

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

# Klausur Grundlagen der Technischen Informatik

25. März 2015

Name	
Matrikelnummer	
Studienrichtung	

Aufgabe	1	2	3	4	5	$\Sigma$
max. Punkte	15	15	20	15	15	80
erreichte Punkte						
<b>Note</b>						

## 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 verfare 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 25. März 2015

.....  
Unterschrift

Kopiervorlage: nur für Fachschaften

## Aufgabe 1 (Zahlensysteme)

(15 Punkte)

- a) Wie lautet die kleinste bzw. größte darstellbare Zahl einer  $n$ -stelligen vorzeichenlosen Hexadezimalzahl? (1 Punkt)
- b) Wie lautet die kleinste bzw. größte darstellbare Zahl einer  $n$ -stelligen Oktalzahl in Zweierkomplement-Darstellung? (1 Punkt)
- c) Konvertieren Sie die Dezimalzahl  $156_{10}$  in eine vorzeichenlose 8-stellige Binärzahl. (1 Punkt)
- d) Subtrahieren Sie  $100_{10}$  von der Zweierkomplement-Zahl  $1110\ 0100_2$ , d.h.  $1110\ 0100_2 - 100_{10}$ . Die Berechnung und das Ergebnis sind in Zweierkomplement-Darstellung mit 8 Bit Genauigkeit durchzuführen. (2 Punkte)
- e) 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. (3 Punkte)
1. Mit den Konstanten "wahr" und "falsch" sowie UND- und ODER-Gattern lässt sich jede beliebige Schaltfunktion realisieren.  wahr  falsch
  2. Mit den Konstanten "wahr" und "falsch" sowie 2:1-Multiplexern lässt sich jede beliebige Schaltfunktion realisieren.  wahr  falsch
  3. Wenn bei der Addition in der Zweierkomplement-Darstellung ein Übertrag bei der Berechnung der höchstwertigen Stelle auftritt, dann ist ein Überlauf vorhanden, und somit ist das Ergebnis der Berechnung ungültig.  wahr  falsch

- f) Stellen Sie die Zahlen 0 bis 7 in Gray-Code dar, und erklären Sie das Einsatzgebiet für die Gray-Codierung. (2 Punkte)

- g) In dieser Aufgabe soll mit 8-Bit Gleitkommazahlen gearbeitet werden. Diese werden analog zum IEEE-Format gebildet. Das Format der Gleitkommazahl sieht dabei wie folgend aus:  
*Vorzeichen (1 Bit), Exponent (3 Bit), Mantisse (4 Bit)*

<i>V</i>	<i>E</i>	<i>M</i>
7	6 4 3	0

- Führen Sie nun die Multiplikation der beiden in diesem Format dargestellten Gleitkommazahlen 1 100 1100 und 1 010 1001 aus. Ein möglicher Genauigkeitsverlust des Ergebnisses soll durch Aufrunden behandelt werden. (5 Punkte)

Kopiervorlage: nur für Fachschaften

## Aufgabe 2 (Minimierung von Schaltfunktionen)

(15 Punkte)

a) Gegeben sei die Schaltfunktion  $f_1(x_3, x_2, x_1, x_0) = x_0 \bar{x}_1 \bar{x}_2 + x_0 \bar{x}_1 \bar{x}_3$  in disjunktiver Form (DF).

1. Entwickeln und zeichnen Sie eine Gatterschaltung von  $f_1(x_3, x_2, x_1, x_0)$ , die nur aus ODER-Gattern und Invertern besteht. (3 Punkte)

2. Bestimmen Sie eine konjunktive Form (KF) von  $f_1(x_3, x_2, x_1, x_0)$ . (1 Punkt)

3. Realisieren Sie  $f_1(x_3, x_2, x_1, x_0)$  als CMOS-Schaltung mit möglichst wenig Transistoren. Alle Eingänge stehen nur in der nicht invertierten Form zur Verfügung. (5 Punkte)

