



Prüfungsklausur Digitaltechnik und Rechnersysteme

für die Bachelor-Prüfung im Sommersemester 2020

.....
Name Vorname Matrikelnummer

☐ letzter Prüfungsversuch

Hinweise für alle Aufgaben

Lassen Sie bitte die Aufgabenblätter zusammengeheftet und führen Sie die Rechnungen auf dem dafür vorgesehenen Platz im Aufgabentext durch. Falls dieser Platz nicht ausreicht, können Sie zusätzliche Blätter erhalten (bitte dann entsprechend referenzieren).

Zugelassene Hilfsmittel

- Handgeschriebene Formelsammlung max. ein DIN A4 Blatt, beidseitig
- Referenzblatt »MIPS32 Befehlssatz (Auszug) und Register«
- nicht-programmierbarer Taschenrechner

Bearbeitungszeit: 90 Minuten

Maximalpunktzahl: 90

Erreichte Punkte:

1	2	3	4	5	6	7	Σ

Aufgabe 1: Verständnisfragen (10 Punkte)

Alle Aufgabenteile können unabhängig voneinander gelöst werden!

a) Gegeben sind die folgenden 4 Bit Binärzahlen. Geben Sie deren Dezimaldarstellung an, wenn diese jeweils im Zweierkomplement- bzw. Vorzeichen-Betrag-Zahlen interpretiert werden.

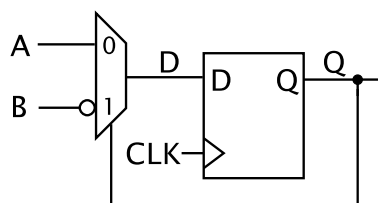
Binärdarstellung	Dezimalwert bei Kodierung im	
	Zweierkomplement	Vorzeichen-Betrag
0101		
1110		

b) Gegeben ist ein Code mit 4 Symbolen sowie deren Wahrscheinlichkeiten:

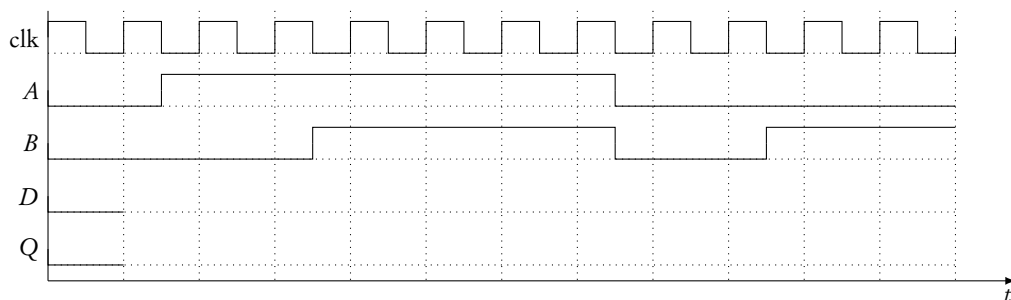
Symbol	Wahrscheinlichkeit
A	0,1
B	0,2
C	0,3
D	0,4

Zeichnen Sie einen Binärbaum für eine Huffman-Codierung und geben Sie eine mögliche Codierung an.

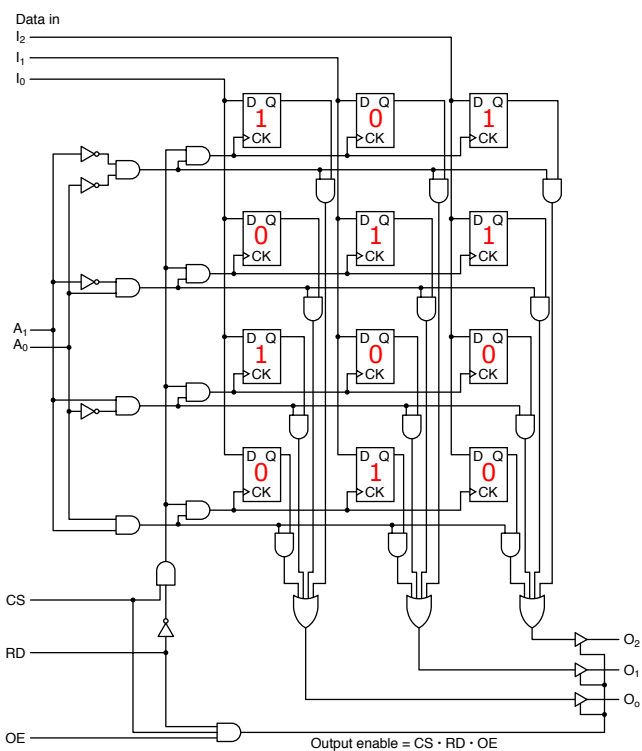
c) Gegeben ist die folgende Schaltung eines synchronen Automaten:



Ergänzen Sie das unten angegebene Timing-Diagramm:



d) Gegeben ist die Schaltung eines 3×4 Bit Speichers, die gespeicherten Inhalte der Flip-Flops sind rot dargestellt:



Wie lautet die Ausgabe O wenn die Adresse $A = 10_2$ im Lesebetrieb ($RD = 1$) mit $CS = OE = 1$ angelegt wird?

O =

Aufgabe 2: Boolesche Algebra und Wahrheitstabellen (10 Punkte)

Alle Aufgabenteile können unabhängig voneinander gelöst werden!

- a) Vereinfachen Sie die folgenden drei Ausdrücke mit Hilfe der Booleschen Algebra und geben Sie das Ergebnis in disjunktiver Normalform (DNF) an.

$$xy + x(wz + w\bar{z}) =$$

$$\overline{\overline{\overline{a+b+c+d+e}}} =$$

$$(x + \bar{y} + \bar{z})(\bar{x} + \bar{z}) =$$

b) Gegeben sei die Wahrheitstabelle der Funktion $f(a, b, c,)$:

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

b1) Ermitteln Sie die kanonische disjunktive Normalform (KDNF) von $f(a, b, c)$.

$f =$

b2) Ermitteln Sie die kanonische konjunktive Normalform (KKNF) von $f(a, b, c)$.

$f =$

b3) Vereinfachen Sie die Funktion mit Hilfe der Booleschen Algebra soweit wie möglich und geben das Ergebnis in disjunktiver Normalform (DNF) an.

$f =$

b4) Geben Sie die Funktion einer NAND-Realisierung der Schaltung an (keine minimierte Form nötig).

$f =$

Aufgabe 3: KV Minimierung (20 Punkte)

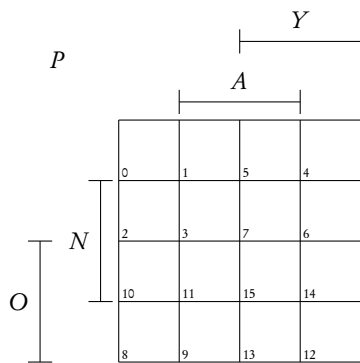
Gegeben sind die folgenden Booleschen Funktionen:

$$P = (Y + \overline{O}) \cdot \overline{Y} \cdot \overline{O} \cdot \overline{A} + N$$

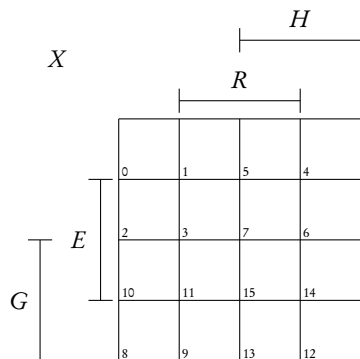
$$X = R \left(\overline{\overline{G}H \cdot \overline{E}} + H(E + G) \right)$$

$$Q = (\overline{Y} + O) \cdot (Y + N)$$

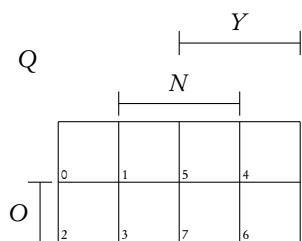
- Tragen Sie die folgenden Funktionen in die zugehörigen Karnaugh-Veitch-Diagramme ein.
- Markieren Sie alle Primterme und geben deren Typ an.
- Ermitteln Sie die minimierten Funktionen in disjunktiver Normalform (DNF).



