

Vorname: \_\_\_\_\_

Name: \_\_\_\_\_

Matr.-Nr.: \_\_\_\_\_

Note: \_\_\_\_\_

28.08.2006, 14.00 Uhr - 16.00 Uhr

**UNIVERSITÄT KARLSRUHE**  
**Institut für Industrielle Informationstechnik**  
- Prof. Dr.-Ing. habil. K. Dostert -

**Vordiplomprüfung im Fach**

***Mikrorechnertechnik***

Die Prüfung umfasst **10 Aufgaben auf 24 Seiten**.

Bitte schreiben Sie auf **alle** Lösungsblätter Ihren Namen und Ihre Matrikelnummer.

Geben Sie bei allen Aufgaben den kompletten Lösungsweg an! Die alleinige Nennung des Endresultats ist zur Erlangung der vollen Punktzahl einer Aufgabe nicht ausreichend.

Die Verwendung eigenen Papiers ist nicht erlaubt.

Bei Bedarf kann zusätzliches Schreibpapier bei der Aufsicht angefordert werden.

Als Hilfsmittel sind Schreib- und Zeichenzeug sowie Taschenrechner mit zu Beginn der Klausur gelöschtem Speicher zugelassen.

Aufgabe:	1	2	3	4	5	6	7	8	9	10	gesamt
Punkte:											
Erreichbare Punktzahl:	10	9	11	10	10	11	9	9	12	9	100

**10 Punkte**

- [illegible]

**Fortsetzung der 1. Aufgabe:**

Eine spezielle Anwendung von Speichern ist die schnelle speicherbasierte Multiplikation.

- d) Gegeben sei ein speicherbasierter Multiplizierer der über ein ROM der Größe 3,75 GByte verfügt. Welche maximale Bitbreite können die beiden Faktoren annehmen?

**Hinweis:** Gehen Sie davon aus, dass beide Faktoren dieselbe Bitbreite annehmen und dass die Ausgangsbitbreite passend gegeben ist. Belegen Sie ihr Ergebnis anhand einer Berechnung.

**9 Punkte**

MSB							LSB
1	1	1	0	0	0	1	0

a) Vorzeichenlose Integerdarstellung!

b) Vorzeichenbehaftete Integerdarstellung!

c) Fraktalzahendarstellung!

**Fortsetzung der 2. Aufgabe:**

Im Folgenden wird mit einem 32 bit-Signalprozessor gearbeitet. Ein Register des Prozessors enthalte folgende Bitfolge:

1100 0100 1001 1000 0000 0000 0000 0000

- d) Geben Sie den Dezimalwert des Registerinhalts an, wenn der Prozessor das Gleitkommaformat mit einfacher Genauigkeit nach IEEE-P754 verwendet!

- e) Welchem Dezimalwert entspricht diese Bitfolge, wenn der Mikrorechner Fraktalzahlendarstellung verwendet?

- f) Stellen Sie die Zahl  $-\left(10 + \frac{3}{16}\right)$  als Gleitkommazahl gemäß dem IEEE-P754 Single-Precision-Standard dar!

### Aufgabe 3: Beschreibung einer FSM

11 Punkte

Das Steuerwerk stellt ein wichtiges Kernstück eines jeden Mikrorechners dar. Im Folgenden wird ein Ausschnitt aus einer Mikrosequencer-Steuerung als Teil des Steuerwerks in Form eines endlichen Automaten (FSM) betrachtet. Abbildung 3.1 zeigt das Blockschaltbild einer solchen FSM. Die FSM bildet den Befehlscode (OP-Code) auf eine Folge von Zuständen (A) ab. Außerdem sind Eingänge für den Takt, für eine Bedingung (COND) sowie zum Rücksetzen (RESET) vorhanden.

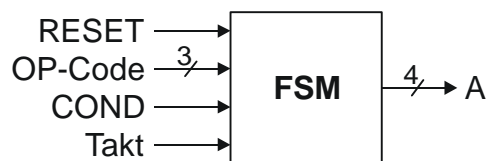


Abbildung 3.1: Blockschaltbild einer FSM als Bestandteil eines Mikrosequencers

Es werden neun Zustände betrachtet. Tabelle 3.1 zeigt die Zuordnung der Zustände zu den Befehlen, wobei für einen Befehl, je nach Komplexität, bis zu vier Zustände benötigt werden.

Zustand	Befehl	Beschreibung
s1	RES	Reset
s2	BRA_1	Verzweigung
s3	BRA_2	
s4	JMP	Sprung
s5	JMS_1	Sprung in Unterprogramm
s6	JMS_2	
s7	JMS_3	
s8	JMS_4	
s9	–	Ruhezustand

Tabelle 3.1: Zuordnung der Zustände der FSM zu den Befehlen

In Tabelle 3.2 ist ein Ausschnitt aus der Flusstabelle zur Beschreibung der FSM angegeben. Dabei sind die Zustände mit  $s1 \dots s9$ , die Folgezustände mit  $f1 \dots f9$  bezeichnet. Der Eingangsvektor  $x = (OP3, OP2, OP1, COND)$  enthält 3 bit für den Befehlscode und ein Bedingungsbit. Das *RESET*-Bit wurde der Übersichtlichkeit halber weggelassen. Der Eingangsvektor bestimmt, ausgehend vom aktuellen Zustand  $s1 \dots s9$ , jeweils den Folgezustand  $f1 \dots f9$ . Irrelevante Bedingungen sind durch „–“ gekennzeichnet.

- Vervollständigen Sie den Zustandsgraphen in Abbildung 3.2, indem Sie in den Knoten die Zustände und Befehle gemäß Tabelle 3.1 eintragen!
- Vervollständigen Sie den Zustandsgraphen in Abbildung 3.2, indem Sie die fehlenden Zustandsübergänge als Kanten mit den zugehörigen Zustandsübergangsbedingungen gemäß Tabelle 3.2 eintragen!

### Fortsetzung der 3. Aufgabe:

relevant =	OP3	OP2	OP1	COND;	
s9, x	0	0	0	-	,f1;
s1, x	-	-	-	-	,f9;
s9, x	0	1	0	-	,f2;
s9, x	0	1	1	1	,f2;
s2, x	-	-	-	-	,f3;
s3, x	-	-	-	-	,f9;
s9, x	1	0	0	-	,f4;
s4, x	-	-	-	-	,f9;
s9, x	1	0	1	1	,f4;
s9, x	1	1	0	-	,f5;
s5, x	-	-	-	-	,f6;
s6, x	-	-	-	-	,f7;
s7, x	-	-	-	-	,f8;
s8, x	-	-	-	-	,f9;