**VDD**

---

**GND**

---

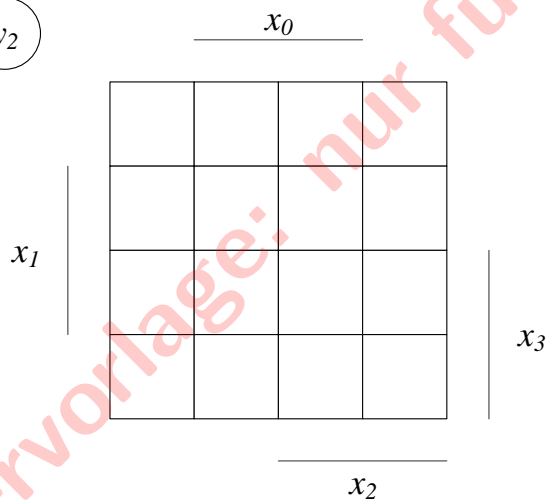
- b) Nachfolgend soll ein Schaltnetz entworfen werden, das das Vorzeichen einer 4-Bit Zweierkomplementzahl im Bereich -4 bis 4 invertiert. So wird beispielsweise eine -3 als Eingangszahl eine 3 als Ausgangszahl zur Folge haben.

1. Spezifizieren Sie zunächst die Schaltfunktion durch das Ausfüllen der nachfolgende Funktionstabelle. (2 Punkte)

Eingangszahl	Zweierkomplement				Ausgangszahl	Zweierkomplement			
	$x_3$	$x_2$	$x_1$	$x_0$		$y_3$	$y_2$	$y_1$	$y_0$
-4					4				
-3					3				
-2					2				
-1					1				
0					0				
1					-1				
2					-2				
3					-3				
4					-4				

2. Bestimmen Sie nun alle Primimplikanten des Ausgangs  $y_2$  mit Hilfe eines Symmetriediagramms. (4 Punkte)

$y_2$



Primimplikanten:

## Aufgabe 3 (Automaten und Flipflops)

(20 Punkte)

- a) Im Folgenden soll ein Kartoffelsackautomat entworfen werden. Dieser ist wie folgt spezifiziert: Ein Kartoffelsackautomat ermöglicht einem Landwirt, seine erntefrischen Kartoffeln unabhängig von Öffnungszeiten und ohne Personaleinsatz zu verkaufen. Der Kartoffelsackautomat verfügt über vier verschiedene Eingaben, die jedoch nicht gleichzeitig anliegen können. Durch Einwurf des korrekten Geldbetrages (Eingabe *ICorrect*) wird eine von drei Gitterboxen geöffnet, und der Kunde kann den darin enthaltenen Kartoffelsack entnehmen (Ausgabe *Potato*). Falls kein Kartoffelsack mehr im Automat enthalten ist, wird der eingeworfene Betrag wieder ausgegeben (Ausgabe *Money*). Wird ein falscher Geldbetrag eingeworfen (Eingabe *IWrong*), so wird unabhängig vom Füllstand auch dieser Betrag wieder ausgegeben. Die Eingabe *IPotato* repräsentiert das Bestücken des Automaten mit einem neuen Kartoffelsack (Ausgabe *NoOut*). Ist der Automat zum Geldentleeren und Säubern im Wartungsmodus (Eingabe *IMaint*), wird durch die Ausgabe *Maint* signalisiert, dass der Automat nicht einsatzbereit ist. Liegt das zugehörige Eingangssignal *IMaint* nicht mehr an, so kann davon ausgegangen werden, dass der Automat vollständig bestückt ist und den Regelbetrieb fortsetzen kann (Ausgabe *NoOut*).