P =



X =



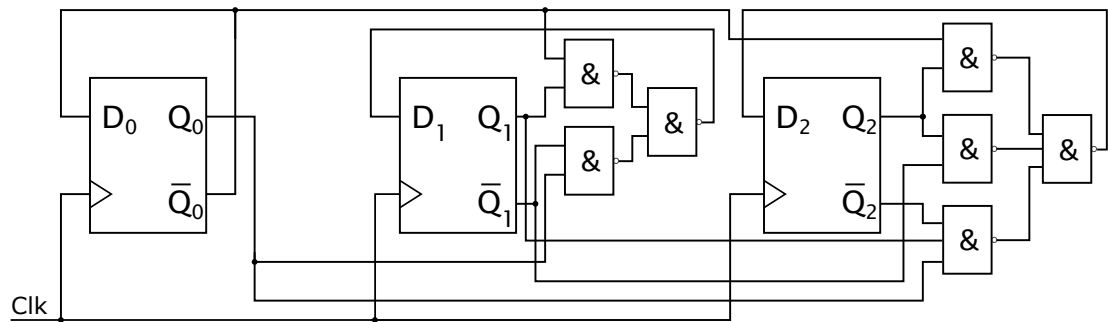
Q =

d) Zeichnen Sie das Schaltbild der minimierten Funktion Q .

e) In welchen der drei Funktionen P , X und Q können strukturelle Hazards auftreten? Geben Sie eine Lösung ohne strukturelle Hazards für diese Funktionen in disjunktiver Normalform (DNF) an.

Aufgabe 4: Automatenanalyse (17 Punkte)

Gegeben ist folgende Schaltung eines Automaten:



a) Geben Sie die folgenden Zustandsübergangsfunktionen in Abhängigkeit der Flipflop-Ausgänge Q_0 , Q_1 und Q_2 als DNF an.

$D_0 =$

$D_1 =$

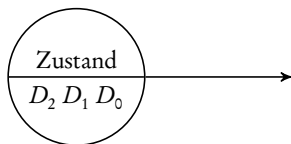
$D_2 =$

b) Geben Sie die Zustandsübergangstabelle an.

Q_2	Q_1	Q_0	D_2	D_1	D_0
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

c) Zeichnen Sie das Zustandsdiagramm des Automaten. Verwenden Sie die nachfolgende Notation.

Notation:



d) Welche Funktion realisiert der Automat?

.....

.....

Aufgabe 5: Taschenrechner (10 Punkte)

Es soll ein Automat entwickelt werden, welcher unzulässige Eingaben eines Taschenrechners erkennt und ein entsprechendes Fehlersignal erzeugt. Der Taschenrechner verfügt über einen Ziffernblock, Operatortasten (+, −, *, ÷) und eine Eingabetaste (=) zur Ausführung der Berechnung.

Die Betätigung einer beliebigen Ziffer (0...9) des Ziffernblocks wird mit dem Signal $z = 1$ signalisiert. Bei Betätigung einer der Operatortasten wird das Signal $o = 1$ ausgegeben und die Betätigung der Eingabetaste wird mit $e = 1$ signalisiert. Haben alle Signale z , o und e den Wert '0', so ist keine Taste gedrückt. Der Wert jedes Tastensignals wird bei Tastendruck für einen Takt gesetzt und anschließend zurückgesetzt, auch wenn die Taste weiter gedrückt wird. Zudem ist ausgeschlossen, dass mehrere Tasten gleichzeitig gedrückt werden können.

