„Mikrorechner-Technik“ – Kurs 1708

Lösungsvorschläge zur

Klausur im WS 2004/05

am 05.02.2005

(Ohne Gewähr für die Fehlerfreiheit)

Zu Aufgabe 1:

Geben Sie für die folgenden Aussagen an, ob sie richtig oder falsch sind.

	richtig	falsch
a) Für die Realisierung eines **DDR-RAMs** *(Double Data Rate)* muß die Zugriffsgeschwindigkeit auf seine Speicherzellen gegenüber konventionellen DRAM-Zellen verdoppelt werden. *Die Zugriffsgeschwindigkeit bleibt gleich. Es wird nur doppelt so schnell auf die Register im Speicherbaustein zugegriffen, die mehr als ein Datum bereithalten.*		X
b) **Digitale Signalprozessoren** verfügen oft über mehrere schnelle Parallelports, über die sie mit anderen Digitalen Signalprozessoren kommunizieren können. *Die sog. Link Ports mit typ. 8 bit Breite erlauben Übertragungsraten zw. DSPs, die von der Taktfrequenz der Prozessoren abhängen. Beispiele: Analog Devices SHARC, Texas Instruments TMS320C40/44.*	X	
c) Der Controller eines Cache-Speichers wird auch als **Speicherverwaltungseinheit** bezeichnet. *Die Speicherverwaltungseinheit übernimmt die Adreß-Umsetzung bei der virtuellen Speicherverwaltung.*		X

d) Geben Sie für die **FIFO-** und **LRU-Ersetzungsstrategie** bei der Cache-Verwaltung an, wofür die Abkürzungen stehen und welchen Eintrag sie ersetzen, wenn Platz für einen neuen Eintrag benötigt wird:

FIFO: First-In, First-Out: Es wird der <u>zuerst eingeschriebene</u> Eintrag verdrängt.

LRU: Least-Recently Used: Es wird der Eintrag verdrängt, auf den <u>am längsten nicht mehr</u> zugegriffen wurde.

e) Der Vorgang, der zur Erhaltung der gespeicherten Information in einem **dynamischen RAM** nötig ist, wird als *Auffrischen (Refresh)* bezeichnet und muß regelmäßig in Zeitintervallen mit einer Dauer von *2 ms* bis *64 ms bzw. 128 ms* durchgeführt werden.

f) Wie werden die speziellen Rechenwerke eines **Digitalen Signalprozessors** (DSP) genannt, die die Auswahl der Operanden im Speicher zur Aufgabe haben ?

Adreß-Rechenwerke (Address Generation Units)

Wie viele dieser Rechenwerke muß ein DSP wenigstens besitzen ? (Begründung!)

Erforderliche Anzahl: 2, da in vielen DSV-Algorithmen pro Takt auf zwei Operanden zugegriffen werden muß.

Zu Aufgabe 2: Datentypen und Datenformate

a) Für die <u>vorzeichenbehafteten</u> Festpunktzahlen Z im 1.15-Format gilt die folgende Darstellung in Potenzschreibweise zur Basis 2:

$$Z = d_0 \cdot (-2^0) + d_1 \cdot 2^{-1} + d_2 \cdot 2^{-2} + ... + d_{15} \cdot 2^{-15} \quad .$$

In Binärschreibweise gilt: $Z = d_0.d_1...d_{15}$ mit $d_i \in \{0, 1\}$

Natürlich kann man die Zahlen auch als <u>vorzeichenlose</u> ganze Zahlen *(Integer)* interpretieren und erhält dann:

$Z_I = d_0d_1...d_{15}$ mit $0 \leq Z_I \leq 65535$. Dabei sei $d_0 = MSB(Z_I)$. *(Most Significant Bit)*

