



BASIS ELEKTROTECHNIEK 1 – WEEK 6

overtref jezelf



AGENDA

- ▶ Sequentiële Logica
- ▶ Latches en Flip-flops



SEQUENTIËLE LOGICA

Sequentiële logica

- Verschil met combinatorische logica kunnen uitleggen
- Flip-flops kunnen beschrijven
- Uitgangen kunnen bepalen

Combinatorisch versus Sequentieel

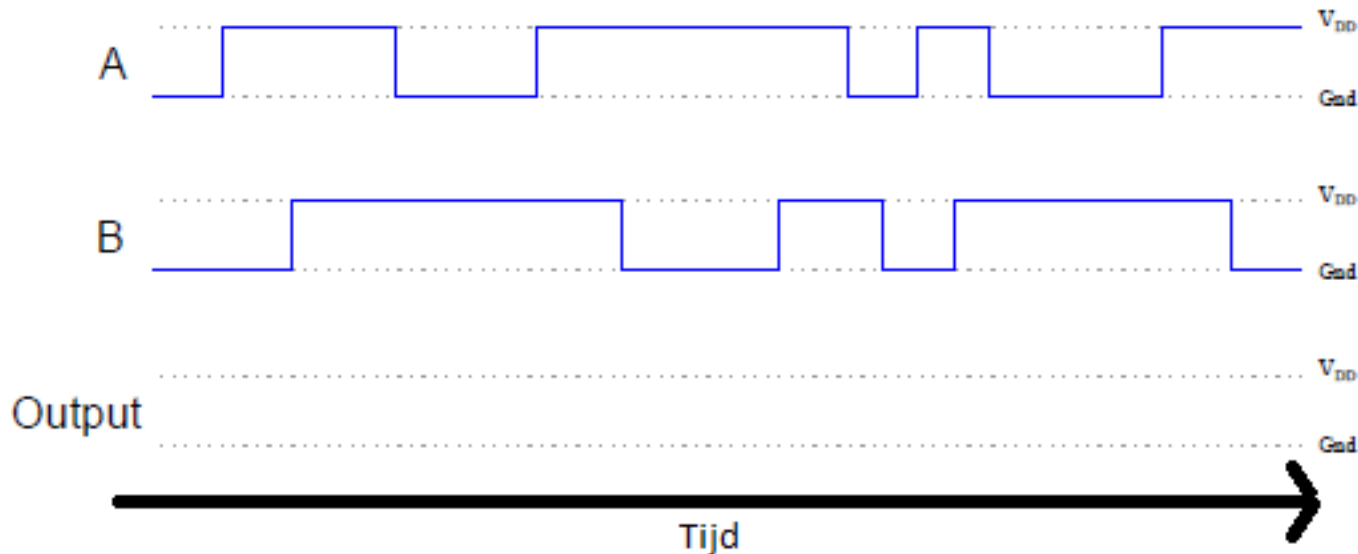
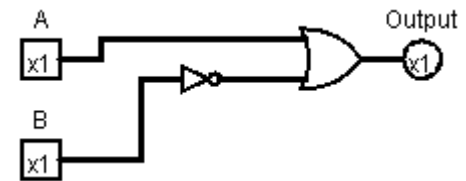
- **Combinatorisch**
 - Uitgang enkel afhankelijk van huidige ingang(en)
 - Parallel, zonder tijd
- **Sequentieel**
 - Uitgang afhankelijk van huidige EN vorige ingangen
 - Geheugen: Toestand (van de schakeling) onthouden
 - Maakt vaak gebruik van een klok voor synchronisatie van de schakeling

Tijdsdiagrammen

- Met behulp van tijdsdiagrammen kunnen we verschillende signalen weergeven in de tijd.

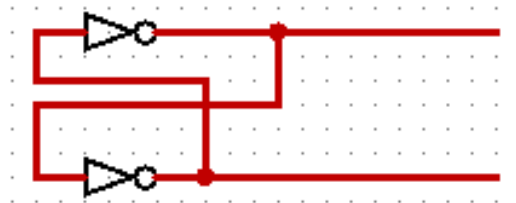
- Oefening: Vul figuur 1 aan:

$$Output = A + \overline{B}$$



Latches en Flipflops

- Het basis element van sequentiële logica
- Heeft 2 toestanden. Of '0' of '1'
 - 1-Bit geheugen element
- Doel: Vasthouden/onthouden van een waarde
- Maakt gebruik van Logica met *feedback*
 - Ook wel *multivibrator* genoemd



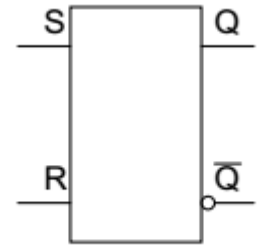
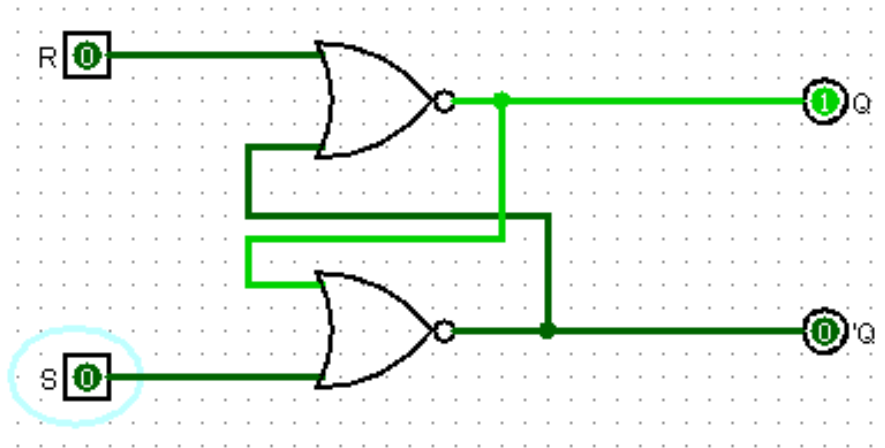
Varianten

- SR-Latch
- Gated SR-Latch
- SR-Flipflop
- D-Latch
- D-Flipflop
- JK-Flipflop

SR Latch

- We moeten de waarde kunnen instellen

- Set/Reset Latch



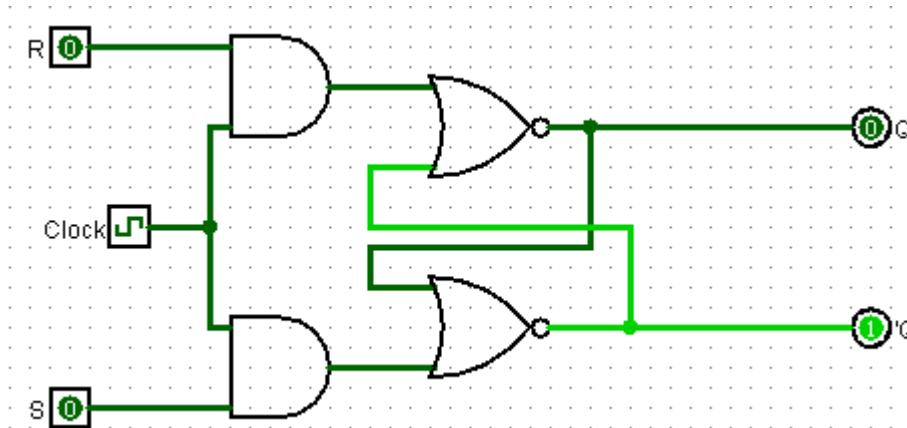
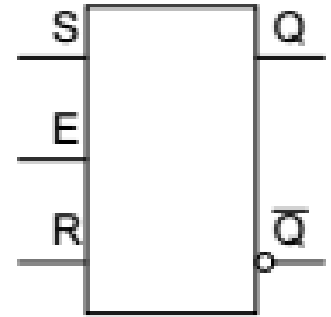
S	R	Q_n
0	0	Q_{n-1}
0	1	0
1	0	1
1	1	X

- Waarheidstabel

- Subscript n verwijst naar de huidige toestand
 - Subscript n-1 verwijst naar vorige toestand
 - De toestand voordat de ingangssituatie veranderde

Gated SR Latch

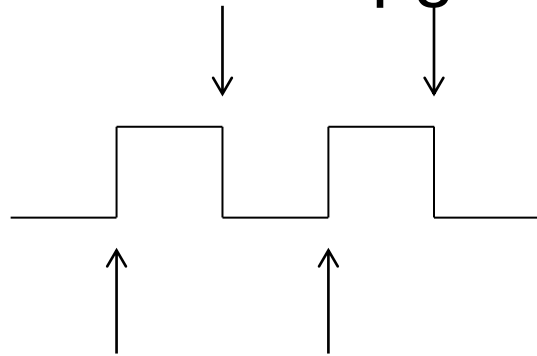
- IC's synchroon laten lopen in grotere schakeling
- "Enable" schakelt de ingangen actief of inactief



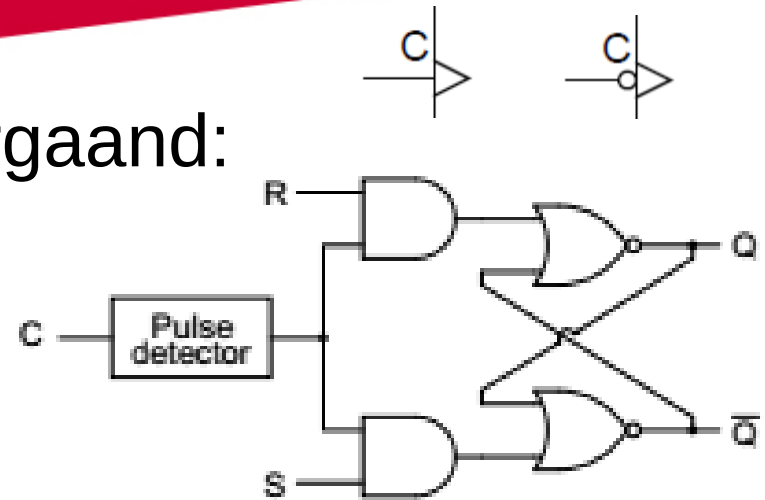
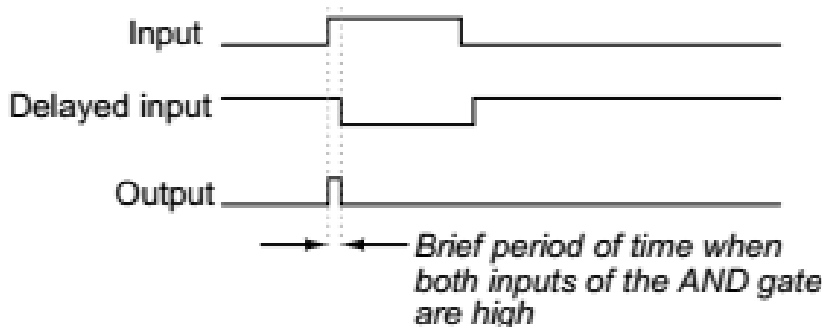
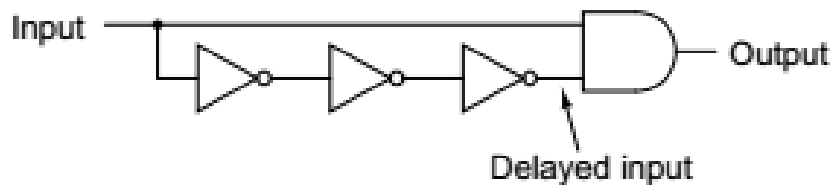
S	R	Enable	Q_n
0	0	1	Q_{n-1}
0	1	1	0
1	0	1	1
1	1	1	X
X	X	0	Q_{n-1}

Flanken en Flankdetectie

- Flanken: opgaand en neergaand:

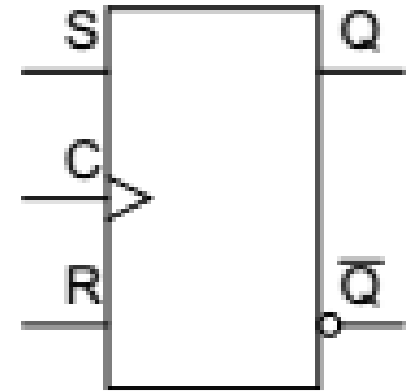


- Voorbeeld simpele flankdetectie schakeling:



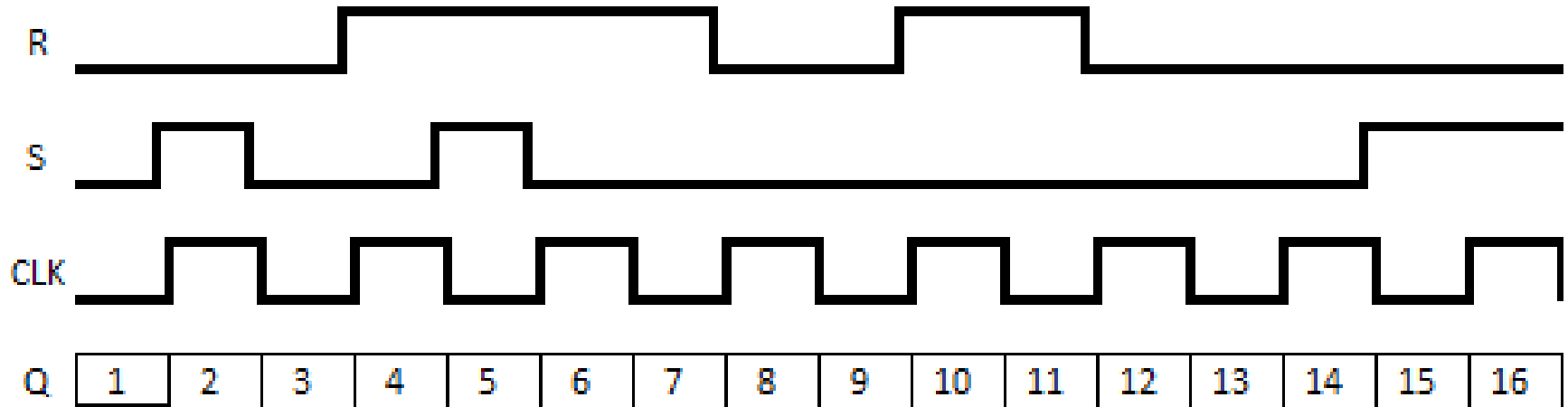
SR Flipflop

- Bijna dezelfde werking
 - In plaats van *enable* een *klok ingang*
- Schakelt alleen bij een opgaande (of soms neergaande) *klokflank*
 - Ingang wordt bekeken
 - Uitgang wordt ingesteld op basis van de ingangen
- Oefening: Vul figuur 2 aan



S	R	Clk	Q_n
0	0	↑	Q_{n-1}
0	1	↑	0
1	0	↑	1
1	1	↑	X
X	X	-	Q_{n-1}

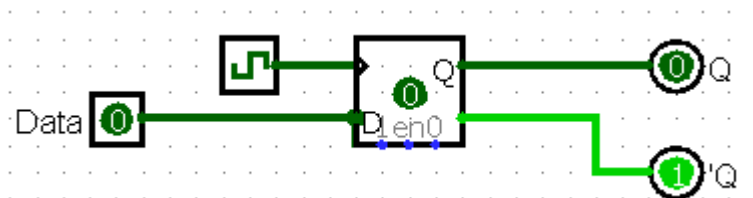
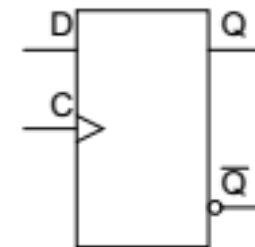
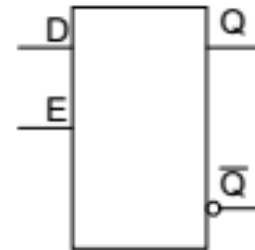
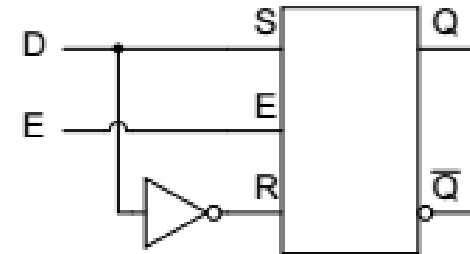
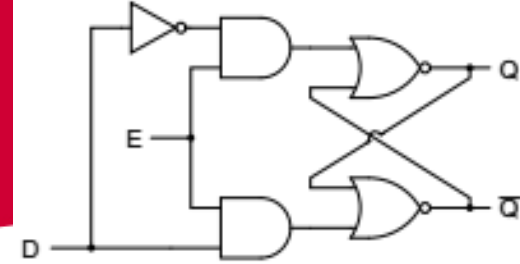
Figuur 2



S	R	Clk	Q_n
0	0	↑	Q_{n-1}
0	1	↑	0
1	0	↑	1
1	1	↑	X
X	X	-	Q_{n-1}

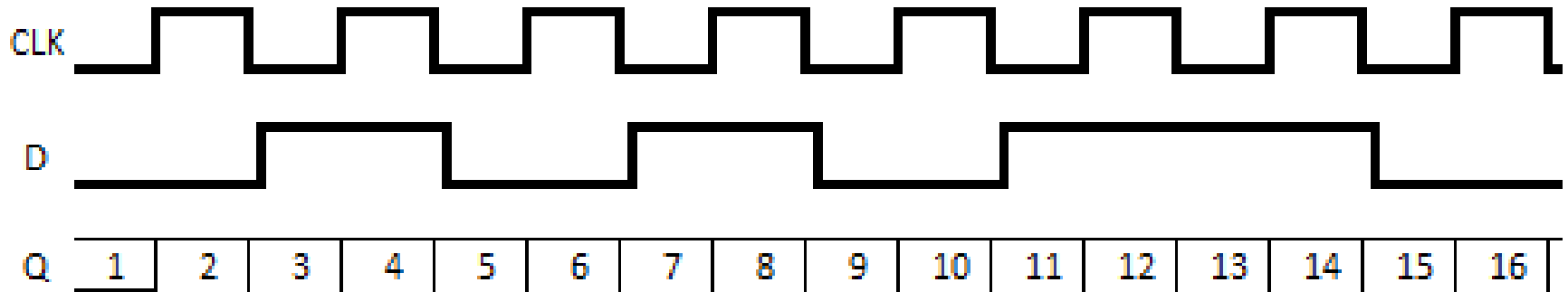
D-Latch en D-Flipflop

- Varianten van de SR-latch en SR-Flip-flop
- Combinatie van S en R in D(ata)
 - $S = D, R = \overline{D}$
 - Vertraagt ingangssignaal met 1 klokperiode
 - Bijvoorbeeld voor het synchroniseren van combinatorische signalen.
- Oefening: Vul figuur 3 aan, gebruik deze schakeling:



D	Clk	Q_n
0	↑	0
1	↑	1
X	-	Q_{n-1}

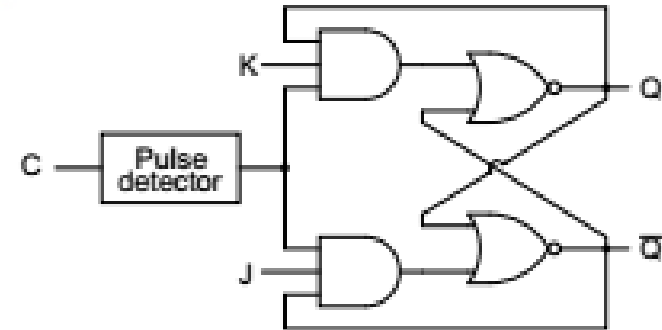
Figuur 3



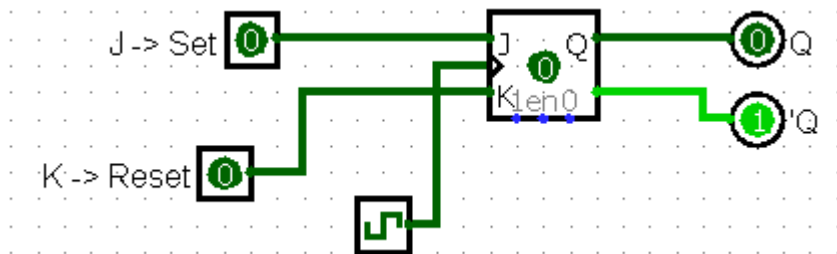
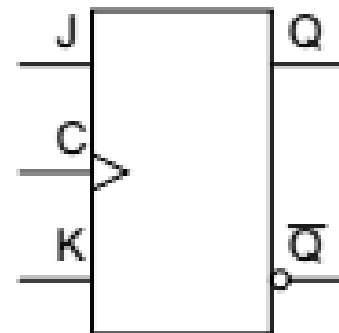
D	Clk	Q_n
0	↑	0
1	↑	1
X	-	Q_{n-1}

JK-Flipflop

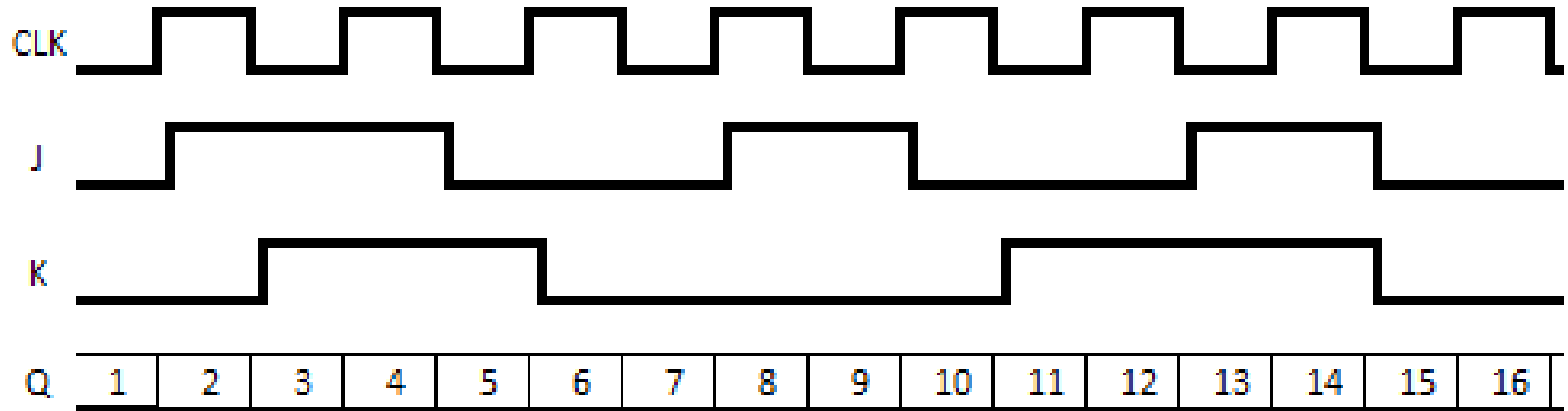
- Variant van de SR Flip-flop
 - Clocked
 - J = Set
 - K = Reset
- $J = 1$ en $K = 1 \rightarrow Toggle$ output.
- Oefening: Vul figuur 4 aan.



J	K	Clk	Q_n
0	0	$\uparrow$	Q_{n-1}
0	1	$\uparrow$	0
1	0	$\uparrow$	1
1	1	$\uparrow$	$\overline{Q_{n-1}}$
X	X	—	Q_{n-1}

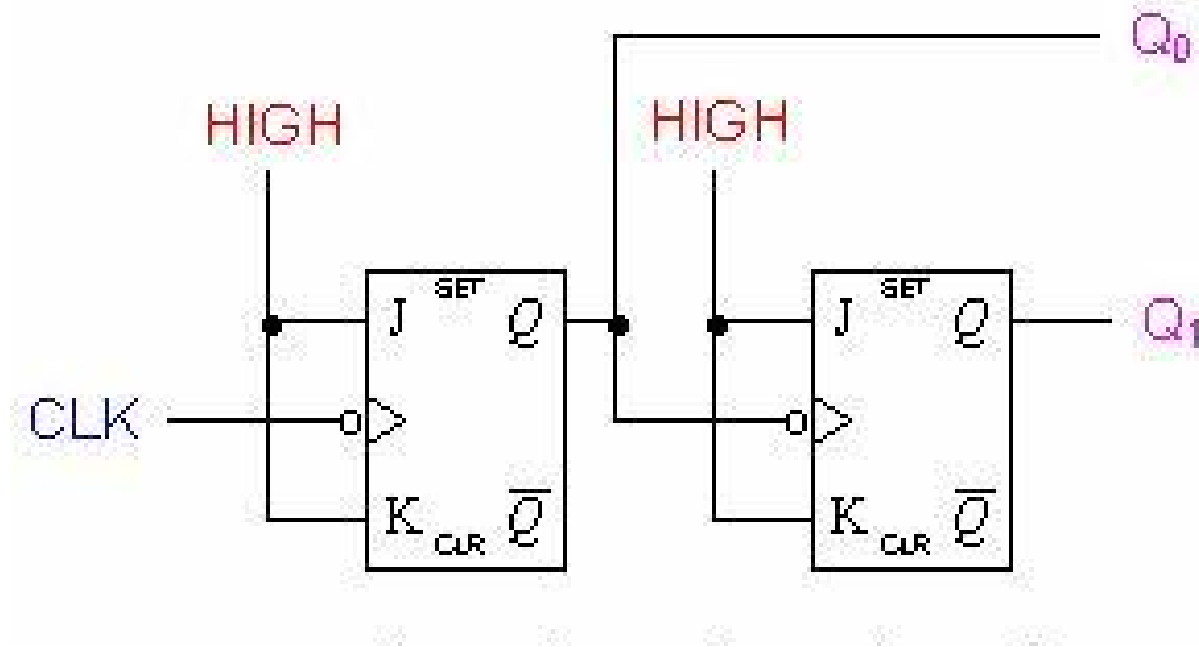


Figuur 4

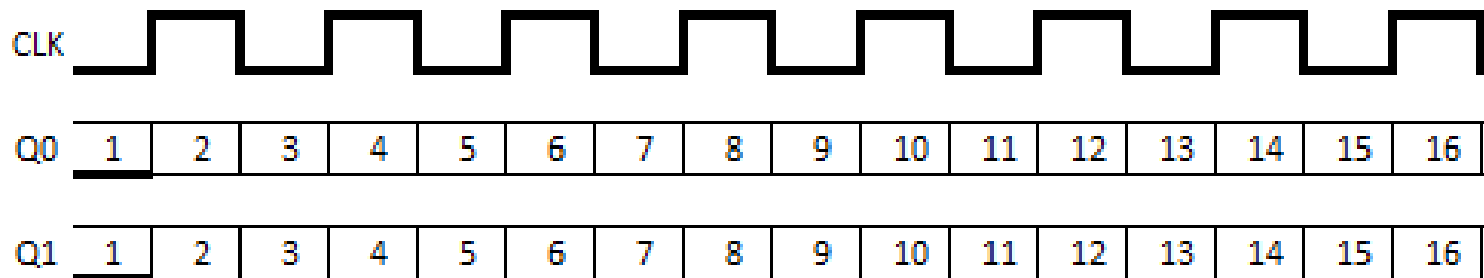


J	K	Clk	Q_n
0	0	↑	Q_{n-1}
0	1	↑	0
1	0	↑	1
1	1	↑	$\overline{Q_{n-1}}$
X	X	—	Q_{n-1}

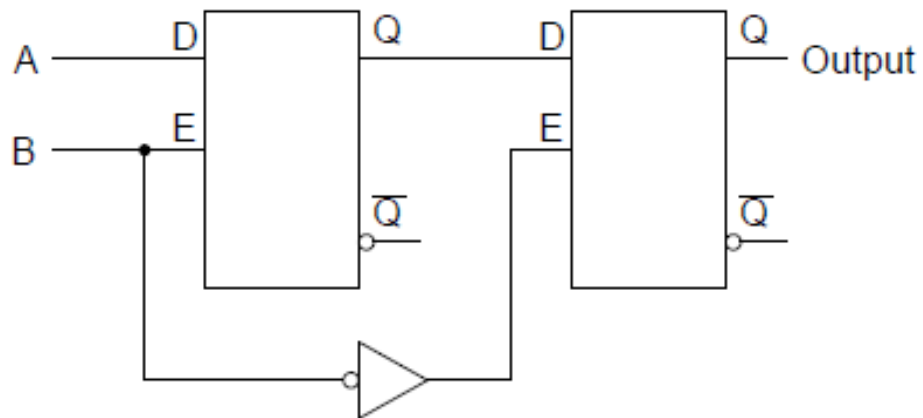
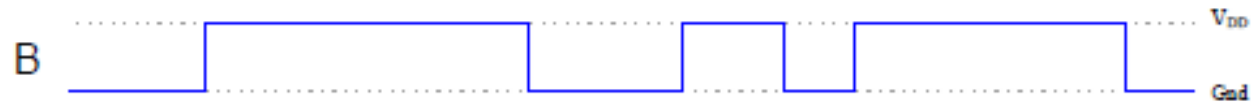
Oefening: Vul figuur 5 aan.



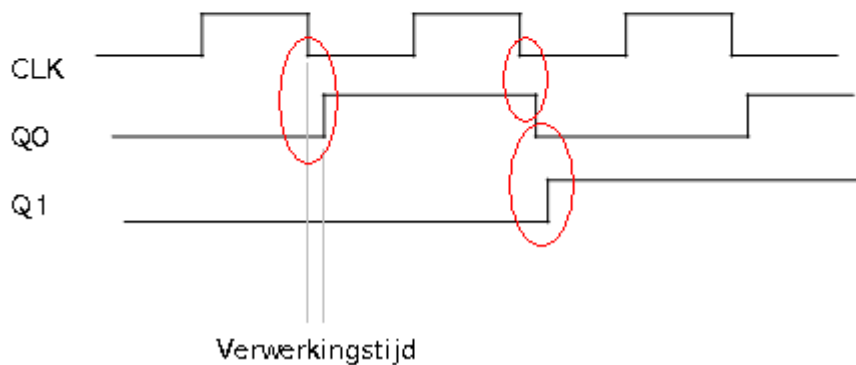
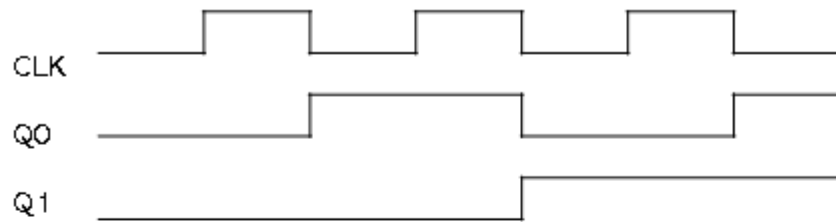
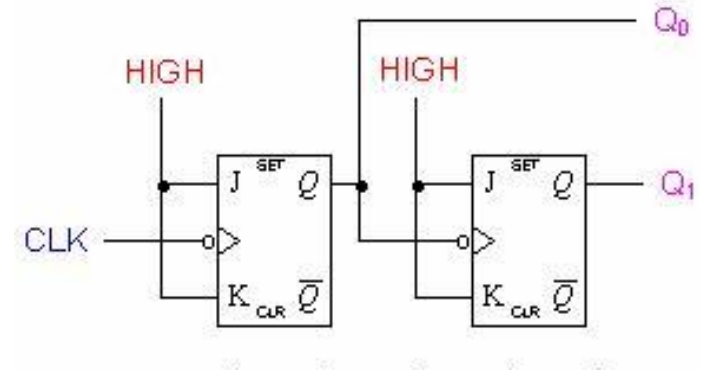
J	K	Clk	Q_n
0	0	$\uparrow$	Q_{n-1}
0	1	$\uparrow$	0
1	0	$\uparrow$	1
1	1	$\uparrow$	$\overline{Q_{n-1}}$
X	X	—	Q_{n-1}



Oefening: Vul figuur 6 aan.



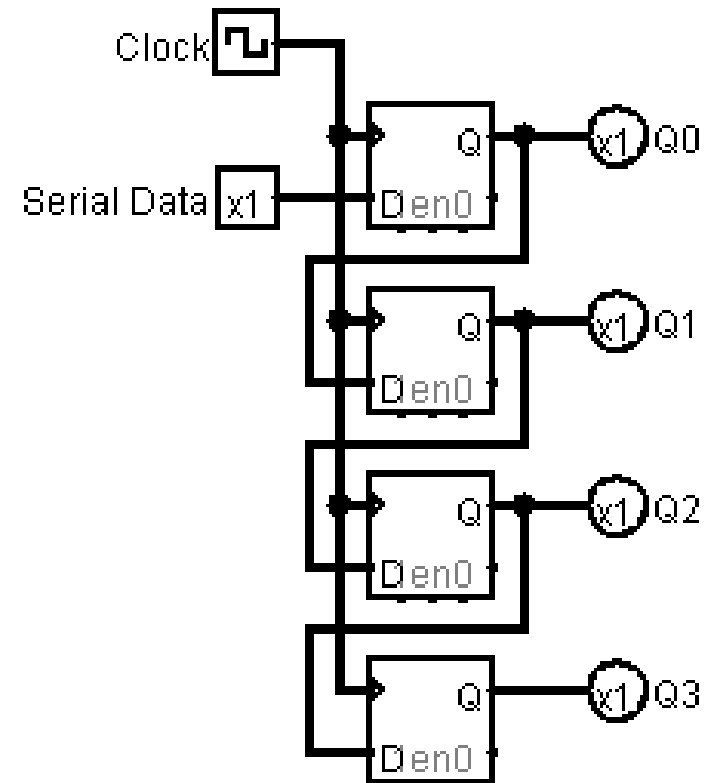
Vertragingen



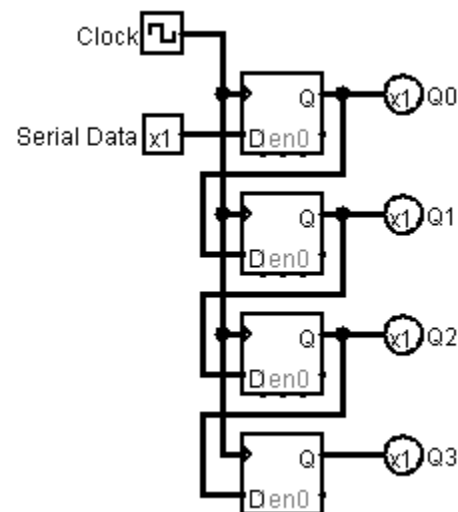
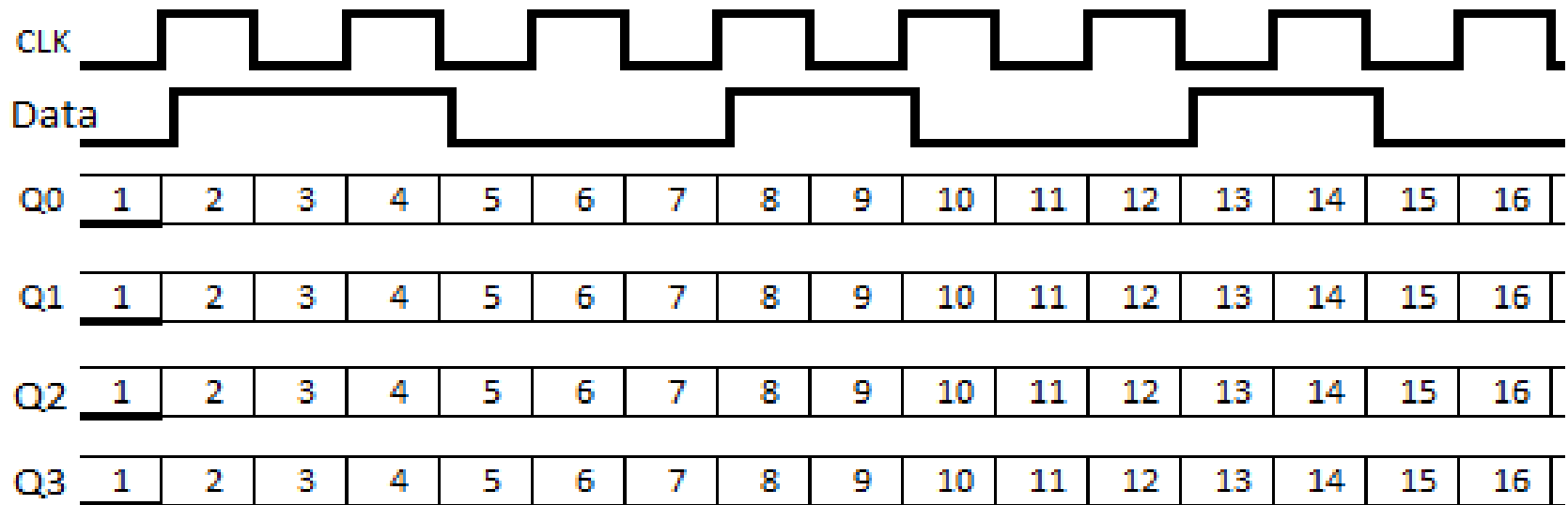
J	K	Clk	Q_n
0	0	↑	Q_{n-1}
0	1	↑	0
1	0	↑	1
1	1	↑	$\overline{Q_{n-1}}$
X	X	—	Q_{n-1}

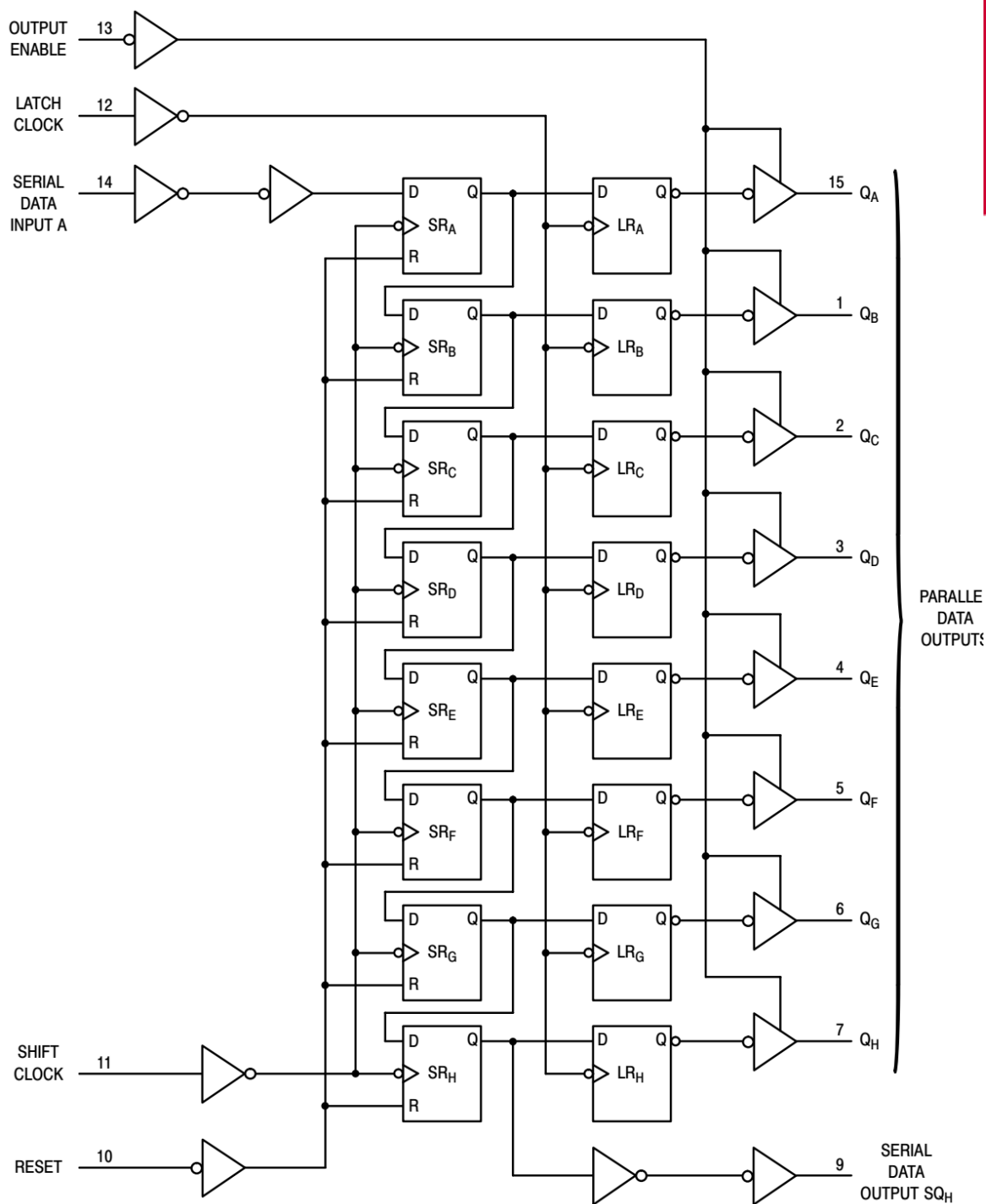
Voorbeeld / Oefening:

- 4-bits Serial In, Parallel Out (Shift Register)
- Oefening: Vul figuur 7 aan
- Vraag: Vanaf reset, hoeveel kloktikken zijn er nodig voordat de ingangsdata Q3 kan beïnvloeden? De ingang kan 1 of 0 zijn.



Figuur 7

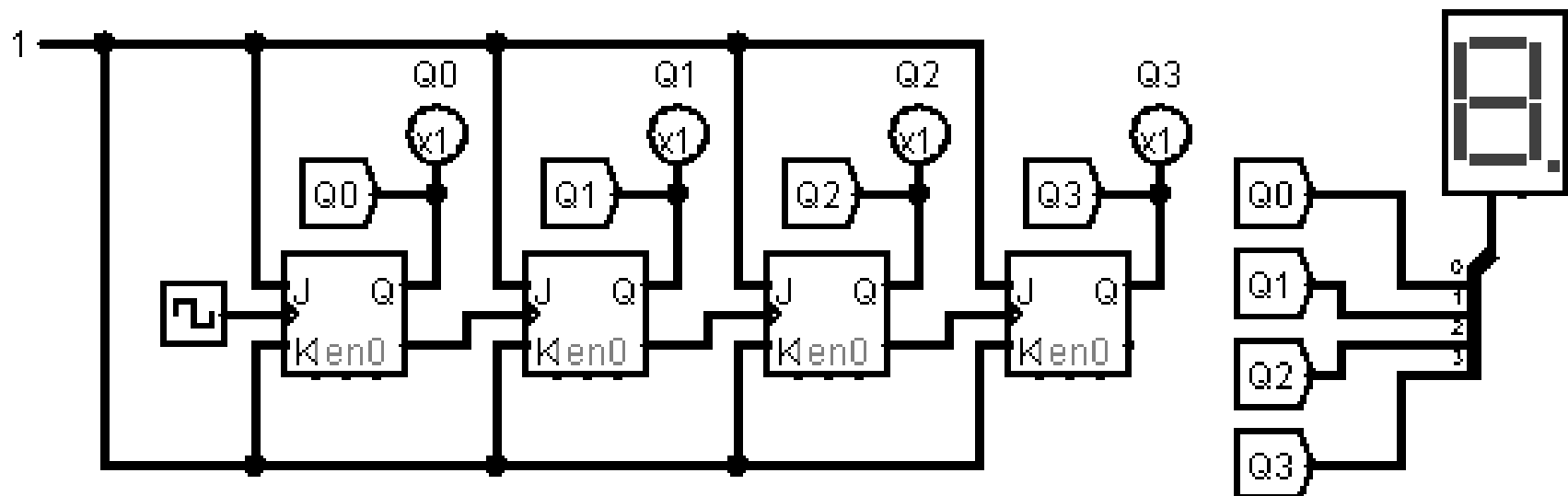




In de praktijk:

- 8-bit shiftregister 74HC595

Oefening: Vul figuur 8 aan. Wat doet deze schakeling?



Q0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
----	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Q1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
----	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Q2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
----	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Q3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
----	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Meer informatie:

- Meer informatie over latches en flip-flops, en bron van diverse afbeeldingen:

“Lessons In Electric Circuits Volume IV – Digital”
by Tony R. Kuphaldt

- Beschikbaar op het netwerk

Samenvatting

- Latches en Flip-flops worden gebruikt om “*state*” bij te houden in een schakeling.
 - Uitgangen zijn afhankelijk van de huidige ingangen en de state (eerdere ingangen).
- Latches hebben vaak een *enable* ingang.
 - Op de enable kan ook klok worden aangesloten!
- Flip-flops veranderen alleen op een *flank* d.m.v. een flankdetectie circuit.
- Met behulp van tijdsdiagrammen kunnen we verschillende signalen weergeven in de tijd.

State Machines

- Met behulp van flip-flops kunnen we statemachines realiseren.
 - De flip-flops worden gebruikt om toestanden te onthouden
 - Huidige toestand + huidige ingang(en) bepalen volgende toestand
 - Denk bijvoorbeeld aan een slagboom of een frisdrankautomaat
- Volgende week gaan we zien hoe dit werkt.
 - Let op: Statemachines worden niet behandeld in het boek van Hambley.

Huiswerk en voorbereiding volgende week:

- Behandeld: Hambley § 7.1 t/m 7.6
- Huiswerkopgaven week 6 uit Hambley:
P7.76, P7.77, P7.78, P7.79, P7.80, P7.83, P7.84
- Huiswerk: Logisim opdrachten week 5/6 (Wiki)
- Op de wiki staan extra opgaven om te oefenen en ter voorbereiding op het tentamen



overtref jezelf