

Wichtig: Das ist eine rekonstruierte Klausur, bei Fragen  
NICHT an den Professor wenden!

Technische Informatik - Sommersemester 2025

## 1. Bitpattern

Für eine 8-Bit Ganzzahl mit Vorzeichen, wie sieht das Bit-Pattern für die Dezimalzahl -6 (minus sechs) im Zweierkompliment aus?

Pattern -6 =  
 1. zuerst Zahl 6 bilden: 6 = 0000 0110  
 2. Zahl invertieren  $\Rightarrow 1111\ 1001$   
 3. +1 addieren  $\Rightarrow 1111\ 1010 = -6$

## 2. Assembly

boo ruft bar auf  
 boo:  $r_1 = r_1 + 5$   
 $r_0 = r_0 + r_1$   
 bar:  $r_0 = r_0 \cdot r_1$   
 (somit mal 7)  
 nach pop{pc}  
 $r_0 = r_0 + 1$

Funktion:

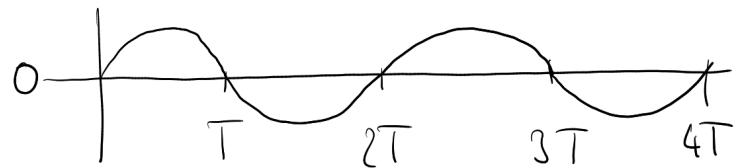
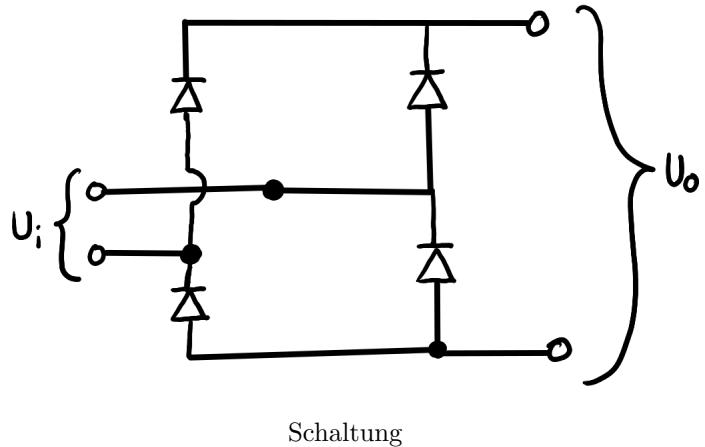
$$f(x, y) = 7 \cdot (x + (y + 5)) + 1$$

A	B	C
<pre>.syntax unified .global boo .global bar  boo:     push {lr}     adds r1, r1, #5     adds r0, r0, r1     bl bar     adds r0, r0, #1     pop {pc}  bar:     push {lr}     movs r1, #7     muls r0, r1, r0     pop {pc}</pre>	<pre>.syntax unified .global boo .global bar  boo:     push {lr}     adds r1, r1, #5     adds r0, r0, r1     bl bar     adds r0, r0, #1     pop {pc}  bar:     movs r1, #7     muls r0, r1, r0     bx lr</pre>	<pre>.syntax unified .global boo .global bar  boo:     adds r1, r1, #5     adds r0, r0, r1     bl bar     adds r0, r0, #1     bx lr  bar:     push {lr}     movs r1, #7     muls r0, r1, r0     pop {pc}</pre> <p>C landet in einem unendlichen Loop</p>

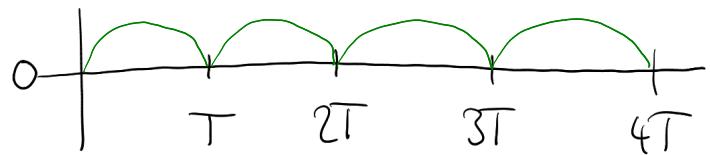
Die Assembly-Codes oben beinhalten zwei Funktionen boo und bar. A und B berechnen das gleiche. Geben Sie den Ausdruck in üblicher mathematischer Schreibweise mit Funktion an. C ist fehlerhaft. Warum?

### 3. Diode

Die Schaltung enthält vier gewöhnliche Dioden. Dargestellt ist die Input-Spannung  $U_i$ . Skizzieren Sie unten die Output-Spannung:



$U_i$ : Soll einen Sinus darstellen



$U_o$ : Hier einzeichnen

## 4. KV-Diagramm

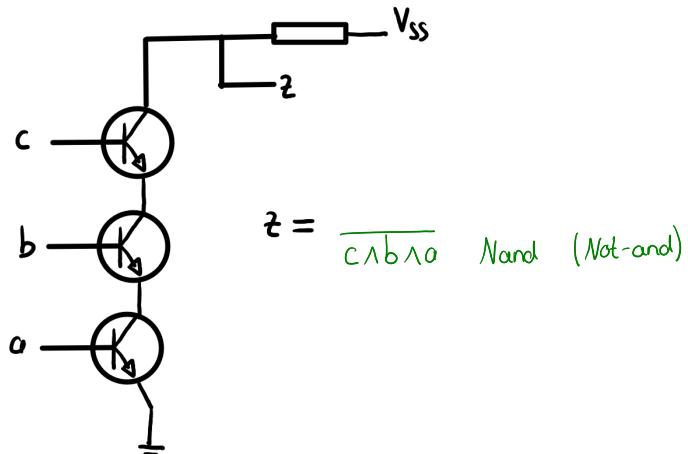
Geben Sie für dieses KV-Diagramm einen boolschen Ausdruck an, der ohne Hazard ist.

	$x_1$	$\bar{x}_1$	$\hat{x}_1$	$\tilde{x}_1$	
$x_2$	1		1	1	$x_4$
$\bar{x}_2$		1	1		$x_4$
$\tilde{x}_2$		1	1		$\bar{x}_4$
$\hat{x}_2$					$\bar{x}_4$
	$x_3$	$x_3$	$\bar{x}_3$	$\bar{x}_3$	

$$(x_1 \wedge x_4 \wedge x_2) \vee (\bar{x}_3 \wedge x_4 \wedge x_2) \vee (\bar{x}_1 \wedge x_4 \wedge \bar{x}_3) \vee (\bar{x}_1 \wedge \bar{x}_2)$$

## 5. Transistor

Dies ist eine Schaltung aus drei bipolaren NPN-Transistoren. Schreiben Sie  $z$  als boolsche Funktion der Eingaben  $a$ ,  $b$  und  $c$



Schaltung

Boolsche Funktion:

## 6. Binomische Formel

Im Folgenden nehmen wir für die Daten Fließkommazahlen einfacher Genauigkeit an (32 Bit = 4 Byte). Wir betrachten die binomische Formel für Vektoren:

$$(\vec{a} + \vec{b}) \cdot (\vec{a} + \vec{b}) = \vec{a} \cdot \vec{a} + 2 \vec{a} \cdot \vec{b} + \vec{b} \cdot \vec{b}$$

Welche Intensität hat der Ausdruck (1)

$$(\vec{a} + \vec{b}) \cdot (\vec{a} + \vec{b}) = \frac{3}{2 \cdot 4} = \frac{3}{8}$$

3 Rechenoperationen  
→  $\vec{a}$  und  $\vec{b}$  lacken

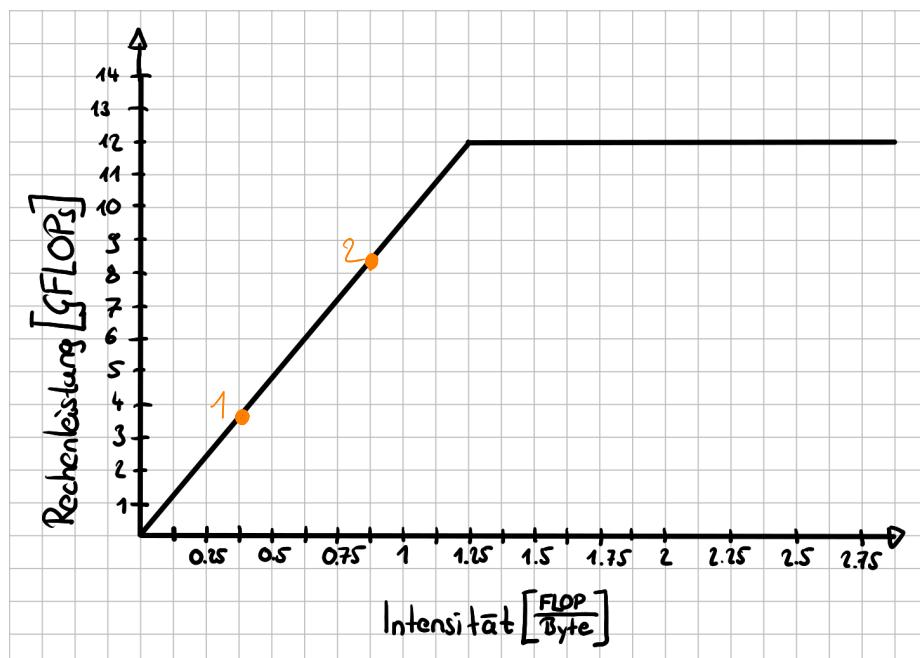
Welche Intensität hat der Ausdruck (2)

