

Universität Erlangen-Nürnberg
Technische Fakultät
Lehrstuhl für Hardware-Software-Co-Design
Prof. Dr.-Ing. Jürgen Teich

Klausur Technische Informatik 1

14. September 2007

Name	
Matrikelnummer	
Studienrichtung	

Aufgabe	1	2	3	4	5	Σ
max. Punkte	20	10	10	20	20	80
erreichte Punkte						
Note						

Aufgabe 1 (Zahlendarstellungen und Rechnerarithmetik) (20 P)

Die nebenstehende Tabelle gibt eine Implementierungsvorschrift zur Multiplikation zweier vorzeichenloser Binärzahlen nach der Schulmethode an.

$$\begin{array}{r}
 a_2 a_1 a_0 \times b_2 b_1 b_0 \\
 \hline
 a_2 b_0 \ a_1 b_0 \ a_0 b_0 \\
 + \quad a_2 b_1 \ a_1 b_1 \ a_0 b_1 \\
 + \quad a_2 b_2 \ a_1 b_2 \ a_0 b_2 \\
 \hline
 p_5 \ p_4 \ p_3 \ p_2 \ p_1 \ p_0
 \end{array}$$

- a) Zeichnen Sie die Implementierung eines Multiplizierers unter Verwendung von UND-Gattern und Volladdierern, der die obige Methode direkt umsetzt. (4 Punkte)

Im Folgenden soll der Multiplizierer ausschließlich zum Quadrieren benutzt werden ($a_2 = b_2$, $a_1 = b_1$, $a_0 = b_0$). Dieses erlaubt gegenüber dem Multiplizierer aus a) erhebliche Vereinfachungen.

- b) Geben Sie an, wie sich die gegebene Tabelle für den Fall des Quadrierens vereinfachen lässt. Wie viele UND-Gatter und wie viele Volladdierer werden minimal zur Implementierung des Quadrierers benötigt? (3 Punkte)
- c) Zeichnen Sie das Schaltnetz ihres Quadrierers aus Aufgabe b). (3 Punkte)

Berechnen Sie die folgenden Aufgaben unter Angabe des Lösungswegs:

- d) Quadrieren Sie die Gleitkommazahl 1101 1101 1101 1101. Die Zahl besitzt (von links gelesen) ein Vorzeichenbit, 7 Bit für den Exponenten und 8 Bit für die Mantisse. Vergleichbar zum IEEE 754-Format beträgt der Bias 63 und die Mantisse ist in der Form $1, M$ codiert. Geben Sie das Ergebnis in dieser Zahlendarstellung an. (4 Punkte)
- e) Quadrieren Sie die Zahl 1234_5 ohne dabei die Basis 5 zu verlassen. (3 Punkte)
- f) Dividieren Sie 170_{10} durch 5_7 im Binärsystem. (3 Punkte)

Aufgabe 2 (Flipflop-Schaltungen) (10 Punkte)

Durch zusätzliche Logik können verschiedene Flipflop-Typen ineinander überführt werden. Entwerfen Sie ein Schaltnetz, welches vor den Eingang eines D-Flipflops geschaltet wird und dafür sorgt, dass sich die Gesamtschaltung wie ein JK-Flipflop verhält. Lösen Sie dazu folgende Aufgaben:

- a) Geben Sie die Funktionstabelle für ein JK-Flipflop an. (1 Punkt)
- b) Geben Sie die charakteristische Funktion eines D-Flipflops an. (1 Punkt)
- c) Geben Sie Funktionstabelle für das gesuchte Schaltnetz an. (2 Punkte)
- d) Bestimmen Sie eine konjunktive Minimalform der gesuchten Funktion durch Verwendung eines Symmetriediagramms. (2 Punkte)
- e) Geben Sie eine Schaltfunktion für Ihre Lösung aus d) an, die ausschließlich NOR-Operationen benutzt. (2 Punkte)
- f) Zeichnen Sie die Schaltung, welche sich wie ein JK-Flipflop verhält und aus einem D-Flipflop und NOR-Gattern besteht. (2 Punkte)

Aufgabe 3 (Schaltalgebra)

(10 Punkte)

Gegeben ist die Funktion $f(a, b, c, d) = bd + \bar{a}\bar{b}d + \bar{b}\bar{c}d + abc$

- Bestimmen Sie die disjunktive Minimalform von f mit Hilfe des Quine/McCluskey-Verfahrens. (5 Punkte)
- Entwickeln Sie f nach dem Entwicklungssatz der Schaltalgebra. Führen Sie dieses für die Variablenordnung a, b, c, d aus. Stellen Sie ferner f als BDD (Binary Decision Diagram) dar. (5 Punkte)

Aufgabe 4 (VHDL)

(20 Punkte)

Jeder Prozessor beinhaltet in seinem Kern mehrere Register, ein Adressrechenwerk zur Berechnung des Programmzählers und eine ALU (engl. arithmetic logic unit). In dieser Aufgabe sollen diese Kernkomponenten unabhängig voneinander in VHDL implementiert werden.

