



*DIGTEC/2021-2022*

Jesse op den Brouw

**DIGTEC**

Schuifregisters

**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:

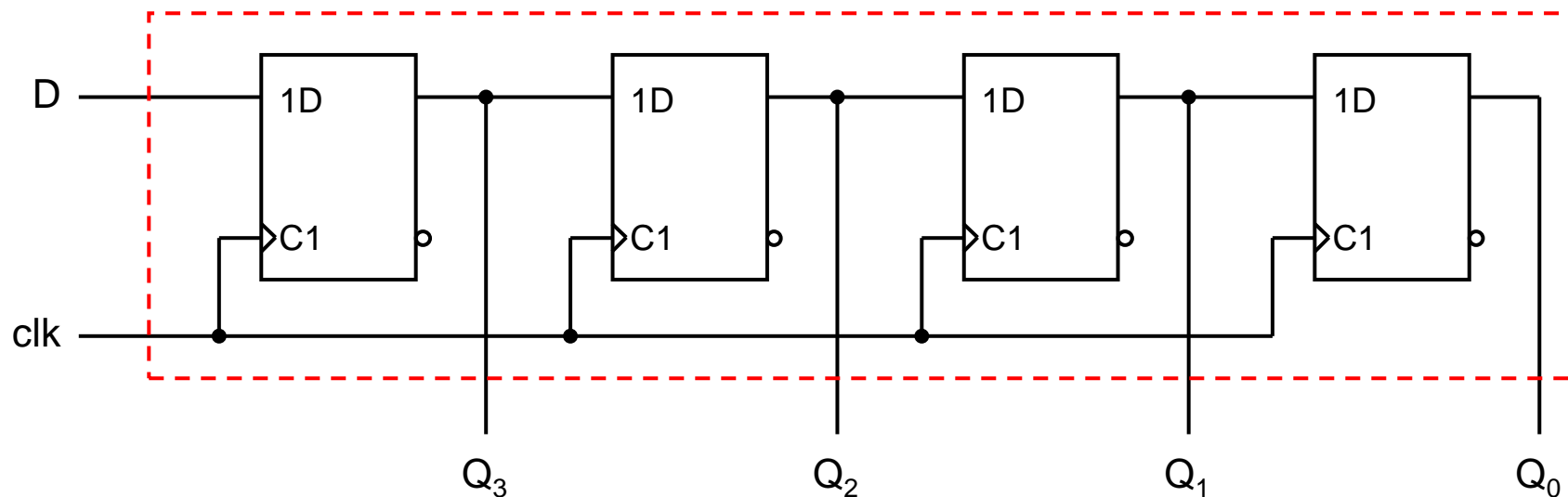
RS232 – minst significante bit eerst

Ethernet – meest significante bit eerst.

- Goed opletten dus.

# Schuifregisters

- Een schuifregister is een groep (D-)flipflops waarvan de data-uitgang van een flipflop is verbonden met de data-ingang van de naastgelegen flipflop. Hieronder een Serial-In-Parallel-Out (SIPO)-schuifregister, *msb* eerst.