$$\vec{a} \cdot \vec{a} + 2 \vec{a} \cdot \vec{b} + \vec{b} \cdot \vec{b} = \frac{7}{2 \cdot 4} = \frac{7}{8}$$

Ganz wichtig, bei Ausdruck 2 kommt sicher  $\frac{7}{8}$  raus, auch wenn nur 6 Rechenoperationen zu sehen sind.  
Keiner hat damals verstanden warum?

Zeichnen Sie beide Ausdrücke in das Roofline-Diagramm ein, wobei Sie annehmen, dass beide nur durch die Speicherbandbreite limitiert sind und ansonsten optimal implementiert wurden.

Erwarten Sie, dass sich die Laufzeit (also die Zeit in Sekunden, bis der Ausdruck berechnet ist) der beiden Ausdrücke unterscheidet? Wenn ja, wie? Wenn nein, warum nicht?



Nein, da beide von der Bandbreite begrenzt sind. Heißt, beide sind nicht von Rechenoperationen, sondern pro Byte/Sekunde beschränkt. Also werden beide in der gleichen Anzahl an Sekunden ermittelt, da diese jeweils mehr GFLOPS/Sekunde berechnen.

Roofline-Modell

## 7. UART

Code A ist der PIO-Code einer UART zum Senden. Skizzieren Sie, was Sie am Pin 0 messen würden. Beschriften Sie dabei welche Bits Daten, Start und Stop-Bit sind. Funktioniert Code B auch? Bitte Begründung falls nicht.

```
A
import rp2
from machine import Pin

@rp2.asm_pio(out_init=
(rp2.PIO.OUT_HIGH, rp2.PIO.OUT_HIGH))
def senden():
    set(x, 7)
    pull()
    set(pins, 0) [9] # Startbit
    label("mal")
    out(pins, 1) [8]
    jmp(x_dec, "mal")
    set(pins, 1) [9] # Stopbit

p = Pin(0, Pin.OUT)
p.high()

sm = rp2.StateMachine(
    0, senden, out_base=p,
    set_base=p, freq=2000,
    out_shiftdir=rp2.PIO.SHIFT_RIGHT
)

sm.active(1)
sm.put(0xa3) = 1010 0011
```

```
B
import rp2
from machine import Pin

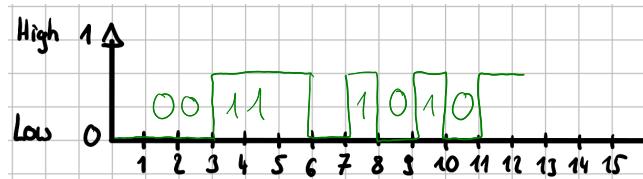
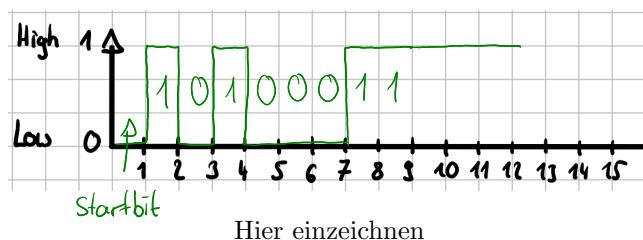
@rp2.asm_pio(out_init=
(rp2.PIO.OUT_HIGH, rp2.PIO.OUT_HIGH))
def senden():
    set(x, 3)
    pull(block)
    set(pins, 0) [9] # Startbit
    label("mal")
    out(pins, 1) [8]
    jmp(x_dec, "mal")
    set(pins, 1) [9] # Stopbit

p = Pin(0, Pin.OUT)
p.high()

sm = rp2.StateMachine(
    0, senden, out_base=p,
    set_base=p, freq=2000,
    out_shiftdir=rp2.PIO.SHIFT_RIGHT
)

sm.active(1)
sm.put(0x3)=0011
sm.put(0xa)=1010
```

