

MSRT Ausarbeitung

Erklären Sie den Einsatz von Speicherelementen in logischen Schaltungen insbesondere in Schaltwerken. Beschreiben Sie beispielhaft wo und wie Schaltwerke zum Einsatz kommen können.

Übersicht – Schaltwerk

Schaltnetz: Der Ausgangszustand hängt nur vom momentanen Eingangszustand ab.

Schaltwerk: Der Zustand hängt zusätzlich vom Vorzustand ab. Um dies zu realisieren muss ein Speicherelement den Vorzustand zwischenspeichern.

Synchrones SW: Zustandsänderungen können nur im Taktraster stattfinden

Asynchrones SW: Zustandsänderungen können immer stattfinden (Problem: kurzfristig unerwünschte Zwischenwerte = Laufzeiteffekt)

Synchrones Schaltwerk Aufbau:

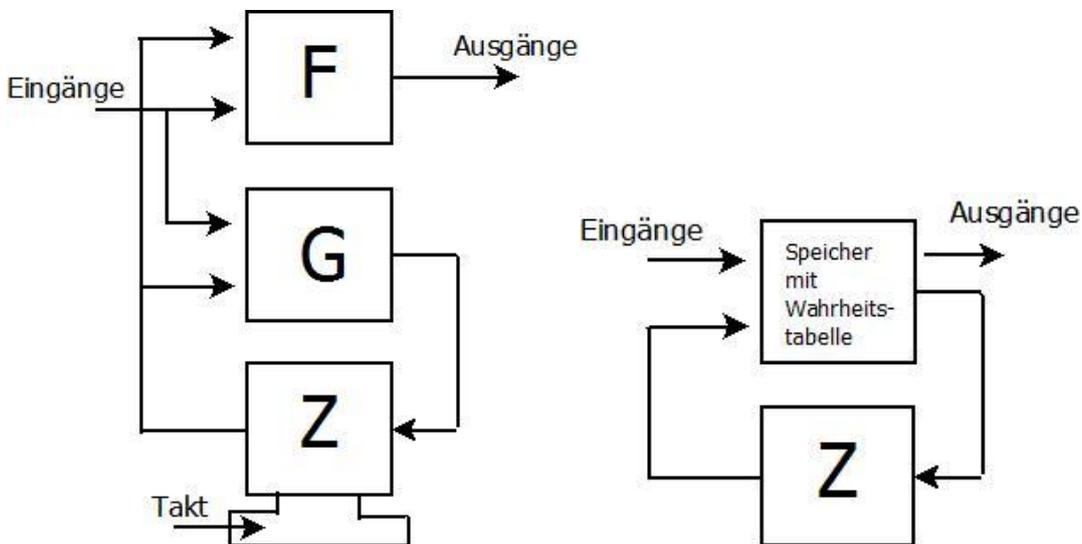
Hier sieht man, dass die Ausgangsfunktion (F) vom aktuellen Eingang und vom Vorzustand abhängt.

Im rechten Bild kann die einfachere Umsetzung mit einem EPROM (electrically programable read only memory) Speicherelement sehen. Hier wird keine Gatterschaltung benötigt.

Z ... Zustand (Speicherelement)

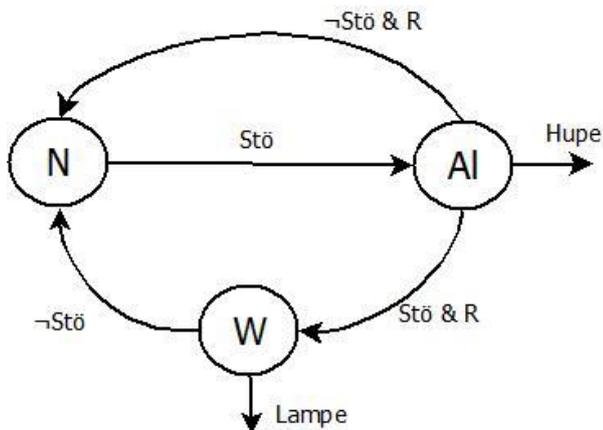
F ... Ausgangsfunktion: $F = f(\text{Zustand}, \text{Eingänge})$

G ... Übertragungsfunktion: $G = f(\text{Zustand}, \text{Eingänge})$



Beispiel Schaltwerk

Zustandsdiagramm: Anwendungsbeispiel eines Geräts



Störung	Stö
Reset	R
N	Normalbetrieb
AI	Alarmzustand
W	Wartungsbetrieb

1. Zustandskodierung: Hier wird festgelegt welcher Zustand welche Bitkombination bekommt
Nicht genutzte Kombinationen werden mit X gekennzeichnet. [Anzahl der Zustände $\leq 2^z$ -Bit]

Z1	Z0	Zustand
0	0	N
0	1	AI
1	0	W
1	1	X

2. Anfertigen einer Wahrheitstabelle.

Als erstes müssen alle möglichen Kombinationen der Eingänge/Zustände aufgeschrieben werden.

Aus diesen Kombinationen ergibt sich immer ein neuer Folgezustand (siehe Zustandsdiagramm)

Nun müssen noch die neuen Zustände der Ausgänge eingetragen werden (siehe Zustandskodierung)

Zustand	Z1	Z0	Stö	R	Folge	Z1neu (D3)	Z0neu (D2)	L (D1)	H (D0)	Adresse	Programm EPROM
N	0	0	0	0	N	0	0	0	0	0	0
	0	0	0	1	N	0	0	0	0	1	0
	0	0	1	0	AI	0	1	0	0	2	4
	0	0	1	1	AI	0	1	0	0	3	4
AI	0	1	0	0	AI	0	1	0	1	4	5
	0	1	0	1	N	0	0	0	1	5	1
	0	1	1	0	AI	0	1	0	1	6	5
	0	1	1	1	W	1	0	0	1	7	9
W	1	0	0	0	N	0	0	1	0	8	2
	1	0	0	1	N	0	0	1	0	9	2
	1	0	1	0	W	1	0	1	0	A	A
	1	0	1	1	W	1	0	1	0	B	A
	1	1	0	0	X	X	X	0	0	C	X
	1	1	0	1	X	X	X	0	0	D	X
	1	1	1	0	X	X	X	0	0	E	X
	1	1	1	1	X	X	X	0	0	F	X

Um später eine Logik-Schaltung entwickeln zu können muss man herausfinden wie die Funktionen der neuen Zustände und Ausgänge aussehen.

Hierzu verwendet man zb das Karnaugh-Veitch-Diagramm:

		Z0				
R	0	0	X	0	Stö	
	0	0	X	0		
	0	1	X	1		
	0	0	X	1		
		Z1				

		Z0				
R	0	1	X	0	Stö	
	0	0	X	0		
	1	0	X	0		
	1	1	X	0		
		Z1				

$$Z1_{neu} = R * Z0 * Stö + Stö * Z1$$

$$Z0_{neu} = \neg Z0 * Stö * \neg Z1 + Z0 * \neg R$$

$$Lampe = Z1 \quad Hupe = Z0$$

Diese Funktionen werden nun in eine Logikschaltung umgesetzt die hier zur Einfachheit halber in einer „Black-Box“ steckt.

Hier kann man gut sehen, dass die neuen Zustände immer in einem Speicherelement zwischengespeichert werden.

Rechts kann man die Umsetzung mit einem 16x4 (64b) EPROM sehen.

16 = Anzahl der Speicherplätze

4 = Ausgänge (Spalten)