Der Automat soll als Moore-Automat realisiert werden und ein Gültigkeitssignal v (*valid*) ausgeben, solange die Eingabe gültig ist. Sobald eine Eingabe *ungültig* ist, soll dies mit $v = 0$ signalisiert werden, ansonsten soll immer $v = 1$ ausgegeben werden. Der Taschenrechner zeigt in diesem Fall den Fehler an und wartet auf die Betätigung einer beliebigen Taste, woraufhin eine erneute Eingabe erfolgen kann.

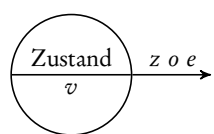
Eine gültige Eingabe hat *immer* den folgenden Ablauf:

1. Eingabe einer Zahl durch ein- oder mehrfaches Betätigen einer Ziffern-Taste
2. Eingabe der Operation (+, −, *, ÷) durch *einfaches* Betätigen einer Operatortaste
3. Eingabe einer weiteren Zahl durch ein- oder mehrfaches Betätigen einer Ziffern-Taste
4. Ausführung der Berechnung durch Eingabetaste (=) *oder* die Eingabe weiterer Operationen durch Sprung nach Schritt 2

Jede Abweichung von diesem Ablauf muss mit Ungültigkeit ($v = 0$) signalisiert werden. Beispielsweise wäre die Eingabe »123+321*321=« gültig, während die Eingaben »123+*«, »123+321*«, »123=« oder »+« zu einem Fehler führt.

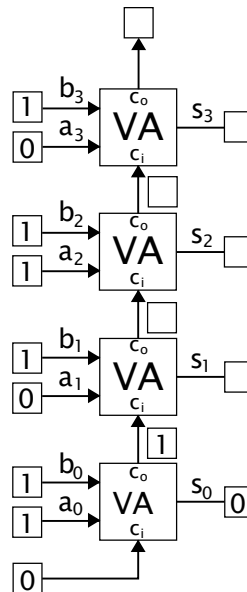
Geben Sie ein geeignetes Zustandsdiagramm für den gesuchten Moore-Automaten an. Geben Sie den Zuständen sinnvolle Namen. Verwenden Sie *don't cares*, um nur sinnvolle Kombinationen der Eingangssignale anzugeben. Verwenden Sie für Ihren Moore-Automaten die nachfolgende Notation.

Notation:



Aufgabe 6: Addierer (8 Punkte)

a) Gegeben ist die Anordnung eines 4-Bit Ripple-Carry Addierers. Ergänzen Sie die fehlenden Ausgangswerte in den dafür vorgesehenen Kästchen.



b) Vervollständigen Sie die unten angegebene Wahrheitstabelle eines Volladdierers mit den Operanden a und b , dem Übertrags-Eingang (*carry-in*) c_i sowie dem Summen-Ausgang s und dem Übertrags-Ausgang (*carry-out*) $c_o = c_{i+1}$:

a	b	c_i	c_o	s
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

c) Welchen wesentlichen Vor- und Nachteil hat ein Carry-Lookahead Addierer gegenüber einem Ripple-Carry Addierer?

Vorteil:

Nachteil:

Aufgabe 7: MIPS Assembler (15 Punkte)

a) Erläutern Sie die Besonderheit einer Akkumulator-Maschine

.....

.....

b) Erläutern Sie den Hauptunterschied zwischen den Befehlssatzarchitekturen CISC und RISC.

.....

.....

c) Gegeben ist das folgende Assembler-Programm:

```
ori $s0,$zero,6
ori $s1,$zero,7
mul $s2,$s0,$s1
andi $s2,$s2,0x0F
```

Wie lautet der Inhalt von Register \$s2 nach Ausführung des Programms?

\$s2

d) Das folgende Programm (gegeben in C bzw. Java Syntax) berechnet die Fakultät f von n , also $f = n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$.

```
f=1;
n++;
for(int i=1; i != n; i++)
{
    f = f * i;
}
```

Entwerfen Sie ein MIPS-Assembler-Programm welches diese Funktionalität umsetzt. Gehen Sie davon aus, dass die Variablen n und f als 32 Bit Integer Variablen deklariert sind welche in den Registern \$s0 und \$s1 gespeichert sind.