DIGTEC/2021-2022

Jesse op den Brouw

DIGTEC

Geheugen

DE HAAGSE HOGESCHOOL

Geheugen

- Tot nu toe zijn alleen *combinatorische* schakelingen behandeld.
- Bij deze schakelingen zijn de uitgangen alleen afhankelijk van de ingangen.
- Schakelingen die geheugenwerking hebben worden *sequentiële* schakelingen genoemd.
- De uitgangen van deze schakelingen zijn afhankelijk van de ingangen én de inhoud van het geheugen.
- De inhoud van het geheugen kan weer aangepast worden.

Geheugen

- Er zijn verschillende soorten geheugenelementen:
- Een (direct werkende) SR-latch
- Een gated latch (SR- en D-)
- Flankgevoelige flipflop (D- en JK-)

SR-latch

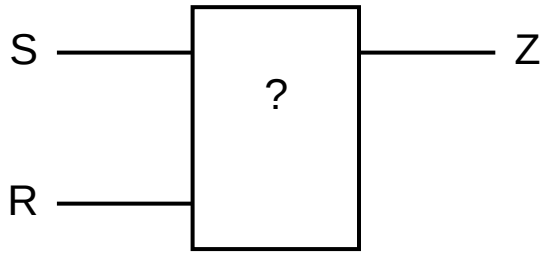
- Een geheugenelement heeft de volgende acties:
- Een actie waarin het geheugenelement gezet (set) wordt.
- Een actie waarin het geheugenelement gewist (reset) wordt.
- Een actie waarin de laatst ingebrachte stand van het geheugenelement onthouden wordt.
- Deze drie acties kunnen met twee (stuur-)bits gecodeerd worden.

SR-latch

- De namen van deze twee signalen zijn S (set) en R (reset), deze namen hebben een directe relatie met de acties.
- De uitgang wordt in de regel Z genoemd, sommige boeken gebruiken Q.
- Het geheugenelement is te tekenen als een black box.
- Van dit te ontwerpen geheugenelement kan een functietabel worden opgesteld.

SR-latch

- Hieronder de black box en de functietabel.

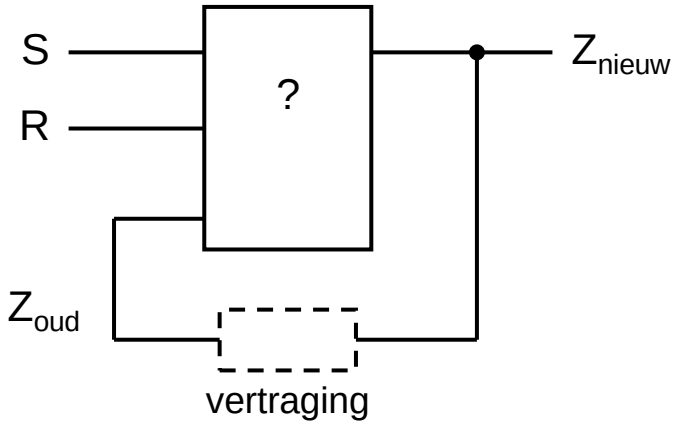


S	R	functie
0	0	onthouden
0	1	reset
1	0	set
1	1	niet gebruikt

- NB: voor één van de acties moet altijd een *combinatie* van S en R worden aangeboden.
- NB: de combinatie SR = 11 wordt (voorlopig) niet gebruikt.

SR-latch

- We kunnen de schakeling als volgt modelleren:



- De vertraging zorgt ervoor dat de waarde van Z_{oud} nog even beschikbaar blijft, als de nieuwe waarde op Z_{nieuw} gegenereerd wordt.

DE HAAGSE HOGESCHOOL

SR-latch

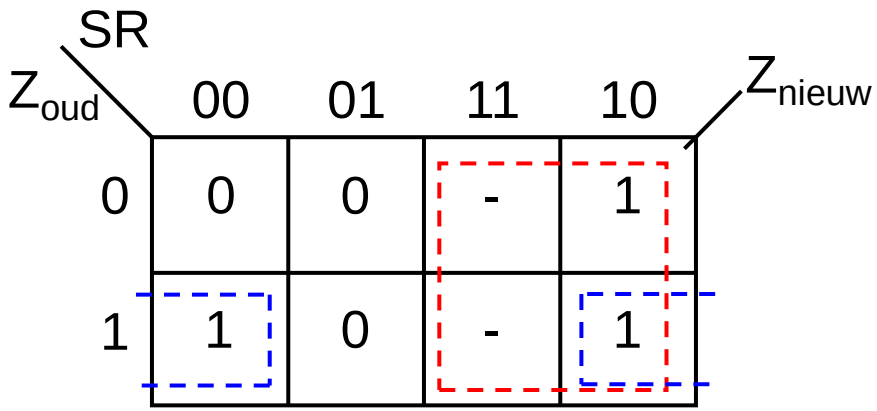
- Rechts is de waarheidstabel.
- Bij SR = 00 moet de nieuwe waarde de oude waarde volgen.
- Bij SR = 01 moet de nieuwe waarde op 0 worden gezet.
- Bij SR = 10 moet de nieuwe waarde op 1 worden gezet.
- Bij SR = 11 wordt de nieuwe waarde als don’t care gespecificeerd.

S	R	Z_{oud}	Z_{nieuw}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

SR-latch

- Van deze waarheidstabel kan een Karnaughdiagram worden opgesteld:
- De bijbehorende functie is:

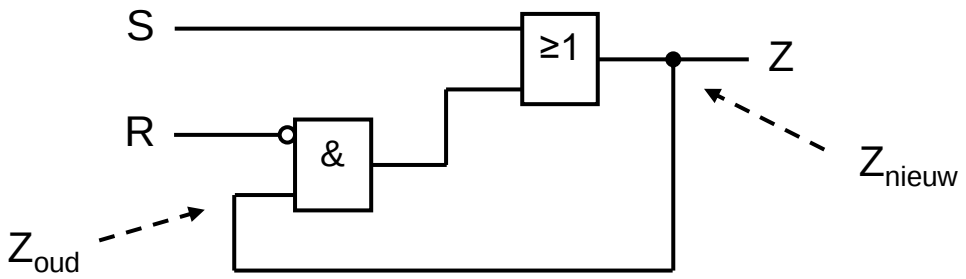
$$Z_{nieuw} = S + \overline{R} \cdot Z_{oud}$$



- De functie laat duidelijk zien dat Z_{nieuw} een functie is van Z_{oud}.
- De ingangscombinatie SR = 11 is na uitwerking gebruikt als set-operatie. Het geheugenelement heeft een *overheersende set*.

