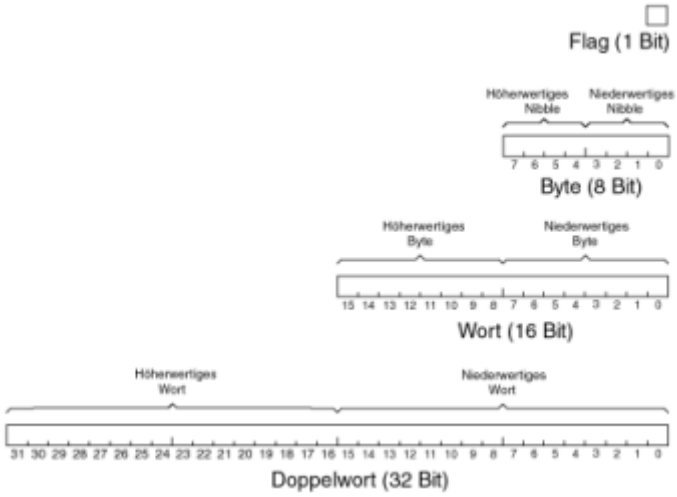
Register und Speicherbereiche

Ein Register ist eine Gruppe von Flipflops (1 Bit-Speicher) mit gemeinsamer Steuerung. Register umfassen meist 8, 16, 32 oder 64 Bit.

- 1 Bit – 1 Flag
- 4 Bit – 1 Nibble
- 8 Bit – 1 Byte
- 16 Bit – 1 Wort (bei 80686 Prozessoren, das hat historische Gründe)



Die Bits der Register sind numeriert, vom LSB (LeastSignificantBit) zum MSB (MostSignificantBit).

Die Register des 80x86

1985 kommt Intels 8086-Prozessor auf den Markt, der ersten 16-Bit-Prozessor. (Das Hauptrechenregister hatte 16 Bit, daher die Einheit "Wort" s.o.)

Auf dem Weg zu den heutigen x86 Prozessoren wurde diese Architektur stets erweitert – bis auf heute 64 Bit.

Das folgende Bild zeigt die Register für die 32Bit Prozessoren:

Registernamen beginnen mit einem E für "extended", weil diese Register von 16 auf 32 Bit erweitert wurden.

Für diese acht Register gilt, dass jeweils die unteren 16 Bit unter dem Namen des früheren 16-Bit-Registers angesprochen werden können.

Damit haben die neueren Prozessoren stets alle Register ihrer Vorgänger, nutzen deren Fähigkeiten aber nicht aus.

Bei den vier Allzweckregistern EAX, EBX, ECX und EDX lassen sich die unteren 16 Bit als AX, BX, CX und DX ansprechen, und diese zusätzlich auch noch byteweise als AL und AH, BL und BH, CL und CH, DL und DH (L für „Low“, H für „High“).

Bei den aktuellen 64Bit Prozessoren gibt es die Register rax, rbx ... die wiederum die eax, ebx Register „beinhalten“.

- **AX** ist der primäre Akkumulator; er wird bei der Eingabe/Ausgabe und den meisten arithmetischen Anweisungen verwendet. Bei einer Multiplikationsoperation beispielsweise wird ein Operand je nach Größe des Operanden im Register EAX oder AX oder AL gespeichert.
- **BX** wird als Basisregister bezeichnet, da es bei der indizierten Adressierung verwendet werden kann.
- **CX** wird als Zählregister bezeichnet, da in den Registern ECX und CX die Schleifenanzahl bei iterativen Operationen gespeichert wird.
- **DX** wird als das Datenregister bezeichnet. Es wird auch bei Eingabe-/Ausgabeoperationen verwendet. Es wird auch zusammen mit dem AX-Register und DX für Multiplikations- und Divisionsoperationen mit großen Werten verwendet.

Flags

Das EFlag-Register unterscheidet sich von an den anderen Registern. Die Flipflops in diesen Registern

werden einzeln gesteuert, jedes Flipflop hat eine bestimmte Bedeutung – es ist ein Flag (Flagge, Fähnchen).

Sprechweise:

- „Flag gesetzt“ bedeutet Flag=1; auch „ein Flag setzen“
- „Flag gelöscht“ bedeutet Flag=0; auch: „der Befehl löscht das Flag“

Es gibt zwei Gruppen von Flags: Statusflags und Steuerflags.

Statusflags sind Flags, die der Prozessor nach arithmetischen oder bitweise logischen Operationen setzt, um etwas über das Resultat dieser Operation auszusagen.

Der Programmierer kann diese Flags dann in bedingten Sprungbefehlen abfragen und Programmverzweigungen vom Zustand der Flags abhängig machen.

Statusflags im Detail

Speicherbereiche reservieren

Im Abschnitt .data werden Speicherbereiche mit Vorbelegung definiert, im Abschnitt .bss solche, die nicht initialisiert werden.

Bei der Deklaration muss angegeben werden, wie groß die Bereiche sein sollen.

.data	.bss	Size
db	resdb	1 Byte
dw	resdw	2 Byte
dd	resdd	4 Byte
dq	resdq	8 Byte

Beispiele:

- Zaehler2 DB 0 ;Def. der Byte-Variablen Zaehler2, Vorbesetzung mit 0
- Startchar DB 65 ;Vorbesetzung mit ASCII-Zeichen #65 = ’A’
- Startchar DB ’A’;gleiche Wirkung, besser lesbar
- Regmaske DB 00110101b ;Vorbesetzung mit binärem Wert (Bitmuster)
- Pixely DW 01AFh ;Wort-Variable, Vorbesetzung mit hexadezimalem Wert

Mit einer Anweisung können auch gleich mehrere Speicherplätze gleichen Typs, also Felder , definiert werden:

- Meldung1 DB ’Divisionsfehler! ;Vorbesetzung mit einer Zeichenkette,das Feld erhält 16 Byte Speicherplatz
- Meldung1 DB ’Hallo Welt’,13,10 ;Vorbesetzung mit einer Zeichenkette, und Steuerzeichen, 12

Byte Speicherplatz

Das haben wir beim Hallo-Welt Programm gemacht.

In der .bss-Sektion sieht dass folgendermaßen aus:

```
antwort resb 5 ; 5 beschreibbare Bytes ohne spezielle Vorbelegung.
```

assembler03.odp	336.2 KiB	13.09.2021 17:55
assembler03.pdf	287.4 KiB	13.09.2021 17:55
auswahl_400.png	23.6 KiB	13.09.2021 16:26
auswahl_401.png	98.0 KiB	13.09.2021 16:29

From:
https://info-bw.de/ -

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:techinf:assembler:register:start

Last update: **13.09.2021 17:55**