

# MUSTER

Prüfungsbogen: 0

EvaExam

Klausur Mikroprozessorsysteme SS 2020



Hochschule Darmstadt  
Fachbereich Informatik

Prof. Dr. Jens-Peter Akelbein  
Mikroprozessorsysteme SS2020



Bitte so markieren: ☐ ☒ ☐ ☐ ☐ Bitte verwenden Sie einen Kugelschreiber oder nicht zu starken Filzstift. Dieser Fragebogen wird maschinell erfasst.  
Korrektur: ☐ ☒ ☐ ☒ ☐ Bitte beachten Sie im Interesse einer optimalen Datenerfassung die links gegebenen Hinweise beim Ausfüllen.

Bitte ausfüllen (Die Angabe des Namens ist freiwillig.):

Matrikel-Nummer für den Prüfungsbogen Nr.: 0:

Vorname:

Nachname:

Für die eindeutige Zuordnung der Prüfung übertragen Sie bitte Ihre Prüfungsteilnehmer-ID gewissenhaft in die dafür vorgesehenen Felder. Alle Seiten sind vollständig individualisiert und nicht mit anderen Prüfungen tauschbar.

|   |                          |                          |                          |                          |                          |
|---|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
|   |                          |                          |                          |                          |                          |
| 0 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 1 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 4 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 6 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 7 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 8 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 9 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

## Hinweise zur Klausur

Als Hilfsmittel ist nur **1 doppelseitig handbeschriebenes DIN A4-Blatt** (keine Kopie) zugelassen.

Das Deckblatt ist mit Angabe der Matrikelnummer und des Namens abzugeben.

Nutzen Sie die vorgesehenen freien Flächen der Klausur für Nebenrechnungen. **Bewertet werden nur die Angaben innerhalb der gekennzeichneten Flächen und Markierungen** zur Beantwortung.

**Verwenden Sie** einen **Kugelschreiber** oder einen **nicht zu dicken Faserschreiber**. Verwenden Sie keinen Bleistift oder dicken Filzstift. Markieren Sie angekreuzte Kästchen mittig mit einem Kreuz. Malen Sie bei falschen Markierungen das Kästchen vollständig aus. Schauen Sie sich die Beispiele für Markierung und Korrektur auf der Seite oben. Nicht eindeutige markierte Antworten werden nicht gewertet.

**Prüfungsdauer:** 90 Minuten

Die Zulassung zur Modulprüfung Mikroprozessorsysteme steht unter dem Vorbehalt, dass die Prüfungsvorleistung (PVL) bestanden wurde. Sollte die PVL nicht erfolgreich erbracht worden sein, so gilt das Klausurergebnis weder in positiver noch in negativer Hinsicht. Eine "Heilung" der fehlenden Zulassungsvoraussetzung ist nicht möglich. Sollten Sie nicht auf der Prüfungsliste aufgeführt sein, bestätigen Sie die erfolgreiche Testierung der Vorleistung als Zulassung zur Klausur mit Ihrer Unterschrift. Hiermit bestätige ich, dass ich die Prüfungsvorleistung mit Erfolg bestanden habe.

Unterschrift: \_\_\_\_\_

Ende der Hinweise

## 1. Mikroprozessorsysteme (18 Punkte)

**Hinweis:** Kreuzen Sie an, welche Aussagen richtig sind. Falsche Aussagen kreuzen Sie nicht an. Die Teilaufgaben werden als richtig gewertet, wenn jeweils alle Aussagen innerhalb der Teilaufgabe richtig bewertet wurden.

### 1.1 Überprüfen Sie folgende Aussagen zu **Mikrocontrollern**.

- ☐ Mikrocontroller unterscheiden sich von Mikroprozessoren dadurch, dass sie für spezifische Anwendungsfälle bzw. Produkte zugeschnitten sein können.
- ☐ Mikrocontroller besitzen eine fest vorgegebene Speicherausstattung, die Teil des Chips ist.
- ☐ Heutige Mikrocontroller werden heute oft auch System-on-Chip SoC genannt, weil alle benötigten Hardwarekomponenten für ein Gerät bereits in einem Chip vorhanden sind.

### 1.2 Überprüfen Sie folgende Aussagen zu **Peripheriebausteinen**.

- ☐ Ein Mikrocontroller enthält Peripheriebausteine wie Timer und serielle Schnittstellen.
- ☐ Ein Mikrocontroller enthält Peripheriebausteine wie Sensoren und Aktoren.
- ☐ Es lassen sich Varianten von Mikrocontrollern wählen, die nur die für eine Anwendung notwendigen Peripheriebausteine enthalten.

### 1.3 Überprüfen Sie folgende Aussagen zur **hardwarenaher Programmierung**.

- ☐ In der Programmiersprache C gibt es das Schlüsselwort volatile. Hierdurch wird dem Compiler angegeben, dass sich Variablen jederzeit durch externe Einflüsse wie durch Hardwarebausteine verändern können.
- ☐ Hardwarebausteine, die in einen Speicherbereich „gemapped“ sind, lassen sich als zusammengesetzter Datentyp in C als struct deklariert darstellen.
- ☐ Die explizite Typumwandlung (casting) von Zeigern in C führt in der hardwarenahen Programmierung oft zu Fehlern und sollte unbedingt vermieden werden.

### 1.4 Überprüfen Sie folgende Aussagen zum **direkten Speicherzugriff** (Direct Memory Access DMA)

- ☐ DMA nennt man das Prinzip, dass der Prozessorkern auf Peripheriebausteine in gleicher Form wie auf den Arbeitsspeicher direkt über Adressen zugreifen kann.
- ☐ DMA entlastet den Prozessorkern bei der Übertragung von größeren Datenmengen zwischen Arbeitsspeicher und Peripherie.
- ☐ DMA ist ein Grundprinzip jedes Von-Neumann-Rechners und findet sich in jedem Mikroprozessor und Mikrocontroller.

