DIGTEC/2021-2022

Jesse op den Brouw

DIGTEC

Herhaling week 1 t/m 7

DE HAAGSE HOGESCHOOL

Binair talstelsel

- Het omrekenen van een binair getal naar het decimale equivalent gaat eenvoudig.

exponent = positie cijfer

$$
\begin{aligned}
1101_2 &= 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 \\
&= 8 \quad + 4 \quad + 0 \quad + 1 \\
&= 13_{10}
\end{aligned}
$$

grondtal

$$
\begin{aligned}
11011010_2 &= 1\cdot 2^7 + 1\cdot 2^6 + 1\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^1 \\
&= 128 + 64 + 16 + 8 + 2 \\
&= 218_{10}
\end{aligned}
$$

DE HAAGSE HOGESCHOOL

Decimaal naar binair

- Als voorbeeld wordt 53_{10} omgezet:

53 ÷ 2	=	26 rest 1	minst significante cijfer
26 ÷ 2	=	13 rest 0	
13 ÷ 2	=	6 rest 1	uitlezen
6 ÷ 2	=	3 rest 0	
3 ÷ 2	=	1 rest 1	
1 ÷ 2	=	0 rest 1	meest significante cijfer
<u>0</u>			

- Het binaire equivalent is dan 110101_2

DE HAAGSE HOGESCHOOL

Hexadecimaal

- Een voorbeeld van een hexadecimaal getal:

$$3FA5_{16}$$

- Dit kan worden omgezet naar een decimaal getal:

$$3FA5_{16} = 3 \cdot 16^3 + 15 \cdot 16^2 + 10 \cdot 16^1 + 5 \cdot 16^0 = 16293_{10}$$

- Merk op: hetzelfde systeem als decimaal en binair, nu echter met grondtal 16.

Decimaal naar hexadecimaal

- Omzetten van decimaal naar hexadecimaal gaat op eenzelfde wijze als omzetten van decimaal naar binair, alleen moet er nu gedeeld worden door 16:

40796 ÷ 16	=	2549 rest 12	→	C
2549 ÷ 16	=	159 rest 5		
159 ÷ 16	=	9 rest 15	→	F
9 ÷ 16	=	0 rest 9		
0				

uitlezen

- Het hexadecimale equivalent is dus $9F5C_{16}$

DE HAAGSE HOGESCHOOL

Voorbeeld met twee variabelen

- Rechts is een waarheidstabel van een functie gegeven.
- (Dit is natuurlijk de functie van een OR).
- Het invullen gaat als volgt. Begin bij f_0 en werk door naar f_3. Het resultaat is een z- of n-patroon.

a	b	s
0	0	0
0	1	1
1	0	1
1	1	1

b \ a	0	1
0	0	1
1	1	1

Met drie variabelen

- Hieronder is een waarheidstabel en een Karnaughdiagram gegeven voor drie variabelen. Merk op dat aangrenzende mintermen in slechts één variabele verschillen.

a	b	c	s
0	0	0	f_0
0	0	1	f_1
0	1	0	f_2
0	1	1	f_3
1	0	0	f_4
1	0	1	f_5
1	1	0	f_6
1	1	1	f_7

aangrenzend

c \ ab (s)	00	01	11	10
0	f_0	f_2	f_6	f_4
1	f_1	f_3	f_7	f_5

DE HAAGSE HOGESCHOOL

Voorbeeld met drie variabelen

- De omranding is als volgt:
- De groene omranding levert $a \cdot b$
- De blauwe omranding levert $a \cdot c$
- De rode omranding levert $b \cdot c$
- De totale functie is $s = a \cdot b + a \cdot c + b \cdot c$

s

c \ ab	00	01	11	10
0	0	0	1	0
1	0	1	1	1

DE HAAGSE HOGESCHOOL

Voorbeeld met drie variabelen

- Gegeven hetzelfde Karnaughdiagram
- De rode omranding levert $\overline{a} \cdot \overline{c}$
- De groene omranding levert $a \cdot b$
- De blauwe omranding levert $\overline{b} \cdot c$
- De totale functie is $s = a \cdot b + \overline{a} \cdot \overline{c} + \overline{b} \cdot c$

s

c \ ab	00	01	11	10
0	1	1	1	0
1	1	0	1	1

Met vier variabelen

- Hiernaast is een Karnaughdiagram voor vier variabelen gegeven.

- Let op de verdeling van de functiewaarden

a	b	c	d	s
0	0	0	0	f_0
0	0	0	1	f_1
0	0	1	0	f_2
0	0	1	1	f_3
0	1	0	0	f_4
0	1	0	1	f_5
0	1	1	0	f_6
0	1	1	1	f_7
1	0	0	0	f_8
1	0	0	1	f_9
1	0	1	0	f_{10}
1	0	1	1	f_{11}
1	1	0	0	f_{12}
1	1	0	1	f_{13}
1	1	1	0	f_{14}
1	1	1	1	f_{15}

cd \ ab (s)	00	01	11	10
00	f_0	f_4	f_{12}	f_8
01	f_1	f_5	f_{13}	f_9
11	f_3	f_7	f_{15}	f_{11}
10	f_2	f_6	f_{14}	f_{10}

Karnaughdiagrammen

- Gegeven het Karnaughdiagram aan de rechterkant.

- Voor het uitwerken van de functie zijn vier termen nodig.

- De functie is:

$$s = c \cdot d + \bar{a} \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$$

s

cd \ ab	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	1	1	1	1
10	0	1	0	1

Optellen

- Optellen in het binaire stelsel is identiek aan optellen in het decimale stelsel. In totaal moeten er per kolom drie bits worden opgeteld.

- In het voorbeeld worden twee 8-bit getallen opgeteld.

carry's	1	1	1	1	0	0	0	0	
getal A →	0	1	1	0	1	1	0	0	
getal B →	0	1	0	1	1	0	1	0	+
	1	1	0	0	0	1	1	0	

Full adder

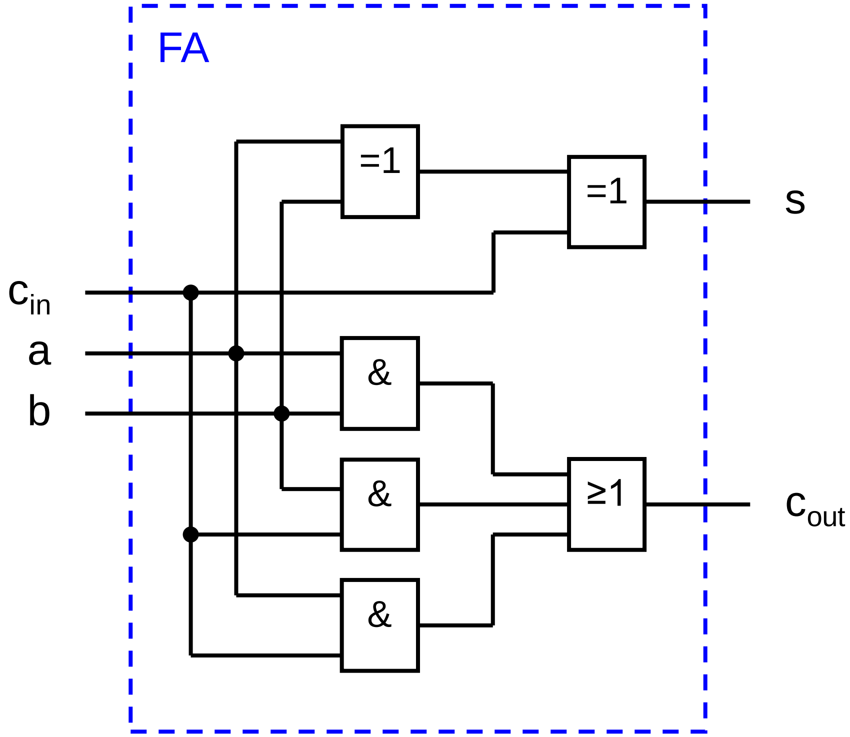
- Een full adder is in staat om drie bits op te tellen.
- De variabelen a en b zijn de bits van de getallen.
- De variabele c_{in} is de *inkomende* carry.
- c_{out} is de *uitgaande* carry.
- s is de sombit.

c_{in}	a	b	c_{out}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

DE HAAGSE HOGESCHOOL

Full adder

- Het schema kan als volgt worden opgebouwd.

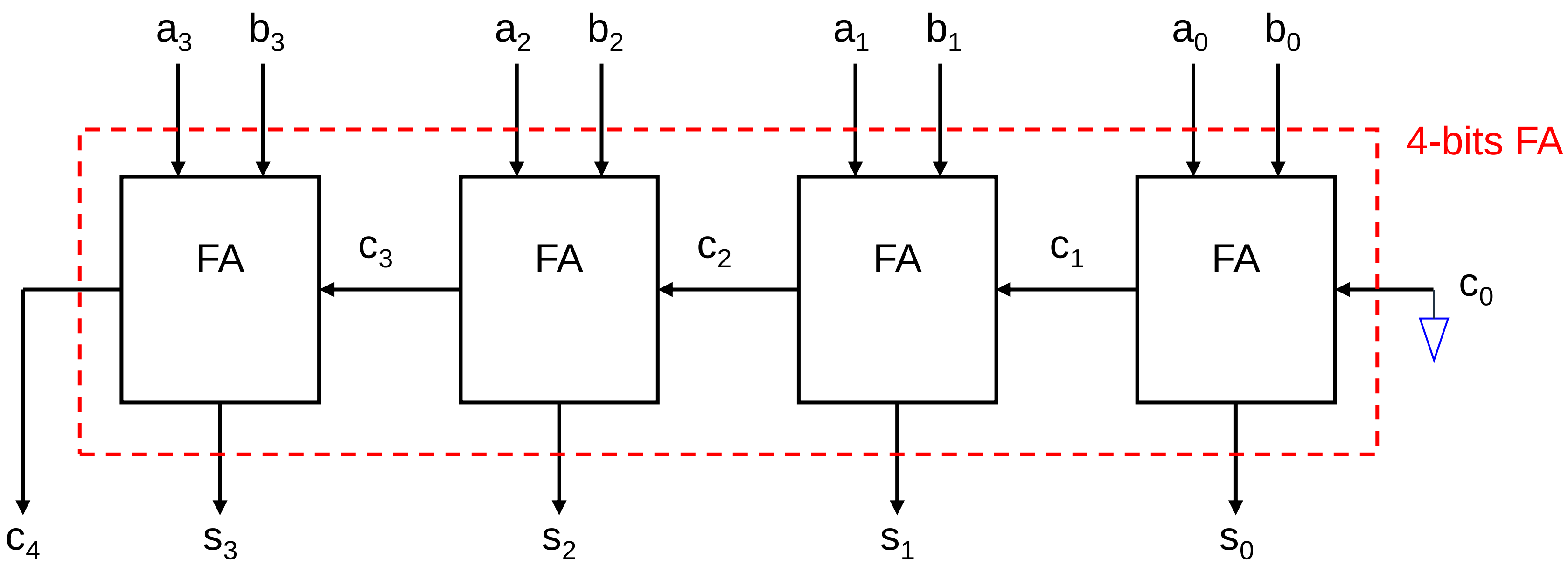


$$s = c_{in} \oplus (a \oplus b)$$

$$c_{out} = a \cdot b + a \cdot c_{in} + b \cdot c_{in}$$

4-bits Full Adder

- Een 4-bits opteller kan worden opgebouwd uit een *cascadeschakeling* van 1-bit full adders.



c_4 kan als 5^e sombit gebruikt worden

Two's complement

- Een positief getal wordt negatief gemaakt door het positieve getal af te trekken van een macht van 2, bijvoorbeeld 2^4 (16, 4-bits getal).

- Als voorbeeld +5 → – 5

$$\begin{array}{r} 10000_2 \\ 0101_2 \\ \hline 1011_2 \end{array} - \qquad \begin{array}{r} 2^4 \\ +5 \\ \hline 2^4 - 5 \end{array} -$$

- Het binaire getal 1011_2 is de *representatie* van -5_{10}. Merk op dat de getallen *modulo* 2^4 zijn!

DE HAAGSE HOGESCHOOL

Two’s complement

- We kunnen nu de getallen gebruiken.
- Als voorbeeld: +5 + (–5) = 0

$$\begin{array}{rl} 0101 & \quad +5 \\ 1011 & \quad 2^4 - 5 \\ \hline 1)\ 0000 & \quad 2^4 + 0 \\ \downarrow & \\ 0000 & \quad 0 \end{array} \; +$$

- De uitgaande carry moet geschrapt worden, die is het resultaat van de wijze waarop de two's complement representatie werkt.

Two’s complement

- Een binair getal aftrekken van 1...1 is heel gemakkelijk.
- Het antwoord is namelijk de inverse van de afzonderlijke bits van dat getal (*flip bits*). Daarna moet er 1 bij opgeteld worden (*add 1*). Dit werkt ook van negatief naar positief.
- Als voorbeeld +5 → –5 en –5 → +5

0101	flip bits	1011	(–5)
1010		0100	
1 +	add 1	1 +	
1011		0101	(+5)

Two's complement

- Er is geen tegengesteld getal voor –8. Er is slechts één representatie van 0.

1000	flip bits	(–8)		0000	flip bits	(+0)
0111				1111		
1	add 1			1	add 1	
1000		(+8?)		1 0000		(–0)

- Two's complement heeft een asymmetrisch bereik.

Two’s complement

- Het omzetten van een two's complement naar decimaal kan eenvoudig door het tekenbit als *negatief* gewicht te gebruiken. De overige bits hebben hetzelfde gewicht als bij unsigned.

- Voorbeeld: 1011 $\rightarrow$ $-1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = -8 + 3 = -5$

  0110 $\rightarrow$ $0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 0 + 6 = 6$

Two’s complement

- Decimaal naar two’s complement:

- Positief getal: omzetten als bij unsigned (herhaald delen door 2)

- Negatief getal:
  - Bepaal aantal bits, bv. 8
  - Trek getal af van 2^8
  - Zet uitkomst om alsof het unsigned is.

- Dus -57 met 8 bits: $2^8 - 57 = 199$
  - Nu 199 omzetten als unsigned (met 8 bits). Levert mogelijk leidende 1-en.

Unsigned versus two's complement

- Hoe de getallen gebruikt/gelezen moeten worden, hangt af van de *interpretatie* van de gebruiker. Voor de hardware is er geen verschil*.

unsigned	bits	two's compl.	dec
+15	1111	$2^4 - 1$	–1
+15 +	1111 +	$2^4 - 1$ +	–1 +
+16 +14	1 1110	$2^4 + 2^4 - 2$	–2
+14	1110	$2^4 - 2$	–2

* En dat was ook de bedoeling!

Overflow

Enige rekenvoorbeelden

+5	0101	–8	1000
+5 +	0101 +	–8 +	1000 +
+10	1010 (= -6)	–16	1 0000 (= 0)
–7	1001	+4	0100
–4 +	1100 +	–8 +	1000 +
–11	1 0101 (= +5)	–4	1100

De uitgaande 1 (carry!) moet genegeerd worden.

Overflow

- Het valt aan te tonen dat overflow kan worden gedetecteerd door:

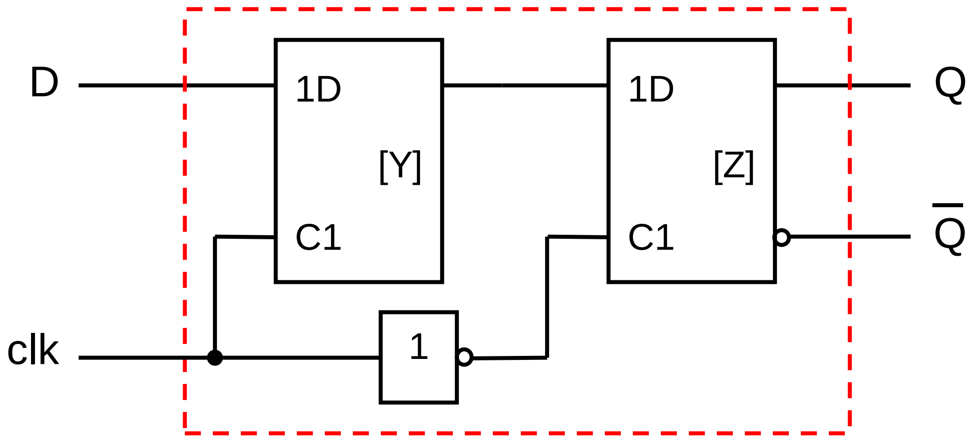
$$V = C_4 \oplus C_3$$

- Er is dus overflow als de inkomende carry van de tekenbit en de uitgaande carry van de tekenbit ongelijk zijn.

- Dit geldt zowel voor optellen als voor aftrekken.

D-flipflop

- Hieronder de gerealiseerde D-flipflop. Deze flipflop klokt de data in op de neergaande flank.



D-flipflop

- De D-flipflop neemt de data aan de ingang over op de *neergaande flank*.

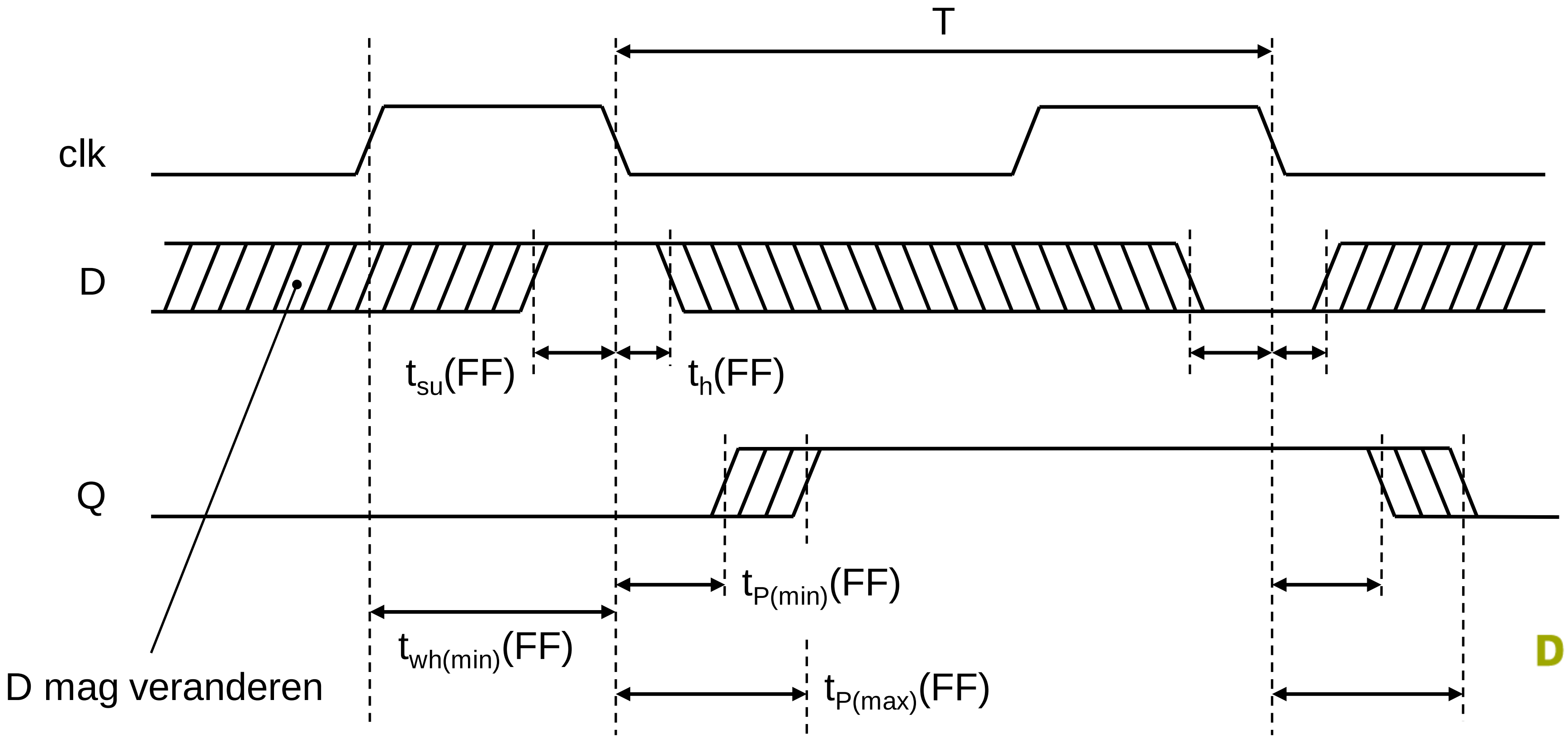
- Het timing-voorschrift wordt *flankgestuurde* of *edge-triggered timing* genoemd.

- 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

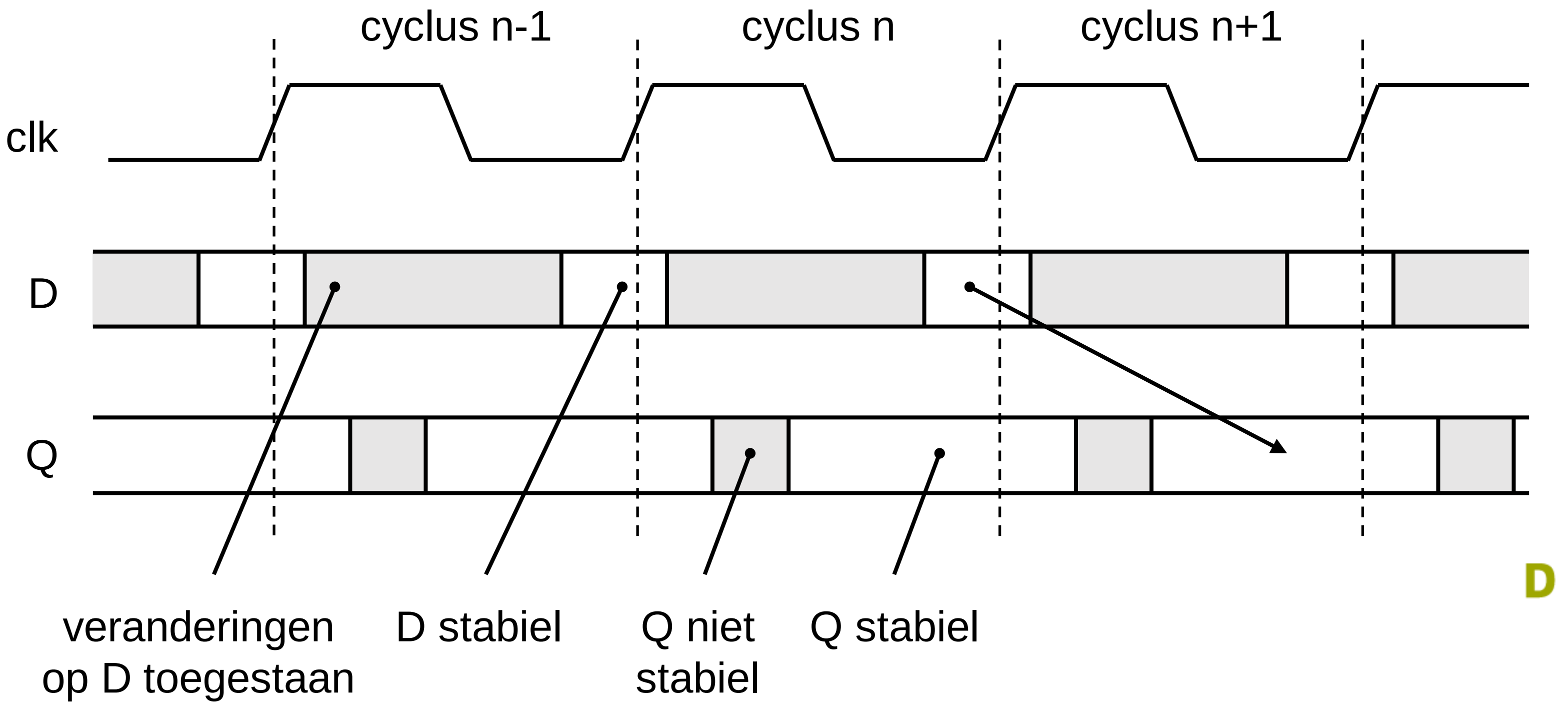
- Hieronder de timing van de negative edge-triggered D-flipflop.



DE HAAGSE HOGESCHOOL

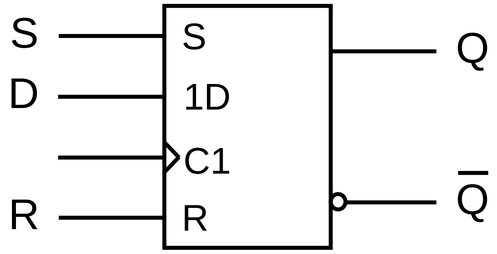
D-flipflop

- 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: $Q^{n+1} = D^{n}$

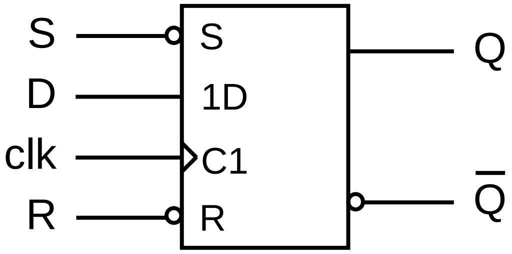


D-flipflop

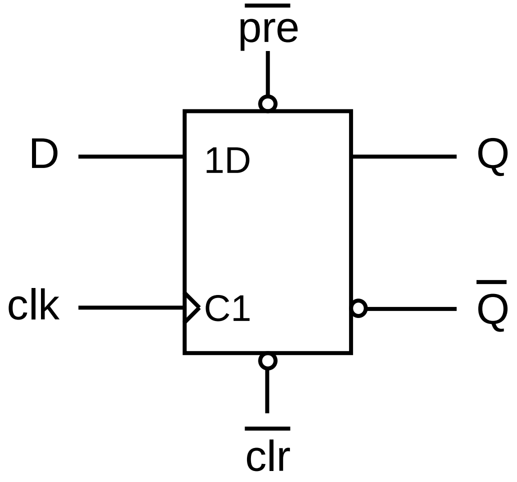
- Hieronder de symbolen van de D-flipflop met asynchrone reset en set.



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



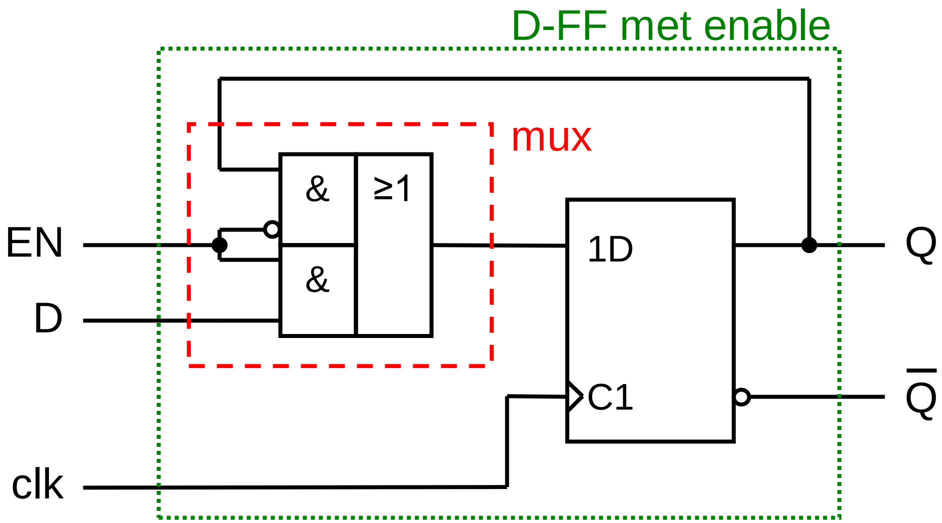
positive edge-triggered D-flipflop with asynchronous active low set and reset



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

D-flipflop

- Flipflop met enable: de flipflop klokt zijn eigen waarde óf een nieuwe waarde in door middel van een multiplexer.



