KIT-Fakultät für Informatik
Prof. Dr. Mehdi B. Tahoori, Prof. Dr.-Ing. Jörg Henkel

Lösungsblätter zur Klausur

Digitaltechnik und Entwurfsverfahren (TI-1)

und

Rechnerorganisation (TI-2)

am 18. August 2021, 8:00 – 10:00 Uhr

Name:	Vorname:	Matrikelnummer:

Digitaltechnik und Entwurfsverfahren (TI-1)	
Aufgabe 1	von 10 Punkten
Aufgabe 2	von 10 Punkten
Aufgabe 3	von 8 Punkten
Aufgabe 4	von 9 Punkten
Aufgabe 5	von 8 Punkten

Rechnerorganisation (TI-2)	
Aufgabe 6	von 7 Punkten
Aufgabe 7	von 10 Punkten
Aufgabe 8	von 13 Punkten
Aufgabe 9	von 7 Punkten
Aufgabe 10	von 8 Punkten

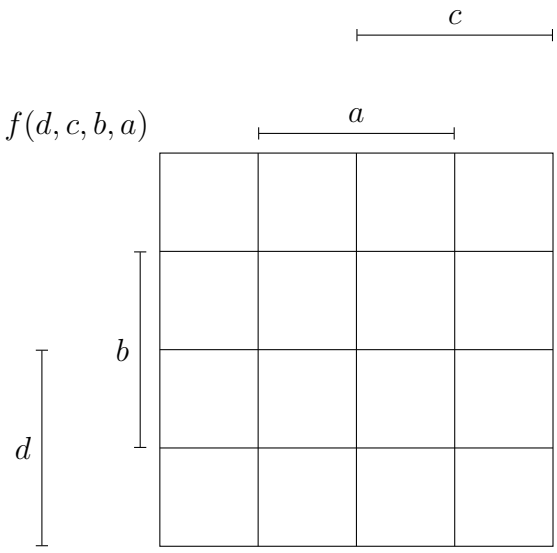
Gesamtpunktzahl:	

	Note:

Aufgabe 1 *Schaltfunktionen*

1. Konjunktive Normalform (KNF):

2.



Primimplikanten:

3. Disjunktive Minimalform von $f(d, c, b, a)$:

4. Zweistufige disjunktive Form von $g(c, b, a)$:

Aufgabe 2 *CMOS-Technologie*

1. Umgeformte Schaltfunktion und Transistor-Schaltbild:

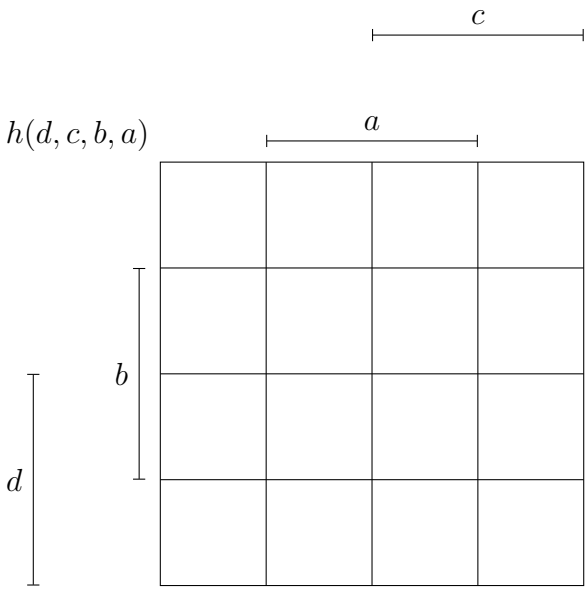
2. Aufbau eines pMOS-Transistors:

3. Unterschied zwischen n-Kanal- und einem p-Kanal-MOSFET:

Aufgabe 3 *Laufzeiteffekte*

1. Totzeitmodell:

2. KV-Diagramm:



3. Realisierung, die frei von allen statischen Strukturhasards ist:

Begründung:

Aufgabe 4 *Schaltwerke*

1. Automatentyp:

   Begründung:

2. Ansteuerfunktion:

   Zustandsübergangsgleichung:

   Ausgabefunktion:

3. Automatengraph des Schaltwerks:

4. Automatengraph mit minimaler Anzahl Zustände:

5. Zustandsübergangsgleichungen:

Aufgabe 5 *Rechnerarithmetik*

1. Die Basen s und r:

2. Der dezimale Wert der größten Zahl:

   -

   -

   -

3. Ausnahmeregel für die Null im IEEE-754-Standard:

4. Serieller Multiplizierer nach der PPS-Methode:

Aufgabe 6 *Die Programmiersprache C*

1. Implementierung `addTwo(int *array, int n)`:

```
int addTwo(int *array, int n)
{









}
```

2. Implementierung `calcSum(int *array, int n)`:

```
int calcSum(int *array, int n)
{









}
```

3. Implementierung `revArr(int *array, int n)`:

```
int revArr(int *array, int n)
{









}
```

Aufgabe 7 *MIPS-Assembler*

1. MIPS steht für:

2. Anzahl Bits für ein Befehlswort:

3. Unterschied Maschinensprache und Assemblersprache:

4. 2 niedrigstwertigen Bits:

5. Laden von `0xF03D 0909` ins Register `$s0`:

6. Inhalte der Zielregister:

Befehl	Zielregister = Wert (z.B. `$s6 = 0x0000 F00A`)
`ori $s1, $zero, 0x2021`	
`sll $s2, $s1, 1`	
`slti $s3, $s2, 0x4043`	
`sub $s4, $s3, $s2`	

Aufgabe 8 *Pipelining*

1. Datenabhängigkeiten:

   •

   •

   •

2. Belegung der Register nach Ablauf des Programms und Zustand der Pipeline:

Takt	`IF`	`DE`	`OF`	`EX`	`WB`	`$t0`	`$t1`	`$t2`
1	S1	—	—	—	—	3	6	8

Anzahl der Takte:

3. Belegung der Register bei sequentieller Bearbeitung des Programms:

$t0	$t1	$t2

4. Behebung der Pipelinekonflikte durch Einfügen von NOP-Befehlen:

Anzahl der Takte:

Aufgabe 9 *Speicherbausteine*

1. Problem:

2. Transistor-Schaltbild einer 1-Bit Speicherzelle eines statischen RAM-Bausteins (SRAM):

3. Zugriffszeit:

   Zykluszeit:

4. Magnetische Speicher in einer Speicherhierarchie:

Aufgabe 10 *Virtuelle Speicherverwaltung*

1. Unterteilung der virtuellen Adresse:

2. Physikalische Adressen:

Virtuelle		Physikalische	
Adresse	Seitennummer	Seitennummer	Adresse
512			
4095			
4097			
4198			
8191			
8192			
8400			
0			

3. Beschleunigung durch TLB:

4. Breite des *Tags*: