

Zulässiges Material: Tabelle der booleschen Funktionen, Zahlentabelle

0-16 (binär/oktal/dezimal/hexadezimal), KV-Diagramm-Formular, Intel-Befehlstabelle.

Dauer: 90 Minuten. Aufgabenblatt bitte mit Name beschriften und als Deckblatt mit abgeben. Reklamationen nur bei Rückgabe. Bei Nichtabholung 1 Punkt Abzug.

**Aufgabe 1**

Wandeln Sie die Hexadezimalzahl 7A5E3 in das Oktalsystem. *J. Blatt 7*

Punkte: 1/2 0

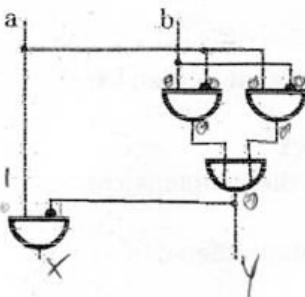
**Aufgabe 2**

Ein Intel-Prozessor führe die nebenstehenden Befehle aus. Was steht an den mit „?“ markierten Stellen jeweils in Register AL (binär oder hexadezimal!) sowie im Carry- und Overflow-Flag?

```
mov AL, 5Bh
add AL, -41
; ?
mov AL, 5Bh
sub AL, 25h
; ?
```

Punkte: 2 1/2 1/2 0

**Aufgabe 3**

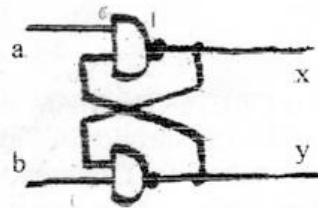


Analysieren Sie die abgebildete Schaltung. Stellen Sie die Wertetafel für die Ausgänge auf. Wie heißt diese Ihnen bekannte Schaltfunktion? Welche Schaltzeit hat die Schaltung bei einer Gatterdurchlaufzeit von  $t_G = 3,8 \text{ ns}$ ?

Punkte: 1 1/2 1 0

**Aufgabe 4**

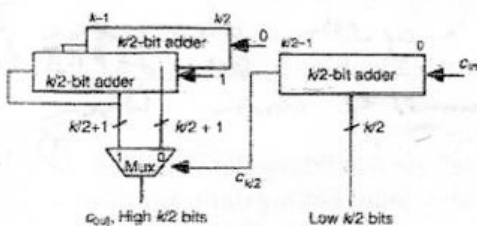
Charakterisieren Sie das Verhalten der abgebildeten Schaltung: Nehmen Sie an, die Eingangswerte a und b werden in der Reihenfolge der Tabelle durchlaufen; bestimmen Sie jeweils die Ausgangswerte und tragen sie in die Tabelle ein. Wie heißt so eine Schaltung?



| a | b | x | y |
|---|---|---|---|
| 0 | 1 |   |   |
| 1 | 1 |   |   |
| 1 | 0 |   |   |
| 1 | 1 |   |   |
| 0 | 0 |   |   |

Punkte: 1 1/2 1 0

**Aufgabe 5**



Der Conditional-Sum-Addierer berechnet in der ersten Ebene für jede Stelle die Summe zweimal, nämlich in jeweils einer Variante für den Fall, dass der Übertrag aus der vorherigen Stelle 0 bzw. 1 ist. Hierfür wird jeweils ein Halbaddierer oder Volladdierer benötigt. Es entstehen Zwischenergebnisse der Länge 1 Bit.

Punkte: 1 1/2 0

In jeder weiteren Ebene werden jeweils 2 Zwischenergebnisse der Länge  $k/2$  Bit zu einem neuen Zwischenergebnis der Länge  $k$  Bit zusammengefasst, wobei aus den zuvor ermittelten Ergebnisvarianten diejenigen ausgewählt werden, die zu den nunmehr ermittelten Überträgen passen (s. Bild, MUX). Die Schaltung jeder Ebene (Addierer, MUX) sei 2-stufig realisiert. In jedem Schritt entstehen also Zwischenergebnisse der doppelten Länge, bis zum Schluss das Gesamtergebnis in der vollen Länge berechnet wurde. Wie schnell berechnet der Conditional-Sum-Addierer mit  $t_G = 3,8 \text{ ns}$  Zahlen der Längen 128 Bit bzw. 256 Bit?

**Aufgabe 6**

Beachten Sie die Rückseite!

Punkte: 0

| Bewertung | Punkte | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   |
|-----------|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Note      |        | 1,0 | 1,3 | 1,7 | 2,0 | 2,3 | 2,7 | 3,0 | 3,3 | 3,7 | 4,0 | 5,0 |

### Aufgabe 7

Konstruieren Sie einen synchronen Zähler (eine Schaltung mit zwei Flipflops), der zyklisch die zweistelligen Zustände 00, 11, 10, ... durchläuft. Der vierte Zustand kommt dabei nicht vor; sein Nachfolgezustand kann für die Konstruktion beliebig gewählt werden.