Tabelle 3.2: Flusstabelle zur Beschreibung der FSM

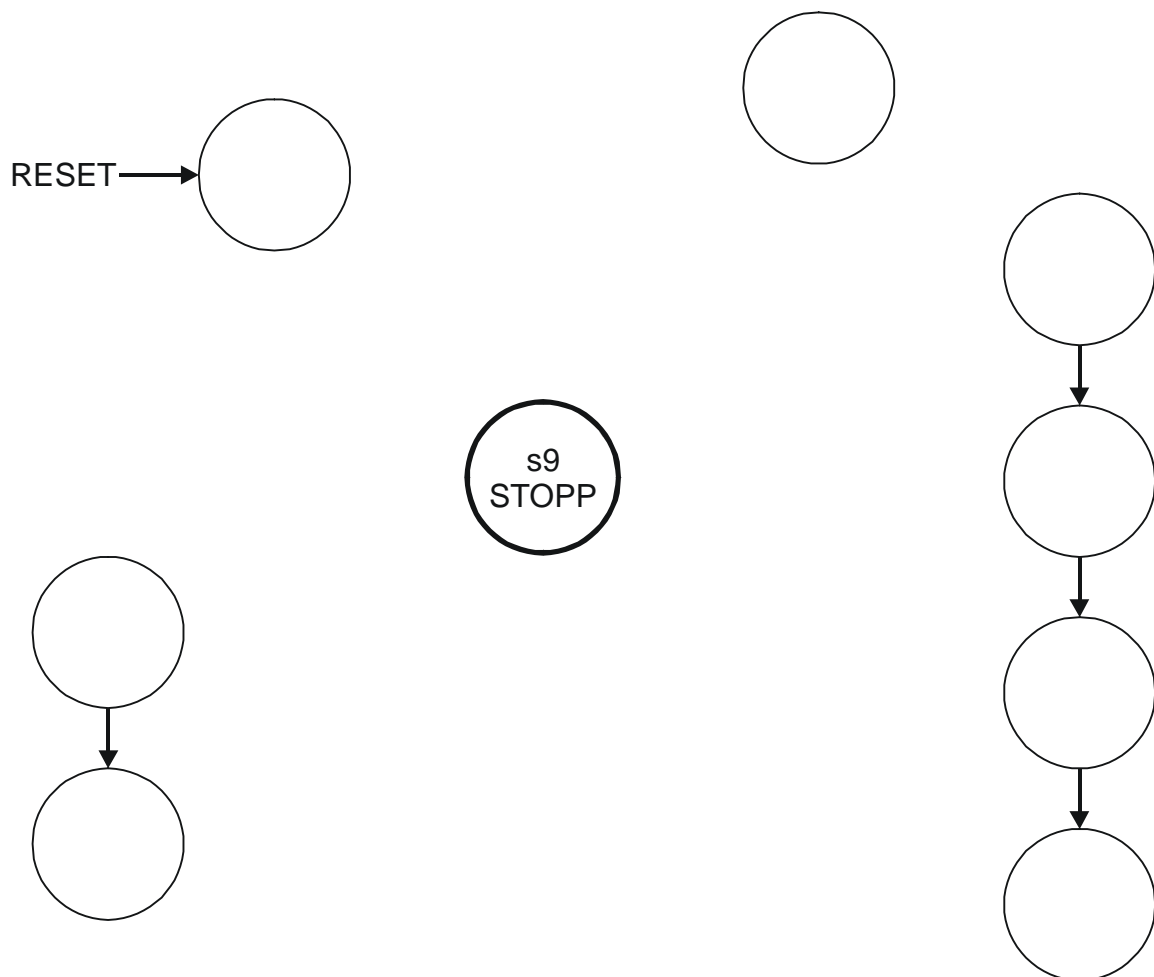


Abbildung 3.2: Zustandsgraph zur Beschreibung der FSM

**Fortsetzung der 3. Aufgabe:**

In Tabelle 3.3 sind die Befehlscodes einiger Befehle angegeben, die mit der FSM realisiert werden können.

- c) Ordnen Sie den Befehlscodes in Tabelle 3.3 die Befehle „Reset“, „Sprung in Unterprogramm“, „bedingter Sprung“ und „unbedingte Verzweigung“ zu!

Befehl	OPC3	OPC2	OPC1
	0	1	0
	1	1	0
Reset	0	0	0
	1	0	1

Tabelle 3.3: Zuordnung von Befehlen zu Befehlscode



**10 Punkte**

Nachfolgende Abbildung 4.1 zeigt das Schaltsymbol eines Transfergates mit Ein- und Ausgang, dem Takteingang sowie dem negierten Takteingang.

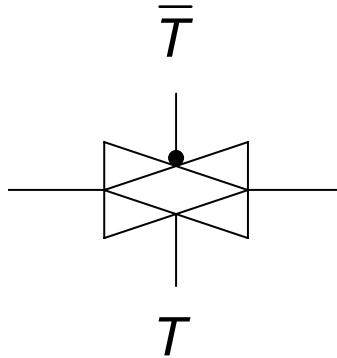


Abbildung 4.1: Transfergate

- [illegible]

**Fortsetzung der 4. Aufgabe:**

Gegeben sei im Folgenden die Schaltung aus Abbildung 4.2.