SR-latch met overheersende set

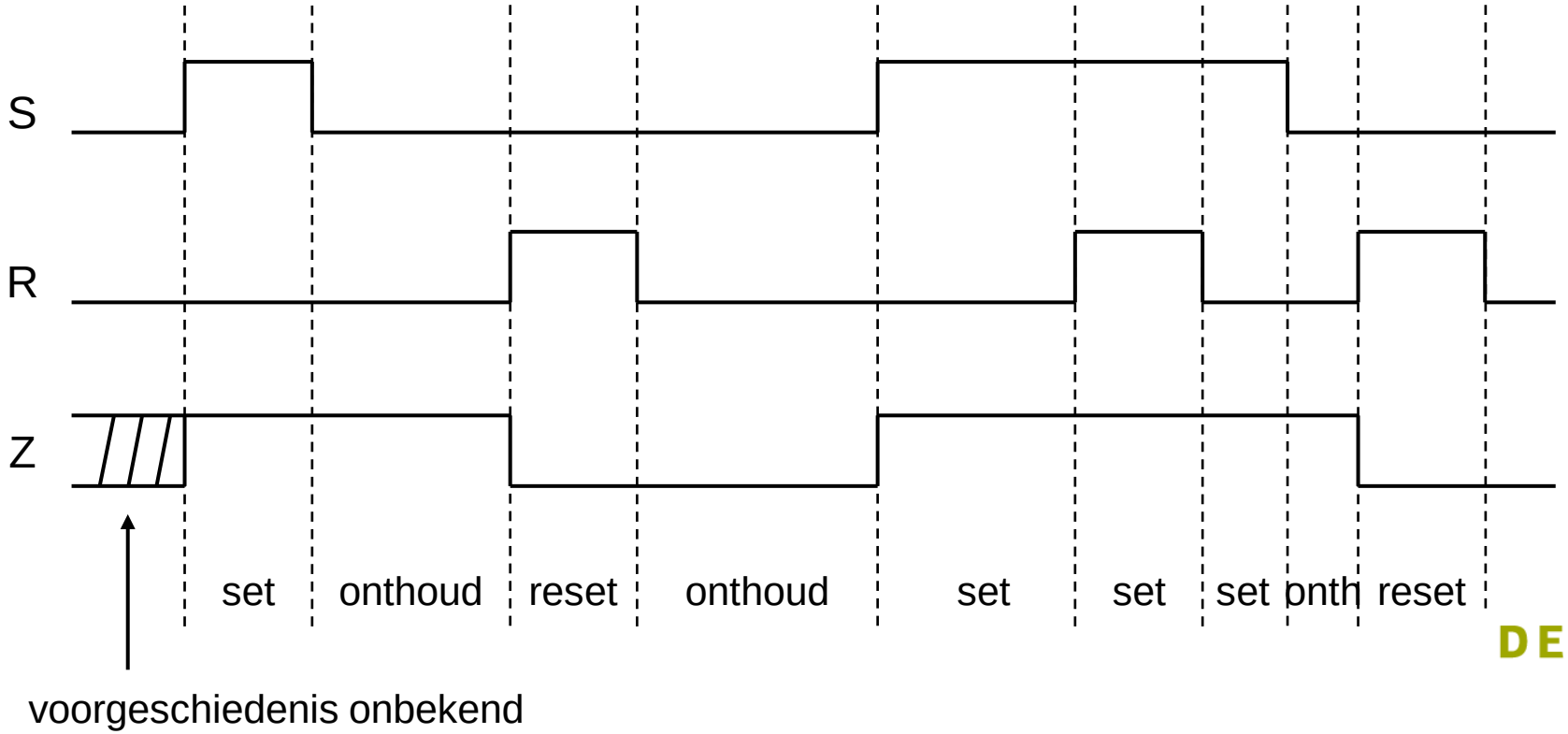
- Nu kan het schema getekend worden. Merk op dat de uitgang Z wordt teruggekoppeld als een ingang.



- Kenmerkend voor deze *SR-latch* is de *lus* in de schakeling.
- De poorten zorgen voor vertraging.

SR-latch met overheersende set

- In het *signaaldiagram* is het gedrag van het SR-latch goed weer te geven.



DE HAAGSE HOGESCHOOL

SR-latch met overheersende set

De functie voor het SR-latch kan worden omgewerkt tot een NAND-NAND-vorm:

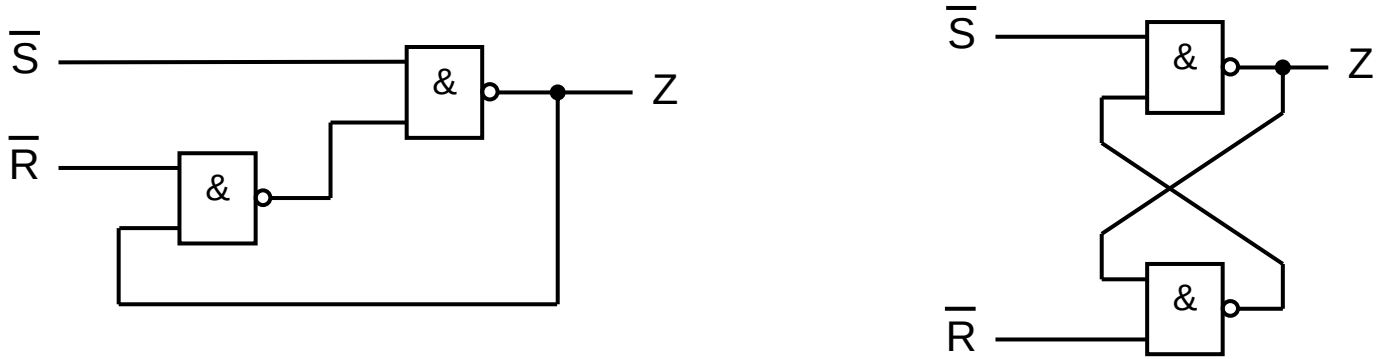
$$Z_{nieuw} = S + \overline{R} \cdot Z_{oud} = \overline{\overline{S} \cdot (\overline{\overline{R} \cdot Z_{oud}})}$$

De schakeling is nu te bouwen met 2 NANDs.

Voor de stuursignalen zijn wel beide inversen nodig.

SR-latch met overheersende set

- De schakeling is als volgt te tekenen.

