

# Online-Klausur

---

Name: ..... Matrikelnummer: .....

Studiengang: ..... Unterschrift: .....

Bearbeitungszeit: 90 Minuten

Gesamtpunktzahl: 60 Punkte

1. Verwenden Sie keinen Bleistift oder Rotstift!
  2. Unleserliche Angaben werden nicht gewertet!
  3. Verwenden Sie einen Notizbogen für Nebenrechnungen!
  4. Geben Sie den Rechenweg klar strukturiert und leserlich an!
  5. Lösungen ohne Angabe des Rechenwegs werden nicht gewertet!
  6. Nur Lösungen in den Lösungsfeldern werden gewertet!
- Nutzen Sie ggf. die Lösungsboxen der englischen Version!

# Online-Examination

---

Name: ..... Student number: .....

Branch of studies: ..... Signature: .....

Working time: 90 Minutes

Total points: 60 Points

1. Do not use lead pen or red ink!
2. Unreadable sections are not assessed!
3. Use an additional sheets for side calculatipons!
4. Give your solutions clearly structured and readable!
5. Solutions without calculation path are not assessed!
6. Solutions will be assessed only if given in the text boxes!  
Use the text boxes of the german version if required!

## Selbstständigkeitserklärung

### Persönliche Angaben

Name: \_\_\_\_\_  
(Last name)

Vorname: \_\_\_\_\_  
(First name)

Matrikelnummer: \_\_\_\_\_  
(Student-ID)

Studiengang: \_\_\_\_\_  
(Program)

### Angaben zur Prüfung

Name der Prüfung: \_\_\_\_\_  
(Title of the exam)

Prüfer: \_\_\_\_\_  
(Examiner)

Prüfungsdatum: \_\_\_\_\_  
(Exam date)

Hiermit versichere ich, dass ich die oben bezeichnete Leistung selbstständig und ohne unzulässige fremde Hilfe sowie ohne Heranziehung nicht zugelassener Hilfsmittel bearbeitet habe. Mir ist bewusst, dass der Verstoß gegen prüfungsrechtliche Regelungen über die Täuschung bei der Erbringung von Prüfungsleistungen und die Abgabe einer unrichtigen Versicherung geahndet wird.

I declare that I have worked on the above-mentioned assessment independently and without unauthorized assistance. I also confirm that I have not used any non-permissible resources. I am aware that the violation of examination regulations on cheating during examinations or a false declaration is punished.

Ort, Datum: \_\_\_\_\_  
(Place, date)

Unterschrift: \_\_\_\_\_  
(Signature)

**Aufgabe 1: Digitale Arithmetik (20 Punkte)**

1.1 Führen Sie folgende Rechnung im Einerkomplement Code durch.

- a) Nutzen Sie die Eigenschaften des Codes. Rechnen Sie ausführlich! (4 Punkte)  
b) Beurteilen Sie die Korrektheit des Ergebnisses. (4 Punkte)

$$01011,101 - 10100,111$$

	01011,101	
+	01011,000	invertierter Wert [1P]
	010110,000	Übertrag [0.5P]
	010110,101	Ergebnis [2P] mit <b>Überlauf</b> [0.5P]

**Beurteilung des Ergebnisses:** Bei der Addition zweier positiver Zahlen ( $11,625 + 11$ ) kann es zu einer Überschreitung des Zahlenbereichs kommen ( $+(2^{5-1} - 2^{-3}) = +15.875$  bis  $-(2^{5-1} - 2^{-3}) = -15.875$ ) [1P]. Hier werden zwei positive Werte (führende Null) addiert und das Ergebnis ist negativ. Damit kann das Ergebnis (22.625) nur inkorrekt sein [1P]. Der maximal positiv darstellbare Wert ist nicht groß genug, um das Ergebnis korrekt darzustellen [1P]. Das Ergebnis ist mit fünf Vorkomma- und drei Nachkommastellen nicht darstellbar und die Rechnung liefert das falsche Ergebnis [1P].

