Prof. Dr.-Ing. Jürgen Teich
Lehrstuhl für Informatik 12
(Hardware-Software-Co-Design)
Universität Erlangen-Nürnberg

Klausur
Grundlagen der Technischen Informatik

02. April 2014

Name	
Matrikelnummer	
Studienrichtung	

Aufgabe	1	2	3	4	5	Σ
max. Punkte	15	15	20	15	15	80
erreichte Punkte						
Note						

Organisatorische Hinweise

Bitte sorgfältig lesen und die Kenntnisnahme durch Unterschrift bestätigen

---

1. Bitte legen Sie Ihren Studentenausweis bereit.
2. Als Hilfsmittel sind nur Schreibmaterialien und ein beidseitig handbeschriebenes DIN A4-Blatt zugelassen.
3. Schmierpapier wird nicht abgegeben und auch nicht korrigiert.
4. Sie können bei der Aufsicht zusätzliche Bearbeitungsblätter anfordern.
5. Unleserliches wird nicht bewertet.

Erklärung

1. Im Falle einer während der Prüfung auftretenden Prüfungsunfähigkeit zeige ich dies sofort der Aufsicht an und befolge deren Anweisungen. Ich weiß, dass ich die volle Beweislast trage. Ich lasse mir das Formular des Prüfungsamts, das für diese Fälle vorgesehen ist, aushändigen und verfahre nach den dort niedergelegten Richtlinien.
2. Ich weiß, dass im Falle des Täuschungsversuchs oder der Benutzung unerlaubter Hilfsmittel („Unterschleif") der Prüfungsausschuss die Entscheidung treffen kann, die betroffene Prüfungsleistung als mit „nicht ausreichend" bewertet gelten zu lassen.
3. Ich habe die obigen Hinweise zur Kenntnis genommen.

Erlangen, den 02. April 2014 ....

Unterschrift

Einwilligung

Ich bin damit einverstanden, dass mein vorläufiges Ergebnis anonymisiert, jedoch unter Angabe der Matrikelnummer, am Mitteilungsbrett und auf der Webseite des Lehrstuhls für Informatik 12 veröffentlicht wird.

Die Bekanntgabe des vorläufigen Ergebnisses begründet keinen Rechtsanspruch.

Die Bekanntgabe des endgültigen Ergebnisses erfolgt ausschließlich durch das Prüfungsamt.

Erlangen, den 02. April 2014 ....

Unterschrift

Aufgabe 1 (Zahlensysteme) (15 Punkte)

a) Wie lautet der Wertebereich einer n Bit Binärzahl in Zweierkomplement-Darstellung? (1 Punkt)

b) Wie lautet der Wertebereich einer mit n Bit darstellbaren vorzeichenlosen Binärzahl? (1 Punkt)

c) Konvertieren Sie die Dezimalzahl 205_{10} in eine vorzeichenlose Binärzahl. (1 Punkt)

d) Konvertieren Sie die Hexadezimalzahl $FCE2_{16}$ in das Oktalsystem. (1 Punkt)

e) In dieser Aufgabe werden 11 Bit lange Gleitkommazahlen betrachtet. Diese werden analog zum IEEE-Format gebildet. Das Format der Gleitkommazahl sieht dabei wie folgt aus: *Vorzeichen (1 Bit)*, *Exponent (4 Bit)*, *Mantisse (6 Bit)*

V	E	M
10	9 6	5 0

Wandeln Sie die in diesem Format dargestellte Gleitkommazahl 11010000111 in das Dezimalsystem um. (3 Punkte)

f) Beantworten Sie folgende Auswahlfragen. Jede richtige Antwort ergibt einen Punkt, jede falsche Antwort führt zu einem Punktabzug, nicht beantwortete Fragen werden nicht gewertet, weniger als null Punkte sind nicht möglich. (4 Punkte)