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

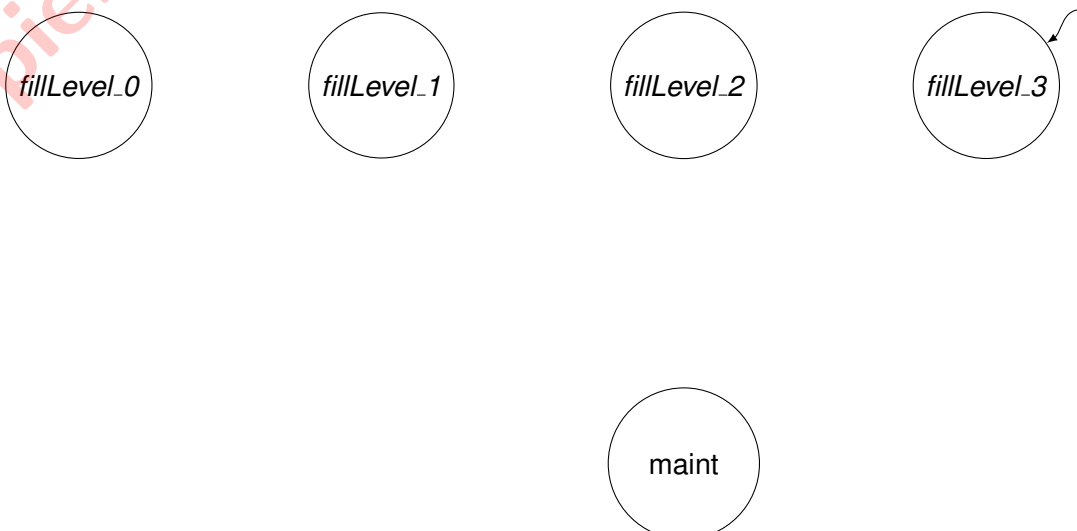
Eingabe	$i_1$	$i_0$
<i>ICorrect</i>	0	0
<i>IWrong</i>	0	1
<i>IPotato</i>	1	0
<i>IMaint</i>	1	1

Eingaben des Automaten

Ausgabe	$o_1$	$o_0$
<i>Potato</i>	0	0
<i>Money</i>	0	1
<i>NoOut</i>	1	0
<i>Maint</i>	1	1

Ausgaben des Automaten

1. Spezifizieren Sie den beschriebenen Kartoffelsackautomaten als Mealy-Automat unter Verwendung der fünf Zustände *fillLevel\_3* (3 Kartoffelsäcke im Automat), *fillLevel\_2*, *fillLevel\_1*, *fillLevel\_0* (kein Kartoffelsack im Automat) und *maint*. Geben Sie den resultierenden Automatengraphen an. (5 Punkte)



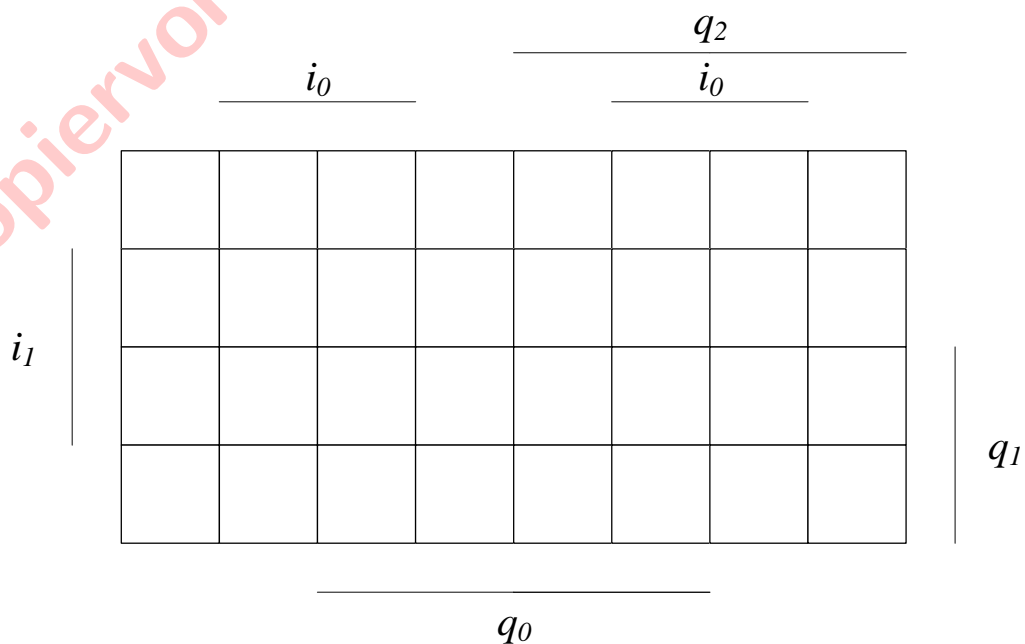


2. Vervollständigen Sie die nachfolgend gegebene Automatentafel unter Verwendung von taktflankengesteuerten D-, T- bzw. JK-Flipflops. (6 Punkte)