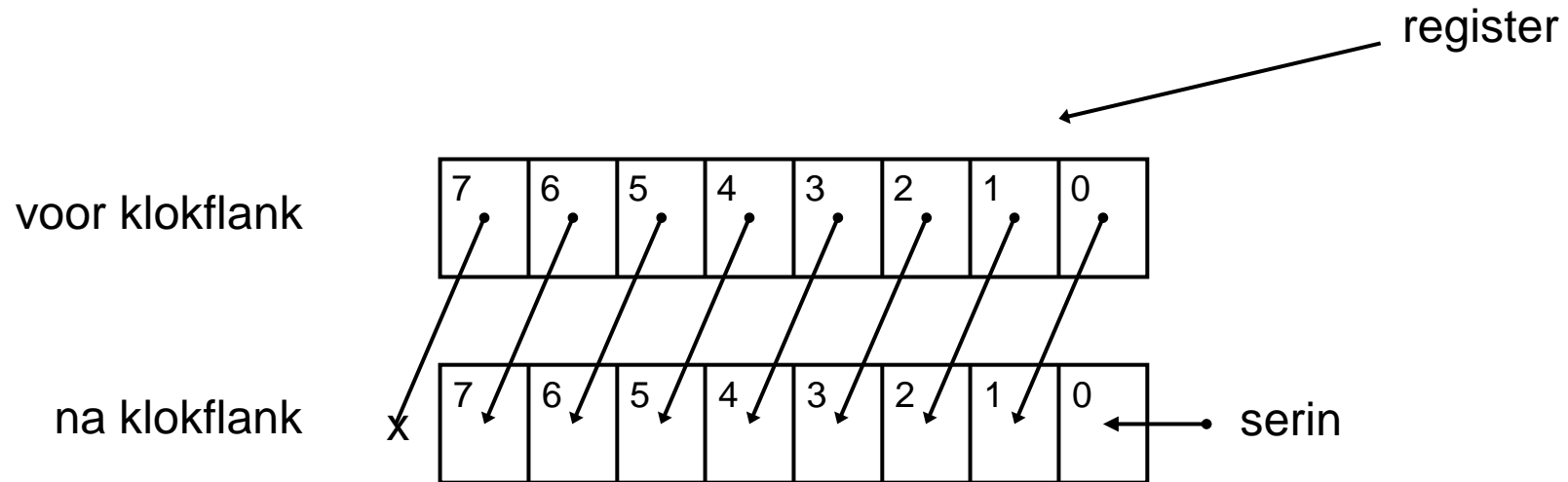
# Schuifregisters

- Bij schuiven praten over *linksom* (left shift) en *rechtsom* (right shift) schuiven.
- 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.



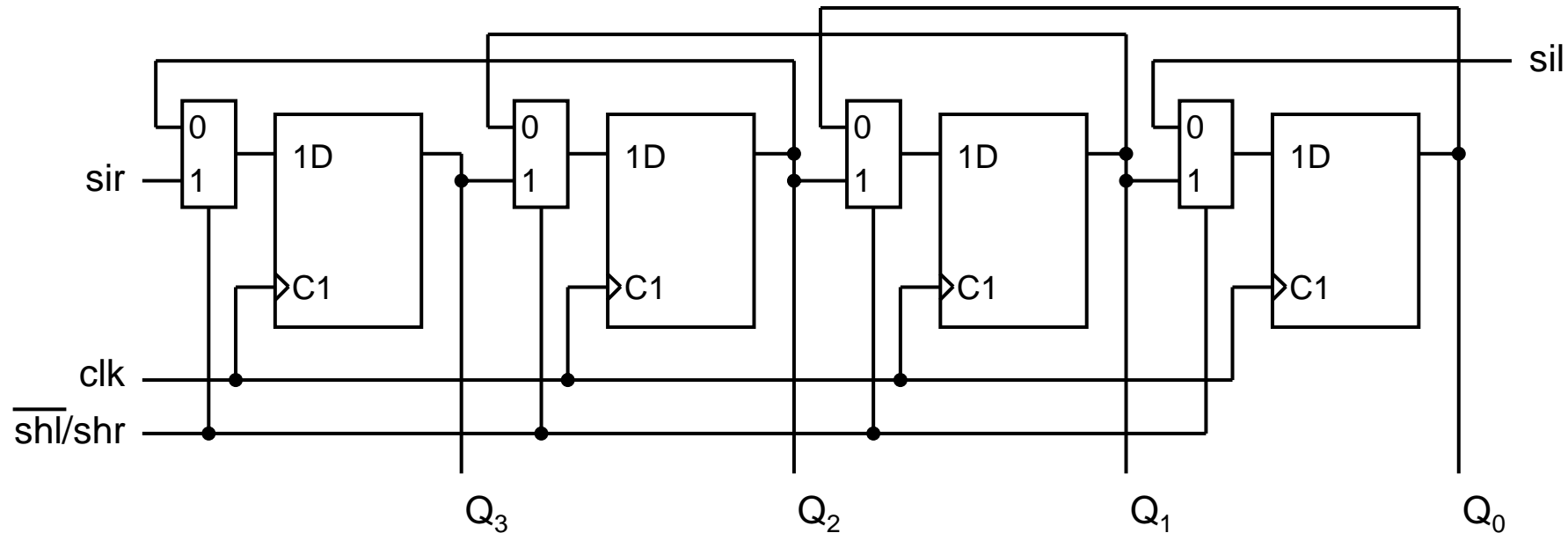
# Schuifregisters

- Een voorbeeld van linksom schuiven. Bits 6 t/m 0 worden één plek naar links geschoven en aangevuld met een extern databit. Bit 7 wordt “eruit geschoven”.



# Schuifregisters

- Daarnaast bestaan er ook *bidirectionele schuifregisters*. Dat zijn schuifregisters die twee kanten kunnen opschuiven.
- De schuifrichting wordt aangegeven met een stuursignaal. Met behulp van multiplexers wordt de juiste ingang geselecteerd.



