



Academie voor Technology, Innovation &  
Society Delft  
Academie voor ICT & Media

# Inleiding Digitale Techniek

Week 7 – Schuifregisters  
Jesse op den Brouw  
INLDIG/2013-2014

DE HAAGSE  
HOGESCHOOL

# Schuifregisters

- In de digitale techniek en met name in de digitale communicatie wordt veel gebruik gemaakt van *seriële transmissie*.
- Het voordeel ten opzichte van *parallele transmissie* is overduidelijk: er is slechts een beperkt aantal verbindingen nodig om informatie (bits) te verzenden of te ontvangen.
- Het nadeel is dat het versturen van een byte of word veel langer duurt dan bij parallele transmissie.

# Schuifregisters

- Voorbeelden van seriële transmissiesystemen:

RS232 (a.k.a. COM-poort in de PC)

Ethernet

I<sup>2</sup>C (a.k.a. Two Wire Interface)

SPI

Infrarood afstandsbediening

USB

SATA

CD/DVD

# Schuifregisters

- Bij het gebruik van seriële transmissie worden schuifregisters gebruikt.
- Bij *verzenden* van data gaat het omzetten van parallel naar serieel als volgt:

Eerst wordt de (parallele) data *geladen*.

Daarna wordt de data serieel *naar buiten geschoven*.

- Dit heet PISO: Parallel In Serial Out

# Schuifregisters

- Bij *ontvangen* van data gaat het omzetten van serieel naar parallel als volgt:

Eerst wordt de data serieel *naar binnen geschoven*.

Daarna wordt de (parallele) data *beschikbaar gesteld*.

- Dit heet SIPO: Serial In Parallel Out

# Schuifregisters

- De grote vraag is altijd:

Welk bit wordt als eerste naar buiten (of binnen) geschoven?

- Dat is systeemafhankelijk:

RS232 – minst significante bit eerst

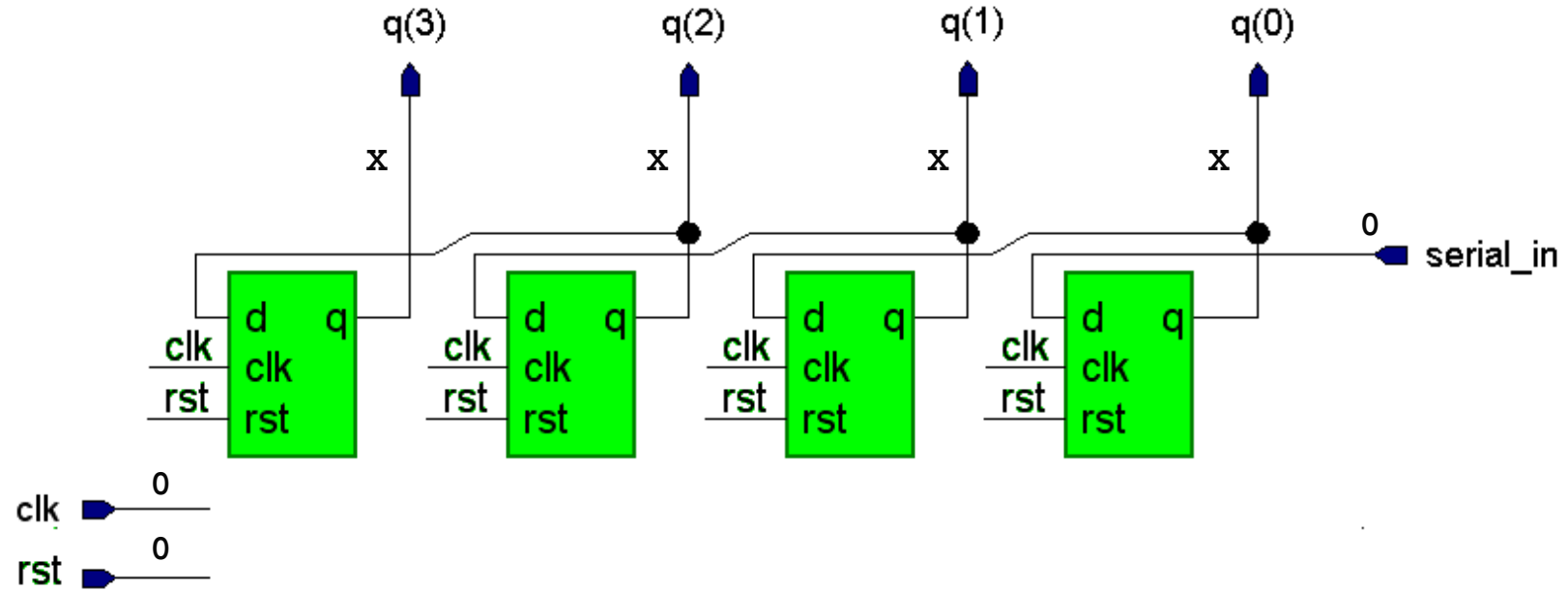
Ethernet – meest significante bit eerst.

- Goed opletten dus.

# Schuifregisters

- Daarnaast bestaan er ook *bi-directionele schuifregisters*. Dat zijn schuifregister die twee kanten kunnen opschuiven.
- Bij het bespreken van schuifregisters wordt gesproken van *linksom schuiven* (left shift) en *rechtsom schuiven* (right shift).
- Linksom is letterlijk zoals op papier zou worden getekend van rechts naar links.
- Rechtsom is letterlijk zoals op papier zou worden getekend van links naar rechts.