- Erstellen Sie ein Modul, welches ein 8-Bit breites Register implementiert. Geben Sie die Schnittstellen in Form einer Entity an und implementieren Sie die Architecture des Moduls. (4 Punkte)
- Gegeben ist die untenstehende Entity eines einfachen Adressrechenwerkes ADR. Implementieren Sie die entsprechende Architecture des Adressrechenwerkes mit dem folgendem Verhalten: Das Modul zählt den Programmzähler (Ausgang PC) bei steigender Taktflanke um 1 weiter, wenn das Signal `count='1'` gesetzt ist. Ansonsten kann der Programmzähler relativ um den im Einerkomplement codierten Wert `offset` springen, wenn das Signal `branch='1'` gesetzt ist. Sind beide Signale nicht gesetzt, so soll der Programmzähler unverändert bleiben, es sei denn, dass das Signal `reset='1'` gesetzt ist. Denn wann immer `reset` aktiv ist, soll, unabhängig von allen anderen Eingangssignalen, der Programmzähler auf Null zurückspringen. (8 Punkte)

```
entity ADR is
port (Takt:    in bit;
      reset:   in bit;
      count:   in bit;
      branch:  in bit;
      offset:  in bit_vector(4 downto 0);
      PC:     out bit_vector(7 downto 0) );
end pc_counter;
```

- Die ALU soll die in der Tabelle angegebenen arithmetischen und logischen Funktionen ausführen. Diese werden mit dem Signal `Sel` ausgewählt und das Resultat liegt am Ausgang `Res` an. Implementieren Sie die Architecture zu der gegebenen Entity der ALU. (8 Punkte)

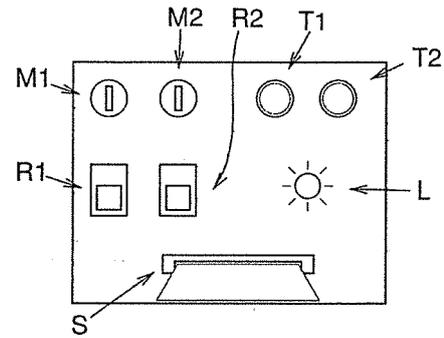
```
entity ALU is
port ( A:    in bit_vector(7 downto 0);
      B:    in bit_vector(7 downto 0);
      Sel:  in bit_vector(1 downto 0);
      Res:  out bit_vector(7 downto 0) );
end ALU;
```

Sel	Funktion
00	A+B
01	A-B
10	A NAND B
11	A NOR B

Aufgabe 5 (Automaten)

(20 Punkte)

Gegeben sei ein Fahrkartenautomat mit folgendem Verhalten: Ein Kunde kann in den Automat 1 Euro Münzen im Münzschlitz M1 sowie 2 Euro Münzen im Münzschlitz M2 einwerfen. Sobald der Wert der einbehaltenen Münzen mindestens dem Fahrkartenpreis von 2 Euro entspricht, beginnt die Lampe L zu leuchten. Nachträglich eingeworfene 1 Euro Münzen werden im Münzrückgabefach R1, 2 Euro Münzen im Münzrückgabefach R2 ausgeworfen.



Drückt der Kunde auf die Taste T1, wenn die Lampe L leuchtet, wird eine Fahrkarte im Fahrkartenschacht S ausgegeben, die Lampe L hört auf zu leuchten und der möglicherweise auftretende Differenzbetrag von 1 Euro wird über das Münzrückgabefach R1 zurückerstattet. Einen Abbruch des Bezahlvorgangs kann der Kunde jederzeit durch Drücken der Taste T2 vornehmen. Der Wert der einbehaltenen Münzen wird in diesem Fall über die Münzrückgabefächer R1 und R2 zurückerstattet, dabei müssen jedoch nicht notwendigerweise die gleichen Münzen zurückgegeben werden, wie einbehalten wurden. Zur Vereinfachung sei Folgendes angenommen: Der Kunde drückt die Tasten nie gleichzeitig und wirft die Münzen stets nacheinander ein. Wenn der Kunde eine Taste drückt, wirft er keine Münze ein und anders herum. Außerdem soll davon ausgegangen werden, dass der Automat über einen unbegrenzten Vorrat von 1 und 2 Euro Münzen verfügt.

- Zeichnen Sie den Automatengraphen eines Mealy-Automaten, der die in der Angabe beschriebene Funktion nachbildet. (6 Punkte)
- Wie viele Bits werden bei einer Binär-Codierung der Ausgabe Ihres Automaten mindestens benötigt? (1 Punkt)
- Geben Sie eine Binär-Codierung für die Zustände Ihres Automaten an, die mit der geringst möglichen Anzahl von Bits auskommt. (1 Punkt)

Betrachten Sie nun folgende Automatentafel mit dem Zustand Q^v , dem Folgezustand Q^{v+1} , der Eingabe X und der Ausgabe Y :

Q^v		X		Q^{v+1}		FF2		FF1		Y
q_2^v	q_1^v	x_2	x_1	q_2^{v+1}	q_1^{v+1}	J	K	J	K	y
0	0	0	0	0	0					0
0	0	0	1	0	1					1
0	0	1	0	1	1					1
0	0	1	1	1	0					1
0	1	0	–	1	0					0
0	1	1	–	0	1					1
1	0	–	0	1	0					0
1	0	–	1	1	1					1
1	1	0	–	1	1					0
1	1	1	–	0	0					1

- Begründen Sie anhand der Automatentafel, um welchen Automatentypen es sich handelt. (1 Punkt)
- Ergänzen Sie die Automatentafel um eine Funktionstabelle für eine Implementierung des Automaten mittels JK-Flipflops. (2 Punkte)

- Gewinnen Sie aus einem Symmetriediagramm einen Booleschen Ausdruck zur Ansteuerung des Ausgangs Y in konjunktiver Minimalform. (5 Punkte)
- Entwerfen Sie für den Booleschen Ausdruck aus Teilaufgabe f) ein Schaltnetz. Verwenden Sie dazu ausschließlich NAND-Gatter. (4 Punkte)