### 1.5 Überprüfen Sie folgende Aussagen zur **Analog-Digital-(AD)-Wandlung**.

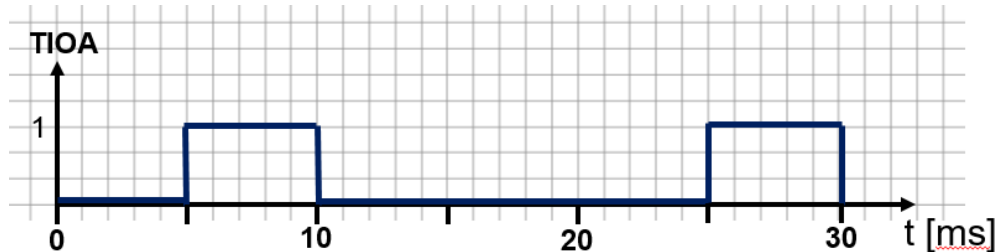
- ☐ AD-Wandlung benötigt man, um die digitalen Signale in einem Mikrocontroller wieder in Analogsignale umzusetzen.
- ☐ Bei der AD-Wandlung ist das Zählverfahren deutlich schneller als das Wägeverfahren und zwar um den Faktor  $\log_2(N)$  bei N Bits.
- ☐ Der maximale Quantisierungsfehler bei einer AD-Wandlung entspricht dem Wert, den die Hälfte des niederwertigsten Bits repräsentiert.

### 1.6 Überprüfen Sie folgende Aussagen zu **Interrupts**

- ☐ Interrupts müssen unterschiedliche Prioritäten haben, weil sonst der Interrupt-Controller nicht entscheiden kann, was zu tun ist.
- ☐ Eine Interrupt-Service-Routine lässt sich in einem Interrupt-Controller registrieren und ist damit frei programmierbar.
- ☐ Der Programmcode einer Interrupt-Service-Routine behandelt ein Compiler wie eine Funktion. Durch Angabe eines Attributs erzeugt der Compiler die spezifischen Aspekte für eine nicht unterbrechbare ISR-Routine.

## 2. Timer-Bausteine (24 Punkte)

Mit dem Timerbaustein des in der Vorlesung vorgestellten Mikrocontrollers AT91M63200 wird ein **Signal erzeugt**. Das Signal wird am Ausgang TIOA gemessen. Es ergibt sich der im folgenden Diagramm dargestellte Signalverlauf, wobei die Messung zu einem Zeitpunkt  $t=0\text{ms}$  startet. Gehen Sie davon aus, dass das **Zählregister** des Timers in dem Diagramm unter anderem zum **Zeitpunkt  $T=10\text{ms}$**  zurückgesetzt wurde.



2.1 Geben Sie die Frequenz des dargestellten Signals TIOA an.

Das Signal hat eine Frequenz von wieviel Hertz?

2.2 Wie ist das Tastverhältnis (Duty Cycle) des Impulses in dem dargestellten Signal?

Geben Sie das Tastverhältnis in Prozent an.

2.3 Sie wollen mit einem Board mit dem Mikrocontroller AT91M63200 ein gleichartiges Signal mit den enthaltenen 16-Bit Timerbausteinen erzeugen. Das Board wird mit einem **Systemtakt von 5 MHz** betrieben. Für den Timerbaustein lassen sich für den Taktvorteiler (Prescaler) die Faktoren 2, 8, 32, 128 oder 1024 konfigurieren.

Wählen Sie den optimalen Prescale-Faktor.

☐ Taktvorteiler 2  
☐ Taktvorteiler 128

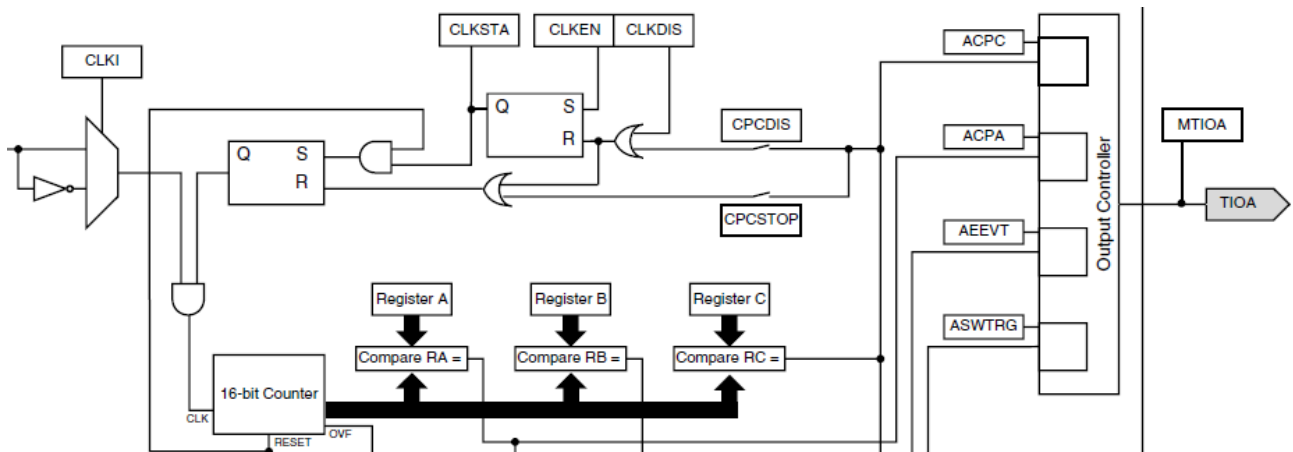
☐ Taktvorteiler 8  
☐ Taktvorteiler 1014

☐ Taktvorteiler 32

2.4 Berechnen Sie aus dem Systemtakt des Boards und optimalen Taktvorteiler die Anzahl an Nanosekunden, die es dauert, bis das Zählregister des Timerbausteins um 1 hochzählt.

Geben Sie die Anzahl an Nanosekunden für eine Inkrementierung des Zählregisters an.

## 2. Timer-Bausteine (24 Punkte) [Fortsetzung]



- 2.5 Berechnen Sie den Wert für das Register C, um ein Signal gleicher Frequenz wie im Signaldiagramm zu erzeugen.

Das Register C wird geladen mit dem Wert

|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|

- 2.6 Berechnen Sie den Wert, mit dem das Register A geladen wird, um das im Diagramm dargestellte Tastenverhältnis zu generieren.

Das Register A wird geladen mit dem Wert

[illegible]

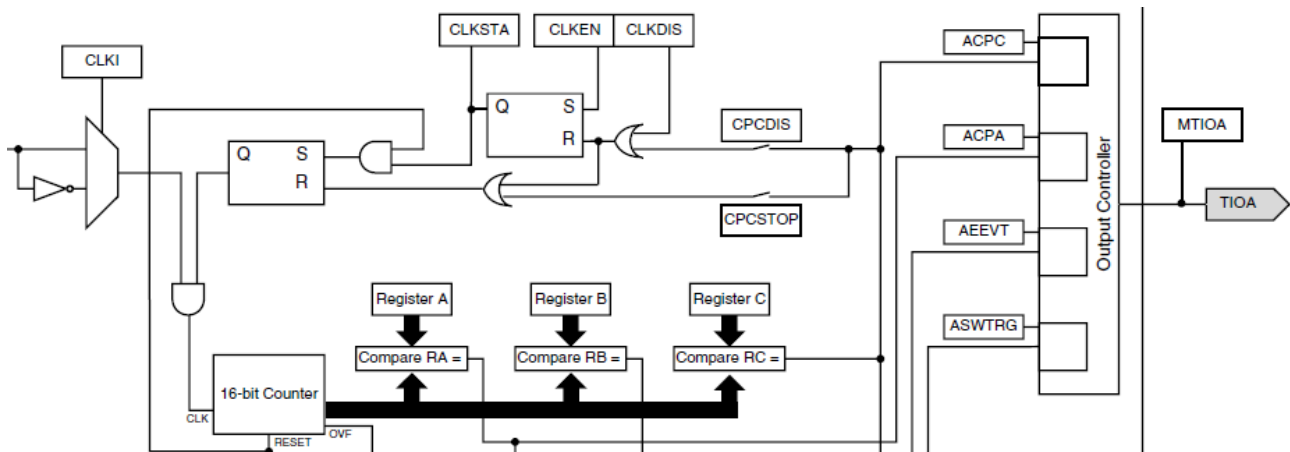
- 2.7 Mit dem Register B können Sie mit dem gleichen Zählregister ein weiteres Signal an dem Ausgang TIOB erzeugen. Mit welchem Wert laden Sie das Register B, um an diesem Ausgang ein Signal gleicher Frequenz aber mit dem **Tastverhältnis von 50%** zu erhalten?

|  |
|--|
|  |
|--|

Das Register B wird geladen mit dem Wert

[illegible]

## 2. Timer-Bausteine (24 Punkte) [Fortsetzung]



Konfigurieren Sie die Signalgenerierung im Output Controller des Timer-Bausteins durch die **Registerteile ACPA und ACPC**. Beachten Sie, dass die Konfiguration das Signal unabhängig vom vorherigen Zustand des Signals im Zeitdiagramm erzeugen soll.

2.8 Mit welchem Wert konfigurieren Sie hierzu den Registerteil ACPA für die Signalgenerierung?

• **ACPA: RA Compare Effect on TIOA**

| ACPA |   | Effect |
|------|---|--------|
| 0    | 0 | none   |
| 0    | 1 | set    |
| 1    | 0 | clear  |
| 1    | 1 | toggle |

- ☐ 0 0  
☐ 0 1  
☐ 1 0  
☐ 1 1

2.9 Mit welchem Wert konfigurieren Sie hierzu den Registerteil ACPC für die Signalgenerierung?

• **ACPC: RC Compare Effect on TIOA**

| ACPC |   | Effect |
|------|---|--------|
| 0    | 0 | none   |
| 0    | 1 | set    |
| 1    | 0 | clear  |
| 1    | 1 | toggle |

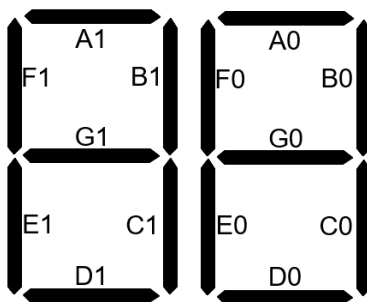
- ☐ 0 0  
☐ 0 1  
☐ 1 0  
☐ 1 1

2.10 Im **Capture-Modus** des Timer-Bausteins messen Sie Zeiten. Welche der Register A, B und C laden Sie zur Konfiguration vor der Messung mit einem Wert, um eine Messdauer festzulegen?

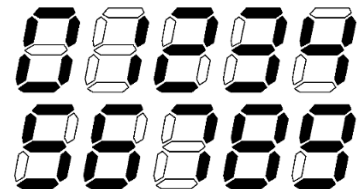
- ☐ Register A  
☐ Register A und B  
☐ Register A und B und C  
☐ Register B  
☐ Register A und C  
☐ Register C  
☐ Register B und C

## 3. Parallele I/O (PIO) mit 7-Segment-Anzeige (16 Punkte)

Eine **7-Segment-Anzeige** mit zwei Stellen soll über die parallelen I/O-Ports angesteuert werden. Der Portbaustein PIOA ist bereits aktiviert und die mit Segmenten beschalteten Portpins als Ausgänge konfiguriert. Die Anschlussbelegung der **high-aktiven** Anzeige an den PIOA-Baustein ist in der folgenden Übersicht gezeigt. Nicht belegte Portpins haben keine Wirkung (don't cares). Angezeigt werden soll jeweils **eine Zahl mit zwei Ziffern von 0 bis 9**.



| Portpin der PIOA | Segment | Portpin der PIOA | Segment |
|------------------|---------|------------------|---------|
| 0                | A0      | 8                | A1      |
| 1                | B0      | 9                | B1      |
| 2                | C0      | 10               | C1      |
| 3                | D0      | 11               | D1      |
| 4                | E0      | 12               | E1      |
| 5                | F0      | 13               | F1      |
| 6                | G0      | 14               | G1      |



Initialisiert sind weiterhin Zeigervariablen zum Zugriff auf die folgenden Register des PIOA-Portbausteins:

**PIOA\_SODR (Set Output Data Register):**

Setzen der (entsprechenden) PIO Ausgänge auf den **HIGH-Pegel**

**PIOA\_CODR (Clear Output Data Register):**

Löschen der (entsprechenden) PIO-Ausgänge auf den **LOW-Pegel**

**PIOA\_ODSR (Output Data Status Register):**

Lesen des Zustands der PIO-Ausgänge

Im folgenden sind Quelltexte aufgezeigt, mit denen jeweils Zahlen angezeigt werden. Tragen Sie nach jedem Codeabschnitt die Zahl als Lösung ein, die jeweils nach Durchlaufen der Programmzeilen angezeigt werden.

```
PIOA_CODR = 0x7F7F;           // Löschen der HIGH-Aktiven 7-Segmentanzeige
PIOA_SODR = 0x5B7D;           // Anzeige: _____
```

3.1 Bitte tragen Sie die angezeigte Zahl an.

## 3. Parallele I/O (PIO) mit 7-Segment-Anzeige (16 Punkte) [Fortsetzung]

```
PIOA_SODR = 0x7F7F;  
PIOA_CODR = 0x7879;    // Anzeige: _____
```

3.2 Bitte tragen Sie die angezeigte Zahl ein.

|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|

```
PIOA_CODR = 0x7F7F;  
PIOA_SODR = 0x006D;  
PIOA_SODR = *PIO_ODR << 8 + 2;    // Anzeige: _____
```

3.3 Bitte tragen Sie die angezeigte Zahl ein.

|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|

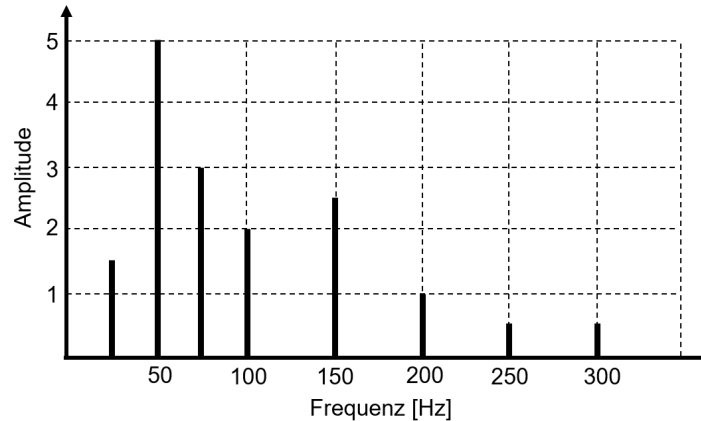
```
PIOA_CODR = 0x7F7F;  
PIOA_SODR = ~0x3099;    // Anzeige: _____
```

3.4 Bitte tragen Sie die angezeigte Zahl ein.

|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|
|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|

## 4. Signalverarbeitung (17 Punkte)

Sie tasten ein Signal  $S(t)$  ab und führen eine Fourieranalyse durch. Hierfür erhalten Sie als Ergebnis folgendes Frequenz-Amplitudendiagramm.



Im folgenden sind eine Reihe von möglichen Signalkomponenten unterschiedlicher Frequenz angegeben. Prüfen Sie, ob und wenn ja mit welcher Amplitude die jeweilige Komponente enthalten ist. Die Variable  $t$  ist in den Komponentenangaben bereits ohne Einheit Sekunde verwendet.

- |                                       |                                                                                          |                                                                                          |                                                            |
|---------------------------------------|------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|------------------------------------------------------------|
| 4.1 Signalkomponente $\sin(25\pi t)$  | <input type="checkbox"/> 0<br><input type="checkbox"/> 1,5<br><input type="checkbox"/> 3 | <input type="checkbox"/> 0,5<br><input type="checkbox"/> 2<br><input type="checkbox"/> 5 | <input type="checkbox"/> 1<br><input type="checkbox"/> 2,5 |
| 4.2 Signalkomponente $\sin(50\pi t)$  | <input type="checkbox"/> 0<br><input type="checkbox"/> 1,5<br><input type="checkbox"/> 3 | <input type="checkbox"/> 0,5<br><input type="checkbox"/> 2<br><input type="checkbox"/> 5 | <input type="checkbox"/> 1<br><input type="checkbox"/> 2,5 |
| 4.3 Signalkomponente $\sin(75\pi t)$  | <input type="checkbox"/> 0<br><input type="checkbox"/> 1,5<br><input type="checkbox"/> 3 | <input type="checkbox"/> 0,5<br><input type="checkbox"/> 2<br><input type="checkbox"/> 5 | <input type="checkbox"/> 1<br><input type="checkbox"/> 2,5 |
| 4.4 Signalkomponente $\sin(100\pi t)$ | <input type="checkbox"/> 0<br><input type="checkbox"/> 1,5<br><input type="checkbox"/> 3 | <input type="checkbox"/> 0,5<br><input type="checkbox"/> 2<br><input type="checkbox"/> 5 | <input type="checkbox"/> 1<br><input type="checkbox"/> 2,5 |
| 4.5 Signalkomponente $\sin(150\pi t)$ | <input type="checkbox"/> 0<br><input type="checkbox"/> 1,5<br><input type="checkbox"/> 3 | <input type="checkbox"/> 0,5<br><input type="checkbox"/> 2<br><input type="checkbox"/> 5 | <input type="checkbox"/> 1<br><input type="checkbox"/> 2,5 |
| 4.6 Signalkomponente $\sin(200\pi t)$ | <input type="checkbox"/> 0<br><input type="checkbox"/> 1,5<br><input type="checkbox"/> 3 | <input type="checkbox"/> 0,5<br><input type="checkbox"/> 2<br><input type="checkbox"/> 5 | <input type="checkbox"/> 1<br><input type="checkbox"/> 2,5 |
| 4.7 Signalkomponente $\sin(250\pi t)$ | <input type="checkbox"/> 0<br><input type="checkbox"/> 1,5<br><input type="checkbox"/> 3 | <input type="checkbox"/> 0,5<br><input type="checkbox"/> 2<br><input type="checkbox"/> 5 | <input type="checkbox"/> 1<br><input type="checkbox"/> 2,5 |
| 4.8 Signalkomponente $\sin(300\pi t)$ | <input type="checkbox"/> 0<br><input type="checkbox"/> 1,5<br><input type="checkbox"/> 3 | <input type="checkbox"/> 0,5<br><input type="checkbox"/> 2<br><input type="checkbox"/> 5 | <input type="checkbox"/> 1<br><input type="checkbox"/> 2,5 |

## 4. Signalverarbeitung (17 Punkte) [Fortsetzung]

- 4.9 Wählen Sie aus der folgenden Liste **sämtliche** Frequenzen aus, mit denen Sie **alle im Signal S(t)** enthaltenen Signalkomponenten abtasten können, ohne dass Artefakte entstehen.

☐ 250 Hz  
☐ 400 Hz  
☐ 550 Hz  
☐ 700 Hz

☐ 300 Hz  
☐ 450 Hz  
☐ 600 Hz  
☐ 750 Hz

☐ 350 Hz  
☐ 500 Hz  
☐ 650 Hz  
☐ 800 Hz

- 4.10 Nun wollen Sie **nicht alle Frequenzen** in dem abgetasteten Signal **erhalten**, sondern durch die Wahl der Abtastfrequenz eine Wirkung vergleichbar eines Tiefpassfilters erreichen. Wählen Sie aus der folgenden Liste **diejenigen** Frequenzen aus, mit denen nach dem Abtasttheorem alle im Signal S(t) enthaltenen Signalkomponenten **bis auf diejenige mit der höchsten Frequenz** abgetastet werden.

☐ 250 Hz  
☐ 400 Hz  
☐ 550 Hz  
☐ 700 Hz

☐ 300 Hz  
☐ 450 Hz  
☐ 600 Hz  
☐ 750 Hz

☐ 350 Hz  
☐ 500 Hz  
☐ 650 Hz  
☐ 800 Hz

- 4.11 Sie tasten das Signal S(t) mit der **Frequenz von 800 Hz** ab und verwenden dabei einen **16-Bit** AD-Wandler. Wieviele Daten haben Sie innerhalb von **10 Sekunden** durch die Abtastung erzeugt? Geben Sie die Menge in **Byte** an.

Die entstandene Datenmenge angegeben in Byte beträgt

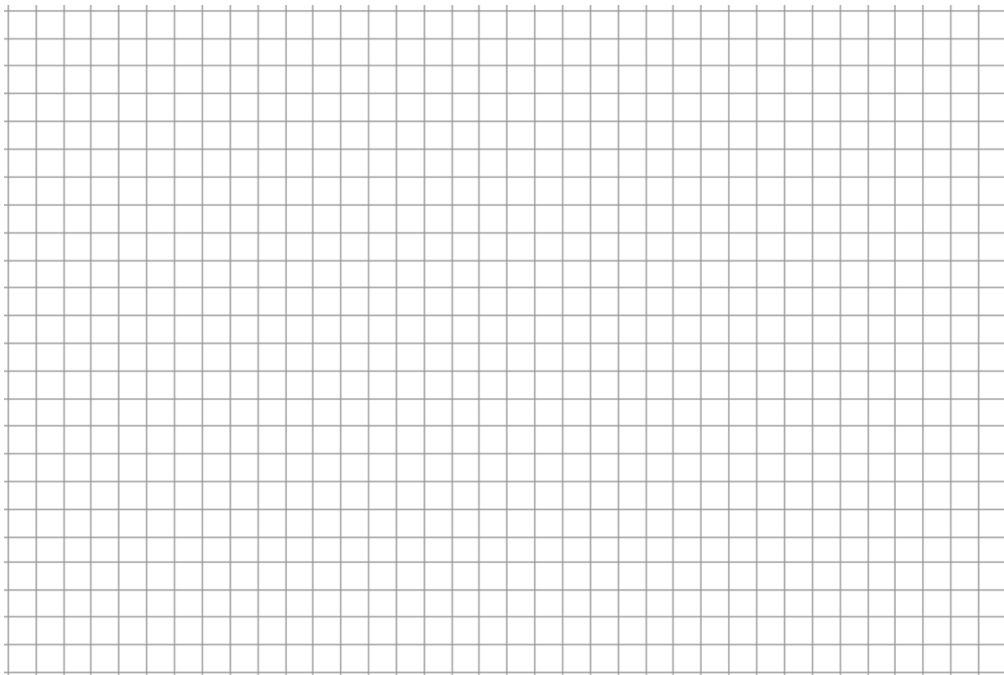
## 5. Cyclic Redundancy Check (12 Punkte)

Für die Übertragung von Daten verwenden Sie CRC4 als standardisiertes Prüfverfahren, welches 4 Bit als Checksumme nutzt. CRC4 ist durch ein Generatorpolynom definiert, welches in mathematischer Darstellung die folgende Form hat:

$$g(x) = x^4 + x + 1$$

- 5.1 Welche Bitkombination für die Berechnung von CRC-Summen ergibt sich, wenn Sie das Generatorpolynom in **binärer Form** angeben. Kreuzen Sie diese Kombination an.
- |                                 |                                |                                   |
|---------------------------------|--------------------------------|-----------------------------------|
| <input type="checkbox"/> 010011 | <input type="checkbox"/> 1001  | <input type="checkbox"/> 11011    |
| <input type="checkbox"/> 10011  | <input type="checkbox"/> 01011 | <input type="checkbox"/> 00010011 |
| <input type="checkbox"/> 1011   | <input type="checkbox"/> 01100 | <input type="checkbox"/> 000000   |
- 5.2 Berechnen Sie die Checksumme mit dem gegebenen Generatorpolynom für CRC4 für die folgende Nachricht
- Nachricht 0 0 1 1 0 0 1 0**
- Geben Sie an, welche Checksumme sich mit CRC4 ergibt.

- |                               |                               |                               |
|-------------------------------|-------------------------------|-------------------------------|
| <input type="checkbox"/> 0000 | <input type="checkbox"/> 0001 | <input type="checkbox"/> 0010 |
| <input type="checkbox"/> 0011 | <input type="checkbox"/> 0100 | <input type="checkbox"/> 0101 |
| <input type="checkbox"/> 0110 | <input type="checkbox"/> 0111 | <input type="checkbox"/> 1000 |
| <input type="checkbox"/> 1001 | <input type="checkbox"/> 1010 | <input type="checkbox"/> 1011 |
| <input type="checkbox"/> 1100 | <input type="checkbox"/> 1101 | <input type="checkbox"/> 1110 |
| <input type="checkbox"/> 1111 |                               |                               |



## 5. Cyclic Redundancy Check (12 Punkte) [Fortsetzung]

5.3 Welche Hardwareelemente benötigen Sie für die Implementierung einer CRC-Generierung mit einem fest vorgegebenen Generatorpolynom?

☐ UND-Gatter

☐ ODER-Gatter

☐ EXOR-Gatter

☐ Inverter

☐ Schieberegister

☐ Multiplexer

☐ Demultiplexer

☐ Volladdierer

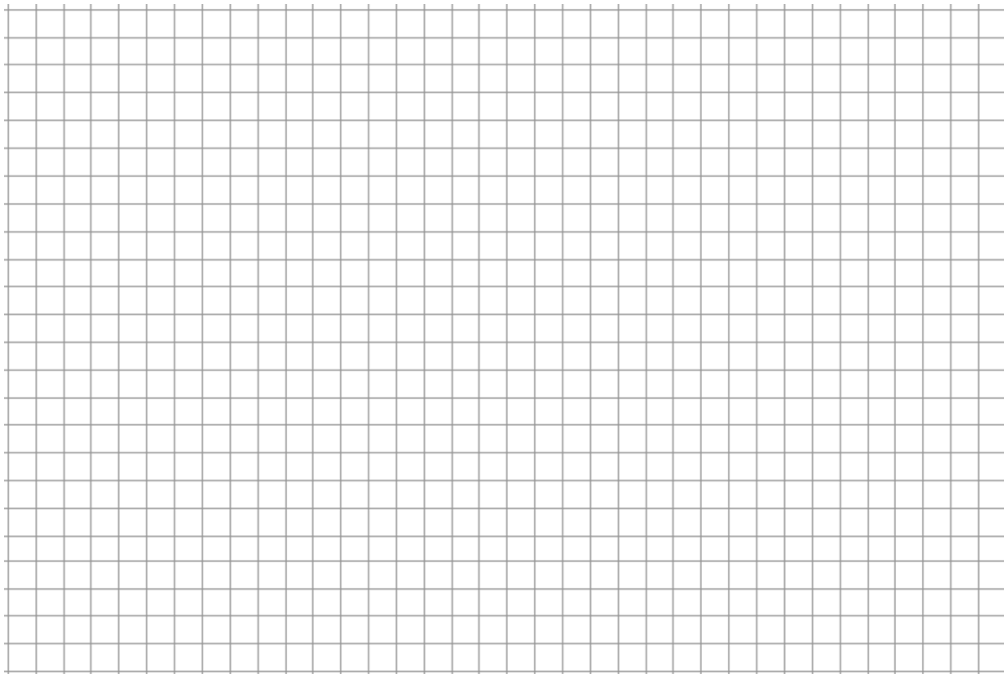
☐ CPU

5.4 Sie erhalten folgende Nachricht mit 4 Bits an Nutzdaten. Berechnen Sie mit Hilfe des CRC4-Verfahrens, ob die Nachricht gültig ist.

**Nachricht 1 1 1 1 0 0 1 0**

☐ Die Nachricht ist gültig.

☐ Die Nachricht ist nicht gültig.

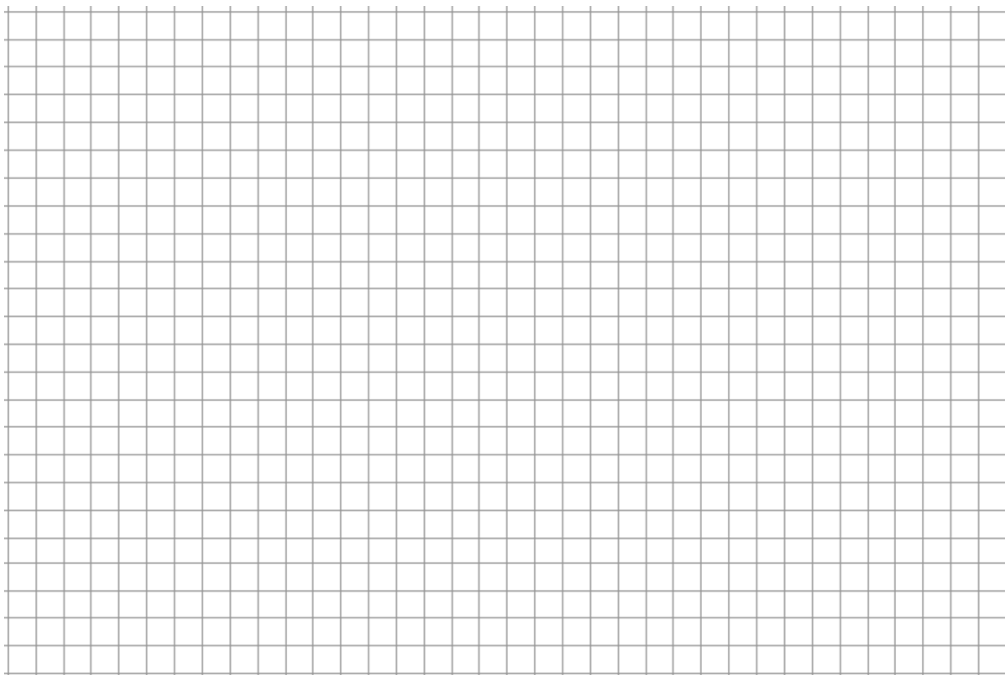
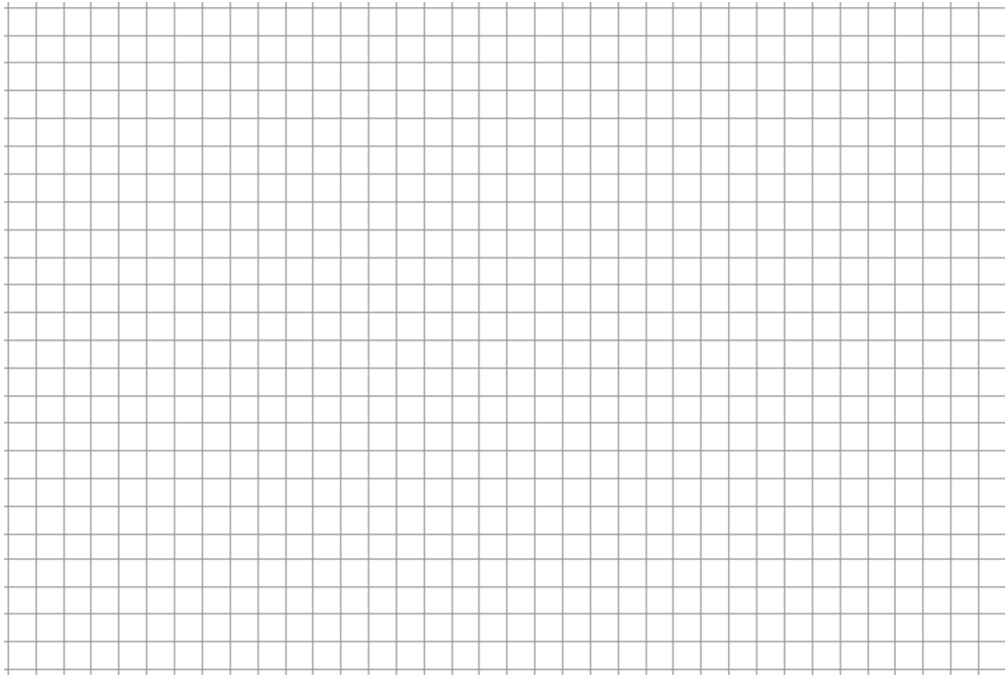


5.5 **Wie viele Iterationen (Rechenschritte)** muss eine CRC-Hardware zur Überprüfung der obigen Nachricht 11110010 durchführen, die einer Polynomdivision entsprechen?

Four dashed boxes for writing the answer.

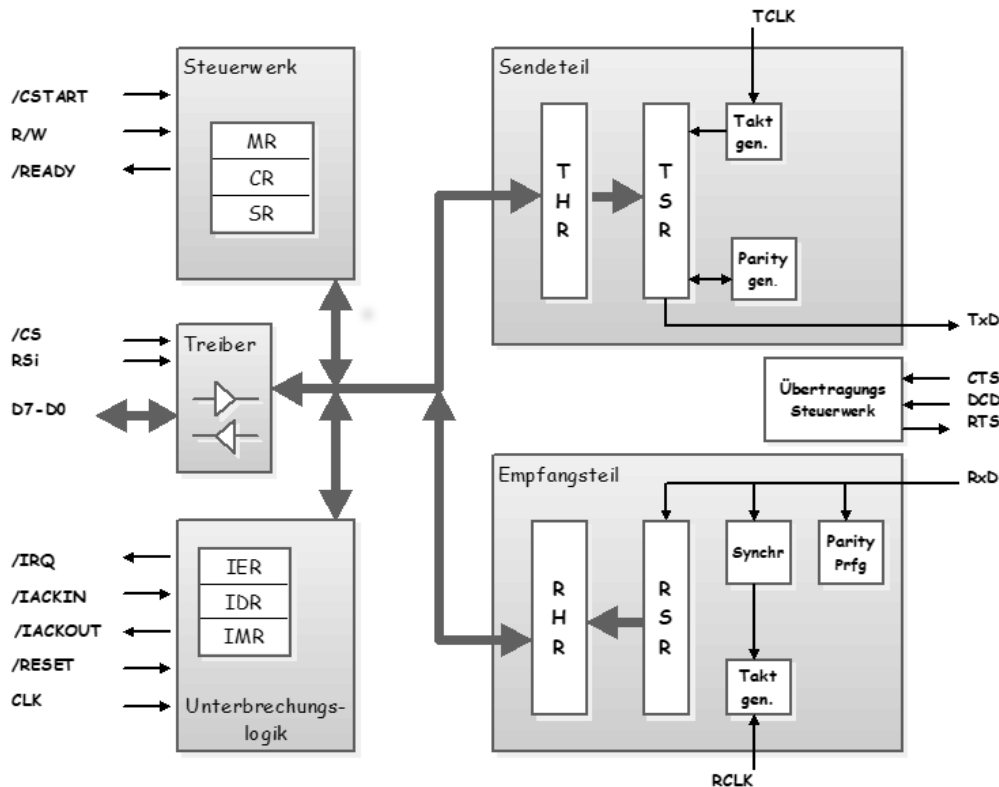
## 5. Cyclic Redundancy Check (12 Punkte) [Fortsetzung]

**Platz für weitere Nebenrechnungen**



## 6. Serielle Schnittstelle und Leitungscodes (15 Punkte)

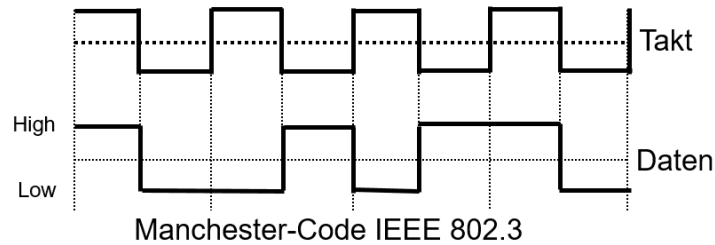
Eine USART-Hardwarekomponente lässt sich durch das folgende Strukturbild beschreiben. Beantworten Sie hierzu Fragen, welche Abläufe bei Datenübertragungen auftreten können.



- 6.1 Bei einer asynchronen Übertragung erfolgt die Synchronisierung, bevor das Startbit des Zeichenrahmens empfangen wurde.  
☐ Wahr ☐ Falsch
- 6.2 Bei dem Empfang von Daten werden diese über Rx D von einem anderen Teilnehmer empfangen.  
☐ Wahr ☐ Falsch
- 6.3 Das Receive Shift Register deserialisiert die Daten, so dass sie als Byte vom Anwendungsprogramm aus dem Receive Hold Register ausgelesen werden können.  
☐ Wahr ☐ Falsch
- 6.4 Das Transmit Hold Register serialisiert die Daten, indem es diese bitweise in das Transmit Shift Register schiebt.  
☐ Wahr ☐ Falsch
- 6.5 Das Übertragungssteuerwerk verwendet CTS, DCD und RTS. Diese werden bei jeder seriellen Schnittstelle verwendet.  
☐ Wahr ☐ Falsch
- 6.6 Über das Mode Register und das Control Register gibt ein Anwendungsprogramm dem Steuerwerk Informationen zur Konfiguration der seriellen Schnittstelle.  
☐ Wahr ☐ Falsch
- 6.7 Die Unterbrechungslogik unterbricht die CPU immer bei jedem empfangenen Zeichen. Dann verarbeitet die CPU das Zeichen weiter.  
☐ Wahr ☐ Falsch

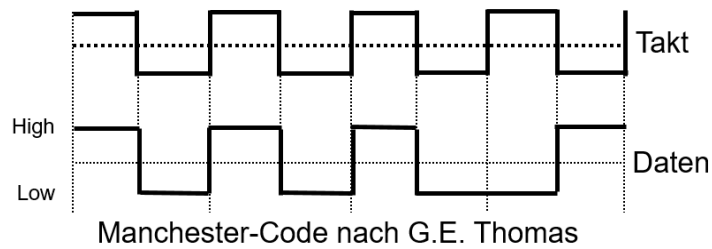
## 6. Serielle Schnittstelle und Leitungscodes (15 Punkte) [Fortsetzung]

Sie erhalten Signale auf einer Leitung, die mit dem Takt der Übertragung dargestellt sind. Die Daten sind mit dem jeweils angegebenen Leitungscode übertragen worden.



6.8 Sie haben die Signale in einer **Manchester-Codierung nach IEEE 802.3** erhalten. Welche Bitkombination ist als Daten enthalten?

- |                               |                               |                               |
|-------------------------------|-------------------------------|-------------------------------|
| <input type="checkbox"/> 0000 | <input type="checkbox"/> 0001 | <input type="checkbox"/> 0010 |
| <input type="checkbox"/> 0011 | <input type="checkbox"/> 0100 | <input type="checkbox"/> 0101 |
| <input type="checkbox"/> 0110 | <input type="checkbox"/> 0111 | <input type="checkbox"/> 1000 |
| <input type="checkbox"/> 1001 | <input type="checkbox"/> 1010 | <input type="checkbox"/> 1011 |
| <input type="checkbox"/> 1100 | <input type="checkbox"/> 1101 | <input type="checkbox"/> 1110 |
| <input type="checkbox"/> 1111 |                               |                               |

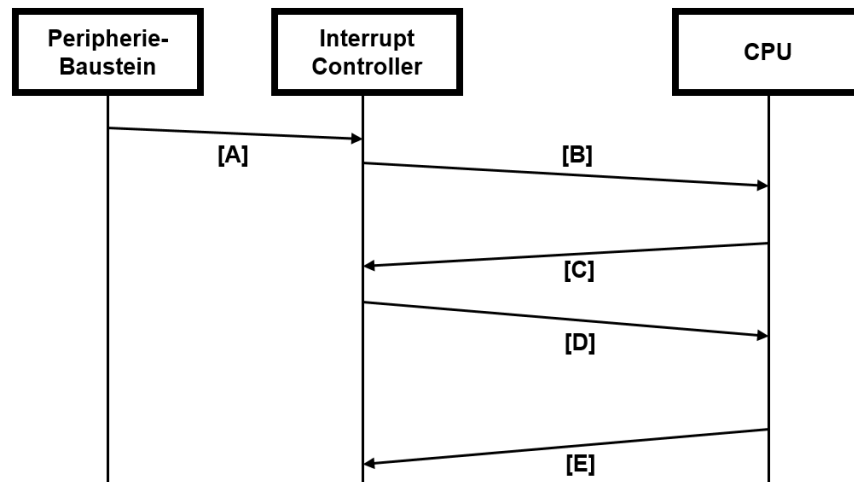


6.9 Sie haben die Signale in einer **Manchester-Codierung nach G.E. Thomas** erhalten. Welche Bitkombination ist als Daten enthalten?

- |                               |                               |                               |
|-------------------------------|-------------------------------|-------------------------------|
| <input type="checkbox"/> 0000 | <input type="checkbox"/> 0001 | <input type="checkbox"/> 0010 |
| <input type="checkbox"/> 0011 | <input type="checkbox"/> 0100 | <input type="checkbox"/> 0101 |
| <input type="checkbox"/> 0110 | <input type="checkbox"/> 0111 | <input type="checkbox"/> 1000 |
| <input type="checkbox"/> 1001 | <input type="checkbox"/> 1010 | <input type="checkbox"/> 1011 |
| <input type="checkbox"/> 1100 | <input type="checkbox"/> 1101 | <input type="checkbox"/> 1110 |
| <input type="checkbox"/> 1111 |                               |                               |

## 7. Hardware-Interrupts (12 Punkte)

Das folgende Bild zeigt den **typischen Ablauf eines Interrupts** zwischen den beteiligten Hardwarekomponenten eines Mikrocontrollers wie des beispielhaft verwendeten AT91M63200. Die Platzhalter [A] bis [E] stehen für die einzelnen Schritte des Ablaufs, der bei der Bearbeitung eines Interrupts durchlaufen wird. Hierfür steht der CPU eine **Interrupt Service Routine ISR** zur Verfügung, deren Sprungadresse im Interrupt Controller konfiguriert wurde.



- 7.1 Überprüfen Sie, **welche Beschreibung im Ablauf** den jeweiligen Schritten [A] bis [E] zuzu ordnen ist. Kreuzen Sie **nur passende** Beschreibungen an. Nicht für alle Schritte muss es eine passende Beschreibung geben.

|                                                                                | <input checked="" type="checkbox"/> [A] | <input checked="" type="checkbox"/> [B] | <input checked="" type="checkbox"/> [C] | <input checked="" type="checkbox"/> [D] | <input checked="" type="checkbox"/> [E] |
|--------------------------------------------------------------------------------|-----------------------------------------|-----------------------------------------|-----------------------------------------|-----------------------------------------|-----------------------------------------|
| Interruptanforderung einer HW-Komponente                                       | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                |
| Unterbrechung des laufenden Assemblerbefehls                                   | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                |
| CPU will den Inhalt des Interrupt Vektor Registers lesen                       | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                |
| Rücksprung aus ISR mittels Branch-Befehl                                       | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                |
| Lädt die Adresse der ISR aus dem Interrupt Vektor Register in ein CPU-Register | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                |

- 7.2 Überprüfen Sie, **welche Angaben** Sie den jeweiligen Schritten [A] bis [E] zuordnen können. Kreuzen Sie **nur passende** Angaben an. Nicht für alle Schritte muss es eine passend Angabe geben.

|                                                                                                                            | <input checked="" type="checkbox"/> [A] | <input checked="" type="checkbox"/> [B] | <input checked="" type="checkbox"/> [C] | <input checked="" type="checkbox"/> [D] | <input checked="" type="checkbox"/> [E] |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|-----------------------------------------|-----------------------------------------|-----------------------------------------|-----------------------------------------|
| Lässt die CPU wieder unterbrechbar und damit reentrant-fähig werden                                                        | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                |
| Beschreibt das End of Interrupt Command Registers mit einem beliebigen Wert.                                               | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                |
| Besitzt eine Priorität von 7 bis 0, so dass der Interrupt Controller parallele Interrupts in eine Reihenfolge bringen kann | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                |
| Erzeugt einen Stack für die ISR                                                                                            | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                |
| Signalisiert der CPU, dass ein Interrupt bearbeitet werden soll                                                            | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                | <input type="checkbox"/>                |