

CS

WEEK 1-2

AGENDA

week	onderwerp	P&H	AT	Dijkstra
1	coderingen en talstelsels representatie van getallen optellen en aftrekken vermenigvuldigen en delen logische poorten schakelingen met poorten geheugen-elementen systeemklok & timers	App. B2, B3, B7, B8, B9 2.4 3.2 t/m 3.5 app B	App. A, B 3.1, 3.2, 3.3	H1 H2
2	typen computers 8 great ideas organisatie van de computer CPU intern, instructies uitvoeren geheugen systeem adres- en databus byte ordering pipelining de AVR MCU	1.1 t/m 1.4 2.12 4.1 t/m 4.5	1.3 2.1, 2.2 3.7	H3 6.1 en 6.2 7.1 en 7.2
3	typen geheugen caching opslag (ssd, harddisk) translating and starting a program parallele architecturen - h/w multi-threading - multicore - GPU	5.2, 5.3 6.4 t/m 6.6	2.2, 2.3 7.3, 7.4 H8	4.1 7.3

AGENDA

- **ASCII en Unicode**
- transistors
- logische poorten
- schakelingen met poorten
- half en full adder
- geheugen elementen
- systeemklok en timers

ASCII

- American Standard Code for Information Interchange
- codering van karakters
- 7 bits code
 - vb: "A" : 0x41
 - vb: "a" : 0x61
- maar wordt meestal als byte opgeslagen
- 128 tekens
 - cijfers
 - letters (A en a)
 - control characters

ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

UNICODE

- ondersteunt alle schriften (zoals Latijns alfabet en 25.000 Chinese karakters en ...)
- ruim 1,1 miljoen tekens
- maakt wereldwijde uitwisseling van karakters mogelijk
- hoe coderen in een stroom van bytes ?
- Unicode Consortium: <http://www.unicode.org>

UTF-8

- 'code units' zijn 8 bit breed
- unicode teken wordt gecodeerd als stroom van bytes
 - 1 .. 4 bytes (variabel !)
- legal values: 0x0000 - 0x10FFFF
 - dus 17 sets van 64k blokken
- veel Westerse letters nemen slechts 1 byte in beslag
- is standaard codering voor XML

UTF-8

- UTF-8 : code units are 8 bit
- msb = 0 : 1-byte sequence
- msb = 110 : 2-byte sequence
- msb = 1110 : 3 byte sequence
- msb = 11110 : 4 byte sequence

Unicode range	Byte1	Byte2	Byte3	Byte4
0000-007F	0xxxxxxx			
0080-07FF	110xxxxx	10xxxxxx		
0800-FFFF	1110xxxx	10xxxxxx	10xxxxxx	
10000-10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

UTF-8

- hoe weet de ontvanger uit hoeveel bytes het karakter bestaat ? A : eerste byte!
- Stel: UTF-8-code begint met : 0xF0
- Q: Hoe lang is de gehele UTF-8 code?
- 4!
- Q: 0xF0 9F 98 82, welke Unicode? Welk symbool? 🤔

Unicode range	Byte1	Byte2	Byte3	Byte4
0000-007F	0xxxxxxx			
0080-07FF	110xxxxx	10xxxxxx		
0800-FFFF	1110xxxx	10xxxxxx	10xxxxxx	
10000-10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

VOORBEELD UTF-8

- the Cent Sign (¢) is Unicode 0x00A2 (1010 0010)
<http://www.unicode.org/charts/charindex.html>

Unicode range	Byte1	Byte2	Byte3	Byte4
0000-007F	0xxxxxxx			
0080-07FF	110xxxxx	10xxxxxx		
0800-FFFF	1110xxxx	10xxxxxx	10xxxxxx	
10000-10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

- so it will be encoded using two bytes,
110xxxxx,10xxxxxx.
- putting the binary right-justified into the 'x' bits results
in 11000010,10100010
- this is hex 0xC2,0xA2, which is the UTF-8 encoding

0080

C1 Controls and Latin-1 Supplement

00FF

	008	009	00A	00B	00C	00D	00E	00F
0	XXX 0080	DCS 0090	NB SP 00A0	◊ 00B0	À 00C0	Ð 00D0	à 00E0	ð 00F0
1	XXX 0081	PU1 0091	¡ 00A1	± 00B1	Á 00C1	Ñ 00D1	á 00E1	ñ 00F1
2	BPH 0082	PU2 0092	¢ 00A2	² 00B2	Â 00C2	Ò 00D2	â 00E2	ò 00F2
3	NBH 0083	STS 0093	£ 00A3	³ 00B3	Ã 00C3	Ó 00D3	ã 00E3	ó 00F3
4	IND 0084	CCH 0094	¤ 00A4	´ 00B4	Ä 00C4	Ô 00D4	ä 00E4	ô 00F4
5	NEL 0085	MW 0095	¥ 00A5	µ 00B5	Å 00C5	Õ 00D5	å 00E5	õ 00F5
6	SSA 0086	SPA 0096	¦ 00A6	¶ 00B6	Æ 00C6	Ö 00D6	æ 00E6	ö 00F6
7	ESA 0087	EPA 0097	§ 00A7	· 00B7	Ç 00C7	× 00D7	ç 00E7	÷ 00F7
8	HTS 0088	SOS 0098	¨ 00A8	 00B8	È 00C8	Ø 00D8	è 00E8	ø 00F8

MEER UITLEG OVER UNICODE

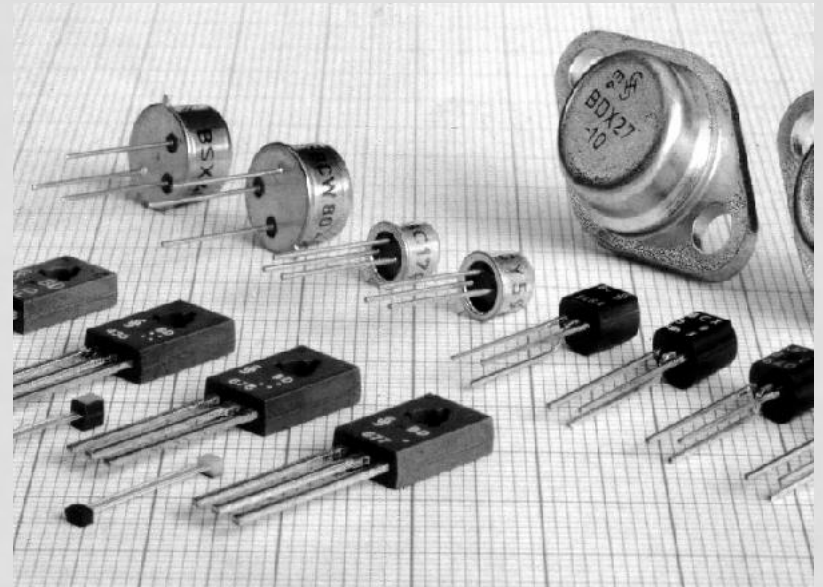
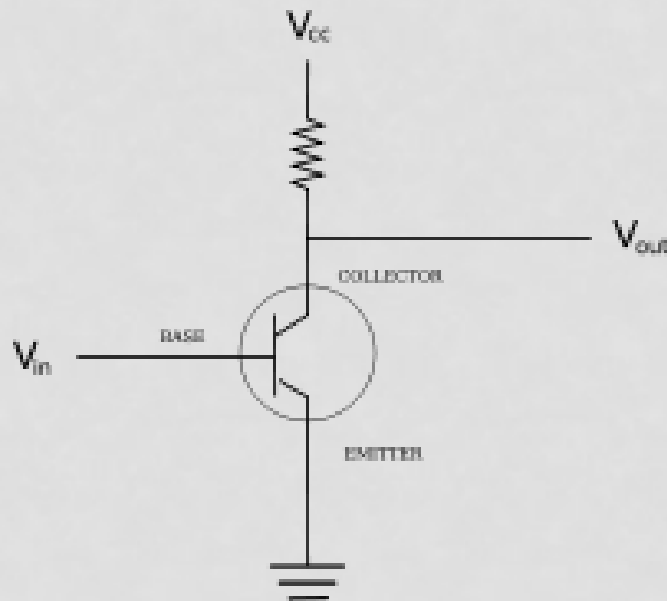
- <http://www.unicode.org/notes/tn23/Muller-Slides.pdf>

AGENDA

- ASCII en Unicode
- **transistors**
- logische poorten
- schakelingen met poorten
- half en full adder
- geheugen elementen
- systeemklok en timers

TRANSISTORS

- 1925 : Lilienfeld (Canada) eerste beschrijving werking field-effect transistor (FET)
- 1947 : AT&T's Bell Labs (Bardeen, Brattain en Shockley) eerste germanium transistor
- 1954 : TI (Teal) eerste silicium transistor
- 1960 : Bell Labs (Kahng and Atalla) eerste MOS transistor



TRANSISTORS

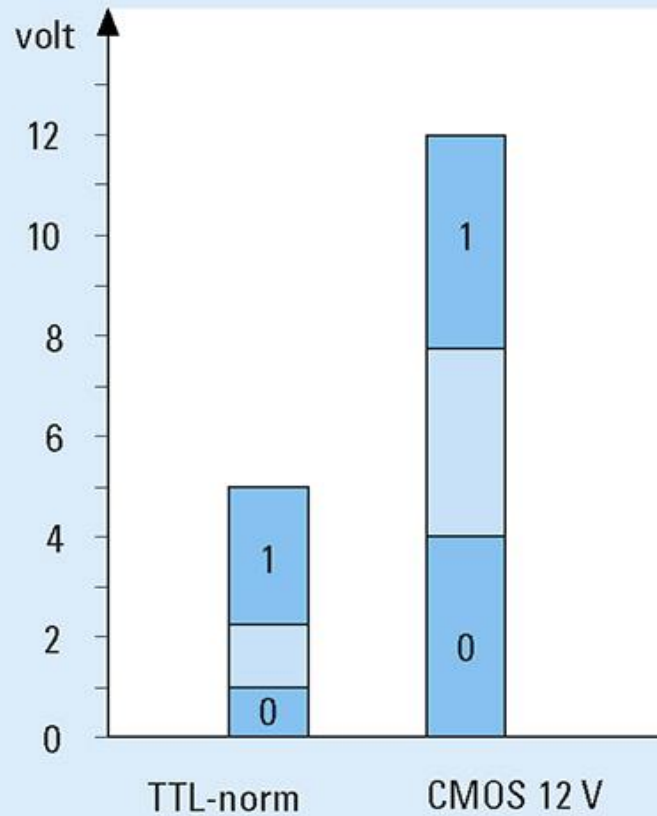
- computers zijn opgebouwd uit digitale elektronische circuits : "poorten" of "gates"
- poorten zijn gemaakt met transistors
- transistor : versterker en schakelaar
- 2 families : CMOS en TTL (74xx)

TRANSISTORS

- poorten werken met 2 niveaus "0" en "1"
- poorten implementeren logische functies
- wat is "0" en wat is "1" ?

