724_{(10)}$. Diese Zahl können wir bei einem 2-Byte-Feld nicht (mehr) eingeben:



Wir müssen dann den Umweg über eine „längere“ Zahl gehen:



Und dann erhalten wir das „richtige“ Ergebnis: