

Hausübung 1 Rechnerarchitektur (16 Punkte)

Hinweise: Schreiben Sie die Lösungen, so weit es möglich ist, auf die Aufgabenblätter. Tragen Sie Name, Matrikelnummer und Studiengang in die nachfolgende Tabelle ein und schreiben Sie auf jedes zusätzlich abgegebene Blatt ihre Matrikelnummer.

Name	Matrikelnummer	Studiengang	Punkte von 16	≥ 40%

Aufgabe 1: Untersuchen Sie in Analogie zu den Folien 13 bis 16 von Foliensatz 1, was bei der Abarbeitung des dritten Befehls auf Folie 13 passiert und füllen Sie dazu die nachfolgende Tabelle aus: 5P

Lade Operand 1:	DA:	D:
Lade Operand 2:	DA:	D:
Operation / Ergebnis:	O:	=
Schreibe Ergebnis:	DA:	D:
Hole Befehlsword:	BA:	D:

(BA /DA- Befehls- bzw. Datenadresse; D – Befehlsword oder übertragende Daten; O – ausgeführte Operation und Ergebnis).

Aufgabe 2: Suchen Sie in der AVR-Befehlssatzbeschreibung AVR_Instructions.pdf auf der Web-Seite der Vorlesung die Befehle für

- die Addition zweier Registerinhalte mit Übertrag (adc)
- die bitweise ODER-Verknüpfung von zwei Operanden (and)
- die Multiplikation zweier Registerinhalte (mul).

Wie lauten für diese Befehle die Operationscodes und in welchen Bits des Befehlswords stehen die Adressen der Operanden? 3P

	Operationscode	Bitnummern der Adresse Rd*	Bitnummern der Adresse Rr*
adc			
or			
mul			

(* – wenn die Registeradresse in den Befehlswordbits 15 bis 10 steht, soll in die Tabelle als Aufzählung der Bitnummern 15, 14, ..., 10 eingetragen werden.)

Aufgabe 3: Das nachfolgende Programm liest ein Byte von Port A, führt Berechnungen durch und gibt das Ergebnis auf Port B aus. Ergänzen Sie im Programmkommentar für jede Zeile, welchen Wert die Anweisung dem Zielregister zuweist, wenn der von Port A eingelesene Wert 0x93 ist? 8P

```
9   main:
10  in r16, 0      ; r16: 0b1001 0011
11  mov r17, r16  ; r17: 0b
12  lsr r17       ; r17: 0b
13  andi r16, 0x50 ; r16: 0b
14  andi r17, 0x50 ; r17: 0b
15  mov r18, r17  ; r18: 0b
16  and r18, r16  ; r18: 0b
17  or r17, r16   ; r17: 0b
18  lsl r17       ; r17: 0b
19  or r17, r18   ; r17: 0b
20  andi r17, 0xF0 ; r17: 0b
21  mov r16, r17  ; r16: 0b
22  swap r16      ; r16: 0b
23  com r16       ; r16: 0b
24  andi r16, 0x0F ; r16: 0b
25  or r17, r16   ; r17: 0b
26  out 8, r17 ; Ausgabe: 0b
```