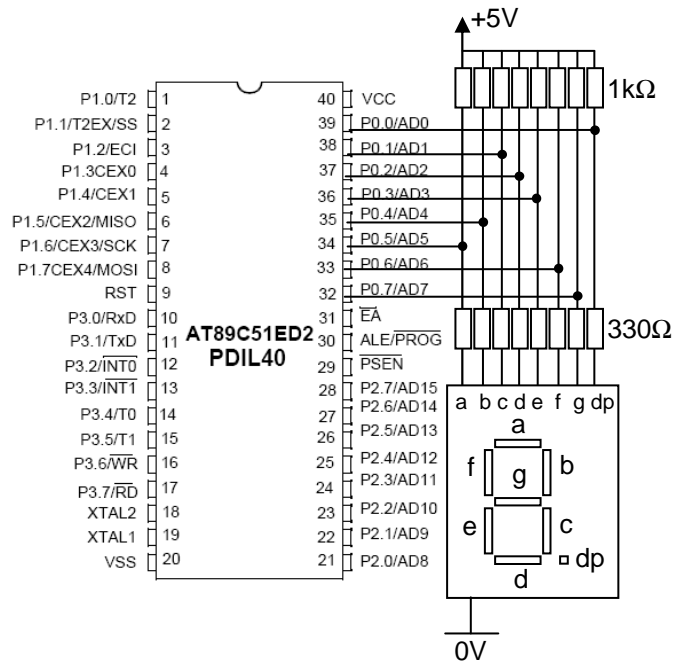


Port	p0.7	p0.6	p0.5	p0.4	p0.3	p0.2	p0.1	p0.0	P0
Segment									hex
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
A									
B									
C									
D									
E									
F									



Die beiden 7-Segment-Anzeigen des Miniboards sind nach dem oben dargestellten Schaltbild an Port 0 (Rechte Anzeige) und Port 2 (Linke Anzeige) des μ Cs angeschlossen. Die einzelnen Segmente beider Anzeigen belegen die gleichen Portbits. Ermittle für Port 0 den **Sieben-Segment-Code!**

Arbeitsaufträge:

1. Ergänze in der Tabelle die Segmentnummern (a, b, c ...) in der entsprechenden Spalte (p0.7, ..)
2. Fülle die Segmente für die jeweiligen Ziffern mit **Bleistift** aus.
3. Ergänze die Spalten mit den Segmentcodes: **0** = Segment **aus**
1 = Segment **ein**
4. Teste deine Codetabelle mit dem abgebildeten Testprogramm
5. Analysiere die Funktion des Befehls: **movc a,@a+dptr**
Informiere dich mit der Befehlsliste oder auf <http://mikrocontroller.rahm-home.de/Befehle.htm>
6. Das Highnibble von P1 soll auf der linken Anzeige (P2) ausgegeben werden.
Hinweis: mit dem Befehl **swap a** werden High-und Lownibble des Akkus vertauscht!

Testprogramm:

```

; Autor:          Rahm, 27.4.2010
; Beschreibung:   Das niederwertige Nibble von P1 wird als Hexziffer auf P0
;                ausgegeben!
;*****
include at89c5131.inc

code at 0

mov    p0,#0      ; Anzeigen dunkelschalten
mov    p2,#0

start: mov    dptr,#siebenSeg ; Adresse der 7-Segment-Codetabelle
loop:  mov    a,p1
      anl    a,#00001111b ; Highnibble löschen!

      movc   a,@a+dptr ; hole 7Segment-Code aus Codetabelle
      mov    p0,a      ; Ausgabe des 7-Segment-Codes an rechter Anzeige (P0)

      jmp   loop      ; sonst, beginne wieder bei start

;**** 7 Segment Codetabelle ****
siebenSeg:
db     7Eh,12h,...

end

```