Beweisen Sie die folgende Umrechnungsvorschrift für Z aus Z_I:

$$Z = Z_I \cdot 2^{-15} - 2 \cdot d_0$$

Herleitung:

$$Z_I \cdot 2^{-15} = (d_0 \cdot 2^{15} + d_1 \cdot 2^{14} + ... + d_{15} \cdot 2^0) \cdot 2^{-15} = d_0 \cdot 2^0 + d_1 \cdot 2^{-1} + ... + d_{15} \cdot 2^{-15}$$

$$Z_I \cdot 2^{-15} - 2 \cdot d_0 = d_0 \cdot 2^0 + d_1 \cdot 2^{-1} + ... + d_{15} \cdot 2^{-15} - 2 \cdot d_0 \cdot 2^0 = d_0 \cdot (-2^0) + d_1 \cdot 2^{-1} + ... + d_{15} \cdot 2^{-15}$$

$$= Z$$

b) Der Vergleich zweier Festpunktzahlen Z und Z‘ kann durch die Vergleicherschaltung (Komparator) einer Integer-ALU durchgeführt werden, indem die beiden Zahlen als vorzeichenlose ganze Zahlen Z_I und Z'_I interpretiert und verglichen werden. Dies sollen Sie hier schrittweise beweisen. Dazu seien $Z \neq Z'$ und wie unter a) $d_0 = MSB(Z_I)$ und $d_0' = MSB(Z'_I)$.

i) Es seien: $d_0 \neq d_0'$, d.h. die beiden Zahlen Z und Z‘ haben entgegengesetzte Vorzeichen.
Zeigen Sie:

$$Z_I > Z'_I \Leftrightarrow Z < Z'$$

$$Z_I > Z'_I \quad \Leftrightarrow \quad d_0 = 1, d_0' = 0 \quad \Leftrightarrow \quad Z < 0, Z' > 0 \quad \Leftrightarrow \quad Z < Z'$$

ii) Es gelte nun: $d_0 = d_0'$, d.h. die beiden Zahlen Z, Z‘ haben dasselbe Vorzeichen.
Zeigen Sie nun mit der Formel aus a):

$$Z_I > Z'_I \Leftrightarrow Z > Z'$$

$$Z_I > Z'_I \Leftrightarrow (Z + 2 \cdot d_0) \cdot 2^{15} > (Z' + 2 \cdot d_0') \cdot 2^{15} \quad \Leftrightarrow Z + 2 \cdot d_0 > Z' + 2 \cdot d_0'$$

$$\Leftrightarrow Z > Z' \qquad \text{da } d_0 = d_0'$$

c) Die Aussage von b) gilt auch für die <u>Beträge</u> von 32-bit-Gleitpunktzahlen nach dem IEEE-754-Standard, d.h. man kann diese dadurch vergleichen, daß man sie als vorzeichenlose ganze Zahlen interpretiert und durch eine Integer-ALU bearbeiten läßt.

i) Geben Sie an, durch welche Operation man den Betrag einer 32-bit-Zahl nach dem IEEE-754-Standard ermitteln kann.

Es reicht, das MSB = 0 zu setzen. Positive Zahlen bleiben dadurch positiv, negative werden positiv.

ii) Gegeben seien nun zwei 32-bit-Zahlen Z und Z‘ und ihre Beträge $|Z|$ und $|Z'|$. Die Beträge werden wie unter a) als vorzeichenlose ganze Zahlen interpretiert und dann mit $|Z|_I$ und $|Z'|_I$ bezeichnet. Zeigen Sie:

$$|Z|_I > |Z'|_I \;\Rightarrow\; |Z| > |Z'|$$

D.h. wenn die als vorzeichenlose ganze Zahlen aufgefaßten Beträge der Zahlen in einem bestimmten Größenverhältnis stehen, so stehen auch die Beträge selbst in diesem Verhältnis.

Sei n, $0 \le n \le 30$, der Index n des höchstwertigen Bits, in dem sich die beiden Zahlen $|Z|_I$ und $|Z'|_I$ unterscheiden. Aus der vorgegebenen Relation folgt dann, daß dieses Bit in $|Z|_I$ gleich 1, in $|Z'|_I$ gleich 0 ist.

1. Fall: $23 \le n \le 30$, d.h. das Bit n gehört zur Charakteristik. Da die Charakteristik im IEEE-754-Format stets positiv ist, folgt, daß die Charakteristik von Z größer ist als die von Z‘, und zwar wenigstens doppelt so groß. Der Betrag der Mantisse ist jedoch kleiner als 2, so daß eine größere Mantisse von Z‘ nicht die größere Charakteristik von Z aufwiegen kann.

2. Fall: $22 \le n \le 0$, d.h. das Bit n gehört zur (vorzeichenlosen) Mantisse. Dann ist die Mantisse von Z größer als die von Z‘. Da n der höchste Index ist, in dem sich beide Beträge unterscheiden, stimmen die Charakteristiken beider Zahlen überein. Das heißt aber, daß $|Z| > |Z'|$ ist.

Aufgabe 4:

a) Ein Mikroprozessor besitze einen jeweils 16 bit breiten Adreß- und Datenbus. Sein integrierter Interruptcontroller verwalte bis zu 7 Interruptquellen, die sich über eine Interruptvektor-Nummer IVN$\in\{1,2,...,7\}$ identifizieren lassen. Die IVN=0 kennzeichnet den Zustand „keine Interruptanforderung“. Im Arbeitsspeicher liege die folgende Interruptvektor-Tabelle, deren Einträge im Little-Endian-Format aufzufassen sind:

Byte Adresse	0	1	2	3	4	5	6	7
A780	d	d	76	70	80	B0	04	C0
A788	A0	C0	66	50	44	20	7C	99

d: don’t care

Der Beginn der Interruptvektor-Tabelle werde durch ein 16-bit-Register IBR festgelegt.

Geben Sie die Belegung des IBR für die oben stehende Interruptvektor-Tabelle an:

IBR = $A780.

Geben Sie die Formel an, nach der aus der IVN und dem Inhalt des Basisregisters IBR die Startadresse der zugehörigen *Interrupt Service Routine* berechnet wird:

Startadresse ISR = IVN * 2 + IBR

Tragen Sie in die folgende Tabelle für alle Interruptvektor-Nummern die Startadresse der *Interrupt Service Routine* (ISR) ein:

IVN	**Startadresse**	**Priorität**
1	7076	↓
2	B080	
3	C004	
4	C0A0	
5	5066	
6	2044	
7	997C	

Die Abarbeitung gleichzeitig anliegender Unterbrechungswünsche werde vom Interruptcontroller durch Prioritäten gesteuert, wobei **eine höherwertige IVN eine höhere Priorität** bedeute.

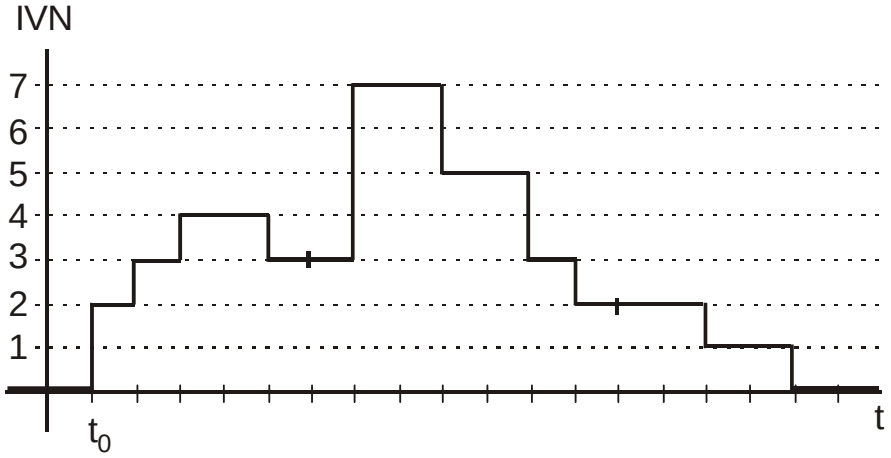
Zeichnen Sie nun in der mit „Priorität“ bezeichneten Spalte der oben stehende Tabelle einen Pfeil von der niedrigsten zur höchsten Priorität!

b) Der Interruptcontroller des Mikroprozessors unterstütze die Verschachtelung von Interrupts, d.h. er erlaube die Unterbrechung einer *Interrupt Service Routine* (ISR1) durch eine andere Routine (ISR2). Diese Routine muß dazu lediglich eine höhere Priorität besitzen, d.h. für ihre Interruptvektor-Nummern muß gelten: **IVN2 > IVN1**.

Ihre Aufgabe ist es nun, die Ausführung einer Reihe von Unterbrechungsanforderungen zu untersuchen:

- Zu Beginn des Beobachtungszeitraums t_0 werde eine ISR mit IVN=2 ausgeführt.
- Während ihrer Bearbeitung trete ein Interrupt der Priorität 3 auf.
- Während der Bearbeitung dieses Interrupts werden weitere Interrupts der Prioritäten 1 und 4 sowie erneut der Priorität 3 angefordert.
- Während der Bearbeitung des zweiten Interrupts der Priorität 3 soll der Reihe nach
  - ein Interrupt der Priorität 7 und <u>während</u> dessen Bearbeitung ein Interrupt der Priorität 5 auftreten,
  - nach Beendigung des Interrupts mit der Priorität 5 ein erneuter Interrupt mit der Priorität 2 erfolgen.
- Nach Abarbeitung aller ISR soll das Hauptprogramm (mit der Priorität 0) fortgesetzt werden.

