

Aufgabe 1

(24 Punkte)

Multiple Choice

Bitte kreuzen Sie alle richtigen Antwortmöglichkeiten an. Es gibt immer (mindestens) eine korrekte Antwort.

Wenn Datentyp Wort als 16 Bit definiert ist, wie viele Worte umfasst dann ein 64-Bit Datentyp?

- 1 2 4 16

(2)

Die Zahl 2B9h...

- entspricht 229d passt in 8-Bit Register passt nicht in 8-Bit Register entspricht 697d

(2)

Ein General-Purpose Mikroprozessor hat immer einen on-chip ROM.

- richtig falsch herstellerabhängig abhängig vom Programmiermodell

falsch!

Ein Mikrocontroller hat...

- on-chip ROM on-chip I/O Ports limitierten on-chip RAM keinen on-board Speicher

(1,5)

Der 8051 ist ein...

- 8-Bit Mikrocontroller 16-Bit Mikrocontroller CISC Prozessor RISC Prozessor

(1,5)

Welche Bitbreite hat das Register DPTR?

- 8 Bit 16 Bit 32 Bit 64 Bit

(2)

Default-mäßig ist welche Registerbank als aktive Registerbank voreingestellt?

- Registerbank 1 Registerbank 2 Registerbank 3 Registerbank 4

(2)

Über wieviele Timer verfügt der (klassische) 8051?

- 1 2 4 keine

(2)

Wie groß ist der Maximalwert, den das Timer-Zählregister speichern kann?

- 65536 65535 28800 256

(2)

Welches Register liegt nicht im SFR-Bereich?

- Stackpointer-Register Datapointer-Register R0-Register Akkumulator-Register

(2)

Das RESET ist normalerweise...

- low und benötigt ein low-Pegel-Signal zum Auslösen high und benötigt ein high-Pegel-Signal zum Auslösen high und benötigt ein low-Pegel-Signal zum Auslösen low und benötigt ein high-Pegel-Signal zum Auslösen

falsch!

Nach Unterscheidung zwischen Primär- und Sekundärmodus haben...

- alle Ports des 8051 eine alternative Funktion einige Ports des 8051 eine alternative Funktion keine Ports des 8051 eine alternative Funktion die Ports entgegengesetzte Polarität

(2)

Aufgabe 2

(30 Punkte)

Umgang mit dem 8051

Welche der folgenden Anweisungen ist (sind) unzulässig? Begründen Sie kurz und knapp ihre Aussage.

- a) MOV R3, #500 *unzulässig!*
 - b) MOV A, #F5h *← Die Null fehlt vor dem F → macht aber nichts!*
 - c) ADD R7, R4 *← 2 Register können nicht miteinander adressiert werden ✓*
 - d) ADD A, #F5h *← Die Null fehlt vor dem F → wäre Komma vor dem!*
 - e) ADD R3, A *✓*
 - f) ADD A, #255h *✓*
 - g) MOV R1, #50 *✓*
- (2)

Ermitteln Sie die Anzahl der ausgeführten Schleifendurchläufe für folgenden Code.

```

MOV R6, #200 199x
Zurück:
MOV R5, #100 99x
Hier:
DJNZ R5, Hier
DJNZ R6, Zurück
    
```

$199 \times 99 = \underline{\underline{19701}}$ (0)

Was ist das Ergebnis des folgenden Codes und wo ist es gespeichert?

```

MOV A, #15h
MOV R2, #13h
ADD A, R2
    
```

28hex ist im Akkumulator ✓ (6)

Identifizieren Sie die Adressierungsart für jede der folgenden Anweisungen.

- a) MOV B, #34h *Direkte Adressierung ✓*
 - b) MOV A, 50h *indirekte Adressierung ✓*
 - c) MOV R0, A *Register-indirekte Adressierung ✓*
 - d) MOV A, @R0 *indirekte Adressierung ✓*
- (4,5)

Folgende Register seien mit den angegebenen Werten beschrieben: A = F0h, B = 56h und R1 = 90h. Führen Sie nun die folgenden Operationen durch und geben Sie jeweils das Ergebnis, sowie das Register an, in dem das Ergebnis gespeichert ist. Beachten Sie, dass alle Anweisungen unabhängig von einander sind.

- a) ANL A, #45H
 $\text{AND } \begin{matrix} A: 1111\ 0000 \\ B: 0100\ 0101 \\ \hline 0100\ 0000 \end{matrix}$ *ist im Akku nach der Operation* ✓ (2)
- b) ORL A, B
 $\text{OR } \begin{matrix} A: 1111\ 0000 \\ B: 0101\ 0110 \\ \hline 1111\ 0110 \end{matrix}$ *ist im Akku nach Operation* ✓
- c) XRL A, R1
- d) ORL A, #99H



Aufgabe 3

(10 Punkte)

Fragen zur maschinennahen Programmierung

1. Was sind die Funktionen von Compiler, Assembler und Linker?

Compiler: übersetzt den Assemblercode bzw C-Code in Maschinsprache also in Bits.

Linker: verknüpft eingebundene Bibliotheken bzw. Unterprogramme ~~und stellt~~ zu einer Datei.

Assembler: übersetzt den Assemblercode in Maschinencode (Bits). (2)

^{nach} zu Compiler: Linker und Assembler sind im Compiler also sind Teile des Compilers.