1.2 Führen Sie eine Vorzeichenerweiterung der Werte A und B auf acht Vorkomma und acht Nachkommastellen im

- a) Binärcode durch. (2 Punkte)  
b) Einerkomplement Code durch. (2 Punkte)

$$A=10100,111 \quad B=01011,101$$

a. Vorzeichenerweiterung der Werte im Binärcode:

$$A = \underline{000}10100,111\underline{00000} \quad [1P]$$

$$B = \underline{0000}1011,101\underline{00000} \quad [1P]$$

b. Vorzeichenerweiterung der Werte im Einerkomplement:

$$A = \underline{111}10100,111\underline{11111} \quad [1P]$$

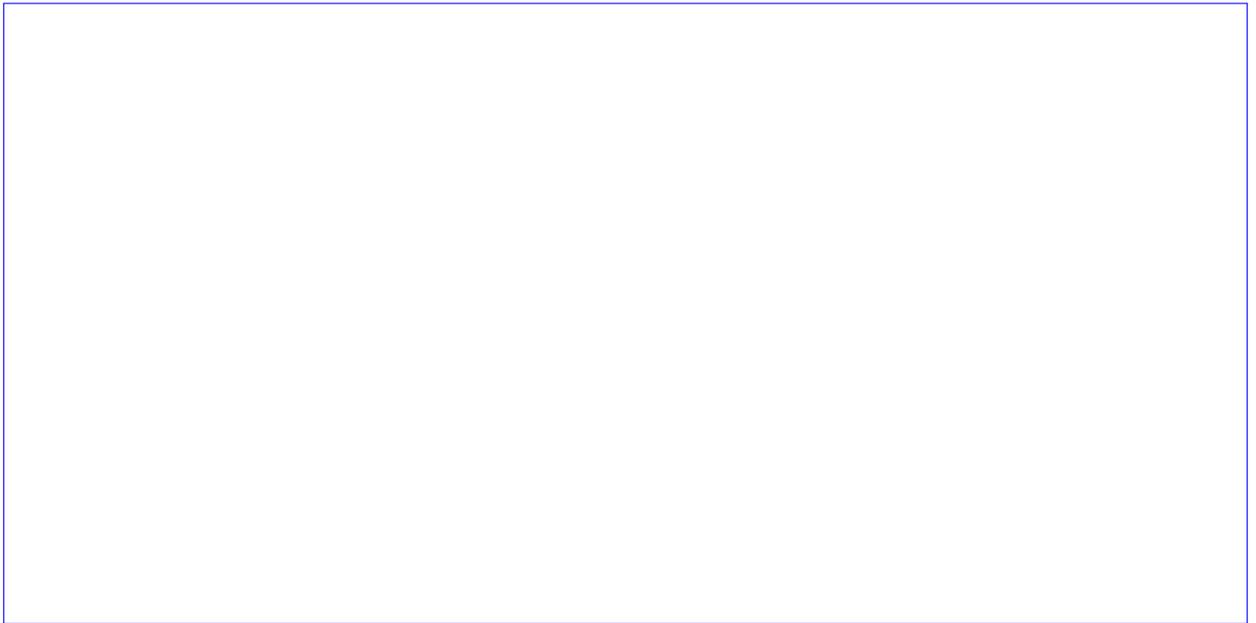
$$B = \underline{0000}1011,101\underline{00000} \quad [1P]$$