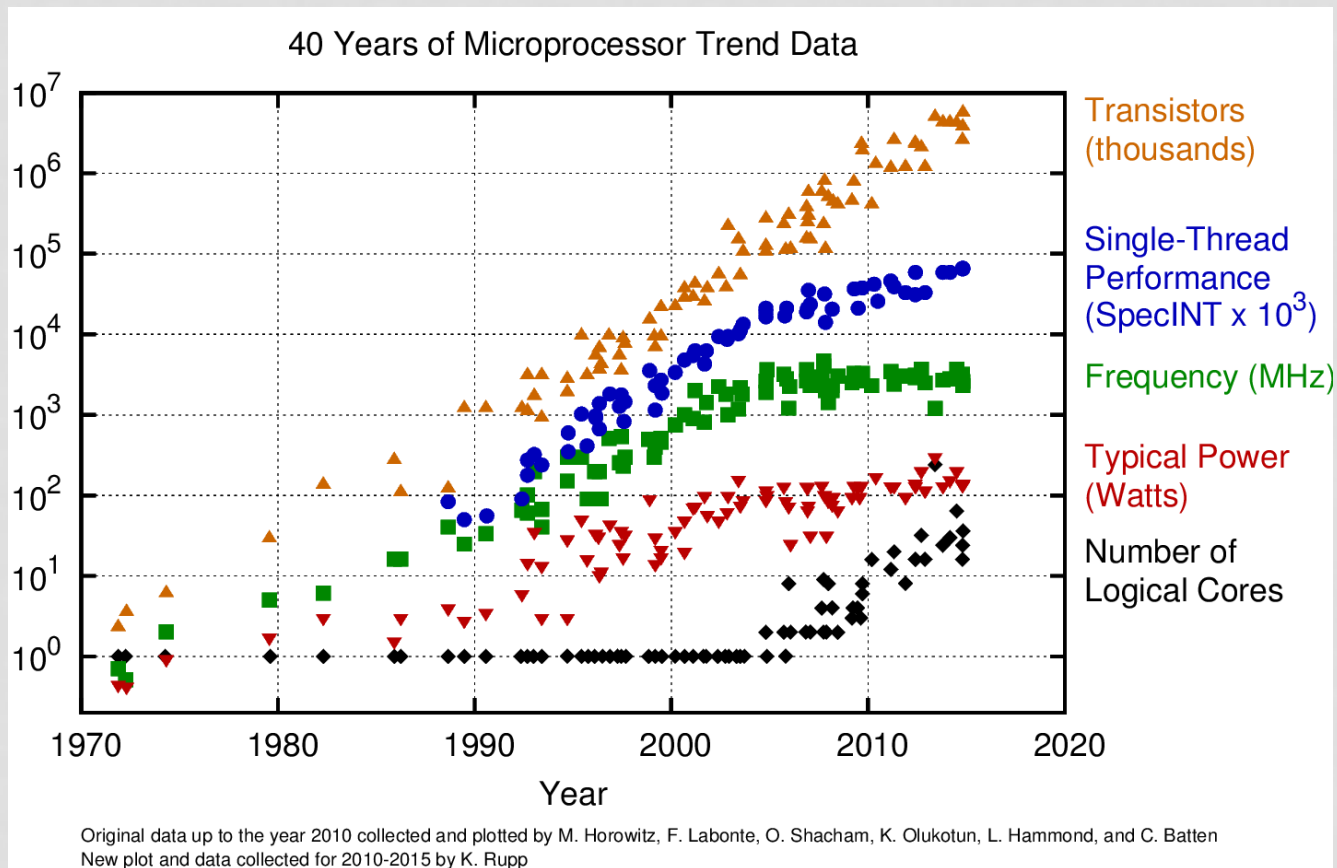
"0" EN "1" NIVEAUS

Figuur 2.3 **Verschillende afspraken**

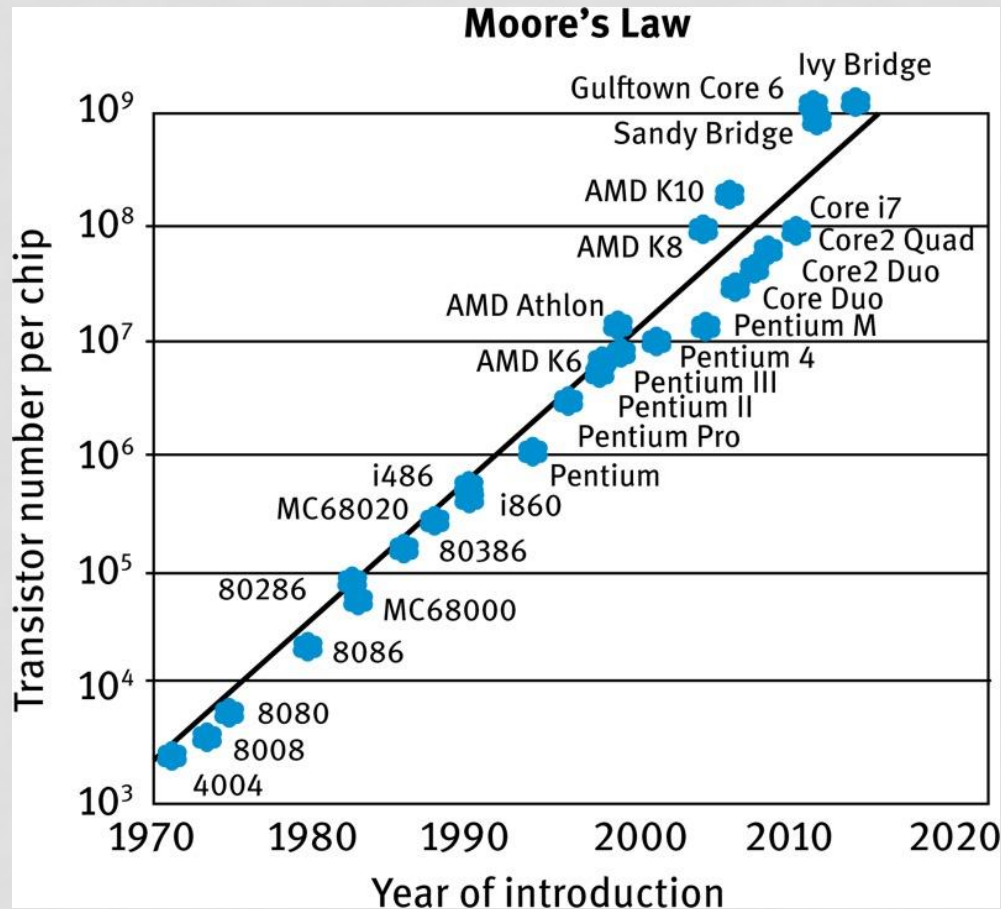


WET VAN MOORE

- Moore (Intel, 1965) : elke 1,5 jaar **verdubbeling** van het aantal transistors op een chip

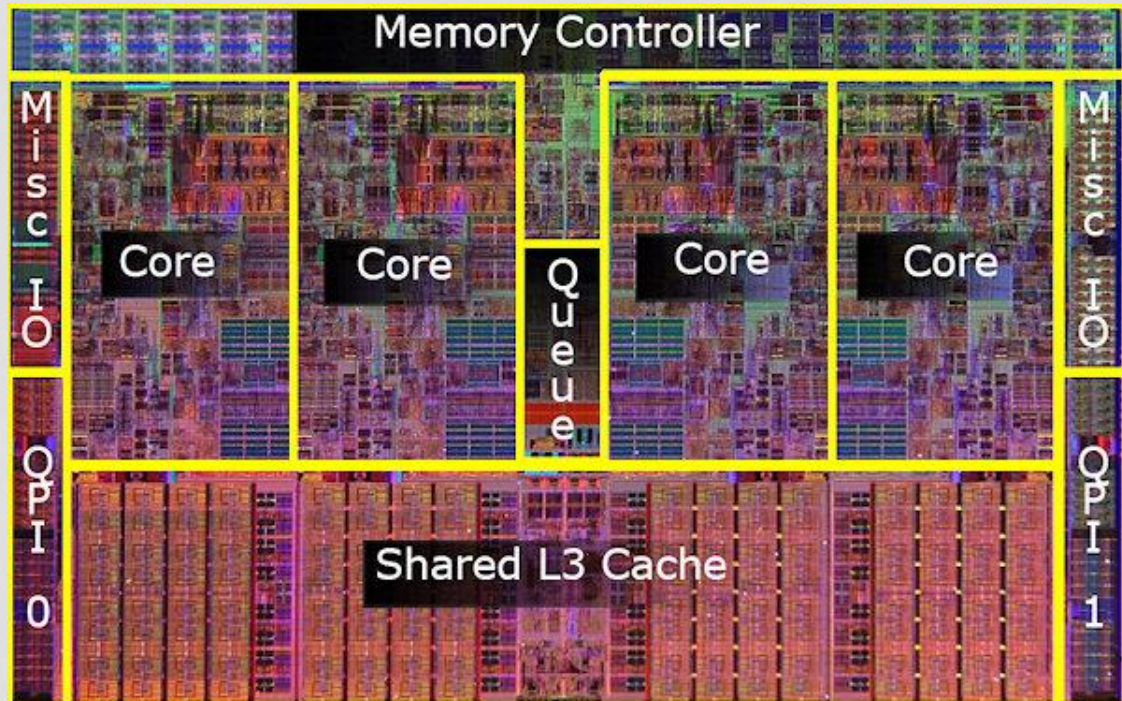
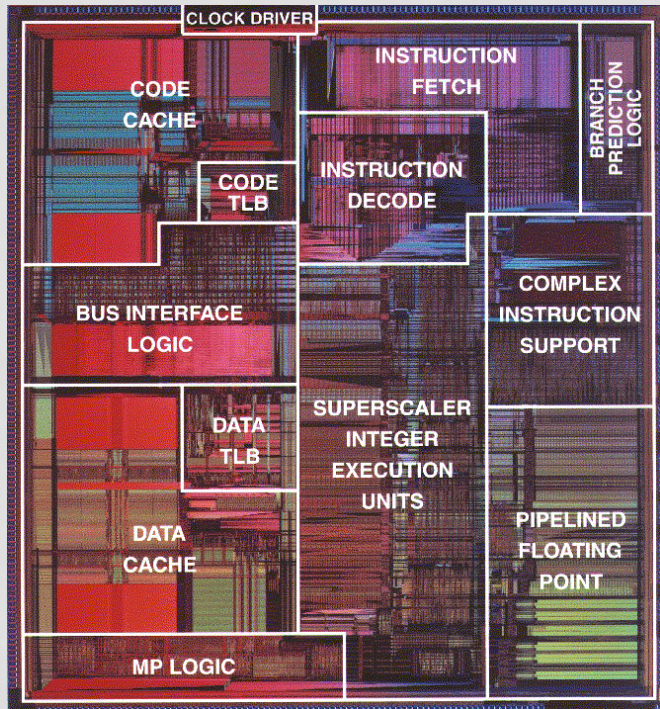


