



Die wichtigsten Register des 8051 sind die Core-Register. Dazu gehören unter anderem die bitadressierbaren Register:

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
ACC	E0h	Accumulator								
B	F0h	B Register								
PSW	D0h	Program Status Word	CY	AC	F0	RS1	RS0	OV	F1	P

Dies sind im Einzelnen:

- Der **Akkumulator** (ACC) ist das wichtigste Rechenregister. Bei Arithmetikbefehlen dient der Akku in der Regel als Quell- und Ergebnisoperand.

Beispiele:

```

add    a, #20           ; acc = acc + 20
rl     a                ; rotiert die Akkubits nach links
anl   a, #00110010b    ; acc = acc && 00110010
  
```

- Das **B-Register** (B) dient als Operandenregister für 8-Bit Multiplikation und Division.

Divisionsbeispiel:

```

mov    a, #Divident
mov    b, #Divisor
div    ab                ; acc = acc / b -> immer Ganzzahlig
                        ; b = mod(acc / b) -> Divisionsrest
  
```

- Das **Programm-Status-Wort** (PSW) beinhaltet eine Reihe von Statusbits (Flags), die den aktuellen Zustand der CPU widerspiegeln. Dies sind im einzelnen:

Flag	Bezeichnung	Beschreibung
CY	Carry-Flag	Übertragsbit für Addition oder Borger für Subtraktion
AC	Auxiliary-Carry	Hilfsübertrag für BCD-Operationen
F0, F1	User-Flags	Frei programmierbare Flags
RS1, RS0	Registerbank Select-Bits	Auswahl der Registerbank R0...R7 Resetwert: RS1=0, RS0=0 → Bank 0 (00h ... 07h)
OV	Overflow-Flag	Multiplikation: OV = 1 → Ergebnis Highbyte in B-Register Division: OV=1 → Division durch Null !
P	Parity-Flag	Das Paritätsbit zeigt nach jedem Befehl an, ob der Akkuinhalt gerade (even) (P=0) oder ungerade (odd) (P=1) ist.

Beispiel für die Verwendung des Overflow-Flags:

```

div    ab
jb     OV, div_by_zero ; OV=1: Divisor (B) = Null
  
```

 Friedrich-Ebert-Schule Esslingen	<b>MIKROCONTROLLER</b>	Name:
<b>2.2.6</b>	<b>Wichtige Register des 8051</b>	Datum:

- Der **Data Pointer** (DPTR) besteht intern aus den 8 Bit -Registern DPH und DPL und dient dazu, eine 16-Bit Adresse zu speichern. Er wird z.B. bei Lesezugriffen auf den Programmspeicher verwendet.  
Das Beispiellisting zeigt wie die symbolische Adresse KONST (0007h) im DPTR gespeichert wird und anschließend der Inhalt dieser Speicheradresse (ASCII-Code 'A' = 41h) in den Akku geladen wird.

Adresse		Maschinencodes	Mnemonics
LOC	OBJ	LINE	SOURCE
		1	INCLUDE AT89C5131.INC
		243	
		244	;
		245	;
0000	900007	246	MOV DPTR, #KONST
0003	E4	247	CLR A
0004	93	248	MOVC A, @A+DPTR
0005	F5A0	249	MOV P2, A
		250	;
		251	;
		252	;
		253	
0007	41	254	KONST: DB 'A'

- Der **Programmzähler** (PC, Programmcounter) hält immer die Adresse (16 Bit) des als nächstes auszuführenden Befehls im Programmspeicher. Nach jedem Befehl wird der **PC**, vom Steuerwerk, auf die nächste Adresse erhöht.  
Er wird insbesondere von Sprung- und Call-Befehlen für die Programmsteuerung verwendet. Kann aber ansonsten nicht direkt manipuliert werden.  
Im Beispiellisting wird mit dem LJMP-Befehl (Long Jump) eine 16 Bit Adresse in den **PC** geschrieben. Die Programmausführung setzt anschließend bei dieser Adresse fort.

Adresse		Maschinencodes	Mnemonics
LOC	OBJ	LINE	SOURCE
		1	INCLUDE AT89C5131.INC
		243	
		244	
		245	EINGABE:
0000	E590	246	MOV A, P1 ; P1 in Akku einlesen
0002	0201CF	247	LJMP AUSGABE
		248	;
		249	;
		250	;
		251	;
		252	ORG 01CFH ; Anfangsadresse Ausgabe
		253	; festlegen
		254	AUSGABE:
01CF	F5A0	255	MOV P2, A ; Akkuinhalt auf P2 ausgeben
		256	