**Exercise 1: Digital Arithmetics****(20 Credits)**

1.1 Perform the following calculation with One's Complement code.

- a) Use the characteristics of the code. Show all details! **(4 Points)**  
b) Assess the correctness of the result. **(4 Points)**

$$01011,101 - 10100,111$$

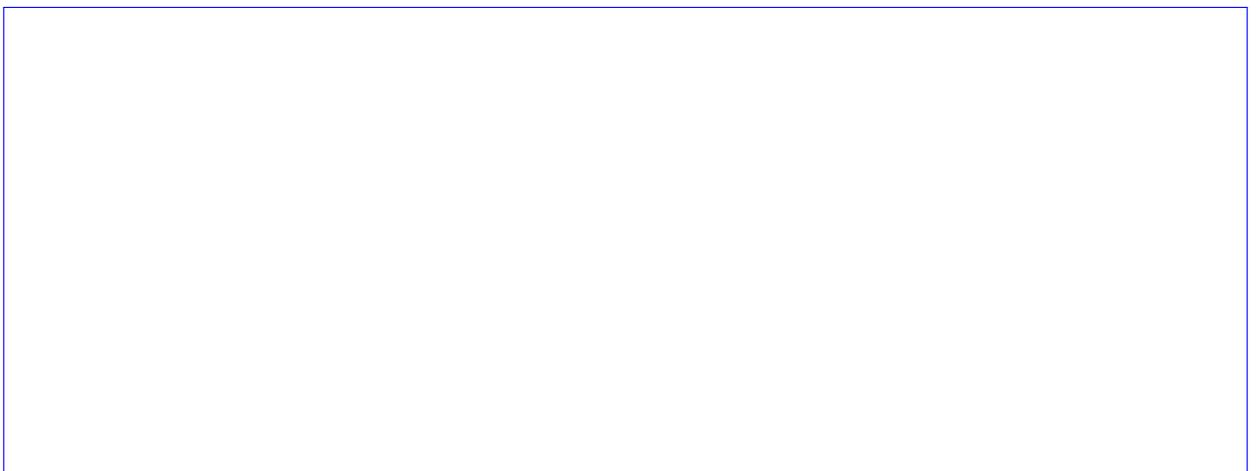


1.2 Perform a sign extension of the values A and B to eight integer and eight fraction positions for values given in

- a) binary code. **(2 Points)**  
b) One's complement code. **(2 Points)**

$$A=10100,111$$

$$B=01011,101$$



- 1.3 Transformieren Sie mit die Zahl  $\boxed{38,625}$  mit dem Euklidischen Algorithmus vom Dezimalsystem in das Binärsystem. (8 Punkte)  
Rechnen Sie ausführlich!

Anzahl Vorkommastellen  $n$  bestimmen:

$$n = \lfloor \log_2 38 \rfloor = \lfloor \frac{\log_{10} 38}{\log_{10} 2} \rfloor = \lfloor 5.25 \rfloor = 5$$

Es sind 5 Vorkommastellen nötig, um die Zahl darzustellen. [2P]

Bestimmung der Stellenwerte nach dem Euklidischen Verfahren.

