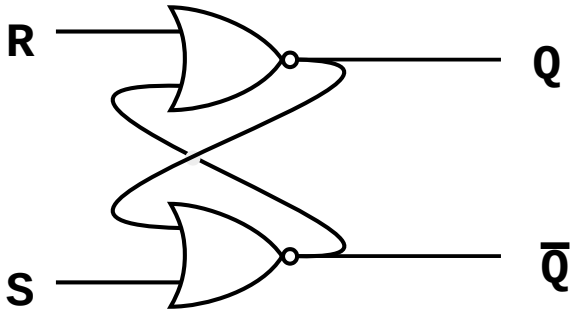
Speicher

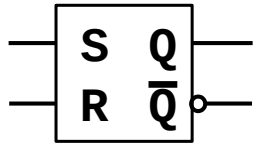
R-S Flip Flop

Wir basteln uns eine Speicherzelle (R-S Flip Flop)

Q : Gespeicherter Zustand
~Q: negierter gespeicherter Zustand
R : Reset löscht das Bit
S : Set setzt das Bit

Die Eingaben (R,S) bieten nur 4 Kombinationen. Überlegen Sie selbst, warum die beiden NOR-Gatter einen 1-Bit Speicher bilden und was die 4 Belegungen für (R,S) bedeuten.





© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Speicher

R-S Flip Flop

Wir basteln uns eine Speicherzelle

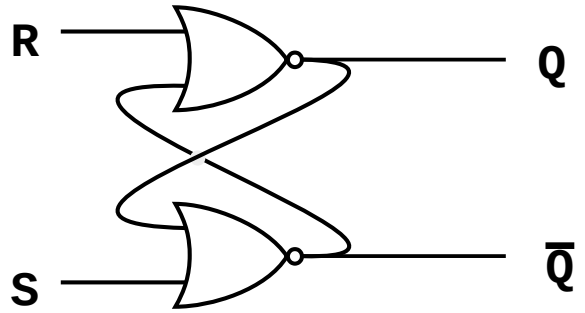
S	R	Q	~Q
0	0	?	?

Problem: Q und ~Q gehen selbst rekursiv in die Schaltung ein. Wir kennen die Werte von Q und ~Q anfangs nicht.

Lösung: Wir nehmen einfach jede Anfangskombinationen an.

		vorher		nachher	
S	R	Q	~Q	Q	~Q
0	0	0	0	1	0
0	**0**	**0**	**1**	**0**	**1**
0	**0**	**1**	**0**	**1**	**0**
0	0	1	1	1	0

Bedeutung der beiden mittleren Zeilen: Bei (S=0/R=0) beibt der Zustand (Q=0/~Q=1) oder (Q=1/~Q=0) erhalten. Liegt also an R/S false an, so behält der Speicher seinen Zustand. Analog kann man alle weiteren Fälle für S/R testen.



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

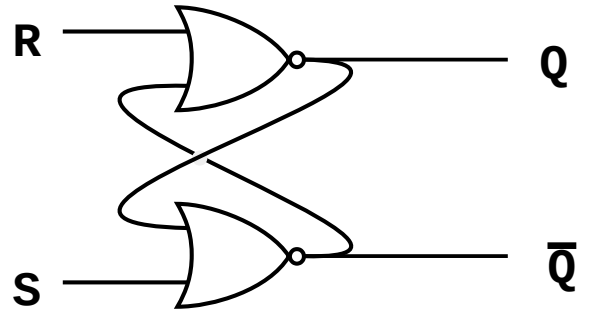
Speicher

R-S Flip Flop

Wir basteln uns eine Speicherzelle

		vorher		nachher	
S	R	Q	~Q	Q	~Q
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1

Bedeutung aller Zeilen: Die Belegung (S=0/R=1) führt in jedem Fall dazu, dass sich (Q=0/~Q=1) einstellt. Der Speicher wird also gelöscht und das ganz unabhängig davon, was Q und ~Q zuvor zufällig gewesen sein könnte.



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

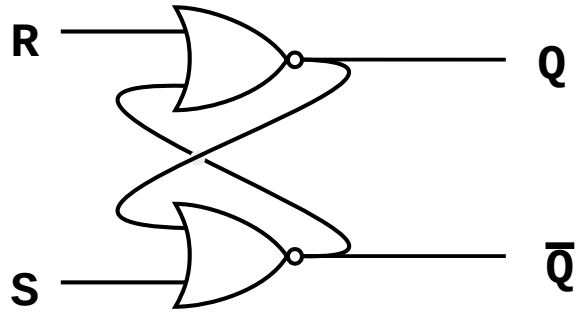
Speicher

R-S Flip Flop

Wir basteln uns eine Speicherzelle

		vorher		nachher	
S	R	Q	~Q	Q	~Q
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0

Bedeutung aller Zeilen: Die Belegung (S=1/R=0) führt in jedem Fall dafür, dass sich (Q=1/~Q=0) einstellt. Das Bit des Speichers wird also gesetzt.



