

Probeklausur Informatik 1

a.)

$$\begin{aligned}87 &= 64 + 16 + 4 + 2 + 1 \\ &= 2^6 + 2^4 + 2^2 + 2^1 + 2^0 \\ &= [0101.0111]_2\end{aligned}$$

$$\begin{aligned}41 &= 32 + 8 + 1 \\ &= 2^5 + 2^3 + 2^0 \\ &= [0010.1001]_2\end{aligned}$$

b.)

Überführung von 41 in Zweierkomplementdarstellung:

Methode 1:

$$\begin{array}{l} \text{Invertieren:} \\ \quad [0010.1001]_2 \\ \quad \text{inv.} \Rightarrow [1101.0110]_2 \end{array}$$

$$\begin{array}{l} \text{Addition von 1:} \\ \quad [1101.0110]_2 \\ \quad + [0000.0001]_2 \\ \quad = [1101.0111]_2 = [-41]_{10} \end{array}$$

Methode 2:

Im Zahlenkreis für 8 Bit gibt es $2^8 = 256$ Zahlen. Um herauszufinden, welche positive Zahl die -41 darstellen würde, muss man nur die 41 von der Gesamtmenge der Zahlen für 8-Bit abziehen.

$$256 - 41 = 215$$

Gesucht ist also eigentlich die Zahl 215.

$$\begin{aligned}215 &= 128 + 64 + 16 + 4 + 2 + 1 \\ &= 2^7 + 2^6 + 2^4 + 2^2 + 2^1 + 2^0 \\ &= [1101.0111]_2\end{aligned}$$

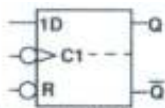
Berechnung von $c = 87 - 41$

Dazu addiert man die 87 und die -41 :

$$\begin{aligned} & [0101.0111]_2 \\ & + [1101.0111]_2 \\ & = [0010.1110]_2 = 2^5 + 2^3 + 2^2 + 2^1 = 32 + 8 + 4 + 2 = 46 \end{aligned}$$

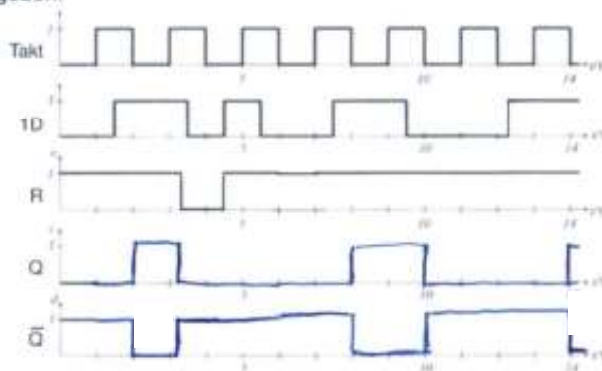
Ergebnis $c = [0010.1110]_2 = 46$

Aufgabe 2:



0 nej. Taktflecke

Ergänzen Sie das nachfolgende Impulsdiagramm. Vernachlässigen Sie dabei Signallaufzeiten zwischen den Ein- und Ausgangssignalen des Speicherbausteins. Die Anfangszustände sind angegeben.



Aufgabe 3:

Zunächst muss man sich überlegen, ab wie vielen Tagen Haft der Häftling bei guter Führung von den beschriebenen Maßnahmen profitiert.

Jenes ist ab 5 Tagen der Fall. Dort würde er dann einen Tag Haft erlassen bekommen. Zum besseren Verständnis sollte man sich eine Tabelle machen, in der man Haft tage und Strafe gegenüberstellt.

Tabelle: Strafe und Haft tage (beides in Tagen)

Strafe	Haft tage
< 5	Haft = Strafe
5	4
6	4
7	5
8	6
9	7
10	8
11	9
12	8

13	9
20	14
30	24

Wichtig für den Algorithmus ist die Berechnung der Haft tage. Aus ihr muss eine Rekursion entwickelt werden. Sie funktioniert wie folgt:

Sitzt ein Straftäter weniger als fünf Tage im Gefängnis, ist die Strafe auch die Zahl der Haft tage. Jenes kann man direkt in Pseudocode schreiben.

```
if Strafe < 5 then
    hafttage = Strafe
```

Für 5 Tage ergibt sich ein Sonderfall, denn hier wird nur ein Tag abgezogen. Auch diesen Fall kann man sofort in den Pseudocode übernehmen. In diesem Fall wird der Pseudocode um den Sonderfall ergänzt:

```
if Strafe ≤ 5 then
    if Strafe = 5 then
        hafttage = 4
    else
        hafttage = Strafe
```

Alle anderen Fälle müssen auf diesen Fall zurückgeführt werden können, damit die Rekursion funktioniert. Um zu sehen, wie man den rekursiven Teil entwickelt, sollte man sich damit beschäftigen, wie die Hafttage berechnet werden.

Beispiel: Strafe = 20 Tage

Sitzt der Häftling 4 Tage ab, werden ihm 2 weitere abgezogen.

$$\rightarrow 20 - 4 = 16 - 2 = 14$$

Er hätte also nach 4 Tagen nur noch 14 statt 16 Tage abzusitzen. Diesen Berechnungsschritt führt man jetzt ein weiteres Mal durch.

$$\rightarrow 14 - 4 = 10 - 2 = 8$$

Nach 8 Tagen in Haft hätte er nur noch 8 vor sich. Der Berechnungsschritt wird noch einmal ausgeführt.

$$\rightarrow 8 - 4 = 4 - 2 = 2$$

Nach 12 Tagen Haft wären noch 2 Tage übrig. An dieser Stelle würde die obere Bedingung für Strafen ≤ 5 greifen und somit wäre die Rekursion beendet.