$$38.625 \text{ div } 2^5 = 1 \text{ Rest } 6.625 \quad [0.5P]$$

$$6.625 \text{ div } 2^4 = 0 \text{ Rest } 6.625 \quad [0.5P]$$

$$6.625 \text{ div } 2^3 = 0 \text{ Rest } 6.625 \quad [0.5P]$$

$$6.625 \text{ div } 2^2 = 1 \text{ Rest } 6.625 \quad [0.5P]$$

$$2.625 \text{ div } 2^1 = 1 \text{ Rest } 0.625 \quad [0.5P]$$

$$0.625 \text{ div } 2^0 = 0 \text{ Rest } 0.625 \quad [0.5P]$$

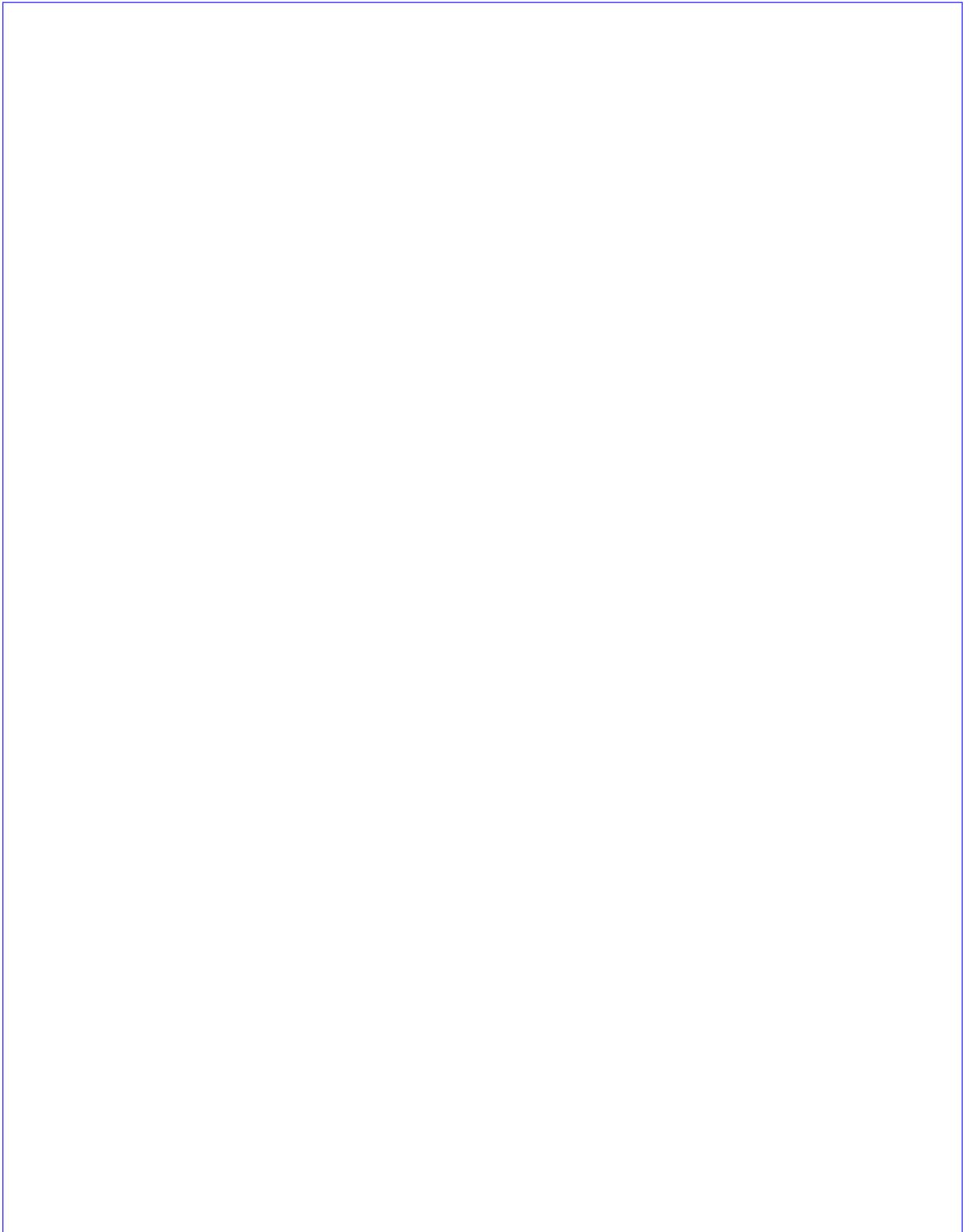
$$0.625 \text{ div } 2^{-1} = 1 \text{ Rest } 0.125 \quad [0.5P]$$

$$0.125 \text{ div } 2^{-2} = 0 \text{ Rest } 0.125 \quad [0.5P]$$

$$0.125 \text{ div } 2^{-3} = 1 \text{ Rest } \mathbf{0} \quad [0.5P] \quad (\text{Algorithmus terminiert!}) [0.5P]$$

