

I.) Inhaltsverzeichnis

I.) Inhaltsverzeichnis.....	1
II.) Definitionen	2
a.) Signale	2
b.) Taktquelle.....	2
c.) Aufbau von Schaltungen	2
d.) Simulation von Schaltungen.....	2
III.) Übungsinhalt	3
IV.) Registerschaltungen.....	3
1.) Übungsaufbau 1 (JK-Schieberegister)	3
a.) Verwendete Komponenten.....	3
b.) Schaltungsaufbau	3
c.) Übertragung von Informationen.....	4
d.) Beantwortung der Fragen	4
2.) Übungsaufbau 2 (Schieberegister mit paralleler Dateneingabe).....	5
a.) Verwendete Komponenten.....	5
b.) Schaltungsaufbau	5
a.) Einlesen von Binärdaten	6
b.) Ermittlung der Funktionsgleichungen	6
3.) Übungsaufbau 3 (4-Bit Schieberegister mit paralleler Dateneingabe)	6
a.) Verwendete Komponenten.....	6
b.) Schaltungsaufbau	7
c.) Einlesen paralleler Informationen mit serieller Übertragung.....	8
4.) Übungsaufbau 4 (Serielle 4-Bit Datenübertragung.....	8
a.) Verwendete Komponenten.....	9
b.) Schaltungsaufbau	9
c.) Übertragung von parallelen Informationen.....	10
a.) Beantwortung der Fragen	10
V.) Multiplexerbetrieb	11
1.) Übungsaufbau (Multiplexing/Demultiplexing)	11
a.) Verwendete Komponenten.....	11
b.) Schaltungsaufbau	11
c.) Übertragung von Informationen.....	12
d.) Beantwortung der Fragen	12
VI.) Anhang	13

II.) Definitionen

a.) Signale

Logisch 0	LOW
Logisch 1	HIGH
Logisch X	HIGH-Z/HIGH-Impedanz

b.) Taktquelle

In dem Protokoll auftretende Bezeichnungen wie HPS Board interner Takt oder generierter Takt sind auf die Taktversorgung des HPS Boards zurückzuführen (siehe Bild).

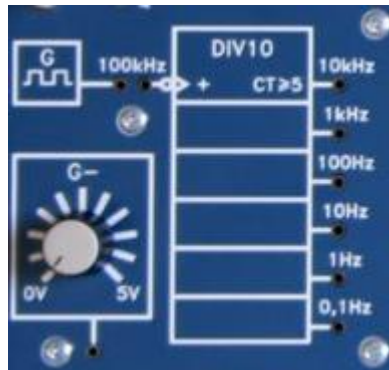


Abb. 1: HPS_DIGIBOARD2 interne Taktversorgung

c.) Aufbau von Schaltungen

Um die Schaltungen zu verstehen wird versucht diese größtenteils mit Logisim oder anderer Software zu erstellen. Da in Logisim und anderen Softwareplattformen nicht immer die gegebenen Bauteile zu finden sind kann es vorkommen dass diese Bauteile durch mehrere unterschiedliche Bauteile nachgebaut werden. Die nachgebauten Schaltungen die zu einem Bauteil zusammengefasst werden sind in den meisten Fällen durch strichlierte Linien zusammengefasst und mit dem Bauteilindex beschriftet.



d.) Simulation von Schaltungen

Die Simulationsbeschriftungen der Schaltungen sind möglicherweise nicht immer mit der originalen Schaltung identisch. Sollte es zu Unklarheiten kommen, sind die vervollständigten Unterlagen im Anhang zur Kontrolle herbeizuziehen. Diese Schaltungen wurden größtenteils in VHDL beschrieben und sind auf <https://github.com/sunriax/labor/digitaltechnik1/>



DIGITALTECHNIK 1

III.) Übungsinhalt

- *Registerschaltungen*
- *Multiplexerbetrieb*

IV.) Registerschaltungen

In der Informationstechnik ist es häufig notwendig parallele Signale über eine serielle Verbindung zu übertragen. Dies birgt den Vorteil das bei der Übertragung von Daten Leitungen durch Serialisierung eingespart werden können. Der Nachteil dabei ist das die Übertragungsgeschwindigkeit mit der Länge des zu übertragenden Wortes im Gegensatz zur parallelen Übertragung sinkt.

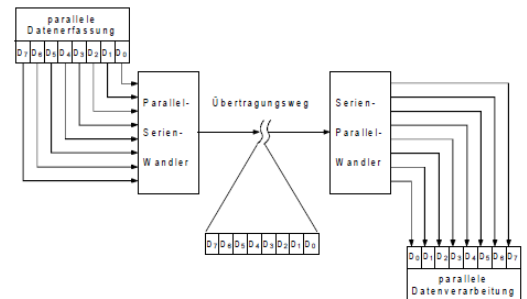


Abb. 2: Übertragung von parallelen Daten seriell

1.) Übungsaufbau 1 (JK-Schieberegister)

Ein Schaltwerk, welches in der Lage ist, Informationen zu speichern und zu verschieben wird als Schieberegister bezeichnet. Bei einem Schieberegister kommen JK-Flipflops zum Einsatz.

a.) Verwendete Komponenten



Abb. 3: HPS_DIGIBOARD2 verwendete Komponenten

Es wurden oben gezeigte (**Abb. 3 Fehler! Verweisquelle konnte nicht gefunden werden.**) JK-Flipflops verwendet und unten beschriebene Schaltung aufgebaut. Dabei wurde jeweils ein händischer und anschließend ein automatisch generierter Takt (HPS Board interner Takt) erzeugt. Zur Visualisierung wurden die LEDs des HPS Boards an die Ausgänge angeschlossen.

b.) Schaltungsaufbau

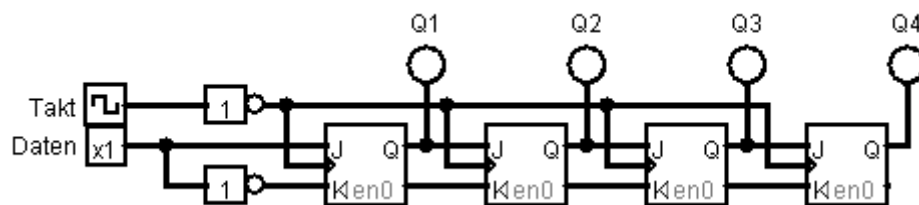


Abb. 4: Registerschaltung mit JK-Flipflops

Hinweis: Der Takt wurde im Schaltbild negiert da auf dem HPS Board die Takteingänge der JK-Flipflops ebenfalls negiert sind! **Daher ist die Negation der Taktleitung nicht auszuführen!**

c.) Übertragung von Informationen

Im ersten Versuch wird die Datenleitung auf logisch 1 (High) gelegt und anschließend werden 5 einzelne Takte (händisch) durchgeführt. Dabei ergibt sich folgende Wertetabelle/Zeitdiagramm:

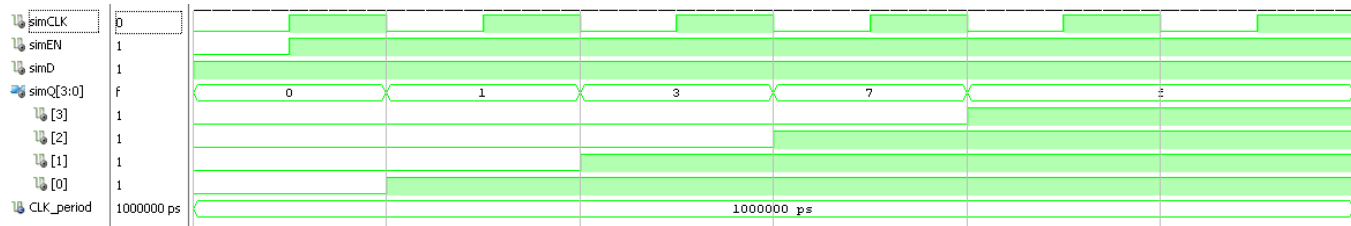


Abb. 5: Registerbetrieb mit Logisch 1 (high) am Dateneingang

C	D	Q ₀	Q ₁	Q ₂	Q ₃
1	1	1	0	0	0
2	1	1	1	0	0
3	1	1	1	1	0
4	1	1	1	1	1
5	1	1	1	1	1

Die Wahrheitstabelle zeigt das durchschieben des High Zustandes durch das Schieberegister. Dabei ist schön ersichtlich die die Information mit jedem Takt um eine Stelle verschoben wird.

Zur weiteren Veranschaulichung der Funktionsweise wurde folgende Information übertragen: **0b1100111** (Eingangsinformation) und eine Wertetabelle sowie ein Zeitdiagramm angelegt.

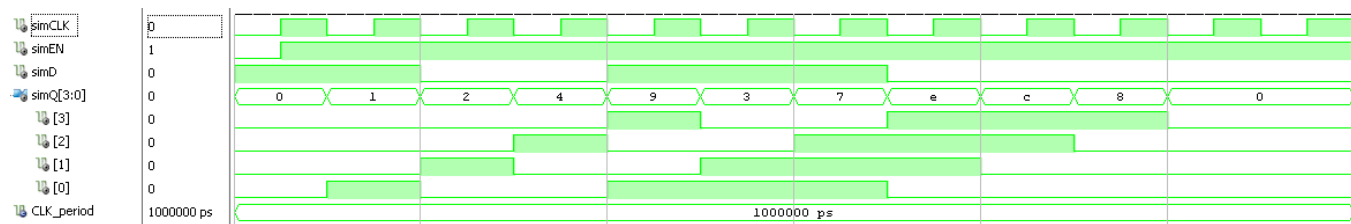


Abb. 6: Registerbetrieb mit Datenwort (0b1100111) am Dateneingang

C	D	Q ₀	Q ₁	Q ₂	Q ₃
1	1	1	0	0	0
2	1	1	1	0	0
3	0	0	1	1	0
4	0	0	0	1	1
5	1	1	0	0	1
6	1	1	1	0	0
7	1	1	1	1	0

Die Wahrheitstabelle zeigt das durchschieben des oben beschriebenen Datenworts. Beginnend mit Takt ein wird das **MSB 0b1** auf **Q₀** geschrieben. Weiterfolgend wird **0b11** im Takt 2 auf **Q₁** und **Q₀** geschrieben dabei bewegt sich das **MSB** um eine Stelle nach rechts. Dies wird mit dem gesamten Datenwort vollzogen bis in Takt sieben das **LSB 0b1100111** am Ausgang **Q₀** anliegt.

d.) Beantwortung der Fragen

- Frage 1: Interpretiere den Takt 5 im oben angeführten Zeitdiagramm (Abb. 5)**

Das Schieberegister wurde mit Setzen des Dateneingangs D=HIGH mit 5 Taktzyklen (fallende Flanke) durchgetaktet. Nach dem 3 Takt bleiben alle Ausgänge (Q₀-Q₃) auf **HIGH**, da das Signal am Dateneingang durch das gesamte Schieberegister getaktet wurde.
- Frage 2: Beschreiben Sie die Arbeitsweise des Schieberegisters**

Mit jedem Takt (fallende Flanke) wird das Signal das sich am Dateneingang (D) befindet auf den Ausgang Q₀ übernommen. Das Signal welches ursprünglich in Q₀ gespeichert war wird auf Q₁ gelegt. Das Signal von Q₁ wird auf Q₂ gelegt und das Signal von Q₂ wird auf Q₃ gelegt und das ursprüngliche Signal verschwindet aus dem Register. Das Schieberegister befindet sich im sogenannten Linksschiebemodus (MSB zuerst).

2.) Übungsaufbau 2 (Schieberegister mit paralleler Dateneingabe)

Um Daten direkt und ohne Serielle Einlesen bedarf es jeweils ein eigenes Flipflop für jeden Eingangskanal welches gesetzt und rückgesetzt werden kann. Durch eine Funktion (Einlesen) kann der Inhalt an den parallelen Eingängen direkt in das Schieberegister übernommen werden und anschließend durch den Betrieb wie im Übungsaufbau 1 weiter getaktet werden. Dies findet z.B. zum Einlesen von Tastzuständen Anwendung.

a.) Verwendete Komponenten

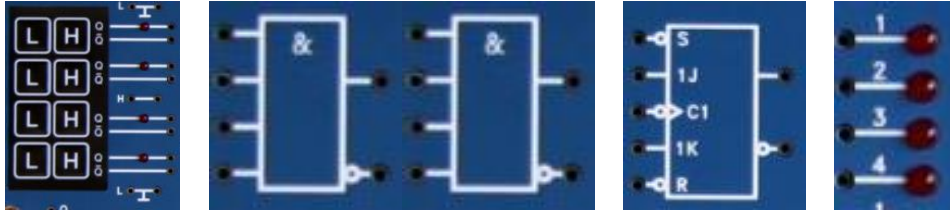


Abb. 7:HPS_DIGIBOARD2 verwendete Komponenten

Es wurden oben gezeigte (**Fehler! Verweisquelle konnte nicht gefunden werden.**) NAND-Verknüpfungen sowie ein beliebiges JK-Flipflop verwendet. erzeugt. Zur Visualisierung wurden die LEDs des HPS Boards an die Ausgänge angeschlossen. Die benötigte Negation für das Einlesen der Binärdaten wurde aus dem Taster L/H erzeugt (negiertes Taster Signal).

b.) Schaltungsaufbau

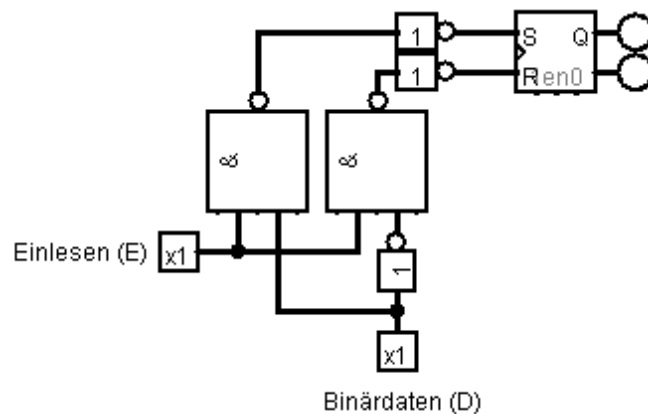


Abb. 8:parallele Dateneingabe in einem Schieberegister

Hinweis: Der Takt wurde im Schaltbild negiert da auf dem HPS Board die Takteingänge der JK-Flipflops ebenfalls negiert sind! **Daher ist die Negation der Taktleitung nicht auszuführen!** Ebenfalls wurden die Eingänge des RS-Flipflops negiert. Dies wäre nicht notwendig, wenn die NAND-Gatter durch UND-Gatter ersetzt werden würden. Um jedoch die Schaltung so darzustellen wie sie auf dem Board aufzubauen ist, wurde auch dieser Teil angepasst! **Die Negation des RS-Flipflops ist daher nicht auszuführen da bereits vorhanden!**

a.) Einlesen von Binärdaten

Im ersten Versuch wird die Datenleitung auf logisch 1 (High) gelegt und anschließend werden 5 einzelne Takte (händisch) durchgeführt. Dabei ergibt sich folgende Wertetabelle/Zeitdiagramm:

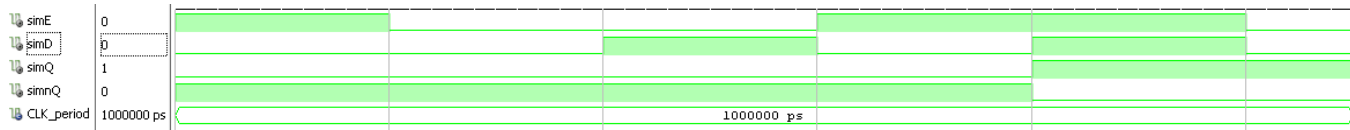


Abb. 9: Eingabe paralleler Daten in das Register

E	D	Q	nQ	Funktion
0	0	Q+1	nQ+1	Eingabe
0	1	Q+1	nQ+1	gesperrt
1	0	0	1	Rücksetzen
1	1	1	0	Setzen

Aus der Wahrheitstabelle ist ersichtlich das ein Einlesen der Daten nur bei gesetztem Eingang (E) möglich ist. Das gesetzte Bit Q/nQ wird am Ausgang solange gehalten bis ein neues Bit eingelesen wird. Wir das Register (siehe nächstes Beispiel) weitergetaktet so kann das Bit später durch parallele Eingabe seriell ausgetaktet werden.

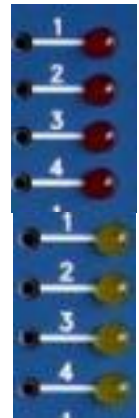
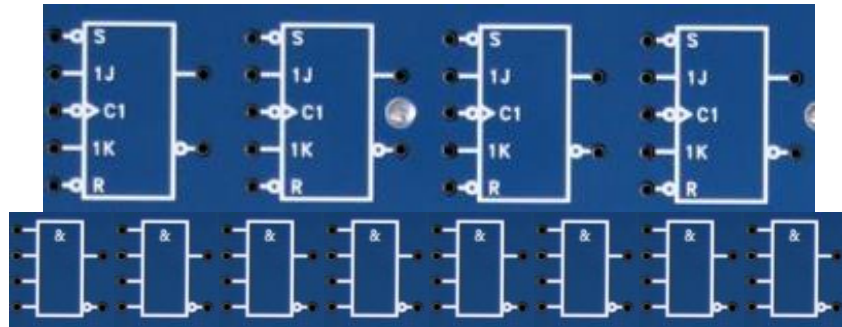
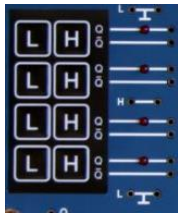
b.) Ermittlung der Funktionsgleichungen

$$\begin{aligned}\bar{S} &= E \wedge D & \text{Die angeführten Funktionsgleichungen dienen dem Verständnis der oben angeführten} \\ S &= \bar{E} \wedge \bar{D} & \text{Schaltung (Abb. 8).} \\ \bar{R} &= E \wedge \bar{D} \\ R &= E \wedge D\end{aligned}$$

3.) Übungsaufbau 3 (4-Bit Schieberegister mit paralleler Dateneingabe)

Einige Schieberegister weisen eine besondere Funktion auf nämlich die der parallelen Dateneingabe. Dies wird in diesem Übungsaufbau demonstriert. Die Daten können parallel eingelesen werden und nach dem Einlesevorgang seriell weitergetaktet werden. Dies ermöglicht das parallel/seriell umsetzen um die Daten zu Serialisieren.

a.) Verwendete Komponenten



In dem Übungsaufbau wurden oben gezeigte NAND-Verknüpfungen und JP-Flipflops verwendet. Die negierten Takteingänge werden hier nicht invertiert das bedeutet das sie aus der Taktquelle bereits negiert herauszuführen sind. Das Einlesen der Daten und weiterschieben wurde zuerst mit einem händischen Takt erzeugt und Anschließend aus der Taktquelle gezogen.

b.) Schaltungsaufbau

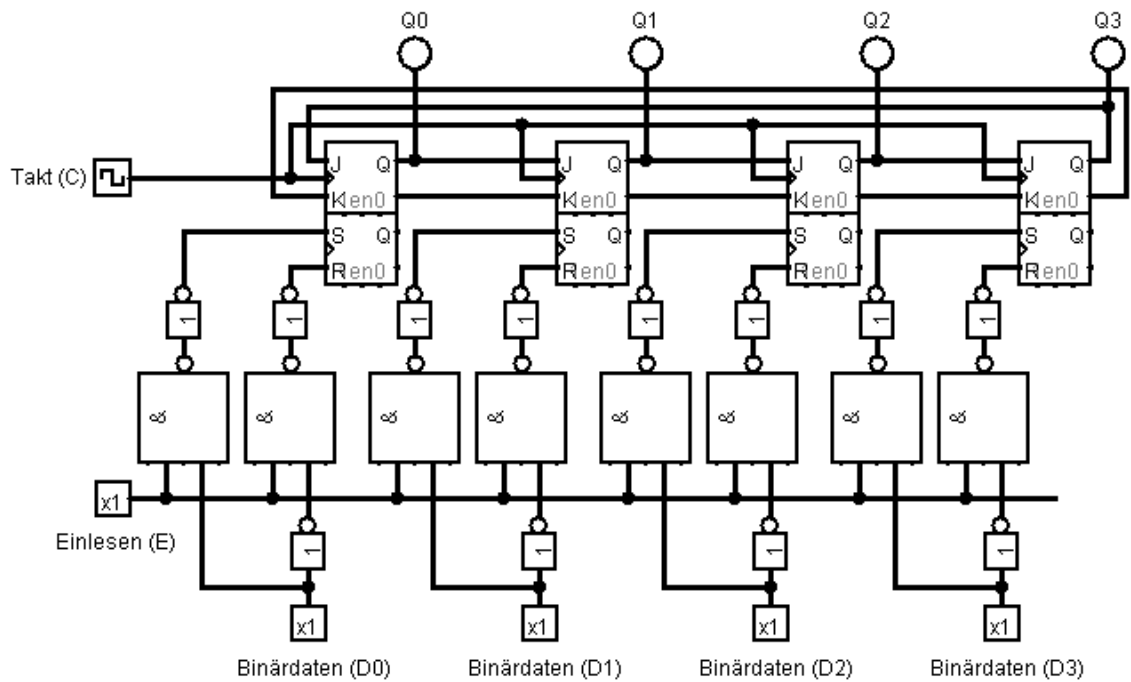


Abb. 10:parallele Dateneingabe in ein Schieberegister

Hinweis: Der Takt wurde im Schaltbild nicht negiert da er in dieser Übungsaufgabe nicht invertiert auszuführen ist. Des Weiteren weisen die S/R Eingänge im Simulationstool keine Negierung auf. Deshalb wurde diese mit einem Inverter versehen. Die Ausgänge Q_3 und nQ_3 wurden auf die Eingänge des ersten JK-Flipflops rückgeführt ($J=Q_3$ und $K=nQ_3$) rückgeführt. Das ermöglicht ein durchschieben der parallel eingelesenen Daten durch das Schieberegister.

