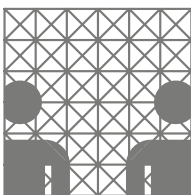


Praktikum: VLSI-Entwurf

3

Full-Custom Design Ein bitserieller Vergleicher



Andreas Mäder

Universität Hamburg – MIN – Fachbereich Informatik
Arbeitsbereich Technische Aspekte Multimodaler Systeme

<http://tams.informatik.uni-hamburg.de>

In diesem Versuch werden Sie einen Vergleicher für zwei Zahlen aufbauen, die bitseriell an den Eingängen der Schaltung eingelesen wird. Dazu wird in einem ersten Schritt eine Automatenbeschreibung erstellt, die dann als Schaltung in TSPC-Technik zu entwerfen ist.

Voraussetzungen

- Grundwissen zu den Stichworten: boole'sche Algebra, Automatenentwurf
- Die Schaltungen aus den Versuchen 1 und 2.

Logische Struktur eines bitseriellen Vergleichers

Es soll eine Zelle entworfen werden, die zwei vorzeichenlose k-bit Zahlen x und y vergleicht und das Ergebnis 0 liefert, wenn $x > y$ ist und 1, wenn $x \leq y$ ist. Die Operanden werden bitseriell mit dem höchstsignifikanten Bit zuerst eingelesen, wobei ein Bit pro Takt verarbeitet wird. Vor der Ausführung des Vergleichs soll der Automat durch Eingabe eines Reset-Signals res auf einen definierten Anfangszustand gesetzt werden können.

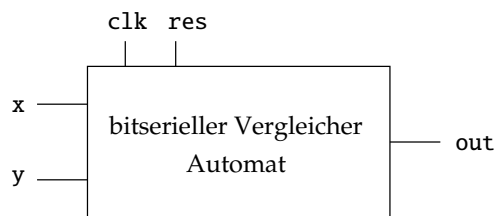


Abbildung 1: Bitserieller Vergleicher

Betrachten wir die gewünschte Funktion des Vergleichers als deterministischen Automaten. Dieser habe drei Zustände, codiert in zwei Bits G und K (G für größer, K für kleiner).

- $K=0, G=0 : x = y$ oder die bisher betrachteten Prefixe der Eingaben von x und y weisen noch keinen Unterschied auf.
- $K=0, G=1 : x > y$
- $K=1, G=0 : x < y$
- $K=1, G=1 : \text{—}$ dieser Zustand wird nicht eingenommen, er ist redundant.

