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

Aufgabenblätter zur Prüfung

Digitaltechnik und Entwurfsverfahren (TI-1)

und

Rechnerorganisation (TI-2)

am 18. August 2021, 8:00 – 10:00 Uhr

- Beschriften Sie bitte gleich zu Beginn jedes Lösungsblatt deutlich lesbar mit Ihrem Namen und Ihrer Matrikelnummer.
- Diese Aufgabenblätter werden nicht abgegeben. Tragen Sie Ihre Lösung deshalb ausschließlich in die für jede Aufgabe vorgesehenen Bereiche der Lösungsblätter ein. Lösungen auf separat abgegebenen Blättern werden nicht gewertet.
- Außer Schreibmaterial sind während der Klausur keine Hilfsmittel zugelassen. Täuschungsversuche durch Verwendung unzulässiger Hilfsmittel führen unmittelbar zum Ausschluss von der Klausur und zur Note „nicht bestanden“.
- Soweit in der Aufgabenstellung nichts anderes angegeben ist, tragen Sie in die Lösungsblätter bitte nur Endergebnisse und Rechenweg ein. Die Rückseiten der Aufgabenblätter können Sie als Konzeptpapier verwenden. Weiteres Konzeptpapier können Sie auf Anfrage während der Klausur erhalten.
- Halten Sie Begründungen oder Erklärungen so kurz und präzise wie möglich. Der auf den Lösungsblättern für eine Aufgabe vorgesehene Platz lässt nicht auf den Umfang einer korrekten Lösung schließen.
- Die Gesamtpunktzahl beträgt 90 Punkte. Zum Bestehen der Klausur sind mindestens 40 Punkte zu erreichen.

Viel Erfolg und viel Glück!

Aufgabe 1 *Schaltfunktionen* (10 Punkte)

Gegeben sei die vollständige Schaltfunktion $f(d, c, b, a)$ durch die folgende Wahrheitstabelle (Tabelle 1):

d	c	b	a	$f(d, c, b, a)$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Tabelle 1: Wahrheitstabelle von $f(d, c, b, a)$

1. Geben Sie die konjunktive Normalform (KNF) von $f(d, c, b, a)$ an. 2 P.
2. Tragen Sie die Schaltfunktion $f(d, c, b, a)$ in das KV-Diagramm des Lösungsblattes ein. Zeichnen Sie alle Prim-Einsblöcke klar und eindeutig ein. Geben Sie die dazugehörigen Primimplikanten an. Unterstreichen Sie alle Kernprimimplikanten. 4 P.
3. Geben Sie eine disjunktive Minimalform (DMF) von $f(d, c, b, a)$ an. 2 P.
4. Das folgende Schaltnetz realisiert die Schaltfunktion $g(c, b, a)$ (Abbildung 1). Geben Sie g in zweistufiger disjunktiver Form an. Vereinfachen Sie soweit wie möglich. 2 P.

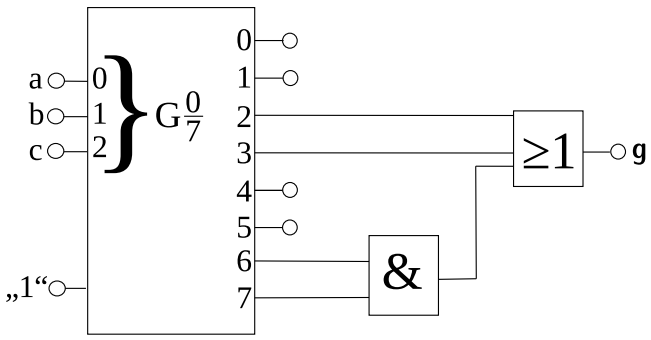


Abbildung 1: Schaltnetz von $g(c, b, a)$

Aufgabe 2 *CMOS-Technologie* (10 Punkte)

1. Die Schaltfunktion [5 P.]

$$y = k(c, b, a) = (a \vee \overline{b}) \wedge \overline{c}$$

soll in der CMOS-Technologie realisiert werden. Es stehen Ihnen zwei NOR-Gatter und ein Inverter-Gatter zur Verfügung. Formen Sie die Schaltfunktion entsprechend um und geben Sie das resultierende Transistor-Schaltbild an.

2. Skizzieren Sie den Aufbau eines pMOS-Transistors. Beschriften Sie alle Bestandteile. Aus Ihrer Zeichnung müssen die unterschiedlich dotierten Zonen und die Anschlüsse des Transistors eindeutig erkennbar sein. [3 P.]

3. Was ist der Unterschied zwischen einem n-Kanal- und einem p-Kanal-MOSFET? Warum werden pMOS-Transistoren in CMOS-Schaltungen so angeordnet, dass sie eine logische Eins an den Ausgang durchschalten, während nMOS-Transistoren eine logische Null durchschalten? [2 P.]

