



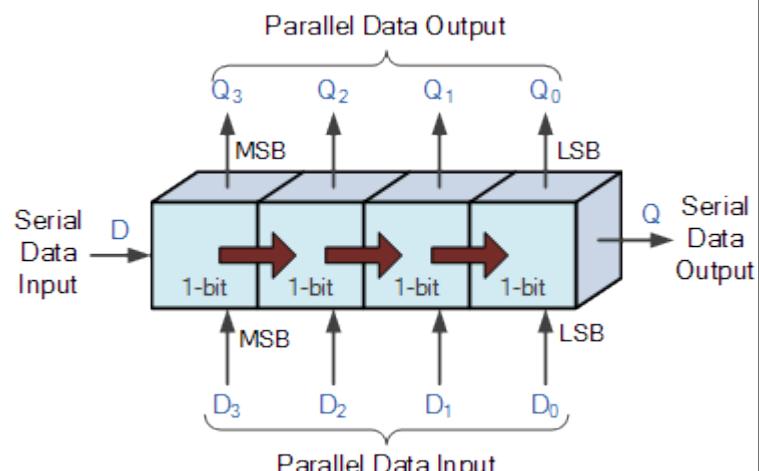
Register

„Die Flipflops stellen im Prinzip Informationsspeicher mit einer Kapazität von einem Bit dar. Größere Speicher können durch Verknüpfung mehrerer Flipflops realisiert werden. In der einfachsten Form werden die mit einer begrenzten Menge von Flipflops aufgebauten Speichereinheiten als Register bezeichnet. Kennzeichnend für derartige Register ist außerdem die gleichzeitige Taktung aller beteiligten Flipflops über ein einziges Taktsignal.“

- Register dienen zum kurzzeitigen Abspeichern von Informationen.
- Es handelt sich i.A. um sehr schnelle, flüchtige Speicherelemente, z.B. D-Ffs.
- Register haben eine geringe Speicherkapazität entsprechend der verwendeten Wortlänge in der jeweiligen Datenverarbeitungsanlage (bzw. Teilen oder Vielfachen davon), z. B. 8, 16, 32 oder 64 Bit.
- Anwendung z.B. zum Zwischenspeichern von Operanden bei arithmetischen Operationen.
- Die Eingabe der Daten kann seriell (serial input, SI) oder parallel erfolgen (parallel input, PI), ebenso die Ausgabe (serial output, SO bzw. parallel output, PO).
- Abhängig hiervon können die Register zum reinen Speichern verwendet werden (Speicherregister) oder auch zum Verschieben der Informationen nach rechts oder links (Schieberegister).¹

Die neben dargestellte Grafik² zeigt die vier verschiedenen Varianten:

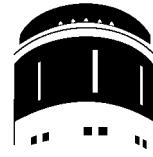
- Serial Input Parallel Output (SIPO) - das Register wird bitweise mit seriellen Daten geladen, wobei die gespeicherten Daten parallel am Ausgang verfügbar sind.
- Serial Input Serial Output (SISO) - die Daten werden seriell „REIN“ und „RAUS“ aus dem Register verschoben, jeweils ein Bit nach dem anderen in linker oder rechter Richtung unter Taktung.
- Parallel Input Serial Output (PISO) - die parallelen Daten werden gleichzeitig in das Register geladen und ein Bit nach dem anderen unter Taktsteuerung aus dem Register seriell verschoben.
- Parallel Input Parallel Output (PIPO) - die parallelen Daten werden gleichzeitig in das Register geladen und gemeinsam mit dem gleichen Takt an ihre jeweiligen Ausgänge übertragen.



Diese vier Varianten werden häufig entweder mit taktflanken gesteuerten D- oder JK-Flipflops realisiert.