1. Mit UND- und ODER-Gattern allein lässt sich jede beliebige Schaltfunktion darstellen.
   ☐ wahr ☐ falsch
2. Mit ODER- und ANTIVALENZ-Gattern (XOR) allein lässt sich jede beliebige Schaltfunktion darstellen. ☐ wahr ☐ falsch
3. Die Addition zweier Zahlen im Zweierkomplement mit 32 Bit ist aufwändiger als die Addition zweier vorzeichenloser Binärzahlen mit 32 Bit. ☐ wahr ☐ falsch
4. Eine Multiplikationseinheit für zwei vorzeichenlose 32 Bit breite Binärzahlen a und b mit einem 32 Bit breiten Ergebnis $s = a \cdot b$ berechnet auch für die Multiplikation zweier 32-Bit-Zahlen im Zweierkomplement ein richtiges Ergebnis, wenn dieses Ergebnis im Wertebereich der 32 Bit Zweierkomplement-Darstellung liegt. ☐ wahr ☐ falsch

g) Gegeben sei eine Recheneinheit (8 Bit ALU) zur Berechnung von $r = x \operatorname{op} y$ mit den Operationen $\operatorname{op} \in \{+, -, >, \geq, <, \leq, =, \neq\}$ sowie den Booleschen Verknüpfungen $\operatorname{op} \in \{\wedge, \vee$ und $\neg\}$. Die Operanden x und y sowie das Ergebnis r sind dabei 8 Bit breite Binärzahlen in Zweierkomplement-Darstellung. Ein Überlauf tritt bei einer Operation auf, wenn das Ergebnis der Operation außerhalb des Wertebereichs einer 8 Bit breiten Binärzahl in Zweierkomplement-Darstellung liegt.

Geben Sie einen Test an, welcher zu wahr evaluiert, wenn bei der Subtraktion $d = a - b$ in der Recheneinheit ein Überlauf auftritt. Dem Test stehen ausschließlich die obigen Operationen der Recheneinheit, Konstanten, darstellbar als 8 Bit breite Binärzahlen in Zweierkomplement-Darstellung, die zwei Operanden a und b, das Ergebnis d, sowie beliebige Zwischenergebnisse zur Verfügung. (4 Punkte)

Ein solcher Test könnte beispielsweise wie folgt aussehen: $d \geq 7 \vee a > 2 \wedge (b \geq a)$

Aufgabe 2 (Minimierung von Schaltfunktionen) (15 Punkte)

a) Geben Sie die Schaltfunktion $f_1(x_3,x_2,x_1,x_0) = \overline{(x_0+x_3)\cdot x_2} + x_1$ als Gatterschaltung an. Dabei stehen Ihnen nur UND-, ODER-Gatter und Inverter zur Verfügung. (2 Punkte)

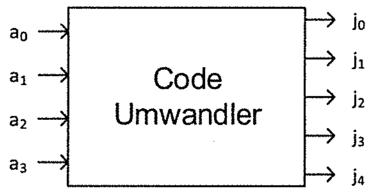
b) Realisieren Sie die Schaltfunktion $f_1(x_3,x_2,x_1,x_0)$ aus Teilaufgabe 2a) als CMOS-Schaltung mit möglichst wenig Transistoren. Dabei stehen Ihnen alle Eingänge nur in der nicht invertierten Form zur Verfügung. (4 Punkte)