Tragen Sie in das folgende Diagramm alle Übergänge zwischen den Interrupt Service-Routinen und ihre Verläufe ein. Gehen Sie vereinfachend davon aus, daß jede Ausführung einer ISR genau <u>zwei</u> Zeiteinheiten dauert und Übergänge nur zu den markierten Zeitpunkten geschehen können. Nehmen Sie weiter an, daß eine während einer Interruptbearbeitung auftretende weitere Anforderung höherer Priorität nach der 1. Zeiteinheit der beiden Zeiteinheiten bedient wird. Kennzeichnen Sie den Übergang von einer ISR zu einer ISR mit derselben IVN durch einen kleinen senkrechten Trennstrich.



c) Kann eine ISR mit der IVN=7 selbst unterbrochen werden? (Begründung!)

Da für keine andere Routine die Priorität größer als 7, also wenigstens 8, sein kann, kann diese ISR nicht unterbrochen werden.

Welche Rolle übernimmt ein Interrupt mit der IVN=7? (Begründung!)

Da eine Anforderung mit IVN=7 jede andere ISR und natürlich auch jedes Programm unterbrechen kann, entspricht sie dem **NMI** (*Non Maskable Interrupt*).

Zu Aufgabe 4: CAN-Bus

a) Auf dem CAN-Bus werden die in der folgenden Tabelle (in Hexadezimalform) dargestellten Bitfolgen (ohne Startbits) beobachtet. Dabei entspreche das MSB *(Most Significant Bit)* dem Bit ID28 des Identifikationsfeldes ID28,....,ID18 bzw. ID28,...,ID0.

Nr.	Bitfolge (in Hexadezimalform)	Format (S/E)	Identifikationen (hex.)	Rahmentyp (D/A)	Datenfeldlänge (DLC)
1	**$4339AC5D00**	E	**$4339AC5D**	A	0
2	**$39A18ABCDE**	S	**$39A**	D	6
3	**$FE9BDCFE10**	E	**$FE9BDCFE**	D	4
4	**$3A7A725900**	E	**$3A7A7259**	A	0
5	**$3C616ADEF0**	S	**$3C6**	D	5

i. Ergänzen Sie die Tabelle um das Format des Identifikationsfeldes S/E („Standardformat“ oder „Erweitertes Format“), um die Identifikationen der Nachrichten und ihren Typ D/A („Datenrahmen“ bzw. „Anforderungsrahmen“) sowie die Länge des übertragenen Datenfeldes DLC. (Der Einfachheit halber können Sie die Kennung – Identifikation – des erweiterten Formats inklusive der eingeschlossenen Bits RTR, IDE, SRR und ‚reserviert’ angeben.)

- Das Format wird durch das IDE-Bit, das höchstwertige Bit der 4. Tetrade von links vorgegeben:
  Standard-Format: IDE=0, Erweitertes Format: IDE = 1.
- Der Rahmentyp wird durch das RTR-Bit festgelegt, das im Standard-Format das niederstwertige Bit der Identifikation ist, im erweiterten Format das niederstwertige Bit der 8.Tetrade von links:
  Datenrahmen: RTR = 0, Anforderungsrahmen: RTR = 1.
- Das Daten-Längenfeld DLC wird im Standardformat durch die vier mittleren Bits der 4. und 5. Tetrade von links, im erweiterten Format durch die vier mittleren Bits in der 9. und 10. Tetrade.

ii. Welche der ermittelten Identifikationen ist ungültig und warum? (Begründung)

Ident.-Nr.: 3 , weil die oberen 7 Bits sämtlich rezessiv, also ‚1’ sind.

iii. In welcher Reihenfolge scheiden die Nachrichten bei der Bus-Arbitrierung aus? Stellen Sie den bitweisen Ablauf der Arbitrierung für die ersten 12 Bits der Identifikationen dar. Kennzeichnen Sie dabei die Bits, in denen die einzelnen Nachrichten unterliegen.