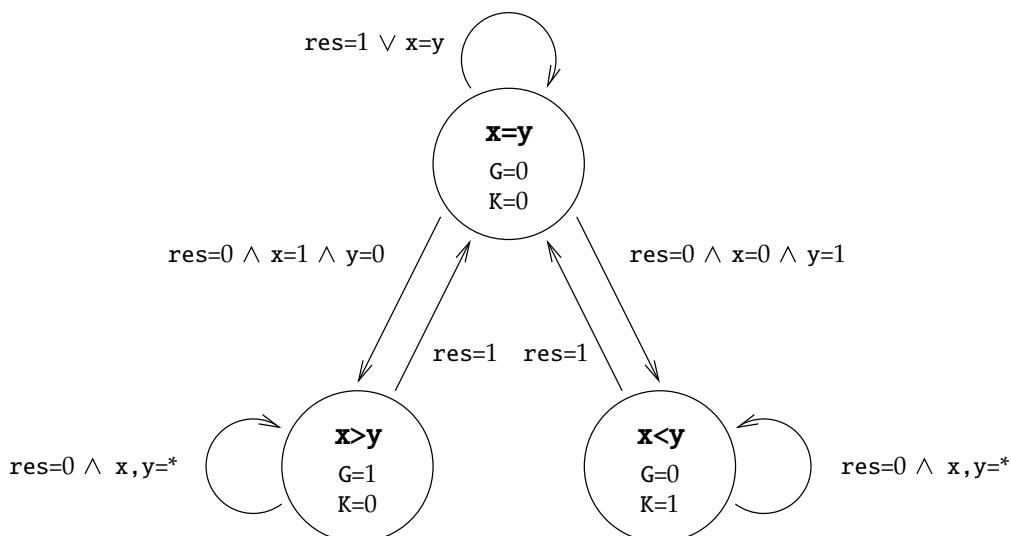


Abbildung 2: Zustandsübergangsgraph des Vergleicherautomaten

Der Zustand $G = K = 0$ ist der Anfangszustand: durch Eingabe des Reset-Signals wird der Vergleicher in diesen Zustand gebracht. Sobald sich das erste Mal ein zu x gehöriges Bit x_n von dem entsprechenden zu y gehörigen Bit y_n unterscheidet, wird einer der beiden anderen Zustände angenommen. Danach bleibt der Automat in diesem Zustand, es ist entschieden, welche Zahl größer war. Erst bei Eingabe des nächsten Reset-Signals wird wieder der Anfangszustand erreicht. Folgende Tabelle beschreibt noch einmal die Übergangsfunktion.

res	x	y	G_{alt}	K_{alt}	G_{neu}	K_{neu}
1	*	*	*	*	0	0
0	*	*	0	1	0	1
0	*	*	1	0	1	0
0	*	*	1	1	*	*
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	1	0
0	1	1	0	0	0	0

Um ein getaktetes Schaltwerk zu erhalten, müssen die Rückkopplungen über Taktglieder laufen — K_{neu} bzw. G_{neu} wird im nächsten Takt zu K_{alt} bzw. G_{alt} . Hierzu können die Flipflops aus Versuch 2 benutzt werden.

Man beachte, dass die Länge der beiden Zahlen nicht festgelegt ist. Sie wird lediglich durch die Eingabefrequenz des Reset-Signals res bestimmt.

Arbeitsweise

1. Überlegen Sie sich die Schaltfunktionen für den Vergleicherautomaten.
2. Entwerfen Sie die Schaltung und Simulieren Sie den Entwurf.

Dabei haben Sie mehrere Möglichkeiten den (hierarchischen) Entwurf durchzuführen. Verschiedene Kombinationen von Schaltplaneingabe + Layoutsynthese (Versuch 2) und Layouteingabe (Versuch 1) sind möglich. Außerdem können die in den ersten Versuchen entworfenen (Teil-)Layouts benutzt werden.

3. Ein kompaktes Layout des vollständig simulierten Entwurfs ist das Gesamtergebnis dieses Versuchs.

Aufgaben

III-1 Überlegen Sie sich anhand der oben aufgezeigten Tabelle, bzw. des Zustandsübergangsgraphen, siehe Abbildung 2, die Funktionen δ (Zustandsübergangsfunktion) und λ (Ausgabefunktion) eines Moore-Automaten.

III-2 Entwerfen Sie ein Layout für den **bitseriellen Vergleich** und überprüfen Sie die Funktion durch geeignete Simulation. Versuchen Sie die Schaltung möglichst klein und schnell zu machen.

Formen Sie dazu die Ausdrücke der Schaltfunktionen in eine Form um, die mit den Entwürfen aus den vorherigen Versuchen realisierbar ist. Noch fehlende Gatter können Sie mit dem Layout-Editor oder der Layoutsynthese erstellen.

Tipps:

- Falls das für die Realisierung günstiger sein sollte, kann der Reset-Eingang (res) als *active low* angenommen werden. Außerdem können andere Zustandscodierungen gewählt werden als die hier mit G und K vorgeschlagene Lösung.
- Um das Design kleiner und schneller zu machen, können Teile der Logik in das Register (siehe Versuch 2) integriert werden.
- Nutzen Sie die Symmetrie der Schaltfunktionen für einen hierarchischen Entwurf.

III-3 Wie müsste die Automaten aussehen, wenn für x und y , nicht vorzeichenlose, sondern 2-Komplement Zahlen eingegeben werden.

Zeichnen Sie den zugehörigen Zustandsübergangsgraphen auf.