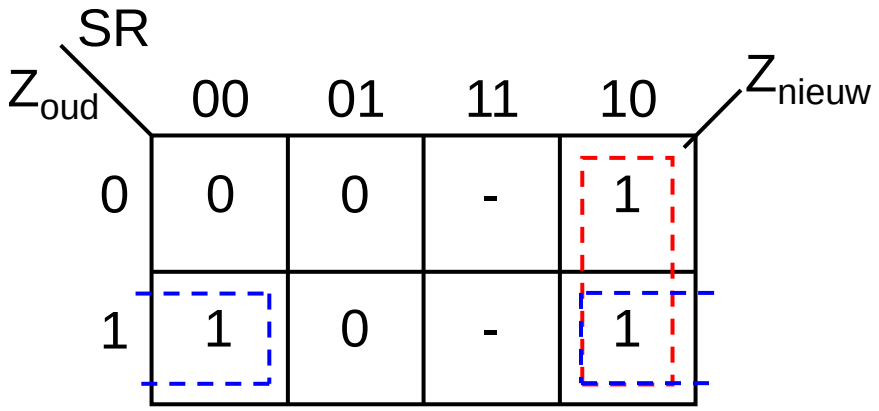


- De tweede variant wordt doorgaans gebruikt in de literatuur.

SR-latch met overheersende reset

- Het Karnaughdiagram van het SR-latch wordt nog eens opnieuw uitgewerkt:
- De bijbehorende functie is nu:

$$Z_{nieuw} = \overline{R} \cdot S + \overline{R} \cdot Z_{oud} = \overline{R} \cdot (S + Z_{oud})$$

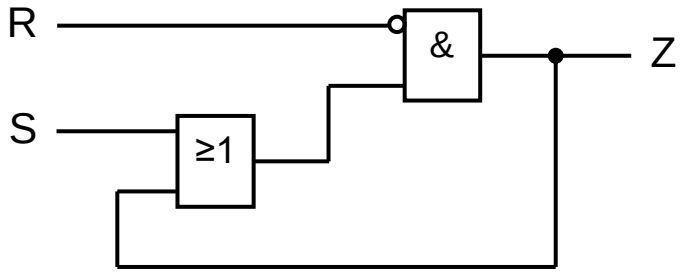


- De ingangscombinatie SR = 11 is na uitwerking gebruikt als resetoperatie. Het geheugenelement heeft een *overheersende reset*.

DE HAAGSE HOGESCHOOL

SR-latch met overheersende reset

- Nu kan het schema getekend worden.



- In feite zijn de poorten van plaats verwisseld. Let op dat R nu boven staat en S onder.

SR-latch met overheersende reset

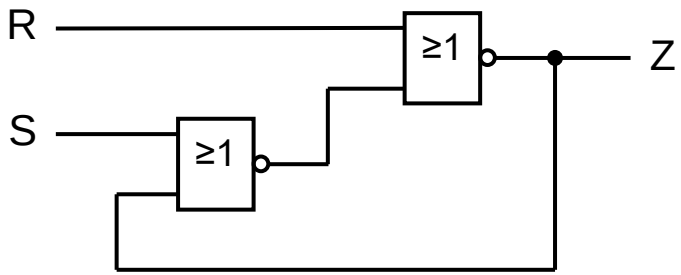
- De functie voor de SR-latch kan worden omgewerkt tot een NOR-NOR-vorm:

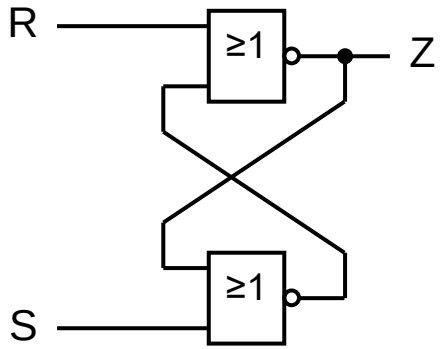
$$Z_{nieuw} = \overline{R} \cdot (S + Z_{oud}) = \overline{R + (\overline{S + Z_{oud}})}$$

- De schakeling is nu te bouwen met 2 NORs.
- Voor de stuursignalen zijn *geen* inversen nodig.

SR-latch met overheersende reset

- De schakeling is als volgt te tekenen.





- De tweede variant wordt doorgaans gebruikt in de literatuur, maar soms zó getekend dat S weer boven staat.

SR = 11

- In veel boeken wordt de combinatie SR = 11 de *verboden stand* genoemd.
- Deze combinatie is dan wel niet verboden, maar wordt in de praktijk nauwelijks gebruikt.
- Daarnaast levert het problemen op als van SR = 11 naar SR = 00 gegaan wordt. Dit is in de praktijk nooit goed te realiseren. Eén van de twee stuursignalen verandert het eerst, zodat onbedoeld een set- of resetoperatie gestart wordt.

Gated SR-latch

- SR-latches zijn transparant voor hun ingangssignalen.
- Eventuele veranderingen van de ingangen heeft direct een gevolg voor de waarde van de uitgang.
- Meestal worden de signalen voor S en R door combinatorische schakelingen opgewekt.
- Deze schakelingen kunnen *hazards** op de uitgangen vertonen, waardoor onbedoelde combinaties voor S en R worden opgewekt.

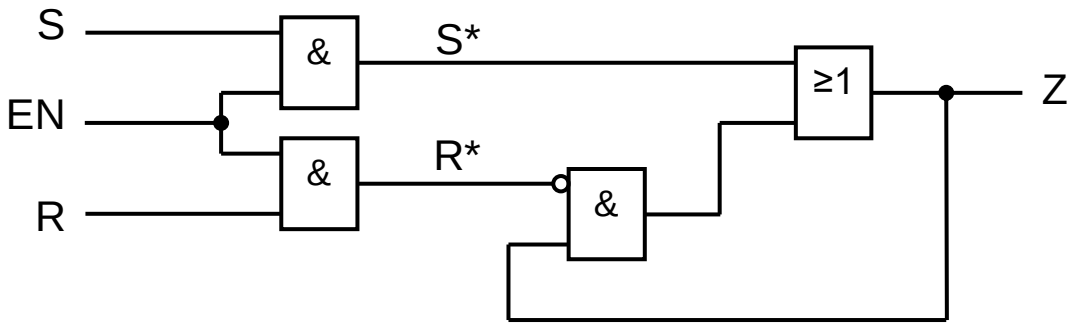
* Onbedoelde tijdelijke uitgangsverandering als gevolg van ingangsveranderingen en verschillen in looptijden in een combinatorische schakeling.

Gated SR-latch

- Beter is het om de ingangen van een SR-latch ongevoelig te maken voor deze verandering.
- De SR-latch wordt uitgebreid met een enable-signaal EN en wordt een gated SR-latch genoemd.
- Als het enable-signaal 0 is, houdt de latch de huidige waarde vast (onthouden).
- Als het enable-signaal 1 is, heeft de latch de werking van een gewone SR-latch.