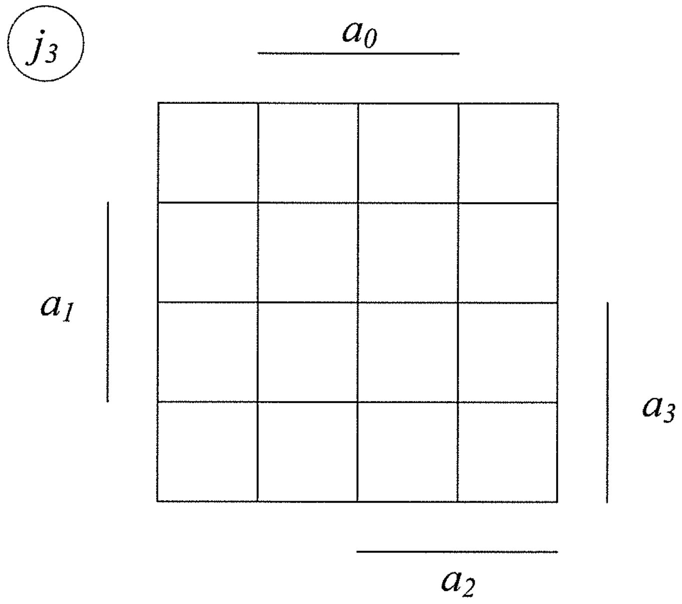
VDD

GND

c) Es soll eine Schaltung zur Umwandlung des Aiken-Codes in den Johnson-Code entwickelt werden. Dazu sei folgende Code-Tabelle gegeben. Bestimmen Sie mit Hilfe eines Symmetriediagramms alle Prim- und Kernimplikanten von j_3. (4 Punkte)

Aiken-Code				Johnson-Code				
a_3	a_2	a_1	a_0	j_4	j_3	j_2	j_1	j_0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	1	1
0	1	0	0	0	1	1	1	1
1	0	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	0
1	1	0	1	1	1	1	0	0
1	1	1	0	1	1	0	0	0
1	1	1	1	1	0	0	0	0





Primimplikanten: Kernimplikanten:

d) Gegeben sei folgendes Symmetriediagramm einer Schaltfunktion $f_2(x_3, x_2, x_1, x_0)$. Bestimmen Sie daraus zuerst eine Nullblocküberdeckung und ermitteln Sie anschließend alle Disjunktiven Minimalformen (DMFs) mittels des Nelson/Petrick-Verfahrens. Die Kosten eines Primimplikanten entsprechen hierbei der Anzahl seiner Literale. Geben Sie alle Zwischenschritte an (Überdeckungstabelle, Petrick-Ausdruck etc.). (5 Punkte)

		x_0	x_0		
	-	-	-	0	
x_1	1	-	1	1	
x_1	0	-	-	1	x_3
	0	0	1	0	x_3
			x_2	x_2	

Aufgabe 3 (Automaten und Flipflops) (20 Punkte)

a) Im Folgenden soll eine Jalousiesteuerung entworfen werden. Diese ist wie folgt spezifiziert: Eine Jalousiesteuerung wird dazu eingesetzt, um die Raumtemperatur in einem bestimmten Temperaturbereich zu halten. Hierzu verfügt das System über vier verschiedene Eingaben, die jedoch nicht gleichzeitig anliegen können. Sobald der eingestellte Höchstwert für die Raumtemperatur erreicht wurde (Eingabe *HT*), soll ein Motor angesteuert werden, der die Jalousie herunter fährt (Ausgabe *MDown*). Die Eingabe *LT* hingegen repräsentiert das Erreichen des eingestellten Minimalwerts für die Raumtemperatur und veranlasst somit das sofortige Herauffahren der Jalousie. Wurde die Jalousie komplett herunter- bzw. heraufgefahren, wird dies durch die Eingabe *STOP* signalisiert, welche das Stoppen des Motors (Ausgabe *MOff*) bewirkt. Zusätzlich besteht die Möglichkeit, in einen manuellen Modus zu wechseln und hierbei die Motoransteuerung zu entkoppeln (Ausgabe *MManual*). Liegt das zugehörige Eingangssignal *MAN* nicht mehr an, so soll die Steuerung ein Hochfahren der Jalousie veranlassen und anschließend den Regelbetrieb fortsetzen.

Die Ein- und Ausgaben sind dabei wie folgt durch binäre Variablen codiert:

Eingabe	i_1	i_0
LT	0	0
HT	0	1
STOP	1	0
MAN	1	1

Eingaben des Automaten

Ausgabe	o_1	o_0
MOff	0	0
MDown	0	1
MUp	1	0
MManual	1	1

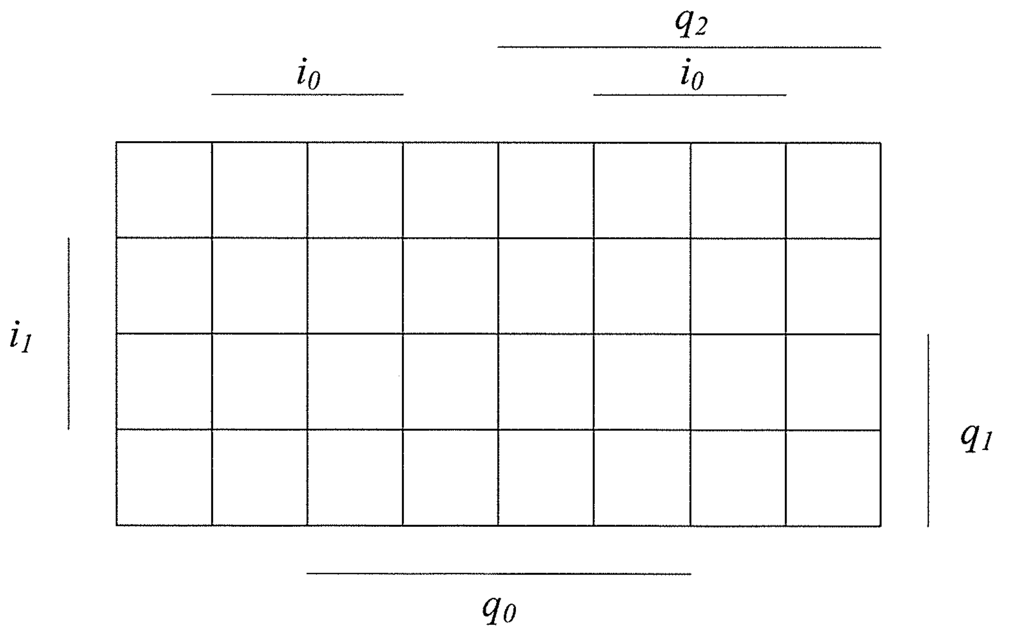
Ausgaben des Automaten

1. Entwickeln Sie die beschriebene Jalousiesteuerung als Mealy-Automat unter Verwendung der fünf Zustände *closed*, *down*, *manual*, *open* und *up* und geben Sie den resultierenden Automatengraphen an. (4 Punkte)

2. Vervollständigen Sie die nachfolgend gegebene Automatentafel unter Verwendung von taktflankengesteuerten D-, JK- bzw. T-Flipflops. (6 Punkte)

Zustandsname	Aktueller Zustand			Eingabe		Nachfolgezustand			Ansteuerfunktion				Ausgabe	
	q_2	q_1	q_0	i_1	i_0	q_2'	q_1'	q_0'	D_2	J_1	K_1	T_0	o_1	o_0
open	0	0	0	0	0									
open	0	0	0	0	1									
open	0	0	0	1	0									
open	0	0	0	1	1									
down	0	0	1	0	0									
down	0	0	1	0	1									
down	0	0	1	1	0									
down	0	0	1	1	1									
closed	0	1	0	0	0									
closed	0	1	0	0	1									
closed	0	1	0	1	0									
closed	0	1	0	1	1									
up	0	1	1	0	0									
up	0	1	1	0	1									
up	0	1	1	1	0									
up	0	1	1	1	1									
manual	1	0	0	0	0									
manual	1	0	0	0	1									
manual	1	0	0	1	0									
manual	1	0	0	1	1									

3. Entwickeln Sie eine disjunktive Minimalform (DMF) der Ansteuerfunktion des JK-Flipflops J_1 unter Verwendung des gegebenen Symmetriediagramms. Geben Sie den resultierenden schaltalgebraischen Ausdruck an. (3 Punkte)
Achten Sie auf Don't-Cares sowie die vorgegebene Variablenordnung!



4. Zeichnen Sie das vollständige Schaltwerk des in Teilaufgabe 2 erstellten Mealy-Automaten unter Zuhilfenahme der Ergebnisse aus Teilaufgabe 3. Gehen Sie ferner davon aus, dass Sie die benötigten Ansteuerfunktionen D_2, K_1 und T_0 als Eingangssignale zur Verfügung haben. Auf die Realisierung der Ausgabe des Automaten kann verzichtet werden. (3 Punkte)

b) Geben Sie den allgemeinen Aufbau eines Moore-Automaten als Blockschaltbild an. (2 Punkte)

c) Worin unterscheiden sich die drei Automatentypen Moore, Medwedew und Mealy? (1 Punkt)

d) Beschreiben Sie die Unterschiede zwischen Schaltnetz und Schaltwerk. (1 Punkt)

