KIT-Fakultät für Informatik
Prof. Dr.-Ing Uwe Hanebeck, Prof. Dr.-Ing. Jörg Henkel

Aufgabenblätter zur Prüfung

Digitaltechnik und Entwurfsverfahren (TI-1)

und

Rechnerorganisation (TI-2)

am 26. Februar 2020, 13:00 – 15:00 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* (7 Punkte)

Gegeben sei die Funktionstabelle der Schaltfunktion f:

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

1. Geben Sie die disjunktive Normalform (DNF) der Schaltfunktion f an. 1 P.

2. Tragen Sie die Schaltfunktion f in das im Lösungsblatt vorbereitete KV-Diagramm ein. Geben Sie alle Primimplikate von f an und zeichnen Sie die zugehörigen Blöcke im KV-Diagramm ein. 3 P.
   Geben Sie für jedes Primimplikat an, ob es sich um ein Kernprimimplikat, ein Wahlprimimplikat oder ein entbehrliches Primimplikat handelt.
   Geben Sie eine konjunktive Minimalform (KMF) der Schaltfunktion f an.

3. Die Schaltfunktion f soll mit Hilfe eines 1:8-Demultiplexers und *möglichst wenigen* weiteren Gattern realisiert werden. Geben Sie das zugehörige Schaltnetz an. 1 P.

4. Die Schaltfunktion f soll mit Hilfe eines 2:1-Multiplexers und *möglichst wenigen* weiteren Gattern realisiert werden. Geben Sie das zugehörige Schaltnetz an. 2 P.

Aufgabe 2 *CMOS-Technologie* (8 Punkte)

1. Realisieren Sie die folgende Schaltfunktion y als CMOS-Schaltnetz **6 P.**

$$y = a\,b \;\vee\; c\,d \;\vee\; e\,f\,g$$

   Die Verwendung von negierten Eingangsvariablen sei nicht erlaubt.

2. Welche Schaltfunktion wird durch das CMOS-Schaltnetz realisiert, dessen n-MOS-Netz in Abbildung 1 näher beschrieben ist? **2 P.**

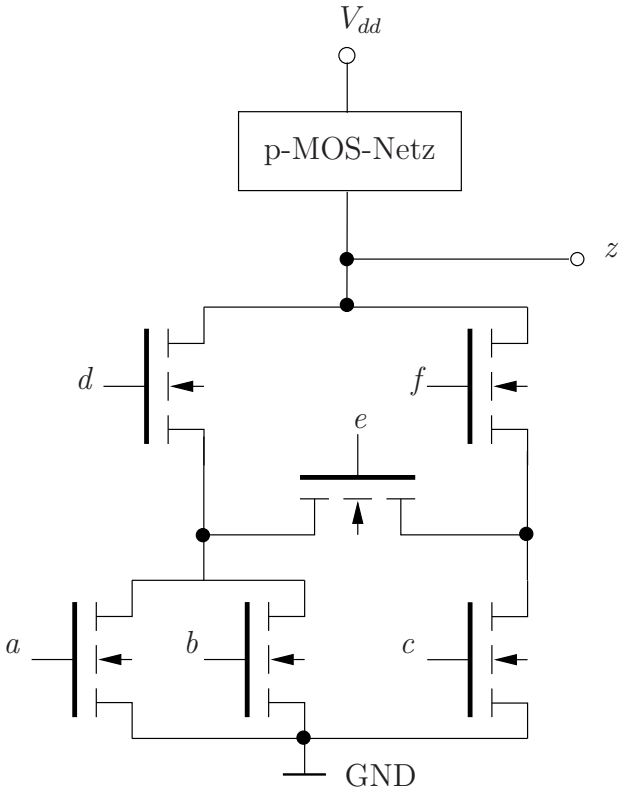

Abbildung 1: CMOS-Schaltnetz

Aufgabe 3 *Laufzeiteffekte* (7 Punkte)

Gegeben ist das in Abbildung 2 dargestellte Schaltnetz. NAND-, NOR- und OR-Gatter haben eine Totzeit von $5\,ns$, das XOR-Gatter von $7\,ns$ und der Inverter von $2\,ns$. Die Eingangsvariablen a, b und c wechseln zum Zeitpunkt $t = 0$ *gleichzeitig* von 0 auf 1.