INTEL COMPUTER FAMILY



https://www.ncbi.nlm.nih.gov/books/NBK321721/figure/oin_tutorial.F3/

PENTIUM 4 / CORE I7

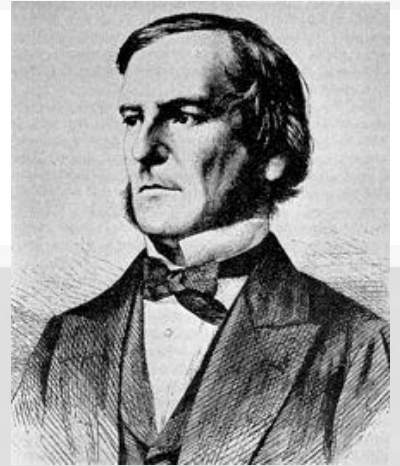


2020: 5 nm technology
1 atom \approx 0.2 – 0.5 nm

AGENDA

- ASCII en Unicode
- transistors
- **logische poorten**
- schakelingen met poorten
- half en full adder
- geheugen elementen
- systeemklok en timers

BOOLESE ALGEBRA

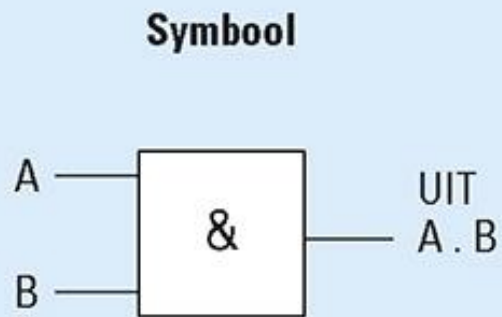


- Boolese algebra of schakel algebra
 - George Boole, 1815-1864
 - digitale schakelingen wiskundig beschrijven
- waarheidstabel : beschrijft uitgang als functie van ingang(en)
- logische operatoren (in C of Java):
 - &&
 - ||
 - !

```
if ((x < 0) || (x > 3)) {  
    ....  
}
```

AND

Figuur 2.4 **AND-poort**

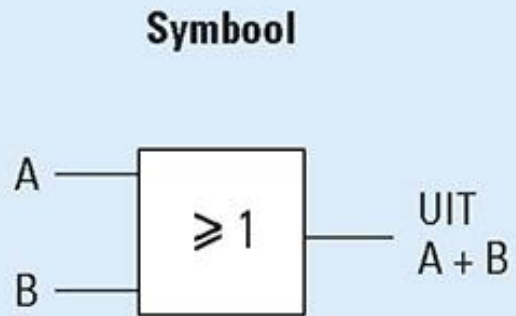


Waarheidstabel

A	B	UIT
0	0	0
0	1	0
1	0	0
1	1	1

OR

Figuur 2.5 **OR-poort**

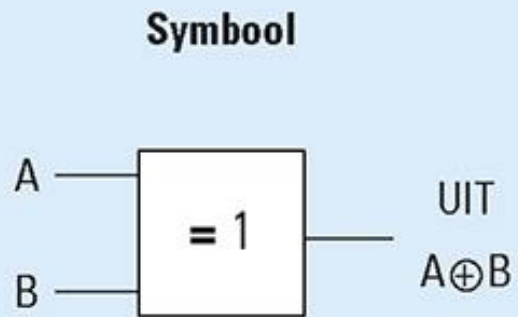


Waarheidstabel

A	B	UIT
0	0	0
0	1	1
1	0	1
1	1	1

(E)XOR

Figuur 2.9 EXOR-poort



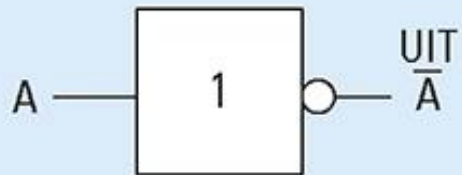
Waarheidstabel

A	B	UIT
0	0	0
0	1	1
1	0	1
1	1	0

INVERTOR

Figuur 2.6 **Inverter**

Symbol

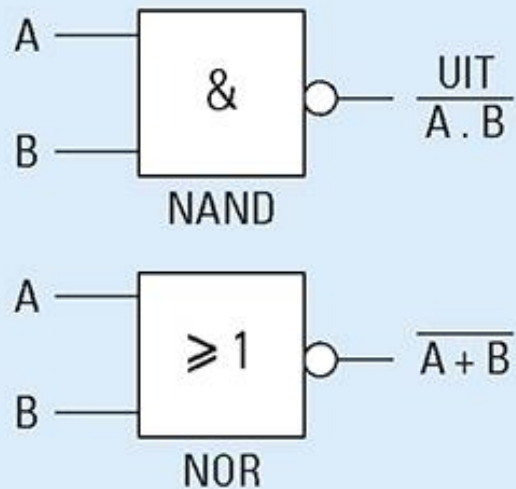


Waarheidstabel

A	UIT
0	1
1	0

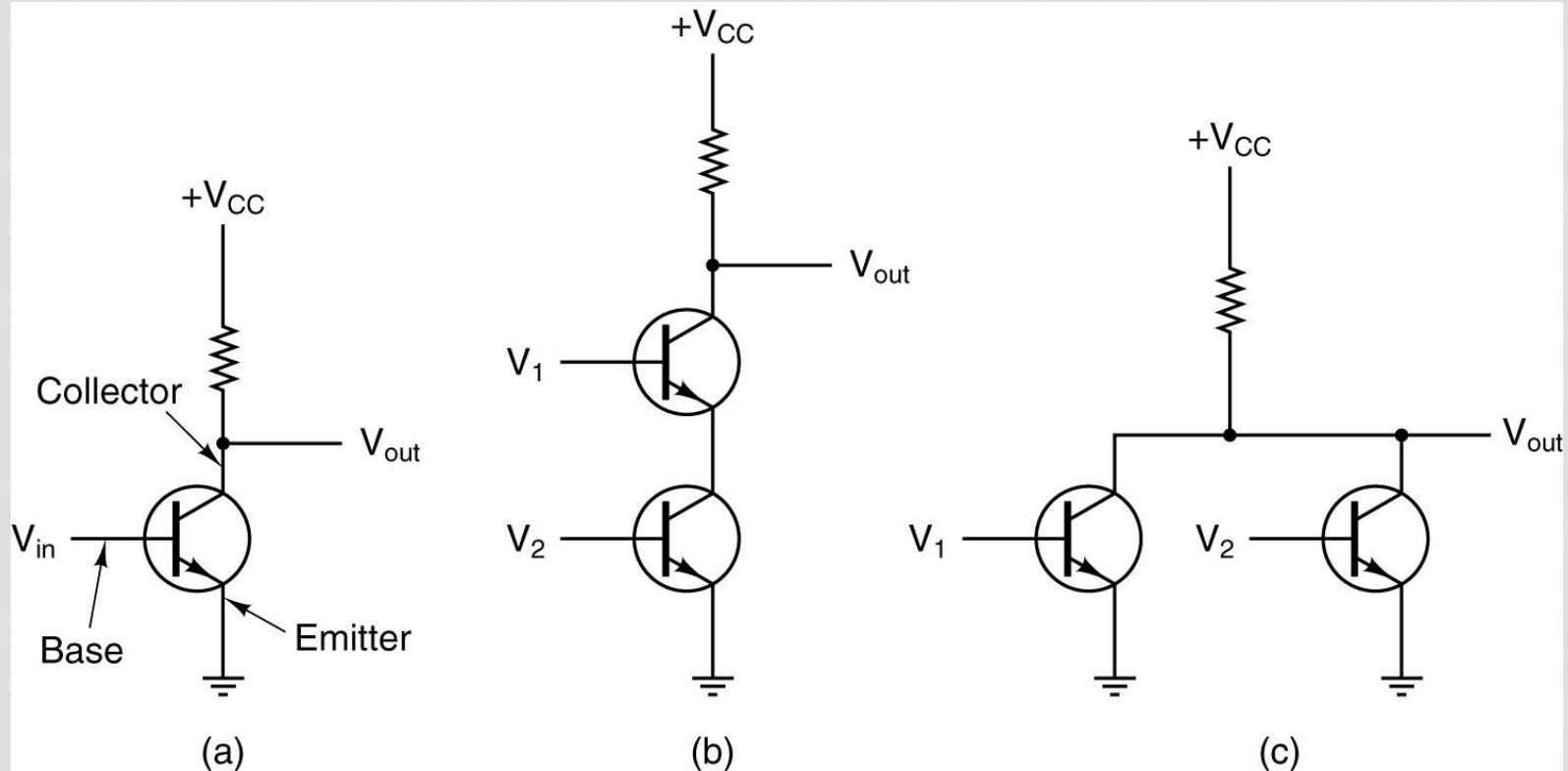
NAND EN NOR

Figuur 2.7 **NAND- en NOR-poorten**



Waarheidstabel

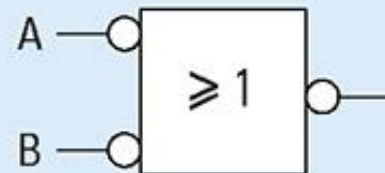
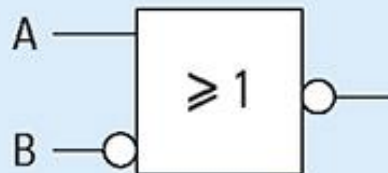
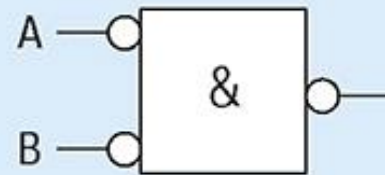
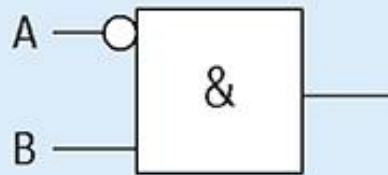
A	B	NAND	NOR
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0



(a) inverter
 (b) NAND gate
 (c) NOR gate

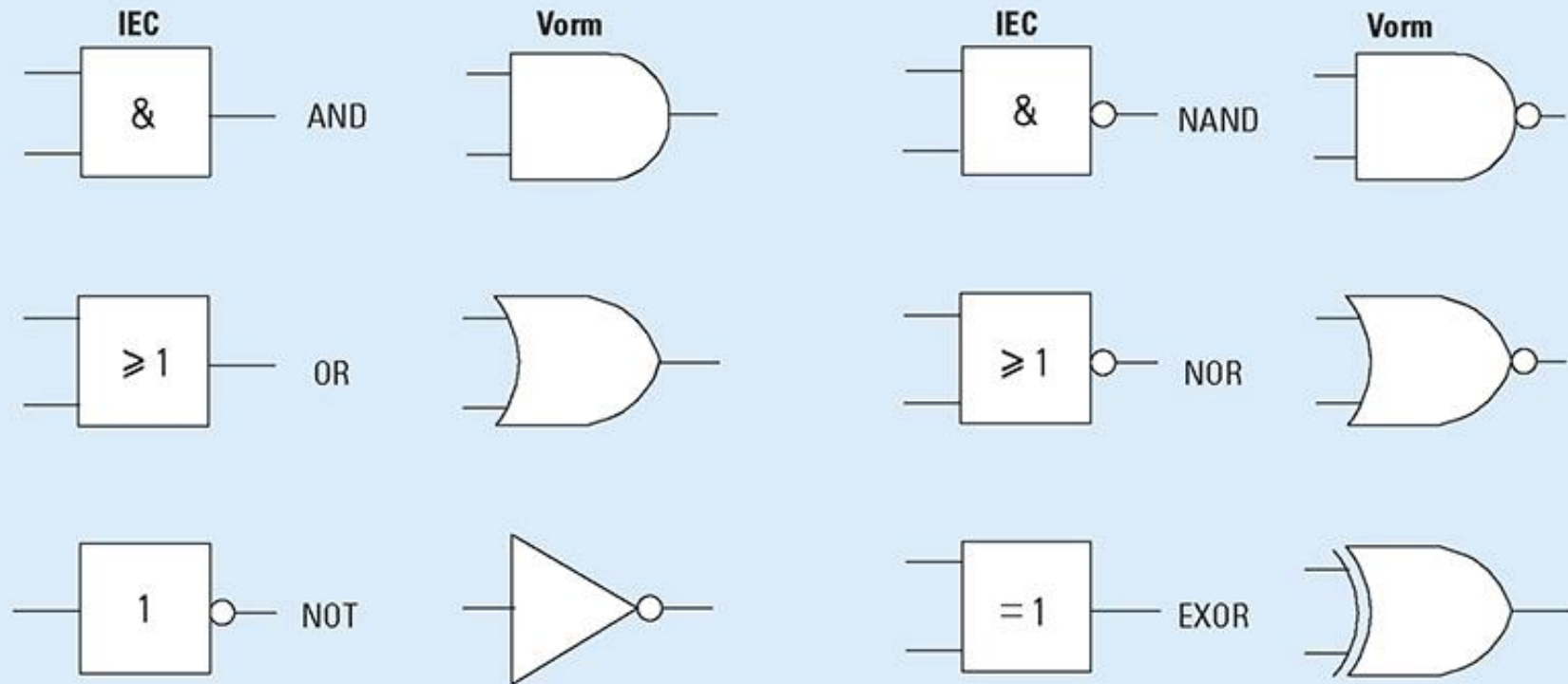
GEÏNVERTEERDE IN- EN UITGANGEN

Figuur 2.8 Geïnverteerde in- en uitgangen



AMERIKAANSE EN EUROPESE NOTATIE

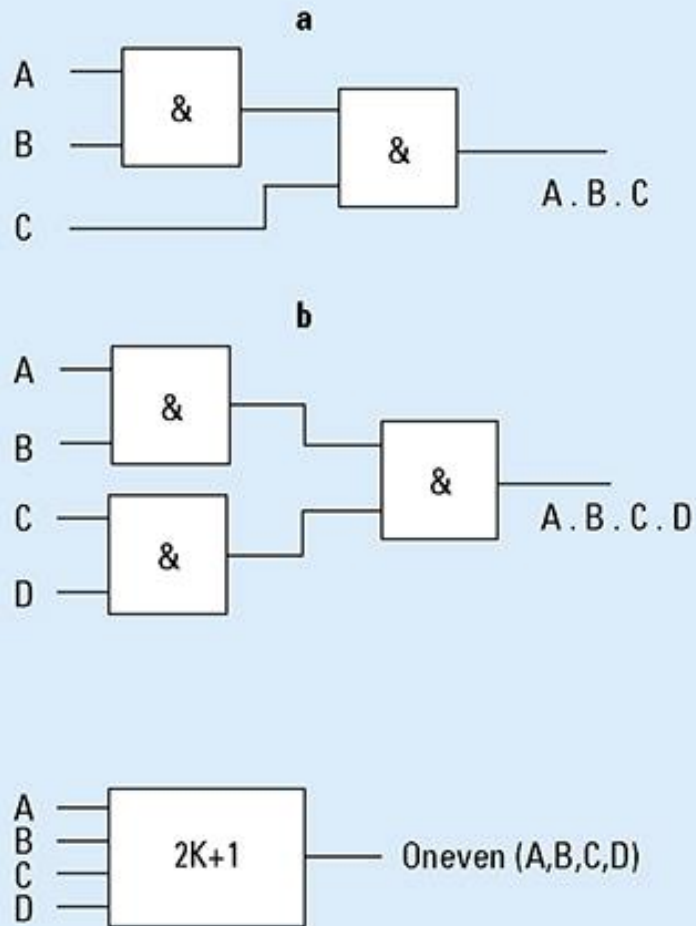
Figuur 2.10 Vorm- en IEC-symbolen



AGENDA

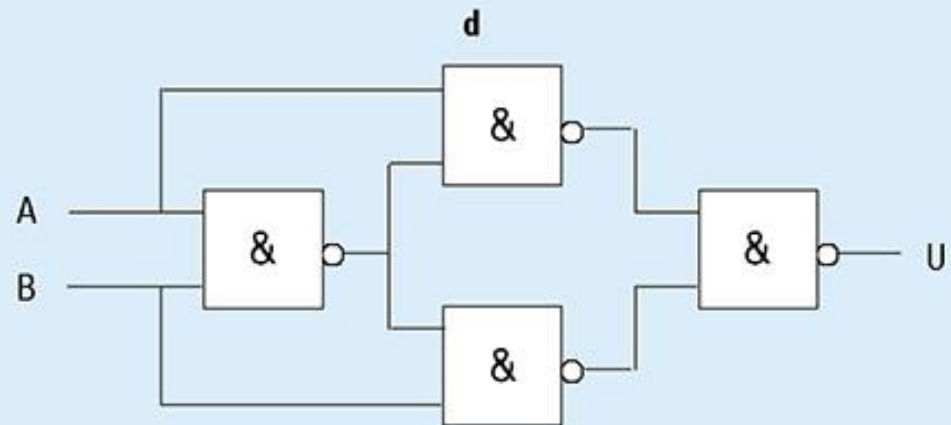
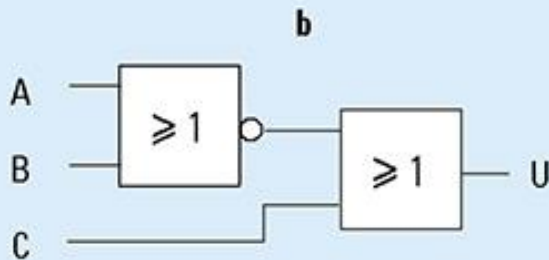
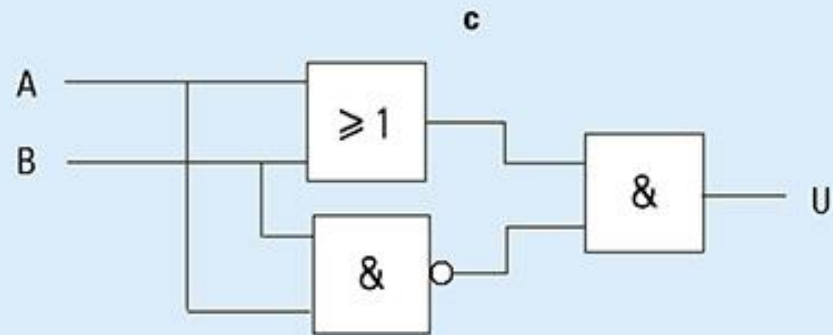
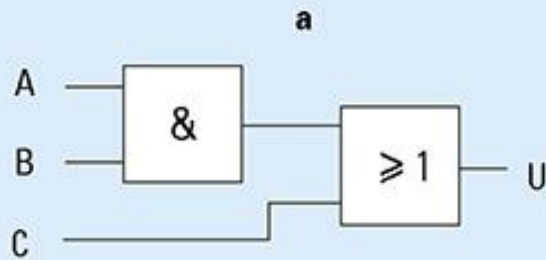
- ASCII en Unicode
- transistors
- logische poorten
- **schakelingen met poorten**
- half en full adder
- geheugen elementen
- systeemklok en timers

Figuur 2.13 Poorten samenstellen

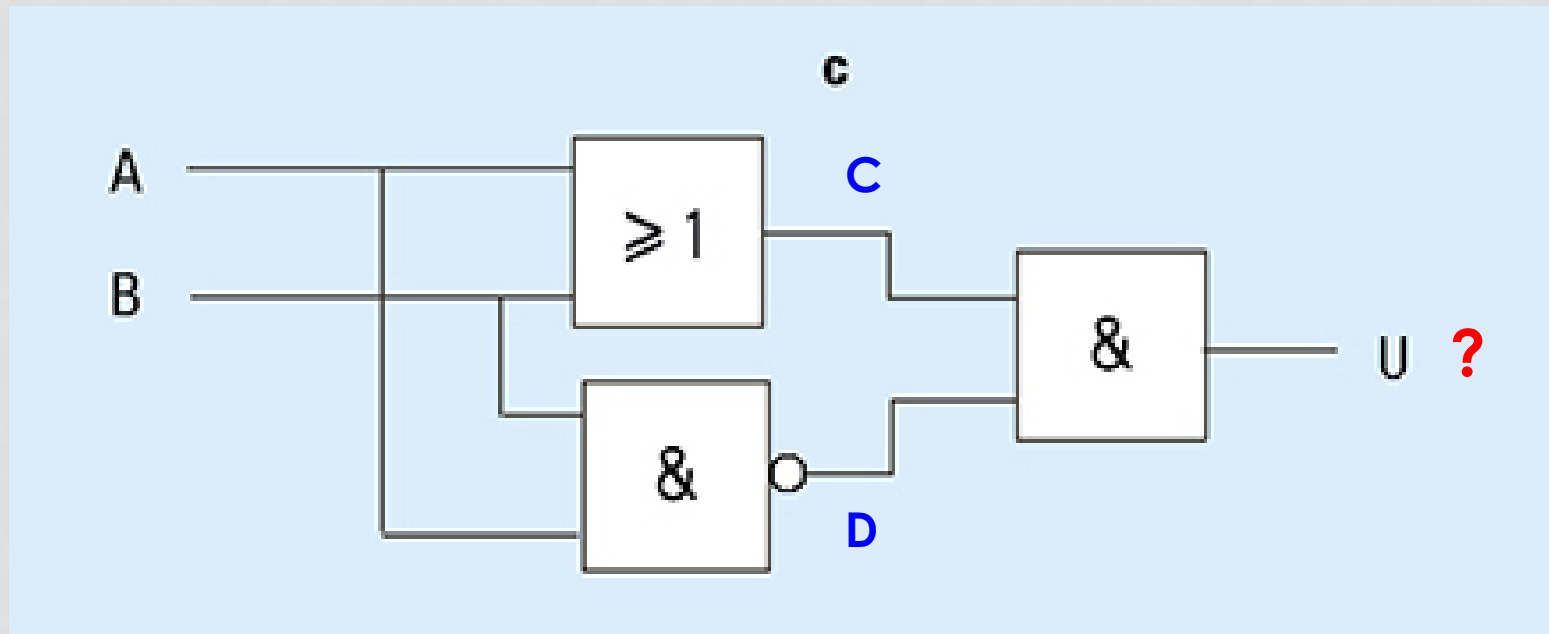


GEEF DE WAARHEIDSTABEL

Figuur 2.14 Poortschakelingen



OPGAVE



UITWERKING

A	B	C	D	U
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

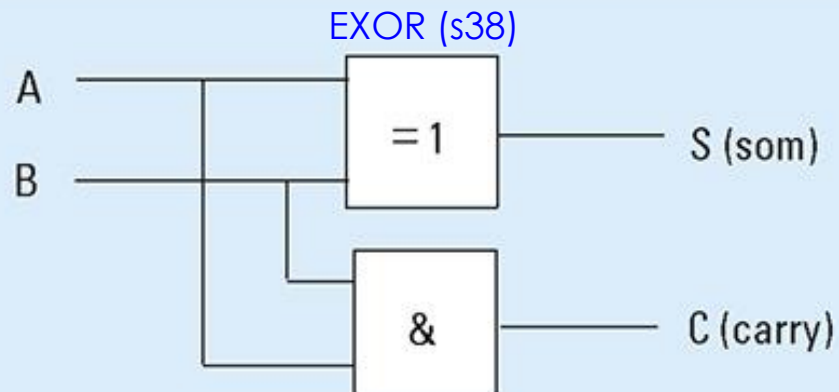
AGENDA

- ASCII en Unicode
- transistors
- logische poorten
- schakelingen met poorten
- **half en full adder**
- geheugen elementen
- systeemklok en timers

HALF ADDER

half : geen ingang voor carry-bit

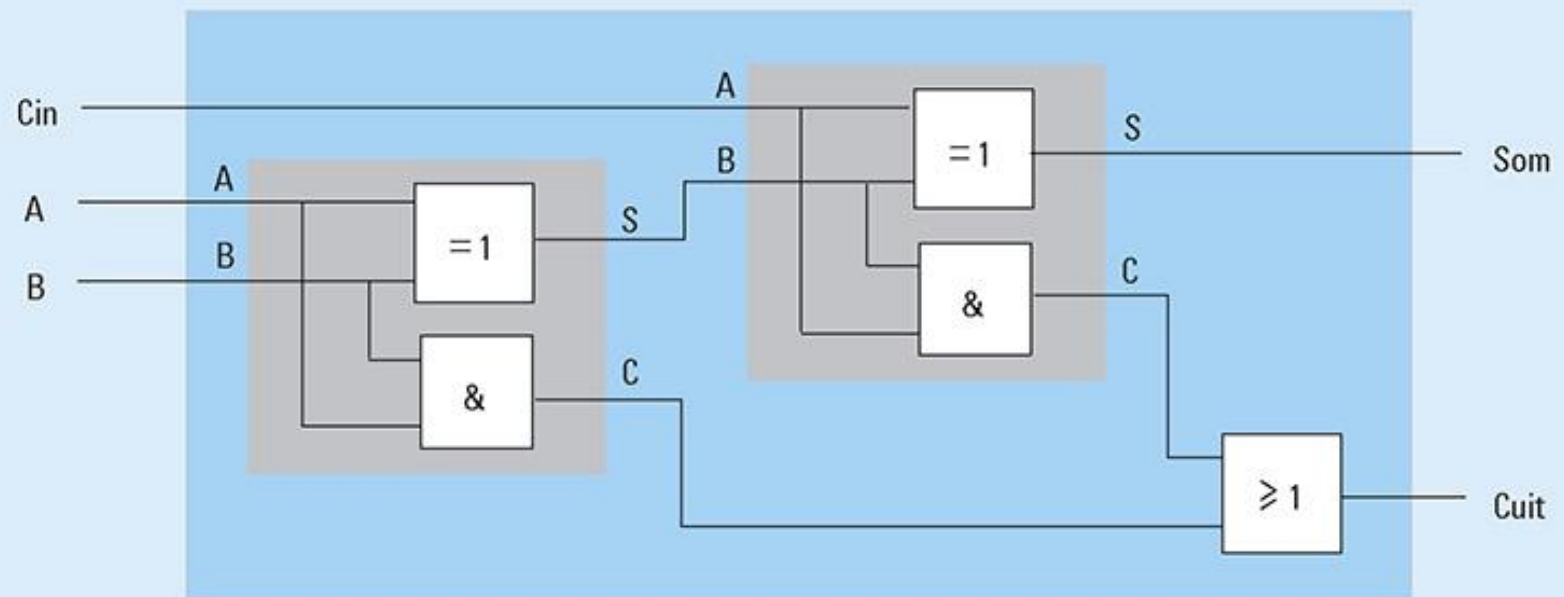
Figuur 2.19 **Half adder**



A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

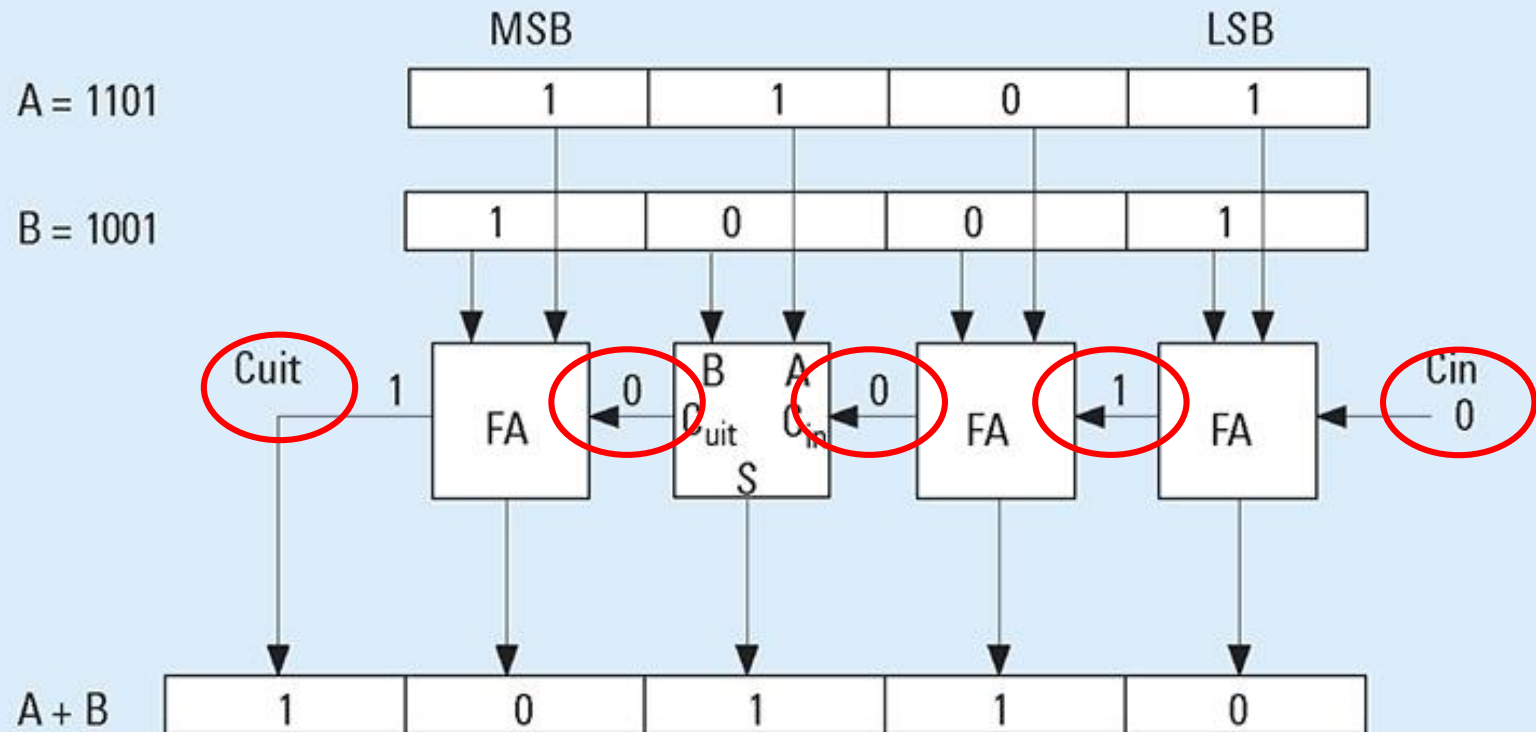
FULL ADDER

Figuur 2.22 Schakeling van de full adder



FULL ADDER

Figuur 2.23 Het optellen van twee getallen



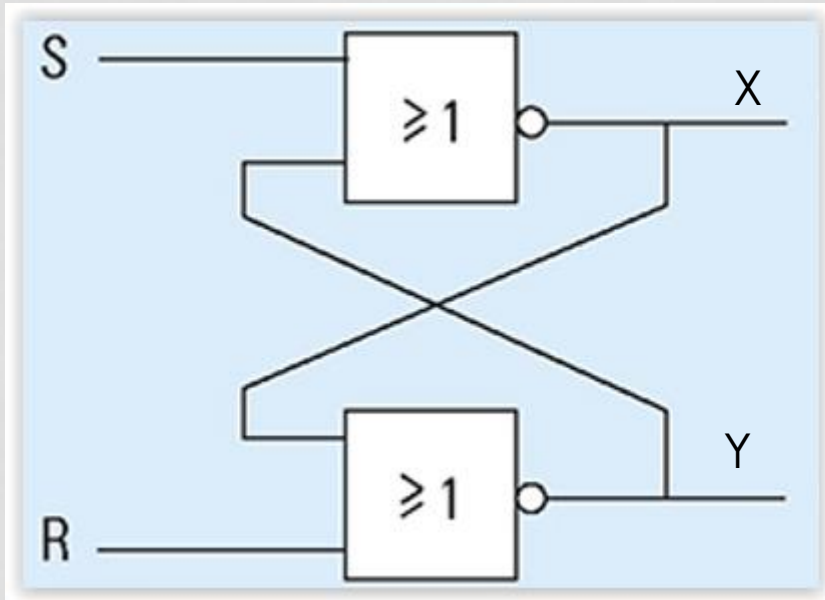
IS ÉÉN CARRY BIT WEL GENOEG ?

- als $A=B=1111$ zijn dan 5 bits genoeg ?
- ja, want $15+15=30 \leq 31$
- $\text{max} + \text{max} = 2 * \text{max}$ (= shift left 1)
- *algemeen* : $2^n - 1 + 2^n - 1 = 2^{n+1} - 2 \leq 2^{n+1} - 1$

AGENDA

- ASCII en Unicode
- transistors
- logische poorten
- schakelingen met poorten
- half en full adder
- **geheugen elementen**
- systeemklok en timers

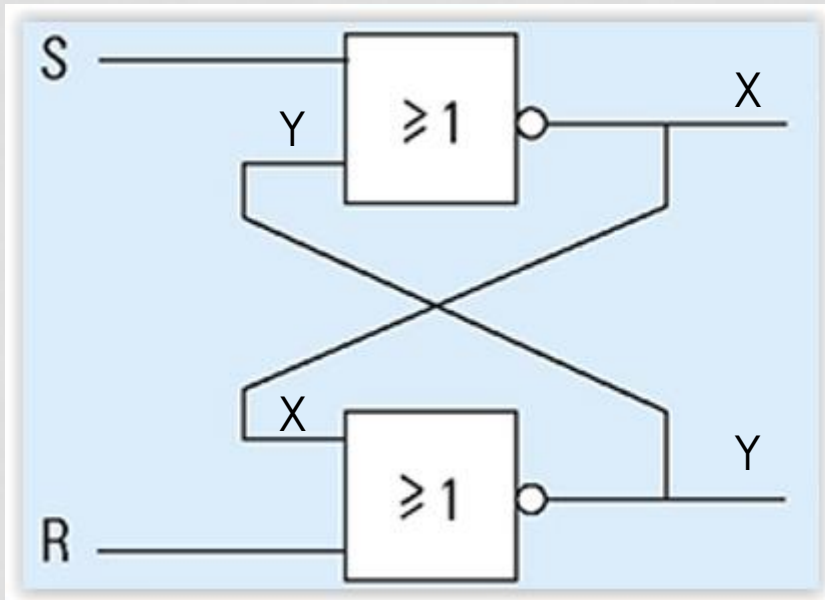
GEEF DE WAARHEIDSTABEL



Waarheidstabel

NOR		
IN		UIT
0	0	1
0	1	0
1	0	0
1	1	0

GEEF DE WAARHEIDSTABEL



Systeem net gestart

Stel R, S en Y en X zijn allemaal 0

- Y = 0 en S = 0 -> X gaat naar 1
- Als X = 1 en R = 0 -> Y blijft 0

Of?

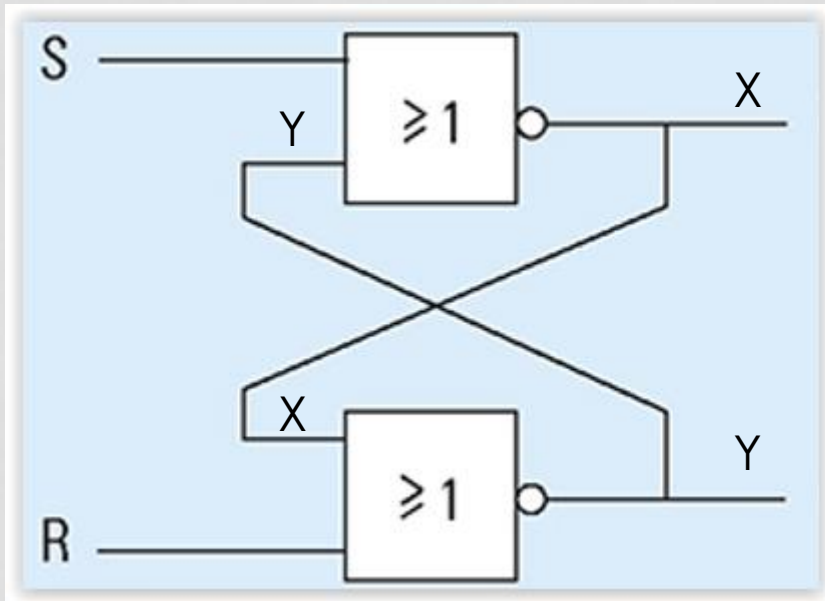
- X = 0 en R = 0 -> Y gaat naar 1
- Als Y = 1 en S = 0 -> X blijft 0

Dus: stabiele .. maar onbekende situatie

Waarheidstabel

NOR		
IN		UIT
0	0	1
0	1	0
1	0	0
1	1	0

GEEF DE WAARHEIDSTABEL



T1

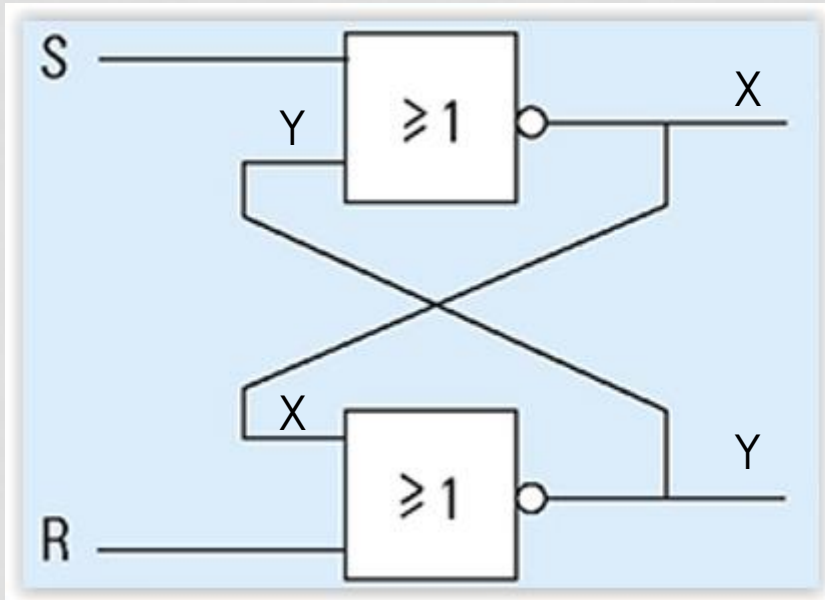
$Y = 0$ (of 1), $X = 1$ (of 0), $R = 0$, S wordt 1