Aufgabe 3 *Laufzeiteffekte* (8 Punkte)

Eine Schaltfunktion $h(d, c, b, a)$ ist durch das Schaltnetz in Abbildung 2 realisiert.

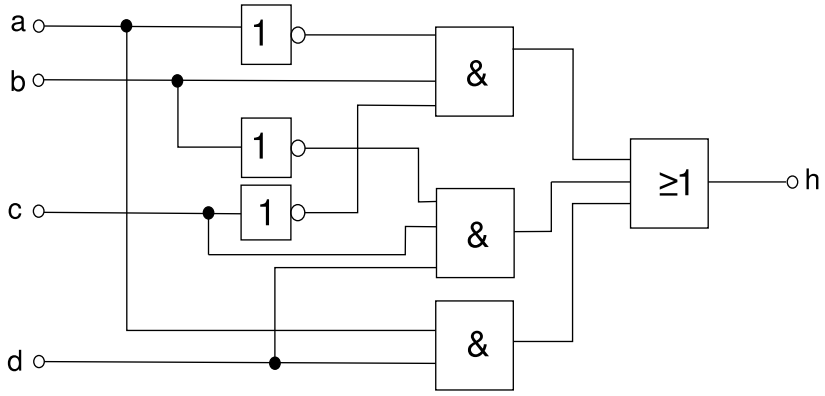


Abbildung 2: Schaltnetz von $h(d, c, b, a)$

1. Geben Sie das endgültige Totzeitmodell des Schaltnetzes an, indem Sie jedem Gatter seinen Verzögerungswert zuweisen und alle Totzeiten zum Eingang des Schaltnetzes verschieben. Tragen Sie alle Pfadvariablen in Ihrer Lösung ein und geben Sie die Werte der Pfadverzögerungen an. Alle verwendeten Gatter besitzen eine Totzeit von $10ns$. 4 P.

2. Übertragen Sie die Schaltfunktion $h(d, c, b, a)$ in das im Lösungsblatt vorbereitete KV-Diagramm und kennzeichnen Sie die Einsblöcke, die bei der Realisierung durch das Schaltnetz in Abbildung 2 verwendet wurden. 2 P.

3. Geben Sie eine Realisierung der Schaltfunktion an, die frei von allen statischen Strukturhazards ist. Begründen Sie Ihre Antwort. 2 P.
   Hinweis: Es reicht aus, wenn Sie den Booleschen Ausdruck einer derartigen Realisierung angeben.

Aufgabe 4 *Schaltwerke* (9 Punkte)

Gegeben sei das in Abbildung 3 dargestellte Schaltwerk mit der Eingangsvariablen x und der Ausgabevariablen y.

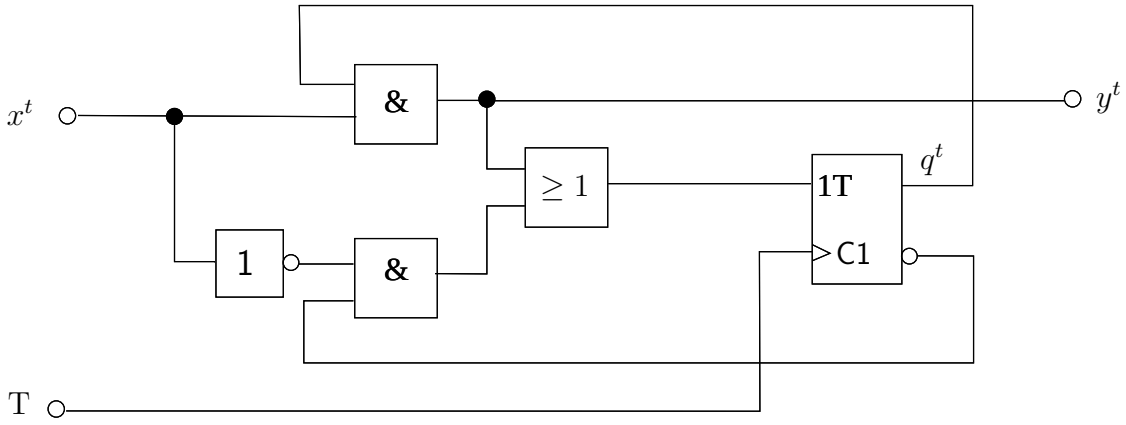


Abbildung 3: Schaltwerk

1. Um welchen Automatentyp handelt es sich? Begründen Sie Ihre Antwort. 1 P.
2. Bestimmen Sie die Ansteuerfunktion des T-Flipflops, die Zustandsübergangsgleichung und Ausgabefunktion. Vereinfachen Sie die Funktionen soweit wie möglich. 2 P.
3. Zeichnen Sie den Automatengraphen des Schaltwerks. 1 P.