Ganz wichtig!  
Hex zu decimal



Hier einzeichnen, falls Code B möglich

Begründung falls nicht:

## 8. Fließkomma

Tabelle: Dezimalzahlen und Bit-Pattern

Dezimal	Bit-Pattern (S E M)
1.00	0 01111 0000000000
2.00	0 10000 0000000000
3.00	0 10000 1000000000
0.50	0 01110 0000000000
-0.25	1 01101 0000000000

In der Tabelle oben sind die Dezimalzahlen und die Bit-Patterns für IEEE-754 Fließkommazahlen mit 16 Bit angegeben.

Die allgemeine Formel mit Vorzeichen  $S$ , Mantisse  $M$ , Breite  $w$ , Exponent  $E$  und Bias  $B$  lautet:

$$(-1)^S \cdot \left(1 + \frac{M}{2^w}\right) \cdot 2^{E-B}$$

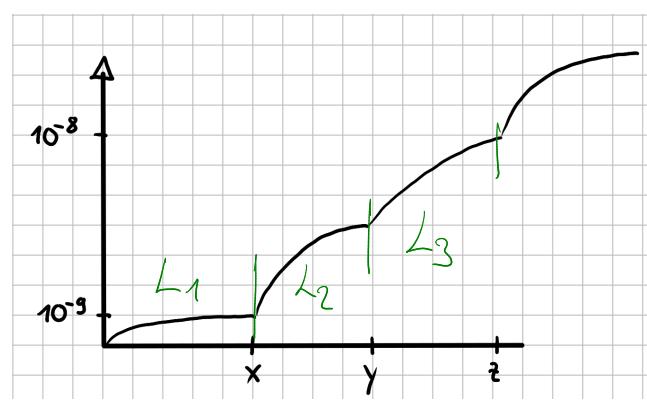
### Fragen

Mit dieser Webseite  
<https://www.h-schmidt.net/FloatConverter/IEEE754de.html> kann man das gut testen

1. Wie ist das Bit-Pattern für 2.75?  1000 0110000000000000
2. Wie ist das Bit-Pattern für 5.0?  1001 0100000000000000
3. Welchen Wert hat die Breite  $w$ ? 10
4. Welchen Wert hat der Bias  $B$ ? 15

## 9. Cache

Bestimmen Sie die Größe der Caches und klassifizieren Sie jeden Cache. Beschreiben Sie außerdem, wie Sie zu Ihrer Lösung gekommen sind.



Tragen Sie hier ein, falls Code 2 unterstützt wird.

Herausgefunden durch  
die Sprünge

$$L_1 = x$$

$$L_2 = y - x$$

$$L_3 = z - y$$

Das ist jetzt vereinfacht  
dargestellt, Achsbeschriftung  
beachten

## 10. PIO Code

Gegeben ist die Schaltung und der Python-Code für MicroPython. Skizzieren Sie grob, nach welchem Muster die LEDs blitzen.

```

from machine import Pin
import rp2
import time

@rp2.asm_pio(set_init=
(rp2 PIO.OUT_LOW, rp2 PIO.OUT_LOW))
def foo():
    set(pins, 0) beide aus
    set(x, 31)
    label("bar")
    nop()
    jmp(x_dec, "bar")

    set(pins, 3) beide an
    set(x, 7)
    label("foo")
    nop()
    jmp(x_dec, "foo")

    set(pins, 1) GPIO15 an
    set(x, 15)
    label("baz")
    nop()
    jmp(x_dec, "baz")

r = Pin(15, Pin.OUT)
s = Pin(16, Pin.OUT)

sm = rp2.StateMachine(
    0, foo, freq=2000, set_base=r)
sm.active(1)

```