# Schuifregisters

- We ontwerpen een schuifregister met *right shift*, *hold* en *parallel load* mogelijkheden.
- We gebruiken dit schuifregister voor het verzenden van seriële data.
- Hiervoor zijn nu twee stuursignalen nodig en 3x1 multiplexers.
- Als voorbeeld de zender-schuifregister voor RS232-communicatie.

# Schuifregisters

- RS232-communicatie gaat als volgt\*):

De zendlijn is 1 in rust.

Eerst wordt een *startbit* naar buiten geschoven. Deze is 0.

Daarna worden de 8 databits naar buiten geschoven.

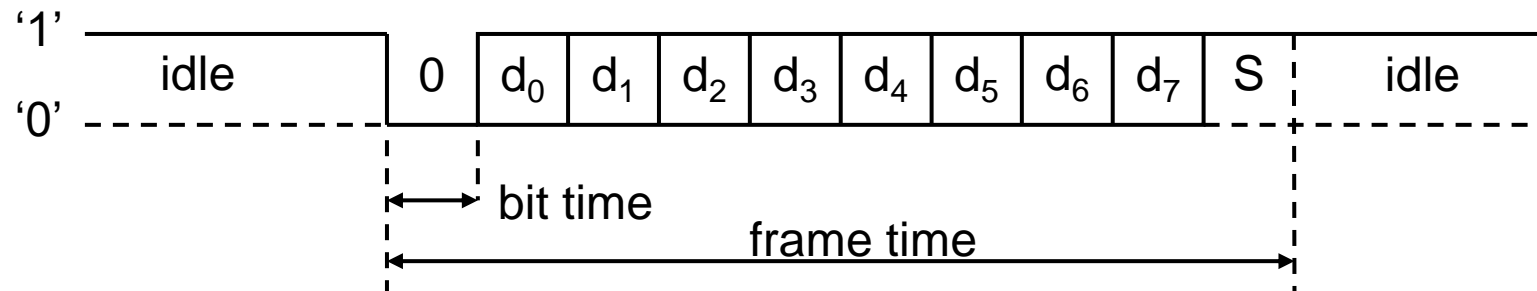
Vervolgens volgt er één stopbit. Deze is 1.

- Dit wordt 8N1 genoemd, er is geen *parity check*.
  - Parity is een één-bits foutdetectie-mogelijkheid, wordt zelden gebruikt.