2. Vergleicht man C und Assembler, welche der beiden Sprachen ist effizienter im Hinblick auf die Code-Generierung (d.h. die Menge an für das Programm verwendeten ROM-Speicher)?

Assembler, weil typischerweise in C Bibliotheken eingebunden werden, in denen viele Funktionen sind die auf dem ROM geschrieben werden, auch wenn man diese gar nicht nutzt.

(2)

3. Nennen Sie 3 Konzepte der Performancesteigerung in einem Rechnersystem.

- Übertaktung von verschiedenen Komponenten z.B. CPU
- Pipeline-Methode verändern z.B. Supersuperpipelining
- weitere zusätzliche CPU-Kerne einbauen

(2)

4. Was bedeutet "Polling von Daten"?

Zyklisches abfragen von empfangenen Daten

etwas ungenau! (1,5)

5. Wie funktioniert das interruptgesteuerte Reagieren auf Ereignisse? Woher "weiß" die CPU wo im Speicher sich die Interrupt Service Routine befindet?

Wird ein Interrupt ausgelöst, wird der Befehl, der gerade abgearbeitet wird noch zu Ende geführt und auf dem Stack wird die ROM-Adresse, bei der unterbrochen wurde gespeichert. Dann wird der Code des Interruptprogramms ausgeführt und anschließend wird ~~es~~ bei der Adresse ~~was~~ die auf dem Stack ist weitergearbeitet. (1,5)

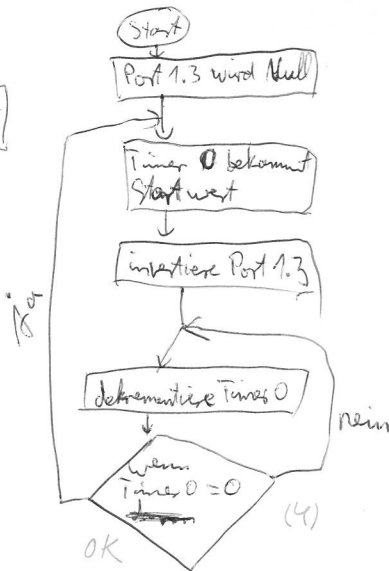
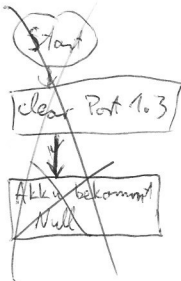
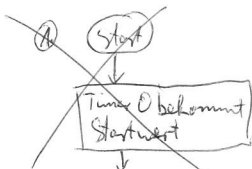
Aufgabe 4

(24 Punkte)

Programmierung in Assembler

Erstellen sie ein Assembler-Programm für den 8051, das ein Rechtecksignal von 1 kHz an Port 1.3 generiert. Dies soll mit Hilfe des Timer 0 geschehen.

1. Erstellen Sie zunächst einen Programmablaufplan für die Aufgabe.
2. Berechnen Sie die für die Timer-Programmierung zugrunde zu legende Frequenz und Zyklusdauer für den Fall, dass der angeschlossene Quarz eine Frequenz von (a) 11,0592 MHz, (b) 20 MHz und (c) 24 MHz hat. Zeigen Sie alle Schritte Ihrer Rechnung.
3. Schreiben Sie den Assembler-Code für die gegebene Problemstellung. Die Referenzfrequenz sei angenommen mit 11,0592 MHz. Wählen Sie einen geeigneten Timer-Modus und begründen Sie die Wahl.
Modus 1 um eine 16 Bit Zahl herunterzuzählen ✓ (1)
4. Welche Verzögerung wird durch den Timer erzeugt?



$1 \text{ kHz} = 1000 \text{ Hz}$

~~$\frac{1}{2} \cdot 1000$~~



Aufgabe 5

(12 Punkte)

Arbeiten mit Interrupts

Bestimmen Sie, ob die folgende Interrupt Service Routine direkt in dem Interrupt-Vektorraum für den externen Interrupt 0 liegen kann und erläutern Sie warum bzw. warum nicht.

```

EX0_ISR:
INC 42h      1 Byte
ADD A, #255  2 Byte
CPL A        1 Byte
MOV R3, A    1 Byte
MOV P2, #FFh 2 Byte
RETI         1 Byte
    +
    8 Byte
    
```

7 Byte
weil 7 Byte Code
 $EX0 \text{ ab- } 0003 \text{ hex} + 0007 = A$
 also geht der Code von 0003 hex bis $000A \text{ hex}$,
 trifft nicht auf $000B \text{ hex}$,
 und ist somit zulässig.
 in $000B \text{ hex}$ ist die Vektoradresse eines anderen Interrupts.

Gegeben sei nun folgendes Programm:

Start:

```

CLR IE0      ; interrupt clear
MOV R0, #00h
CLR IT0      ; interrupt freigeben fahr! für wird werden Pegel- und Flanken-
SETB P3.2   ; Externer Interrupt freigeben triggung unterschied!
MOV IE, #10000001b
    
```

Schleife:

```

SJMP Schleife
    
```

```

EX0_ISR:
INC R0
RETI
    
```

Nehmen wir an, dass der folgende Eingangs-Datenstrom an P3.2 angelegt wurde. Wie ist dann der Inhalt des Registers R0?



R0 ist dann 08hex fahr! (0)