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

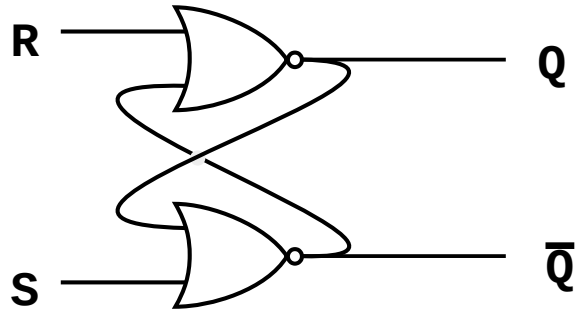
Speicher

R-S Flip Flop

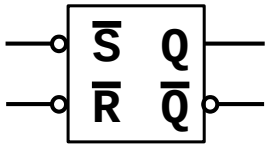
Wir basteln uns eine Speicherzelle

		vorher		nachher	
S	R	Q	~Q	Q	~Q
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Bedeutung aller Zeilen: Die Belegung (S=1/R=1) Löscht Q und ~Q (solange sich S und R nicht ändern). Diese Funktion hat in der Praxis keine Bedeutung, da man S und R praktisch nicht gleichzeitig auf Null schalten kann.



Ebenso mit NAND möglich, dann ist S=R=1 der Haltezustand



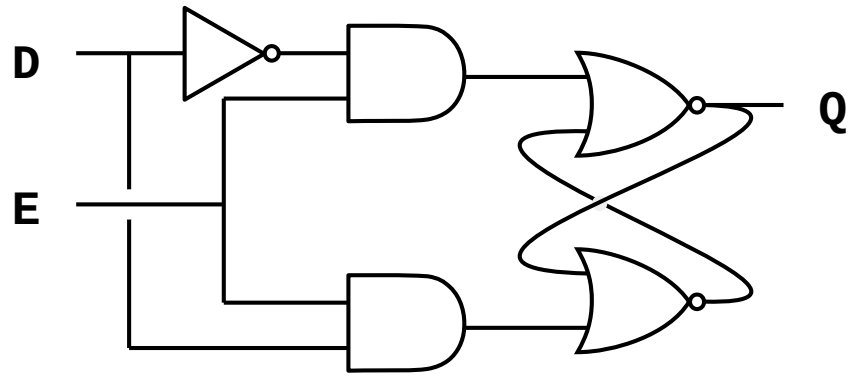
© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

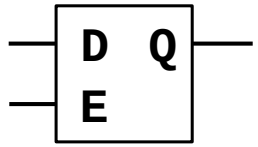
Speicherzelle mit Enable Port

Problem des RS-FF: Es hört ständig zu. Meist hat man nur einen Datenbus. Nicht alle Speicherzellen sollen zuhören. Lösung: Das D-Latch mit einer enable Leitung.

```
E D | Q
---------
0 0 | gespeicherter Zustand bleibt
0 1 | gespeicherter Zustand bleibt
1 0 | 0
1 1 | 1

D : Data
E : Enable (0=nur read/1=write)
```





© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

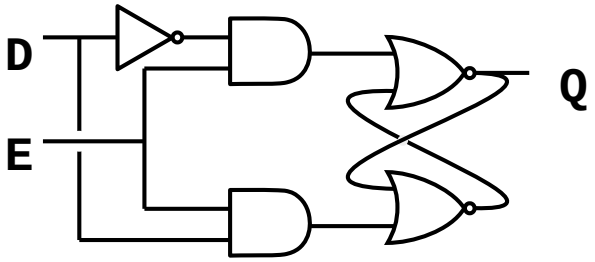
Praktikum

Speicher

D-Latch

1) Bauen Sie ein D Latch als Schaltung in der Simulationssoftware Logisim nach.

2) Angenommen Sie haben 2 D-Latches, eine Adressleitung und eine Datenleitung. Die Adressleitung soll dazu genutzt werden eines der beiden Latches auszuwählen. Mit der Datenleitung soll das Bit gesetzt werden. Erweitern Sie die Schaltung in Logisim so, dass dies geht.
   Mit der obigen Schaltung haben Sie einen sog. Demultiplexer gebaut.



