

# Fachprüfung Digitaltechnik

28. Februar 2017, Uhrzeit: 11:00 - 12:30 Uhr

Dr.-Ing. Wolfgang Heenes



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Name (Nachname, Vorname)	
Matrikelnummer	
Unterschrift	
Prüfung Bitte ankreuzen	<input type="checkbox"/> Bachelor of Science – Informatik <input type="checkbox"/> Diplom – Informatik <input type="checkbox"/> Bachelor of Computational Engineering <input type="checkbox"/> Bachelor of Education – Informatik <input type="checkbox"/> Bachelor Informationssystemtechnik <input type="checkbox"/> Sonstiger Abschluss:
Anzahl abgegebener Zusatzblätter:	

Aufgabe	Punkte	Erreicht
1	10	
2	10	
3	8	
4	12	
5	10	
6	8	
7	10	
8	10	
9	12	
Summe	90	
Note		

## Hinweise:

Überprüfen Sie zunächst, ob Ihre Klausur die Seiten 1 bis 11 besitzt.

Füllen Sie zuerst das Deckblatt aus und halten Sie Studenausweis und Lichtbildausweis bereit.

Sie sollten die Lösungen möglichst direkt in die Klausur eintragen. Reicht der vorhandene Platz nicht aus, so können Sie zusätzliche Blätter verwenden, die Sie zuerst mit Nachname, Vorname und Matrikelnummer kennzeichnen.

Nicht mit Bleistift schreiben und keine roten oder grünen Stifte verwenden.

Erlaubte Hilfsmittel: **keine**

**Viel Erfolg!**

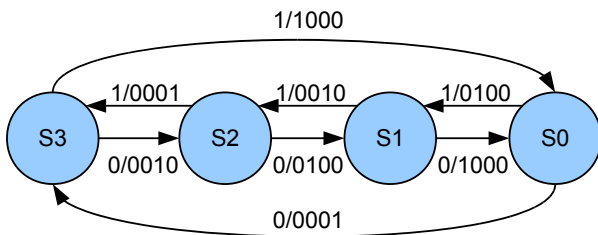
---

## Aufgabe 1: Grundlagen und Begriffe - 10 Punkte

1. Erläutern Sie das **Moore'sche Gesetz**.
2. Welcher Zusammenhang besteht zwischen dem dynamischen Leistungsumsatz ( $P$ ), der Spannung ( $U$ ) und der Taktfrequenz ( $f$ ) bei CMOS-Schaltungen?
3. Die Größe eines Chips (Chip-Fläche) hat einen Einfluss auf die Ausbeute von einem Waver. Erläutern Sie diesen Einfluss. Nennen Sie ein Beispiel für einen Chip, bei dem eine von acht identischen Komponenten defekt sein darf und der trotzdem seinen Einsatz in einem Computersystem findet.
4. Skizzieren Sie das Schichtenmodell eines Computers (wie in der Vorlesung vorgestellt) und erläutern Sie den Vorteil, den Schichtenmodelle im Allgemeinen bieten.
5. Hardware-Beschreibungssprachen (SystemVerilog, Verilog HDL, VHDL) dienen der Beschreibung und Spezifikation von technischen Systemen. Wie nennt man den Vorgang der eine Hardware-Realisierung erzeugt?
6. In der Digitaltechnik-Vorlesung bedeutet die Abkürzung **HDL**?
7. Gegeben ist die Zahl 113 im Dezimalsystem. Geben Sie die Zahl im Binärsystem an (inkl. Lösungsweg).
8. Gegeben ist die Zahl 0111 0011 im Binärsystem. Geben Sie die Dezimalzahl an (inkl. Lösungsweg).

## Aufgabe 2: Mealy-Automat - 10 Punkte

Gegeben ist folgender Zustandsgraph.



1. Beschreiben Sie den Automaten in SystemVerilog. Achten Sie auf die Trennung von kombinatorischer Logik und Speicherelementen bei der Beschreibung. Als Zustandskodierung verwenden Sie eine One-Hot Kodierung. Zur Beschreibung verwenden Sie bitte die symbolischen Bezeichner der Zustände (S0 - S3). Der Reset soll asynchron implementiert werden und auf die negative Flanke reagieren. Startzustand ist der Zustand S0.

2. Wieviele Flip-Flops benötigt man, um einen Zustandsautomaten mit  $n$  Zuständen in One-Hot Kodierung zu realisieren?

### Aufgabe 3: Synthese, FPGA/ASIC - Technologie - 8 Punkte

1. Gegeben ist folgendes SystemVerilog Modul:

```
module zaehler(  
    input clock,  
    input up_down,  
    output reg [3:0] qa  
);  
    initial qa = 4'b0;  
    integer direction;  
    always_ff @(posedge clock) begin  
        if (up_down) begin  
            direction = 1;  
        end else begin  
            direction = -1;  
        end  
        qa <= qa + direction;  
    end // end always  
endmodule
```

Welches SystemVerilog Konstrukt ist nicht synthesesfähig? Wie müssen Sie das SystemVerilog Modul erweitern, damit der Zähler in Hardware die gleiche Funktionalität hat, wie in der Simulation?

2. Nennen Sie mindestens zwei Design-Constraints der Logiksynthese.
3. Welches sind die logischen Grundelemente eines FPGAs? Skizzieren Sie die Grundelemente und die wichtigsten Verbindungen.
4. Was versteht man unter einem Softcore-Prozessor? Nennen Sie einen Vorteil und einen Nachteil gegenüber einer ASIC-Realisierung.

## Aufgabe 4: Boole'sche Algebra, Logikminimierung - 12 Punkte

1. Vereinfachen Sie folgenden boole'schen Ausdruck durch Anwendung von Axiomen und Theoremen.  

$$Y = A(AB + ABC)$$

2. Vereinfachen Sie folgenden boole'schen Ausdruck durch Anwendung von Axiomen und Theoremen.  

$$Y = \overline{(A + \overline{BD})C}$$

3. Gegeben ist folgende boole'sche Funktion (in Tabellenform) und folgendes K-Diagramm welches Sie zur Logikminimierung nutzen sollen. Geben Sie die minimale Funktion an. Markieren Sie die Primimplikanten im K-Diagramm.

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Y CD \ AB	AB			
	00	01	11	10
00				
01				
11				
10				

Geben Sie die minimale Funktion an: Y =

---

## Aufgabe 5: Speicherelemente - 10 Punkte

1. Beschreiben Sie kurz den Unterschied zwischen einem Latch und einem Flip-Flop.
2. Skizzieren Sie den Aufbau eines SR-Latches. Zur Verfügung stehen die auf dem Hilfsblatt angegebenen Gatter. Bezeichnen Sie die Eingänge und Ausgänge.
3. Benennen Sie die Zeitvorgaben, die eingehalten werden müssen, damit ein Flip-Flop richtig funktioniert. Was passiert, wenn diese Zeitvorgaben **nicht** eingehalten werden? Wie nennt man diesen Zustand?
4. Implementieren Sie in SystemVerilog ein ladbares 4-Bit Register, welches asynchron rücksetzbar ist. Das Register reagiert auf positive Taktflanken.

## Aufgabe 6: Zustandsübergangstabelle - 8 Punkte

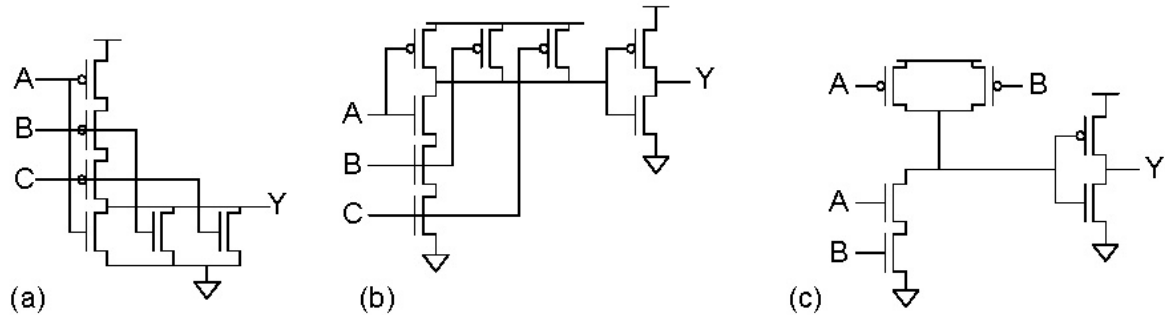
Gegeben ist eine Zustandsübergangstabelle:

Zustand		Eingang			nächster Zustand	
$S_1$	$S_0$	Start	B-ready	T-ready	$S'_1$	$S'_0$
0	0	0	X	X	0	0
0	0	1	X	X	0	1
0	1	X	0	X	0	1
0	1	X	1	X	1	0
1	0	X	X	0	1	0
1	0	X	X	1	0	0

Realisieren Sie diese Zustandsübergangstabelle. Verwenden Sie dazu D Flip-Flops und die notwendigen Gatter.

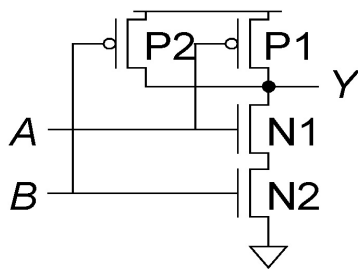
## Aufgabe 7: CMOS-Gatter - 10 Punkte

- Im Folgenden wird ein kombinatorischer Logikbaustein mit drei Eingängen betrachtet, der die Funktion NOR realisiert. Welche der angegebenen Realisierungen mit Feldeffekttransistoren stellt die Funktionalität eines NOR-Gatters da.



Antwort:

- Die Realisierung eines Gatters mit Feldeffekttransistoren ist in folgender Abbildung gegeben:



Geben Sie in der folgenden Tabelle für alle möglichen Werte von A und B an, ob die Feldeffekttransistoren „ein- oder ausgeschaltet“ sind. Einschaltet wird durch die Angabe einer 1 in der Tabelle gekennzeichnet, Ausgeschaltet durch eine 0. Welche logische Funktion erfüllt das Gatter? Tragen Sie die logischen Ausgangswerte ebenfalls in die Tabelle ein.

A	B	P1	P2	N1	N2	Y
0	0					
0	1					
1	0					
1	1					



### Aufgabe 8: Modellrechner - 10 Punkte

Betrachtet wird der in der Vorlesung vorgestellte Modellrechner (s. Hilfsblatt).

1. Gegeben ist folgendes Assemblerprogramm.

```
0000 0000 : .....
0000 0101 : .....
0100 0000 : .....
0000 0111 : .....
1101 0000 : .....
0000 0000 : .....
```

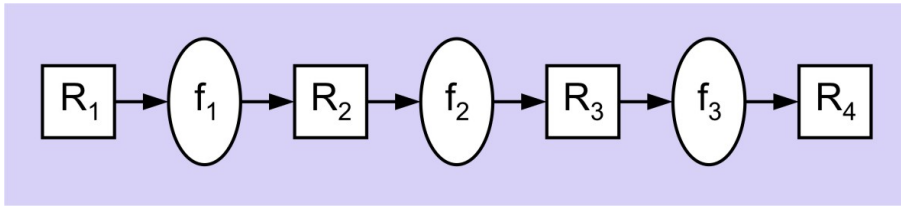
Disassemblieren Sie das Programm und geben Sie die Mnemonics und Operanden an.

2. Beschreiben Sie kurz die Wirkung des Programms. Welche Probleme würde es bei einer wiederholten Ausführung geben?

3. Geben Sie alle Steuersignale (Mikrooperationen) für Speicher, Multiplexer und Register an, welche bei der *Abarbeitung* (inklusive Befehl holen und dekodieren) des Befehls **add #5** verwendet werden.

## Aufgabe 9: Register-Transfer-Logik - 12 Punkte

Gegeben ist folgende Pipeline in Register-Transfer-Logik:



Die kombinatorische Logik zwischen den Registern soll folgende Funktionen realisieren.

- $f_1$ : vervierfachen
- $f_2$ : plus 3
- $f_3$ : quadrieren

Die Pipeline berechnet also  $R_4 = (4 \cdot R_1 + 3)^2$ .

1. Beschreiben Sie die Pipeline in SystemVerilog. Die Funktionen sollen als `function` realisiert werden. Die Register R1, R2, R3 und R4 sind jeweils 16-Bit breit. Das Überlaufen der Register kann vernachlässigt werden. Wenn ein Steuersignal `ldreg1` gesetzt ist, soll über den Port `in` von außen ein Wert in das Register R1 geladen werden. Das Ergebnis der Berechnung soll über den Port `out` nach außen geführt werden.

Die Pipeline soll außerdem einen asynchronen Reset-Eingang `areset` haben, mit dem die Pipeline zurückgesetzt werden kann. Wird `areset` gesetzt, soll die Berechnung sofort gestoppt werden und solange keine neue Berechnung begonnen werden, bis `areset` wieder den Wert 0 annimmt. Da die Pipeline einige Takte benötigt um das Ergebnis zu berechnen, liegen am Ausgang zwischenzeitlich falsche Werte an. Das Module soll deshalb ein Signal `valid` besitzen. `valid` soll genau dann den Wert 1 haben, wenn der Wert an `out` ein korrektes Ergebnis eines geladenen Wertes ist.

