

Der Algorithmus zur Entwicklung der Schaltung kann aus der Aufgabenstellung entnommen werden. Als Input stehen 8 Bit zur Verfügung, wodurch sich ein Maximalwert von 255 ergibt. Die numerische Darstellung benötigt 12 Bit (drei BCD-Ziffern, je ein Nibble pro Segment). In den Diagrammen sind Ideen dargestellt, um die effizienteste Variante der Schaltung zu bestimmen. Durch die Analyse der maximalen Anzahl von ADD3-Gattern kann die Anzahl der benötigten ADD3-Blöcke abgeschätzt werden.

Hunderter	Zehner	Einer

Input
Aktionen

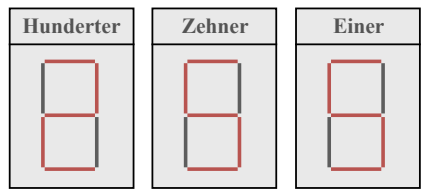
									1	1	1	1	1	1	1	1	Shift
								1	1	1	1	1	1	1	1	1	Shift
							1	1	1	1	1	1	1	1	1		Shift
						1	1	1	1	1	1	1	1	1			Add 3 (011)
					1	0	1	0	1	1	1	1	1	1			Shift
				1	0	1	0	1	0	1	1	1	1				Add 3 (011)
				1	1	0	0	0	0	1	1	1	1				Shift
				1	1	0	0	0	1	1	1	1					Shift
				1	1	0	0	0	1	1	1	1					Add 3 (011)
				1	0	0	1	0	0	1	1	1	1				Shift
				1	0	0	1	0	0	1	1	1	1				Add 3 (011)
				1	0	0	1	0	1	0	1	0	1				Shift
				1	0	0	1	0	1	0	1						

Die Binärzahl wird immer ein Stück nach links (abhängig von der grafischen Darstellung) rein in das BCD Register geschoben. Man beginnt beim MSB. Nach jeder Schiebeoperation, die genau um eine Stelle schiebt, wird kontrolliert, ob innerhalb eines BCD-Nibbel der Dezimalwert der Bits  $\geq 5$  ist. Wenn ja wird der Dezimalwert 3 dazuaddiert. Anschließend wird die Aktion solange fortgesetzt, bis auch das LSB verarbeitet wurde.

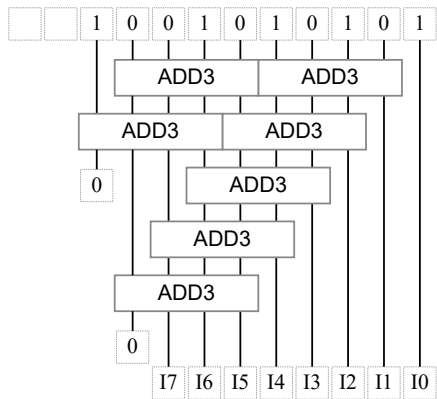
[illegible]

Insgesamt kommt man bei 8 Input-Bits auf maximal 7 mögliche Rechenoperationen. Da laut Algorithmus definiert ist, dass erst bei  $\geq 5$  addiert wird, fallen wenn nur zwei Bits im jeweiligen Nibble vom BCD-Register eingeschoben wurden, diese jeweils weg. Bei der Hunderterziffer wird es nie zu einem ADD3 kommen mit 8 Input-Bit, denn der Maximalwert dieser Ziffer beträgt Zwei (0010). Da dem Input ganz klar seine Adresse zugeordnet werden kann, lässt sich eine schematische Darstellung daraus kreieren.

# Schematische Darstellung



BCD-Output



Binär-Input

## Ergebnis

Nach Analyse der beiden Skizzen entspringt dieses Schema. Der Grundalgorithmus der mit den beiden Kernfunktionen ist hier klar wiederzuerkennen. Das vergleichen mit dem Inputwert und das addieren mit 3 (011) übernimmt der ADD3 Block. Die Schaltlogik hierzu muss noch ermittelt werden. Das Shiften ist durch die Verdrahtung der ADD3-Blöcke mit den In- und Outputs gegeben. Aus rein theoretischer Sicht sollte dieses Schema funktionieren.