Gated SR-latch

- De schakeling ziet er dan als volgt uit.



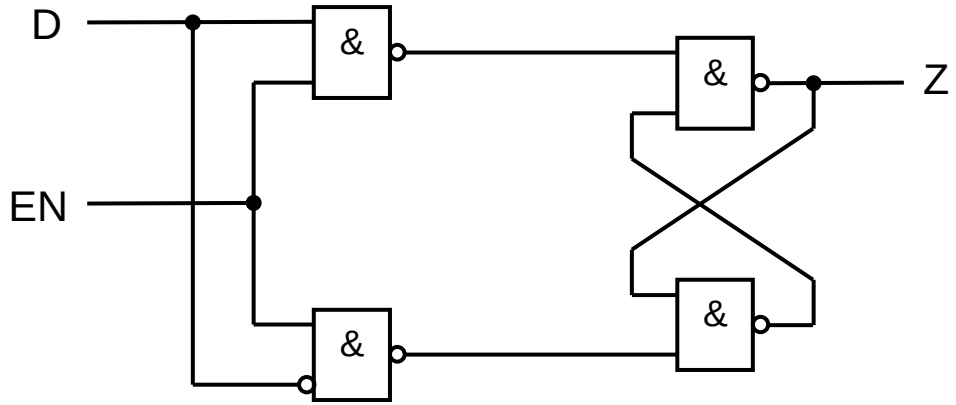
- Tijdens EN = 0 is de latch niet meer transparant.
- Tijdens EN = 1 heeft de latch de werking van een SR-latch.

Gated D-latch

- De nu ontwikkelde latch heeft eigenlijk vijf onthoudstanden: SR = 00 tijdens EN = 1 en EN = 0 (waarbij SR = don’t care).
- De combinatie SR = 00 kan dus geschrapt worden, want die wordt overgenomen door EN = 0.
- De combinatie SR = 11 wordt toch niet gebruikt en kan dus ook geschrapt worden.
- Er blijven twee combinaties over: SR = 01 voor reset en SR = 10 voor set (tijdens EN = 1).
- Nu blijkt dat S en R complementair zijn.

Gated D-latch

- De waarde van R is te verkrijgen door de inverse van S te nemen. De namen S en R verdwijnen (er is immers nog maar één ingang over) en wordt nu D.



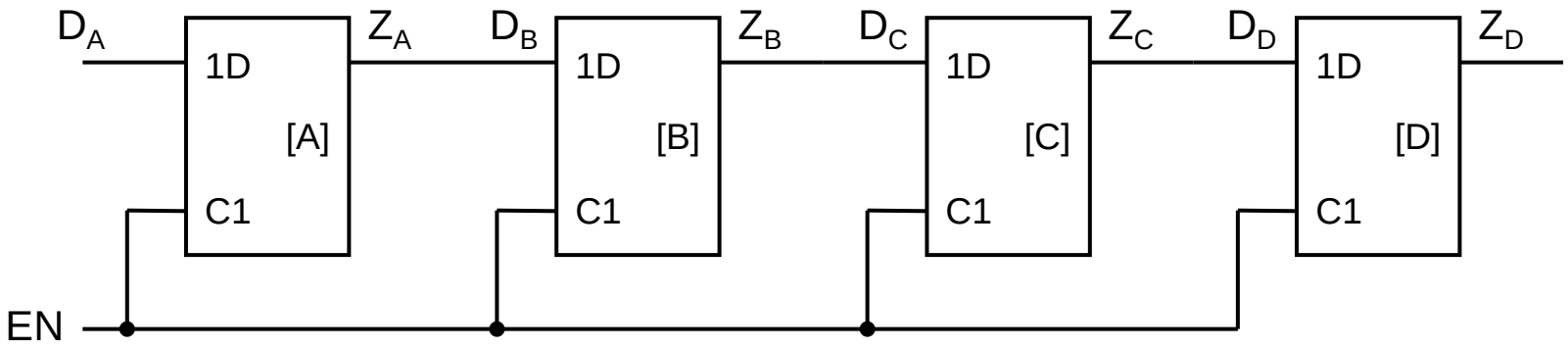
- De hier gepresenteerde schakeling (met NANDs) heet *gated D-latch*.

D-flipflop

- De gated D-latch heeft als nadeel dat D stabiel moet worden gehouden zolang EN = 1.
- Op basis van deze latch is het niet makkelijk om data van één latch in een andere over te brengen.
- Een veel voorkomende schakeling is een *schuifregister*. Hierbij worden een aantal identieke geheugenelement zó geschakeld dat de data geschoven kan worden onder besturing van een enablesignaal.

D-flipflop

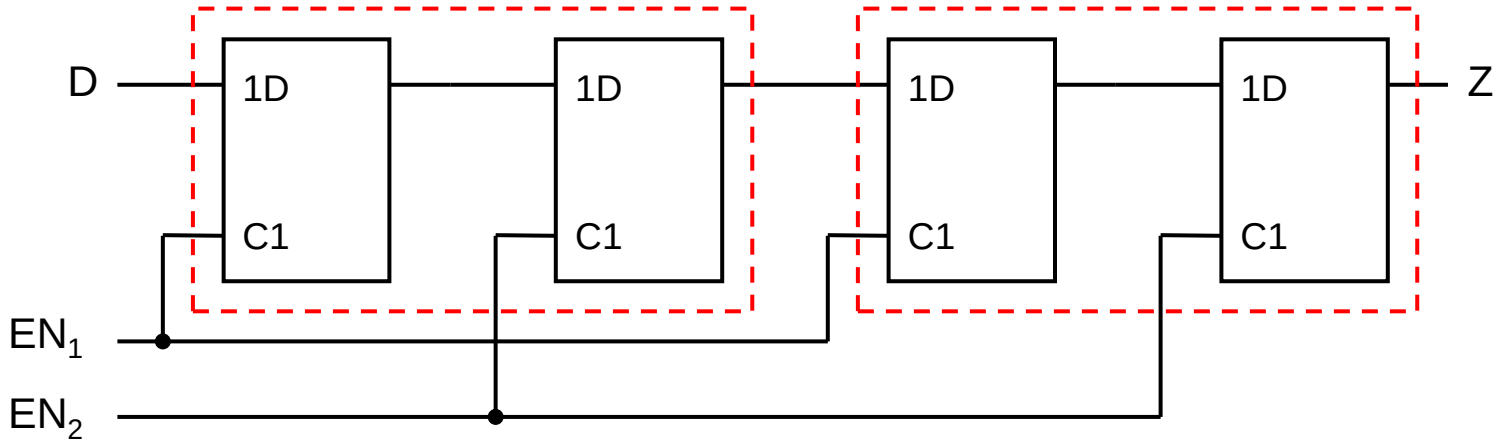
- Onderstaande is een poging tot een schuifregister.