Aufgabe 4 (Codierung und Arithmetik) (15 Punkte)

a) Das Alphabet einer Quelle wurde optimal Huffman-codiert. Alle codierten Zeichen sollen nun nacheinander übertragen werden. Für die Übertragung wird außerdem an jedes Huffman-codierte Zeichen X ein Paritätsbit p angehängt, also $X \to (X,p)$. Folgende Zeichen wurden empfangen:

$$11\,1, \quad 010\,0, \quad 001\,1, \quad 10\,0, \quad 11\,0$$

Bei der Übertragung können pro Zeichen nur Einfachfehler aufgetreten sein. Geben Sie die korrigierten Zeichen an und zeichnen Sie den Codierungsbaum. (2 Punkte)

b) Beim Hamming-Code werden den Symbolen X einer Quelle zusätzliche Prüfbits in Form eines Prüfvektors Y angehängt, also $X \to (X,Y)$.

1. Welche Art von Fehlern können mit Hamming-Codes erkannt werden? (1 Punkt)

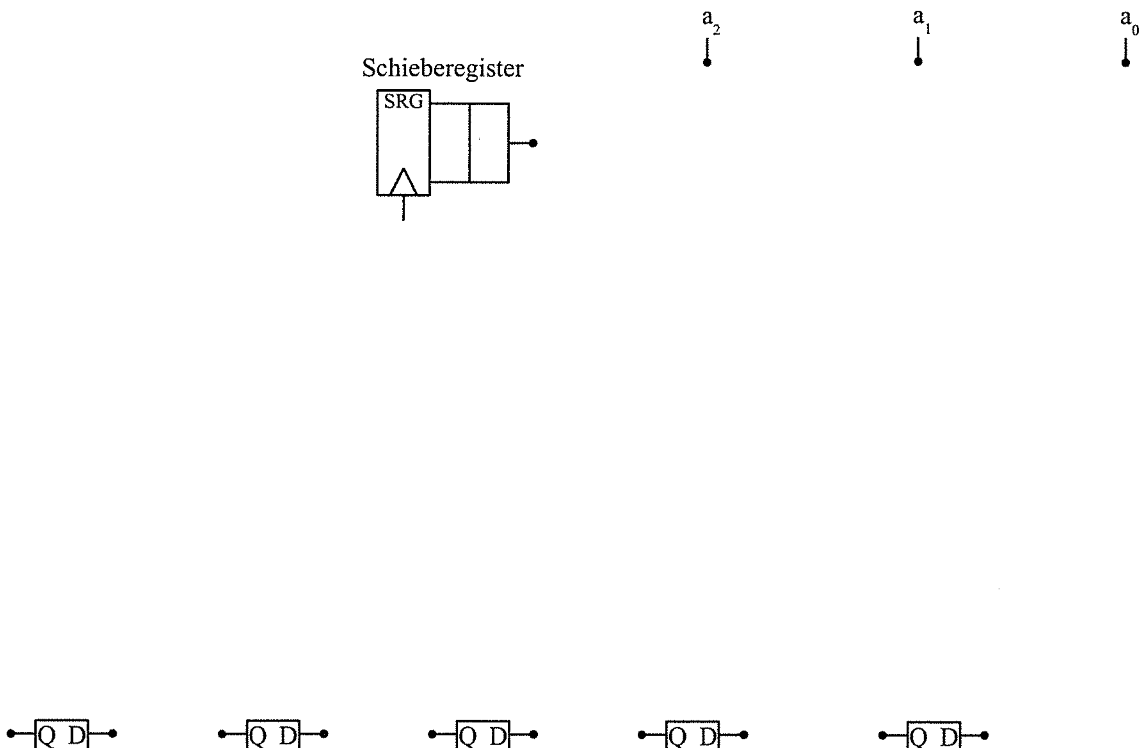
2. Sichern Sie die aus Hexadezimalzahlen bestehende Nachricht `ACD9` zeichenweise gegen Übertragungsfehler, indem Sie den Prüfvektor $Y = (y_2, y_1, y_0)$ berechnen und die neue Nachricht angeben. (2 Punkte)

