

Vorlesungsprüfung aus Digitales Design

2. Juni 2015

Die Arbeitszeit beträgt 1,5 Stunden. Als Hilfsmittel sind ausnahmslos Schreibzeug, Lineal und (nicht programmierbarer) Taschenrechner erlaubt. Schreiben Sie Ihre Antworten und Lösungen (inkl. Lösungsweg!) mit Füllfeder oder Kugelschreiber (nicht rot, KEIN Bleistift!) und streichen Sie alles durch, was nicht zur Beurteilung herangezogen werden soll. Ein Abbruch der Prüfung nach Erhalt der Angaben führt in jedem Fall zu einer Beurteilung.

Tragen Sie Namen, Kennzahl und Matrikelnummer **zu Beginn** der Prüfung in die Tabelle ein und beschriften Sie jedes Blatt, das Sie abgeben möchten rechts oben mit Namen und Matrikelnummer.

Familienname:	Vorname:
Kennzahl:	Matrikelnummer:

Viel Erfolg!

Die nachfolgende Tabelle nicht beschriften!

Beispiel		Mögliche Punkte	Erhaltene Punkte
Theoriefragen		48	
Rechenbeispiel	1	18	
	2	18	
	3	12	
Gesamt		96	

Theoriefragen

Frage 1:

Nennen Sie 5 Vorteile der Digitalisierung von Signalen bzw. Information!

Frage 2:

Sie wollen ein Logiksignal vom Ausgang eines Gatters G1 (Ausgangswiderstand $R = 100\Omega$) zum Eingang eines Gatters G2 (Eingangskapazität $C = 0,1\text{pF}$) leiten (Leitung ideal). Schätzen Sie ab, welches Delay sich durch die RC-Konstante ergibt!

Frage 3:

Was besagt das Theorem von Shannon? Wo wird es praktisch verwendet?

Frage 4:

Was versteht man unter Bonding?

Frage 5:

Erläutern Sie, wie physikalische Defekte zu dynamischen Fehlern führen können!

Frage 6:

Wozu verwendet man eine Antifuse?

Frage 7:

Was versteht man unter infant mortality?

Frage 8:

Was besagt die „Rule of Ten“?

Beispiel 1

Entwerfen Sie einen 16-bit Pseudo-Zufallsgenerator mit folgenden Schritten:

- (a) Stellen Sie den Aufbau eines 16-bit Schieberegisters aus Flip-Flops dar. Schließen Sie jeweils Takteingang, Dateneingang und Ausgang korrekt an!
- (b) Ergänzen Sie diese Schaltung geeignet zu einem LFSR mit dem Polynom
$$X^{16} = X^8 \oplus X^4 \oplus X^3 \oplus X^0$$
(Hinweis: verwenden Sie einen Block namens „odd Parity“ bzw. „even Parity“ für die Realisierung eines XOR mit mehr als 2 Eingängen)
- (c) Stellen Sie dar, wo man die Zufallszahl abgreifen kann
- (d) Was ist der Unterschied zwischen diesem Pseudo-Zufallsgenerator und einem „echten“ Zufallsgenerator (bezogen auf die gelieferten Zahlenfolgen am Ausgang)?
- (e) Nehmen Sie an, der Generator sei auf den Wert „0110110100111100“ initialisiert. Welchen Wert hat der Ausgang nach 3 Taktzyklen (= aktiven Taktflanken)?

Rechenbeispiele

Beispiel 1

Gegeben ist die Multiplexer-basierte Implementierung einer Funktion $F(A,B,C)$ laut Abbildung 1.1.