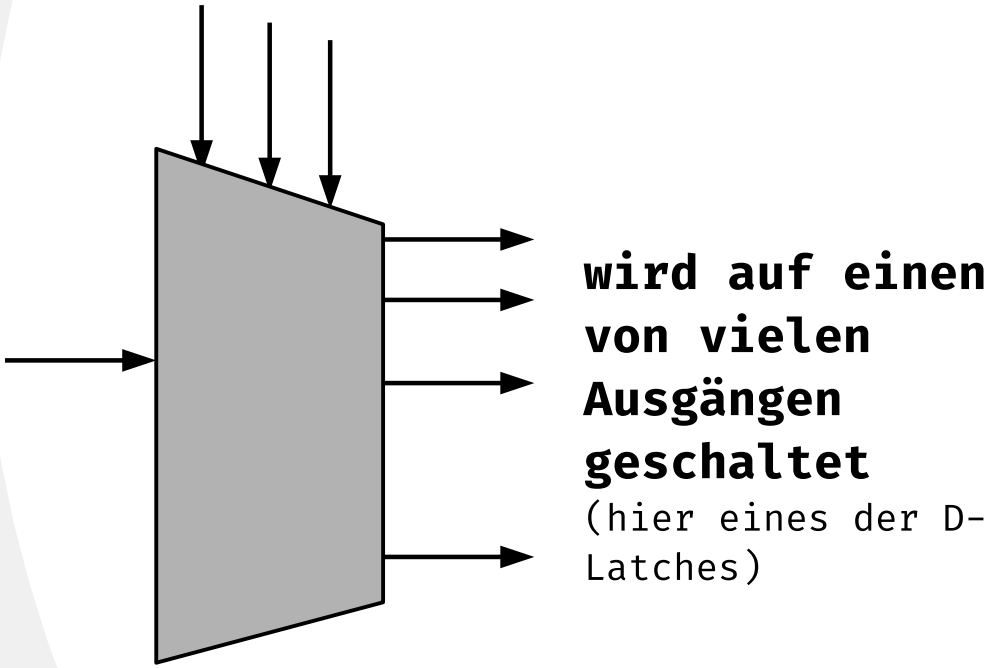
© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Demultiplexer

Eine Eingangsleitung wird auf eine von vielen Ausgangsleitungen „demultiplext“. Die Wahl der Ausgangsleitung geschieht mittels des Adressbusses.

Auf wie viele Ausgänge kann durchgeschaltet werden, wenn der Adressbus n Leitungen besitzt?

ein Eingang
(hier das Bit das gespeichert werden soll)



© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Adressierbarer Speicher

2) (…)
   Mit der obigen Schaltung haben Sie einen sog. Demultiplexer gebaut. Wie sieht die Schaltung in Logisim aus, wenn 4 D-Latches adressiert werden müssen? Nutzen Sie Cut&Paste für die Verdoppelung der D-Latches und erweitern Sie unseren Mini-Adressbus mit bisher nur einer Leitung sinnvoll. Wie erreichen Sie, dass jede Speicherzelle einzeln adressiert werden kann?

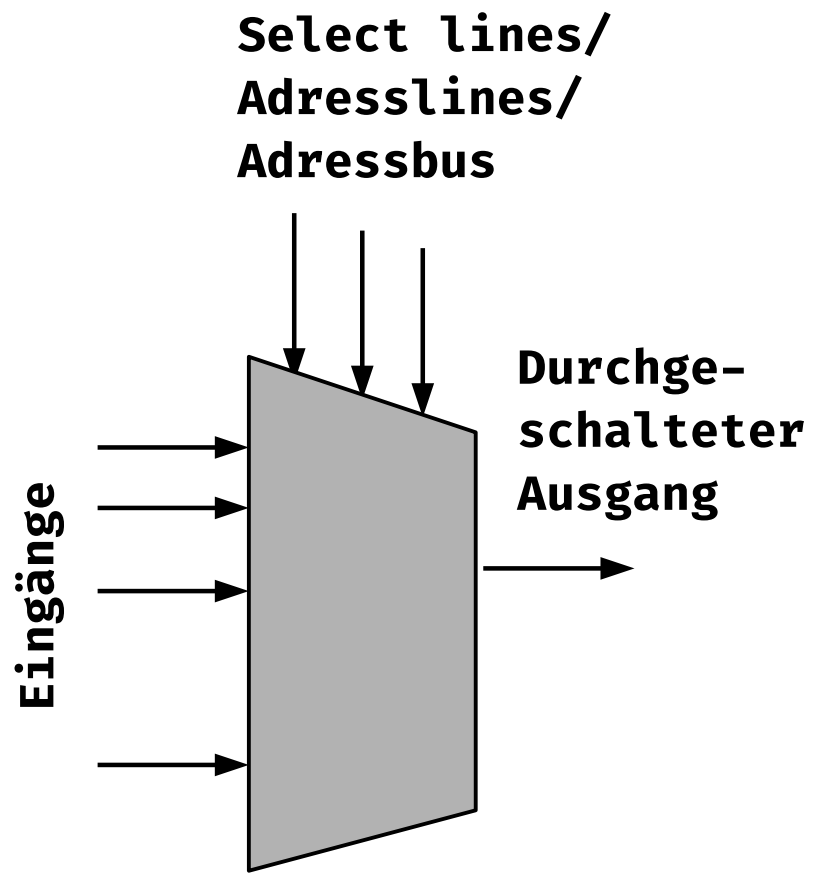
3) Wahrscheinlich zeigt Ihre Schaltung im Moment den Zustand jedes Latches mit einer LED an. In einem echten Computer wäre aber nur die adressierte Speicherzelle mit der CPU verbunden. Erweitern Sie Ihre Schaltung so, dass nur der Zustand des gerade adressierten Latches an einer einzigen zentralen LED angezeigt wird. Damit haben Sie einen Multiplexer gebaut.

© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Multiplexer

Viele Eingangsleitungen werden auf eine Ausgangsleitung „gemultiplext“. Die Wahl der Eingangsleitung geschieht mittels des Adressbusses.

Einen Multiplexer kann man klassisch mit einer Wahrheitstabelle bauen. Oder einfach direkt mit AND- und OR-Gattern.

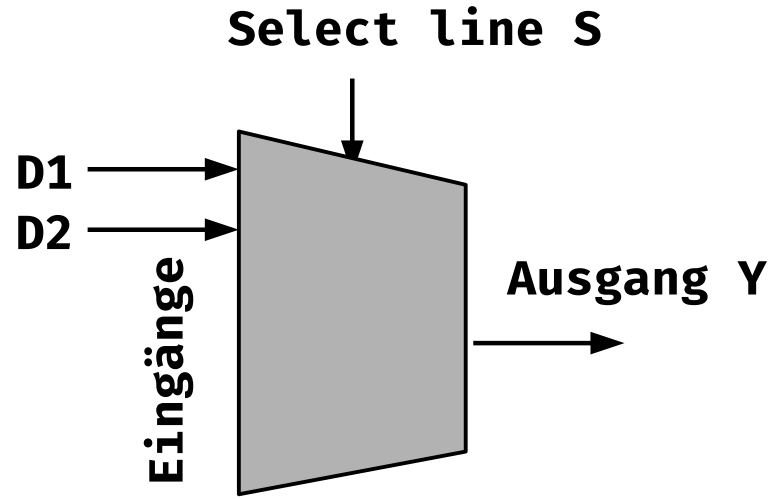


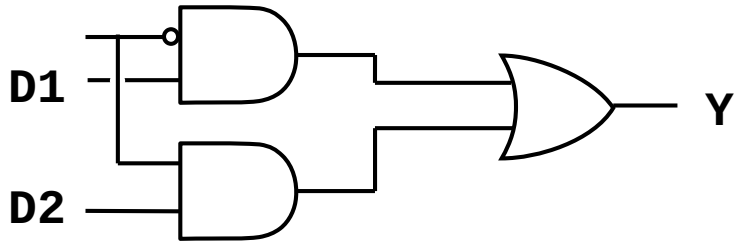
© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Multiplexer

S	D2	D1	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$Y = \sim S\&\sim D2\&D1 \mid \sim S\&D2\&D1 \mid S\&D2\&\sim D1 \mid S\&D2\&D1$$
$$= S\&D2\&(\sim D1 \mid D1) \mid \sim S\&D1\&(\sim D2 \mid D2)$$
$$= S\&D2 \mid \sim S\&D1$$





© Thomas Haenselmann, Prof. Dr. rer. nat. habil.

Praktikum

Speicher

Adressierbarer Speicher

4) Testen Sie Ihre Schaltung. Vielleicht fällt Ihnen beim Test ein Problem beim Lesen der zuvor gespeicherten Zustände auf. Welches?

5) Versuchen Sie das Problem durch Erweiterung der Schaltung zu lösen.

6) Implementieren Sie die Schaltung in Verilog. Lagern Sie dazu das D-Latch in ein eigenes Modul aus und testen dieses zuerst.
Hinweis: Schleifen (Loops) von wechselseitig miteinander verbundenen Elementen müssen erst erlaubt werden. Aktivieren Sie die folgenden beiden Kommentarzeilen in Ihrer constraints-Datei durch Löschen des #.
set_property ALLOW_COMBINATORIAL_LOOPS TRUE
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets btnL_IBUF]

   Versuchen Sie in Ihrer Verilog Implementierung keinen doppelten Code zu schreiben. Ggfs. hilft Ihnen ein wire bei der Wiederverwendung.