1 <http://www.netzmafia.de/skripten/digitaltechnik/flipflop.html> Kapitel 7.6 Register abgerufen am 17.02.2019

2 <https://www.electronics-tutorials.ws/de/sequentielle/schieberegister.html> abgerufen am 17.02.2019
© Uwe Homm Version vom 18. Februar 2019 D:\Users\Uwe\Documents\(\Schule)\Lehrgang_Neu(BG TI)\(12BGTI-Q2) Digitale Steuerungstechnik\Skript\Digitaltechnik\09 Register.odt

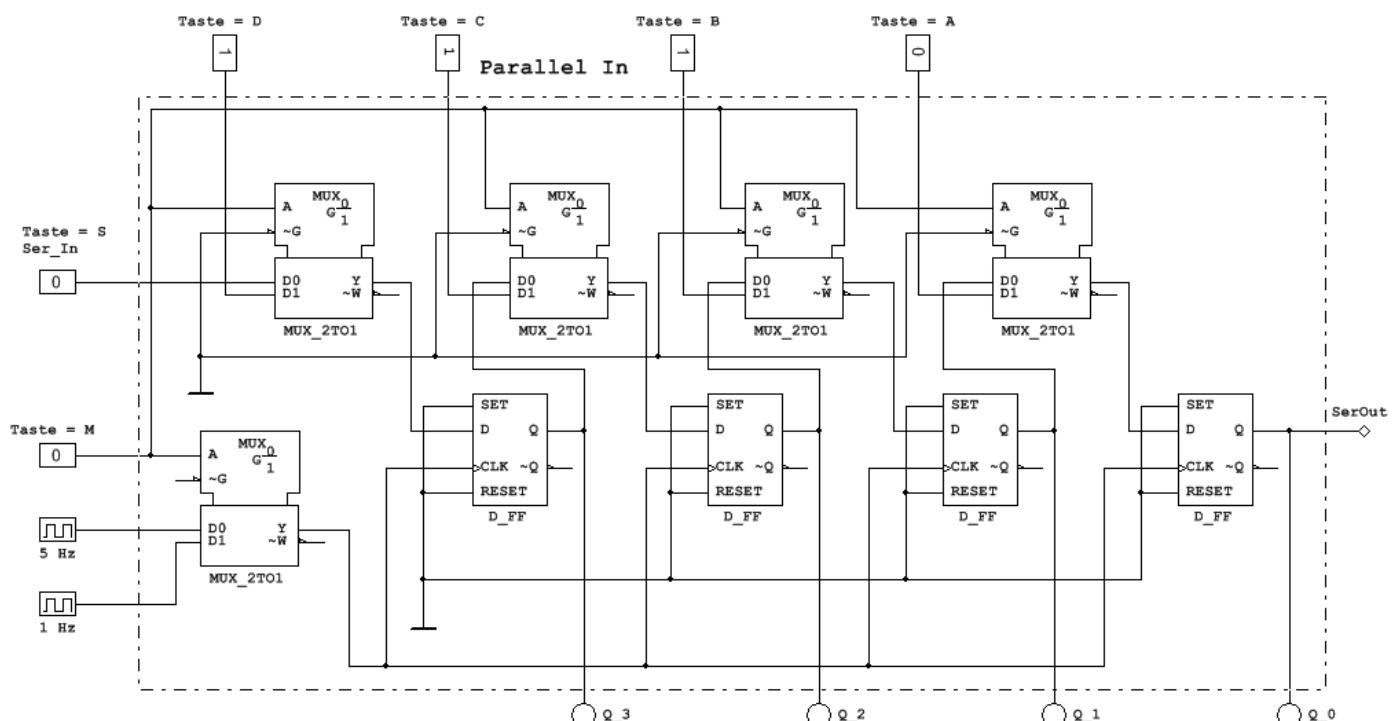


Anwendungen

Parallel Input Serial Output (PISO)

Diese Betriebsart ist gängig für die serielle Datenübertragung der in einem digitalen System parallel vorliegenden Daten. Das zu versendende Datum wird parallel in das Register geladen und anschließend mittels eines Taktsignals zeitlich hintereinander (seriell) an das entfernte System übertragen.