c) Multiplizieren Sie die beiden vorzeichenlosen Binärzahlen $B = 0110$ und $A = 01010$ durch Anwenden derjenigen Methode aus der Vorlesung, welche die Implementierung eines Multiplizierers mit möglichst geringem Hardwareaufwand und geringer Laufzeit ermöglicht. Geben Sie die einzelnen Schritte explizit an. (2 Punkte)

d) In dieser Aufgabe soll die Verzögerungszeit eines sequentiellen Multiplizierers ermittelt werden, der 3 Bit Multiplikanten $A = (a_2, a_1, a_0)$ mit 2 Bit Multiplikatoren $B = (b_1, b_0)$ multipliziert und das Ergebnis $P = (p_4, p_3, p_2, p_1, p_0)$ liefert.

1. Ergänzen Sie das Schaltbild des Multiplizierers unter ausschließlicher Verwendung von UND-, ODER- und Antivalenz (XOR)-Gattern mit zwei Eingängen sowie Halbaddierern und Volladdierern.

   Kennzeichnen Sie außerdem, welchen Zellen des Schieberegisters anfangs b_1 und b_0 zugeordnet sind und in welchen D-Flipflops letzendlich die Bits p_i, $i = 0, \ldots, 4$ des Ergebnisses stehen. (4 Punkte)



2. Nehmen Sie an, dass UND- und ODER-Gatter eine Verzögerungszeit von $2\ ns$, Antivalenzgatter von $4\ ns$, Halbaddierer von $3\ ns$ und Volladdierer von $7\ ns$ haben. Geben Sie die Laufzeit zur Berechnung der kombinatorischen Logik an und markieren Sie den längsten Pfad im Schaltbild. (1 Punkt)

3. Geben Sie nun die Laufzeit der kombinatorischen Logik an, wenn keine Halbaddierer und Volladdierer mehr als funktionale Einheiten zur Verfügung stehen, sondern diese aus den verbleibenden Gattern (UND-, ODER- und Antivalenzgatter) realisiert werden müssen. (3 Punkte)

Aufgabe 5 (VHDL) (15 Punkte)

Der Manchester-Code ist ein häufig verwendetes Codierungsschema für die serielle Datenübertragung von Informationsbits. Der Wert '0' wird durch eine steigende Flanke repräsentiert, der Wert '1' wird hingegen durch eine fallende Flanke codiert. Ein Beispiel eines Datenstroms ist in Abb. 1 dargestellt.

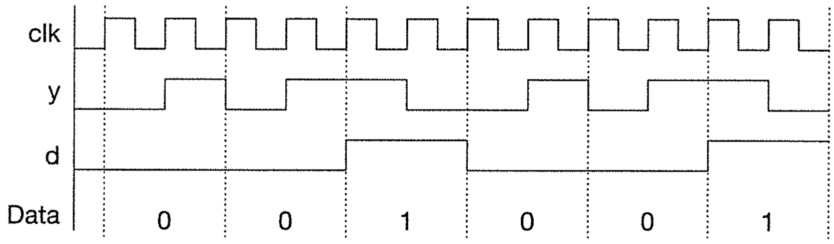


Abbildung 1: Beispiel eines Manchester-codierten Datenstroms.

In dieser Aufgabe gilt es einen synchron getakteten Schaltungsblock `encoder` zur Manchester-Codierung in VHDL zu implementieren. Der Schaltungsblock hat neben dem 1 Bit breiten Eingangsdatensignal `d` einen weiteren Eingang `v`. Hat `v` den Wert '1', so soll der Wert von `d` codiert und auf dem Ausgang `y` ausgegeben werden. Hat `v` hingegen den Wert '0', so soll keine Codierung vorgenommen werden und der Ausgang `y` auf '0' gesetzt werden. Das Verhalten des Codierers ist in Form des Automaten in Abb. 2 gegeben.

