

# TI II

Sommersemester 2010

PD Dr. Katinka Wolter

---

## 3. Aufgabenblatt

Ausgabe Abgabe

14.05.10 28.05.10

Kontakt bei Fragen

Matthias Dräger, Viet Do, Marco Jeschke, Uwe Kuehn, Markus Rudolph

{mdraeger/do/kuehn/mjeschke/rudolph}@mi.fu-berlin.de

### Problem 1: Gleitkomma-Darstellung (2+2+2+2=8)

Stellen Sie die Zahlen  $2/3$  und  $-2/7$  als IEEE single unter Verwendung der

- „round-to-even“-Regel
- Rundung zum nächsten Gleitkommawert in Richtung 0
- Rundung zum nächsten Gleitkommawert in Richtung  $\infty$
- Rundung zum nächsten Gleitkommawert in Richtung  $-\infty$

dar.

### Problem 2: Carry-lookahead-Addierer (2+2+2=6)

- Lösen Sie die rekursive Berechnung von  $ü_4$  dem Skript entsprechend in Und- und Oder-Verknüpfungen von  $g_i$  und  $p_i$  mit  $i \leq 4$  auf.
- Berechnen Sie  $g_i$  und  $p_i$  mit  $i \leq 4$  für  $a_{4..1} = 0011$  und  $b_{4..1} = 1101$ .
- Berechnen Sie die ersten vier Ziffern  $s_i$  mit  $i \leq 4$  der Summe von a und b aus b) entsprechend der Carry-lookahead-Addierer-Methode.

### Problem 3: Windmessung mittels PAL (4+4=8)

- Konstruiere durch Programmierung eines PALs einen Decoder für die in Abbildung (a) dargestellte 10-Segment-Anzeige.  
Die Tabelle gibt die möglichen Eingangssignale und die entsprechende Anzeige an. Benutze jeweils den Ausgang  $s_i$  des PALs zur Ansteuerung des Segmentes  $i$ .  
Gib neben der Herleitung der Lösung eine Darstellung der programmierten Matrix auf kariertem Papier an.
- Konstruiere die Anzeigeeinheit eines digitalen Windrichtungsmessers (vgl. Abbildung (b)). Die Signale der an der Windfahne befestigten Sensoren sollten durch eine 2-stellige 10-Segment-Anzeige sichtbar gemacht werden. Gehe davon aus, daß zwei Decoderbausteine der in Aufgabenteil (a) spezifizierten Funktionalität zur Verfügung stehen. Die Signale  $x_0$ ,  $x_1$  und  $x_2$  der Sensoren sollen durch ein programmiertes PAL in Eingangssignale für die Decoderbausteine übersetzt werden. Nimm an, daß ein Sensor eine 1 ausgibt, falls er sich über einem weißen Feld befindet (analog schwarz = 0). Benutze ein PAL des in Aufgabenteil (a) verwendeten Typs und gib dessen Programmierung an (Herleitung, Angabe der Matrix auf kariertem Papier). Wähle für die Darstellung der Richtungen N, W, S und O jeweils die rechte 10-Segment-Anzeige.

# TI II

Sommersemester 2010  
PD Dr. Katinka Wolter

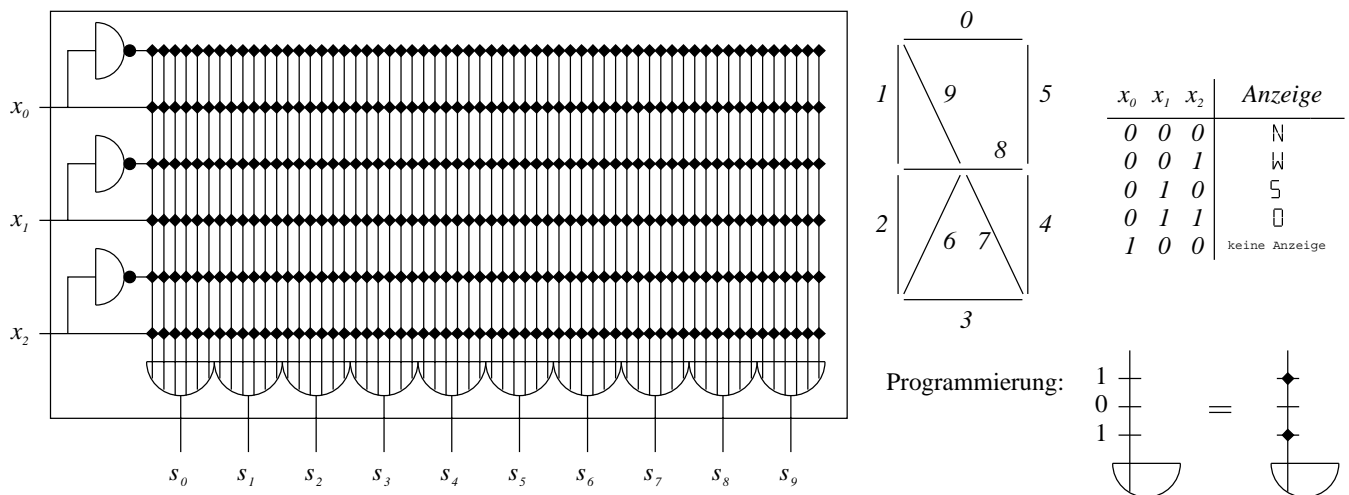


Abbildung (a)