1.	_0_	_**1**_	_0_	_0_	_0_	_0_	_1_	_1_	_0_	_0_	_1_	_1_
2.	_0_	_0_	_1_	_1_	_1_	_0_	_0_	_1_	_1_	_0_	_1_	_0_
3.	_**1**_	_1_	_1_	_1_	_1_	_1_	_1_	_0_	_1_	_0_	_0_	_1_
4.	_0_	_0_	_1_	_1_	_1_	_0_	_**1**_	_0_	_0_	_1_	_1_	_1_
5.	_0_	_0_	_1_	_1_	_1_	_**1**_	_0_	_0_	_0_	_1_	_1_	_0_
Buszustand:	_0_	_0_	_1_	_1_	_1_	_0_	_0_	_1_	_1_	_0_	_1_	_0_

(Die weiß unterlegten, fettgedruckten Stellen zeigen an, wann die zugehörigen Nachrichten unterliegen und ausscheiden.)

Reihenfolge des Ausscheidens: 3 (ungültig) – 1 – 5 – 4

iv. Welche binären Werte müssen im Akzeptanz-Kennungsregister AKR und Akzeptanz-Maskenregister AMR eines Knotens stehen, der nur Nachrichten vom Typ Nr. 2 empfangen soll?

AKR = 00111001_2 = $39, da es mit den oberen 8 Bits der Kennung übereinstimmen muß.

AMR = 00000000_2 = $00, da jede Stelle an der Auswahl beteiligt sein muß, also keine Stelle maskiert werden darf.

v. Welcher Wert steht nach dem Arbitrierungsverfahren im *Arbitration Lost Capture Register* ALCR des Knotens, der die Nachricht Nr. 5 aussenden wollte?

ALCR = 23 = $17, da dieser Knoten in der Bitposition 6 von links unterliegt, die dem Identifikationsbit ID23 entspricht.

b) Wie im Kurs beschrieben, wird auf dem CAN-Bus das Verfahren des *Bit Stuffings* durchgeführt, bei dem der Sender nach jeweils 5 Bits gleicher ‚Polarität' (rezessiv bzw. dominant) ein Bit der entgegengesetzten Polarität einfügt, das vom Empfänger dann wieder entfernt werden muß. Kennzeichnen Sie in den folgenden Teilaufgaben alle Bits, die durch das *Bit Stuffing* bei der Übertragung auf dem CAN-Bus hinzugefügt werden. (Dabei sei jeweils angenommen, daß die Bits in der Reihenfolge von links nach rechts übertragen werden, d.h. das am weitesten links stehende Bit zuerst, das am weitesten rechts stehende zuletzt.)

i. Welche Bitfolge wird auf dem CAN-Bus übertragen, wenn der Sender die folgenden, in binärer Form vorliegenden Data übertragen will ?

Sender CAN-Bus

0010011101111110100000000000110... ⇒ 0010011101111110101000001000001110

ii. Welche Daten übernimmt der Empfänger, wenn er auf dem CAN-Bus die folgende Bitfolge liest ?

CAN-Bus Empfänger

10100000100111110001000000100110... ⇒ 1010000000111110010000001 10.....

c) Ein Knoten sende nach seinem Anschalten an den CAN-Bus eine Folge von korrekten (d.h. fehlerfreien) und fehlerhaften Nachrichten, die durch die Kennung k bzw. f bezeichnet werden:

Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13
Kennung	k	k	f	k	f	k	k	k	f	k	k	k	k
Zähler	0	0	8	7	15	14	13	12	20	19	18	17	16

i. Tragen Sie in die letzte Zeile der oben stehenden Tabelle den jeweils vorliegenden Wert des Sende-Fehlerzählers *(Tx Error Counter)* ein.