- Dit gaat fout als EN van 0 naar 1 gaat. *Alle* latches zijn dan *transparant* en data op D_A wordt direct naar Z_D doorgevoerd.

D-flipflop

- Dit transparant zijn kan worden tegengehouden door twee latches afwisselend te plaatsen, elk met een eigen EN. Er zijn dan ook twee EN-signalen nodig.



- Voorwaarde voor juiste werking: EN1 en EN2 mogen niet tegelijk 1 zijn.

D-flipflop

- Als EN1 = 0 en EN2 = 0 onthouden alle latches. Geen enkele latch is transparant.
- Als EN1 = 1 en EN2 = 0 zal de eerste latch data overnemen van ingang D en de derde latch data overnemen van de tweede latch. De eerste en derde latch zijn transparant. De inhoud van de tweede en vierde latch zijn ongewijzigd.
- Als EN1 = 0 en EN2 = 1 zal de tweede latch data overnemen van de eerste latch en de vierde latch data overnemen van de derde latch. De inhoud van de eerste en derde latch zijn ongewijzigd.
- Voor het maken van een 4-bit schuifregister zijn dus acht latches nodig.

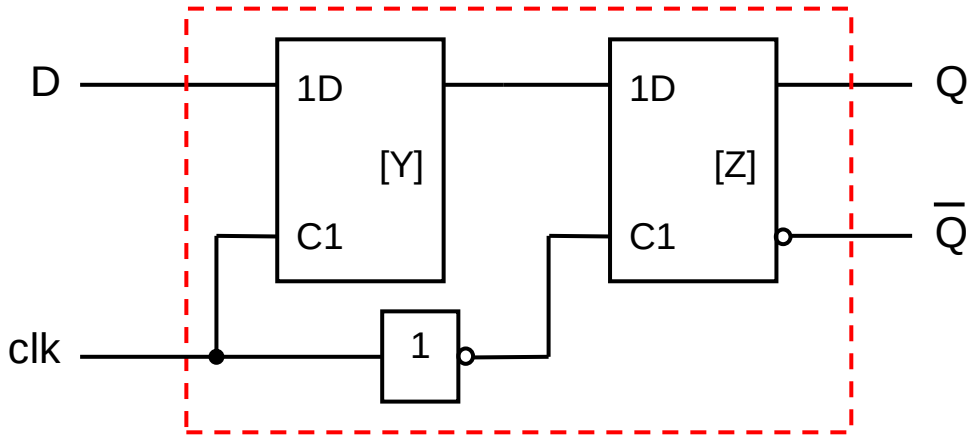
D-flipflop

- Een geheugenelement op basis van twee latches die niet tegelijk transparant zijn wordt een *flipflop* genoemd, in dit geval een D-flipflop.
- Door de constructie van de flipflop met twee latches wordt dit meester-en-slaaf (*master-slave*) genoemd*).
- De schakeling kan nog wat eenvoudiger worden door het gebruik van een NOT-poort zodat $EN_2 = \overline{EN_1}$

nb: maar welke is nou de master en welke de slave?

D-flipflop

- Hieronder de gerealiseerde D-flipflop. De meeste (losse) flipflops hebben ook een inverse uitgangswaarde beschikbaar.