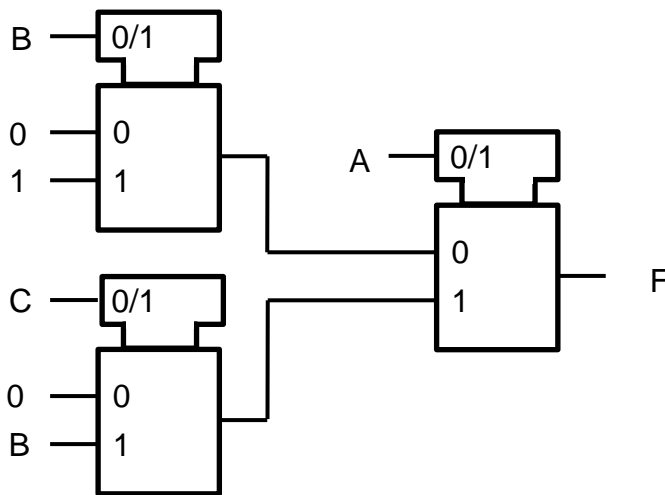


Abbildung 1.1: Implementierung der Funktion F mittels Multiplexern

- Ermitteln Sie *anhand der Schaltung* (nicht Wahrheitstabelle) die Funktion $F(A,B,C)$!
- Zeichnen Sie das KV-Diagramm und geben sie die *konjunktive* Normalform an!
- Geben Sie für die verwendeten Multiplexer eine Realisierung mittels Transmission Gates und Invertern an (mit geeigneter Entkopplung von Eingang und Ausgang)! Wie viele Transistoren benötigt ein Multiplexer, wie viele die gesamte Implementierung von $F(A,B,C)$ lt. Abbildung 1.1 (für interne Verbindungen können Sie die Entkopplung nun einsparen)?
- Entwerfen Sie eine alternative Realisierung von $F(A,B,C)$ mittels AOI bzw. OAI! Welche Variante (OAI oder AOI) ist günstiger und warum?
- Stellen Sie den Aufbau Ihrer Lösung als Transistorschaltung aus p-Stack und n-Stack dar (vergessen Sie nicht, den Ausgang zu kennzeichnen)! Falls Sie Inverter benötigen, geben Sie einmal den Aufbau eines Inverters aus Transistoren an, und verwenden Sie in Ihrer Schaltung für $F(A,B,C)$ zur besseren Übersichtlichkeit nur das Invertersymbol.
- Wie viele Transistoren benötigen Sie nun für die Implementierung von $F(A,B,C)$? Vergessen Sie nicht, die Transistoren für die Inverter zu berücksichtigen!

Beispiel 2

Gegeben ist der in Abbildung 2.1 dargestellte Synchronizer mit 500MHz Takt (f_{clk}). Ein asynchrones Eingangssignal (f_{input}) mit 93MHz liegt am Eingang von Flip-Flop $FF1$. Der Ausgang von $FF1$ geht direkt an den Eingang von Flip-Flop $FF2$. Laut Datenblatt haben die beiden Flip-Flops folgende Parameter:

setup time $t_{SU} = 220\text{ps}$; hold time $t_H = 160\text{ps}$; clock to output delay $t_{CO} = 350\text{ps}$;
 metastability characteristics: $\tau_C = 50\text{ps}$; $T_0 = 100\text{ps}$

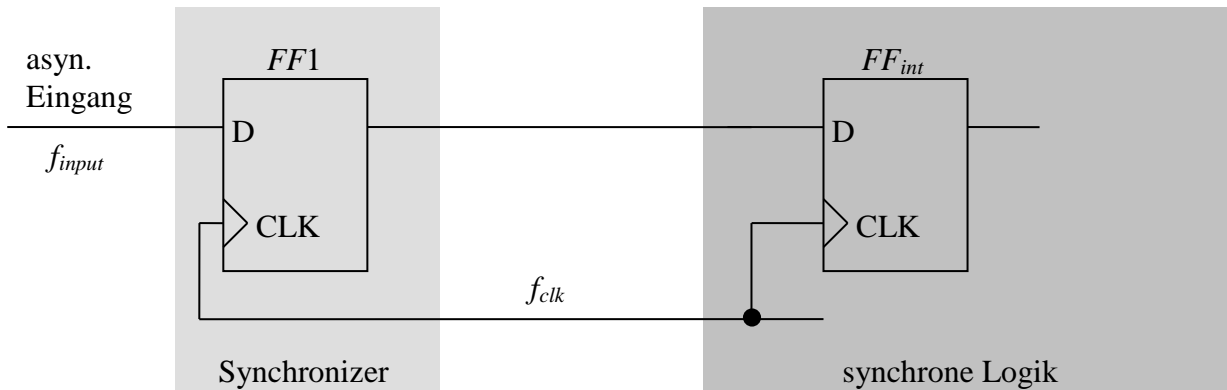


Abbildung 2.1

(a) Wie groß ist die Resolution Time für FF_{int} ? Welche MTBU ist zu erwarten?

Zur Erhöhung der MTBU soll der Synchronizer zweistufig gemacht werden, indem ein weiteres Flip-Flop, $FF2$, mit identischen Eigenschaften zwischen den Ausgang von $FF1$ und den Eingang von FF_{int} geschaltet wird. Damit ergibt sich die in Abbildung 2.2 dargestellte Schaltung.

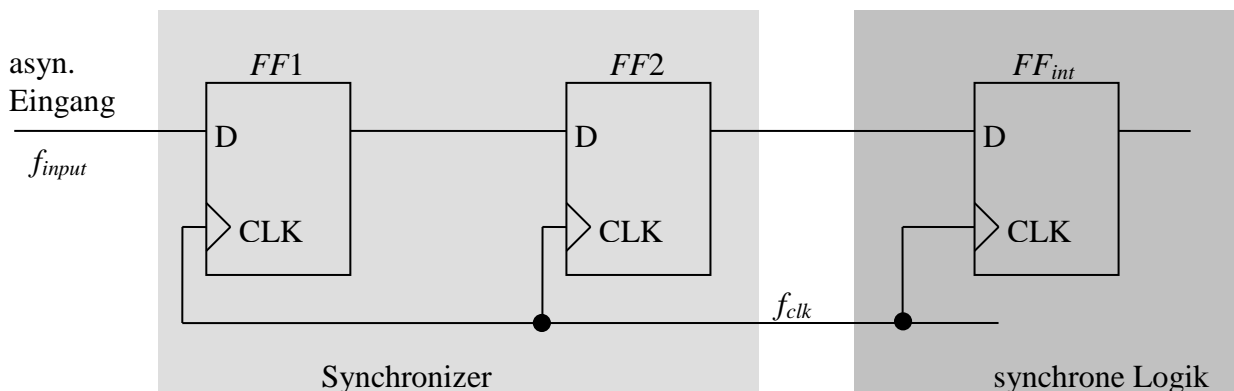


Abbildung 2.2

- (b) Welche Resolution Time und MTBU ergeben sich mit diesem neuen Synchronizer?
- (c) Ihr Chef möchte, dass Sie $FF2$ mit invertiertem Takt ansteuern, damit die Latenzzeit des Synchronizers vermindert wird. Er argumentiert, dass seine Lösung zwar schlechtere MTBU aufweist als die nach Abb. 2.2, aber immer noch bessere als jene nach Abb. 2.1. Ist das korrekt? Begründen Sie ihre Antwort!

Rechenbeispiele

Beispiel 1

Gesucht ist eine Schaltung, für ein 2-aus-3 Threshold Gate mit Hysterese: Sie soll den Ausgang (y) genau dann auf 1 setzen, wenn an zumindest zwei der drei Eingänge (a,b,c) logisch 1 anliegt. Der Ausgang wird genau dann auf 0 gesetzt, wenn an allen drei Eingängen logisch 0 anliegt. Für alle anderen Eingangskombinationen wird der letzte gesetzte Wert gehalten (Hysterese).

- (a) Da die Schaltung ein „Gedächtnis“ für den letzten gesetzten Wert benötigt, brauchen Sie ein Speicherelement. Verwenden Sie dafür ein SR-Latch mit Beschaltung laut Abbildung 1.1. Erstellen Sie KV-Diagramm und DNF für die kombinatorischen Blöcke „set“ und „reset“!

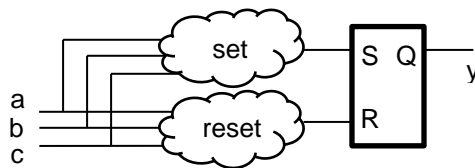


Abbildung 1.1

Nehmen Sie für (b) bis (e) an, Sie hätten zur Implementierung nur NOR-Gatter (mit bis zu 4 Eingängen) und Inverter zur Verfügung.

- (b) Wie würden Sie das SR-Latch mit NOR Gattern realisieren (Schaltplan)?
- (c) Beim SR-Latch gibt es eine illegale Eingangskombination für R und S. Welche ist dies und wie verhält sich Ihr Latch dann?
- (d) Kann diese Kombination in Ihrer Implementierung des Threshold Gate auftreten? Falls ja, für welches (a,b,c)? Falls nein, warum nicht?
- (e) Betrachten Sie die kombinatorische Funktion für die „set“ Logik: Wie können Sie diese nur mit NOR und Invertern realisieren (Schaltplan)?
- (f) Ihr Chef hat vor seinem Urlaub die in Abb. 1.2 dargestellte Implementierung für die set-Logik skizziert. Beschalten Sie die Eingänge e1 ... e4 passend!

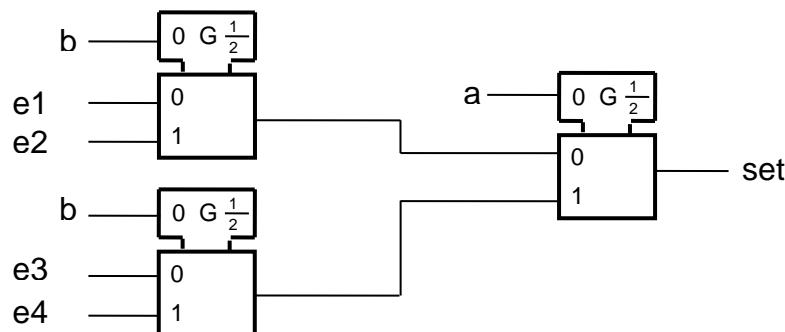


Abbildung 1.2

- (g) Erarbeiten Sie eine alternative Implementierung der gesamten Threshold-Funktion mittels LUT (ohne Verwendung eines SR-Latch!): Erstellen Sie dazu die Wahrheitstabelle $y(a,b,c,y')$.
- (h) Wie viele Eingänge und Ausgänge benötigt die LUT, wie werden diese beschaltet (Skizze)? Geben Sie den Speicherinhalt der LUT an!

Beispiel 3

Gegeben ist die Schaltung in Abbildung 3.1.

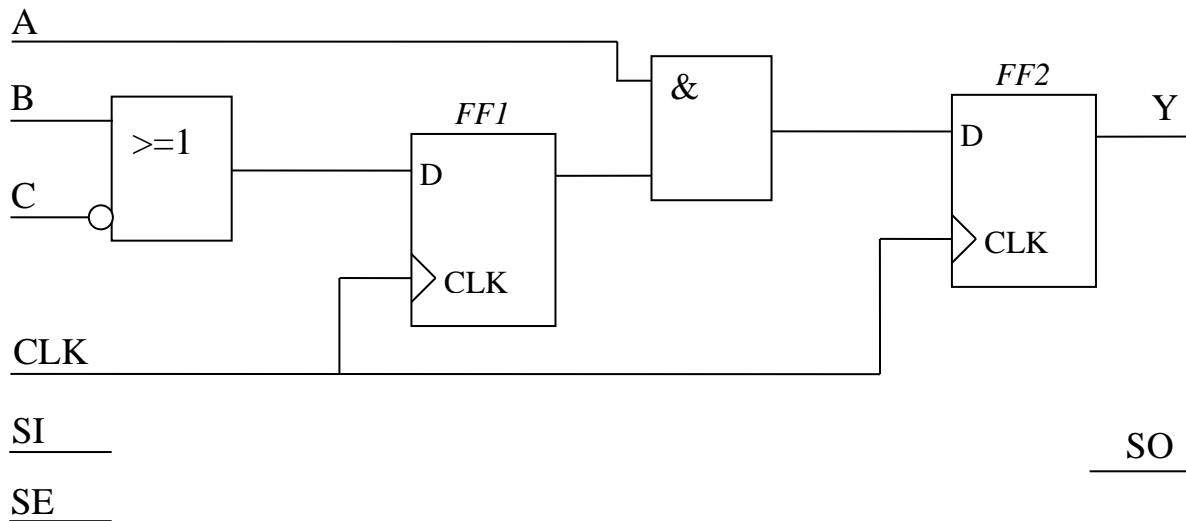


Abbildung 3.1

- Wie müssen Sie ein D-Flip-Flop erweitern, um daraus ein Scan-Register zu machen? Skizzieren Sie diese Erweiterung!
- Erweitern Sie die beiden Flip-Flops in Abbildung 3.1 entsprechend!
- Bilden Sie mit den beiden so entstandenen Scan-Registern eine Scan Chain. Schließen Sie die Signale SI (scan in), SO (scan out) und SE (scan enable) entsprechend an.
- Betrachten Sie das AND-Gate: Welche Testmuster müssen Sie an dieses Gatter anlegen, um alle Stuck-at Fehler an seinen Eingängen und seinem Ausgang zu finden?
- Beschreiben Sie den Ablauf eines Scan Tests anhand der Schaltung für das Beispiel eines SA0 am unteren Eingang des AND-gates.
 - Welchen Pegel müssen Sie dort anlegen um den Fehler zu aktivieren?
 - Welche Sequenz von Signalen an SE, SI und CLK benötigen Sie, um das zu erreichen?
 - Welche Sequenz von Signalen müssen Sie als nächstes anlegen, um den Fehler an SO sichtbar zu machen?