Zustandsname	Aktueller Zustand			Eingabe		Nachfolgezustand			Ansteuerfunktion				Ausgabe	
	$q_2$	$q_1$	$q_0$	$i_1$	$i_0$	$q'_2$	$q'_1$	$q'_0$	$D_2$	$T_1$	$J_0$	$K_0$	$o_1$	$o_0$
fillLevel_3	0	0	0	0	0									
fillLevel_3	0	0	0	0	1									
fillLevel_3	0	0	0	1	0									
fillLevel_3	0	0	0	1	1									
fillLevel_2	0	0	1	0	0									
fillLevel_2	0	0	1	0	1									
fillLevel_2	0	0	1	1	0									
fillLevel_2	0	0	1	1	1									
fillLevel_1	0	1	0	0	0									
fillLevel_1	0	1	0	0	1									
fillLevel_1	0	1	0	1	0									
fillLevel_1	0	1	0	1	1									
fillLevel_0	0	1	1	0	0									
fillLevel_0	0	1	1	0	1									
fillLevel_0	0	1	1	1	0									
fillLevel_0	0	1	1	1	1									
maint	1	0	0	0	0									
maint	1	0	0	0	1									
maint	1	0	0	1	0									
maint	1	0	0	1	1									

3. Entwickeln Sie eine konjunktive Minimalform (KMF) der Ansteuerfunktion  $J_0$  des JK-Flipflops unter Verwendung des gegebenen Symmetriediagramms. Geben Sie den resultierenden schaltalgebraischen Ausdruck an. (4 Punkte)

*Achten Sie auf Don't-Cares sowie auf 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  $D_2$ ,  $T_1$  und  $K_0$  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)

Kopiervorlage nur für Fachschaften

## Aufgabe 4 (Codierung und Arithmetik)

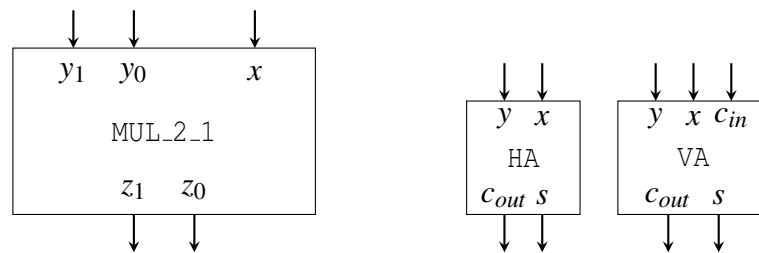
(15 Punkte)

a) Strukturierte ( $k$ -aus- $m$ )-Codes bestehen aus  $m$ -stelligen Codewörtern mit jeweils genau  $k$  Einsen. Zum Beispiel besitzt der (2-aus-3)-Code die Codewörter 011, 101 und 110.

1. Was ist die minimale Hammingdistanz  $HD_{min}$  der Codewörter eines ( $k$ -aus- $m$ )-Codes? (1 Punkt)
  
2. Für die Übermittlung von Codewörtern des (2-aus-4)-Codes werden immer vier Codewörter zu einer Nachricht zusammengefasst. Bei der Übertragung solcher Nachrichten können Einfachfehler auftreten. Durch ein Blocksicherungsverfahren ist es jedoch möglich, Einfachfehler zu korrigieren. Zeigen Sie beispielhaft, wie diese Codierung für die Übertragung der Codewörter 1001, 0110, 0101, 0110 aussieht. Dabei sollen so wenig zusätzliche Paritätsbits (mit even parity) wie möglich hinzugefügt werden. (3 Punkte)
  
3. Welches weitere Verfahren kann verwendet werden, um Einfachfehler bei der Übertragung von Codewörtern korrigieren zu können? (1 Punkt)

Kopiervorlage: nur für Fachschüler

- b) Gegeben sei die digitale Schaltung  $MUL\_2\_1$  eines Multiplizierers, der 2-Bit Multiplikanden mit 1-Bit Multiplikatoren multipliziert. Außerdem seien die digitalen Schaltungen eines Halbaddierers (HA) und eines Volladdierers (VA) gegeben.