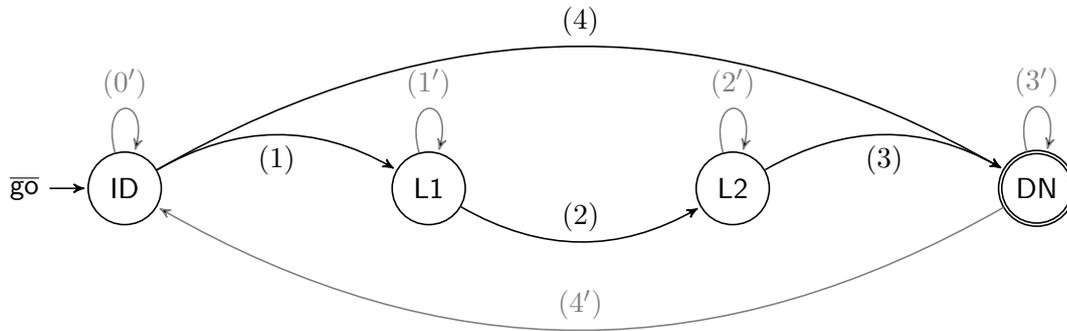
Ergebnis:  $38.625_{10} = 100110.101_2$  [1P]

- 1.3 Transform the number  $38,625$  with the Euclidian algorithm from decimal system to binary system. (8 Points)  
Show all details of the transformation!



**Aufgabe 2: Deterministische Automaten (20 Punkte)**

Gegeben sei ein deterministischer endlicher Automat für einen vereinfachten zweistufigen Page-Table-Walk einer Memory-Management Einheit.



$$\begin{aligned}
 (0') &= (\overline{\text{pt1\_done}} \vee \overline{\text{pt2\_done}}) \wedge \overline{\text{tlb\_hit}} \\
 (1) &= \overline{\text{pt1\_done}} \wedge \overline{\text{pt2\_done}} \wedge \overline{\text{tlb\_hit}} & (1') &= \overline{\text{pt1\_done}} \\
 (2) &= \text{pt1\_done} & (2') &= \overline{\text{pt2\_done}} \\
 (3) &= \text{pt2\_done} & (3') &= \text{pt1\_done} \wedge \text{pt2\_done} \\
 (4) &= \text{tlb\_hit} & (4') &= \overline{\text{pt1\_done}} \vee \overline{\text{pt2\_done}}
 \end{aligned}$$

1. Erstellen Sie die Zustandsübergangstabelle für den Automaten. (8 Punkte)  
Verwenden Sie Don't-Care Variablen, um die Tabelle klein zu halten!

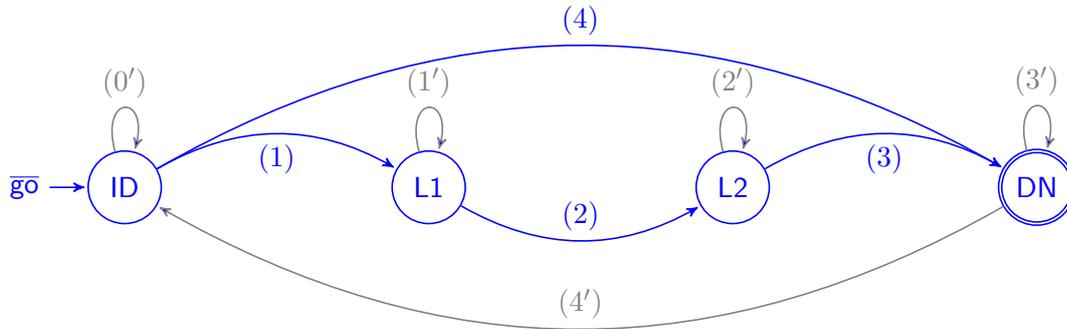
go	pt1_done	pt2_done	tlb_hit	$S_t$	$S_{t+1}$
0	x	x	x	x	ID
1	1	x	0	ID	ID
1	x	1	0	ID	ID
1	0	0	0	ID	L1
1	x	x	1	ID	DN
1	0	x	x	L1	L1
1	1	x	x	L1	L2
1	x	0	x	L2	L2
1	x	1	x	L2	DN
1	1	1	x	DN	DN
1	0	x	x	DN	ID
1	x	0	x	DN	ID