Ein solches Register wird daher als Schieberegister bezeichnet. Die binäre Information wird durch das Taktsignal von einem Flipflop zum nächsten übertragen. Der gespeicherte Wert des Flipflops mit der Wertigkeit 2^0 erscheint dabei auf der Sendeleitung.



Die obige Schaltung demonstriert die Funktionsweise eines solchen PISO-Registers.

Mit dem Modus-Eingang M wird eingestellt, ob die vier Flipflops parallel geladen werden ($M=1$) oder ob sie im Schiebemodus arbeiten ($M=0$).

Bei $M=1$ werden die Flipflop-Eingänge mit dem Anschluss D1 des 2-zu-1 Multiplexer verbunden, bei $M=0$ hingegen mit dem Ausgang des vorherigen Flipflops. Lediglich der D-Eingang der Flipflop-Stufe 2^3 (ganz links) wird in diesem Fall mit dem Eingang **Ser_In** verbunden.

Der Multiplexer, an dem die beiden Taktsignale angeschlossen, wählt bei $M=1$ für den Ladevorgang den langsamen Takt aus, bei $M=0$ den schnelleren Takt für den Schiebevorgang.

Serial Input Parallel Output (SIPO)

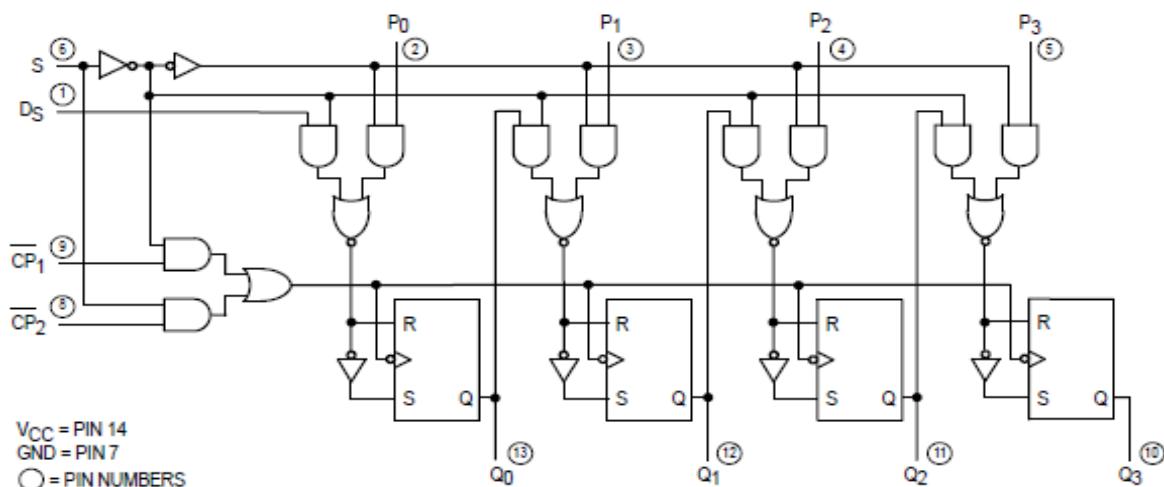
Aufgrund des seriellen Eingangs **Ser_In** kann diese Schaltung mit einer kleinen Modifikation auch als SIPO-Register verwendet werden. Hierfür müssen die vier Ausgänge der Flipflops (an denen momentan die Kontrolllampen angeschlossen sind) ebenfalls heraus geführt werden.

Genau diese Anordnung findet sich im TTL-IC 74LS95.

Hier ein Auszug aus dessen Datenblatt:

**SN54/74LS95B**

LOGIC DIAGRAM

**FUNCTIONAL DESCRIPTION**

The LS95B is a 4-Bit Shift Register with serial and parallel synchronous operating modes. It has a Serial (D_S) and four Parallel (P_0 - P_3) Data inputs and four Parallel Data outputs (Q_0 - Q_3). The serial or parallel mode of operation is controlled by a Mode Control input (S) and two Clock Inputs (CP_1) and (CP_2). The serial (right-shift) or parallel data transfers occur synchronous with the HIGH to LOW transition of the selected clock input.

