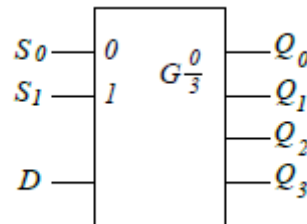


- **Das zugrundeliegende Schaltnetz: Selbsttestaufgabe 2.8**

Bauen Sie aus einem 1-zu-4 Demultiplexer einen 2-Bit Adressdecoder. Hierzu müssen Sie lediglich in folgender Abbildung die Ein und Ausgänge des Schaltzeichens neu beschriften.



Musterlösung:

Die Steuereingänge des Demultiplexers werden mit den Adresseingängen verbunden. Der Dateneingang wird fest mit dem Wert 1 verbunden. Die Ausgänge des Demultiplexers entsprechen den Ausgängen des Adressdecoders.

Quelle: Computersysteme I (2017), Kapitel 2.8 Multiplexer

- **Die Hades Simulation:**

Der als letzte Seite angehängte Screenshot zeigt die Hades Umsetzung des Schaltnetzes. Links sind die Eingangssignale A_1, A_0 platziert, ebenso wie die fest verdrahtete 1 für den Dateneingang, wie in der Aufgabe beschrieben. Rechts stehen die vier Ausgangssignale für Q_0, \dots, Q_3 . Initial sind zu Anschauungszwecken A_1, A_0 mit 0/1 vorbelegt, entsprechend wählt der Demultiplexer Q_1 . Durch Klick auf die Schalter belegt man A_1, A_0 mit den Werten 0 (grau) und 1 (rot) und kann beobachten, wie der Demux einen Adressdecodierer realisiert.

- **Die Simulation besteht aus folgenden Komponenten:**

- 2 Ipins (switch)
- 1 1:4 DEMUX
- 1 constant1
- 4 Opins (LED)

- **Besonderheit:**

In dieser Simulation wurde ein *Subdesign* verwendet. Das Symbol *1:4 DEMUX* versteckt die Implementierung des zugrundeliegenden 1 zu 4 Demultiplexers (siehe Simulation 15 dieser Reihe) und verfügt über die volle Funktionalität des enthaltenen Subdesigns. Wie bei allen verwendeten Subdesigns gilt auch hier: Klickt man mit der rechten Maustaste auf das Symbol und wählt im erscheinenden Popup-Menü den Eintrag *edit*, so öffnet sich das enthaltene Subdesign im Editor.