000101100000

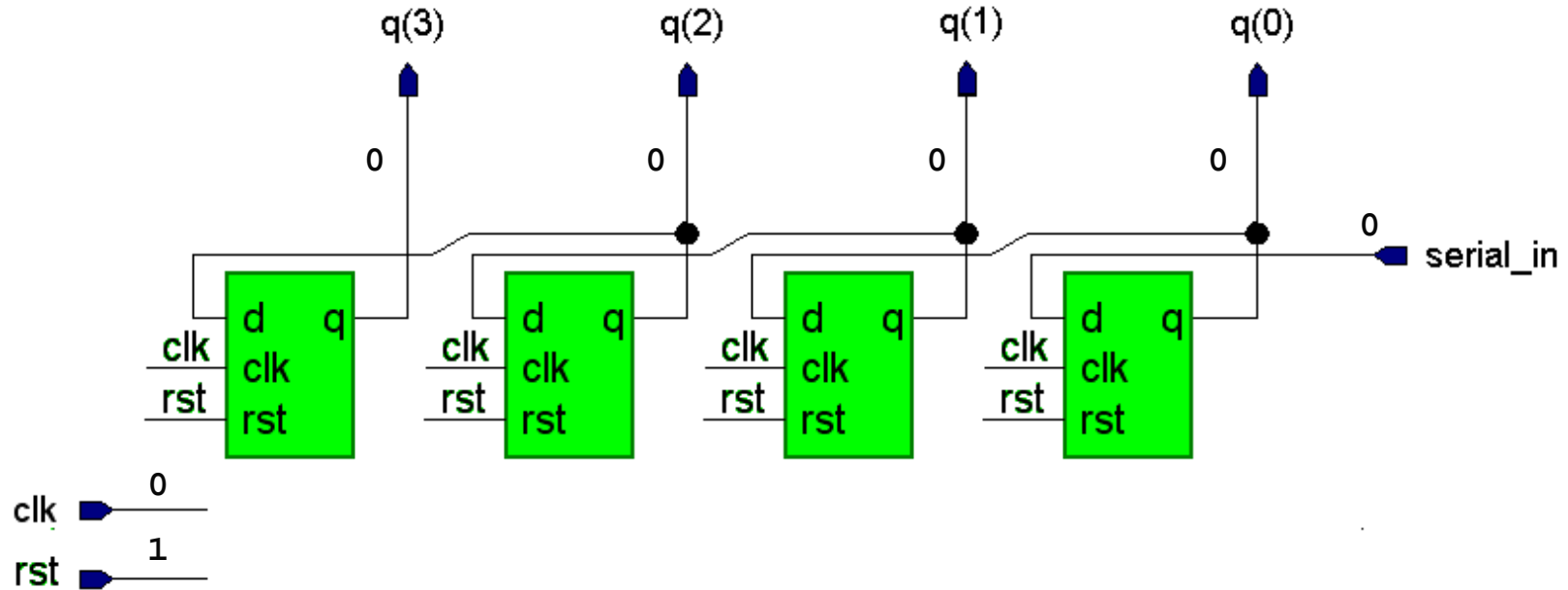


Serial Input Parallel Output Left Shiftregister





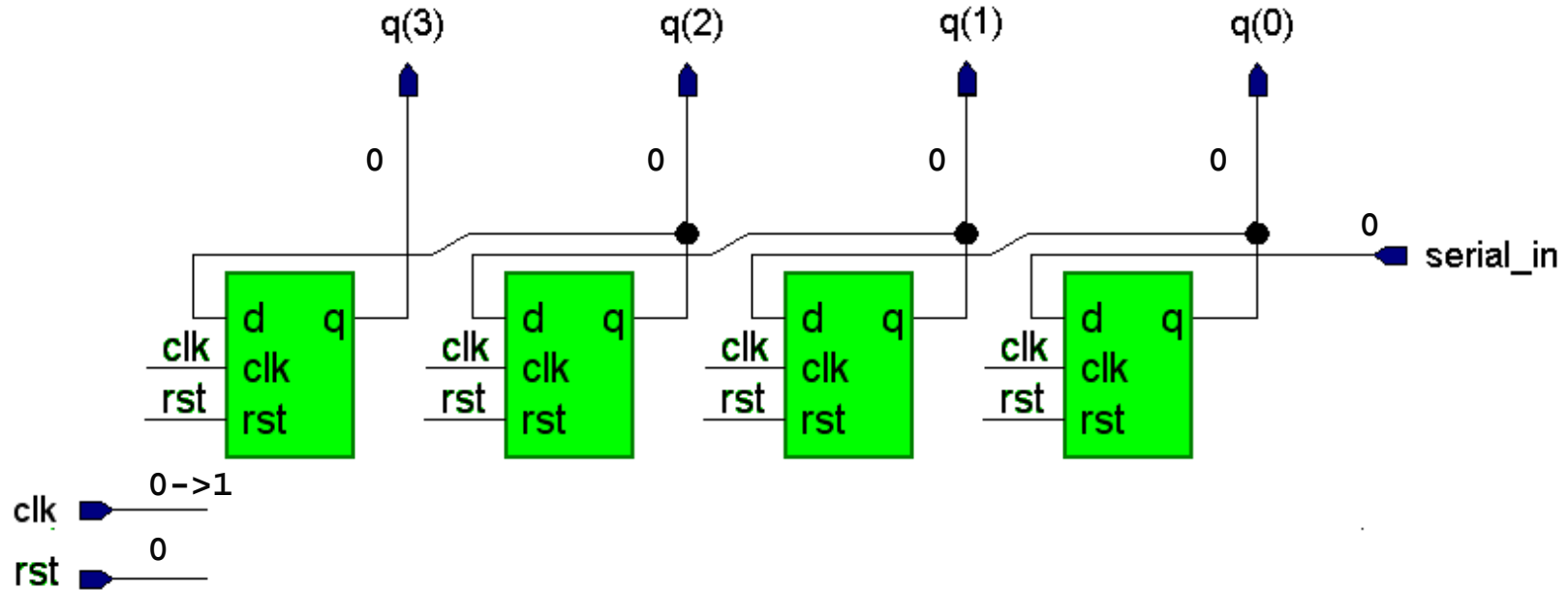
000101100000



Serial Input Parallel Output Left Shiftregister



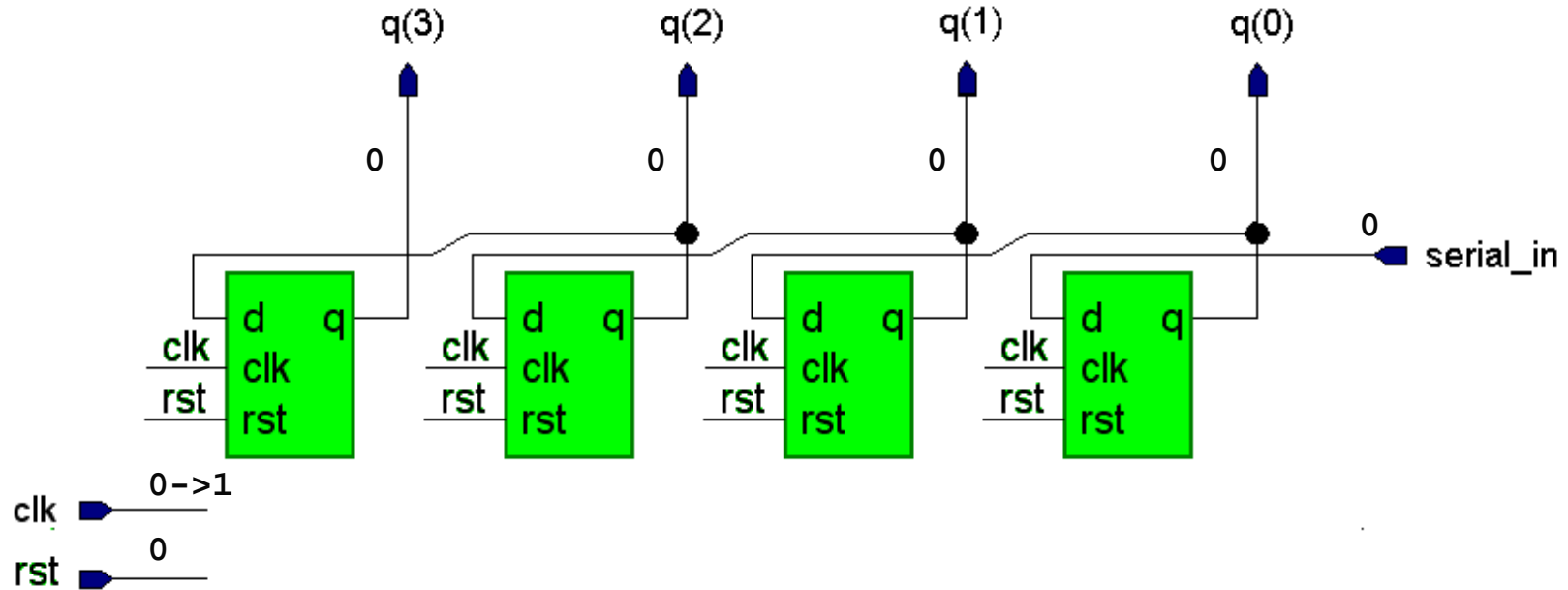
000101100000



Serial Input Parallel Output Left Shiftregister



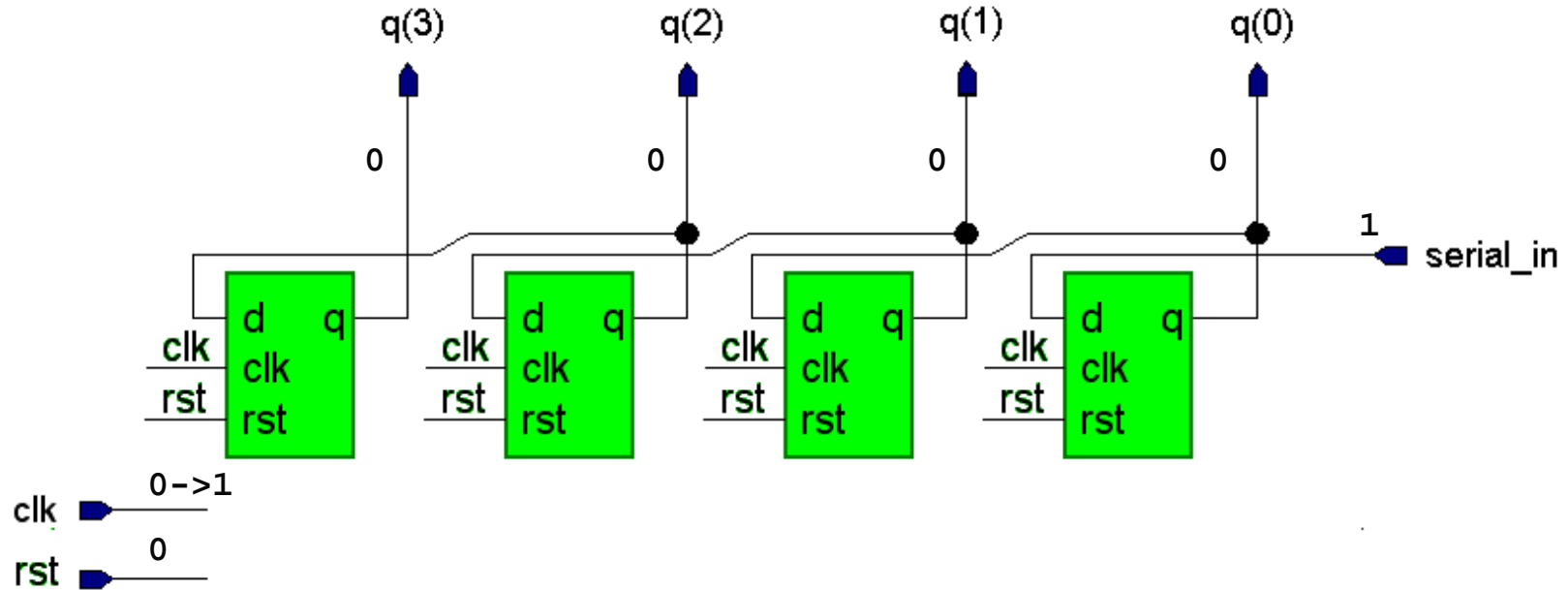
000101100000



Serial Input Parallel Output Left Shiftregister



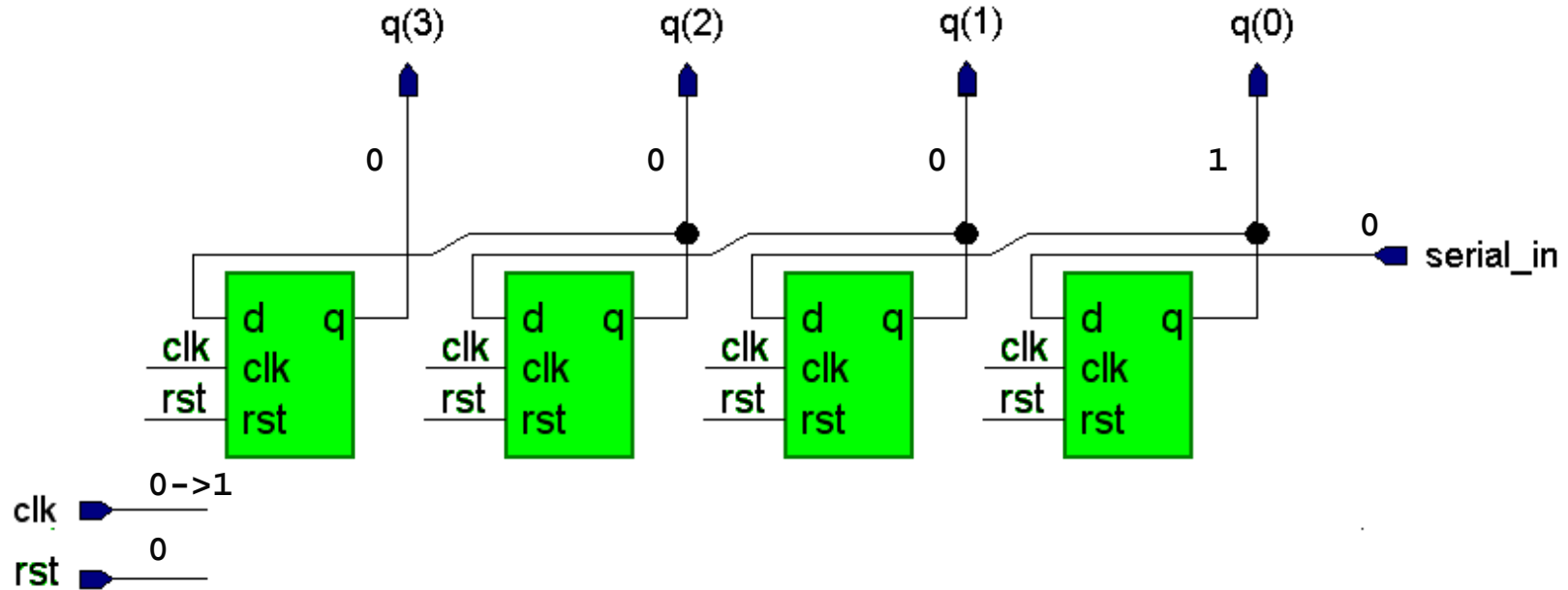
000101100000



Serial Input Parallel Output Left Shiftregister



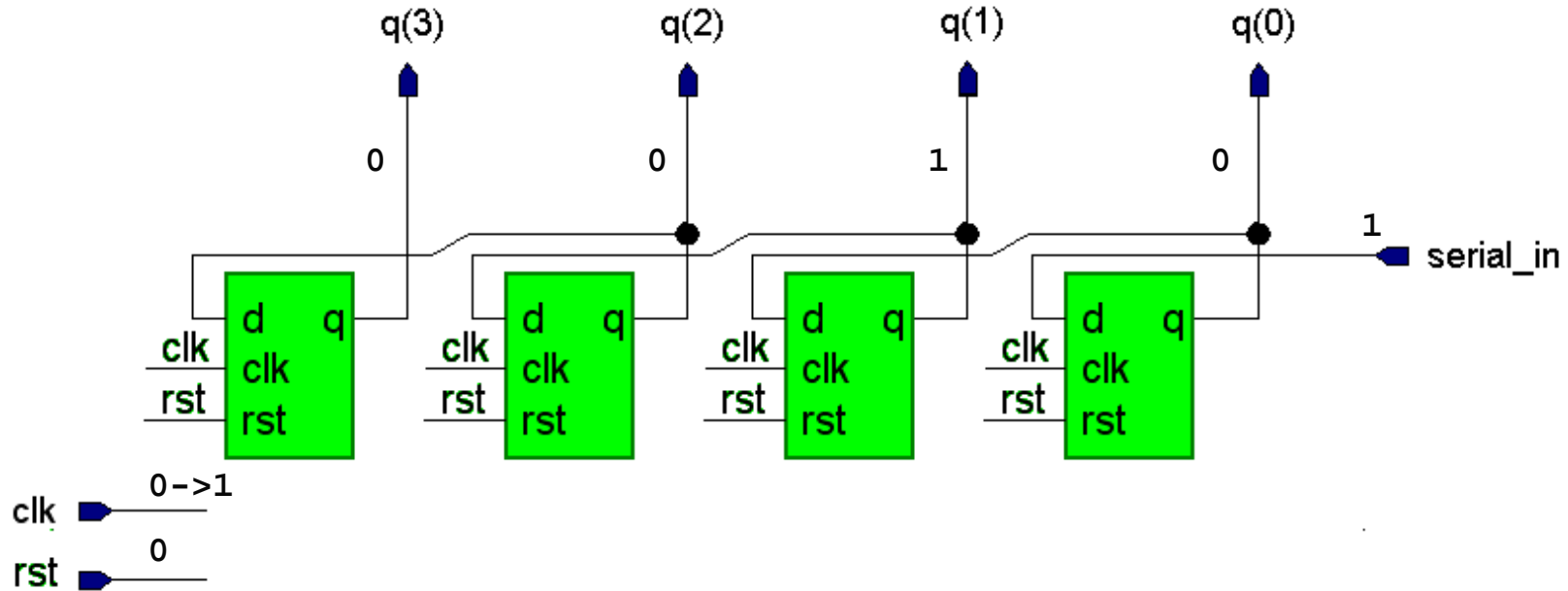
000101100000



Serial Input Parallel Output Left Shiftregister



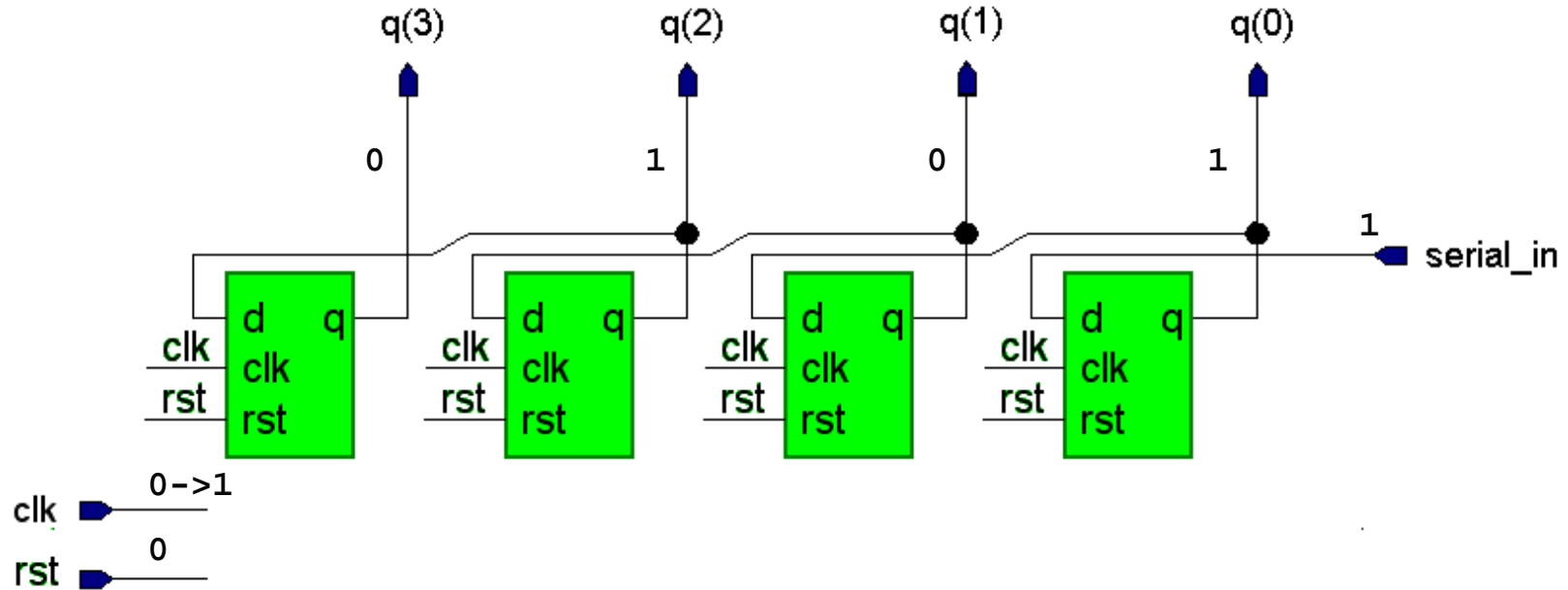
000101100000



Serial Input Parallel Output Left Shiftregister



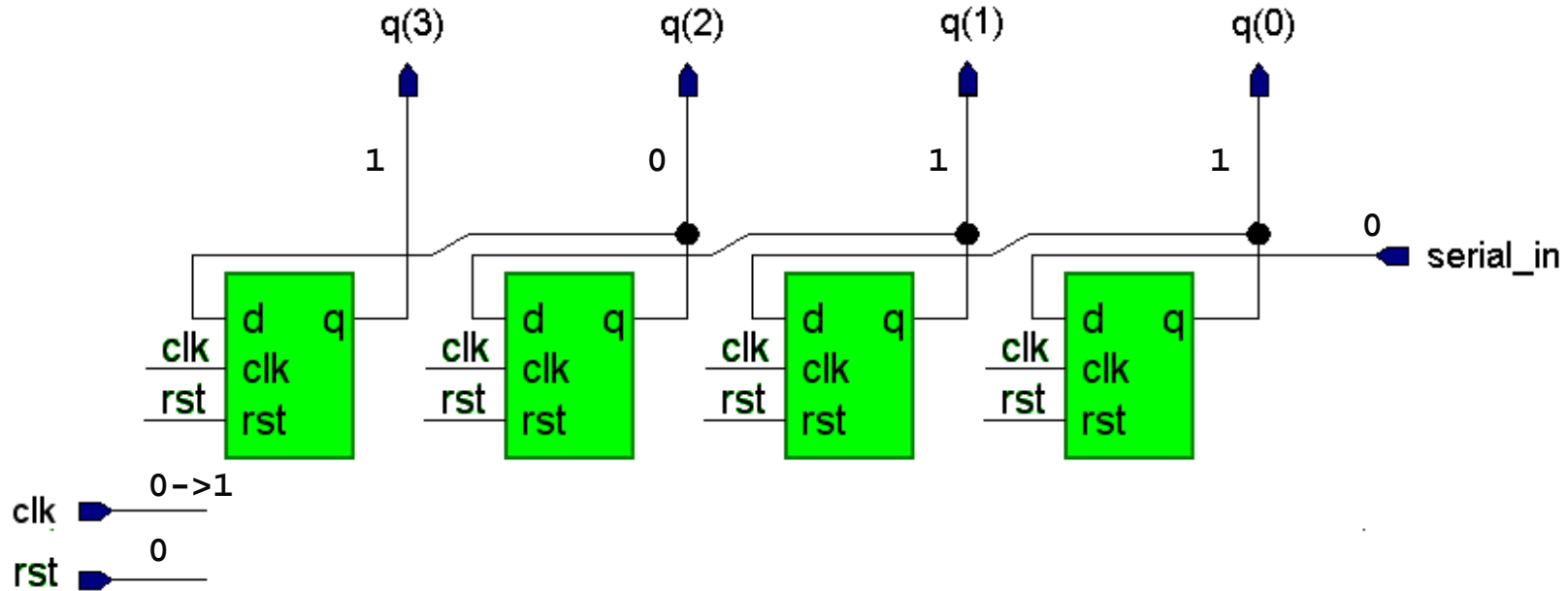