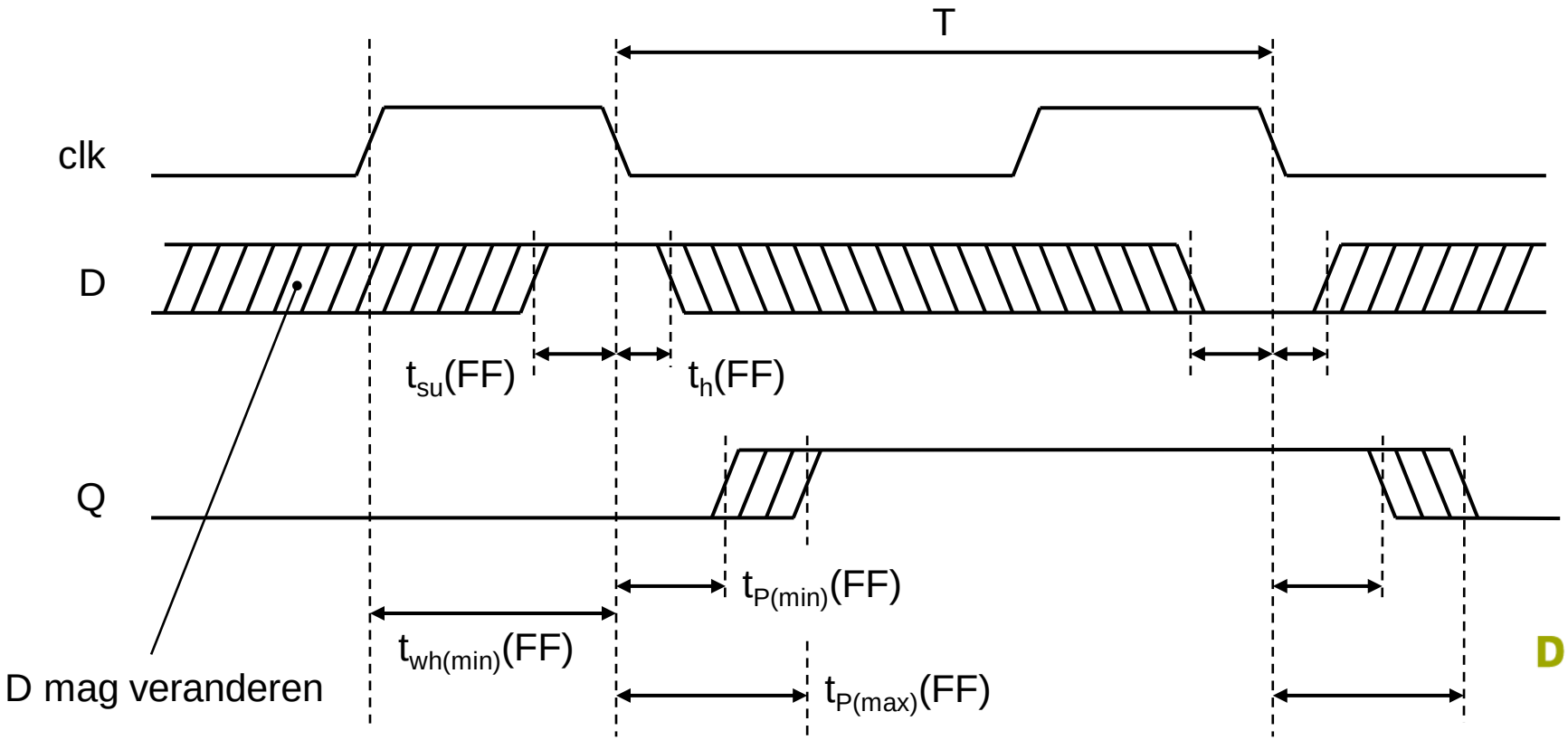
DE HAAGSE HOGESCHOOL

D-flipflop

- De D-flipflop neemt de data aan de ingang over op de *neergaande flank*.
- Het timing-voorschrift wordt *flankgestuurde* of *edge-triggered timing* genoemd.
- Bij een edge-triggered timing refereren alle timing-parameters aan dezelfde flank van het kloksignaal. Dit wordt de actieve flank genoemd.
- Het tijdinterval waarin de data klaar moet staan is in de regel zeer klein, in de orde van enkele ns.

D-flipflop

- Hieronder de timing van de negative edge-triggered D-flipflop.



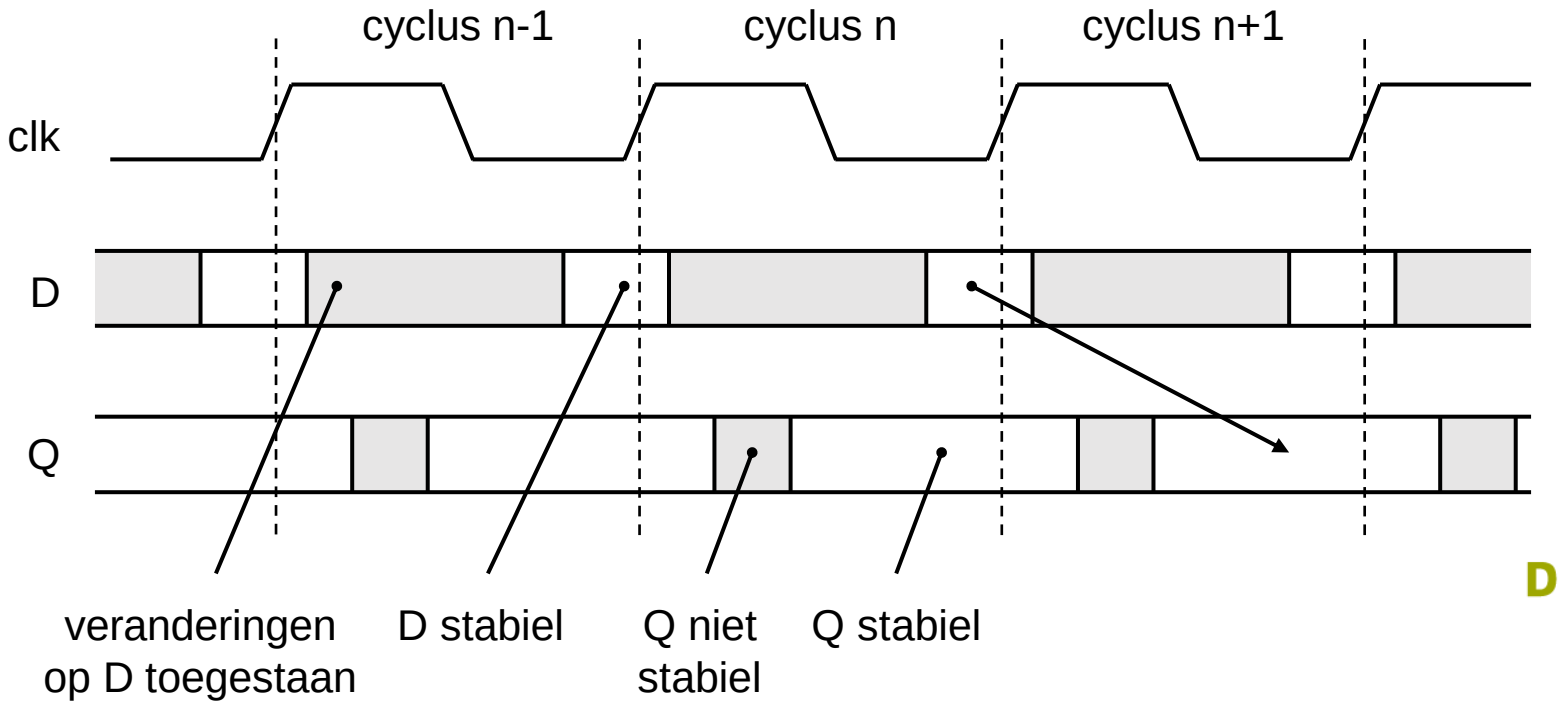
DE HAAGSE HOGESCHOOL

D-flipflop

$t_{P(min)}(FF)$	De minimale vertragingstijd van de uitgang van de flipflop t.o.v. de actieve klokflank.
$t_{P(max)}(FF)$	De maximale vertragingstijd van de uitgang van de flipflop t.o.v. de actieve klokflank.
$t_{su}(FF)$	De setuptijd van de D-ingang van de flipflop t.o.v. de actieve klokflank.
$t_h(FF)$	De holdtijd van de D-ingang van de flipflop t.o.v. de actieve klokflank.
$t_{wh(min)}(FF)$	De minimale pulsbreedteduur van het kloksignaal.
T	De periodeduur van het kloksignaal.

D-flipflop

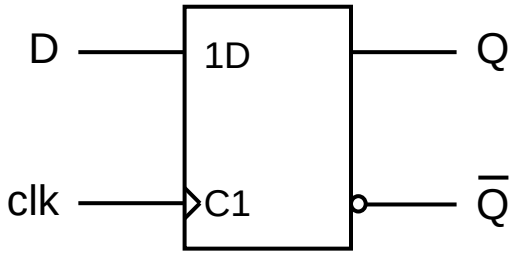
- In de functie van de D-flipflop wordt het kloksignaal *clk* niet meer expliciet meegenomen. Om aan te geven dat een huidige waarde van D pas ná de klokflank op Q beschikbaar is, wordt gebruik gemaakt van de volgende schrijfwijze: $Q^{n+1} = D^{n}$



DE HAAGSE HOGESCHOOL

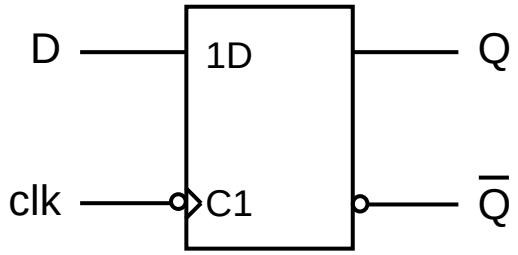
D-flipflop

- Hieronder de symbolen van de D-flipflop.



positive edge-triggered
D-flipflop

werking D afhankelijk
van de opgaande flank
van C



negative edge-triggered
D-flipflop

werking D afhankelijk
van de neergaande flank
van C

D-flipflop

- Het ontwerpen van een betrouwbare flipflop is een lastige aangelegenheid en moet gedaan worden door ervaren ontwerpers.
- Het zelf ontwerpen van een flipflop met losse poorten of in een beschrijvingstaal als VHDL is niet aan te raden.
- Bij gebruik van *reconfigureerbare logica* (FPGA, CPLD) is het helemaal niet gewenst, de fabrikant heeft al kant-en-klare flipflops aangebracht. Door gebruik van *library modules* kan een flipflop gerealiseerd worden.