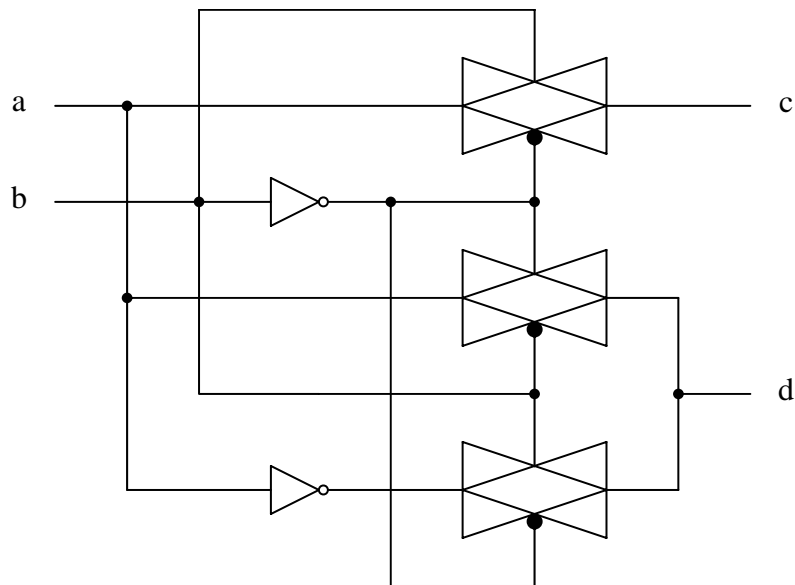


Abbildung 4.2: Schaltung mit Transferegates

- c) Vervollständigen Sie das Timingdiagramm aus Abbildung 4.3 entsprechend den vorgegebenen Eingangspegeln a und b!

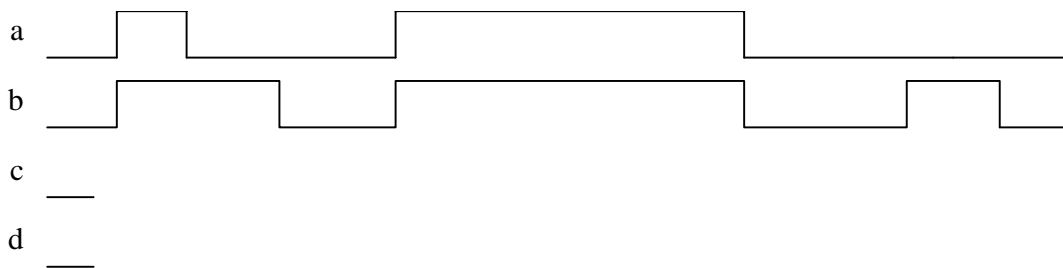


Abbildung 4.3: Timingdiagramm

- d) Welche Funktion wird mittels der Schaltung aus Abbildung 4.2 realisiert? Füllen Sie die zugehörige Wahrheitstabelle 4.1 aus!

a	b	c	d

Tabelle 4.1: Wahrheitstabelle

## Aufgabe 5: Arithmetik-Schaltungen

10 Punkte

In Abbildung 5.1 ist ein aus NAND-Gattern aufgebautes Schaltnetz gegeben. Jedes der NAND-Gatter hat eine Durchlaufzeit von 3 ns.

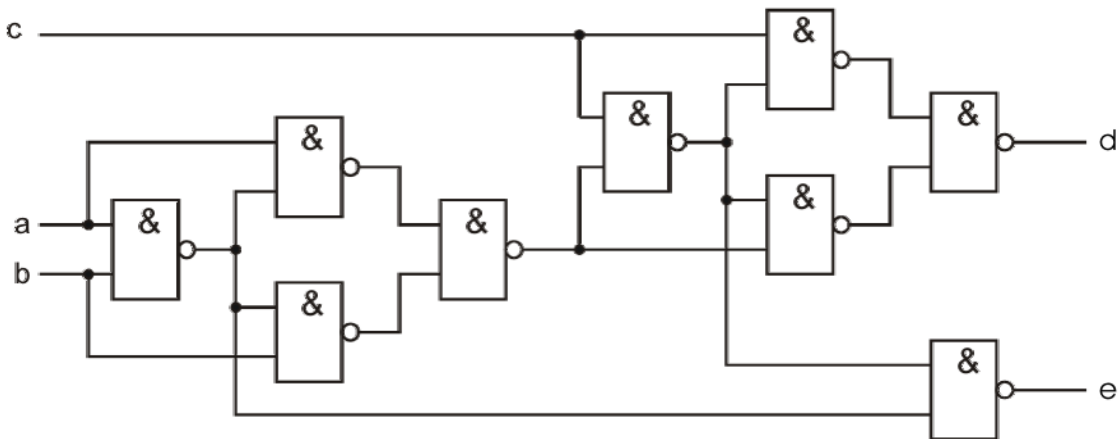


Abbildung 5.1: Schaltnetz mit NAND-Gattern

- a) Vervollständigen Sie die Funktionstabelle 5.1 dieses Schaltnetzes!

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Tabelle 5.1: Funktionstabelle des Schaltnetzes mit NAND-Gattern

- b) Welche arithmetische Operation wird durch das obige Schaltnetz realisiert? Was stellen dabei der Ausgang *s* und was der Ausgang *c<sub>out</sub>* dar?

**Fortsetzung der 5. Aufgabe:**

- c) Wie lange benötigt das Schaltnetz zur Ausführung seiner Funktion?

Vor die Eingänge  $a$ ,  $b$  und  $c$  und hinter die Ausgänge  $d$  und  $e$  des in Abbildung 5.1 gegebenen Schaltnetzes werden nun Register geschaltet, die mit einem gemeinsamen Taktsignal  $clk$  betrieben werden. Das Ergebnis ist ein synchrones Schaltnetz. Für die weiteren Berechnungen sei angenommen, dass alle Register verzögerungsfrei arbeiten.

- d) Wie groß ist die maximale Frequenz  $f$  des Taktsignals  $clk$  für das mit Registern erweiterte Schaltnetz?

### Aufgabe 6: Serielle Schnittstelle des 80C51

11 Punkte

Unter Verwendung des integrierten Timers 1 bietet der Mikrocontroller 80C51 vielfältige Möglichkeiten der Baudratengenerierung für die serielle Datenübertragung. Dazu soll im Folgenden die Betriebsart 1 der seriellen Schnittstelle verwendet werden. Diese erlaubt die asynchrone Datenübertragung mit variabler Baudrate, wobei als Zeitbasis die Überlaufrate von Timer 1 dient.

- a) Erläutern Sie die Begriffe „Simplex-“, „Halbduplex-“ und „Vollduplexbetrieb“ im Zusammenhang mit der seriellen Datenübertragung!

Die Taktfrequenz des Mikrocontrollers betrage 10 MHz. Um die Baudratengenerierung mit möglichst geringer Prozessorbelastung durchzuführen, soll Timer 1 im Autoreload-Mode betrieben werden. Die serielle Schnittstelle werde in der Betriebsart (Mode) 1 betrieben.

- b) Berechnen Sie die Baudrate in Abhängigkeit vom Wert des Steuerbits SMOD, falls das Reloadregister TH1 von Timer 1 den Wert 80 (hexadezimal) enthält!

**Fortsetzung der 6. Aufgabe:**

- c) Geben Sie in Tabelle 6.1 die notwendigen Einstellungen der Spezialfunktionsregister SCON, TCON, TMOD und IE an und kennzeichnen Sie irrelevante Bits durch „X“!

	Bit 7				Bit 0			
SCON								
TCON								
TMOD								
IE								

Tabelle 6.1: Initialisierung der Register SCON, TCON TMOD und IE

- d) Berechnen Sie die maximale Baudrate, die in der angegebenen Betriebsart (Schnittstelle in Mode 1, Timer 1 im Autoreload-Mode) generiert werden kann!

## Aufgabe 7: Mikrocontroller-Programmierung

9 Punkte

In dieser Aufgabe soll ein Quasi-Zufallszahlengenerator mit dem 80C51 programmiert werden. Zur Berechnung der Zufallszahl wird der Algorithmus nach D.H. Lehmer (1948) angewandt. Ausgehend von einem ganzzahligen Startwert berechnet der Algorithmus iterativ eine Zufallszahl nach folgender Gleichung 7.1:

$$X_{n+1} = (a_1 \cdot X_n + a_2) \bmod a_3 \quad (\text{Gleichung 7.1})$$

**Hinweis:** Die Modulo-Rechenoperation (mod) berechnet den Rest einer ganzzahligen Division (gemäß obiger Formel durch  $a_3$ )!

- a) Welchen Wert darf  $a_3$  maximal annehmen? Begründen Sie Ihre Antwort!
- b) Es sei folgende Anfangsbelegung gegeben:  $X_0 = 3$ ,  $a_1 = 37$ ,  $a_2 = 53$ ,  $a_3 = 100$ . Welcher Wert  $X_2$  wird ausgehend von dieser Belegung erzeugt?

Im Folgenden sollen die für die obige Berechnung benötigten Unterprogramme entwickelt werden. Dazu wird der Startwert  $X_n$  als Konstante im Hauptprogramm gesetzt und anschließend das Unterprogramm *GETRND* aufgerufen. Dieses initialisiert mittels *INITRND* den Quasi-Zufallsgenerator und berechnet den Folgewert  $X_{n+1}$ .

- c) Vervollständigen Sie im nachfolgenden Programmabschnitt die sieben mit ..... gekennzeichneten Stellen, so dass die oben beschriebene Funktionalität entsteht!

---

Fortsetzung der 7. Aufgabe:

```
;*****
DSEG

FIRSTRND EQU      ..... ; Anfangswert  $X_n$  für Zufallsgenerierung

RANDOM: DS 1

;*****
CSEG
;*****

CALL GETRND


; Zufallszahl initialisieren

INITRND:
    MOV      ....., #FIRSTRND      ; festen Anfangswert setzen
    RET
;*****

; Zufallszahl berechnen (Algorithmus nach D.H. Lehmer (1948))
;  $X_{n+1} = (a_1 * X_n + a_2) \bmod a_3$ , mit  $a_1 = 37$ ,  $a_2 = 53$ ,  $a_3 = 100$ 

GETRND:
    CALL INITRND      ; den Wert  $X_n$  initialisieren
    MOV      A, RANDOM ; den Wert  $X_n$  in Akku A laden
    MOV      B, #37
    .....      ; das Produkt  $a_1 * X_n$  berechnen
    ADD      A, #53
    MOV      B, #100
    DIV      ..... ; Modulo-Berechnung
    .....      ; den Wert  $X_{n+1}$  nach RANDOM schreiben
    RET

;*****
```



## Aufgabe 8: Architektur von DSPs und Analyse von Assemblercode

9 Punkte

Der DSP 56000 zeichnet sich unter anderem durch vielseitige Möglichkeiten der Registeradressierung sowie durch die MAC-Operation aus.

Tabelle 8.1 gibt einen Auszug aus der Register- und Speicherbelegung des DSPs an. Ausgehend von dieser Speicherbelegung werden die folgenden 3 Befehle ausgeführt:

```
MOVE X:(R0+N0), X0    Y:(R4), Y0    ; Befehl 1
MOVE X:(R0)+N0, X0    Y:R4, Y0      ; Befehl 2
MOVE X:R0+N0, X0      Y:-(R4),Y0    ; Befehl 3
```

Register	Inhalt		Register	Inhalt		Adresse	X-Speicher	Y-Speicher
R0:	\$0002		A2:	\$00		\$0005	\$900000	\$060000
M0:	\$0004		A1:	\$200000		\$0004	\$A00000	\$050000
N0:	\$0003		A0:	\$000000		\$0003	\$B00000	\$040000
R4:	\$0005					\$0002	\$C00000	\$030000
M4:	\$FFFF		X0:	\$400000		\$0001	\$D00000	\$020000
N4:	\$0010		Y0:	\$400000		\$0000	\$E00000	\$010000

Tabelle 8.1: Auszug aus der Anfangsbelegung von Registern und Speichern

- a) Vervollständigen Sie die Registerbelegung in Tabelle 8.2 nach der Ausführung der 3 Befehle!

Register	Inhalt nach Befehl 1	Inhalt nach Befehl 2	Inhalt nach Befehl 3
R0			
X0			
R4			
Y0			

Tabelle 8.2: Registerbelegung nach Ausführung der Befehle

**Fortsetzung der 8. Aufgabe:**

Die Register enthalten im Folgenden die Anfangsbelegung gemäß Tabelle 8.1. Betrachtet wird nun die folgende MAC-Operation:

MAC X0, Y0, A

- b) Geben Sie das in A stehende Ergebnis nach Ausführung der MAC-Operation an! Beachten Sie, dass der DSP 56000 mit Fraktalzahlendarstellung arbeitet! Geben Sie die Zwischenschritte Ihrer Lösung an!

A2	A1	A0
\$	\$	\$

## Aufgabe 9: Simulation von VHDL-Modellen

12 Punkte

Gegeben sei der folgende VHDL-Code, der näher untersucht werden soll.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity exam is
port (clk    : in  std_logic;
      reset  : in  std_logic;
      a, b   : in  integer range 0 to 127;
      c: out integer range 0 to 145160);
end exam;

architecture exam_arch of exam is

signal d: integer range 0 to 145160;

begin

process(clk, reset)
begin

    if reset = '0' then
        d      <= 0;
        result <= 0;
    else
        if clk'event and clk = '0' then
            d <= a*b+d;
        end if;
    end if;
end process;

c <= d;

end exam_arch;
```

- a) Welche Funktionseinheit wird durch den VHDL-Code beschrieben?

### Fortsetzung der 9. Aufgabe:

Im Folgenden soll schrittweise eine Testbench zum Test des VHDL-Modells mit einem Simulator entworfen werden.

- b) Entwerfen Sie die Entity der Testbench mit dem Namen *test*. Was ist bei der Entity einer Testbench zu beachten?

- c) Gegeben sind die Verläufe der Eingangssignale nach Abbildung 9.1, welche die Testbench generieren soll. Ergänzen Sie die folgende Architecture der Testbench so, dass die in Abbildung 9.1 dargestellten Eingangssignale erzeugt werden! Instantiieren Sie mittels *port map* zudem die bereits deklarierte Komponente *exam*!

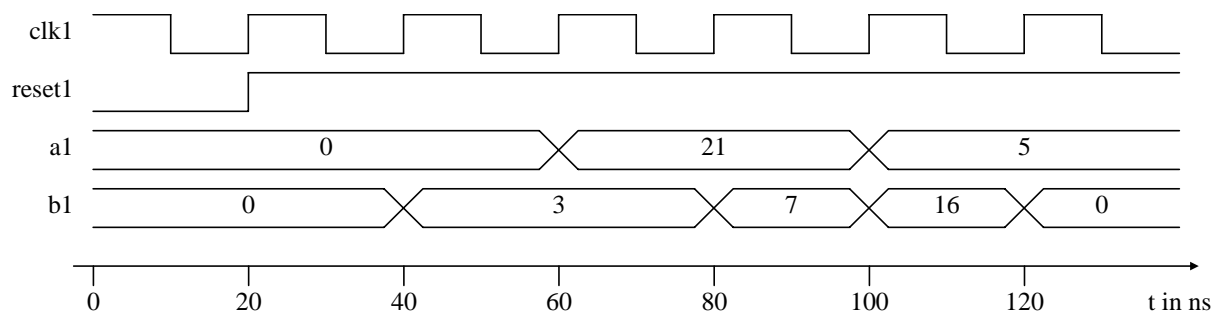


Abbildung 9.1: Timingdiagramm der Eingangssignale

```
architecture test_arch of test is

    signal clk1 : std_logic;
    signal reset1: std_logic;
    signal a1, b1: integer range 0 to 127;
    signal c1 : integer range 0 to 145160;

    component exam
    port (clk : in std_logic;
          reset : in std_logic;
          a, b : in integer range 0 to 127;
          c: out integer range 0 to 145160);
    end component;
```

**Fortsetzung der 9. Aufgabe:**

begin

```
process
begin
clk1 <= '1';
wait for 10 ns;
clk1 <= '0';
wait for 10 ns;
end process;

reset1 <= '0',
        '1' after 20 ns;

a1 <=
```

b1 <=

end test\_arch;

- d) Berechnen Sie das Ergebnis  $d$ , das nach Ablauf der Simulation mit den Eingangssignalen nach Abbildung 9.1 zur Verfügung steht!

**Fortsetzung der 9. Aufgabe:**

e) Vervollständigen Sie das Signal *c1* des nachfolgenden Timingdiagramms in Abbildung 9.2, wenn die Eingangssignale entsprechend Abbildung 9.1 gewählt werden! Was ist bei der Zuweisung von Signalen zu beachten?

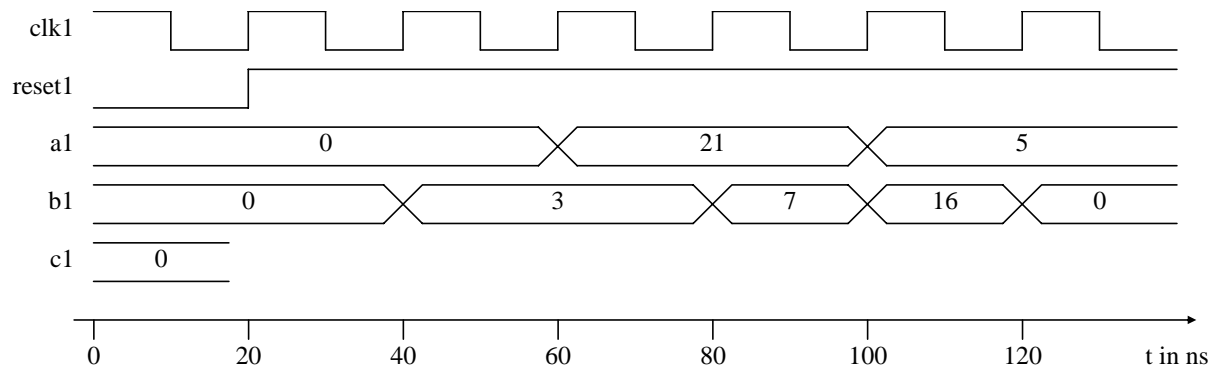


Abbildung 9.2: Timingdiagramm der Eingangssignale

## Aufgabe 10: Logikbeschreibung mit VHDL

9 Punkte

Der nachfolgend gegebene VHDL-Code ist zu untersuchen.

```
entity gatter is
port(a, b: in bit;
      c: out bit)
end gatter;

architecture g1 of gatter is
begin
c <= '1' when a = '0' and b = '0' else
      '0' when a = '0' and b = '1' else
      '0' when a = '1' and b = '0' else
      '1' when a = '1' and b = '1' else
      '0';
end g1;
```

- a) Wie viele Ein- und Ausgänge hat diese Schaltung und von welchem Typ sind die Ein- und Ausgänge?

- b) Geben Sie die Logiktablelle und die logische Verknüpfungsgleichung der Schaltung an!

Gegeben ist folgendes Komplexgatter:

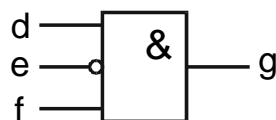


Abbildung 10.1: Komplexgatter mit dem Namen „k\_gatter“

**Fortsetzung der 10. Aufgabe:**

- c) Vervollständigen Sie den nachfolgenden VHDL-Code derart, dass das in Abbildung 10.1 angegebene Komplexgatter damit beschrieben wird!

```
entity k_gatter is
port (d, e, f: IN  bit;
      g      : OUT bit);
end k_gatter;

architecture g2 of k_gatter is
begin
```

```
end g2;
```