Entwerfen Sie einen Moore-Zustandsautomaten mit einer **minimalen** Anzahl von Zuständen, welcher eine beliebig lange Dualzahl bitweise einliest (Variable e) und kontinuierlich das logische ODER der letzten zwei eingelesenen Werte bildet und ausgibt (Ausgabevariable a).

4. Zeichnen Sie den Automatengraphen. 3 P.
5. Die Zustände seien mit den Zustandsvariablen $q_0, q_1, \ldots$ dual kodiert. Bestimmen Sie die Zustandsübergangsgleichungen in disjunktiver Minimalform. 2 P.

Aufgabe 5 *Rechnerarithmetik* (8 Punkte)

1. Finden Sie die Basen r und s so dass gilt: 1 P.

$$14_r \;=\; 132_s$$

2. Geben Sie den dezimalen Wert der größten Zahl an, die repräsentiert werden kann mit 3 P.

   - 14 binären Stellen,
   - 6 oktalen Stellen,
   - 4 hexadezimalen Stellen.

   Geben Sie das Endergebnis *nicht* in exponentieller Schreibweise an.

3. Warum wird zur Darstellung der Zahl 0 als normalisierte Zahl im IEEE-754 Standard eine Ausnahmeregel benötigt? 1 P.

4. Die PPS-Methode (*partial product sum*) zur seriellen Multiplikation zweier n-Bit Zahlen kann mit Hilfe eines n-Bit Addierers, eines n-Bit Registers B, zweier n-Bit Schieberegister A und P und eines Flipflops E implementiert werden. 3 P.

   Aus der Vorlesung ist ein 4-Bit-Multiplizierer nach dieser Methode bekannt. Skizzieren Sie den Aufbau und den Datenfluss des Multiplizierers.

Aufgabe 6 *Die Programmiersprache C* (7 Punkte)

Gegeben seien die folgenden Codeschnipsel in C. Vervollständigen Sie auf dem Lösungsblatt die Funktionen so, dass die gewünschte Funktion realisiert wird. Das Ergebnis ist als Beispiel im Kommentar rechts der Funktion zu sehen. Achten Sie darauf, dass Ihr Code nicht nur für dieses Beispiel funktioniert. Die Verwendung von externen Bibliotheken sei nicht gestattet.

```
int main()
{
```

1. Implementieren Sie die Funktion `addTwo(int *array, int n)`. Sie addiert die Zahl 2 auf den ersten n Speicherstellen, auf die der Pointer `array` zeigt. [2 P.]

```
int a[] = {1,2,3,4,5};
addTwo(a, 5); // {3,4,5,6,7}
```

2. Implementieren Sie die Funktion `calcSum(int *array, int n)`. Sie summiert die ersten n Speicherstellen, auf die der Pointer `array` zeigt. Das Ergebnis wird an die erste Stelle gespeichert. [2 P.]

```
int b[] = {3,6,9,18,27};
calcSum(b,5); // {63,6,9,18,27}
```

3. Implementieren Sie die Funktion `revArr(int *array, int n)`. Sie invertiert die Reihenfolge der ersten n Speicherstellen, auf die der Pointer `array` zeigt. [3 P.]

```
int c[] = {1,2,3,4,5};
revArr(c, 5); // {5,4,3,2,1}
```

```
}
```

Aufgabe 7 *MIPS-Assembler* (10 Punkte)

1. Für was steht die Abkürzung MIPS? 1 P.
2. Wie viele Bits sieht das MIPS-Befehlsformat für ein Befehlswort vor? 1 P.
3. Was ist der Unterschied zwischen Maschinensprache und Assemblersprache? 2 P.
4. Welche Werte haben die 2 niedrigstwertigen Bits einer Wortadresse? 1 P.
5. Bei der MIPS-Architektur können nur 16-Bit Operanden durch MIPS-Befehle in die Register geladen werden. Geben Sie eine Folge von MIPS-Befehlen an, mit welcher der 32-Bit Operand `0xF03D 0909` ins Register `$s0` geladen werden kann. 2 P.
6. Geben Sie für das folgende MIPS-Programmstück den Inhalt des Zielregisters nach der Ausführung des jeweiligen Befehls in hexadezimaler Schreibweise an. 3 P.