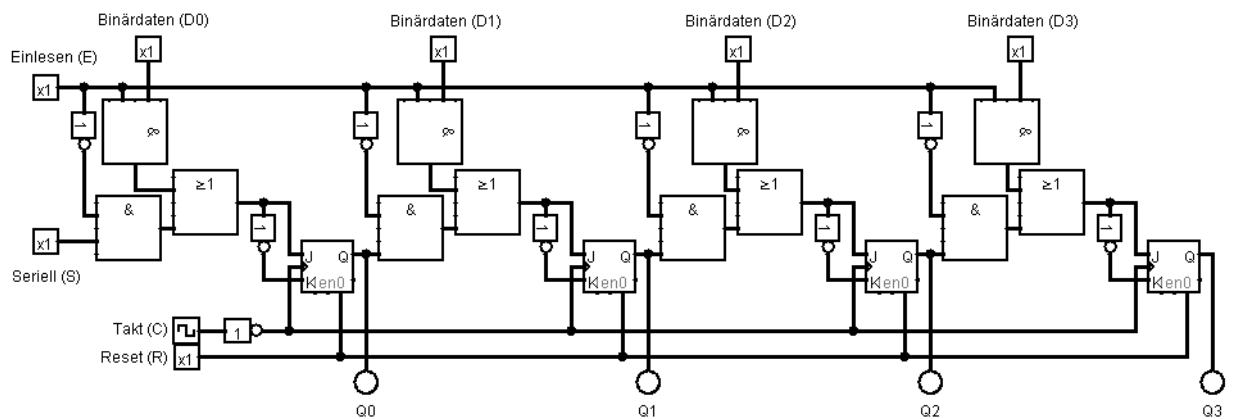
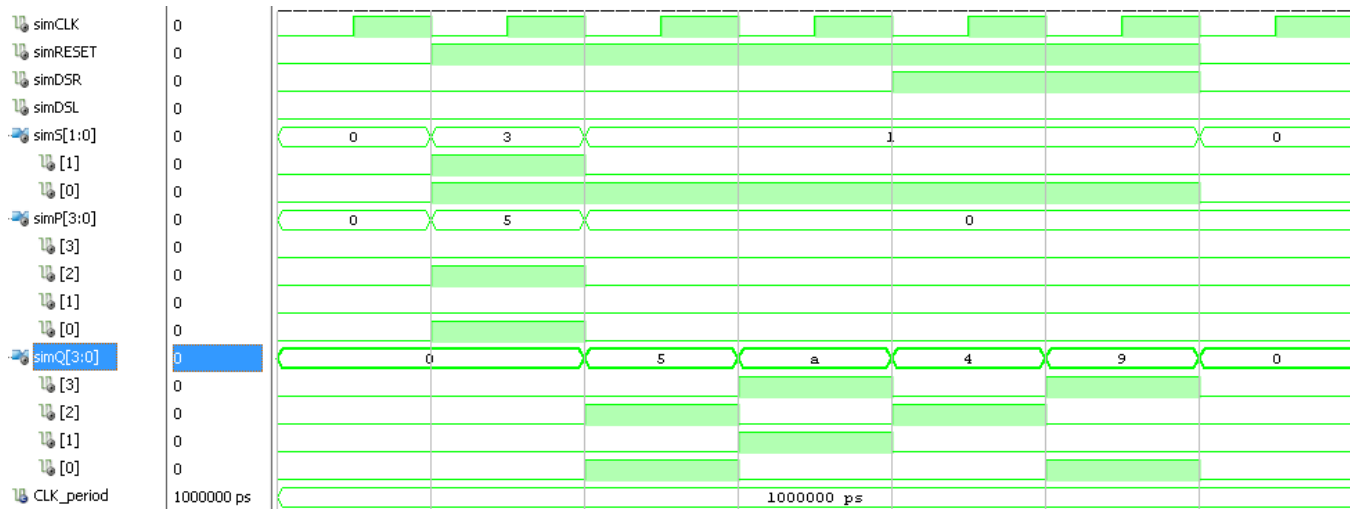


Abb. 11:mögliche interne Realisierung der parallelen Dateneingabe (ohne RS/SR-FlipFlop)

c.) Einlesen paralleler Informationen mit serieller Übertragung

Die Daten werden parallel eingelesen und anschließend über das Schieberegister durchgetaktet. Im 5. Takt wird noch zusätzlich ein eingetaktetes **HIGH** übernommen. Da für diese Simulation das Schieberegister SN74LS194 herangezogen wurde sind Unterschiede in den Signalbezeichnungen zu erkennen.



C	RESET	D _{SR}	D _{SL}	SimS ₁	simS ₀	simP ₀	simP ₁	simP ₂	simP ₃	Q ₀	Q ₁	Q ₂	Q ₃	Status
1	0	X	X	X	X	X	X	X	X	0	0	0	0	Reset
2	1	X	X	1	1	0	0	1	1	0	0	0	0	Parallel
3	1	0	X	0	1	X	X	X	X	1	0	1	0	Seriell
4	1	0	X	0	1	X	X	X	X	0	1	0	1	
5	1	1	X	0	1	X	X	X	X	0	0	1	0	
6	1	1	X	0	1	X	X	X	X	1	0	0	1	
7	0	X	X	X	X	X	X	X	X	0	0	0	0	Reset

Hinweis: Signalanpassung C=Takt, D_{SR}=Serielle Dateneingabe (im Übungsaufbau nicht vorhanden da Daten direkt vom letzten JK-Flipflop wieder ins Erste JK-Flipflop übernommen werden). S₁/S₀ sind für **parallele/serielle Einlesefunktion**. Paralleles Einlesen: **HIGH/HIGH** (entspricht dem einlesen Eingang E), seriell einlesen: **LOW/HIGH** (entspricht der Funktion Daten seriell einzulesen und weiterzutakten). P₀₋₃ sind parallele Dateneingänge, Q₀₋₃ sind die Ausgänge des JK-Schieberegisters.