1. Entwerfen Sie unter ausschließlicher Verwendung der gegebenen Bausteine das Schaltnetz eines Multiplizierers, der 2-Bit Multiplikanden  $a_1 a_0$  mit 3-Bit Multiplikatoren  $b_2 b_1 b_0$  multipliziert und das Ergebnis  $p_4 p_3 p_2 p_1 p_0$  liefert. (6 Punkte)

Kopiervorlage: nur für Fachschaffern

2. Welche der folgenden Schaltungselemente werden zwingend zusätzlich benötigt, um aus den gegebenen Bausteinen einen sequentiellen Multiplizierer zu bauen? (1 Punkt)
- Schieberegister     Modulo-3-Zähler  
 Komparator         Flipflop
3. Entwerfen Sie das Schaltnetz des Multiplizierers `MUL_2_1` unter ausschließlicher Verwendung von NOR-Gattern mit zwei Eingängen. Verwenden Sie dazu so wenig NOR-Gatter wie möglich. (3 Punkte)

Kopiervorlage: nur für Fachschaften

Aufgabe 5 (VHDL)

(15 Punkte)

- a) RS-232 ist ein Standard zur seriellen Datenübertragung. Gemäß dieses Standards soll in dieser Aufgabe ein schneller Sender mit einem langsamen Empfänger mittels eines 9-poligen, sogenannten D-Sub-Anschlusses verbunden werden (vgl. Abbildung 1).

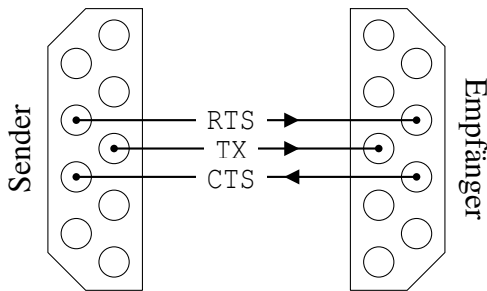


Abbildung 1: Verbindung via 9-poligem D-Sub-Stecker

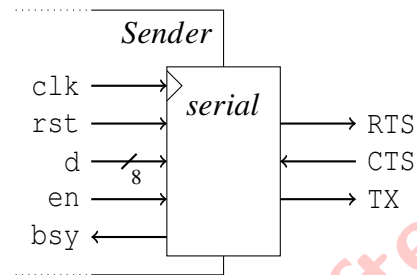


Abbildung 2: Schematische Darstellung des Serialisierers

Es werden nur drei der neun Leitungen benötigt. Möchte der Sender Daten an den Empfänger senden, setzt er RTS (*request to send*) auf 1 und wartet, bis der Empfänger als Antwort CTS (*clear to send*) auf 1 setzt. Dann schickt der Sender 10 Bits seriell (d.h. nacheinander, Bit für Bit) über TX: zuerst ein Startbit  $S = 0$ , dann 8 Datenbits (ein Byte, beginnend mit dem niedrigwertigsten Bit), und schließlich ein Endbit  $E = 1$ , das die erfolgreiche Übertragung signalisiert.

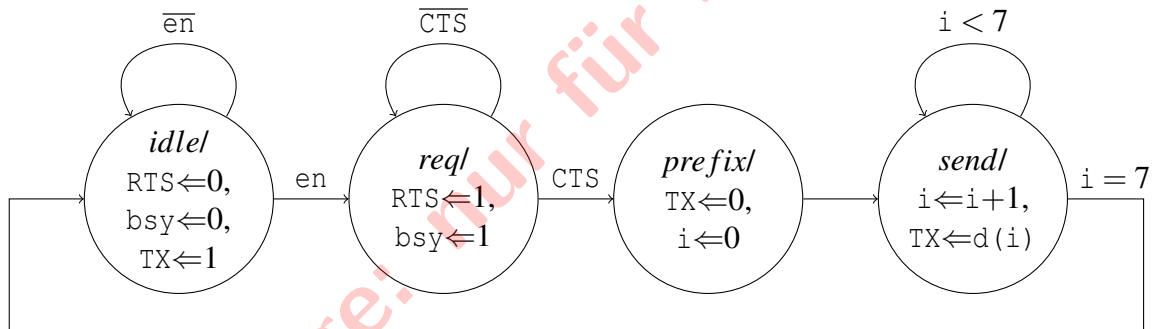
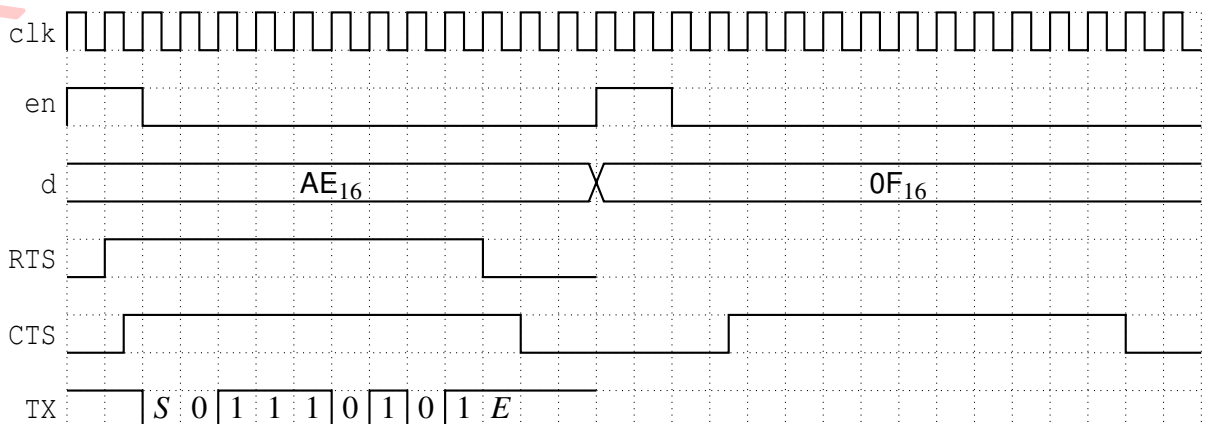