D-flipflop

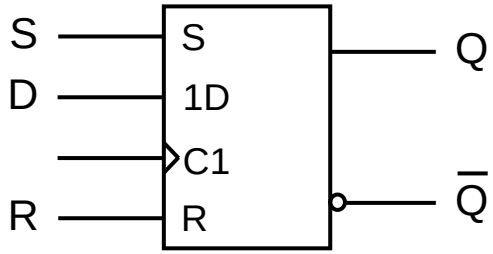
- Voor de logische werking van een schakeling maakt het niet uit op welke flank een flipflop z'n data inklokt.
- Het door elkaar gebruiken van positive en negative edge-triggered flipflops is meestal niet de bedoeling.
- Het gebruik van flipflops met verschillende timing-parameters levert problemen op.
- Binnen één IC is dat echter niet het geval.

D-flipflop

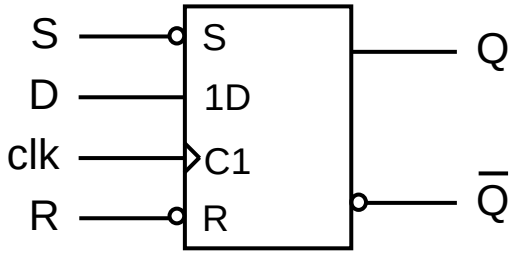
- Bij het opkomen van de voedingsspanning is het onduidelijk wat de stand van een flipflop wordt.
- Vandaar dat flipflops worden voorzien van twee *asynchrone* ingangen die de stand onafhankelijk van de klok vastleggen.
- De *reset*-ingang zorgt ervoor dat de stand van de flipflop logisch 0 wordt.
- De *set*-ingang zorgt ervoor dat de stand van de flipflop logisch 1 wordt.

D-flipflop

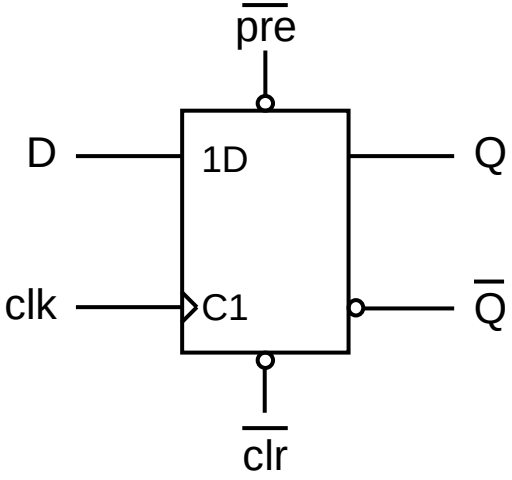
- Hieronder de symbolen van de D-flipflop met asynchrone reset en set.



positive edge-triggered D-flipflop with asynchronous active high set and reset



positive edge-triggered D-flipflop with asynchronous active low set and reset



positive edge-triggered D-flipflop met asynchrone active low preset en clear

DE HAAGSE HOGESCHOOL

D-flipflop

- Soms moet de stand van een flipflop behouden blijven “over de klokflanken heen”, dus de stand verandert niet ondanks dat er een actieve klokflank passeert.
- Dat kan op twee manieren:

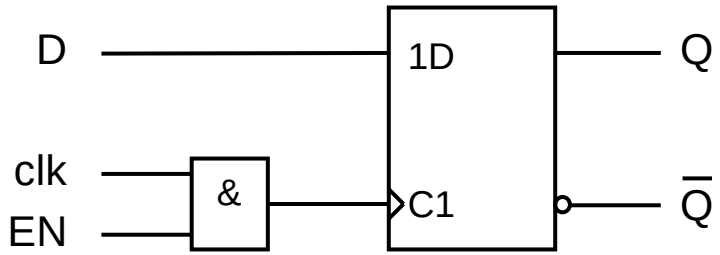
    Schakel de klok uit zodat geen klokflank passeert.

    Zorg ervoor dat de flipflop zijn eigen stand opnieuw inklokt.

- Dit kan beide met een enable-signaal.

D-flipflop

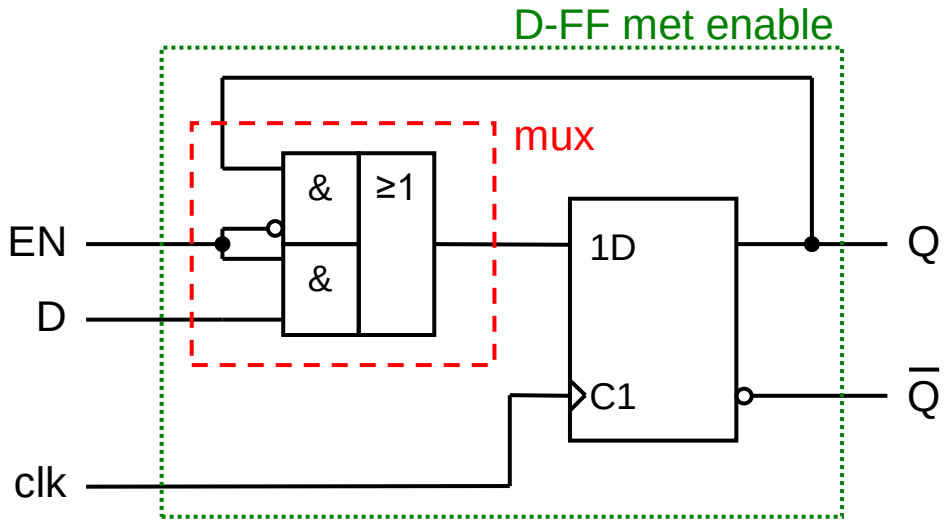
- Oplossing: schakel de klok uit.



