Rechnersysteme und -netze: Probeklausur

Winter 2019

Die Probeklausur besteht aus drei Teilen, für die je ca. 45 Minuten zur Verfügung stehen und die je Inhalte zu circa einem Drittel der Vorlesung abfragen.
Als Hilfsmittel ist neben Stiften und Papier die Reference Card erlaubt.

Name	Immatrikulationsnummer

Aufgabe 1	Aufgabe 2	Aufgabe 3	Bonusaufgabe	Summe (Teil 1)
/18	/16	/12	/ 2	/46(+2)

Aufgabe 1 Boolesche Operationen (4 + 6 + 8 Punkte)

Wir betrachten die drei binären Booleschen Operationen NAND(a, b) (auch $a \mid b$, "Gegenteil des logischen Und"), NOR(a, b) (auch $a \downarrow b$, "Gegenteil des logischen Oder"), und XNOR(a, b) (auch $\neg(a \oplus b)$, "Gegenteil des exklusiven Oder"). Lösen Sie die folgenden Teilaufgaben mit Hilfe Boolescher Formeln, **nicht** durch Schaltpläne mit (Logik-)Gattern! (Sie dürfen eine Negation auch durch einen Querstrich über einen Variablennamen bzw. einer Teilformel darstellen, aber schreiben Sie bitte die Konjunktion als $\wedge$ und die Disjunktion als $\vee$.)

a) Stellen Sie die Operatoren NAND, NOR, und XNOR mit Hilfe der Operatoren Konjunktion $\wedge$, Disjunktion $\vee$ und Negation $\neg$ dar!

b) Zeigen Sie, wie der XNOR-Operator durch den NAND-Operator ausgedrückt werden kann!

c) Zeigen Sie, daß die Operationenmenge O, die nur aus den Operationen Implikation $\rightarrow$ (siehe Wahrheitstafel rechts) und Negation $\neg$ besteht (d.h., $O = \{\rightarrow, \neg\}$), vollständig ist!

a	b	$a \rightarrow b$
0	0	1
0	1	1
1	0	0
1	1	1

Aufgabe 2 Minimierung Boolescher Funktionen (6 + 6 + 4 Punkte)

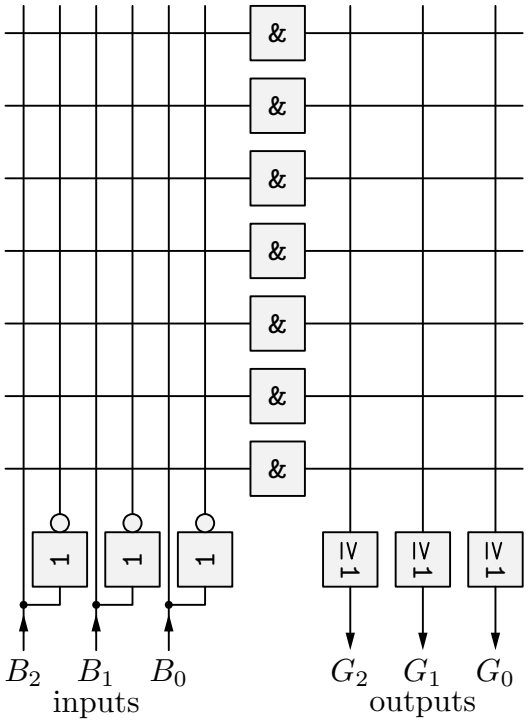
Gegeben sei die folgende Übersetzungstabelle eines dreistelligen Binärkodes $[B_2B_1B_0]$ in einen dreistelligen Gray-Kode $[G_2G_1G_0]$:

Binärkode			Gray-Kode		
B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

a) Fassen Sie die Spalten des Gray-Kodes als Boolesche Funktionen der Spalten des Binärkodes auf und geben Sie für G_0 die disjunktive Normalform an! Wäre es sinnvoll, stattdessen die konjunktive Normalform zu verwenden? Begründen Sie Ihre Antwort!