Punkte: 2 1/2 1 1/2 0

### Aufgabe 8

- Auf welche Weise trägt die dargestellte Interruptroutine dazu bei, dass andere Programme durch ihre Aktivierung nicht im Ablauf gestört werden? Markieren Sie!
- Welche Anweisungen fehlen? Fügen Sie sie an den richtigen Stellen ein!

```

intr_handler:
  push AX      push BX
  mov AX, 1; zeige Interrupt an
  mov flag, AX
  mov BX, 92Ah
  mov intr_type, BX
  pop AX
  iret        ; Rückkehr aus Interrupt
  
```

Punkte: 0 1/2 0

### Aufgabe 9

Welche Komponente eines Rechners führt jeweils die beschriebene Aufgabe aus? Wählen Sie aus folgenden Komponentennamen die jeweils speziellste: ALU, CPU, CSU, MMU, FPU, Cache

Punkte: 0 1/2 1 1/2 0

- CPU Fordere nächsten Befehl aus dem Speicher an
- Cache Prüfe, ob angefordertes Speicherwort schneller zur Verfügung gestellt werden kann
- CPU Erhöhe Programmzähler nach der Befehlsabarbeitung
- CPU Rette bei Interruptbearbeitung Rücksprungadresse auf dem Stack
- MMU Ermittle aus der Adresse, an der ein Programm Daten lesen will, die Hauptspeicheradresse, an der sich diese befinden
- MMU Prüfe, ob ein angeforderter Speicher-Lese- oder Schreibzugriff stattfinden darf
- ALU 3+5
- FPU 3.7+2.8

### Aufgabe 10

Ein Computer habe eine 1.5GHz-CPU, 768 MB DRAM als Hauptspeicher, 1MB SRAM als Cache, eine Festplatte mit 80GB. Er wird hauptsächlich für Textverarbeitung genutzt. Die Nutzer klagen allgemein über „langsam“ Verhalten des Computers.

Punkte: 3/4 0

Welche Maßnahme zur Geschwindigkeitsverbesserung würden Sie empfehlen und warum?

- Austausch der CPU durch eine 3GHz-CPU
- Austausch des Hauptspeichers durch SRAM *statischer RAM ist schneller als dynamische RAM.*
- Ergänzung des Hauptspeichers um 1GB ? *768MB RAM sollte genug sein, damit der PC nicht anlegen muss*
- Einbau einer weiteren Festplatte *mag sein aber wenn Text-...*
- erwartete Leistungsverbesserung durch empfohlene Maßnahme: *bei 2GB RAM langsam ist, muss es daran liegen*

### Aufgabe 11

Die virtuelle Speicherverwaltung MMU führt eine „Seitentabelle“ zur Abbildung der virtuellen Adressen des Programms auf reale Adressen des Hauptspeichers. Jeder Eintrag darin kann verschiedene Attribute haben. Welche der folgenden Attribute gibt es und was bewirken sie?

Punkte: 2 1/2 1 1/2 0

- ausgelagert *Speichereinträge auf Datenträger gespeichert*
- gültig *Im Speicher vorhanden,*
- sticky *Darf nicht verschoben werden.*
- slimy *ausgelagert*
- dirty *Kein Zugriff möglich.*
- muddy
- schreibgeschützt *Darf nicht verändert werden - nur Lesezugriff (read-only)*

1)  $7A5E3_{(16)}$

0111 | 1010 | 0101 | 1110 | 0011

1722743<sub>(8)</sub>

2)  $5B_{(16)} - 47_{(16)} \rightarrow$

$01011011$   
 $+ 11101011$   
1100110010 ← Erstes Fragezeichen

Overflow-Flag ist gesetzt  
 Carry-Flag ist gesetzt

47<sub>16</sub> → 20<sub>10</sub>  
 5<sub>16</sub> → 5<sub>10</sub>  
 20<sub>10</sub> - 5<sub>10</sub> = 15<sub>10</sub> = 00101001<sub>2</sub>  
 EK → 11010110  
 ZK → 11010111

$5B_{(16)} \rightarrow$   $01011011$   
 $+ 11101011$   
1100110010 ← zweites Fragezeichen

Overflow- und Carry-Flag sind gesetzt

$25_{(16)} \rightarrow$   $00100101$   
 $\rightarrow EK$  11011010  
 $\rightarrow ZK$  11011011

3)  $t_{GS} = t_G \cdot n \rightarrow t_{GS} = 3,8ns \cdot 2 = 7,6ns$  Gatterdurchlaufzeit 6<sup>ns</sup>

$7,6ns \rightarrow KZ$

| a | b | x | y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

$x = a \cdot b$

Zustand von  $x$  liegt immer an  $x$  an, egal welchen Zustand  $y$  hat.

~~Zustand~~

~~x registriert wie es ist.~~

Halbaddierer ✓

4.) Ist ein Flip-Flop. ✓

| a | b | x                       | y |
|---|---|-------------------------|---|
| 0 | 1 | 1                       | 0 |
| 1 | 1 | unerwünschter Zustand ✗ |   |
| 1 | 0 | 0                       | 1 |
| 1 | 1 | unerwünschter Zustand ✗ |   |
| 0 | 0 | 1                       | 1 |

$60 - 6 = 54 = 48$

5.)  $t_b = 3,8 \text{ ns}$   $t_{cs} = (\log_2 n + 1) \cdot 2 \cdot t_b$   
 128 Bit  
 $t_{cs} = (7 + 1) \cdot 2 \cdot 3,8 \text{ ns} = \underline{60,8 \text{ ns}}$  ✓

Nebenrechnungen  
 $18 \cdot 3,8$   
 $22,8 \cdot 6 \cdot 3,8$   
 $8 \cdot 2 = 16 \cdot 3,8 \text{ ns}$   
 $10 \cdot 3,8 =$

256 Bit  
 $t_{cs} = (8 + 1) \cdot 2 \cdot 3,8 \text{ ns} = \underline{68,4 \text{ ns}}$  ✓

~~$38 + 22,8 =$~~   
 ~~$18 \cdot 3,8$~~   
 $18 \cdot 3,8 \text{ ns}$   
 $60,8 + (2 \cdot 3,8)$   
 $69,6 + 7,6$   
 $67,6 + 1,8$   
 $68,4$   
 $67,6$   
 $3,8$   
 $60,8$   
 $7,6$

7.)