000101100000



Serial Input Parallel Output Left Shiftregister



000101100000

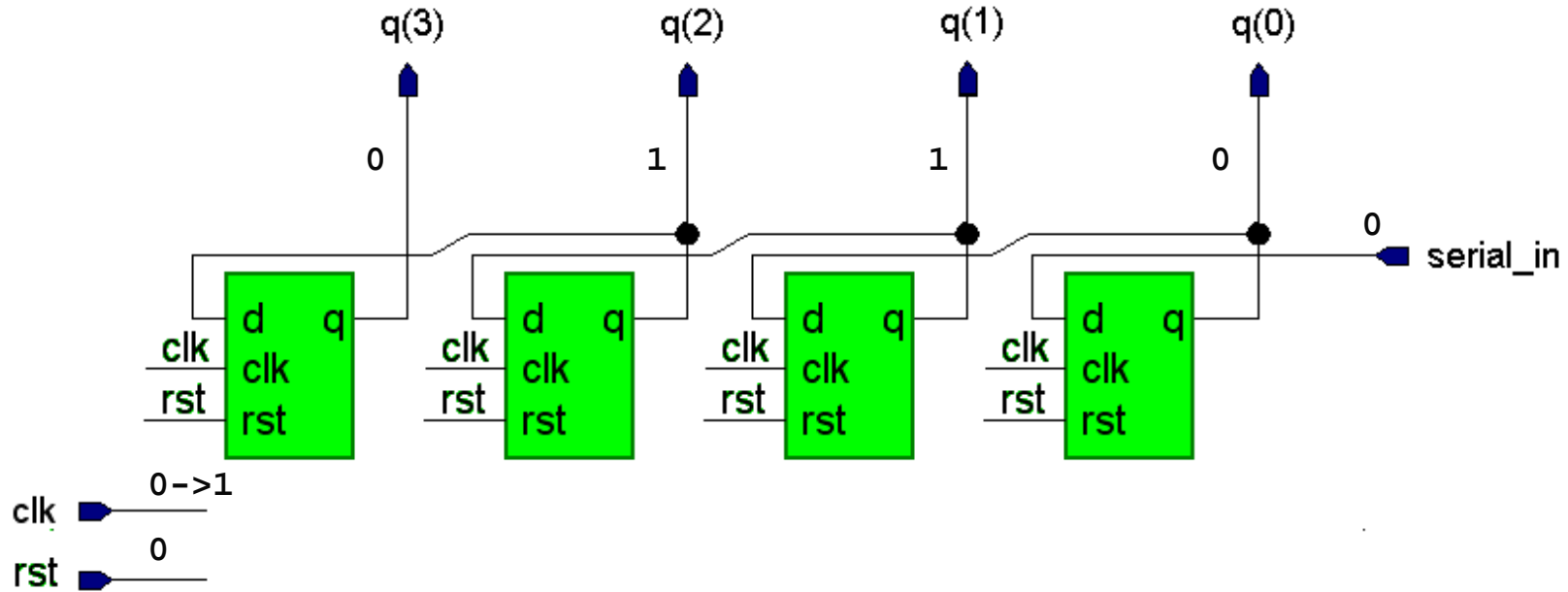


Serial Input Parallel Output Left Shiftregister





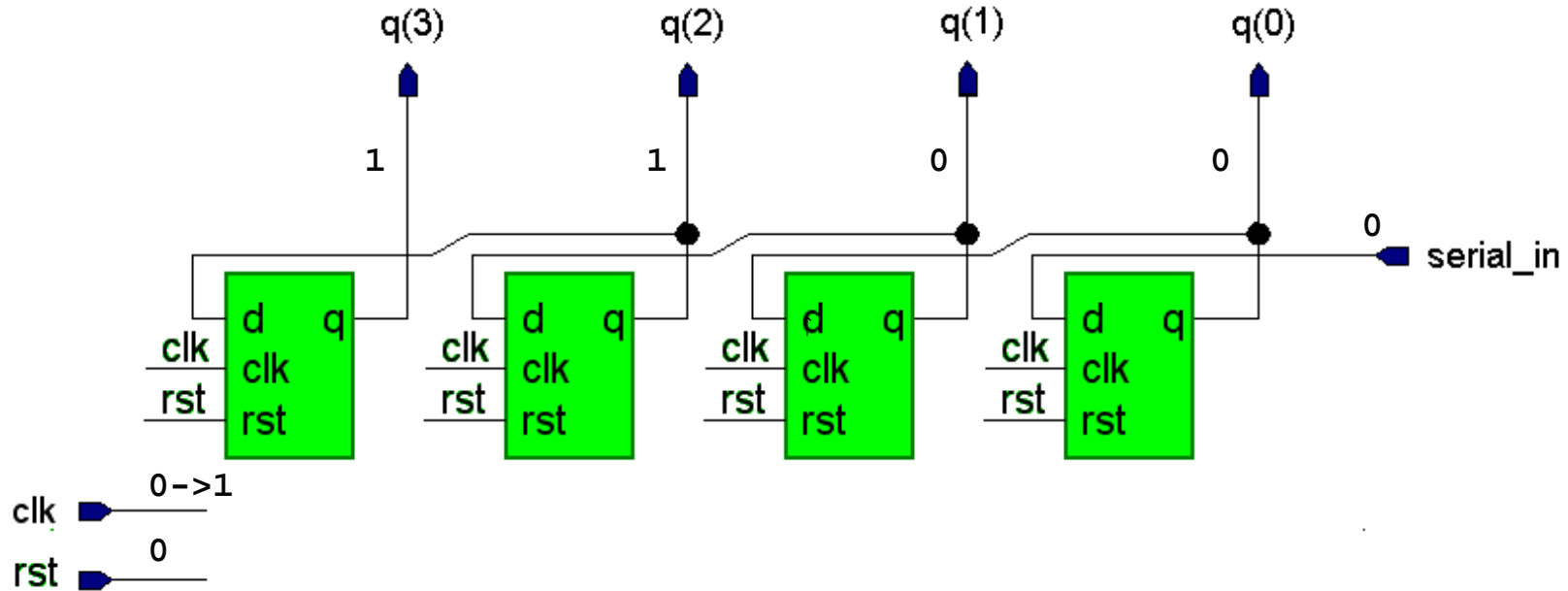
000101100000



Serial Input Parallel Output Left Shiftregister



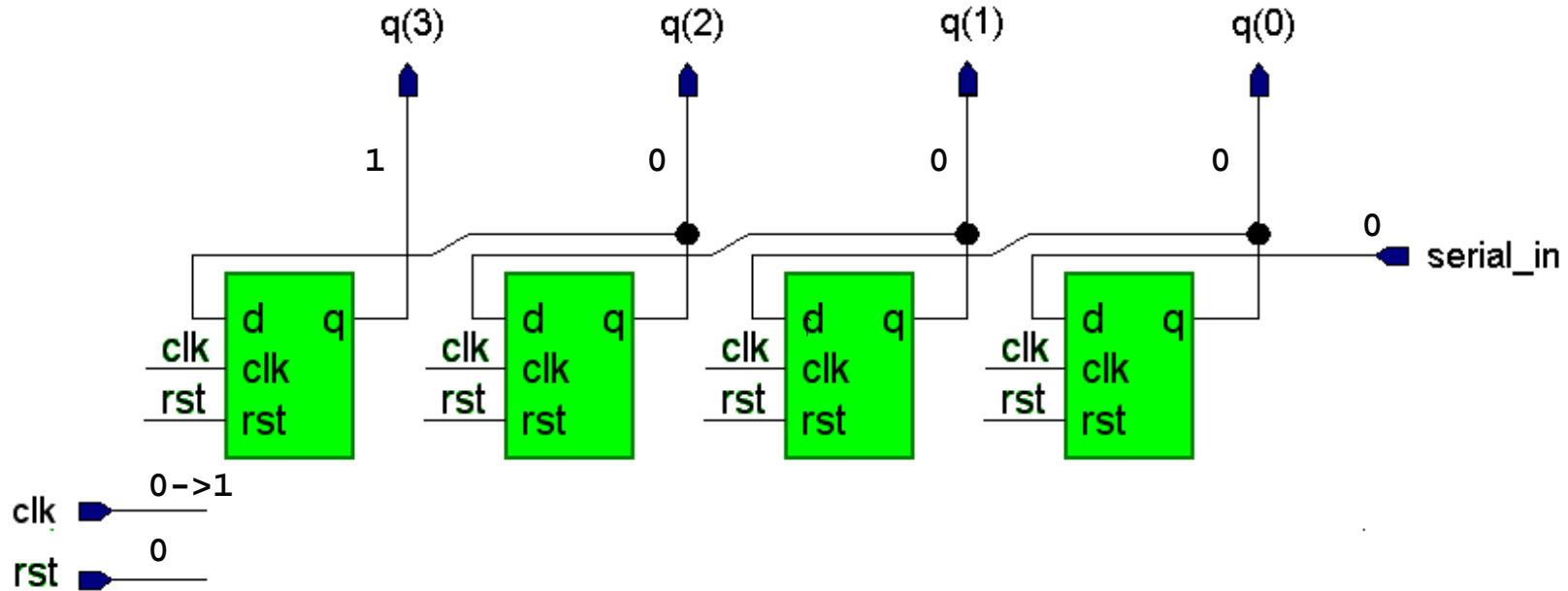
000101100000



Serial Input Parallel Output Left Shiftregister