Abbildung 3: Verhalten des Serialisierers als endlicher Automat

Im Rahmen dieser Aufgabe soll der Serialisierer aus Abbildung 2 realisiert werden, welcher Teil des Senders ist und ein 8-Bit Datensignal  $d$  serialisiert an den Empfänger schickt, sobald das Enable-Signal  $en$  aktiviert wird. Solange der Serialisierer mit dem Senden beschäftigt ist, sei das Signal  $bsy$  auf 1 gesetzt.  $rst$  sei ein asynchrones Reset-Signal. Das konkrete Verhalten ist bereits durch den endlichen Automaten in Abbildung 3 vorgegeben und ist exemplarisch in folgendem Wellenformdiagramm aufgetragen:



1. Vervollständigen Sie das Wellenformdiagramm für die Signale  $RTS$  und  $TX$ . (Hinweis: das 8 Bit breite Signal  $d$  ist zusammengefasst dargestellt. 14 Takte lang liegt  $AE_{16}$  an, danach 16 Takte lang  $0F_{16}$ .) (2 Punkte)
2. Implementieren Sie den Serialisierer in VHDL gemäß des Automaten in Abbildung 3. Die erforderlichen Bibliotheken seien bereits eingebunden. Nutzen Sie das folgende Code-Skelett und gehen Sie davon aus, dass Sie das  $i$ -te Bit von  $d$  mit dem Ausdruck  $d(i)$  auslesen können. (6 Punkte)

```
entity serial is  
port (clk, rst, en, CTS : in std_logic;  
      d : in std_logic_vector(7 downto 0);  
      bsy, RTS, TX : out std_logic);  
end serial;
```

```
architecture behavior of serial is  
subtype bit_count is integer range 0 to 7;  
signal i : bit_count;
```

```
begin
```

Kopiervorlage: nur für Fachschaften

**end** behavior ;

3. Manche Synthesewerkzeuge sind nicht in der Lage, den Ausdruck  $d(i)$  direkt zu synthetisieren, da  $i$  nicht statisch (d.h. nicht konstant) ist. Implementieren Sie deshalb einen Multiplexer in Form einer Funktion, die das  $i$ -te Bit von  $d$  zurückliefert. (2 Punkte)

**function** get\_bit (

**begin**

**end** get\_bit ;

- b) Bearbeiten Sie die folgenden Aufgaben und Fragen zu VHDL und den digitalen Schaltungsentwurf im Allgemeinen.

1. Erklären Sie die in VHDL geltenden syntaktischen und semantischen Unterschiede zwischen Signalen und Variablen. (2 Punkte)

2. Wie unterstützt VHDL die Wiederverwendung von Hardware-Beschreibungen? (1 Punkt)

3. Was bezwecken Testbenches und wie funktionieren sie? (1 Punkt)

4. Worin unterscheiden sich synchrone und asynchrone Schaltungen? (1 Punkt)