b) Minimieren Sie die Boolesche Funktion $G_1 = f(B_2, B_1, B_0)$ unter Bezugnahme auf die Einsen in der Ausgabe **entweder** mit Hilfe eines Karnaugh–Veitch-Diagramms **oder** mit Hilfe des Quine–McCluskey-Algorithmus!

c) Implementieren Sie die Berechnung des Gray-Kodes (alle drei Spalten) mit Hilfe eines (programmierbaren) Logikarrays! Verwenden Sie dazu das unten gezeigte Schema!



Aufgabe 3 Halbsubtrahierer $(4 + 4 + 4$ Punkte)

Bei der Subtraktion zweier Binärziffern (Bits) entsteht u.U. ein negativer Übertrag, den wir hier "Borger" nennen wollen.

a) Geben Sie die Wertetabelle für den Halbsubtrahierer, der zwei Binärziffern voneinander abzieht, mit den Ausgängen Differenz D und Borger B an!

b) Geben Sie die Booleschen Funktionen für Differenz D und Borger B an!

c) Zeichnen Sie, unter Verwendung der Ihnen bekannten (Logik-)Gatter, eine Schaltung von (Logik-)Gattern, die den Halbsubtrahierer implementiert!

Bonusaufgabe Variablennamen (2 Punkte)

Wie heißen die beiden Zeichen φ und ψ?

Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Bonusaufgabe	Summe (Teil 2)
/ 6	/ 20	/ 14	/ 10	/ 6	/ 50(+6)

Aufgabe 1 Addieren von Binärzahlen (2 + 4 Punkte)

a) Berechnen Sie die Summe der beiden vorzeichenlosen Binärzahlen $x = \texttt{1010011}_2$ und $y = \texttt{1011011}_2$.

b) Berechnen Sie die Differenz $x - y$ der oben gegebenen Zahlen!

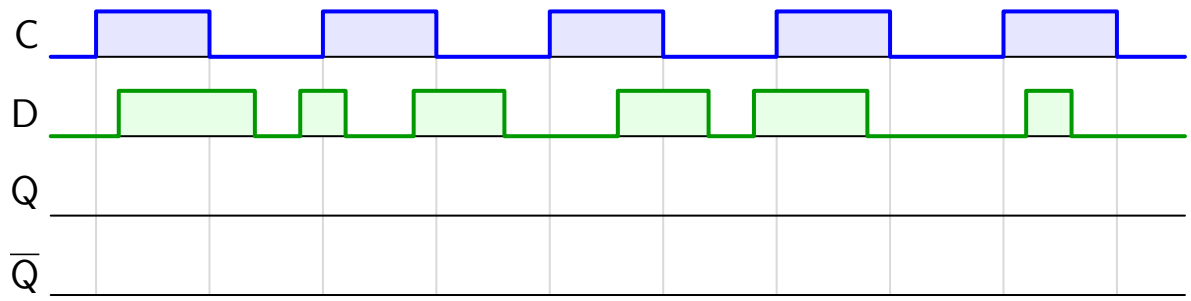
Aufgabe 2 Multiplizieren (14 + 6 Punkte)

a) Verifizieren Sie die Rechnung $7_{10} \times (-2)_{10} = -14_{10}$ im Binärsystem!