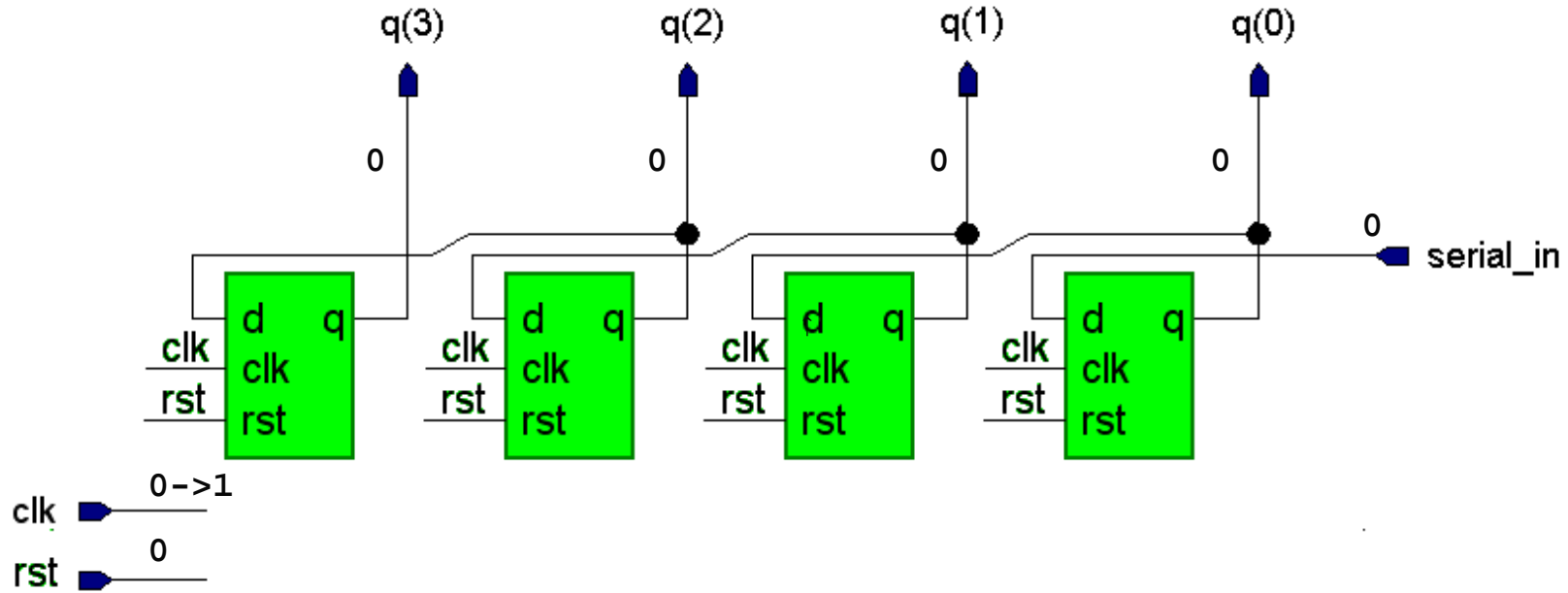
000101100000



Serial Input Parallel Output Left Shiftregister



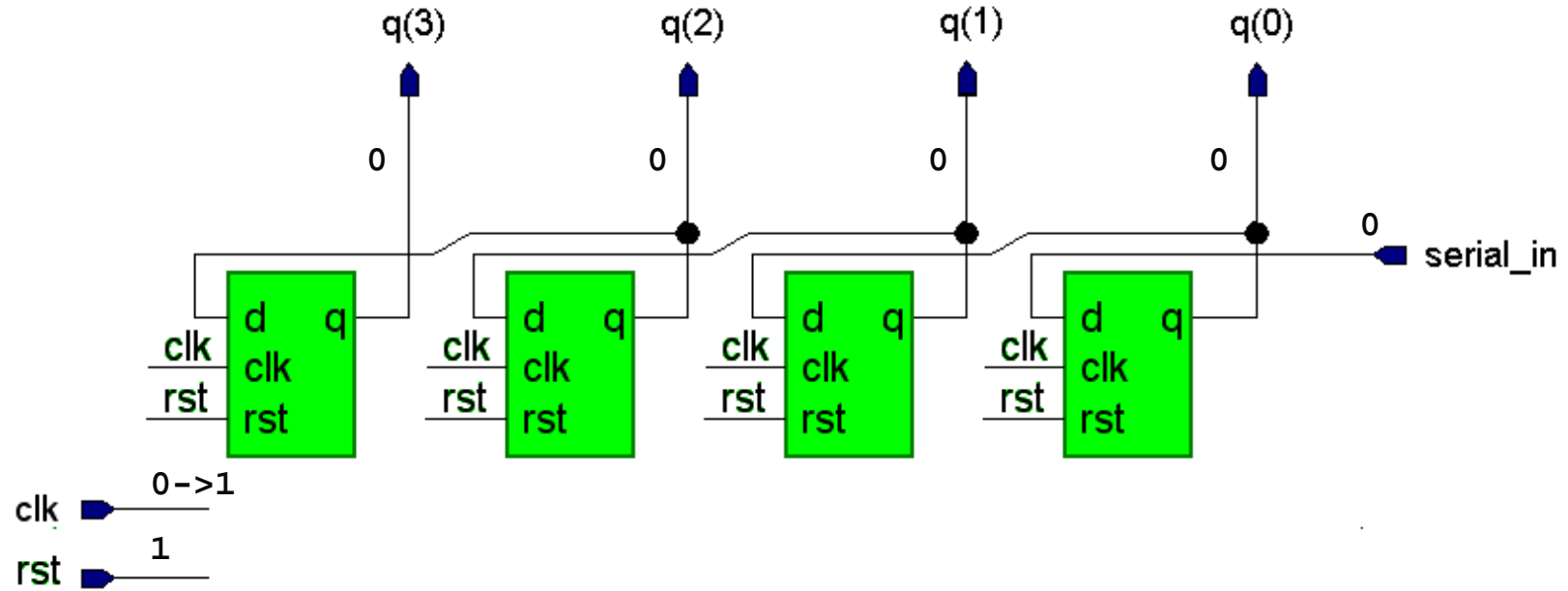
000101100000



Serial Input Parallel Output Left Shiftregister



000101100000



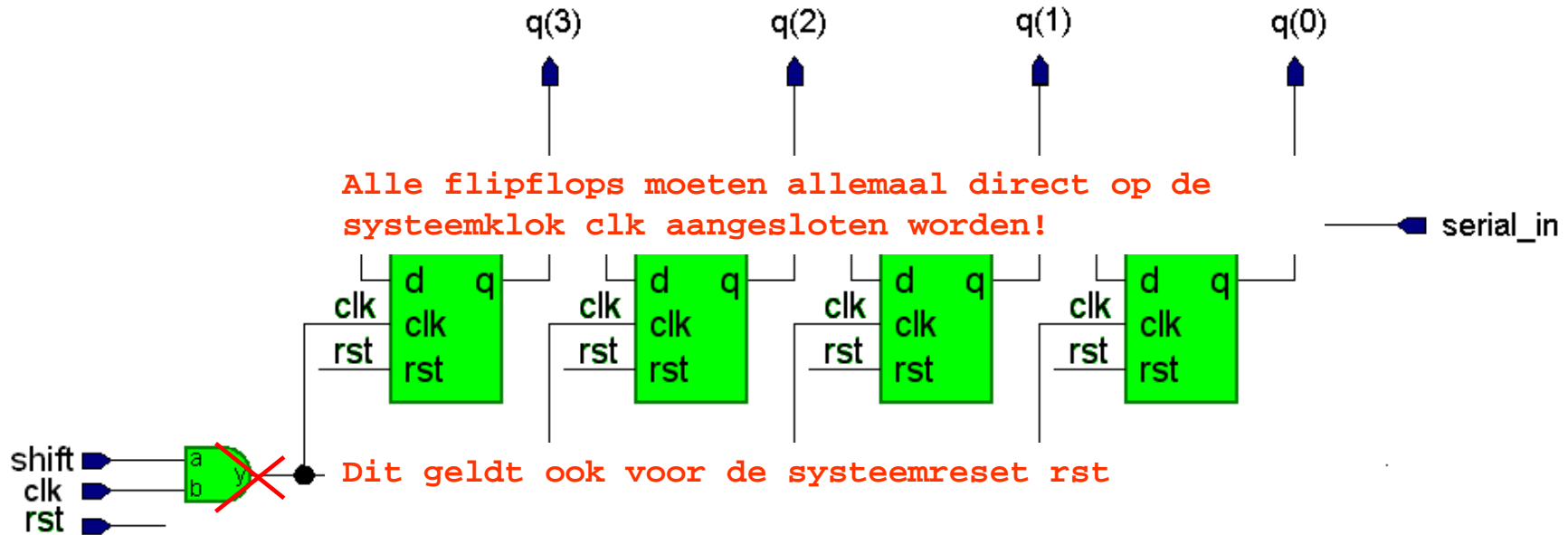
Serial Input Parallel Output Left Shiftregister



# Schuifregisters

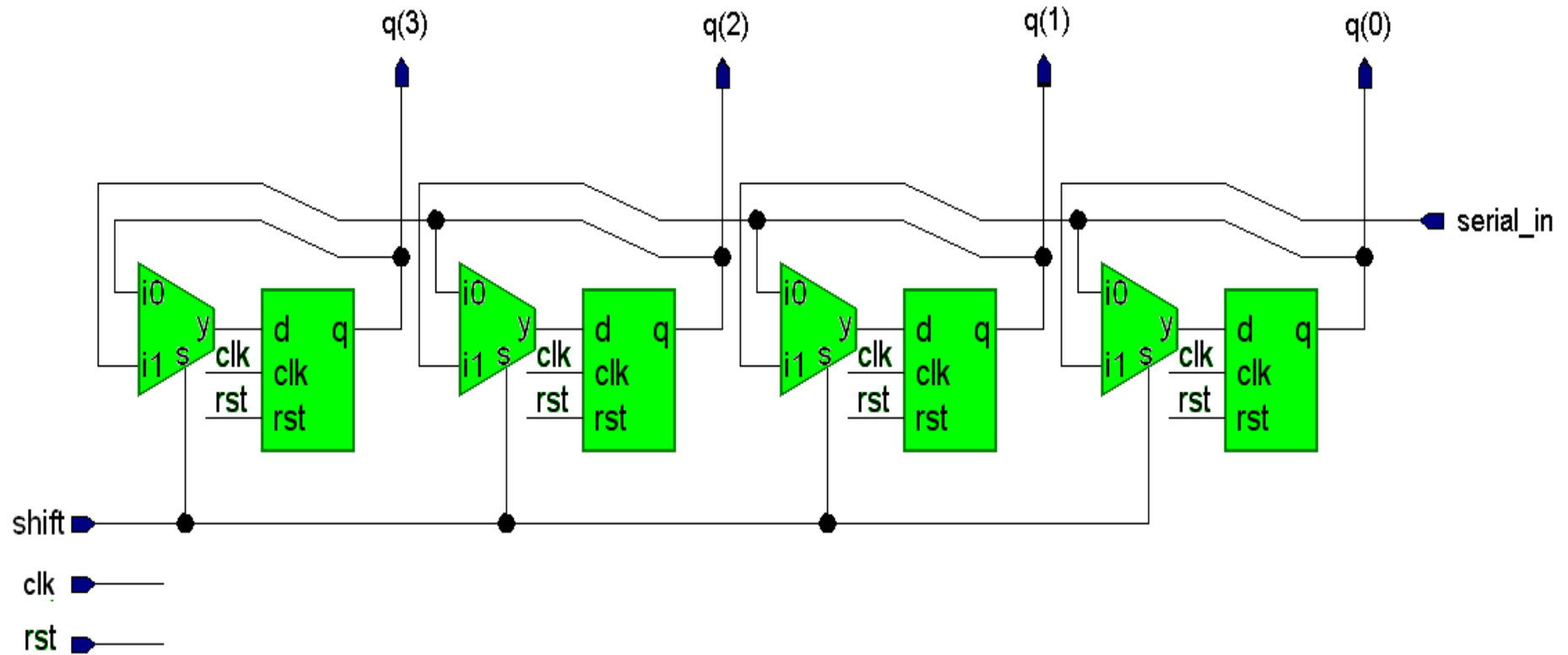
- Na het inlezen van de seriële data, is het beschikbaar als parallelle data.
- Meestal moet die data even vastgehouden worden, dus er mag dan niet geschoven worden.
- Dat kan eenvoudig gerealiseerd worden door multiplexers toe te voegen en een stuurlijn.
- Dit signaal wordt *shift* genoemd. Bij  $shift = 0$  wordt de data vastgehouden (onthouden), bij  $shift = 1$  wordt er geschoven.

