Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl. Inform. M. Klein
Dipl. Inform. T. In der Rieden

1. Teilklausur Informatik II

(Datum: 21.06.2003)

---

1. **Aufgabe** **(4 + 6 Punkte)**

   Beweise:

   (a) $\sum_{i=0}^{n} i = \frac{n \cdot (n+1)}{2}$

   (b) $\sum_{i=0}^{n} i^2 = \frac{n \cdot (n+1) \cdot (2n+1)}{6}$

2. **Aufgabe** **(4 Punkte)**

   Seien $A := \{1, 2, 3, 5\}$ und $B := \{1, 2, 4, 6\}$.
   Gib die folgenden Mengen und ihre Mächtigkeit an:

   (a) $A \cup B$.

   (b) $A \setminus (A \cap B)$.

   (c) A^2.

   (d) $A \times B$.

3. **Aufgabe** **(5 Punkte)**

   Beweise durch Induktion über $\#B$:

   $\#(A \cup B) = \#A + \#B - \#(A \cap B)$.

4. **Aufgabe** **(5 Punkte)**

   Seien $a, b, c \in \mathbb{N}$. Beweise durch Induktion über c:

   $a + (b + c) = (a + b) + c$.

   Benutze dazu nur die Definition der Addition:

   $a + 0 = a,\ a + (b + 1) = (a + b) + 1$.

5. **Aufgabe** **(4 + 4 Punkte)**

   (a) Sind die folgenden Zeichenreihen vollständig geklammerte Boolesche Ausdrücke? Begründe deine Antwort.

   i. $((x_1 \vee (\sim x_3)) \wedge (x_2 \vee x_3))$.
   ii. $(\sim (x_1 \wedge \sim (x_2 \wedge \sim x_3)))$.
   iii. $(x_1 \vee (\sim x_1) \wedge (x_2 \vee (\sim x_2))$.
   iv. $(((x_2 \vee 0) \wedge (\sim x_3)) \vee (1 \wedge x_1))$.

   (b) Werte die vollständig geklammerten Booleschen Ausdrücke aus (a) sowie deren Teilausdrücke unter folgender Belegung aus:
   $\varphi(x_1) = 0,\ \varphi(x_2) = 1,\ \varphi(x_3) = 1$.

6. **Aufgabe** **(2 + 2 Punkte)**

   Beweise:

   (a) $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$.

   (b) $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$.

Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl. Inform. M. Klein
Dipl. Inform. T. In der Rieden

1. Teilklausur Informatik II

(Datum: 21.06.2003)

---

7. **Aufgabe** **(2 + 3 + 4 Punkte)**

Die Funktionen f_1 und f_2 seien wie folgt definiert:

x_1	x_2	x_3	f_1	f_2
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

(a) Bestimme die vollständige disjunktive Normalform von f_1.

(b) Bestimme die vollständige rekursive Funktionszerlegung von f_2.
(Zur Erinnerung: $f(x_1, ..., x_n) = (x_1 \wedge f(1, x_2, ..., x_n)) \vee (\overline{x_1} \wedge f(0, x_2, ..., x_n)))$.

(c) Gib einen Schaltkreis an, der die Funktionen f_1 und f_2 berechnet mit Gesamtkosten ≤ 7.

8. **Aufgabe** **(2 + 4 Punkte)**

(a) Gib die Definition des Trägers T_f einer Schaltfunktion f an.

(b) Für $a \in \{0,1\}^n$ definieren wir $$m(a) := \bigwedge_{i=1}^{n} X_i^{a_i}$$

Beweise den Darstellungssatz: Für jede n-stellige Schaltfunktion f gilt
$$f(x_1, \ldots, x_n) \equiv \bigvee_{a \in T_f} m(a)$$