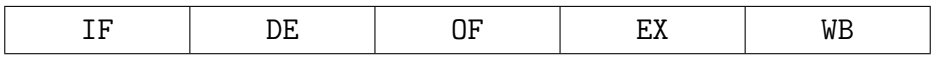
```
ori   $s1, $zero, 0x2021
sll   $s2, $s1, 1
slti  $s3, $s2, 0x4043
sub   $s4, $s3, $s2
```

Aufgabe 8 *Pipelining* (13 Punkte)

Gegeben sei das sequentielle Programmstück:

```
S1:  add $t0, $t1, $t2    ; $t0 = $t1 + $t2
S2:  sub $t1, $t0, $t2    ; $t1 = $t0 - $t2
S3:  add $t2, $t1, $t1    ; $t2 = $t1 + $t1
S4:  mul $t0, $t1, $t2    ; $t0 = $t1 * $t2
```

Dieses Programmstück wird von einem RISC-Prozessor mit folgender fünfstufiger Pipeline ausgeführt:



Die einzelnen Stufen haben folgende Bedeutung :

IF: Instruction Fetch - Befehl holen

DE: Decode - Befehl dekodieren

OF: Operand Fetch - Operanden holen

EX: Execute - Befehl ausführen

WB: Write Back - Ergebnis speichern

Dabei ist ein Schreibvorgang in ein Zielregister erst am Ende der WB-Stufe abgeschlossen.

1. Bestimmen Sie alle Datenabhängigkeiten innerhalb dieses Programmstücks. Geben Sie zu jeder Datenabhängigkeit die beiden beteiligten Befehle, das ursächliche Register und den Typ der Datenabhängigkeit an. [5 P.]

2. Zu Beginn des Programmstücks seien die Register folgendermaßen belegt: [4 P.]

$t0	$t1	$t2
3	6	8

   Geben Sie die Registerbelegung nach Ablauf des Programmstücks an. Tragen Sie hierzu in die Tabelle auf dem Lösungsblatt den Zustand der Pipeline bei jedem Taktzyklus und der Register *nach* jedem Taktzyklus ein.

   Wie viele Takte werden benötigt, um das Programm abzuarbeiten?

3. Wie wäre die Belegung der Register, wenn der Prozessor keine Pipeline besäße, sondern die Befehle rein sequentiell abarbeitet? [1 P.]

4. Die einzige Methode, die Pipelinekonflikte bei diesem Prozessor zu beheben, sei das Einfügen von `NOP`-Befehlen (*No Operation*) in den Befehlsstrom. [3 P.]

   Fügen Sie möglichst wenige `NOP`-Befehle in das Programmstück ein, sodass es zu keinen Konflikten mehr kommt und das Ergebnis dem der sequentiellen Ausführung entspricht. Geben Sie das modifizierte Programmstück an.

   Wie viele Takte werden nun benötigt?

Aufgabe 9 *Speicherbausteine* (7 Punkte)

1. Welches Problem ist bemerkbar, wenn man die Entwicklung der Verarbeitungsgeschwindigkeit von Prozessorn mit der Entwicklung der Zugriffsgeschwindigkeit von DRAM-Speicherchips des Hauptspeichers vergleicht? 1 P.

2. Skizzieren Sie das Transistor-Schaltbild einer 1-Bit Speicherzelle eines statischen RAM-Bausteins (SRAM). 3 P.

3. Was versteht man unter Zugriffszeit und Zykluszeit eines Speicherbausteins? 2 P.

4. In welchen Ebenen der Speicherhierarchie kommen magnetische Speicher zum Einsatz? 1 P.

Aufgabe 10 *Virtuelle Speicherverwaltung* (8 Punkte)

Gegeben sei eine Speicherverwaltungseinheit (MMU) mit einer Seitengröße von 4 KiByte, 8 virtuellen Seiten und 4 physikalischen Seiten (Frames). Der aktuelle Ausschnitt der Seitentabelle ist in Tabelle 2 angegeben.

Virtuelle Seitennummer	Physikalische Seitennummer
0	1
1	3
2	-
3	-
4	2
5	-
6	0
7	-

Tabelle 2: Seitentabelle

1. Skizzieren Sie die Unterteilung der 32 Bit breiten virtuellen Adresse. 1 P.

2. Ermitteln Sie die physikalischen Adressen zu den folgenden virtuellen Adressen: 4 P.

   512, 4095, 4097, 4198, 8191, 8192, 8400, 0

   Füllen Sie dazu die auf dem Lösungsblatt bereit gestellte Tabelle aus. Geben Sie die Adresse in dezimaler Schreibweise an. Rechnen Sie dazu das Ergebnis explizit aus.

Zur Beschleunigung der Adressberechnung soll ein Cache-Speicher als *Translation Lookaside Buffer (TLB)* eingesetzt werden, der die 32 zuletzt benutzten Einträge aus dem Seitentabellen-Verzeichnis und der Seitentabelle speichert.

3. Unter welchen allgemeinen Bedingungen wird eine Beschleunigung der Adressumsetzung durch einen *Translation Lookaside Buffer (TLB)* erreicht? 1 P.

4. Wie breit ist der *Tag* eines Cache-Eintrags? Gehen Sie dabei von einer n Bit breiten virtuellen Adresse, einer m Bit breiten physikalischen Adresse und einer Seitengröße von 4 KiByte aus. 2 P.