[Punktevergabe: 1P f. Resetbedingungen, 2P f. Transitionen aus ID, je 1P f. Transitionen aus L1 und L2, 2P f. Transitionen aus DN, 1P für korrekte und ordentliche Darstellung der Tabelle]

**Exercise 2: Deterministic Automata**

**(20 Punkte)**

Given a deterministic finite automata for a simplified two-stage page table-walk engine of a memory-management unit.



	$(0') = (\overline{pt1\_done} \vee \overline{pt2\_done}) \wedge \overline{tlb\_hit}$
$(1) = \overline{pt1\_done} \wedge \overline{pt2\_done} \wedge \overline{tlb\_hit}$	$(1') = \overline{pt1\_done}$
$(2) = pt1\_done$	$(2') = pt2\_done$
$(3) = pt2\_done$	$(3') = pt1\_done \wedge pt2\_done$
$(4) = tlb\_hit$	$(4') = \overline{pt1\_done} \vee \overline{pt2\_done}$

1. Create the state-transfer table for the automata.

**(8 Points)**

Use don't care variables to keep the table small!

2. Erstellen Sie eine Zustandskodierungstabelle für den Automaten. **(2 Punkte)**  
Verwenden Sie eine Gray-Codierung, die im Reset-Fall nicht einzuhalten ist!

Zustandsname	Zustandskodierung $S(1 : 0)$
ID	00
L1	01
L2	11
DN	10

[Punktevergabe: 0.5P f. 2 Stellen im Zustandsvektor und eindeutige Zuweisung von Zustandscodes, 1P für Gray-Codierung, 0.5P für korrekte und ordentliche Darstellung der Tabelle]

3. Erstellen Sie die Zustandsübergangsfunktion in disjunktiver Normalform. **(10 Punkte)**

Einsetzen der Zustandskodierung in die Zustandsübergangstabelle ergibt die Wahrheitstabelle der Zustandsübergangsfunktion.

go	pt1_done	pt2_done	tlb_hit	$S_t$	$S_{t+1}$
0	x	x	x	x	00
1	1	x	0	00	00
1	x	1	0	00	00
1	0	0	0	00	01
1	x	x	1	00	10
1	0	x	x	01	01
1	1	x	x	01	11
1	x	0	x	11	11
1	x	1	x	11	10
1	1	1	x	10	10
1	0	x	x	10	00
1	x	0	x	10	00

$$S_{t+1}(1) = (go \wedge tlb\_hit \wedge \overline{S_t(1)} \wedge \overline{S_t(0)}) \vee (go \wedge \overline{pte1\_done} \wedge \overline{S_t(1)} \wedge \overline{S_t(0)}) \vee (go \wedge \overline{pte2\_done} \wedge \overline{S_t(1)} \wedge \overline{S_t(0)}) \vee (go \wedge pte2\_done \wedge \overline{S_t(1)} \wedge \overline{S_t(0)}) \vee (go \wedge pte1\_done \wedge pte2\_done \wedge \overline{S_t(1)} \wedge \overline{S_t(0)})$$

$$S_{t+1}(0) = (go \wedge \overline{pte1\_done} \wedge \overline{pte2\_done} \wedge \overline{tlb\_hit} \wedge \overline{S_t(1)} \wedge \overline{S_t(0)}) \vee (go \wedge \overline{pte1\_done} \wedge \overline{S_t(1)} \wedge \overline{S_t(0)}) \vee (go \wedge pte1\_done \wedge \overline{S_t(1)} \wedge \overline{S_t(0)}) \vee (go \wedge \overline{pte2\_done} \wedge \overline{S_t(1)} \wedge \overline{S_t(0)})$$

[Punktevergabe: 4P f. korrekte und ordentliche Wahrheitstabelle, je 3P für eine korrekte Zustandsübergangsfunktion]

2. Create a state-encoding table for the automata. **(2 Points)**  
Use a gray encoding, not to be maintained under reset condition!



3. Derive the state-transfer function as a sum of products. **(10 Points)**



**Aufgabe 3: Digitaler Schaltungsentwurf (20 Punkte)**

Die folgende Wahrheitstabelle beschreibt die Ausgabefunktionen eines Zustandsautomaten.

S(1)	S(0)	idle	sel_pt1	sel_pt2	sel_idx
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

1. Bestimmen Sie die vier Ausgabefunktionen in Konjunktiver Form. (4 Punkte)  
Um welche Architektur handelt es sich bei dem Zustandsautomaten? (1 Punkt)

$$\begin{aligned}
 \text{idle} &= S(1) \vee S(0) && [1P] \\
 \text{sel\_pt1} &= S(1) \vee \overline{S(0)} && [1P] \\
 \text{sel\_pt2} &= \overline{S(1)} \vee S(0) && [1P] \\
 \text{sel\_idx} &= \overline{S(1)} \vee \overline{S(0)} && [1P]
 \end{aligned}$$

Der Automat ist als MOORE-Architektur implementiert. [1P]

2. Bestimmen Sie die Funktion für `idle` im  $(\overline{\wedge})$ -Operatorensystem. (2 Punkte)  
Geben Sie zu jedem Transformationsschritt das angewendete Rechengesetz der boole'schen Algebra an! (1 Punkt)

$$\begin{aligned}
 \text{idle} &= S(1) \vee S(0) \\
 &= \overline{\overline{S(1) \vee S(0)}} \quad (\text{Identität}) [0.5P] \\
 &= \overline{\overline{S(1)} \wedge \overline{S(0)}} \quad (\text{Regel von de Morgan}) [0.5P] \\
 &= \overline{S(1) \overline{\wedge} \overline{S(0)}} \quad (\text{Definition der NAND - Funktion}) [0.5P] \\
 &= (S(1) \overline{\wedge} S(1)) \overline{\wedge} (S(0) \overline{\wedge} S(0)) \quad (\text{Negation aus NAND - Funktion}) [0.5P]
 \end{aligned}$$

[+1P, falls alle Regeln richtig benannt wurden, 0.5P falls zwei Regeln richtig benannt wurden.]

### Exercise 3: Digital Circuit Design

(20 Punkte)

The following truth table determines the output functions of a state machine.

S(1)	S(0)	idle	sel_pt1	sel_pt2	sel_idx
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

1. Give the four output functions as product of sums.

(4 Points)

What type of architecture has the automata?

(1 Point)

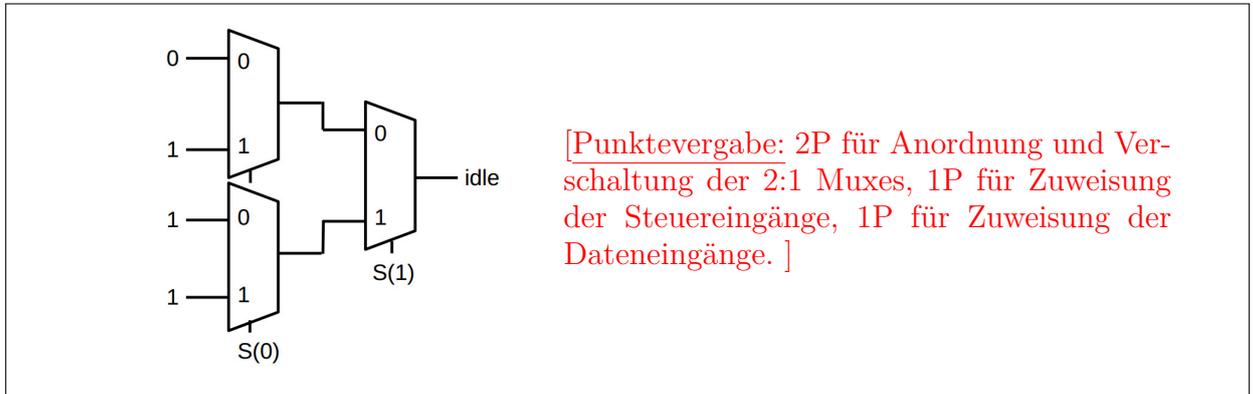
2. Give the function of `idle` in the  $(\overline{\wedge})$  operator system.