000101100000



Serial Input Parallel Output Left Shiftregister



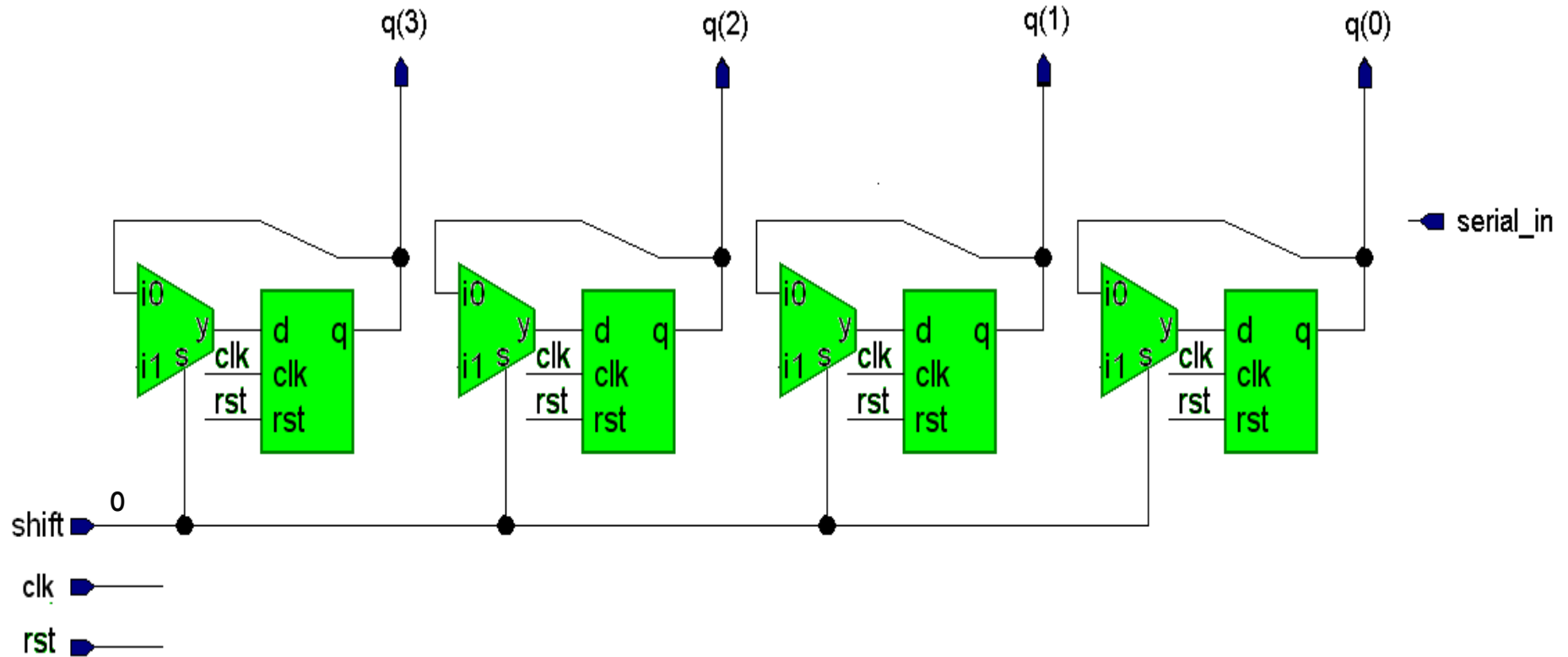


Serial Input Parallel Output Left Shiftregister





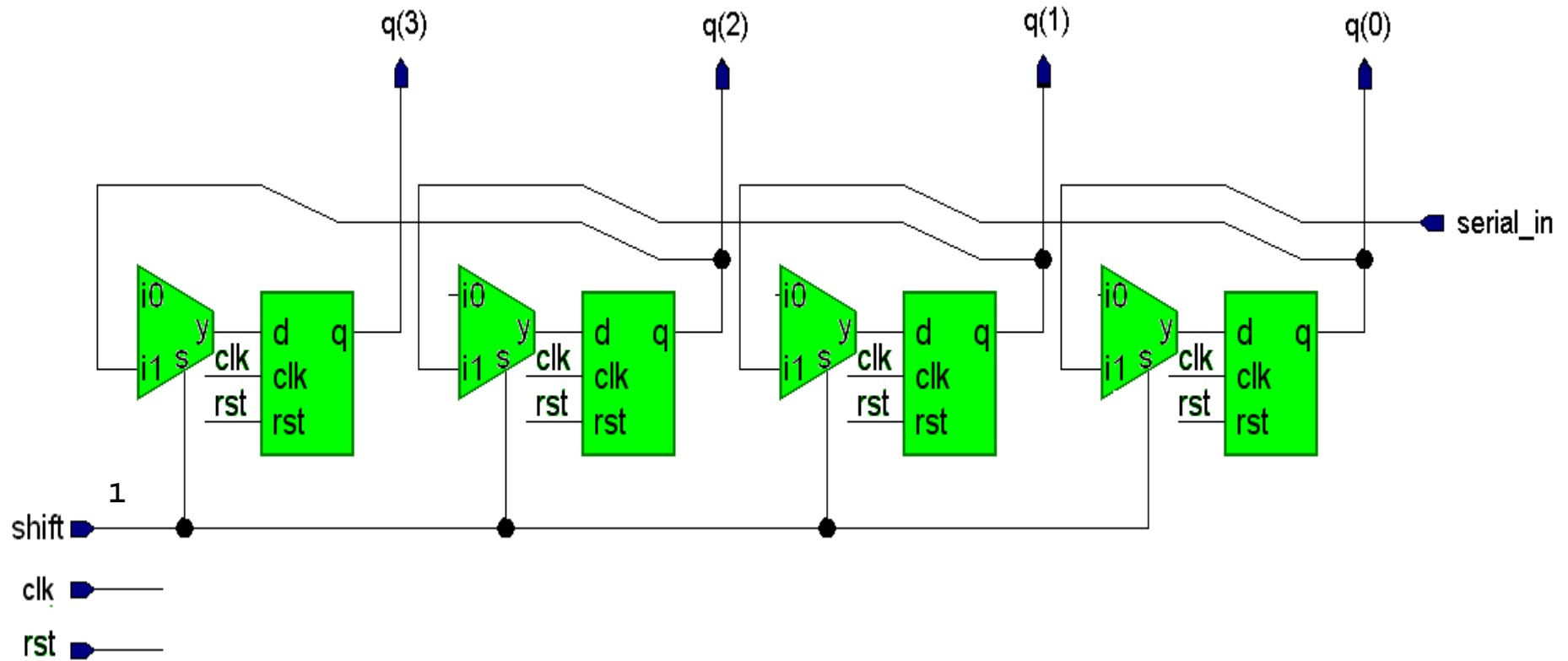
Onthouden (shift = 0)



Serial Input Parallel Output Left Shiftregister



Schuiven (shift = 1)



Serial Input Parallel Output Left Shiftregister



# Schuifregisters

- Het schuifregister kan nu onthouden en schuiven, dus deze is geschikt als ontvanger voor seriële data.
- Als er data uit geschoven moet worden, heeft het schuifregister ook een laadmogelijkheid nodig.
- Hiervoor zijn nu twee stuursignalen nodig en 3x1 multiplexers.
- Als voorbeeld een schuifregister voor RS232-communicatie.

# Schuifregisters

- RS232-communicatie gaat als volgt\*):

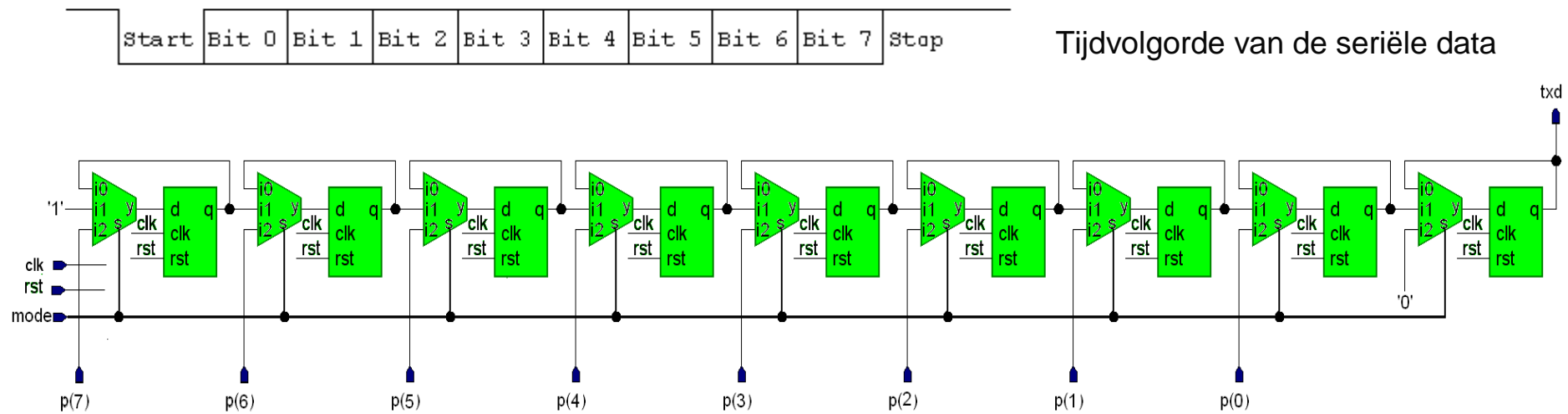
Eerst wordt een *startbit* naar buiten geschoven. Deze is 0.  
Daarna worden de 8 databits naar buiten geschoven.  
Vervolgens volgt er één stopbit.

