

Handout zum Programmierkurs II (Assembler) Zentrale Register einer 8051 CPU

Einleitung

Dieses Handout fasst die zentralen Register des 8051 CPU-Kerns zusammen, wie sie auch vom C504 verwendet werden. Dies sind:

- Akkumulator **A** (zentrales Register zum Rechnen und für den Datentransfer)
- Hilfsakkumulator **B**
- Programmstatuswort **PSW**
- Stackpointer **SP**
- Datenzeiger **DPTR** (zur Adressierung des externen Speichers)
- Programmzähler **PC** (kann nur indirekt über Sprungbefehle manipuliert werden)

Akkumulator A

Der Akkumulator nimmt beim 8051 eine zentrale Stellung ein. Sämtliche Berechnungen und Datentransfers benutzen den Akkumulator. Für arithmetische oder logische Operationen muss einer der Operanden im Akkumulator bereitgestellt werden. Das Ergebnis wird wiederum im Akkumulator abgelegt. Im Programmquelltext wird der Akkumulator mit **A** bezeichnet.

Das Akkumulatorregister kann auch im internen Speicher angesprochen werden, außerdem sind seine Bits einzeln adressierbar.

| Name | 2 ⁷ | 2 ⁶ | 2 ⁵ | 2 ⁴ | 2 ³ | 2 ² | 2 ¹ | 2 ⁰ | Adresse |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|---------|
| ACC | ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 | E0h |
| | E7h | E6h | E5h | E4h | E3h | E2h | E1h | E0h | Bitadr. |

Hilfsakkumulator B

Speziell für die Multiplikation und die Division existiert der Hilfsakkumulator **B**. In **B** muß bei der Multiplikation einer der Faktoren abgelegt werden. Falls das Produkt größer als 255 (0xFF) ist, dann enthält **B** nach der Multiplikation das höherwertige Byte des Ergebnisses. Bei einer Division enthält **B** nach Ausführung der Operation den ganzzahligen Divisionsrest.

| Name | 2 ⁷ | 2 ⁶ | 2 ⁵ | 2 ⁴ | 2 ³ | 2 ² | 2 ¹ | 2 ⁰ | Adresse |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|---------|
| B | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 | F0h |
| | F7h | F6h | F5h | F4h | F3h | F2h | F1h | F0h | Bitadr. |

Programmstatuswort PSW

Im **PSW** sind mehrere Kennzeichnungsbits zusammengefaßt, die Auskunft über den Zustand der CPU oder das Ergebnis einer Operation geben. Das **PSW** ist bitadressierbar. Die Flags des **PSW** haben folgende Bedeutung:

| | |
|-------------------------------|--|
| CY (Carry) | Übertragsbit, zeigt Bereichsüberschreitungen bei Addition oder Subtraktion an |
| AC (auxiliary carry) | Hilfsübertrag für die Addition von BCD-Zahlen. |
| F0 (flag 0) | Die CPU nutzt F0 nicht, der Benutzer kann es als Bitvariable benutzen. |
| RS1 RS0 (registerbank select) | Auswahl von einer aus 4 Registerbänken (Im Assemblerkurs nicht nötig, der Monitor kümmert sich automatisch darum.) |
| OV (overflow) | Zeigt Vorzeichenwechsel bei Addition im 2er-Komplement an Falls OV nach einer Division gesetzt ist, so war der Quotient 0 und das Ergebnis ist undefiniert |
| P (parity) | P ist gesetzt, falls A eine gerade Anzahl Bits enthält |

| Name | 2 ⁷ | 2 ⁶ | 2 ⁵ | 2 ⁴ | 2 ³ | 2 ² | 2 ¹ | 2 ⁰ | Adresse |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|---------|
| PSW | CY | AC | F0 | RS1 | RS0 | OV | --- | P | D0h |
| | D7h | D6h | D5h | D4h | D3h | D2h | D1h | D0h | Bitadr. |

Stackpointer SP

Der Stackpointer **SP** beim 8051 ist nur 8 Bit breit, da der Stack im internen Speicher angelegt wird und dort 8 Bit zur Adressierung reichen. Der Stack wird zur Ablage von Rücksprungadressen benutzt. Eine explizite Verschiebung des Stacks ist im Assemblerkurs nicht nötig. Auch kann der **SP** nicht bitadressiert werden.

| Name | 2 ⁷ | 2 ⁶ | 2 ⁵ | 2 ⁴ | 2 ³ | 2 ² | 2 ¹ | 2 ⁰ | Adresse |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|---------|
| SP | SP.7 | SP.6 | SP.5 | SP.4 | SP.3 | SP.2 | SP.1 | SP.0 | 81h |

Datenzeiger DPTR

Der sogenannte „Data Pointer“ (**DPTR**) ist 16 Bit breit und wird benutzt, um Daten im externen Datenspeicher indirekt zu adressieren. Der Benutzer hat in der Regel keinen direkten Zugriff auf die vollen 16 Bit, aber über 2 getrennte Register kann das höherwertige und das niederwertige Byte des **DPTR** manipuliert werden. Der **DPTR** zeigt immer auf einen 8 Bit Wert in einer Speicherstelle im Adressbereich von 0x0000 bis 0xffff.

Eine 16 Bit Adresse kann mit folgendem Befehl direkt in den **DPTR** geladen werden:

```
mov dptr, #<label>
```

| Name | 2 ⁷ | 2 ⁶ | 2 ⁵ | 2 ⁴ | 2 ³ | 2 ² | 2 ¹ | 2 ⁰ | Adresse |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|---------|
| DPL | DP.7 | DP.6 | DP.5 | DP.4 | DP.3 | DP.2 | DP.1 | DP.0 | 82h |
| DPH | DP.15 | DP.14 | DP.13 | DP.12 | DP.11 | DP.10 | DP.9 | DP.8 | 83h |