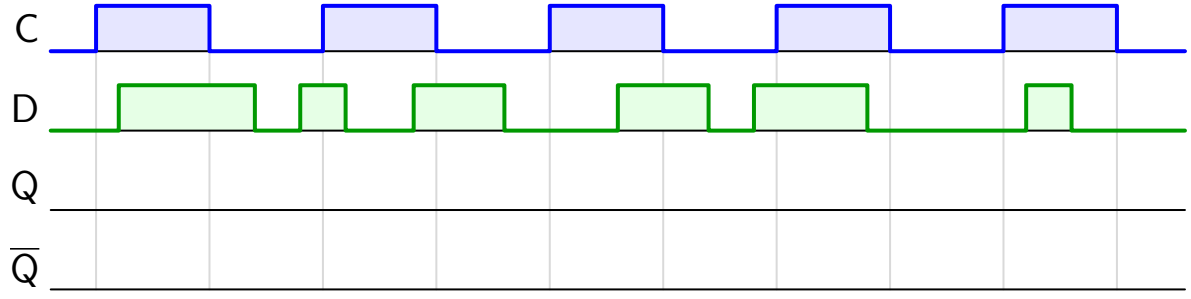
b) Berechnen Sie im Dezimalsystem $9990_{10} \times 88_{10}$ unter Verwendung einer „Dezimalversion“ von Booths Algorithmus!

Aufgabe 3 Signalverlauf Master-Slave-D-Riegel/-Flipflop (5 + 5 + 4 Punkte)

a) Bestimmen Sie für die unten gezeigten Eingaben C (Takt) und D (Daten) die Ausgaben Q und $\overline{\text{Q}}$ für einen Master-Slave-D-**Riegel**!



b) Bestimmen Sie für die unten gezeigten Eingaben C (Takt) und D (Daten) die Ausgaben Q und $\overline{\text{Q}}$ für ein Master-Slave-D-**Flipflop**!



c) Zeichnen Sie das Schaltzeichen für einen Master-Slave-D-**Riegel**!

Aufgabe 4 Schieberegister (8 + 2 Punkte)

a) Konstruieren Sie aus D-Flipflops (als Schaltzeichen, **nicht** als Gatterschaltung) ein 4-Bit-Schieberegister!

b) Geben Sie eine Einsatzmöglichkeit eines Schieberegisters an!

Bonusaufgabe Hack-Architektur (2 + 4 Punkte)

a) Wodurch unterscheiden sich die Harvard- und die von-Neumann-Architektur?

b) Geben Sie vier verschiedene Belegungen der Steuerbits der Hack-ALU an, die als Ausgabe -1 erzeugen (und zwar unabängig von den Eingaben `x` und `y` der ALU)!

Zur Erinnerung: Die Steuerbits `zx`, `nx`, `zy`, `ny`, `f` und `no` der Hack-ALU kodieren die auszuführende Operation:

`zx`	Setze Eingabe `x` $= 0$.
`nx`	Bilde Einerkomplement der Eingabe `x`.
`zy`	Setze Eingabe `y` $= 0$.
`ny`	Bilde Einerkomplement der Eingabe `y`.
`f`	Wählt zwischen Addition (1) und bitweiser Konjunktion (0) als Operation.
`no`	Bilde Einerkomplement der Ausgabe `out`.

Aufgabe 1	Aufgabe 2	Bonusaufgabe	Summe (Teil 3)
/ 26	/ 24	/ 6	/ 50(+6)

Aufgabe 1 Hack-Assemblersprache ($4 + 4 + 8 + 6 + 4$ Punkte)

```
  @a
  M=1
  @b
  M=0
  @R0
  D=M
  @i
  M=D
(LOOP)
  @i
  MD=M-1
  @STORE
  D;JLT
  @a
  D=M
  @b
  MD=D+M
  @a
  M=D-M
  @LOOP
  0;JMP
(STORE)
  @b
  D=M
  @R1
  M=D
(END)
  @END
  0;JMP
```

Gegeben sei das links gezeigte Programm in Hack-Assemblersprache.

a) Welchen Inhalt hat die Symboltabelle des Hack-Assemblers am Ende der Übersetzung dieses Programms? (Geben Sie von den vordefinierten Symbolen nur die an, die auch im links gezeigten Programm auftreten!)

b) Übersetzen Sie die beiden Hack-Assemblerbefehle „`MD=M-1`“ und „`D;JLT`“, die im links gezeigten Programm auftreten, in Hack-Maschinensprache!