nb: er zijn ook andere mogelijkheden

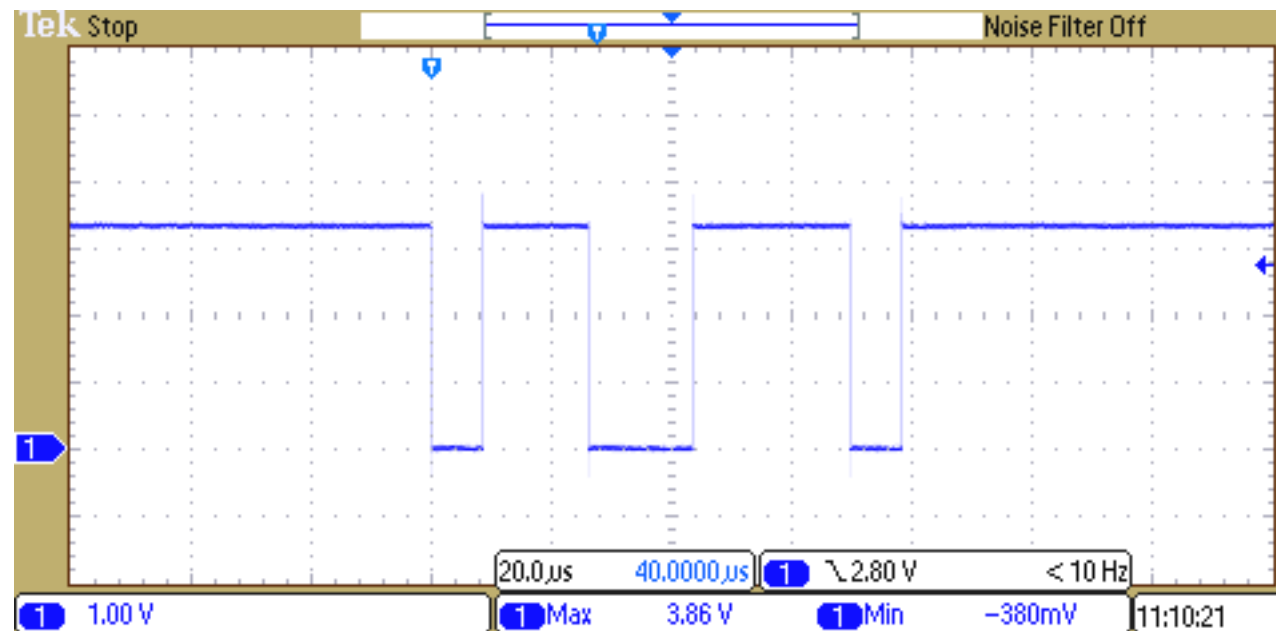
# Schuifregisters

- De seinsnelheid wordt uitgedrukt in bits per seconde (bps) en wordt meestal *baud rate* genoemd.
- Bekende waarden zijn 9600 bps en 115200 bps.
- Hieronder is het frame-formaat te zien.



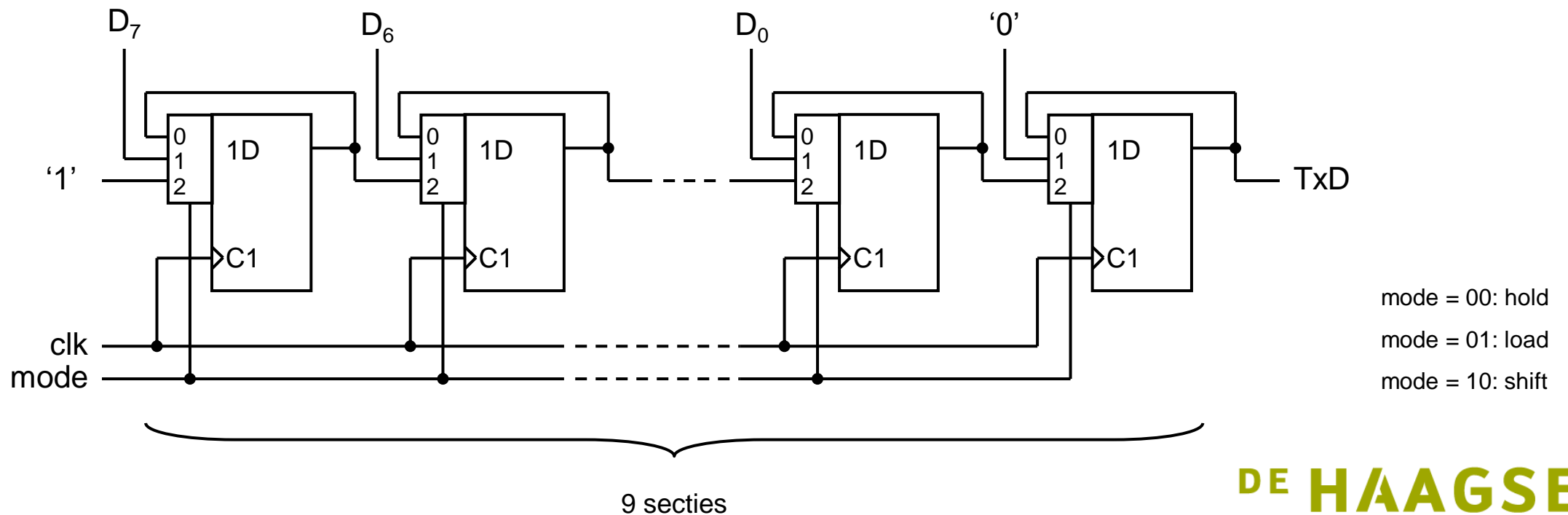
# Schuifregisters

- Hieronder een oscilloscoopbeeld:



# Schuifregisters

- Er worden 9 schuifsecties gemaakt, de eerste wordt geladen met een 0 (startbit), de laatste schuift een 1 in (stopbit).



# Schuifregisters

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

0001011 - het getal (+11 unsigned)

0010110 - eerste keer schuiven (\*2)

0101100 - tweede keer schuiven (\*4)

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

- In een processor: LSL, logical shift left



# 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!
- In een processor: LSR, logical shift right.

# Schuifregisters

- Signed naar links schuiven:

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! Het resultaat mag dan niet van teken veranderen!
- Signed naar links schuiven is hetzelfde als bij unsigned naar links schuiven.

# 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!
- In een processor: ASR, arithmetic shift right.

**let's change**