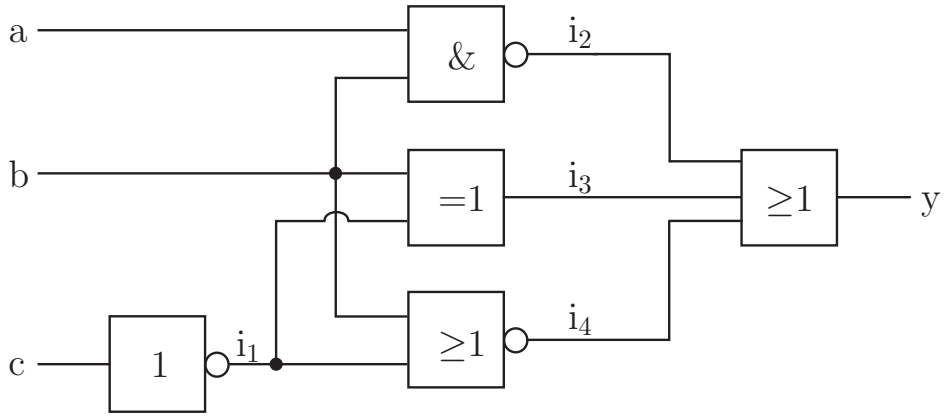

Abbildung 2: Schaltnetz

1. Geben Sie die Verläufe der Signale i_1, i_2, i_3, i_4 und y an, indem Sie das im Lösungsblatt angegebene Zeitdiagramm vervollständigen. 4 P.

2. Treten im Zeitverlauf von y Hasardfehler auf? Falls ja, um welchen Typ handelt es sich bei dem zu Grunde liegenden Hasard? Begründen Sie Ihre Antwort. 3 P.

Aufgabe 4 *Schaltwerke* (15 Punkte)

Gegeben sei ein synchrones Schaltwerk mit zwei flankengesteuerten JK-Flipflops, einer Eingangsvariablen x und einer Ausgangsvariablen y (siehe Abbildung 3).

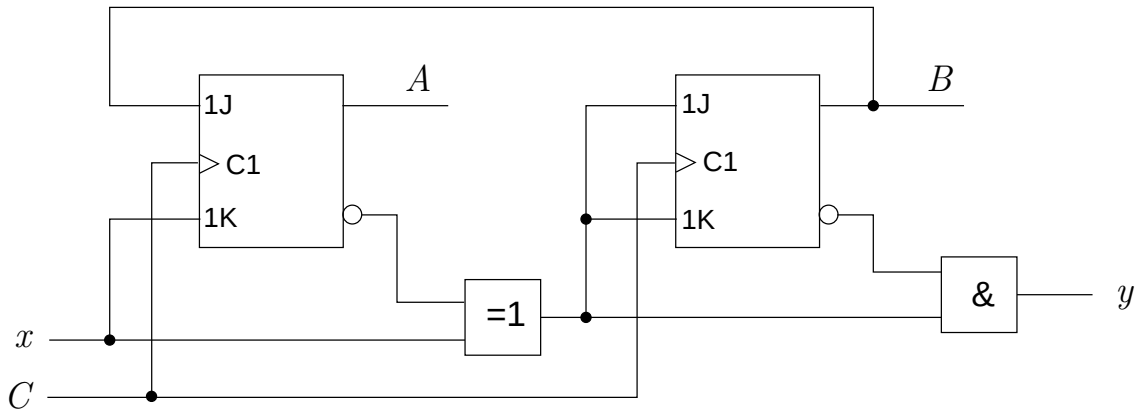

Abbildung 3: Schaltwerk I

1. Wieviele Zustände kann das Schaltwerk maximal haben? 1 P.

2. Stellen Sie die kodierte Ablauftabelle des Schaltwerks auf. Stellen Sie Ihre Lösung schrittweise dar. 6 P.

   Hinweis: Bestimmen Sie die Ansteuerfunktionen der Flipflops, die Überführungsfunktionen und die Ausgabefunktion des Schaltwerks.

In Abbildung 4 ist der Automatengraph eines zweiten synchronen Schaltwerks mit der Eingangsvariablen e, der Ausgangsvariablen a und den Zuständen Z_0, Z_1, Z_2 und Z_3 dargestellt.