(2 Points)

Label every step of the transformation with the corresponding law of boolean algebra!

(2 Points)

3. Implementieren Sie die Funktion für die Ausgabe `idle` mit ausschließ- (4 Punkte)  
lich 2:1 Multiplexern. Zeichnen Sie die Schaltung mit 2:1 Multiplexern.  
Hinweis: Wenden Sie das in der Veranstaltung erläuterte Entwurfsprinzip an!



- 4a. Bestimmen Sie die Funktion für `idle` in Disjunktiver Normalform. (2 Punkte)

$$idle = (\overline{S(1)} \wedge S(0)) \vee (S(1) \wedge \overline{S(0)}) \vee (S(1) \wedge S(0))$$

[Punktevergabe: 0.5P für Zuweisung an `idle`, je 0.5P pro Implikant.]

- 4b. Optimieren Sie die Funktion mit dem Verfahren von Quine-McCluskey. (6 Punkte)

1. Quine'sche Tabellen. [3P]

0. Ordnung:

Num	S(1)	S(0)
0*	1	1
1*	0	1
2*	1	0

1. Ordnung:

Num	S(1)	S(0)
(0,1)	-	1
(0,2)	1	-

Keine weiteren Zusammenfassungen möglich.

2. Überdeckungstabelle erstellen und Kern-Primterme finden. [2P]

Primterm	0	1	2
(0,1)	x	x	
(0,2)	x		x

Beide Terme sind Kern-Primterme. Weitere Primterme gibt es nicht.

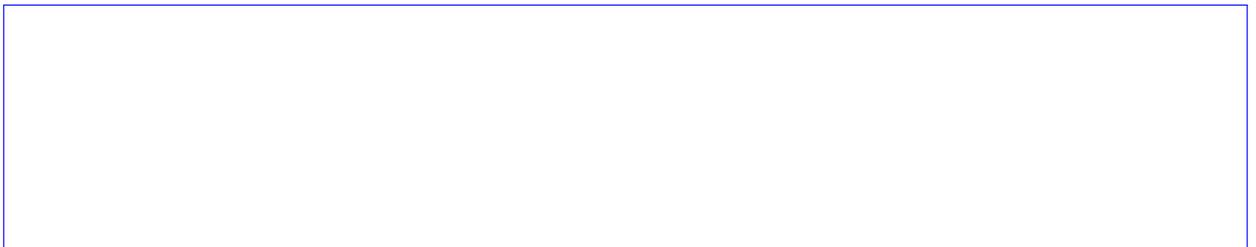
3. Ergebnis:  $idle = S(1) \vee S(0)$  [1P]

(Hinweis: Das entspricht in diesem Fall der konjunktiven Form, da jeder Implikant nur eine Variable enthält.)

3. Implement the function for the output `idle` exclusively with 2:1 multiplexers. Draw the block diagram with 2:1 multiplexers. (4 Points)  
Remark: Apply the design procedure explained in the course!



- 4a. Give the function for `idle` as sum of products. (2 Points)



- 4b. Optimize the function with the Quine-McCluskey algorithm. (6 Points)