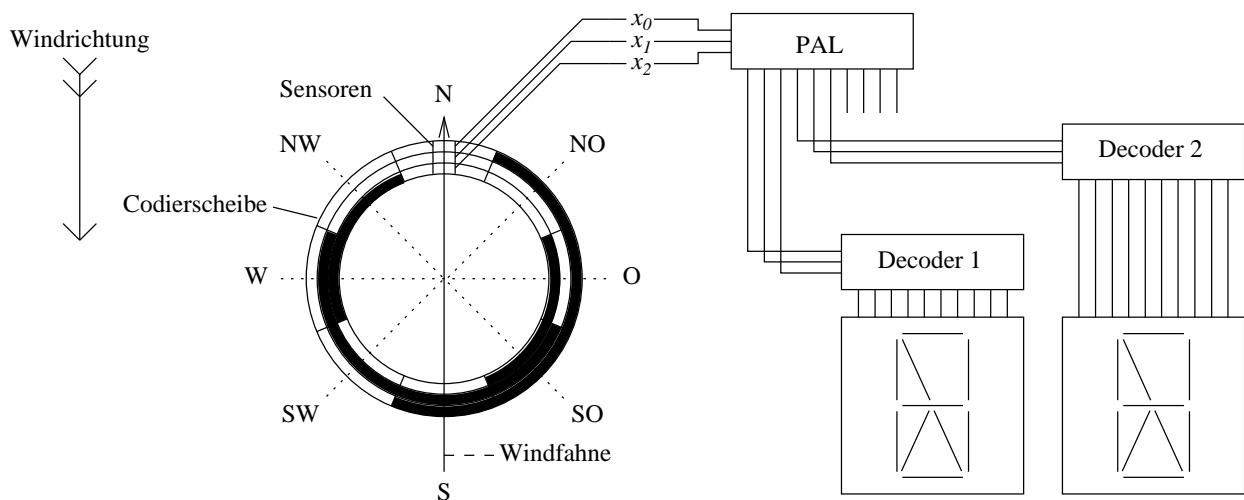


Abbildung (b)

## Problem 4: Halbaddierer und Volladdierer (3+3+4=10)

Ein NOR-Gatter ist ein Schaltnetz mit zwei Eingängen und einem Ausgang. Am Ausgang liegt genau dann eine 1 vor, wenn an keinem Eingang eine 1 anliegt.

- Entwerfen Sie eine Schaltung für einen Halbaddierer (Resultat und Übertrag), der ausschließlich aus NOR-Gattern mit 2 Eingängen besteht (Herleitung!). Verwenden Sie nicht mehr als 5 NOR-Gatter.
- Wie sieht der aus NOR-Gattern bestehende Volladdierer aus (für die Halbaddierer jeweils ein Symbol verwenden)?

# TI II

Sommersemester 2010

PD Dr. Katinka Wolter

---

- c) Wir nehmen nun an, daß ein NOR-Gatter eine Schaltzeit von 20 psec hat. Wann liegt bei Ihrem Halbaddierer das Resultat vor? Wann der Übertrag? Wie sehen die Zeiten für Ihren Volladdierer aus?

## Problem 5: Assembler - Quadratwurzel (6)

Schreiben Sie ein MMIX-Assemblerprogramm, das die folgende Aufgabenstellung löst:

Das Programm soll den ganzzahligen Anteil  $x$  der Quadratwurzel einer gegebenen ganzen Zahl  $z$  berechnen, d.h. gesucht wir  $x$  mit  $x = \max\{x \in \mathbb{Z} \mid x^2 \leq z\}$ , bzw.

$$\text{result} = \lfloor \sqrt{\text{value}} \rfloor .$$

Verwenden Sie dabei die sich aus der Newton-Methode ergebene Intervallschachtelung:

```
a := value
b := 0
while |a - b| > 1
    a := [(a + b)/2]
    b := [value/a]
end_while
result := min{a, b}
```

## Problem 6: MMIX – Unterprogramme (5+5=10)

Informieren Sie sich über die Erstellung von Unterprogrammen in MMIX.

- a) Implementieren Sie ein Unterprogramm für die Berechnung der Fakultätsfunktion  $n! = 1 \cdot 2 \cdot \dots \cdot n$  und geben Sie ein Hauptprogramm an, vom dem das Unterprogramm aufgerufen wird.
- b) Die rekursive Berechnung des Binomialkoeffizienten  $\binom{n}{k}$  kann mittels folgender Funktion durchgeführt werden:

```
function binom (n,k : integer) : integer;
begin
    if (k=0 OR k=n) then binom := 1
        else if k=1 then binom := n
            else binom := binom(n-1,k) + binom(n-1,k-1)
        end;
end;
```

Schreiben Sie ein vollständiges MMIX Assemblerprogramm, welches mittels eines rekursiven Unterprogramms den Binomialkoeffizient  $\binom{n}{k}$  berechnet.

Hinweis: Benutzen Sie PUSHJ und POP.