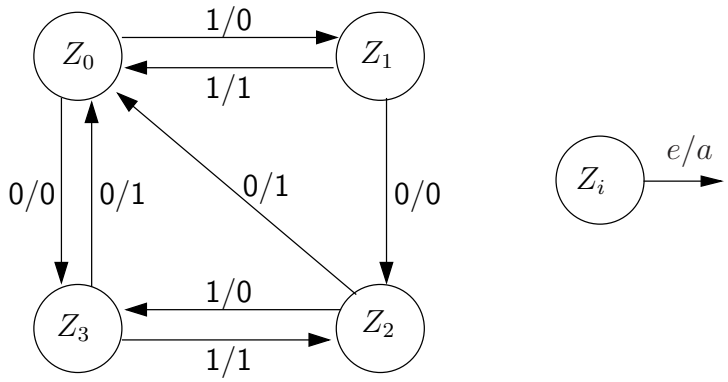

Abbildung 4: Automatengraph vom Schaltwerk II

3. Nehmen Sie an, dass sich das Schaltwerk am Anfang im Zustand Z_0 befindet. Vervollständigen Sie die im Lösungsblatt angegebene Tabelle der Zustands-, Eingabe- und Ausgabefolgen des Schaltwerks. 4 P.

In Abbildung 5 ist ein weiteres Schaltwerk dargestellt.

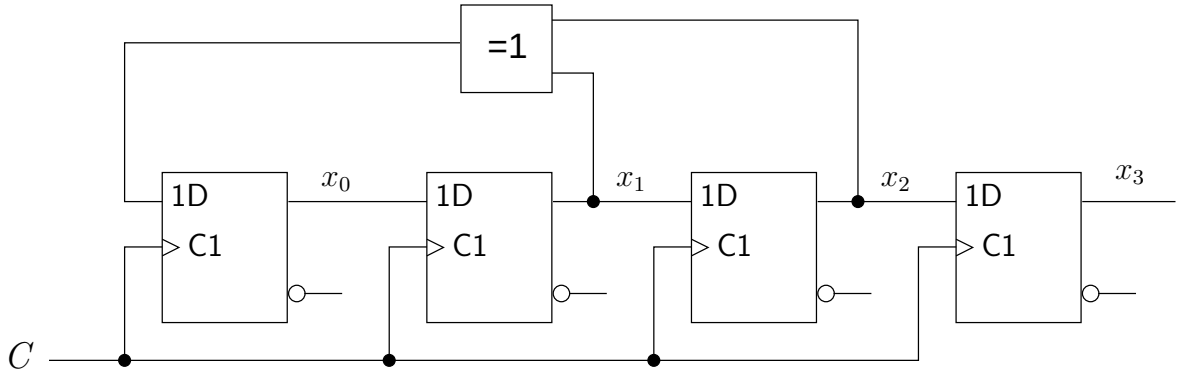

Abbildung 5: Schaltwerk III

4. Vervollständigen Sie die Verläufe der Signale x_0, x_1, x_2 und x_3 im angegebenen Zeitdiagramm im Lösungsblatt. 4 P.

Aufgabe 5 *Rechnerarithmetik* (8 Punkte)

Hinweis: Geben Sie in dieser Aufgabe *immer* den Rechenweg an.

1. Wie viele Prüfbits sind für eine Einzelbit-Fehlerkorrektur in 100 Bit Datenwörtern erforderlich? 1 P.

2. Wandeln Sie die Zahl $21,11_3$ in eine Dezimalzahl um. 1 P.

3. Wandeln Sie die Zahl $F0,A1_{16}$ in eine Zahl zur Basis 8 um. 1 P.

4. Welche Bereiche einer Gleitkomma-Zahl nach dem IEEE-754-Standard (Vorzeichen, Exponent, Mantisse) ändern sich gegebenfalls, falls eine beliebige Zahl mit -4 multipliziert wird. Begründen Sie Ihre Antwort. 2 P.

5. Gegeben sei die folgende 32-Bit Folge 3 P.

   1001 1000 0000 0000 0000 0000 0001 0100

   Was stellt diese Folge dar, wenn sie interpretiert wird als

   (a) BCD-Zahl.

   (b) Vorzeichenlose Dualzahl. Geben Sie den dezimalen Wert an.

   (c) Gleitkomma-Zahl im IEEE-754-Standard in einfacher Genauigkeit. Geben Sie den dezimalen Wert an.

   Hinweis: Sie brauchen Zweier-Potenzen nicht explizit auszurechnen. Ergebnisse können z. B. in der Form $2^i - 2^j + 7$ angegeben werden.

Aufgabe 6 *MIPS-Assembler* (12 Punkte)

1. Schreiben Sie die folgenden C-Kontrollstrukturen in MIPS-Assembler um. 3 P.

```
for (i = 100; i > 0; i--) j = j * i;
```

   Die Variablen `i` und `j` stehen in den Registern `$a0` und `$a1`. Verwenden Sie das Register `$v0` zur Speicherung temporärer Variablen.