4.) Übungsaufbau 4 (Serielle 4-Bit Datenübertragung)

In dieser Übung werden die Erkenntnisse aus den vorherigen Übungen zusammengetragen und ausgenutzt. Es erfolgt eine Serielle 4-Bit Übertragung mit paralleler Dateneingabe, Serialisierung und paralleler Datenausgabe. Der Vorteil dieser Methode ist offensichtlich da durch die Serialisierung die Möglichkeit besteht Datenleitungen einzusparen. Zudem kann eine serielle Datenübertragung deutlich schneller durchgeführt werden als eine parallele. Da bei paralleler Datenübertragung mit hoher Übertragungsgeschwindigkeit Störungen durch Nebenschleifen auftreten können und die gegenseitige ungleiche Kapazität der Leiterbahnen Probleme mit Signallaufzeiten verursachen. Dies kann durch Serialisierung vermieden werden und so eine höhere Übertragungsgeschwindigkeit erzielt werden. Bei gleicher Geschwindigkeit ohne Beachtung von Störungen oder Geschwindigkeitsgrenzen, ist die parallele Übertragung deutlich schneller.

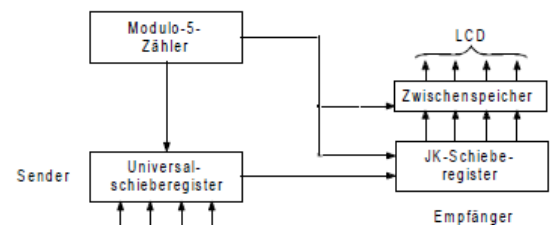


Abb. 12: Serielle 4-Bit Datenübertragung

a.) Verwendete Komponenten

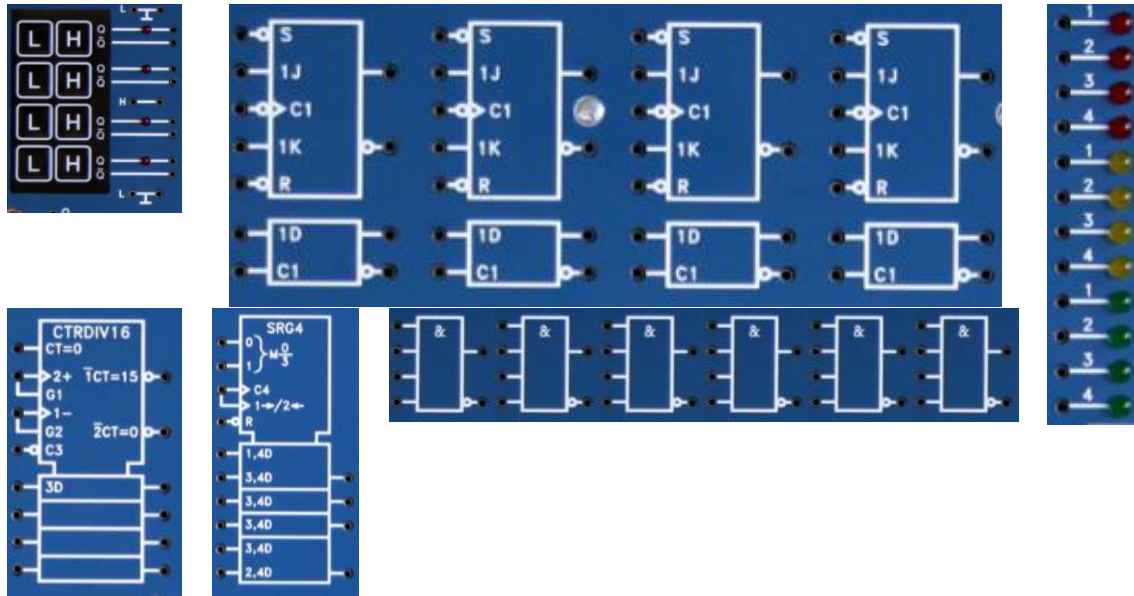
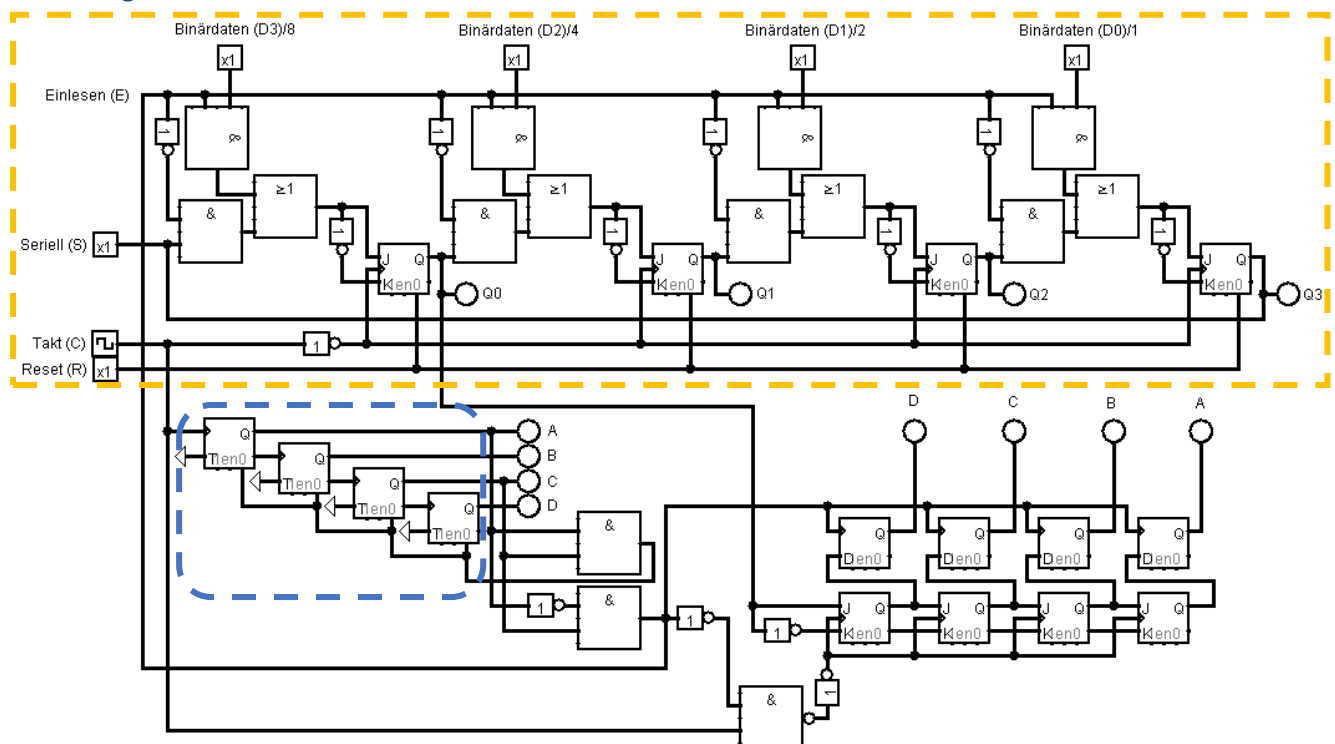


Abb. 13: HPS_DIGIBOARD2 verwendete Komponenten

Es wurden oben abgebildete Symbole zum Aufbau der Schaltung verwendet. Im unten angeführten Schaltplan wurden die Elemente die aus einem einzigen IC bestehen zusammengefasst. Möglicherweise ist die Beschriftung im Schaltungs Aufbau nicht identisch mit der Beschriftung der Komponenten.

b.) Schaltungs Aufbau

Abb. 14: Schaltung für eine 4-Bit serielle Datenübertragung,
Orange=SRG4; Blau=CTRDIV16

Hinweis: Die Schaltung und deren Elemente (SRG4/CTRDIV16) wurden anhand von diversen Schaltungen Simuliert und können in deren Funktionsweise von den originalen Bausteinen abweichen bzw. auch unterschiedlich im Aufbau sein. Des Weiteren kann beim Baustein SRG4 in der Simulation die Schieberichtung nicht nach Belieben geändert werden. Der Zähler CTRDIV16 kann ebenfalls so verschalten das eine Zählung bis 5 erfolgt. Die Schaltung wurde so Simuliert und entspricht dem erwarteten Ergebnis.

c.) Übertragung von parallelen Informationen

Anhand der Simulation lässt sich erkennen, dass das parallele Einlesen 1 Takt und die Übertragung 4 Takte benötigt bis die Daten ins Ausgangsregister übernommen werden. In der Simulation wurden zwei unterschiedliche Datenwörter übertragen:

Datenwort 1: 0b1101

Datenwort 2: 0b1001



Abb. 15: Parallel-Seriell-Parallel Datenübertragung

CLK	RSET	H	Z	E	D ₀	D ₁	D ₂	D ₃	S ₀	S ₁	S ₂	S ₃	M ₀	M ₁	A	B	C	D
1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	1	0	0	0	1	0	1	1	X	X	X	X	1	1	0	0	0	0
4	1	0	0	1	X	X	X	X	1	0	1	1	0	1	0	0	0	0
5	1	0	1	0	X	X	X	X	0	1	1	0	0	1	0	0	0	0
6	1	0	1	1	X	X	X	X	1	1	0	0	0	1	0	0	0	0
7	1	1	0	0	X	X	X	X	1	0	0	0	0	1	1	1	1	1

Hinweis: Die Beschriftung der Simulation ist wie bei vorherigen Simulationen nicht zwingend Identisch!

a.) Beantwortung der Fragen

- Frage 1: Anlegen von Taktfrequenzen von 1Hz und 100kHz, Aufklärung der Unterschiede?
Die Übertragung dauert bei 1Hz wesentlich länger als bei 100kHz. Ab ca. 10kHz erfolgt die Übertragung der eingegebenen Daten für das menschliche Auge zeitgleich.
- Frage 2: Wie viele 4-Bit Informationen können bei 100kHz in der Sekunde übertragen werden?

Da die Übertragung 5 Takte andauert kann daraus die Berechnung

$$Inf = \frac{100kHz}{5 \text{ Takte}} = 20000 \text{ Bit}$$

abgeleitet werden.

V.) Multiplexerbetrieb

Multiplexer wandeln parallel anstehende Daten in serielle Daten um. Dabei ist das Prinzip anders wie bei Registern. Die Daten werden über eine Leitung übertragen. Sender und Empfänger müssen dabei miteinander Synchronisiert werden. Dies wird meistens durch einen Takt oder Adressleitungen erreicht. Über einen Zähler werden die Adressen dabei angewählt und die Datenleitung selektiert und übertragen. Der Empfänger ist dabei auf die selbe Adresse wie der Empfänger eingestellt. Der wesentliche Unterschied zwischen Multiplexing und Registern ist das bei Multiplexing die parallelen Daten des Senders nicht zeitgleich (parallel) beim Empfänger ankommen. Dies liegt daran das immer nur ein Teilnehmer (Eingang) auf den Ausgang übertragen werden kann.

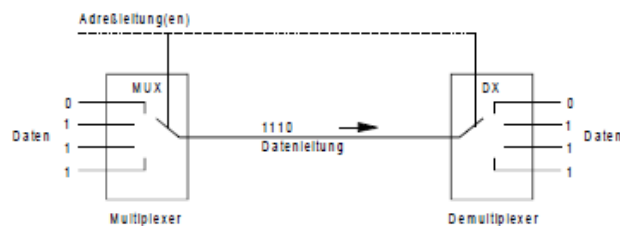


Abb. 16: Übertragung mittels Multiplex-/Demultiplexing

1.) Übungsaufbau (Multiplexing/Demultiplexing)

Übertragung der Daten durch Adressauswahl über einen Asynchronen Zähler der mittels T-Flipflops (JK-Flipflops auf J und K auf **HIGH** gesetzt) aufgebaut ist, betrieben wird.

a.) Verwendete Komponenten

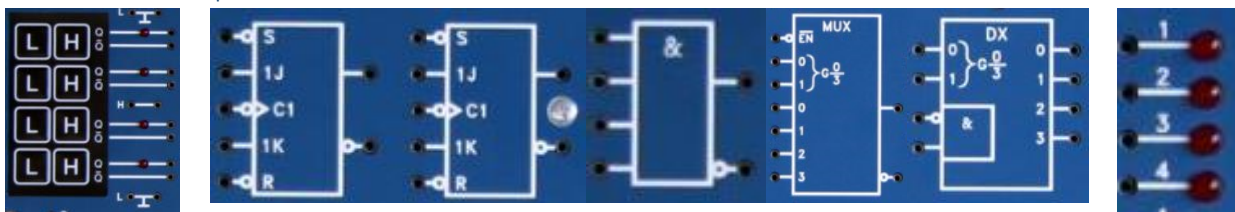


Abb. 17: HPS_DIGIBOARD2 verwendete Komponenten

Es wurden im oberen Bild gezeigte Bausteine verwendet. Für den Multiplexer sowie Demultiplexer wurden fertige Bausteine die auf dem Board zur Verfügung stehen verwendet. Des Weiteren wurden für das Schaltbild im unteren Abschnitt T-Flipflops verwendet. Diese lassen sich aus JK-Flipflops herstellen. Die Eingabe der Daten erfolgt über das **HIGH/LOW** Tastpad.

b.) Schaltungsaufbau

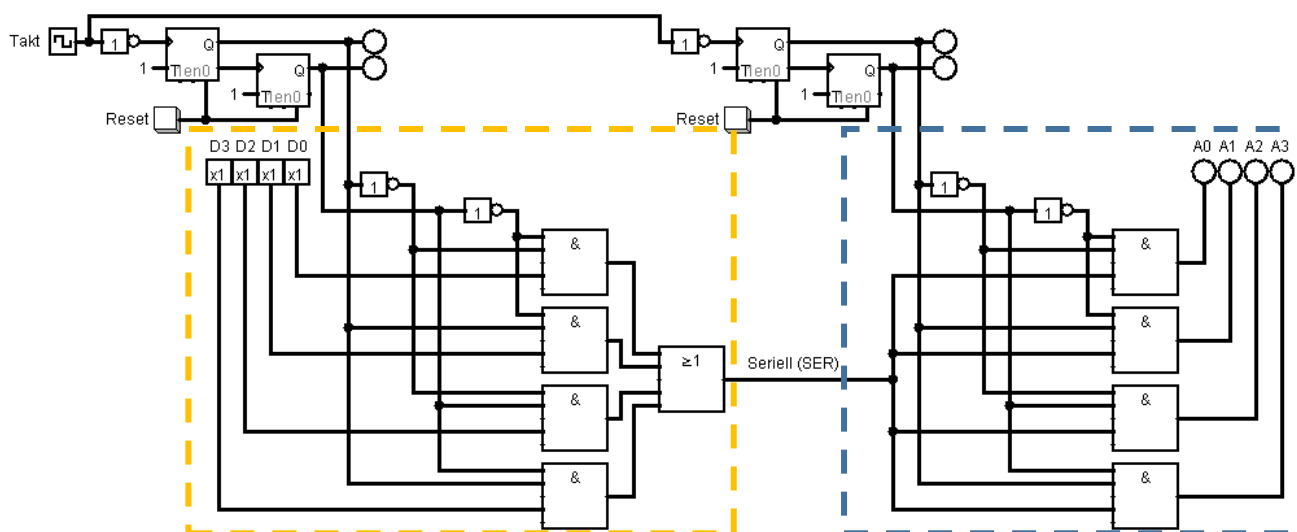


Abb. 18: Multiplexer/Demultiplexer mit Adresszähler

Orange=MULTIPLEXER; Blau=DEMUTIPLEXER

c.) Übertragung von Informationen

In der ersten Simulation wird das Multiplex Verfahren dargestellt und analysiert. Aus dem Zeitdiagramm ist ersichtlich das die Daten nacheinander übertragen werden.

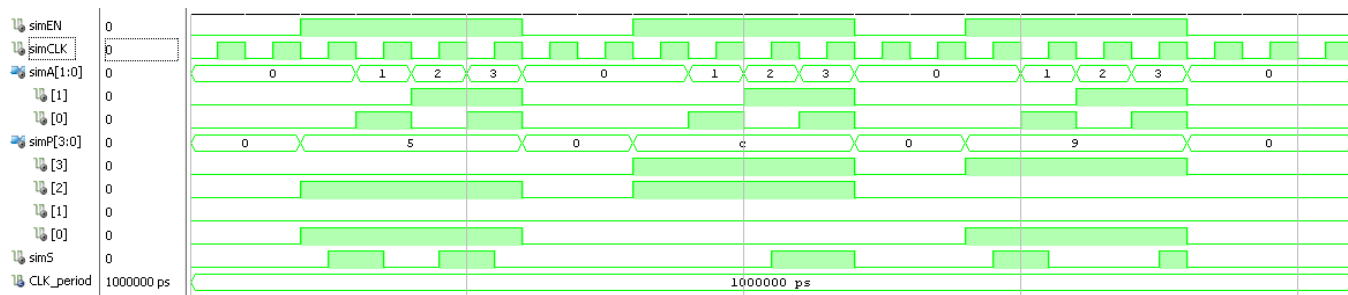


Abb. 19: Multiplexing Betrieb mit 3 unterschiedlichen Datensätzen

In der zweiten Simulation wird das Demultiplexing Verfahren dargestellt und analysiert und die in der vorherigen Simulation dargestellten Daten werden ausgewertet.

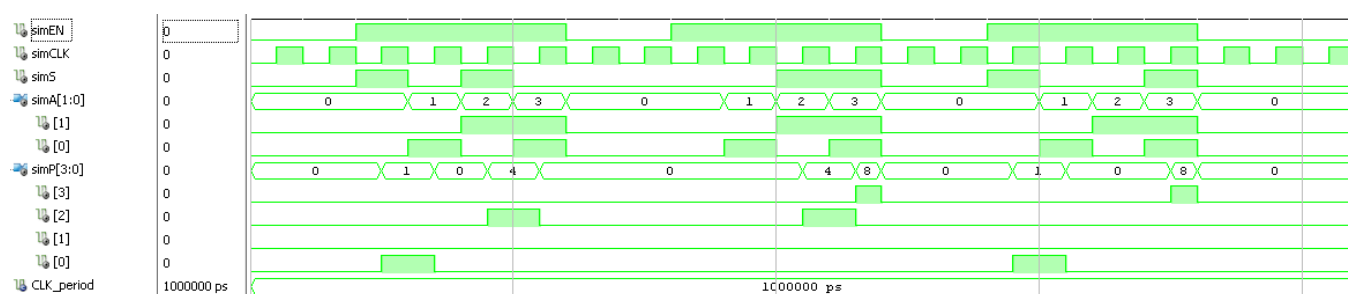


Abb. 20: Demultiplexing Betrieb mit 3 unterschiedlichen Datensätzen

Die Wahrheitstabelle veranschaulicht das durchschalten der Bits D_0 - D_3 zu den unterschiedlichen Adressen. Daraus ist ebenfalls ersichtlich das die Daten nicht zum selben Takt also Zeitgleich am Ausgang anliegen.

C	RSET ₁	RSET ₂	S _{1L}	S _{0L}	S _{1R}	S _{0R}	D ₀	D ₁	D ₂	D ₃	A ₀	A ₁	A ₂	A ₃	
1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	RESET
2	0	0	0	0	0	0	0	1	1	0	0	X	X	X	Bit0
3	0	0	0	1	0	1	0	1	1	0	X	1	X	X	Bit1
4	0	0	1	0	1	0	0	1	1	0	X	X	1	X	Bit2
5	0	0	1	1	1	1	0	1	1	0	X	X	X	0	Bit3

d.) Beantwortung der Fragen

- Frage 1: Funktion des Startsignals?**
 Das Startsignal setzt den asynchronen Zähler auf 0 so dass bei der untersten Adresse mit dem multiplexen/demultiplexen begonnen werden kann.
- Frage 2: Welchen Pegel hat ein Ausgang des Demultiplexer, wenn er nicht adressiert ist?**
 Dies hängt in den meisten Fällen von der Technologie ab. Es besteht die Möglichkeit den Ausgang über einen Pullup Widerstand auf **HIGH** zu halten oder über einen Pulldown Widerstand auf **LOW**. In unserem Übungsaufbau wurde der Ausgang auf LOW gehalten. Ebenso können die Ausgänge über ein Tri-State Register **High-Z** gehalten werden.
- Frage 3: Wie erfolgt die Adressierung von Multiplexer und Demultiplexer?**
 Um Daten richtig zu übertragen ist es notwendig das Multiplexer und Demultiplexer auf die gleiche Adresse eingestellt sind. Ist die Einstellung fehlerhaft so werden die Daten nicht richtig übertragen. Deshalb ist eine Synchronisierung zwischen Multiplexer und Multiplexer notwendig. Wird neben der Datenleitung der Takt mitgeführt so kann dies über Resettaster die den Zähler zurücksetzen eingestellt werden. Die einfachste Methode ist das Mitführen der Adressleitungen.