c) Das links gezeigte Programm werde ausgeführt, wobei zu Beginn im virtuellen Register `R0` der Wert 6 stehen möge. Füllen Sie die folgende Iterationstabelle aus (es werden ggf. nicht alle Zeilen benötigt):

Zeitpunkt/Variablen	`i`	`a`	`b`
Initialisierung			
1. Durchlauf			
2. Durchlauf			
3. Durchlauf			
4. Durchlauf			
5. Durchlauf			
6. Durchlauf			
7. Durchlauf			
8. Durchlauf			

Die Zeitpunkte beziehen sich auf das erste bzw. jedes weitere Erreichen der Markierung `(LOOP)` (bzw. des durch sie markierten Befehls).
Welcher Wert steht am Ende der Ausführung im virtuellen Register `R1`?

d) Erläutern Sie die Berechnung des Programms, indem Sie eine Funktion `int f (int i)` in Pseudocode (oder Jack oder Java oder ...) angeben, die die Berechnung auf äquivalente Weise durchführt!

Welche Funktion berechnet das Programm allgemein?
(Beschreibung in einem Satz!)

e) Nennen Sie zwei Vor- und zwei Nachteile von Assemblersprache!

Aufgabe 2 Virtuelle Maschine ($6 + 8 + 6 + 4$ Punkte)

```
function Main.f 1
label LOOP
  push argument 0
  push constant 0
  lt
  if-goto STORE
  push local 0
  push constant 1
  add
  pop local 0
  push argument 0
  push argument 0
  add
  push constant 1
  add
  pop argument 0
  goto LOOP
label STORE
  push local 0
  return
```

Gegeben sei die links gezeigte Funktion in der Sprache der virtuellen Maschine des Hack-Systems.

a) Übersetzen Sie die Anweisung „`pop argument 0`“, die in der links gezeigten Funktion auftritt, in Hack-Assemblersprache!

b) Die links gezeigte Funktion werde durch die Anweisungen

```
push constant 2048
call Main.f 1
```

aufgerufen. Füllen Sie die folgende Iterationstabelle aus (es werden ggf. nicht alle Zeilen benötigt):

Zeitpunkt/Variablen	`argument 0`	`local 0`
Initialisierung		
1. Durchlauf		
2. Durchlauf		
3. Durchlauf		
4. Durchlauf		
5. Durchlauf		
6. Durchlauf		

Die Zeitpunkte beziehen sich auf das erste bzw. jedes weitere Erreichen der Markierung `LOOP`. Welchen Wert gibt die Funktion `Main.f` bei diesem Aufruf zurück?

(Hinweis: Es ist u.U. vorteilhaft, mit einer Binär- oder Hexadezimaldarstellung der Werte zu arbeiten. Dies erleichtert auch das Lösen der folgenden Teilaufgabe.)

c) Erläutern Sie die Berechnung der Funktion, indem Sie eine Funktion `int f (int i)` in Pseudocode (oder Jack oder Java oder ...) angeben, die die Berechnung auf äquivalente Weise durchführt!

Was berechnet die Funktion `Main.f` allgemein?
(Beschreibung in einem Satz!)

d) Nennen Sie zwei Vor- und zwei Nachteile von virtuellen Maschinen!

Bonusaufgabe ($2 + 2 + 2$ Punkte)

a) Wie lauten Titel und Untertitel des Buches, dem die Vorlesung „Rechnersysteme und -netze“ folgt?

b) Wofür stehen in der Sprache der virtuellen Maschine das n und das k in den Anweisungen

`function` *fnname* n und `call` *fnname* k ?

(Antworten Sie mit „Das n steht für ...“ und „Das k steht für ...“!)

c) Zeichnen Sie einen Parse-Baum für den folgenden Jack-Ausdruck:

```
2+min(x+3*y,0)
```