2. Das folgende Programmstück soll die Summe der Elemente eines Arrays aus 32-Bit Integer-Zahlen in Zweierkomplement-Form berechnen. Das Register `$a0` sei mit der Adresse des Arrays initialisiert; das Register `$a1` sei mit der Anzahl der Array-Elemente initialisiert. Alle anderen Register seien nicht initialisiert. Am Ende soll die Summe in `$v0` stehen. 3 P.

```
array_sum:      li    $t0, 0($a0)
                add   $v0, $v0, $t0
                addi  $a0, $a0, 4
                addi  $a1, $a1, -1
                bgez  $a0, array_sum
```

   Leider haben sich bei der Implementierung einige Fehler eingeschlichen. Finden Sie diese Fehler und korrigieren Sie das Programm, so dass es korrekt arbeitet.

3. Geben Sie für das folgende MIPS-Programmstück den Inhalt des Zielregisters in hexadezimaler Schreibweise nach der Ausführung des jeweiligen Befehls an. 3 P.

```
subi   $s1, $zero, 0x2
srl    $s2, $s1, 4
slti   $s3, $s2, 100
lui    $s4, 0x40
xor    $s5, $s1, $s4
```

4. Gegeben sei das folgende MIPS-Programmstück:

```
        .data
vec:    .word 12, 13, 17, 19, 23, 29, 31, 37, 41, 43

        .text
main:   lw $t1, vec
        lw $t2, vec+0x18
        lw $t3, vec($t1)
        lw $t4, vec+0x14($t1)
```

   (a) Geben Sie die Inhalte der Register `$t1`, `$t2`, `$t3` und `$t4` in hexadezimaler Schreibweise nach der Ausführung des obigen Programmcodes an. 2 P.

   (b) Geben Sie MIPS-Code an, mit dem man die Adresse von `vec` im Register `$s0` speichert. 1 P.

Aufgabe 7 *Pipelining* (10 Punkte)

Gegeben sei eine Pipeline mit den folgenden Stufen:

- 1. Stufe: Befehl holen
- 2. Stufe: Befehl dekodieren und Operanden aus Registern bereitstellen
- 3. Stufe: Befehl ausführen
- 4. Stufe: Zugriff auf Speicher
- 5. Stufe: Ergebnis im Register speichern

Bei Lade/Speicher-Befehle wird in der Stufe 3 die Adresse berechnet und in der Stufe 4 auf den Speicher zugegriffen. Alle anderen Befehlsarten führen in der Stufe 4 keine Operation durch. Das Laden von Operanden in der Stufe 2 und Speichern von Ergebnissen in der Stufe 5 ist jeweils erst am Ende des Taktes abgeschlossen.

Betrachten Sie die folgende Programmsequenz:

```
S1:   add    R5, R2, R2  //   R5 = R2 + R2
S2:   sub    R1, R4, R2  //   R1 = R4 - R2
S3:   sub    R4, R5, R3  //   R4 = R5 - R3
S4:   add    R1, R1, R4  //   R1 = R1 + R4
S5:   addi   R6, R5, 42  //   R6 = R5 + 42
```

1. Bestimmen Sie alle Datenabhängigkeiten in der gegebenen Programmsequenz und bestimmen sie jeweils die Art der Abhängigkeit. 3 P.
2. Welche der gefundenen Abhängigkeiten führen in der obigen Programmsequenz zu Pipelinekonflikten? 3 P.
3. Beseitigen Sie alle Pipelinekonflikte durch Einfügen von möglichst wenigen NOP-Befehlen ohne die Reihenfolge der Befehle oder die Befehle an sich zu verändern. Es genügt, wenn Sie anstelle der gegebenen Befehle den jeweiligen Label (z. B. `S1`) angeben. 2 P.
4. Welches Problem tritt auf, wenn bei der gegebenen Pipeline ein gemeinsamer Daten- und Befehls-Cache benutzt wird, auf den nur ein Lesezugriff pro Takt möglich ist? 2 P.

Aufgabe 8 *Cache-Speicher* (12 Punkte)

(Hinweis: 1 KByte = 1024 Byte, 1 MByte = 1024 KByte, 1 GByte = 1024 MByte)

1. Bei einem Cache-Speicher mit einer Speicherkapazität von 512 KByte ist die Hauptspeicheradresse in ein 15 Bit Tag-Feld, ein 12 Bit Index-Feld und ein 5 Bit Byte-Offset unterteilt. Geben Sie bei der Beantwortung der folgenden Fragen den Lösungsweg an.

   (a) Bestimmen Sie die Blockgröße in Bytes. 1 P.

   (b) Wieviele Einträge besitzt der Cache-Speicher? 1 P.