- Dit wordt 8N1 genoemd, er is geen *parity check*.
- Er worden 9 schuifsecties gemaakt, de eerste wordt geladen met een 0 (startbit), de laatste schuift een 1 in (stopbit).

nb: er zijn ook andere mogelijkheden

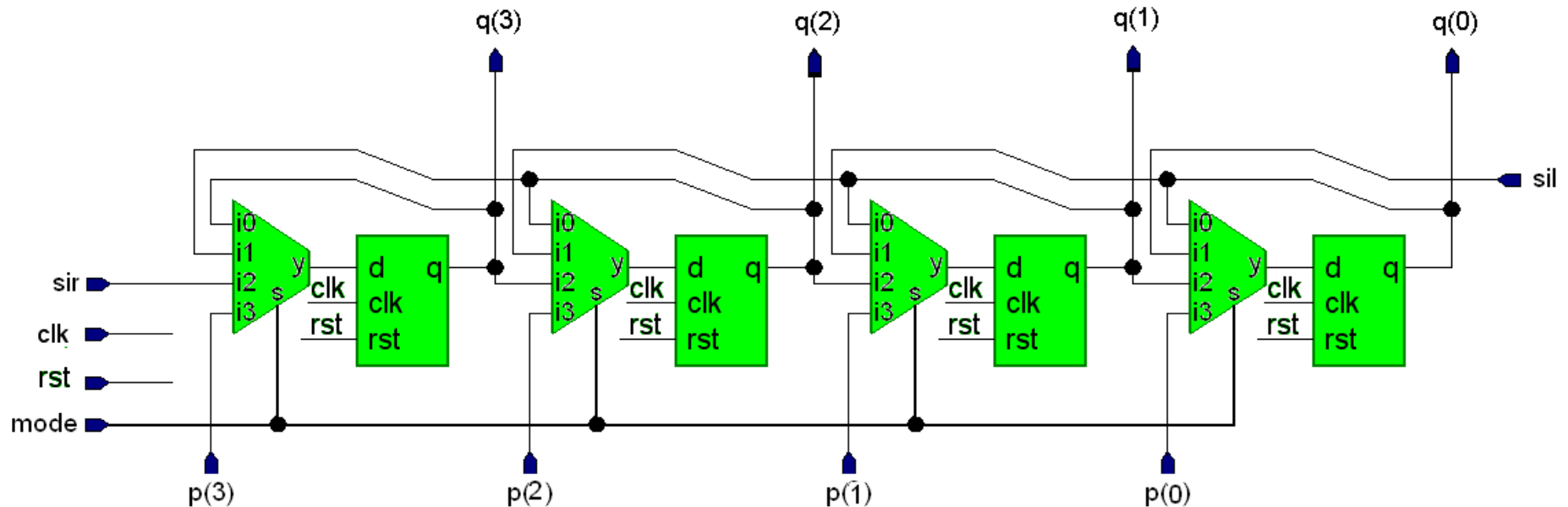
# Schuifregisters

- Hieronder het schuifregister en de data.



# Schuifregisters

- Met twee stuurlijnen zijn echter vier combinaties te maken.
- Het schuifregister is uit te breiden met één extra schuif-mogelijkheid zodat beide kanten kan worden opgeschoven.



Mode = 00 is hold

Mode = 10 is shift right

Mode = 01 is shift left

Mode = 11 is load

# Schuifregisters

- Schuiven heeft ook nog een rekenkundige functie.
- Naar links schuiven en aanvullen met 0 is hetzelfde als vermenigvuldigen met 2.

1011 - het getal (+11 unsigned)

10110 - eerste keer schuiven (\*2)

101100 - tweede keer schuiven (\*4)

1011000 - derde keer schuiven (\*8 = +88 unsigned)

# Schuifregisters

- Schuiven heeft ook nog een rekenkundige functie
- Naar rechts schuiven en aanvullen met 0 is hetzelfde als delen door 2.

1011011 - het getal (+91 unsigned)

0101101 - eerste keer schuiven (/2)

0010110 - tweede keer schuiven (/4)

0001011 - derde keer schuiven (/8 = +11 unsigned)

- De minstwaardige bits gaan wel verloren!



# Schuifregisters

- Signed naar links schuiven gaat ook.

1111011 - het getal (-5 signed)

1110110 - eerste keer schuiven (\*2)

1101100 - tweede keer schuiven (\*4)

1011000 - derde keer schuiven (\*8 = -40 signed)

- Het resultaat moet wel passen!

# Schuifregisters

- Signed naar rechts schuiven gaat ook, maar er is wel een voorziening nodig om het tekenbit te behouden.

1011011 - het getal (-37 signed)

1101101 - eerste keer schuiven (/2)

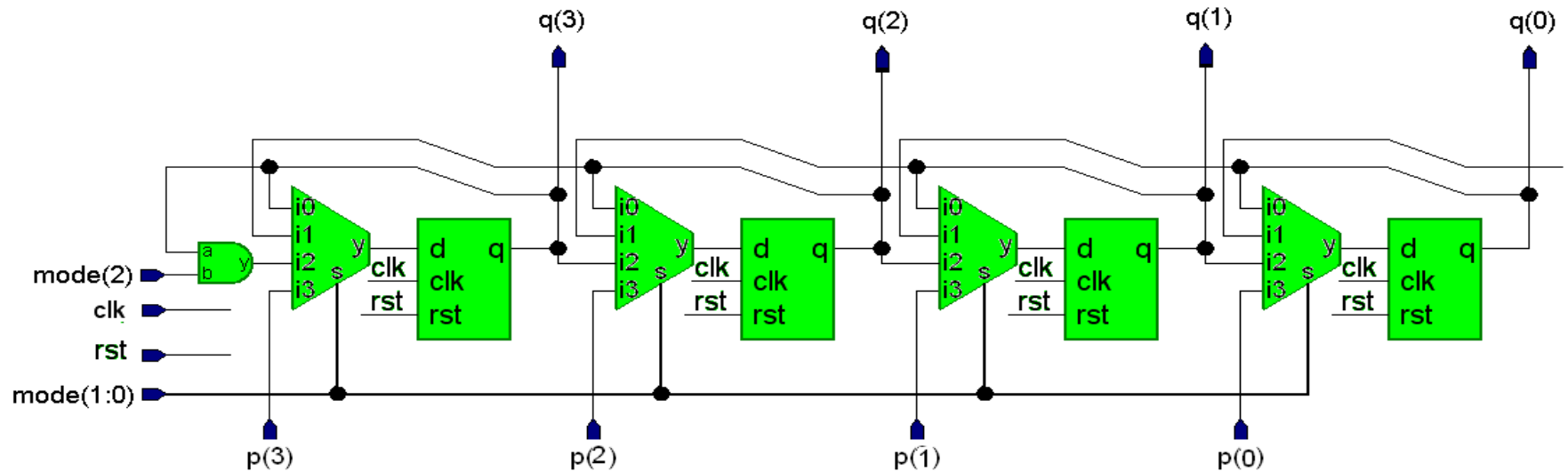
1110110 - tweede keer schuiven (/4)

1111011 - derde keer schuiven (/8 = -5 signed)

- De minstwaardige bits gaan wel verloren!
- Negatieve getallen worden naar beneden (dus nog negatiever) afgerond.

# Schuifregisters

- Hieronder het schema.



- Een extra stuursignaal is nodig om signed/unsigned delen mogelijk te maken.



Academie voor Technology, Innovation &  
Society Delft  
Academie voor ICT & Media

De Haagse Hogeschool, Delft  
+31-15-2606311  
J.E.J.opdenBrouw@hhs.nl  
[www.dehaagsehogeschool.nl](http://www.dehaagsehogeschool.nl)

**DE HAAGSE**  
HOGESCHOOL