

EE1410: Digitale Systemen

BSc. EE, 1e jaar, 2012-2013, 6e hoorcollege

Arjan van Genderen, Stephan Wong, Computer Engineering
22-4-2013

Rooster 4e kwartaal (onder voorbehoud)

week	ma 1 ^e + 2 ^e uur	vr 5 ^e + 6 ^e uur	verder
1	Implementatie technologie	VHDL	
2		Standaard modules	toets
3	VHDL		
4	RTL sys./microproc.		toets
5		werkcollege 3: gr. A1 – B4	
6	werkcollege 3: gr. C1 – D4	werkcollege 4: gr. A1 – B4	
7	werkcollege 4: gr. C1 – D4	vragencollege	

Hoor en vragencollege in zaal B, toetsen woensdag 9:45-10:30 DW zaal 2, werkcollege in zaal DW-PC 1.010 op maandag en DW-PC 1.150 op vrijdag.

Verder ook nog een nieuwe VHDL opdracht !

Hoorcollege 6

- CMOS technologie
- Programmeerbare devices
- Geheugens
- Meer over flip-flops

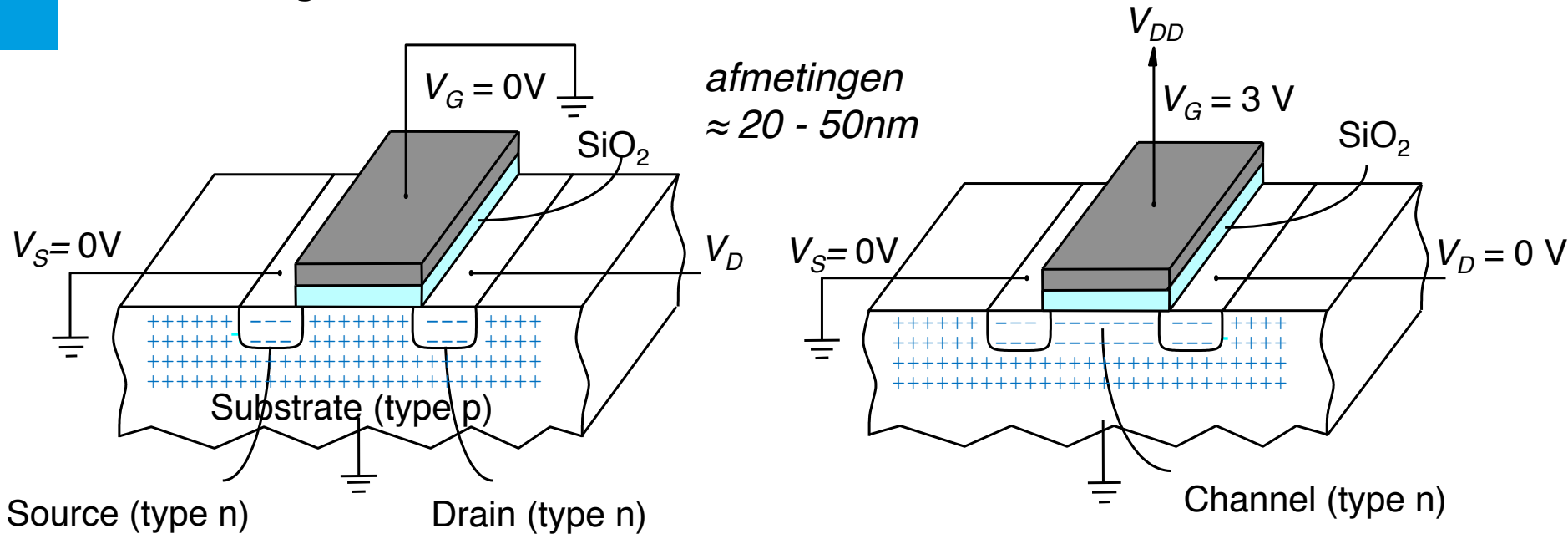
Corresponderende stof in boek "Digital Logic":

3 – 3.1, 3.3, 3.5 - 3.7, 3.8.7, 3.8.8 vanaf fan-out,
3.9, 3.11, 7.4.4 – 7.6, 7.15

CMOS Technologie

De MOS transistor

- CMOS = Complementary MOS (NMOS en PMOS transistors)
- Werking NMOS transistor:



$V_G = 0\text{ V}$: transistor is off

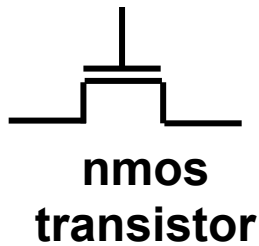
$V_G = 3\text{ V}$: transistor is on

- Werking PMOS transistor (p en n gebieden andersom):

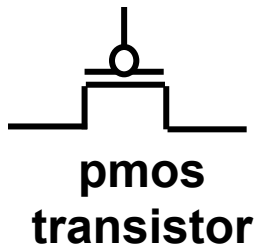
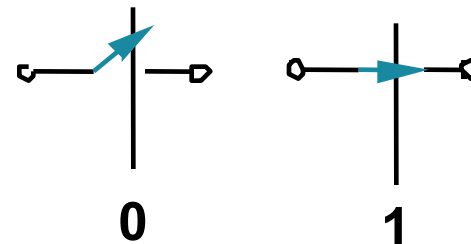
$V_G = 0\text{ V}$: transistor is on

$V_G = 3\text{ V}$: transistor is off

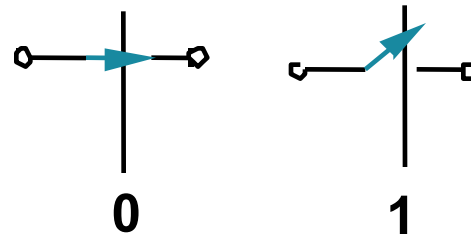
De “digitale abstractie” van een NMOS en PMOS transistor



Normally Open:

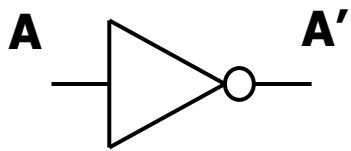


Normally Closed:

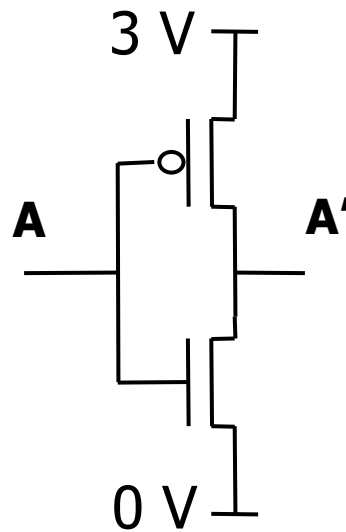


CMOS Inverter

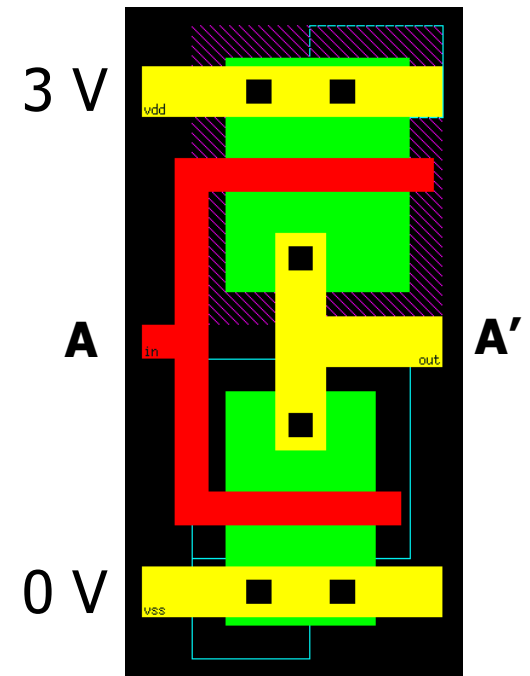
Logisch:



