

SCHALTWERKE

Grundidee der Schaltwerke

Schaltwerke unterscheiden sich von Schaltnetzen durch Rückkopplungen. Während Schaltnetze nur kombinatorische Logik abbilden, besitzen Schaltwerke einen internen Zustand, der über die Zeit erhalten bleiben kann. Mit serieller Verarbeitung werden Informationen nacheinander verarbeitet, nicht parallel.

Schaltwerke bilden die Grundlage für sequenzielle Logik und sind essenziell für Speicher, Zähler und Steuerwerke.

Zustandsgraphen

Ein Zustandsgraph ist ein gerichteter Graph mit:

- **Knoten**: Repräsentieren Zustände des Systems.
- **Kanten**: Zeigen Zustandsübergänge an, ausgelöst durch Eingaben.

Race Conditions

Race Conditions entstehen, wenn mehrere Signale gleichzeitig umschalten und der Endzustand undefiniert wird. Dies kann zu metastabilen Zuständen führen, die das Systemverhalten unvorhersehbar machen.

Vermeidung von Race Conditions

Durch Einführung zusätzlicher Zustände und eines Taktsignals können Race Conditions vermieden werden. Der Takt synchronisiert alle Zustandsübergänge, sodass sich nur ein Signal pro Taktperiode ändert. Änderung des Hamming-Abstand 1.

Diese Methode führt zu synchronen Schaltwerken, die deterministisches Verhalten garantieren.

Taktsignale, Clock und Flankensteuerung

Das Taktsignal oder auch Clock genannt steuert den Zeitpunkt von Zustandsänderungen:

- Steigende Flanke: $(0 \rightarrow 1)$ positive Flanke
- Fallende Flanke: $(1 \rightarrow 0)$ negative Flanke

Flankensteuerung ist robuster als Pegelsteuerung, da sie nur auf kurze Übergangszeitfenster reagiert.

Flipflops

Flipflops sind bistabile Multivibratoren, die einen Bit-Zustand speichern.

Bemerkung

Die Erfindung geht auf Eccles und Jordan im Jahre 1918 zurück.

SR-Flipflop, Set Reset-Flipflop

Das SR-Flipflop besitzt zwei Eingänge: S setzt den Ausgang Q auf 1, R setzt ihn auf 0. Der Zustand $S=R=1$ ist meist verboten, da er zu einem undefinierten Verhalten führen kann. Die charakteristische Gleichung lautet: $Q' = S + \overline{R} \cdot Q$.

D-Flipflop, Delay-Flipflop

Das D-Flipflop vereinfacht die Handhabung, indem es nur einen Dateneingang D besitzt. Bei der aktiven Taktflanke wird der Wert von D übernommen und am Ausgang Q gespeichert. Es umgeht das Problem des verbotenen Zustands des SR-Flipflops.

JK-Flipflop, Jump Kill-Flipflop

Das JK-Flipflop ist eine Weiterentwicklung, die die Vorteile von SR- und T-Flipflops kombiniert. J verhält sich wie S, K wie R. Der entscheidende Vorteil liegt im Verhalten bei $J=K=1$: In diesem Fall toggelt der Ausgang Q bei jedem Takt, das heißt, es wechselt in den jeweils anderen Zustand. Damit ist der bei SR-Flipflops problematische Eingangszustand hier sinnvoll genutzt.

T-Flipflop, Toggle-Flipflop

Ein T-Flipflop kehrt seinen Zustand immer dann um, wenn der T-Eingang aktiv ist $T=1$ und eine aktive Taktflanke auftritt. Es kann einfach aus einem JK-Flipflop realisiert werden, indem J und K miteinander verbunden werden also $J=K=T$.

Taktflankengesteuertes SR-Flipflop

Diese Variante reagiert nur, wenn das Taktsignal C aktiv ist. Es zeigt das Phänomen der Laufzeitverzögerung, oder auch Propagation Delay genannt, zwischen der Eingangsänderung und der sichtbaren Reaktion am Ausgang.

Anwendungen von Flipflops

Diese sind:

- Zähler:

Durch das Hintereinanderschalten oder Kaskadieren von Flipflops entstehen Zähler. Jede Zählerstufe teilt die Frequenz des ankommenden Taktsignals durch zwei. Ein 4-Bit-Zähler, realisiert mit T-Flipflops oder JK-Flipflops im Toggle-Modus, kann somit von 0 (binär 0000) bis 15 (binär 1111) zählen. Zähler sind unverzichtbar in nahezu jedem digitalen System, z.B. für Adressierung, Taktteilung oder Zeitmessung.

- Schieberegister:

In einem Schieberegister sind Flipflops in Reihe geschaltet. Bei jedem Taktimpuls wird der Wert eines Flipflops in das nächste weitergeschoben. Ein Linksshift entspricht einer Multiplikation der gespeicherten Binärzahl mit 2, ein Rechtsshift einer Division durch 2 Ganzzahldivision. Schieberegister werden für die Umwandlung zwischen parallelen und seriellen Datenformaten, zur Erzeugung von Pseudozufallsfolgen und in vielen Kommunikationsprotokollen eingesetzt.

- Serieller Addierer

Ein serieller Addierer führt die Addition zweier Binärzahlen Bit für Bit durch, beginnend mit dem niederwertigsten Bit genauer das least significant Bit, LSB genannt. Er verwendet einen Volladdierer, der die beiden Eingabebits und ein Übertragsbit, Carry-In, addiert und ein Summenbit sowie ein neues Übertragsbit, Carry-Out, erzeugt. Dieses Carry-Out wird in einem Flipflop zwischengespeichert und dient im nächsten Taktzyklus als Carry-In für die Addition der nächsten höherwertigen Bits. Dieses Verhalten lässt sich hervorragend durch einen Mealy-Automaten modellieren, bei dem der Zustand dem aktuellen Übertragsbit entspricht und die Ausgabe, Summenbit genannt, sowohl vom Zustand als auch von der aktuellen Eingabe abhängt.