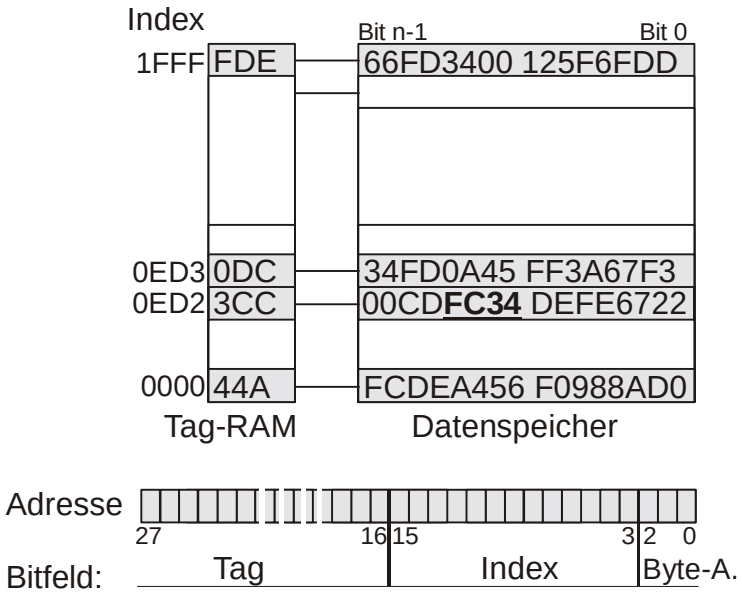
ii. Die fehlerhaft ausgesandten Nachrichten des Knotens seien näherungsweise gleichmäßig in seinem Nachrichtenstrom verteilt. Geben Sie an, wie groß der Prozentsatz

der fehlerhaften Nachrichten höchstens sein darf, wenn der Knoten nicht (fast sicher) in den *Bus Off*-Zustand gehen soll. (Begründung erforderlich.)

Im Mittel müssen nach jeder fehlerhaften Nachricht wenigstens 8 fehlerfreie gesendet werden, um die Erhöhung des Fehlerzählers um 8 wieder rückgängig zu machen. Daraus folgt, daß (im Mittel) höchstens jede 9. Nachricht fehlerhaft sein darf. Dies entspricht einem Prozentsatz von ca. 11%.

Zu Aufgabe 5:

Das folgende Bild zeigt den Aufbau eines direkt abbildenden Caches *(Direct Mapped Cache)* mit einem Ausschnitt seiner aktuellen Belegung.



a) Kennzeichnen Sie im unteren Teil des Bildes die unterscheidbaren Bitfelder einer Speicheradresse ein, wie sie vom Cache-Controller benötigt werden! Benennen Sie diese Bitfelder und ergänzen Sie die Nummer der Bits an den Grenzen der Bitfelder. Aus wie vielen Bits besteht eine Speicheradresse, wenn der gesamte Adreßraum im Cache abgelegt werden kann? Wie groß ist der damit ansprechbare Adreßraum? (Geben Sie für jede Ihrer Angaben eine Begründung bzw. Berechnung an!)

Jeder Eintrag ist 8 byte lang. Daher umfaßt die Byte-Auswahl die unteren 3 Bits.
Es gibt $1FFF+1 = $2000 = 8192 Einträge. Also ist der Index 13 bit lang.
Jeder Tag besteht aus 3 Tetraden, ist also 12 bit lang.
Anzahl der Adreßbits: 12 + 13 + 3 = 28 bit
Adreßraum-Größe: 2^{28} byte = 2^{8} Mbyte = 256 Mbyte

b) Berechnen Sie die Gesamt-Kapazität in kbyte des im Bild dargestellten Caches aus der Kapazität des Tag-RAMs und der des Datenspeichers!

Kapazität des Tag-RAMs:	2^{13} * 1,5 byte = 12 kbyte
Kapazität des Datenspeichers:	2^{13} * 8 byte = 64 kbyte
Gesamt-Kapazität:	(12 + 64) kbyte = 76 kbyte

c) Beantworten Sie die beiden folgenden Fragen und begründen Sie Ihre Antwort!

- Ist im dargestellten Cache-Ausschnitt aus vier Einträgen der aktuelle Wert der Speicherzelle $0DC5660 abgelegt?

nein, weil der Index (Bits 15 – 3) in der gegebenen Speicheradresse $0DC5660 = 00001101110 0 1010 1100 1100 000 gleich $0ACC ist und dieser Index nicht zum dargestellten Ausschnitt des Tag-RAMs (Indizes: $1FFF, $0ED3, $0ED2, $0000) gehört.

- Welchen Wert erhält der Prozessor, wenn er mit der Adresse $3CC7694 einen 16 bit breiten Lesezugriff durchführt?

  Gelesener Wert: $FC34, weil der Index = 0ED2 und darunter der *Tag* $3CC gespeichert ist, der mit dem oberen Teil der Adresse übereinstimmt. Daher liegt ein *Hit* vor. Im Datenspeicher findet sich ab der Byte-Adresse 4 der angegebene 16-bit-Wert.