When the Mode Control input (S) is HIGH, $\overline{CP_2}$ is enabled. A HIGH to LOW transition on enabled CP_2 transfers parallel data from the P_0 - P_3 inputs to the Q_0 - Q_3 outputs.

When the Mode Control input (S) is LOW, CP_1 is enabled. A

HIGH to LOW transition on enabled $\overline{CP_1}$ transfers the data from Serial input (D_S) to Q_0 and shifts the data in Q_0 to Q_1 , Q_1 to Q_2 , and Q_2 to Q_3 respectively (right-shift). A left-shift is accomplished by externally connecting Q_3 to P_2 , Q_2 to P_1 , and Q_1 to P_0 , and operating the LS95B in the parallel mode (S = HIGH).

For normal operation, S should only change states when both Clock inputs are LOW. However, changing S from LOW to HIGH while CP_2 is HIGH, or changing S from HIGH to LOW while CP_1 is HIGH and CP_2 is LOW will not cause any changes on the register outputs.

MODE SELECT — TRUTH TABLE

OPERATING MODE	INPUTS					OUTPUTS			
	S	CP ₁	CP ₂	D _S	P _n	Q ₀	Q ₁	Q ₂	Q ₃
Shift	L	—	X	I	X	L	q ₀	q ₁	q ₂
	L	—	X	h	X	H	q ₀	q ₁	q ₂
Parallel Load	H	X	—	X	P _n	P ₀	P ₁	P ₂	P ₃
Mode Change	—	L	L	X	X	X	X	X	No Change
	—	L	L	X	X	X	X	X	No Change
	—	H	L	X	X	X	X	X	No Change
	—	H	L	X	X	X	X	X	Undetermined
	—	L	H	X	X	X	X	X	Undetermined
	—	L	H	X	X	X	X	X	No Change
	—	H	H	X	X	X	X	X	No Change

L = LOW Voltage Level

H = HIGH Voltage Level

X = Don't Care

I = LOW Voltage Level one set-up time prior to the HIGH to LOW clock transition.

h = HIGH Voltage Level one set-up time prior to the HIGH to LOW clock transition.

P_n = Lower case letters indicate the state of the referenced input (or output) one set-up time prior to the HIGH to LOW clock transition.



Übungen

Nutzen Sie für die Übungen auch das folgende PDF:

https://home.zhaw.ch/~dqtm/dt1/vorlesung/DT109_08_schieberegister.pdf

1. Bauen Sie die MultiSim-Schaltung auf. Testen Sie die Schaltung.
2. Überlegen Sie, wie ein IEC-Symbol für das 74LS95 aussehen kann und skizzieren Sie es.
Hinweis: Orientieren Sie sich an dem IEC Symbol für das 74F194.
3. Informieren Sie sich über sogenannte „Ringzähler“.
Bauen Sie den nachfolgenden Ringzähler mit Multisim auf. Verwenden Sie aus der TTL-Bibliothek das IC 74F194 (4-Bit Universal Schieberegister).
Verwenden Sie für den Logikanalysator die dargestellten Einstellungen!
Informieren Sie sich im Datenblatt, mit welchen Steuersignal-Kombinationen ein 4-Bit Wert geladen werden kann und wie ein Rechtsschieben veranlasst wird.
Laden Sie den Ringzähler mit dem dargestellten Wert „1000“.

Jeder Ausgang des Ringzählers teilt das Takt signal durch 4.
Wie könnte eine Taktteilung 1 zu 3 erreicht werden?

4. Ermitteln Sie, um was es sich bei einem Johnson-Zähler handelt.
Ändern Sie die Schaltung zu einem 4-Bit Johnson Zähler.