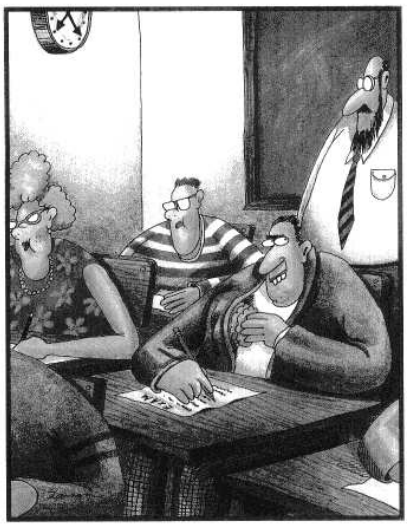
9. **Aufgabe** **(5 + 5 + 8 Punkte)**

Bringe folgende Formeln in eine geschlossene Form ohne Summenzeichen und beweise die Korrektheit deiner Lösung:

(a) $C(1) = 1,\ C(n) = C(n-1) + n$ für $n \in \mathbb{N}$.

(b) $C(1) = 1,\ C(n) = C(n/2) + 2$ für $n = 2^k,\ k \in \mathbb{N}$.

(c) $C(1) = a,\ C(n) = 2 \cdot C(n/2) + 2 \cdot a$ für $n = 2^k,\ a, k \in \mathbb{N}$.



Midway through the exam, Allen pulls out a bigger brain.

Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl. Inform. M. Klein
Dipl. Inform. T. In der Rieden

1. Teilklausur Informatik II

(Datum: 21.06.2003)

---

10. **Aufgabe** **(2 + 6 + 4 Punkte)**
    (a) Gib eine **formale** Definition eines Addierers.
    (b) Konstruiere **rekursiv** einen Addierer deiner Wahl und beweise seine Korrektheit gegen deine Definition. Die Korrektheit des Fulladders kann dabei vorausgesetzt werden.
    (c) Gib Kosten und Tiefe deines Addierers in geschlossener Form an.

11. **Aufgabe** **(2 + 4 + 4 Punkte)**
    Seien $a, b \in \{0,1\}^n,\ s \in \{0,1\}^{n+1},\ \langle a\rangle > \langle b\rangle,\ [s] := [a] + [\overline{b}] + 1$. Beweise:
    (a) $\langle a\rangle = [0a]$.
    (b) $-[b] = [\overline{b}] + 1$.
    (c) $\langle a\rangle - \langle b\rangle = \langle s[n-1:0]\rangle$.

12. **Aufgabe** **(10 + 3 Punkte)**
    Für Zahlen $i \in \{0,\ldots,n-1\},\ n = 2^k$ definieren wir die unäre Darstellung als $0^{n-1-i}10^i$ und die halb-unäre Darstellung als $0^{n-1-i}1^{i+1}$.
    (a) Konstruiere einen Schaltkreises zur Konvertierung von unären Eingaben in halb-unäre Ausgaben mit Kosten $O(n)$ und Tiefe $O(\log n)$.
    (b) Konstruiere einen Schaltkreis zur Konvertierung halb-unärer Eingaben in unäre Ausgaben mit Kosten n und Tiefe 1.

13. **Aufgabe** **(5 Punkte)**
    In der Vorlesung wurde bewiesen, dass man einen zyklischen Rechtsshift um $\langle a\rangle$ Stellen simulieren kann durch einen zyklischen Linksshift um $(\langle\overline{a}\rangle + 1)$ Stellen. Ein kombinierter zyklischer 2^n-Bit Links-/Rechts-Shifter bestand dementsprechend aus einem zyklischen 2^n-Bit Linksshifter, einem n-Bit breiten bitweisen Inverter und einem n-Incrementer. Diesen Incrementer kann man jedoch einsparen.

    Konstruiere einen zyklischen 2^n-Bit Links-/Rechts-Shifter, der nur aus einem zyklischen 2^n-Bit Linksshifter, einem n-Bit breiten bitweisen Inverter und einem 2^n-Bit Multiplexer besteht.

14. **Aufgabe** **(2 + 4 Punkte)**
    In der ALU gibt es keine direkte Hardwareunterstützung für bitweise Negation. Um einen Registerinhalt bitweise zu negieren kann man jedoch andere Befehle "zweckentfremden".

    Es gelten folgende Voraussetzungen:
    - $[GPR(00000)] = 0$.
    - $[GPR(00001)] = -1$.

