

Name:

Punkte: ___/31P

Datum:

Note:

1) Theorie Rechnerarchitektur

1.a) 0-Adressarchitektur

___/3P

Eine gegebene Stackmaschine unterstützt die Befehle push, add und mult. Welche Befehle werden ausgeführt, um die Berechnung $(5+2)*3+1$ auszuführen?

1.b) CISC vs. RISC

___/4P

Ordne die folgenden (typischen) Eigenschaften der jeweiligen Architektur zu

	CISC	RISC
Befehlsausführung meist in einem Takt	<input type="checkbox"/>	<input type="checkbox"/>
Steuerwerk wird oft mittels Mikrocode realisiert	<input type="checkbox"/>	<input type="checkbox"/>
Es gibt meist nur ein oder zwei allgemeine Register	<input type="checkbox"/>	<input type="checkbox"/>
Für Pipelining optimiert	<input type="checkbox"/>	<input type="checkbox"/>
Gilt als "modernere" Architektur	<input type="checkbox"/>	<input type="checkbox"/>
Datentransfer fast nur über Load-Store Befehle	<input type="checkbox"/>	<input type="checkbox"/>
Atmel AVR ist ein typischer...	<input type="checkbox"/>	<input type="checkbox"/>
Die Programme sind im allgemeinen kleiner	<input type="checkbox"/>	<input type="checkbox"/>

1.c) Steuerwerk und Datenpfad

___/4P

Ordne die folgenden Eigenschaften dem Steuerwerk oder dem Datenpfad zu

	Steuerwerk	Datenpfad
Akkumulator	<input type="checkbox"/>	<input type="checkbox"/>
Mikrocode	<input type="checkbox"/>	<input type="checkbox"/>
Carry-Flag	<input type="checkbox"/>	<input type="checkbox"/>
Datenbus	<input type="checkbox"/>	<input type="checkbox"/>
Fetch Instruction	<input type="checkbox"/>	<input type="checkbox"/>
Harvard-Architektur	<input type="checkbox"/>	<input type="checkbox"/>
Programmspeicher	<input type="checkbox"/>	<input type="checkbox"/>
Rechenwerk	<input type="checkbox"/>	<input type="checkbox"/>

2) Praxis Assemblerprogrammierung

```

.include "m16def.inc"

    clr XH
    ldi XL, 0x60
strupper:
    ld R16, X+
    cpi R16, 0x00
    breq end
    cpi R16, 0x61
    brlo strupper
    cpi R16, 0x7B
    brsh strupper
    subi R16, 0x20
    dec XL
    st X+, R16
    rjmp strupper
end:
    rjmp end

```

Speicherauszug:

0x5F	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F	0x70	0x71	0x72
0x37	0x48	0x69	0x20	0x34	0x61	0x68	0x65	0x6C	0x69	0x00	0x20	0x4A	0x5A	0x42	0x6C	0x63	0x52	0x51	0x32

2.a) Wie groß ist das Programm in Bytes? _____/1P

_____ Bytes

2.b) Wie viele Takte benötigt das Programm bis zum Erreichen des Labels end? _____/3P

Hinweis: Notiere die Zwischenschritte der Berechnung!

_____ Takte

2.c) Was steht im Speicher nach der Ausführung? _____/3P

Es reicht aus, die geänderten Bytes einzutragen.

0x5F	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F	0x70	0x71	0x72	

2.d) Welche Werte haben die Register nach der Ausführung? _____/1P

R16	X

2.e) Fragen zum Programm _____/5P

	richtig	falsch
Das Programm zählt die Anzahl der 0x00 Bytes	<input type="checkbox"/>	<input type="checkbox"/>
Das Programm nutzt den Stack	<input type="checkbox"/>	<input type="checkbox"/>
Es wird nur das Register R16 verwendet	<input type="checkbox"/>	<input type="checkbox"/>
Es werden Daten kopiert	<input type="checkbox"/>	<input type="checkbox"/>
Die Laufzeit ist abhängig von den Daten im Speicher	<input type="checkbox"/>	<input type="checkbox"/>
Am Ende der Routine ist der X Zeiger wieder am ursprünglichen Wert	<input type="checkbox"/>	<input type="checkbox"/>
Es werden nur bestimmte Bytes im Speicher verändert	<input type="checkbox"/>	<input type="checkbox"/>
Im Speicher werden 16 Bit Daten im Speicher verarbeitet	<input type="checkbox"/>	<input type="checkbox"/>
Es kann zu einem Überlauf in der ALU kommen	<input type="checkbox"/>	<input type="checkbox"/>
Es können nicht mehr als 255 Bytes an Daten bearbeitet werden	<input type="checkbox"/>	<input type="checkbox"/>

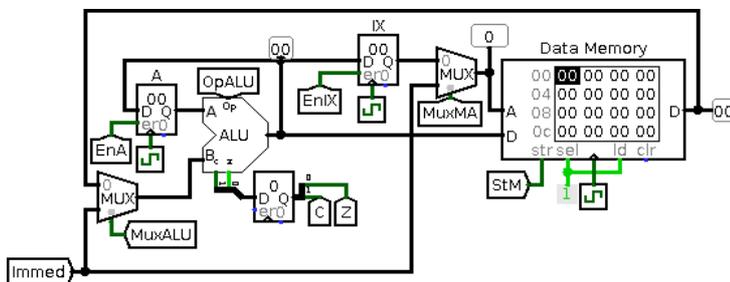
2.f) Assembler Aufgabe

___/3P

Im Register R16:R17 (high:low) steht ein 16 Bit Wert. Dieser soll mit 5 multipliziert werden. Bringe die Befehle in die richtige Reihenfolge.

___ mov R19, R17
 ___ lsl R19
 ___ adc R16, R18
 ___ rol R18
 ___ rol R18
 ___ mov R18, R16
 ___ lsl R19
 ___ add R17, R19

3) Praxis Rechnerarchitektur



ALU	Operation	Beschreibung
000	Result=A	Legt Operand A auf den Ausgang
001	Result=B	Legt Operand B auf den Ausgang
010	Result=A+B	Addiert A und B
011	Result=A-B	Subtrahiert B von A
100	Result=A AND B	Bitweise UND Verknüpfung
101	Result=A OR B	Bitweise OR Verknüpfung
110	Result=A EOR B	Bitweise Exclusive-OR
111	Result=A>>1	Logisches Rechtsschieben von A

3.a) Ordne die Beschreibung richtig zu

___/4P

1. nop – Führt keine Operation aus (No Operation)
2. ld IX, (IX) – Lädt Register IX mit dem Wert an der Adresse IX
3. or A, Imm. - Ver-ODER-t Register A mit Konstante
4. st (IX), Imm. - Speichert die Konstante Imm. an der Adresse IX
5. st (IX), A – Speichert den Wert von Register A an der Adresse IX
6. ld IX,A – Lädt Register IX mit dem Wert aus Register A
7. and A, (IX) – Ver-UND-et Register A mit dem Wert an Adresse IX
8. ld A, (Imm.) - Lädt Register A mit dem Wert an der Adresse Imm.

Nummer	EnA	EnIX	StM	MuxALU	MuxMA	OpALU
	0	1	0	x	x	000
	1	0	0	1	x	101
	1	0	0	0	1	001
	0	0	1	x	0	000
	0	0	0	x	x	x
	0	1	0	0	0	001
	1	0	0	0	0	100
	0	0	1	1	0	001