VI.) Anhang

- *Verbesserungen*
 - * Die gesamten in dieser Übung aufgebauten asynchronen Zähler können durch Synchronzähler ersetzt werden.
 - * Die Logisim Simulationen sind teilweise noch besser anzupassen.
- *Originalschaltungsaufbauten*

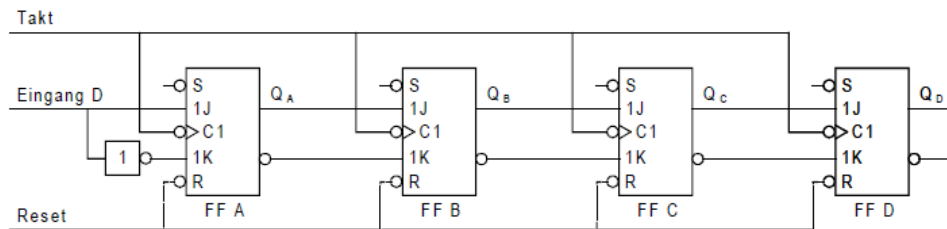


Abb. 21: Übungsaufbau 1 (JK-Schieberegister)

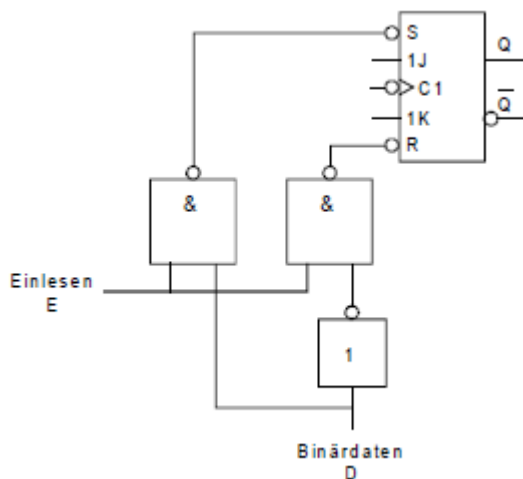


Abb. 22: Übungsaufbau 2 (Schieberegister mit paralleler Dateneingabe)

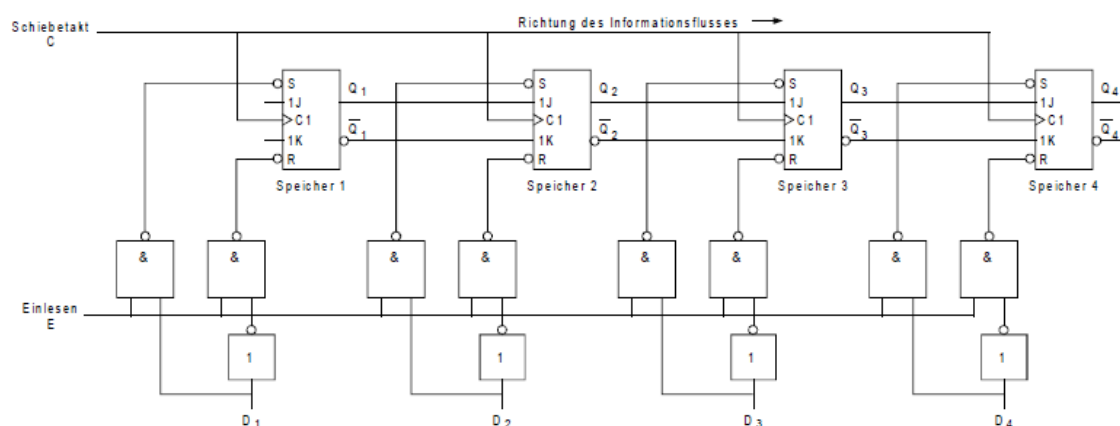


Abb. 23: Übungsaufbau 3 (4-Bit Schieberegister mit paralleler Dateneingabe)

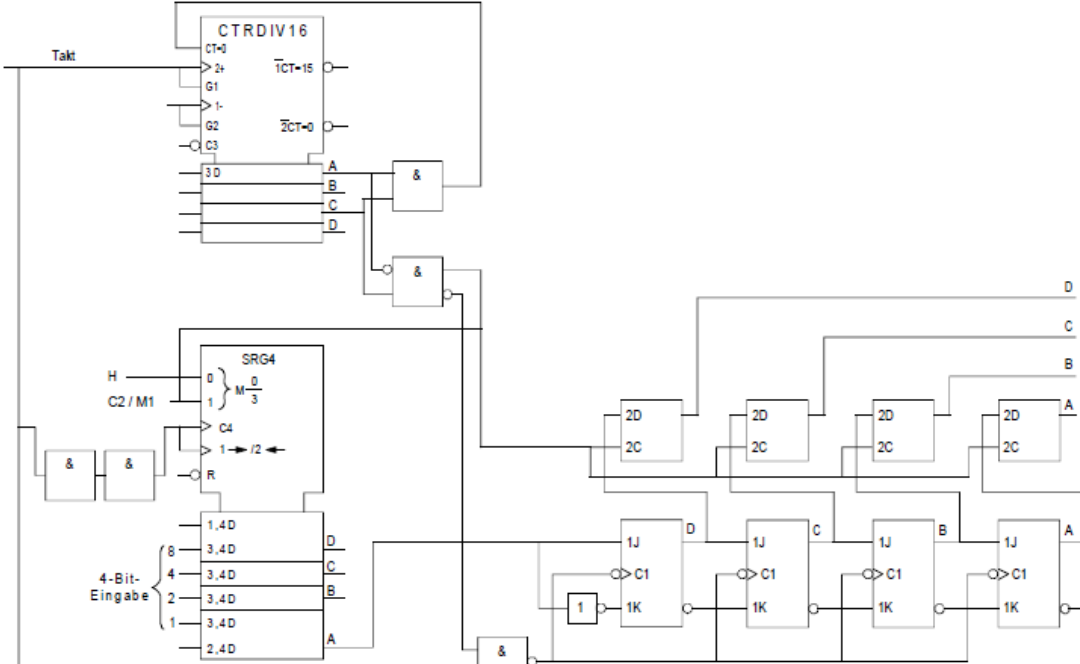


Abb. 24: Übungsaufbau 4 (Serielle 4-Bit Datenübertragung)

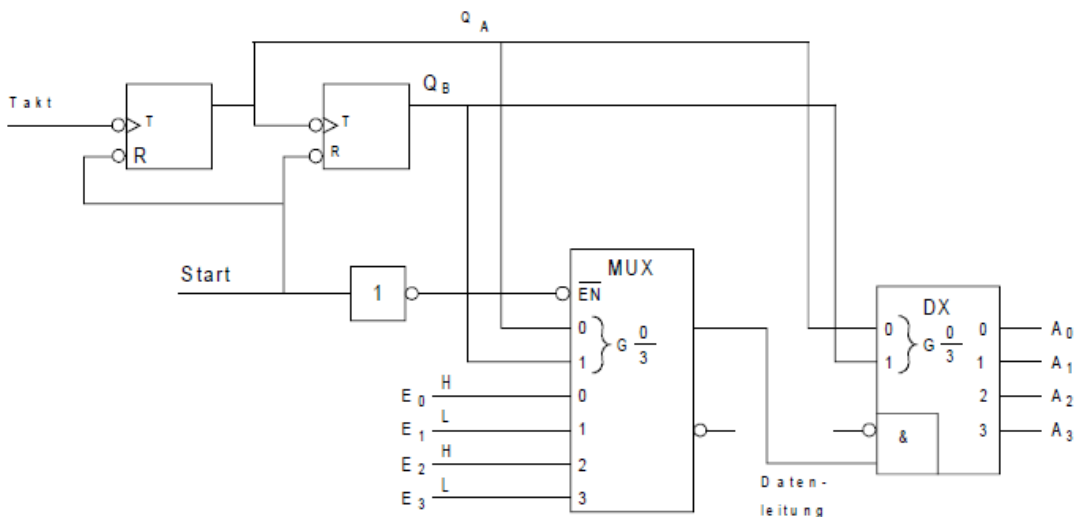


Abb. 25:Übungsaufbau (Multiplexing/Demultiplexing)