Nun ist aber die Frage: Wie berechnet man die Hafttage??

Dazu sind zwei Dinge wichtig. Zum einen die 4 Tage, die der Häftling immer erstmal sitzen muss, um Tage erlassen zu bekommen, zum anderen der Rest an Tagen, der ≤ 5 ist und für den sofort die Strafe bestimmt werden kann.

Für Strafe = 20 Tage gilt daher

Hafttage = $4 + 4 + 4 + 2 = 14$ Tage

Diese Rechnung fasst man nun als Algorithmus Ebenen auf, die jede ein eigenes Ergebnis liefern. Die unterste Ebene ist dabei jene, in der die if- Anweisung greift und in der ein festes Ergebnis geliefert wird.

Alle anderen Ebenen addieren je 4 Tage zum Ergebnis hinzu.

Wie die Ebenen definiert sein müssen, geht aus dem rekursiv einfachsten Fall Strafe = 6 hervor.

Beispiel: Strafe = 6

Hafttage : $6 - 4 = 2 - 2 = 0 \rightarrow 4$ Hafttage

Schreibt man die Rechnung als

Hafttage : $6 - 6 = 0 \rightarrow 4$ Hafttage

Kann man schnell erkennen, wie man die Rekursion definieren muss. Die Unterebene, die aufgerufen wird, muss sechs Tage abziehen, und nach dem Schließen der Unterebene wieder 4 Tage addieren. Aus dieser Erkenntnis lässt sich der folgende Pseudocode entwickeln.

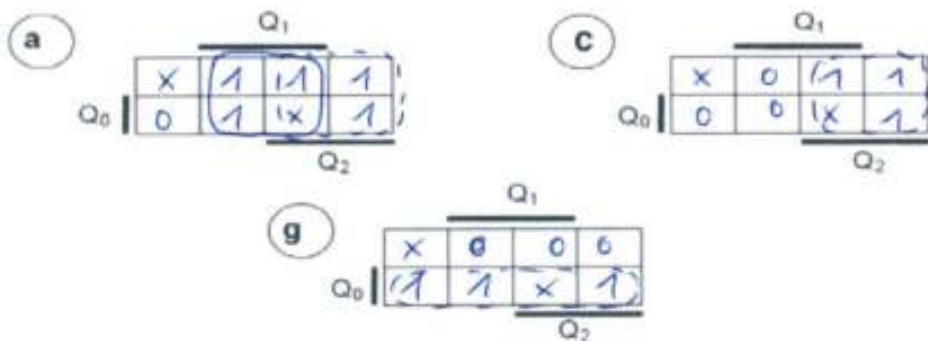
```
function hafttage = RechneHaft (Strafe)
    if Strafe  $\leq$  5 then
        if Strafe = 5 then
            hafttage = 4
        else
            hafttage = Strafe
    else
        hafttage = RechneHaft (Strafe - 6)
        hafttage = hafttage + 4
    fi
End
```

Aufgabe 4:

Funktionstabelle:

Zahl	Q ₂	Q ₁	Q ₀	a	b	c	d	e	f	g
0	0	0	0	x	x	x	x	x	x	x
1	0	0	1	0	0	0	0	0	0	1
2	0	1	0	1	0	0	0	0	1	0
3	0	1	1	1	0	0	0	0	1	1
4	1	0	0	1	0	1	1	0	1	0
5	1	0	1	1	0	1	1	0	1	1
6	1	1	0	1	1	1	1	1	1	0
7	1	1	1	x	x	x	x	x	x	x

KV-Diagramme:



Minimierung:

$$a = Q_1 + Q_2$$

$$c = Q_2$$

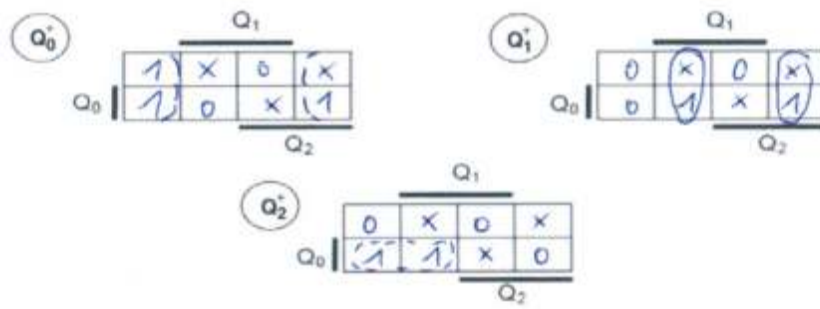
$$g = Q_0$$

Aufgabe 5:

Funktionstabelle:

Zahl	Q ₂	Q ₁	Q ₀	Folgeziffer	Q ₂ ⁺	Q ₁ ⁺	Q ₀ ⁺
0	0	0	0	1	x	x	x
1	0	0	1	5	1	0	1
2	0	1	0	x	x	x	x
3	0	1	1	6	1	1	0
4	1	0	0	x	x	x	x
5	1	0	1	3	0	1	1
6	1	1	0	0	0	0	0
7	1	1	1	x	x	x	x

KV- Diagramme:



Minimierung:

$$Q_0^+ = \bar{Q}_1$$

$$Q_1^+ = (Q_1 \cdot \bar{Q}_2) + (\bar{Q}_1 \cdot Q_2) = Q_1 \oplus Q_2$$

$$Q_2^+ = (Q_0 \cdot \bar{Q}_2)$$