   (c) Wie ist der Cache-Speicher organisiert? 2 P.

2. Es soll ein 2-fach-assoziativer *(2-way set associative cache)* Cache-Speicher mit 256 Sätzen und einer Blockgröße von 8 Byte realisiert werden. Nehmen Sie an, dass die Hauptspeicheradresse 32 Bit breit ist. Zur Verwaltung eines Cacheblocks wird nur ein Statusbit (*Valid*-Bit: V) verwendet. 3 P.

   Bestimmen Sie den insgesamt erforderlichen Speicherbedarf in Byte zur Realisierung dieses Cache-Speichers. Geben Sie dabei den Rechenweg an.

3. Gegeben sei ein direkt-abgebildeter Cache-Speicher (*direct mapped cache*) mit einer Speicherkapazität von 64 Byte und einer Blockgröße von 16 Byte. Als Aktualisierungsstrategie wird ein Durchschreibverfahren *(write-through mit write-no-allocate)* verwendet, bei dem ein CPU-Datum bei einem *write miss* nur in den Speicher geschrieben, aber nicht in den Cache. Bei einem *write hit* wird ein CPU-Datum sowohl in den Cache als auch in den Speicher geschrieben. 5 P.

   Die Adressen der folgenden Lese- und Schreibzugriffe sind in dezimaler Schreibweise angegeben.

Adresse	**0**	**32**	**96**	**16**	**112**	**32**	**16**	**112**	**64**	**0**
read/write	r	r	r	r	r	r	r	w	w	r
Index	0	2								
Tag	0	0								
Hit/Miss	Miss									

   Vervollständigen Sie die obige Tabelle im Lösungsblatt. Verwenden Sie dabei **Miss** für Cache-Miss und **Hit** für Cache-Hit.

Aufgabe 9 *Virtuelle Speicherverwaltung* (11 Punkte)

Ein Rechnersystem enthält eine Speicherverwaltungseinheit (MMU) zur Umsetzung von virtuellen in physikalische Seitenadressen (Abbildung 6). Die MMU bildet einen virtuellen Adressraum von 2^V Bytes auf einen physikalischen Adressraum der Größe 2^M Bytes ab und benutzt dabei Seiten der Größe 2^P Bytes. Nehmen Sie an, dass die MMU eine Byte-Adressierung benutzt und die gesamte Seitentabelle im Hauptspeicher ist.
(Hinweis: 1 KByte = 1024 Byte, 1 MByte = 1024 KByte, 1 GByte = 1024 MByte)

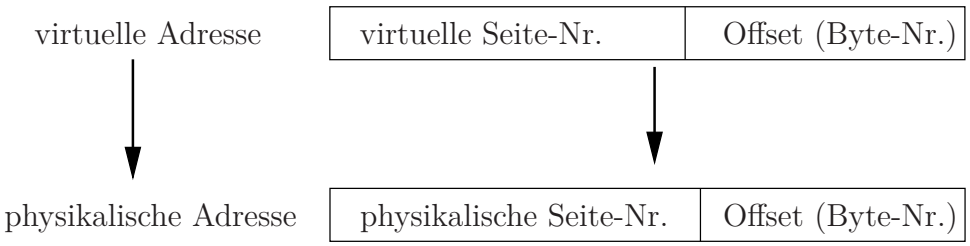

Abbildung 6: Format der virtuellen und physikalischen Adressen

1. Vervollständigen Sie die im Lösungsblatt angegebene Abbildung, indem Sie die Länge der einzelnen Felder sowohl in der virtuellen als auch in der physikalischen Adresse angeben. 2 P.

2. Wieviele Seiten können auf einmal im physikalischen Adressraum gespeichert werden? Wieviele Einträge hat die Seitentabelle? 2 P.

3. Wieviele Bits benötigt ein Eintrag in der Seitentabelle, wenn zu jedem Eintrag zusätzlich zwei Steuerbits benötigt werden? Wieviele Seiten benötigt die gesamte Seitentabelle, wenn $V = 26$, $M = 20$ und $P = 8$ ist? 4 P.

4. In der folgenden Tabelle ist ein Ausschnitt aus der Seitentabelle gegeben. Welchen physikalischen Adressen entsprechen die dezimalen virtuellen Adressen 5348 und 6484, wenn $P = 11$ ist? (Hinweis: $2^{11} = 2048$) 3 P.

Virtuelle Seitennummer	Physikalische Seitennummer
0	4
1	2
2	7
3	5
4	3
5	1
6	8
7	0
⋮	⋮