    (a) Gib **einen** R-Type-Befehl an, der den Inhalt von GPR(00010) bitweise negiert und in Register GPR(00011) abspeichert.
    (b) Gib eine zweite Lösungsmöglichkeit an.

15. **Aufgabe** **(5 Punkte)**
    Angenommen du willst in Register GPR(00010) eine feste Konstante $a \in \{0,1\}^{32}$ abspeichern. Leider gibt es keine Möglichkeit, diese direkt mit einem Befehl in das Register zu schreiben, da Immediate-Konstanten nur maximal 16 Bit lang sein können.

    Gib eine Folge von 2 Befehlen an, die a in Register GPR(00010) speichert.

    Tipp: Du musst 2 unterschiedliche Programme schreiben. Die Fallunterscheidung, die bestimmt, welches deiner Programme auszuführen ist, ist in Abhängigkeit von a anzugeben.

Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl. Inform. M. Klein
Dipl. Inform. T. In der Rieden

1. Teilklausur Informatik II

(Datum: 21.06.2003)

IR[31 : 26]		IR[5 : 0]		Mnem.	Effect
Arithmetic, Logical Operation					
000000	0x00	100000	0x20	add	RD=RS1+RS2
000000	0x00	100001	0x21	addu	RD=RS1+RS2 (no overfl.)
000000	0x00	100010	0x22	sub	RD=RS1-RS2
000000	0x00	100011	0x23	subu	RD=RS1-RS2 (no overfl.)
000000	0x00	100100	0x24	and	RD=RS1 $\wedge$ RS2
000000	0x00	100101	0x25	or	RD=RS1 $\vee$ RS2
000000	0x00	100110	0x26	xor	RD=RS1 $\oplus$ RS2
000000	0x00	100111	0x27	lhg	RD=RS2[15:0] 0^{16}
Test Set Operation					
000000	0x00	101000	0x28	clr	RD=(false ? 1 : 0)
000000	0x00	101001	0x29	sgr	RD=(RS1 > RS2 ? 1 : 0)
000000	0x00	101010	0x2a	seq	RD=(RS1 = RS2 ? 1 : 0)
000000	0x00	101011	0x2b	sge	RD=(RS1 $\geq$ RS2 ? 1 : 0)
000000	0x00	101100	0x2c	sls	RD=(RS1 < RS2 ? 1 : 0)
000000	0x00	101101	0x2d	sne	RD=(RS1 $\neq$ RS2 ? 1 : 0)
000000	0x00	101110	0x2e	sle	RD=(RS1 $\leq$ RS2 ? 1 : 0)
000000	0x00	101111	0x2f	set	RD=(true ? 1 : 0)

Tabelle 1: R-type instruction layout

IR[31 : 26]		Mnem.	d	Effect
Arithmetic, Logical Operation				
001000	0x08	addi		RD=RS1 + imm
001001	0x09	addiu		RD=RS1 + imm (no overflow)
001010	0x0a	subi		RD=RS1 - imm
001011	0x0b	subiu		RD=RS1 - imm (no overflow)
001100	0x0c	andi		RD=RS1 $\wedge$ imm
001101	0x0d	ori		RD=RS1 $\vee$ imm
001110	0x0e	xori		RD=RS1 $\oplus$ imm
001111	0x0f	lhgi		RD=imm 0^{16}
Test Set Operation				
011000	0x18	clri		RD=(false ? 1 : 0)
011001	0x19	sgri		RD=(RS1 > imm ? 1 : 0)
011010	0x1a	seqi		RD=(RS1 = imm ? 1 : 0)
011011	0x1b	sgei		RD=(RS1 $\geq$ imm ? 1 : 0)
011100	0x1c	slsi		RD=(RS1 < imm ? 1 : 0)
011101	0x1d	snei		RD=(RS1 $\neq$ imm ? 1 : 0)
011110	0x1e	slei		RD=(RS1 $\leq$ imm ? 1 : 0)
011111	0x1f	seti		RD=(true ? 1 : 0)

Tabelle 2: I-type instruction layout