- Het enable-signaal mag niet veranderen tijdens clk = 1!
- Het resultaat is een flipflop met pulse-triggered eigenschappen.
- **Doe dit niet, lastige timingeigenschappen!**

D-flipflop

- Oplossing: flipflop klokt zijn eigen waarde óf een nieuwe waarde in door middel van een multiplexer.

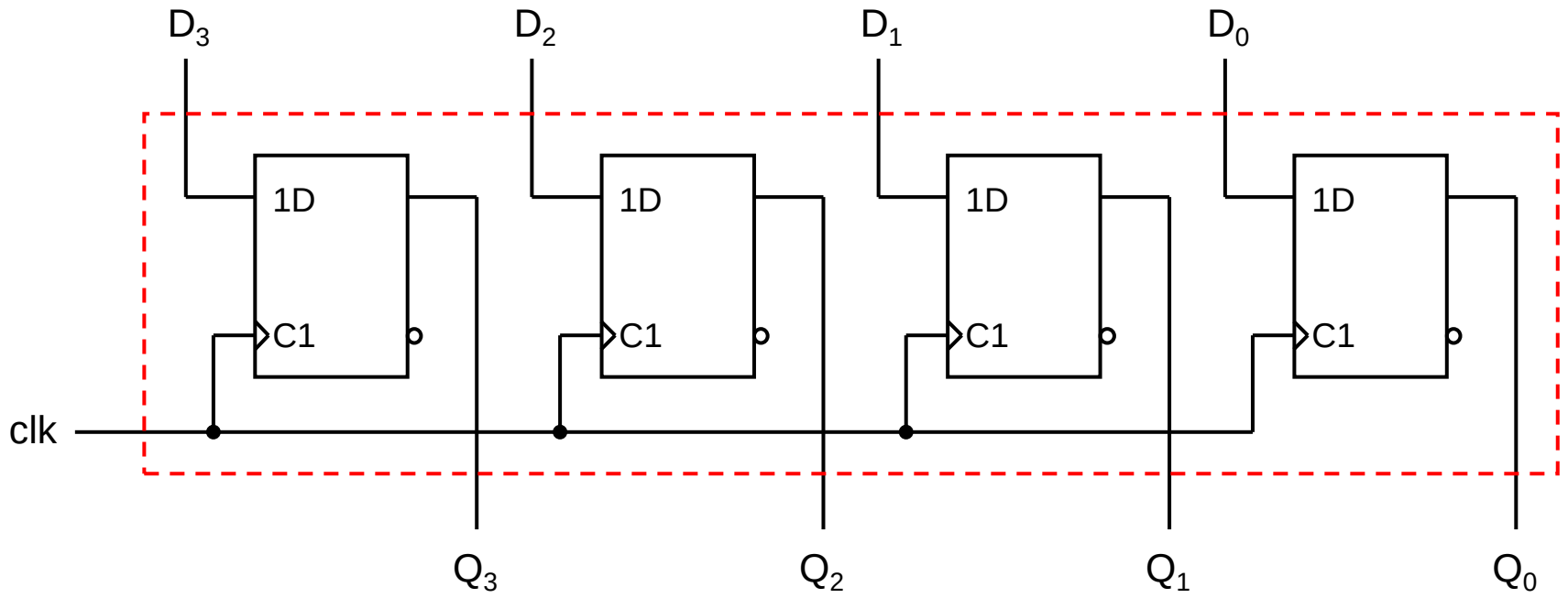


$$Q^{n+1} = \overline{EN^n} \cdot Q^n + EN^n \cdot D^n$$

- Er is wel meer logica nodig dan in het vorige ontwerp, maar het resultaat is een edge-triggered flipflop.

D-flipflop

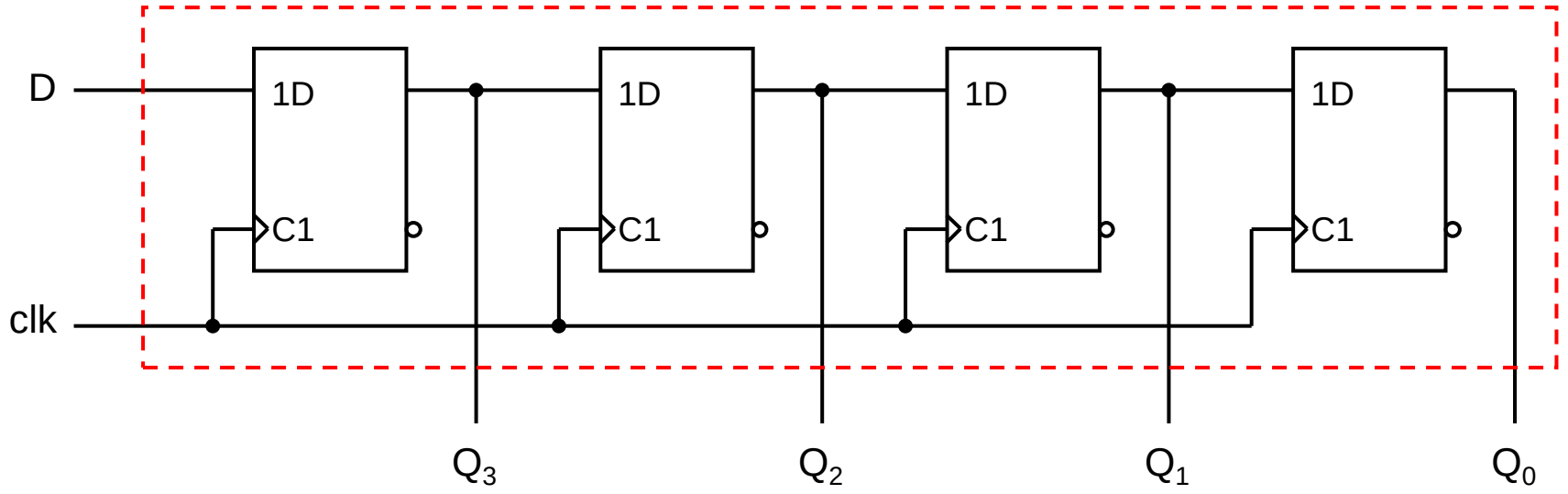
- Een *register* is een groep (D-)flipflops die alle getriggerd worden op hetzelfde kloksignaal, vaak met enable-faciliteit.



DE HAAGSE HOGESCHOOL

D-flipflop

- Een schuifregister is een groep (D-)flipflops waarvan de data-uitgang van een flipflop is verbonden met de data-ingang van de naastgelegen flipflop. Schuifregisters spelen een belangrijke rol in seriële communicatie.



let’s change