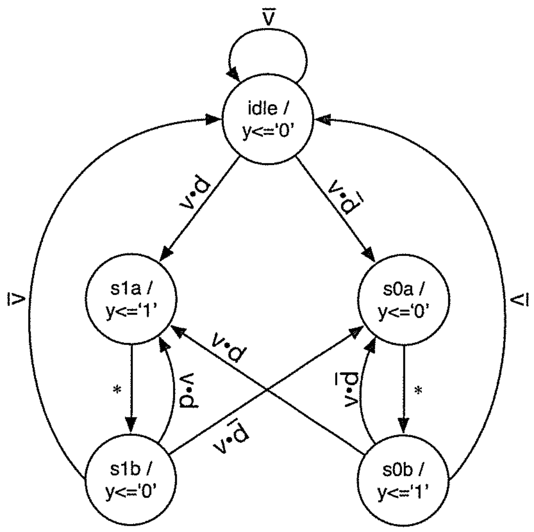


Abbildung 2: Beschreibung des Codierungsvorgangs durch einen Automaten.

a) Geben Sie eine Schnittstellenbeschreibung für den Schaltungsblock `encoder` in Form einer Entity in VHDL an. Verwenden Sie den Datentyp `std_logic` für die Spezifikation der Ein- und Ausgangssignale. (*Auf die Angabe der benötigten Bibliotheken der IEEE kann hierbei verzichtet werden.*) (2 Punkte)

```
entity encoder is



end entity;
```

b) Geben Sie eine Implementierung des Manchester-Codierers in Form einer VHDL-Architecture-Beschreibung für den Schaltungsblock `encoder` an. Verwenden Sie hierzu das vorgegebene Code-Skelett. (*Auf die Angabe der benötigten Bibliotheken der IEEE kann hierbei verzichtet werden.*) (4 Punkte)

```
architecture behavior of encoder is

end architecture;
```

c) Entwerfen Sie einen synchron getakteten Schaltungsblock `decoder` in Form einer VHDL-Architecture-Beschreibung, um einen auf dem Eingang `y` empfangenen Manchester-Datenstrom zu decodieren. Die decodierte Information soll am Ausgang `d` ausgegeben werden. Bei der Datenübertragung kann es jedoch zu Fehlern kommen. Setzen Sie den Ausgang `e` auf den Wert '1', um den Empfang einer fehlerhaften Sequenz anzuzeigen, `d` kann hierbei einen beliebigen Wert haben. Bei fehlerfreien Sequenzen soll `e` hingegen den Wert '0' annehmen. Verwenden Sie das vorgegebene Code-Skelett für Ihre Implementierung. (*Auf die Angabe der benötigten Bibliotheken der IEEE kann hierbei verzichtet werden.*) (6 Punkte)

```
entity decoder is
  port(



  );
end entity;

architecture behavior of decoder is



end architecture;
```

d) In Abb. 3 ist ein möglicher Verlauf des empfangenen Signals y gegeben. Vervollständigen Sie den Verlauf der Signale d und e anhand Ihrer Implementierung aus Teilaufgabe c). (Hinweis: Sollten Sie in Teilaufgabe c) keine Lösung gefunden haben, verwenden Sie die Aufgabenstellung von Teilaufgabe c) als Lösungsgrundlage.) (3 Punkte)

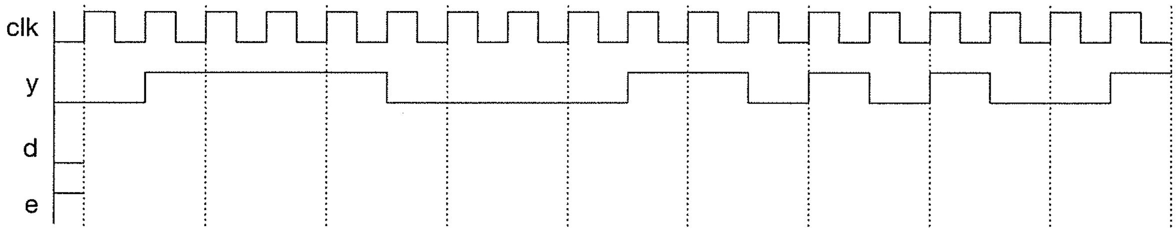


Abbildung 3: Signalverlauf eines empfangenen Manchester-codierten Datenstroms.