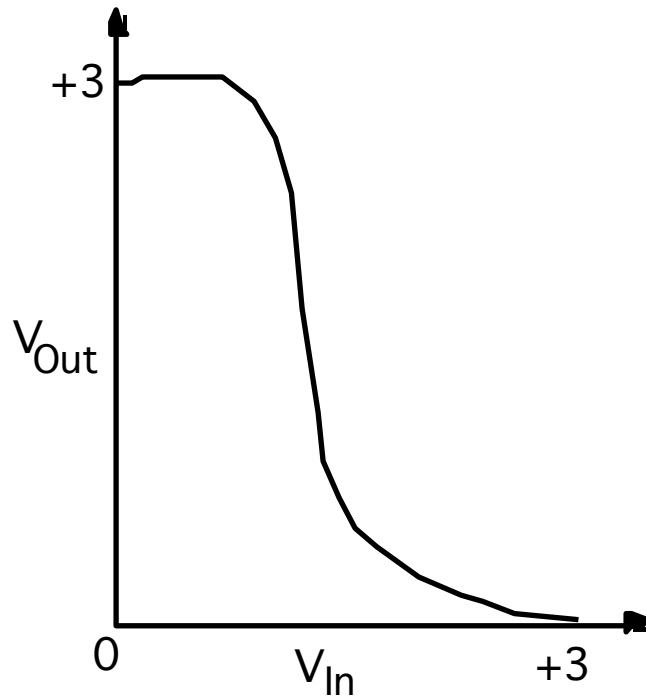
Elektrisch



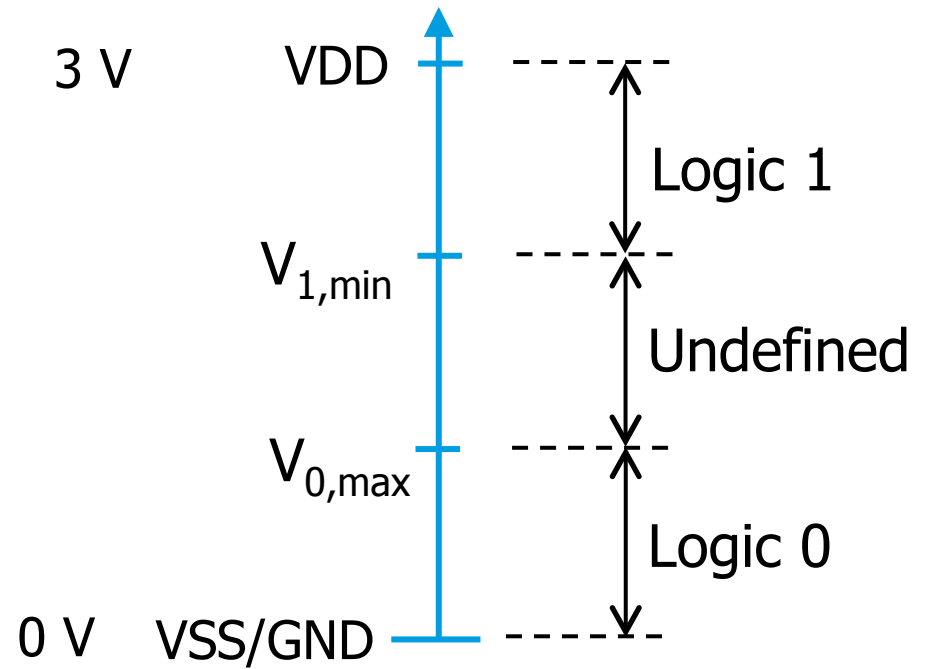
Layout



Gedrag inverter

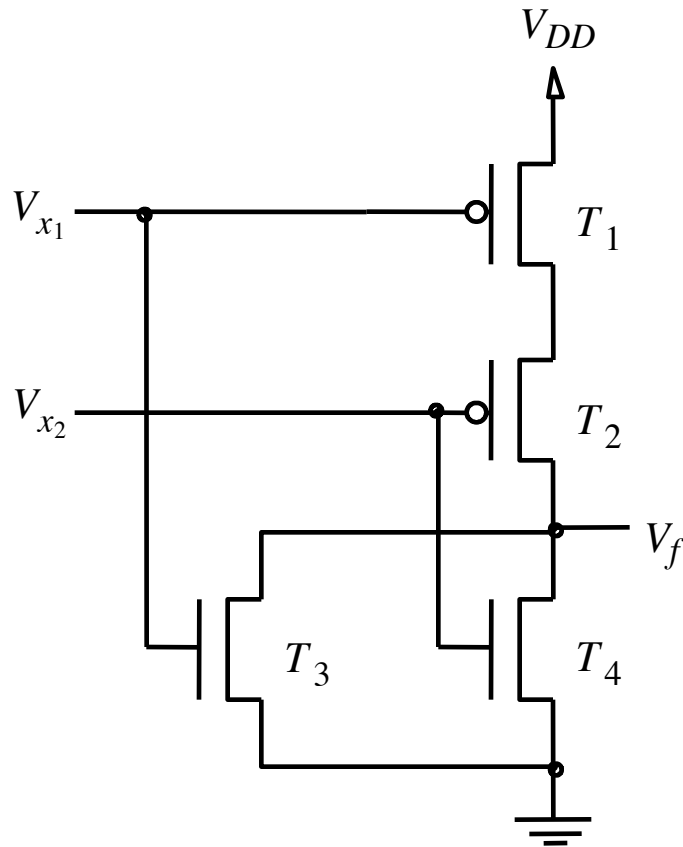


**Werkt eigenlijk analoog
maar de eindwaarden
zijn “digitaal”**



**Voor logische 0 en 1
worden spanningsmarges
aangehouden**

CMOS NOR Gate

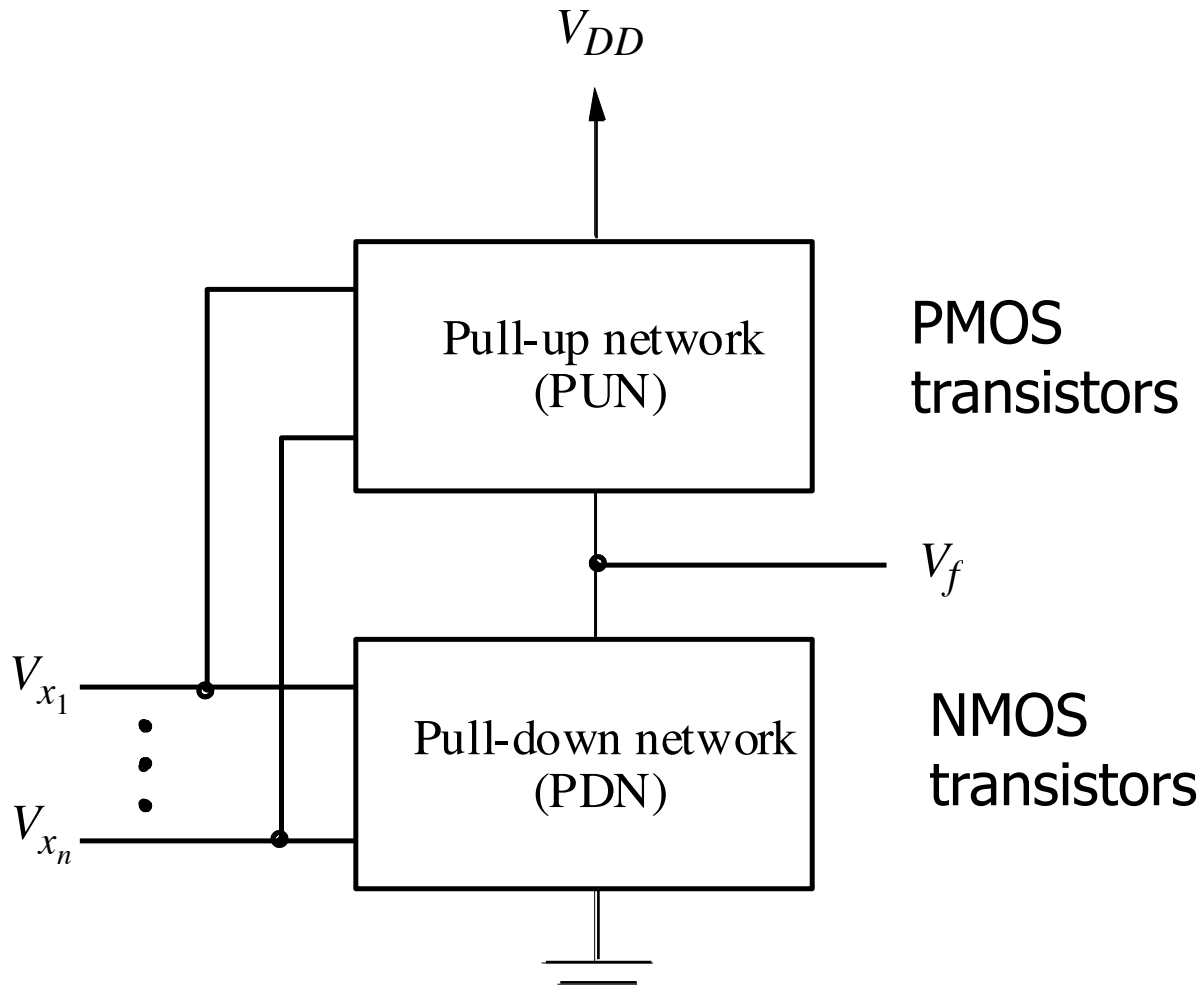


(a) Circuit

x_1	x_2	T_1	T_2	T_3	T_4	f
0	0	on	on	off	off	1
0	1	on	off	off	on	0
1	0	off	on	on	off	0
1	1	off	off	on	on	0

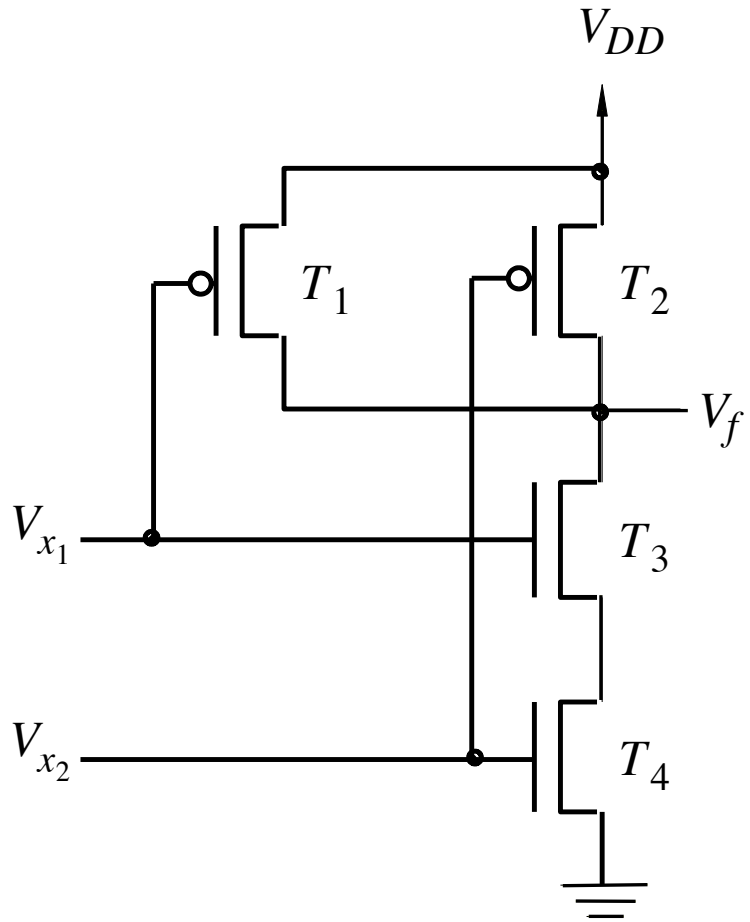
(b) Truth table and transistor states

CMOS Principe



(PUN on, PDN off)
of
(PUN off, PDN on)

CMOS NAND Gate

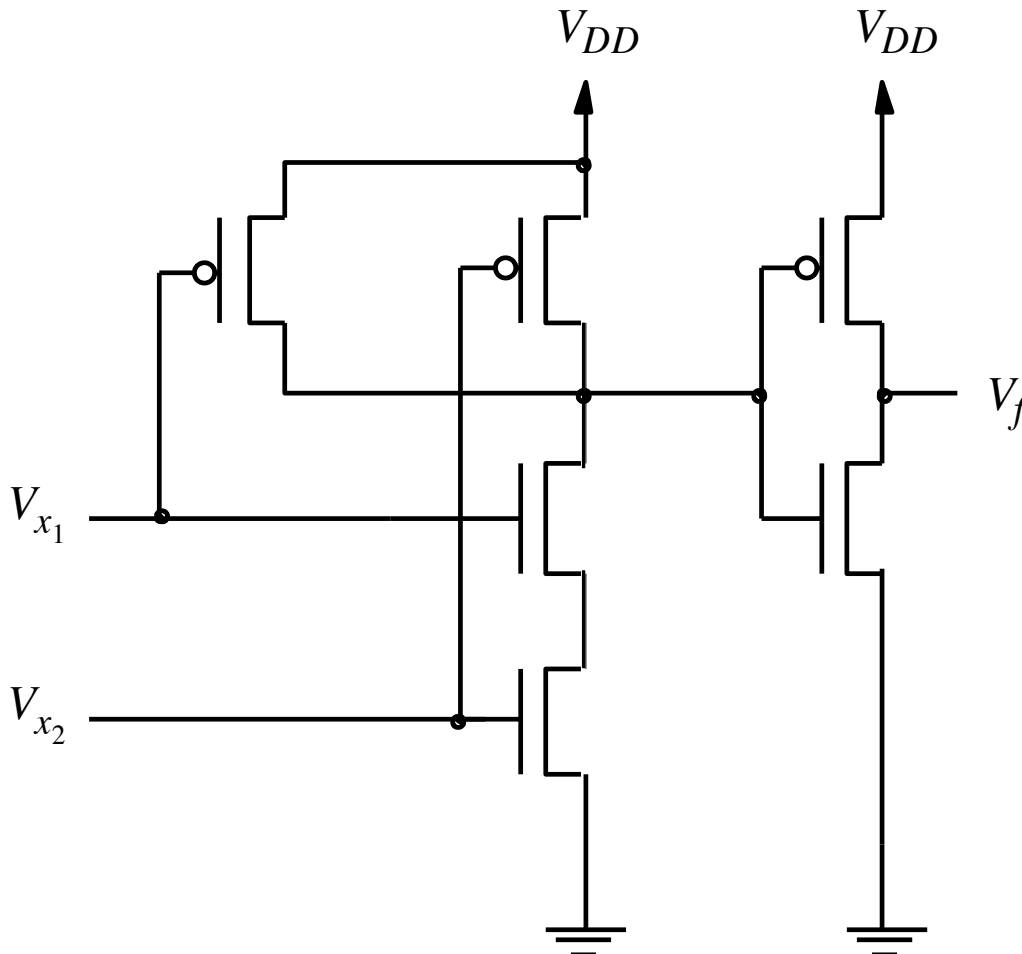


(a) Circuit

x_1	x_2	T_1	T_2	T_3	T_4	f
0	0	on	on	off	off	1
0	1	on	off	off	on	1
1	0	off	on	on	off	1
1	1	off	off	on	on	0

(b) Truth table and transistor states

CMOS AND Gate

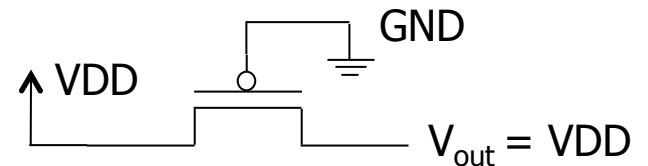
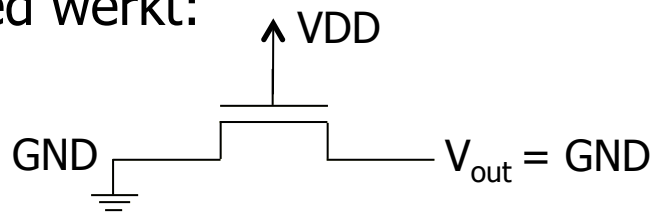


AND heeft 6
transistors !

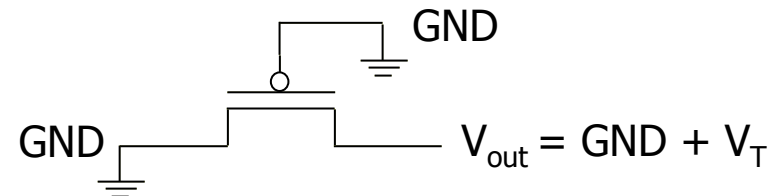
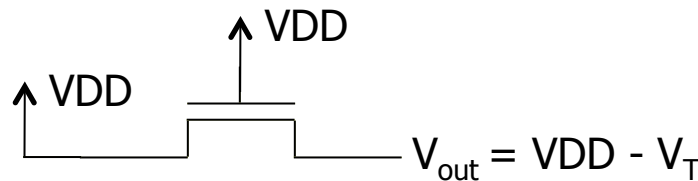
Een NAND heeft
slechts 4 transistors

Pass Transistors

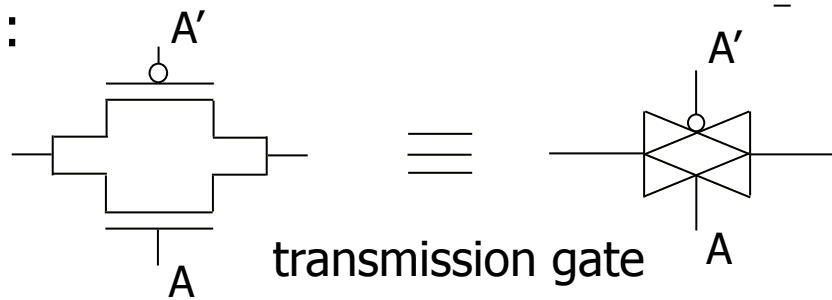
- NMOS en PMOS Transistors kunnen gebruikt worden als schakelaars om verbindingen te maken en 0 of 1 door te geven.
- Goed werkt:



- Maar minder goed werkt (V_T is threshold voltage voor "on" stand):

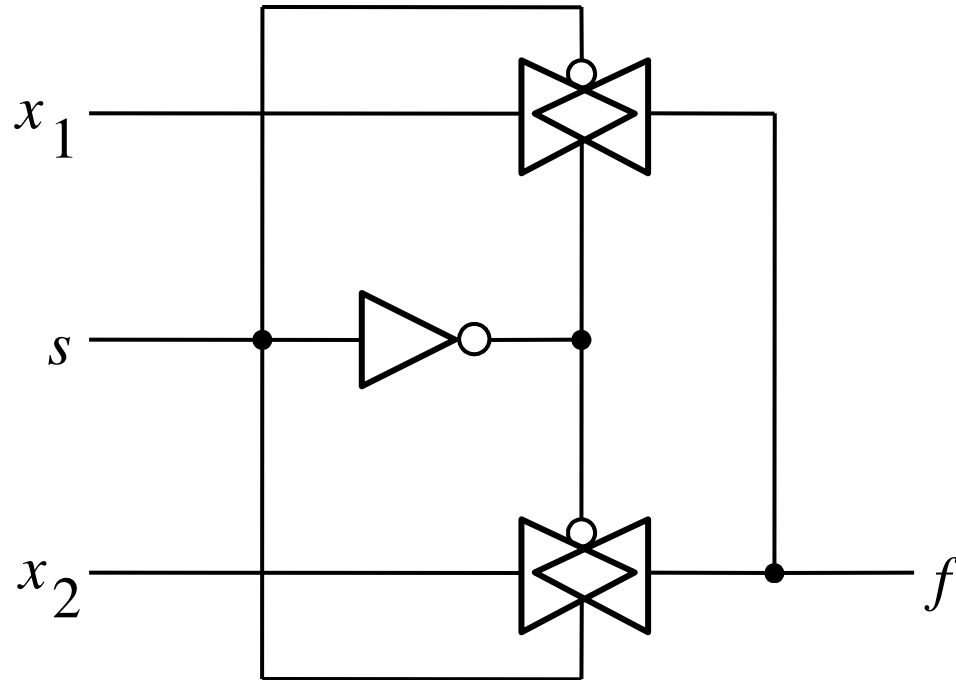


- Oplossing:



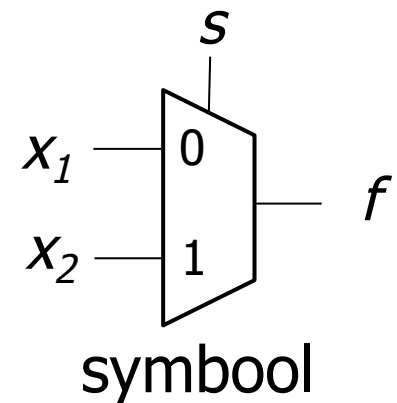
Wanneer $A = 1$:
zowel 0 als 1 worden
onverzwakt doorgelaten

Multiplexor (of selector)



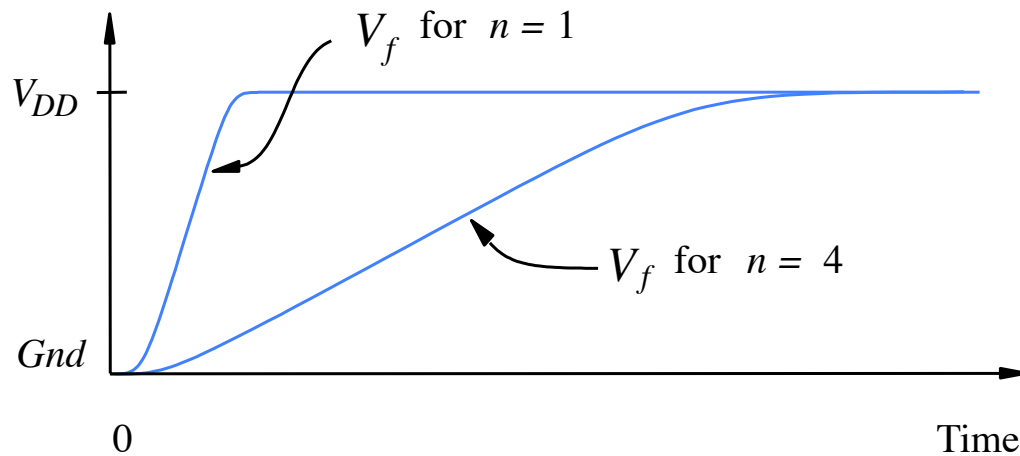
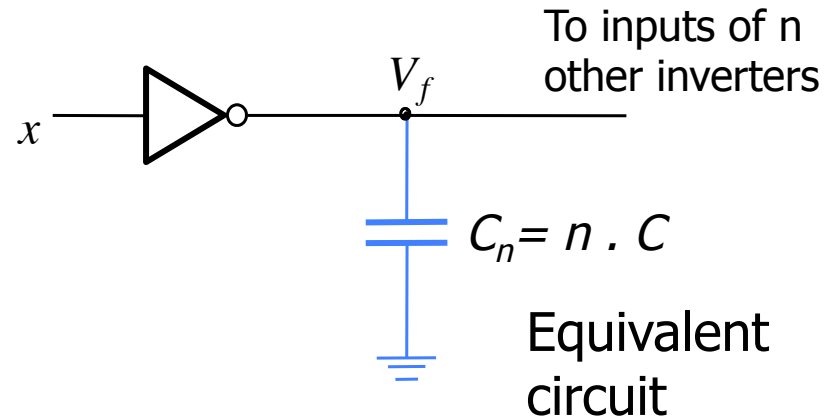
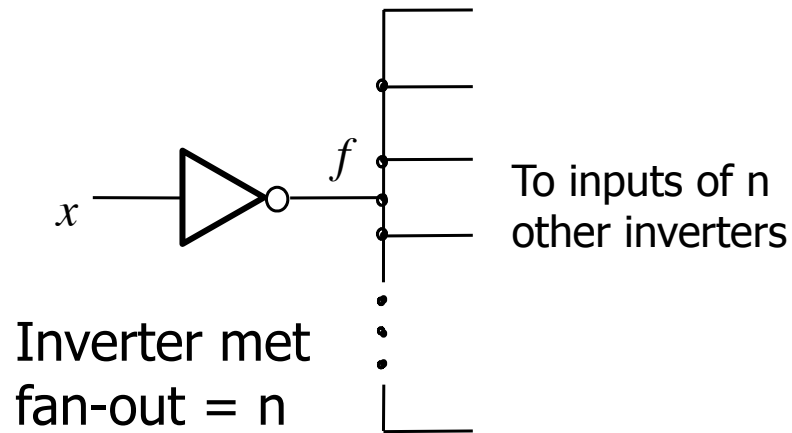
s	f
0	x_1
1	x_2

waarheidstabel



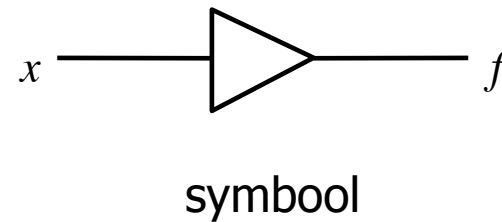
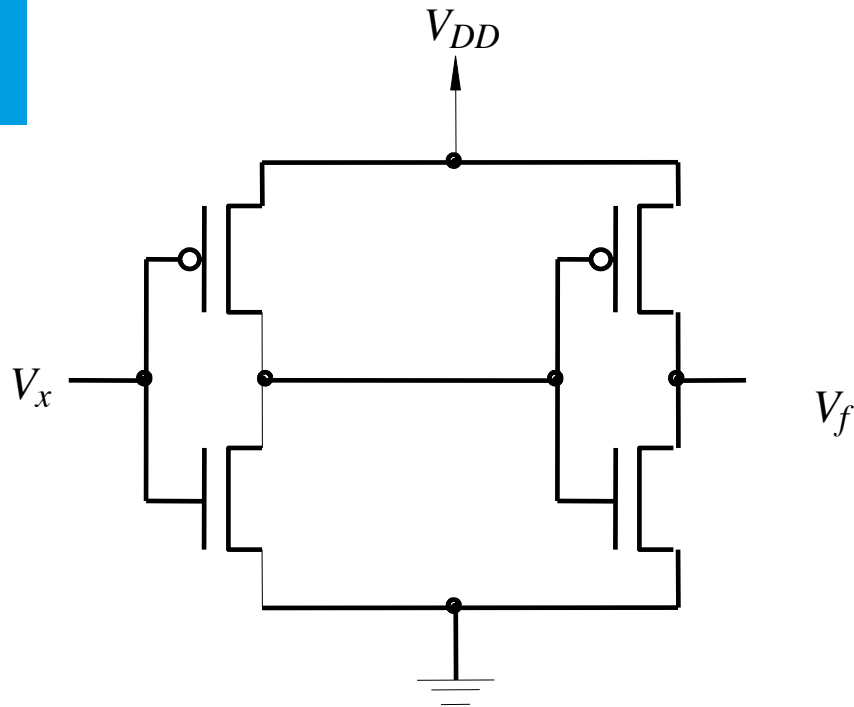
Een 2-to-1 multiplexer opgebouwd met transmission gates.

Fan-out



Vertragingstijd neemt toe bij hogere fan-out

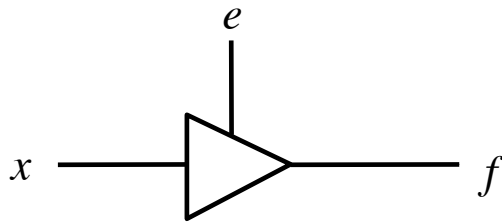
Buffers



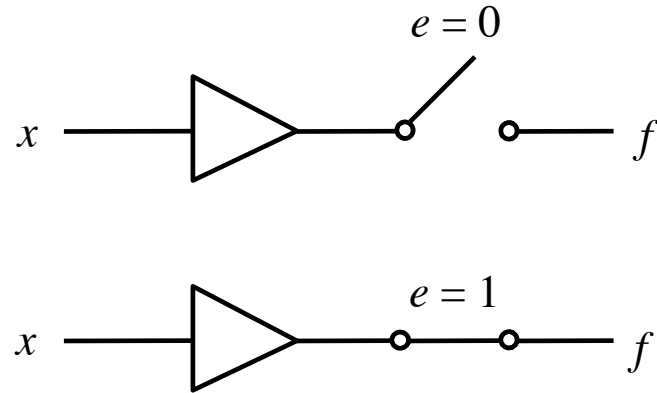
Niet inverterende buffer

Transistors van 2^e inverter zijn breed en kunnen veel stroom leveren (voor weinig vertragingstijd bij een hoge fan-out)

Tri-state buffer



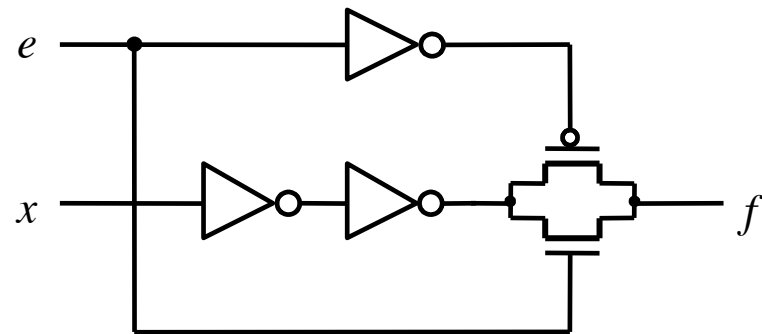
symbool



equivalent circuit

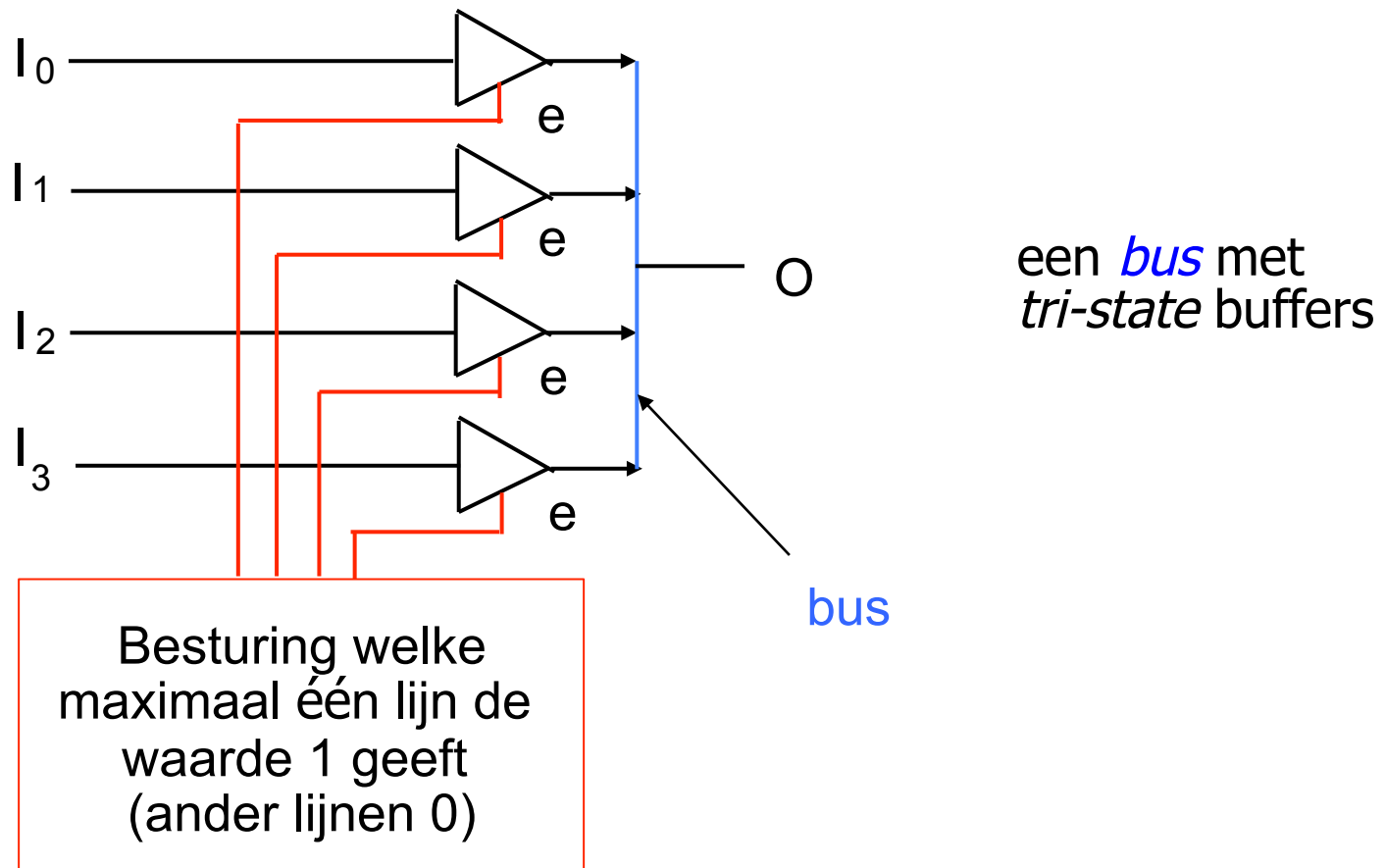
e	x	f
0	0	Z
0	1	Z
1	0	0
1	1	1

waarheidstabel
(Z = hoge impedantie)

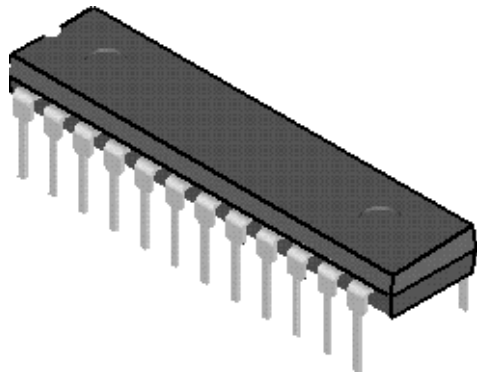


mogelijke implementatie

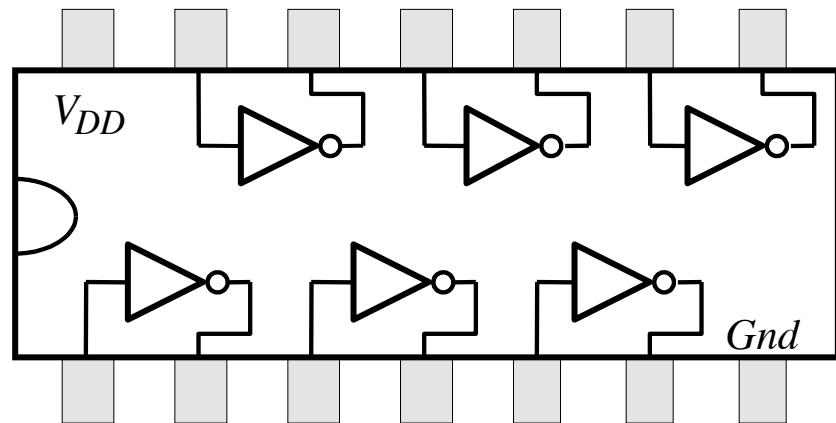
Voorbeeld tri-state buffer toepassing



Losse componenten: 7400 series



Dual-inline package



Structure of 7404 chip

VLSI Chips

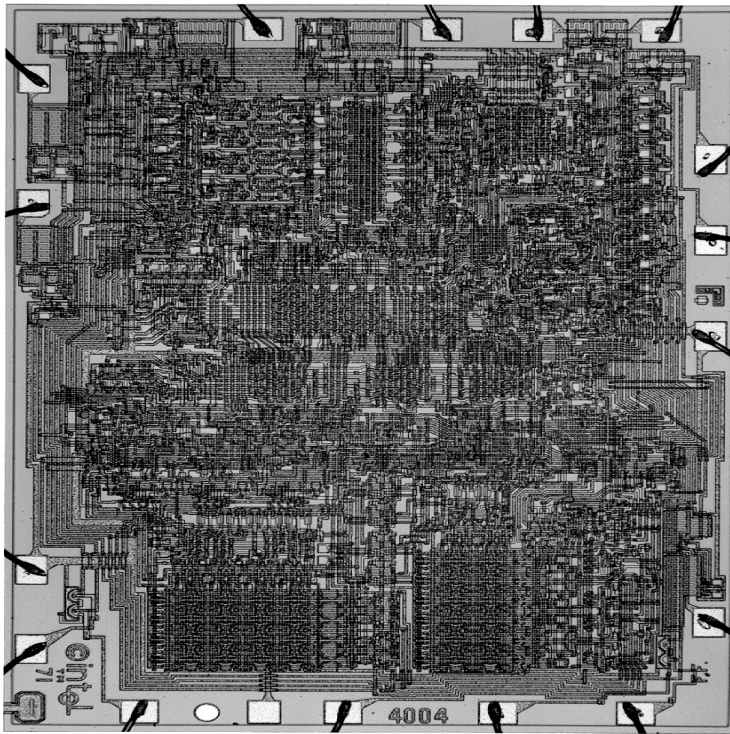
Schakelingen met miljoenen logische poorten



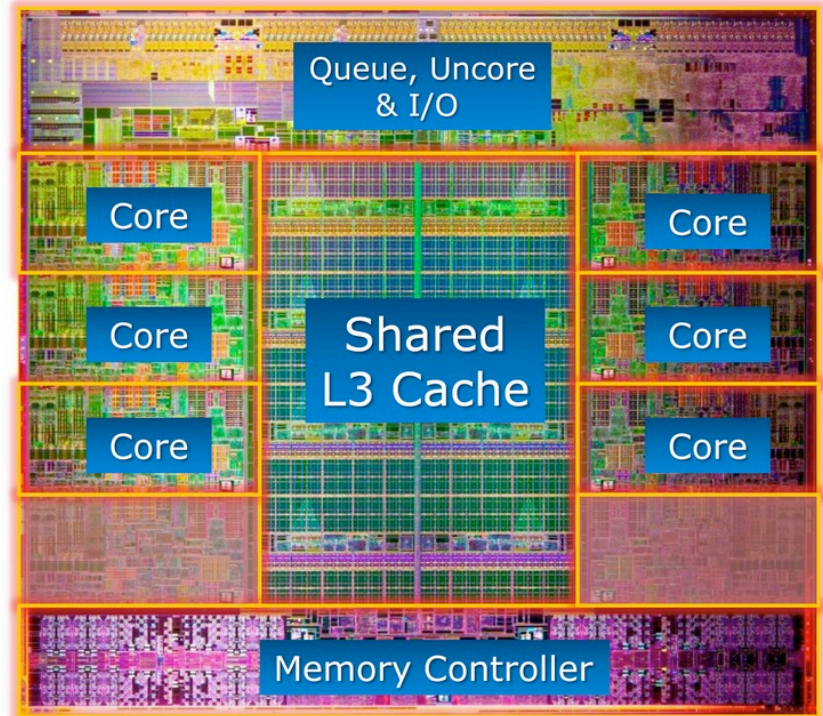
Bijv.

- General Purpose Processor
- Grafische Processor
- Digital Signal Processor
- Geheugen
- FPGA

VLSI Technologie Ontwikkeling



Intel 4004 processor 1971
trans. size 10 μm oppv. 12 mm^2
2.300 transistors 16 pins
740 kHz clock 4 bits
supply 15 V 1 Watt

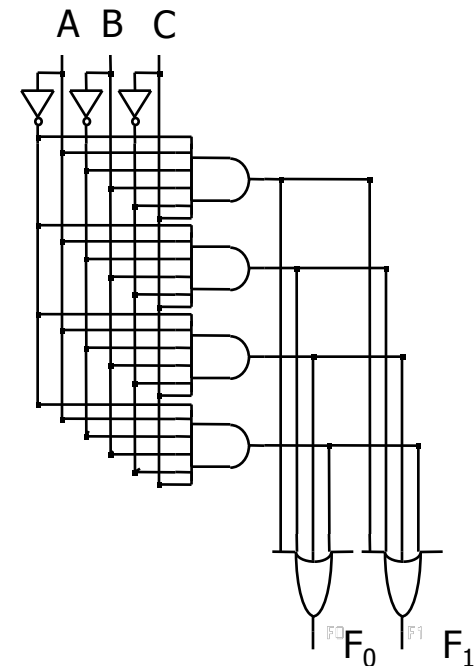
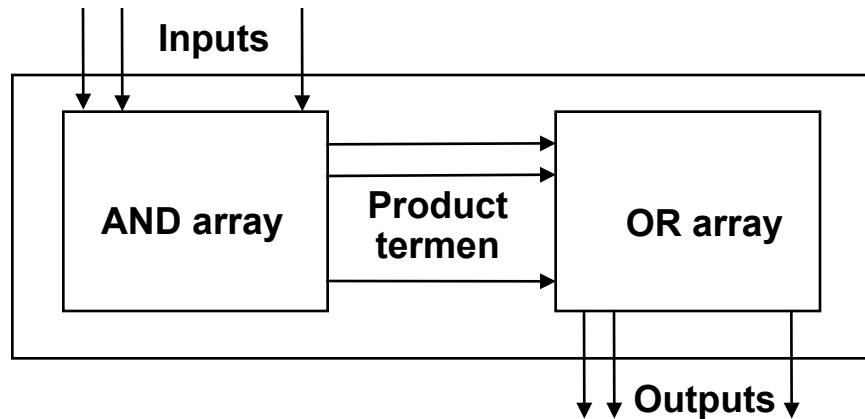


Intel Core i7-3960X processor 2011
trans. size 0.032 μm oppv. 435 mm^2
2,27 x 10⁹ transistors 2011 pins
3.3 GHz clock 64 bits
supply 1.1 V 130 Watt

Programmeerbare devices

PLA's en PAL's

Ge-prefabriceerde bouwblokken van vele AND/OR gate circuits die "personalized" worden door maken of verbreken van tussenverbindingen.

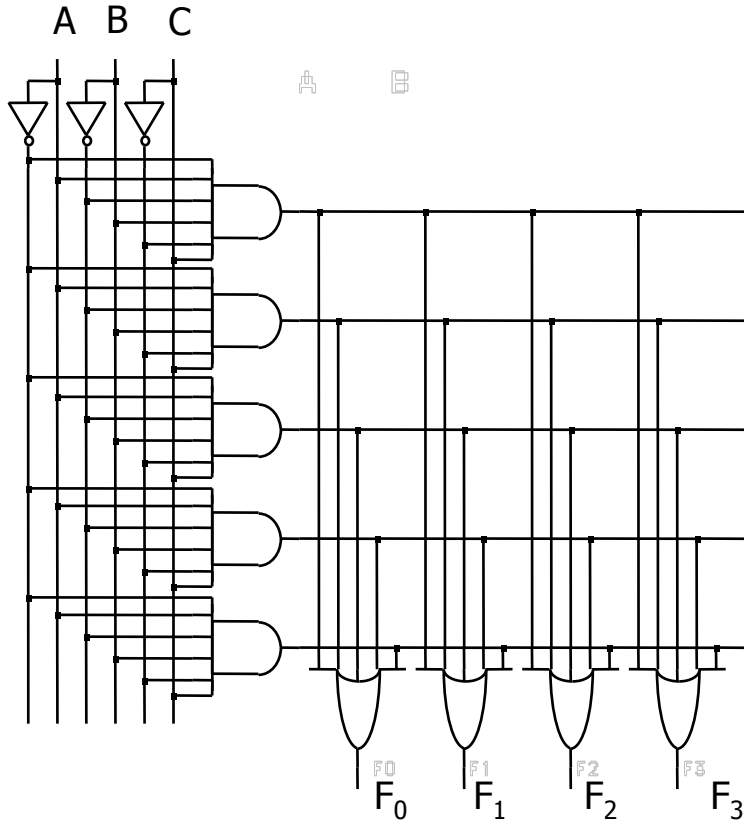


Programmable Logic Array (PLA):
zowel AND als OR arrays personalized ----->

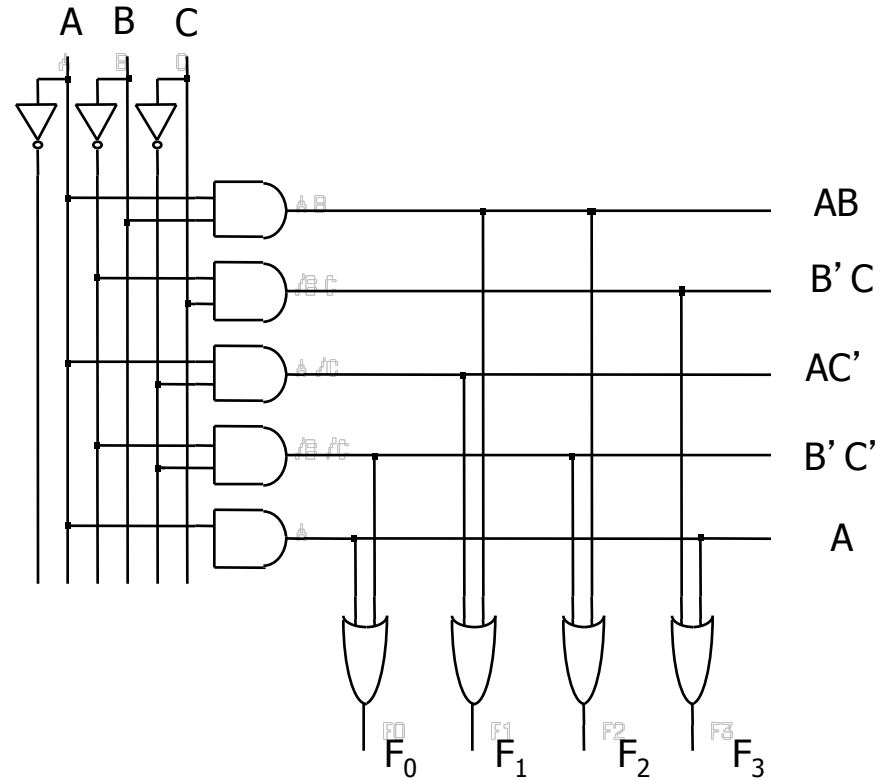
Programmable Array Logic (PAL):
alleen AND array personalized
OR array vast verbonden met subset producttermen

Programmeren PLA

Voor programmeren:



Na programmeren:

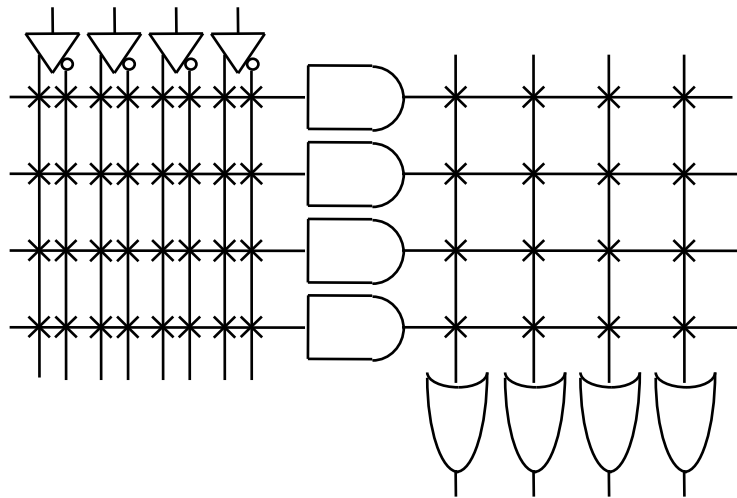


$$\begin{aligned}
 F_0 &= B'C' + A & F_2 &= AB + B'C' \\
 F_1 &= AB + AC' & F_3 &= BC' + A
 \end{aligned}$$

PLA vereenvoudigde notatie

Eenvoudigere notatie m.n. bedoeld voor zeer grote circuits

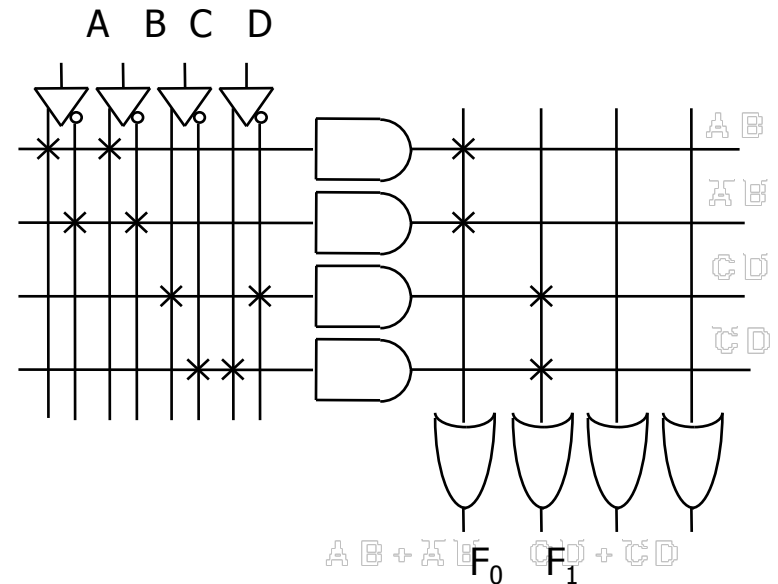
Alle invoerlijnen van ANDs en ORs worden door slechts een enkele lijn voorgesteld:



Bijvoorbeeld

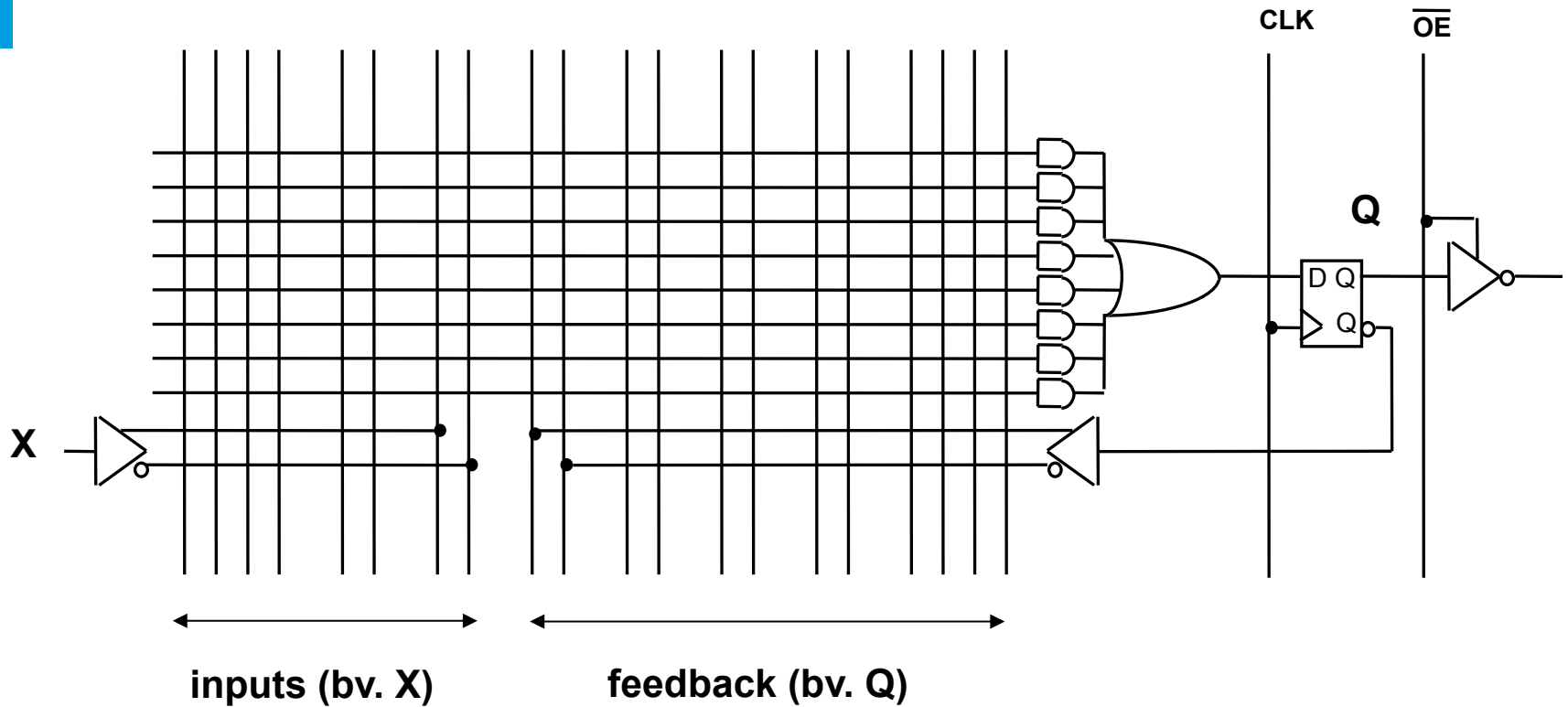
$$F_0 = A B + A' B'$$

$$F_1 = C D' + C' D$$



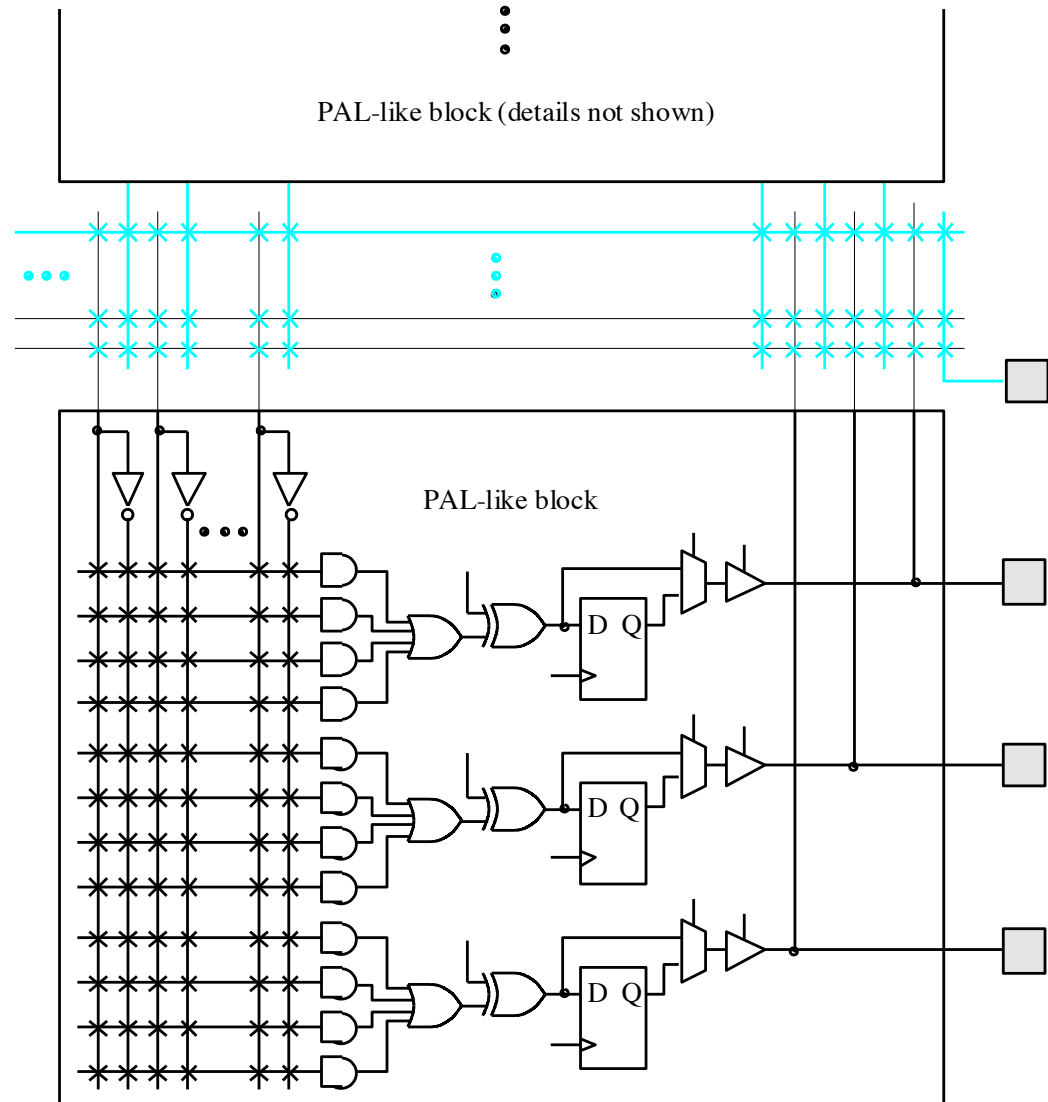
Registered PAL

PAL met registers (D flip flops)



Complex Programmable Logic Device (CLPD)

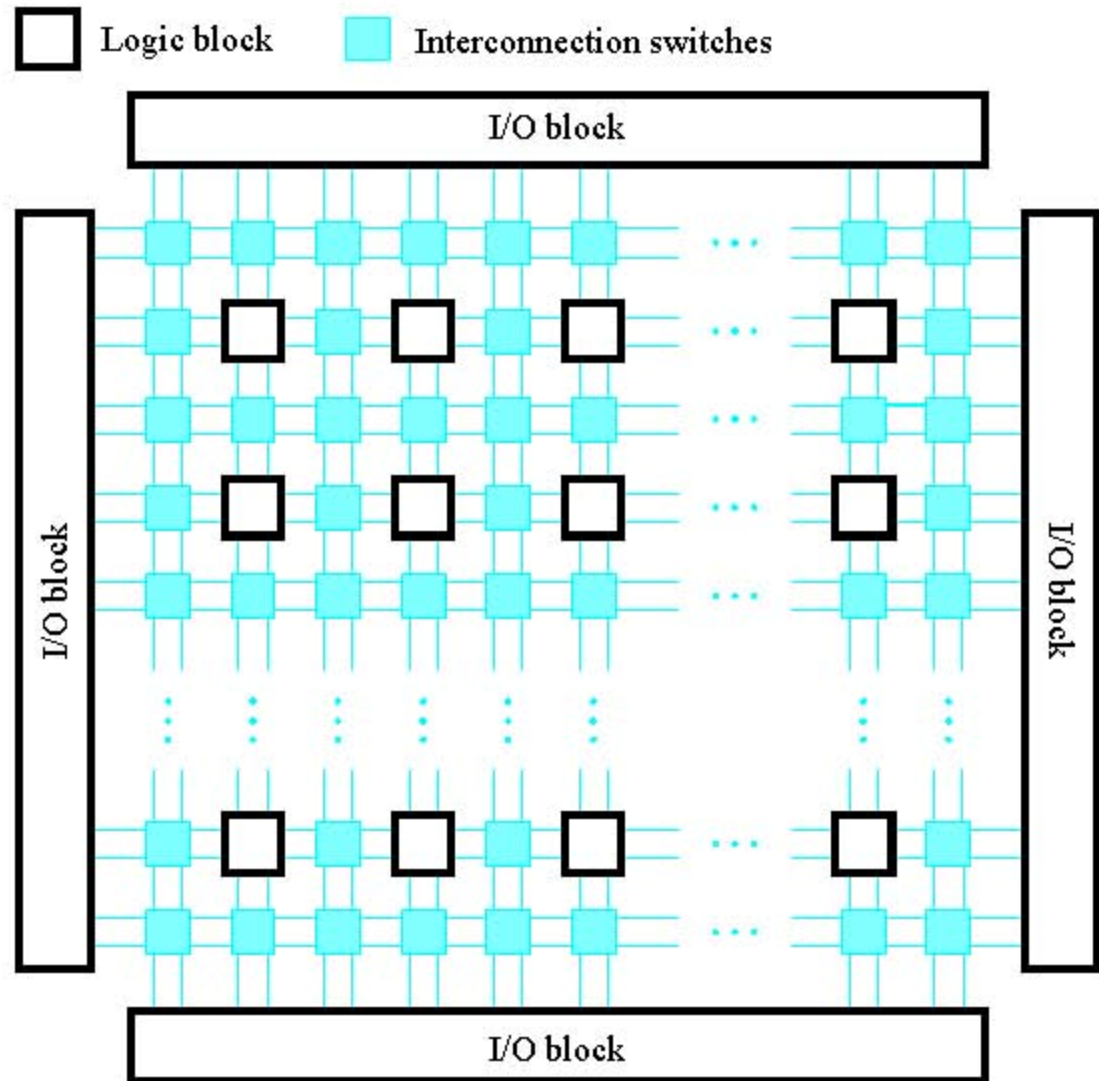
**Op 1 device (chip):
een aantal
registered PAL
blokken, onderling
verbonden d.m.v.
programmeerbare
interconnecties**



Field Programmable Gate Array (FPGA)

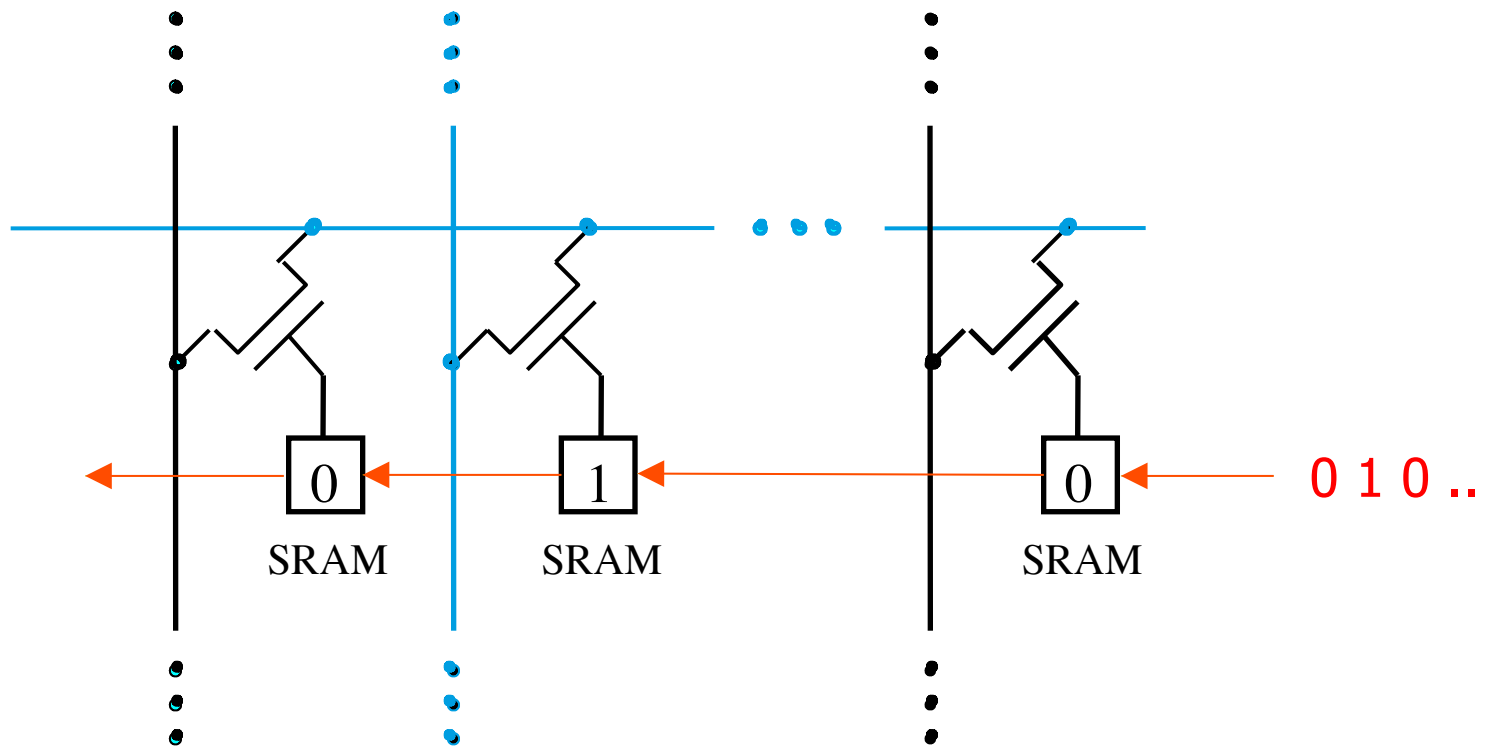
Een grote array van programmeerbare logic blocks, verbonden met programmeerbare interconnecties

Programmering van logic blocks en interconnecties met geheugencellen (welke geschakeld kunnen worden als een groot schuifregister)



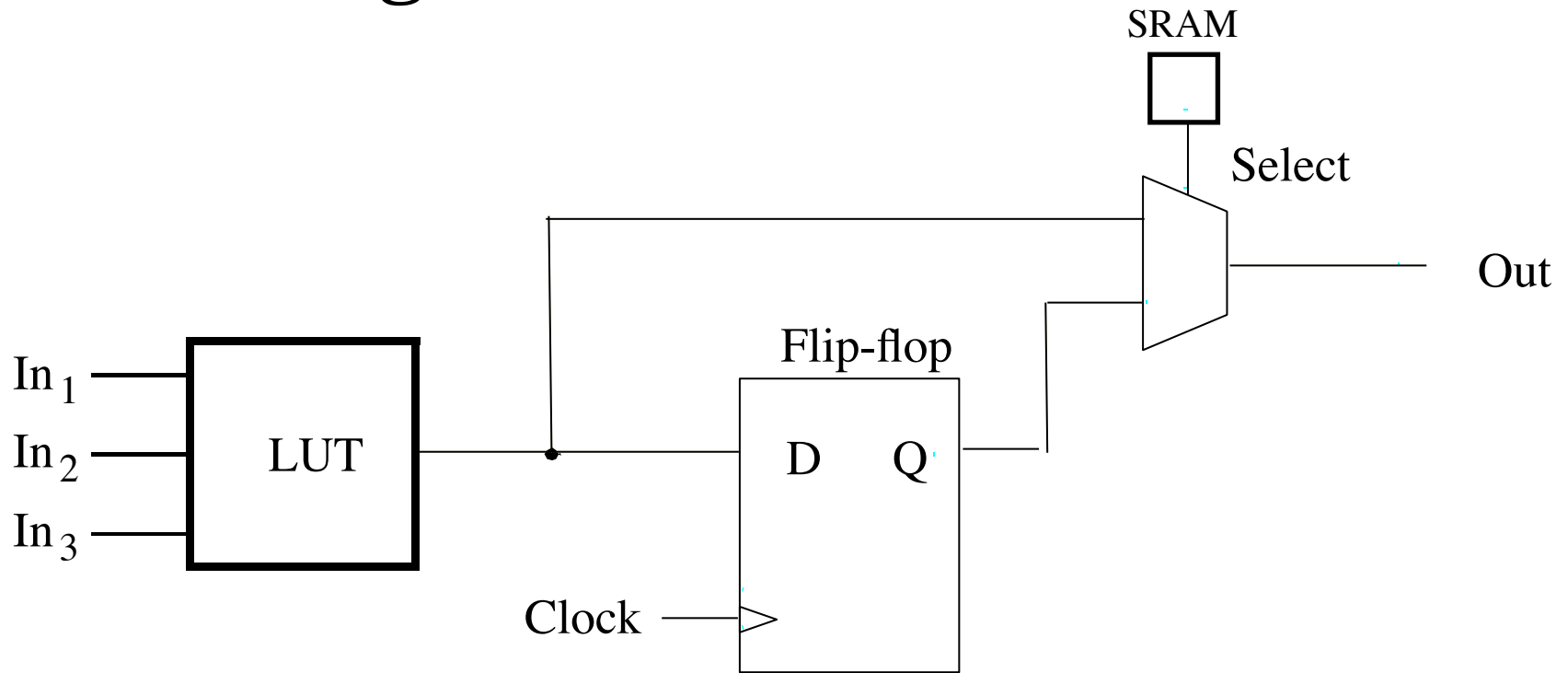
FPGA Interconnecties

Transistor switches worden aangestuurd door geheugencellen



Geheugencellen worden geprogrammeerd door ze te schakelen als schuifregister

FPGA Logic Block



LUT = Look Up Table

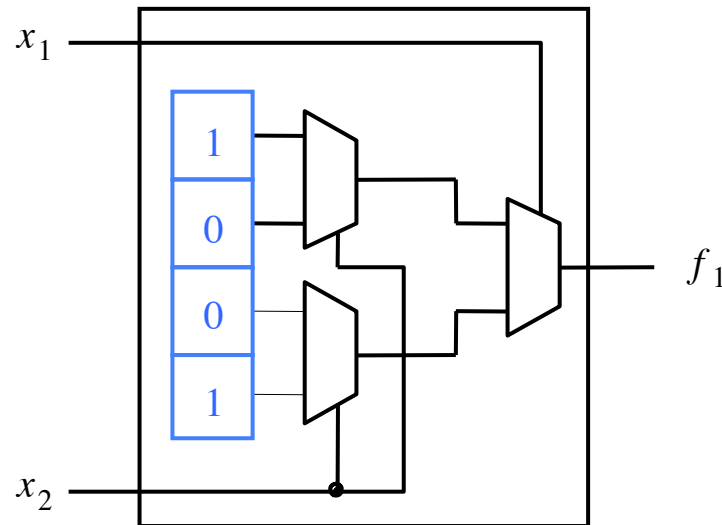
Waarheidstabel voor (in dit geval) 3 ingangen,
geprogrammeerd met geheugencellen

LUT Implementatie

Voorbeeld van geprogrammeerde LUT met 2 ingangen (x_1 en x_2) en 1 uitgang (f_1)

x_1	x_2	f_1
0	0	1
0	1	0
1	0	0
1	1	1

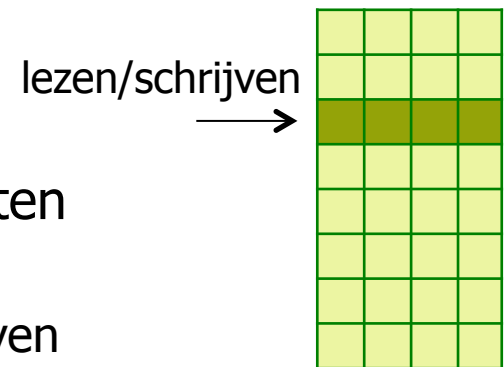
$$f_1 = \bar{x}_1\bar{x}_2 + x_1x_2$$



Geheugen

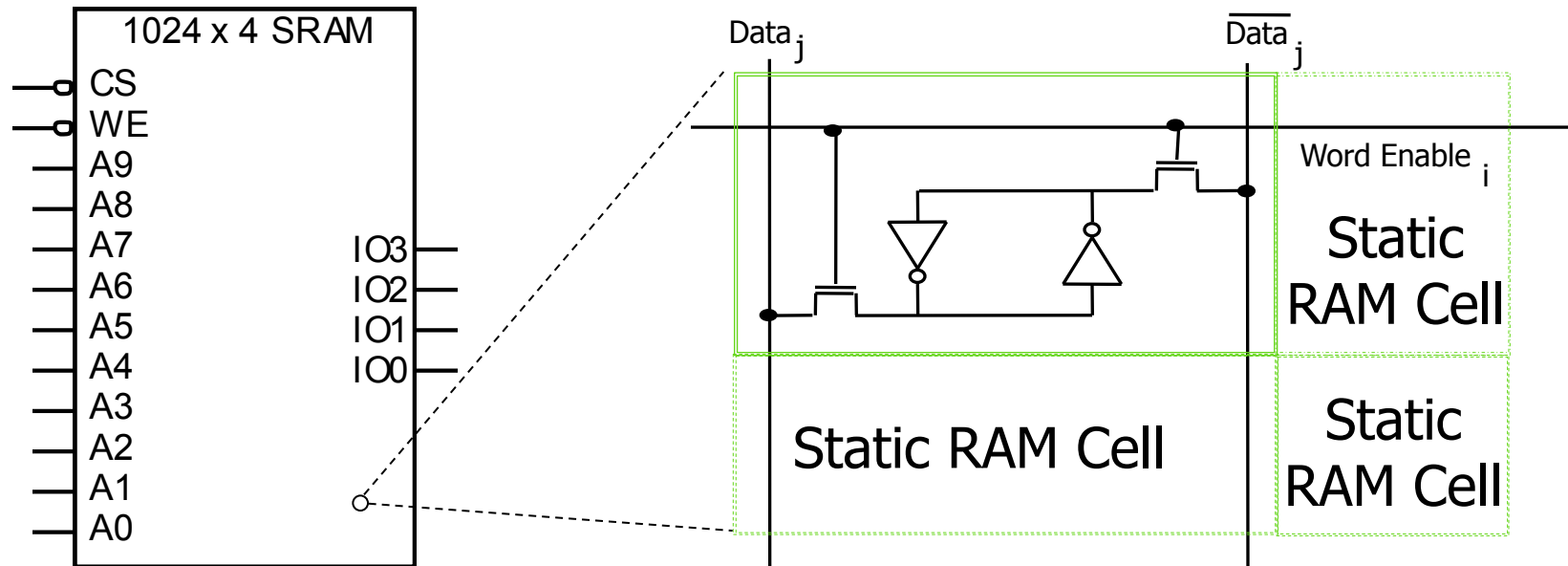
Geheugen

- Flip-flop ideaal als geheugencel voor implementatie van FSMs:
 - Inhoud van **alle** flipflops kan tegelijkertijd binnen 1 klokcyclus worden **gelezen** en **geschreven**
- Voor veel data zijn deze eisen echter niet nodig en kan compacter geheugen worden gebruikt:
 - S-RAM
 - D-RAM
 - EEPROM/Flash
- Deze geheugens bestaan uit een matrix van geheugenelementen waarbij geheugenelementen per rij (woord) te lezen of te schrijven
 - Binnen 1 klokcyclus alleen lezen of alleen schrijven



S-RAM: Static Random Access Memory

- Elk geheugenelement omvat slechts circa 6 transistoren.
- “Static”: informatie in RAM cel hoeft niet periodiek ververs te worden.



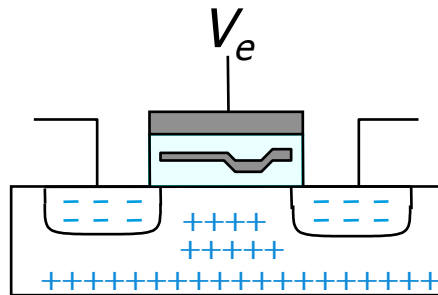
Chip Select Line (CS: 0 = enable gehele chip)

Write Enable Line (WE: 1 = READ, 0 = WRITE)

10 Adreslijnen, 4 Bidirectionele Datalijnen, voor 2^{10} woorden van 4 bits

EEPROM/Flash

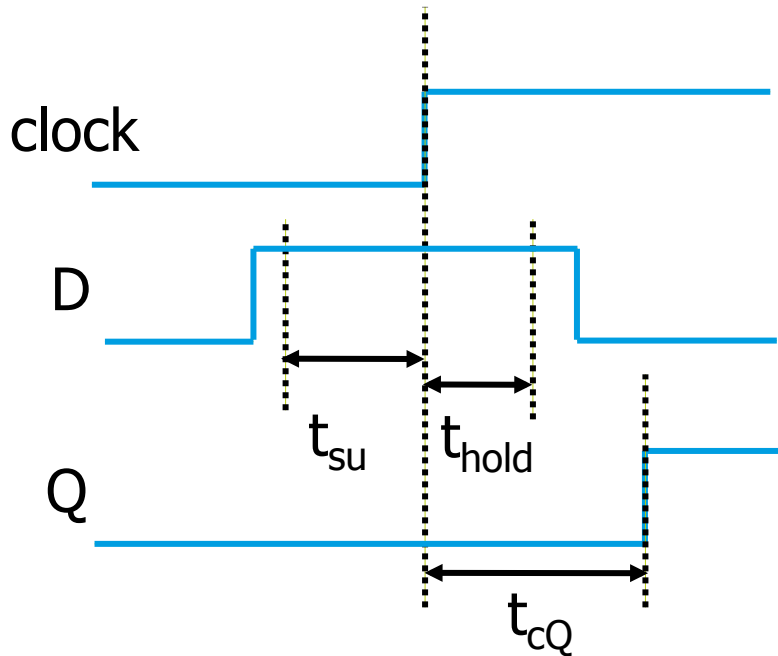
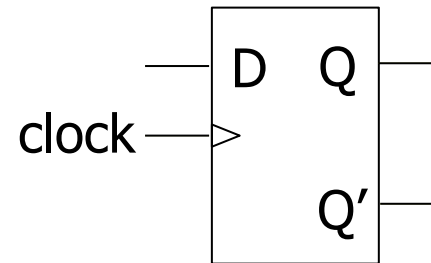
- EEPROM = Electrically Erasable Programmable Read-Only Memory
- Geheugencel bevat EEPROM transistor met twee gates: één gate is "floating".



- Transistor werkt normaal als schakelaar met V_e als gate.
- Wanneer V_e extra hoge spanning krijgt, wordt via tunneling een permanente negatieve lading opslagen op de floating gate.
- Hierdoor is de transistor permanent uit.
- Inhoud geheugen blijft bewaard ook zonder voedingspanning.
- Lezen gaat vrij snel, maar schrijven gaat traag.
- Flash is een geavanceerde/moderne versie van EEPROM.

Meer over flip-flops

Flip-flop timing

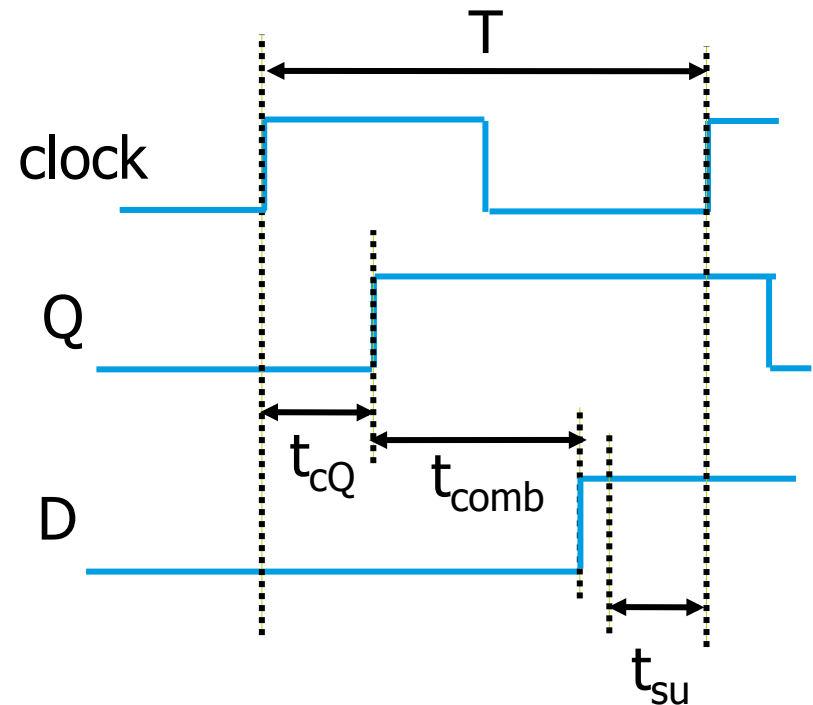
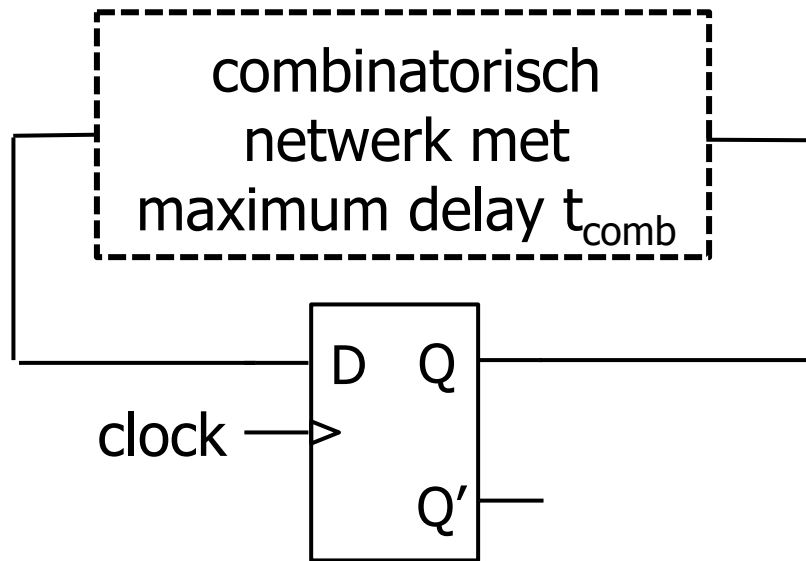


t_{su} = set-up tijd
minimale tijd dat D stabiel moet
zijn voor actieve klokflank

t_{hold} = hold tijd
minimale tijd dat D stabiel moet
zijn na actieve klokflank

t_{ccQ} = vertragingstijd tussen
actieve klokflank en uitgang Q

Maximum clock frequentie



Voor klokperiode T moet gelden $T > t_{cQ} + t_{comb} + t_{su}$

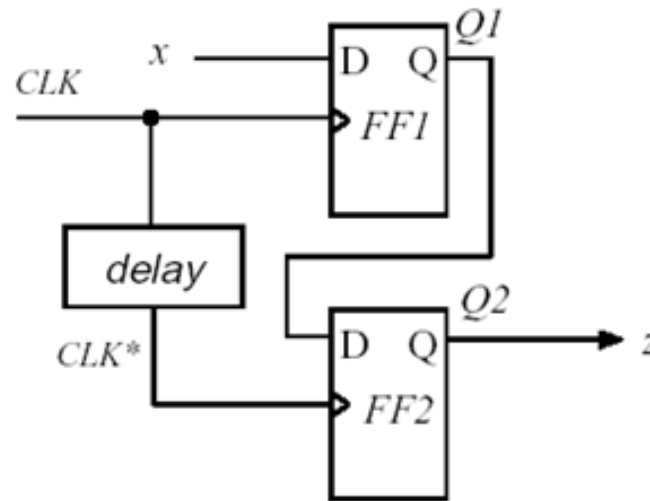
Dus maximale frequentie $F_{max} = 1/T_{min} = 1/(t_{cQ} + t_{comb} + t_{su})$

Bijvoorbeeld: $F_{max} = 1/(1.0 + 5.2 + 0.6 \text{ ns}) = 147 \text{ MHz}$

Verder moet $t_{cQ} + t_{comb} > t_{hold}$ maar dit is normaal geen probleem

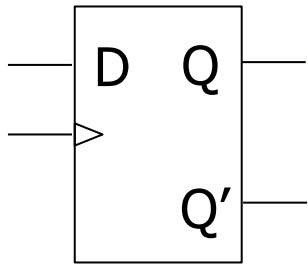
Clock skew

- Door vertraging in de kloklijn leest ingang D van FF2 niet $Q1(t)$ maar $Q1(t+1)$, omdat actieve klokflank voor FF2 komt na verandering in $Q1$ (hold tijd overtreding!)



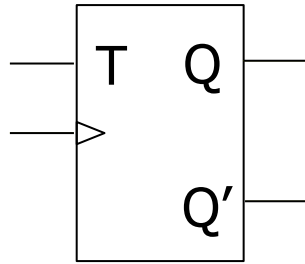
- Probleem kan spelen bij grote kloknetwerken of wanneer er logische schakelingen in het kloknetwerk zitten.

D, T en JK Flip-flop



D	Q ⁺
0	0
1	1

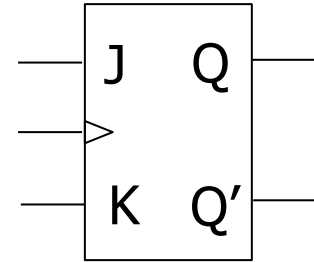
$$Q^+ = D$$



T	Q ⁺
0	Q
1	Q'

$$Q^+ = T \cdot Q' + T' \cdot Q$$

$$= T \oplus Q$$

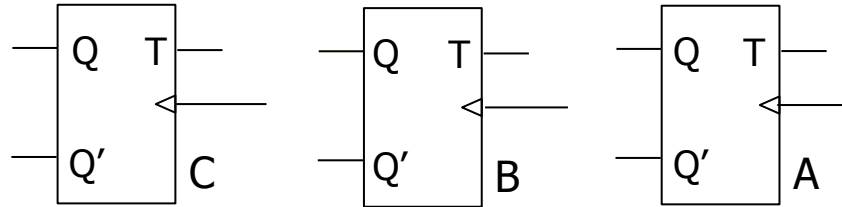
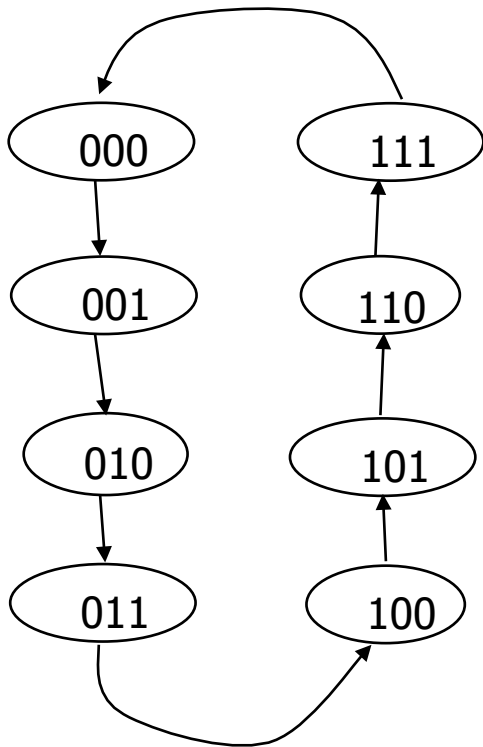


J	K	Q ⁺
0	0	Q
0	1	0
1	0	1
1	1	Q'

$$Q^+ = J \cdot Q' + K' \cdot Q$$

Ontwerpen met T Flip-flops: teller

Stel we willen de teller implementeren met T Flip-flops:



Dan moeten we de juiste aansturing voor T vinden (next-state functie re-mappen)

Re-mapping tabel voor T Flip-flop:

Q	Q ⁺	
	0	1
0	T=0	T=1
1	T=1	T=0

Present State			Next State			T Flip-flop Inputs		
Q _C	Q _B	Q _A	Q _C ⁺	Q _B ⁺	Q _A ⁺	T _C	T _B	T _A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Implementatie teller met T flip-flops

		C			
	CB	00	01	11	10
A	0	1	1	1	1
	1	1	1	1	1
		B			

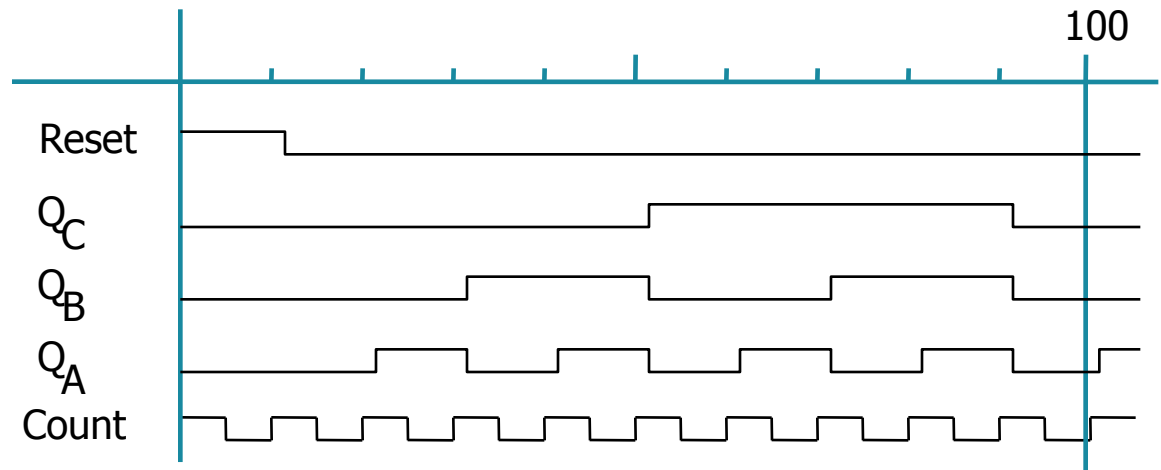
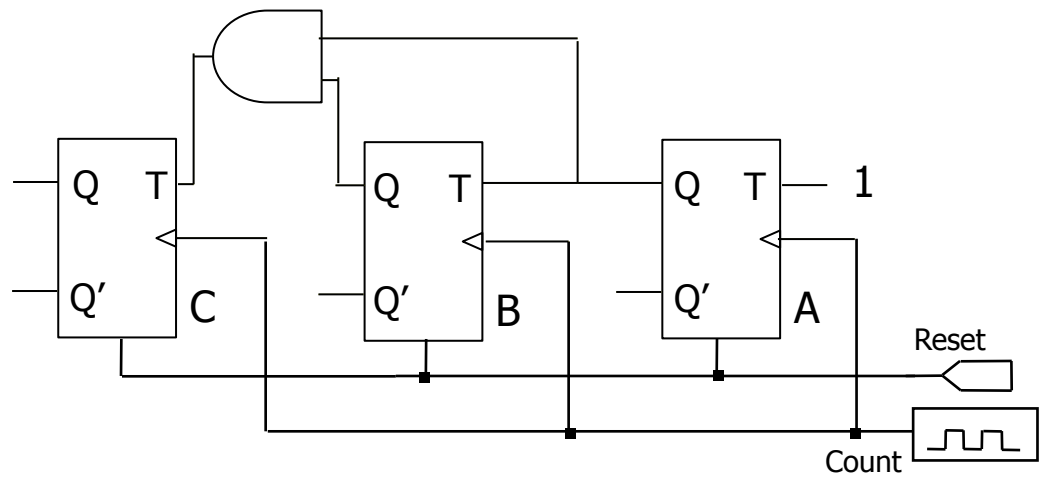
$T_A = 1$

		C			
	CB	00	01	11	10
A	0	0	0	0	0
	1	1	1	1	1
		B			

$T_B = A$

		C			
	CB	00	01	11	10
A	0	0	0	0	0
	1	0	1	1	0
		B			

$T_C = B A$



Samenvatting

- CMOS technologie
- Programmeerbare devices
- Geheugen
- Timing in sequentiële netwerken
- T en JK Flip-flops

Volgende college: [VHDL](#)