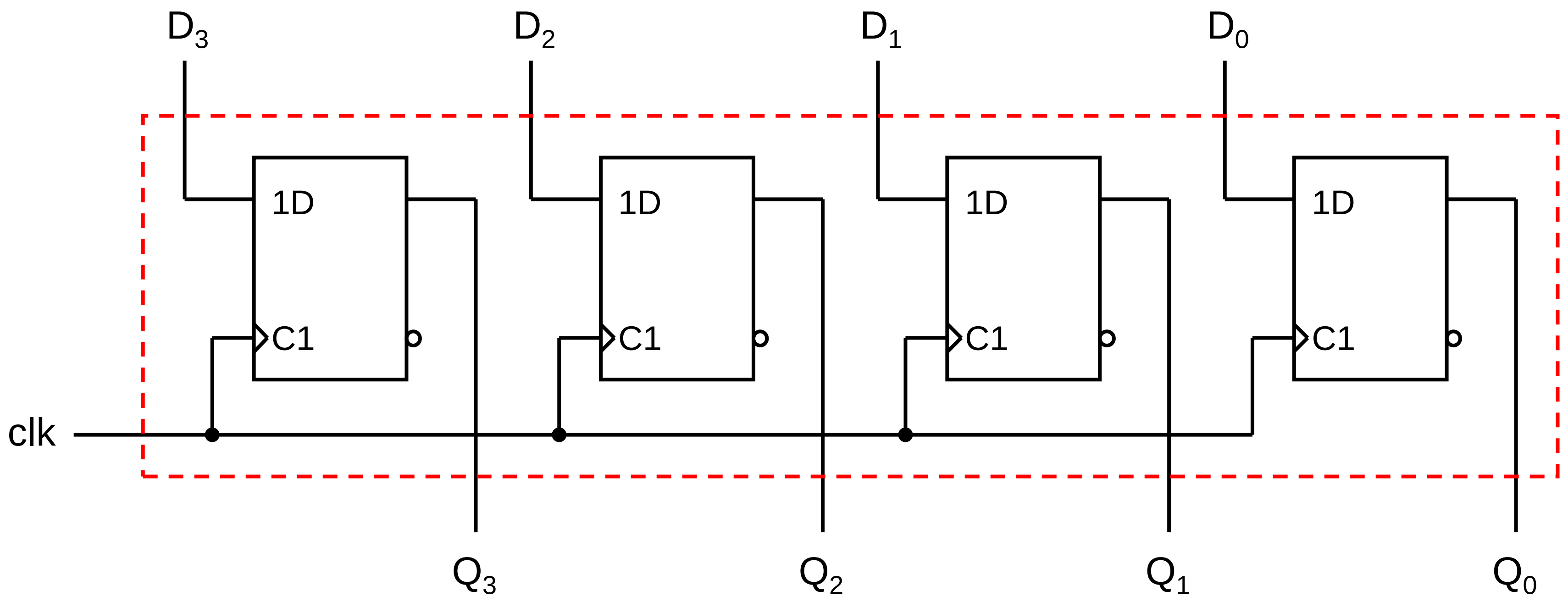
$$Q^{n+1} = \overline{EN^n} \cdot Q^n + EN^n \cdot D^n$$

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

DE HAAGSE HOGESCHOOL

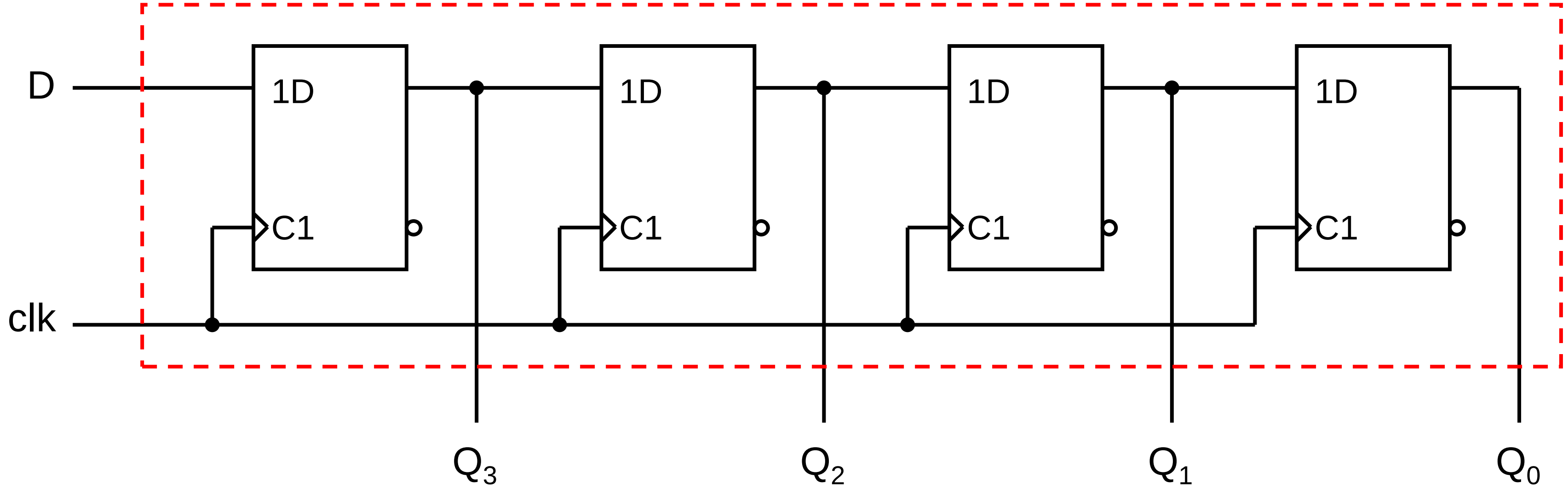
D-flipflop

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



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