- X gaat naar 0 (ongeacht Y !)
- Y gaat naar 1
- Stabiele , bekende situatie

Waarheidstabel

NOR		
IN		UIT
0	0	1
0	1	0
1	0	0
1	1	0

GEEF DE WAARHEIDSTABEL



T2

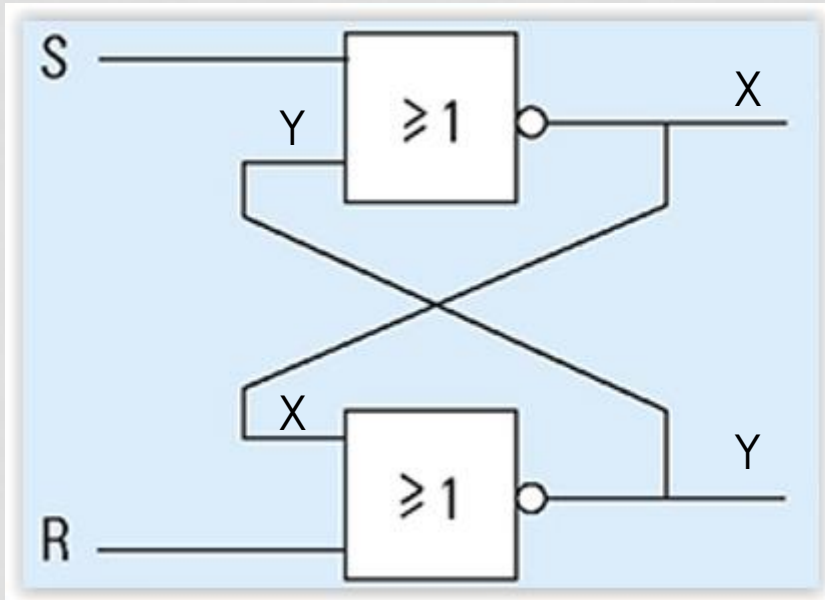
Y = 1, X = 0, R = 0, S wordt (weer) 0

- X blijft 0
- Y blijft 1
- Stabiele, bekende situatie

Waarheidstabel

NOR		
IN		UIT
0	0	1
0	1	0
1	0	0
1	1	0

GEEF DE WAARHEIDSTABEL



T3

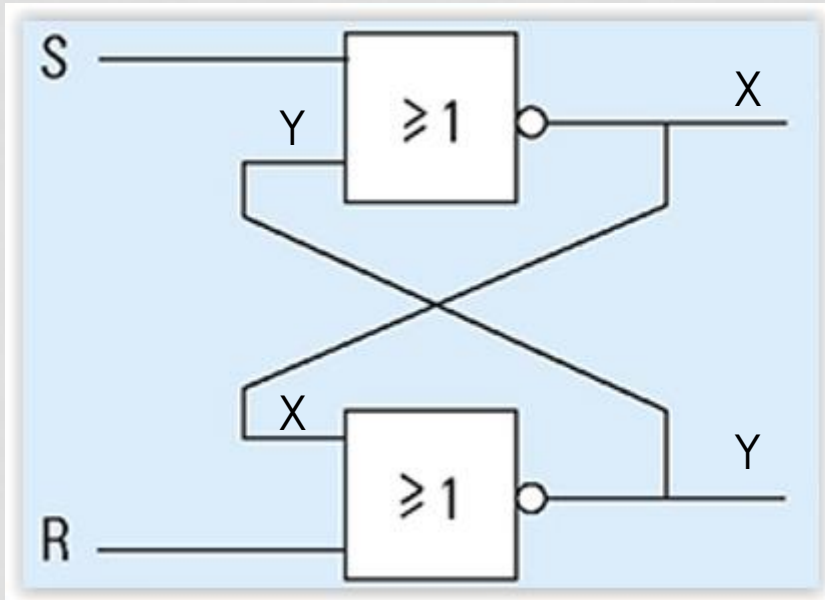
$Y = 1, X = 0, S = 0, R$ wordt 1

- Y wordt 0 (ongeacht X)
- X wordt 1
- Stabiele, bekende situatie

Waarheidstabel

NOR		
IN		UIT
0	0	1
0	1	0
1	0	0
1	1	0

GEEF DE WAARHEIDSTABEL



T4

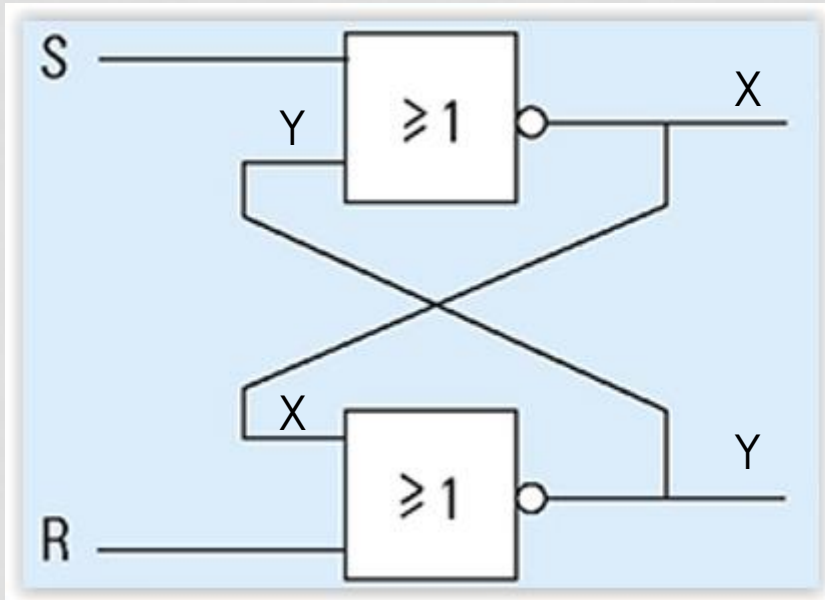
$Y = 1, X = 0, S = 0, R$ wordt (weer) 0

- Y blijft 0
- X blijft 1
- Stabiele, bekende situatie

Waarheidstabel

NOR		
IN		UIT
0	0	1
0	1	0
1	0	0
1	1	0

GEEF DE WAARHEIDSTABEL



T5

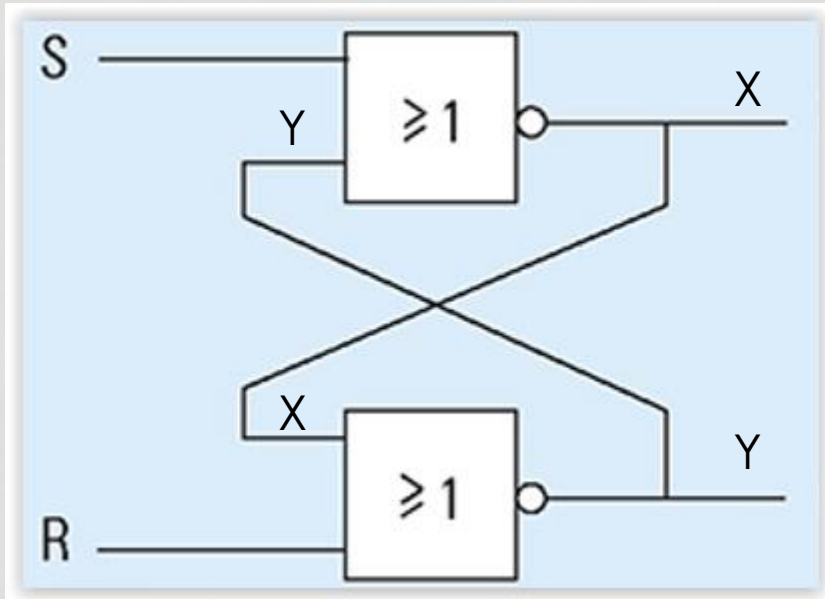
Y = 1, X = 0, S en R worden beide een

- X wordt 0, Y wordt ook 0
- Stabiele, bekende situatie

Waarheidstabel

NOR		
IN		UIT
0	0	1
0	1	0
1	0	0
1	1	0

GEEF DE WAARHEIDSTABEL



Conclusie:

$S = 1$ en $R = 0 \rightarrow Y = 1, X = 0$

$S = 0$ en $R = 1 \rightarrow Y = 0, X = 1$

$S = 0$ en $R = 0 \rightarrow X$ en Y ongewijzigd

$S = 1$ en $R = 1 \rightarrow X$ en Y beide 0

Afspraak: $(S = 1$ en $R = 1)$ niet toegestaan.

Gegeven dat:

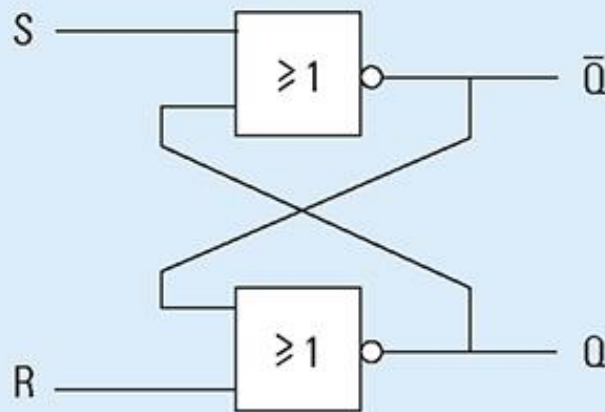
X en Y altijd elkaars inverse

Waarheidstabel

NOR		
IN		UIT
0	0	1
0	1	0
1	0	0
1	1	0

1-BIT GEHEUGENCEL

Figuur 2.36 De set-reset latch

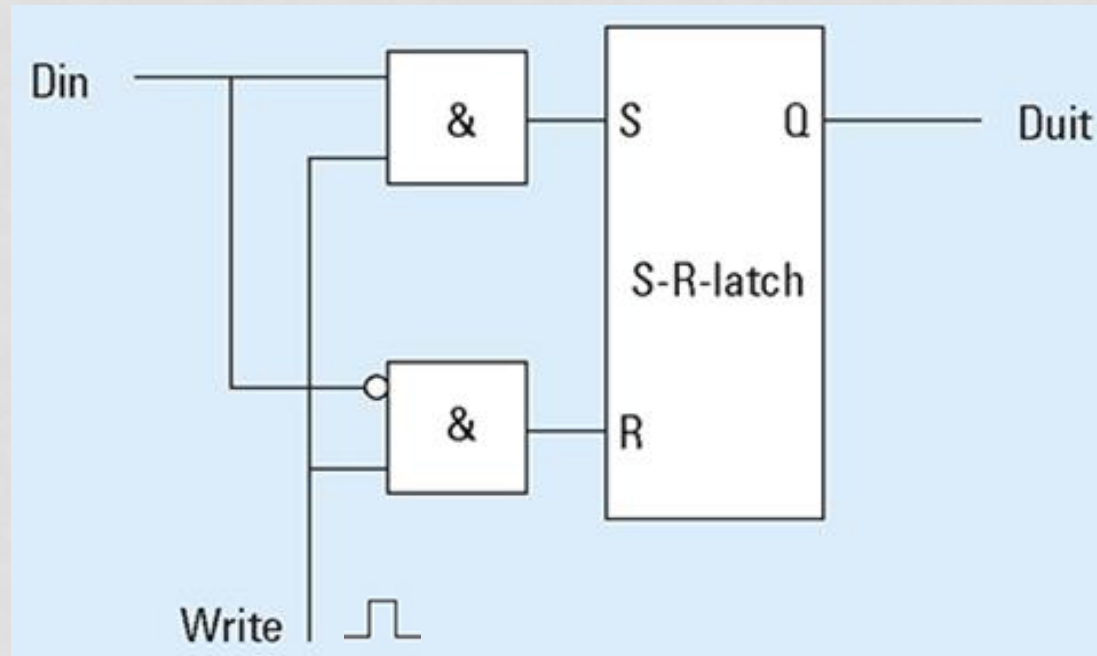


Waarheidstabel

NOR		
	IN	UIT
0	0	1
	1	0
1	0	0
1	1	0

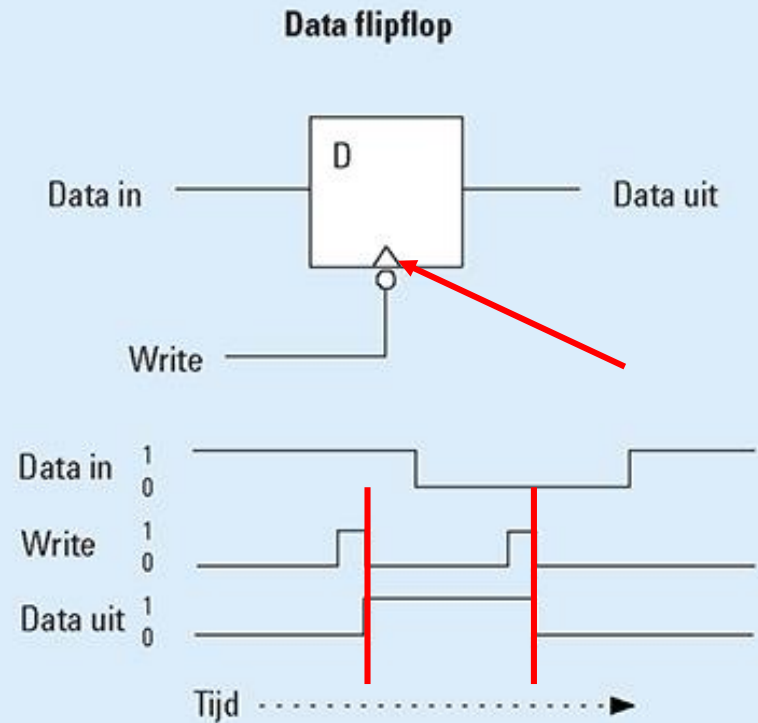
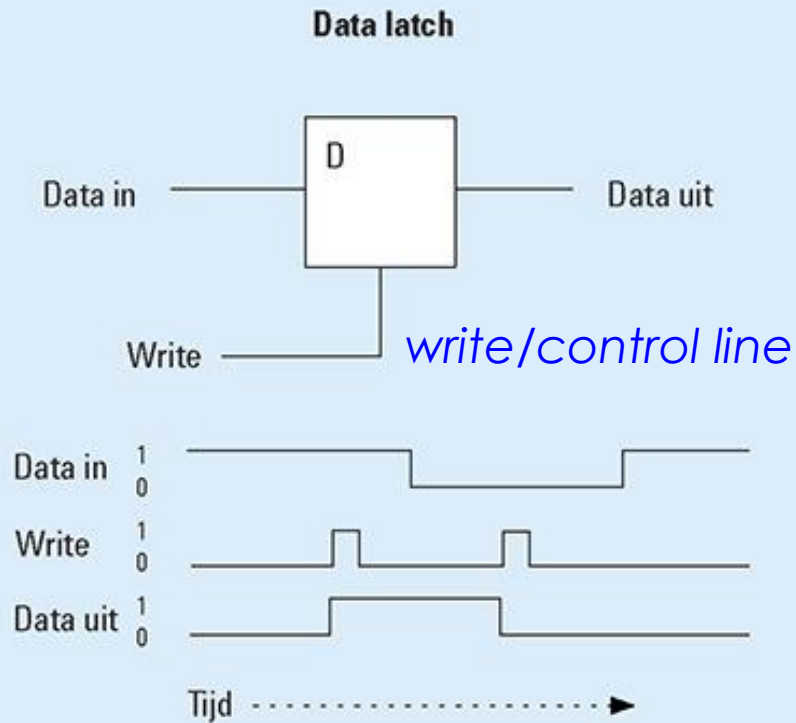
R = S = 0 : **hold** of "store data" (beide NOR's werken als invertor)
S = 1 & R=0 : $\sim Q = 0$ en onderste NOR=invertor, **set** the latch, Q wordt 1
R = 1 & S=0 : **reset** the latch, Q wordt 0
R = S = 1 mag niet !
dus : latch "onthoudt" in de "hold" stand

DE SET-RESET LATCH



LATCH EN FLIP-FLOP

Figuur 2.24 De D-latch en D-flipflop

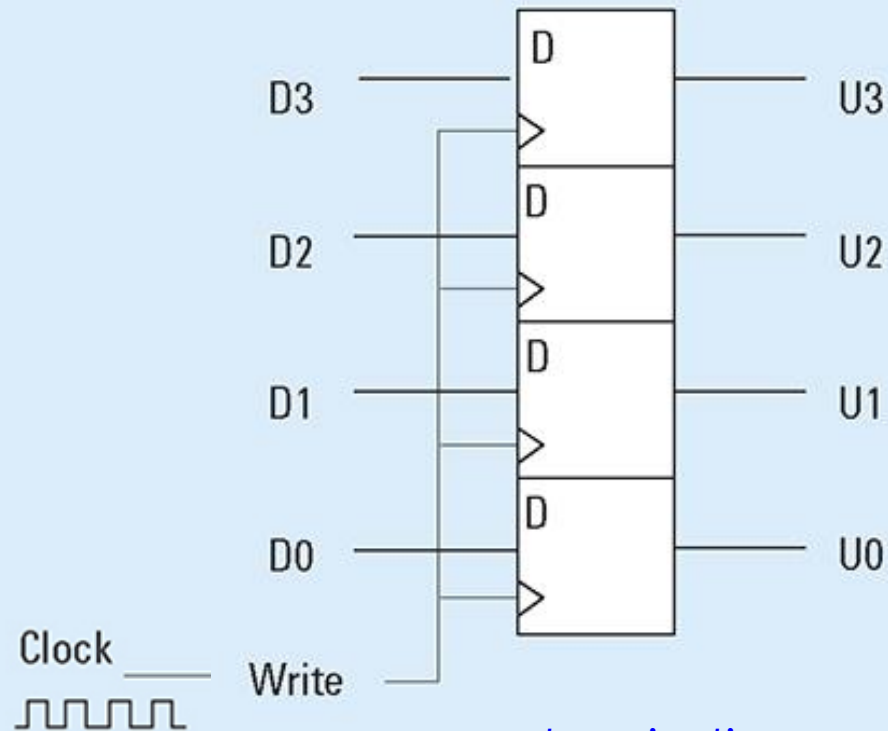


latch : level sensitive

ff : edge sensitive

REGISTER

Figuur 2.25 Register van vier bits

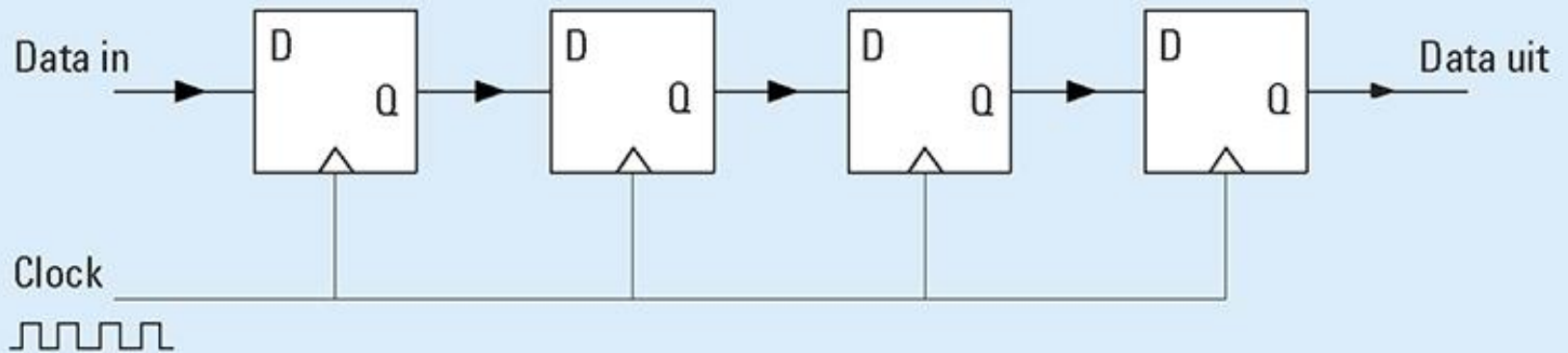


4 flipflops

*synchronisatie
met kloksignaal*

SCHUIFREGISTER

Figuur 2.26 **Schuifregister**



Bij elke voorflank / rising edge schuift het woord 1 positie naar rechts

COMBINATORISCH VS. SEQUENTIEEL

- combinatorisch circuit
 - uitgang is alleen een functie van huidige toestand ingangen
 - schakelingen met poorten zonder feedback
 - voorbeeld : een full-adder
 - NAND-poort delay ≈ 10 nS
- sequentieel circuit
 - uitgang hangt niet alleen af van huidige, maar ook van vorige "toestand "
 - sequentiële schakeling bevat *geheugen*
 - voorbeelden : flip-flop, teller, schuifregister

AGENDA

- ASCII en Unicode
- transistors
- logische poorten
- schakelingen met poorten
- half en full adder
- geheugen elementen
- **systemklok en timers**

SYSTEMKLOK

- elektronisch circuit dat een signaal afgeeft met exacte puls-breedte
 - vaste periode
 - vaste frequentie
- sequentiële schakelingen worden aangestuurd door klok



SYSTEMKLOK

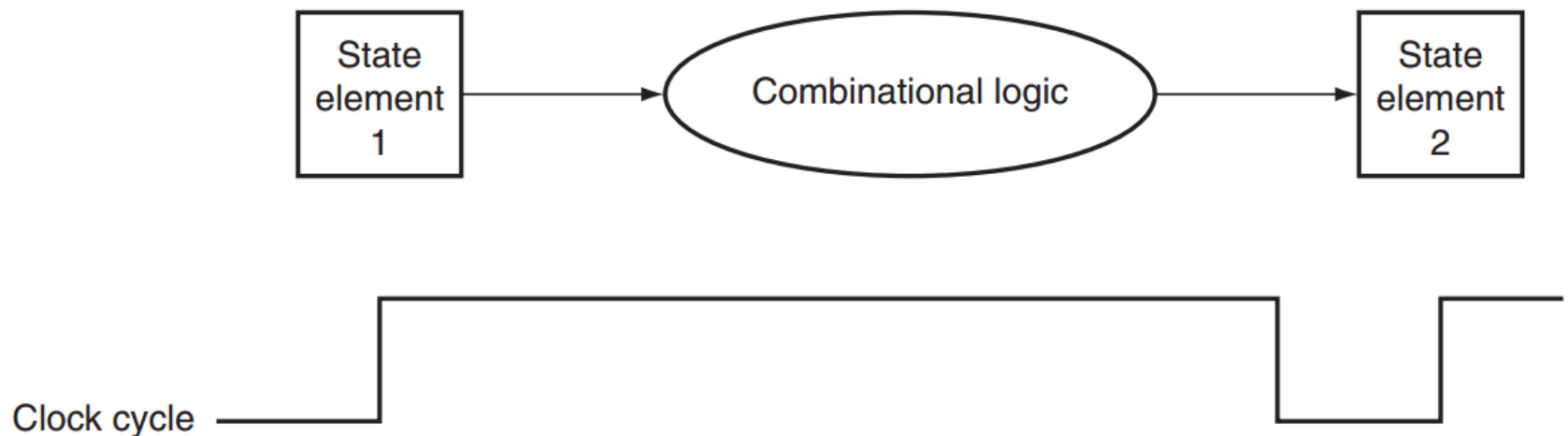
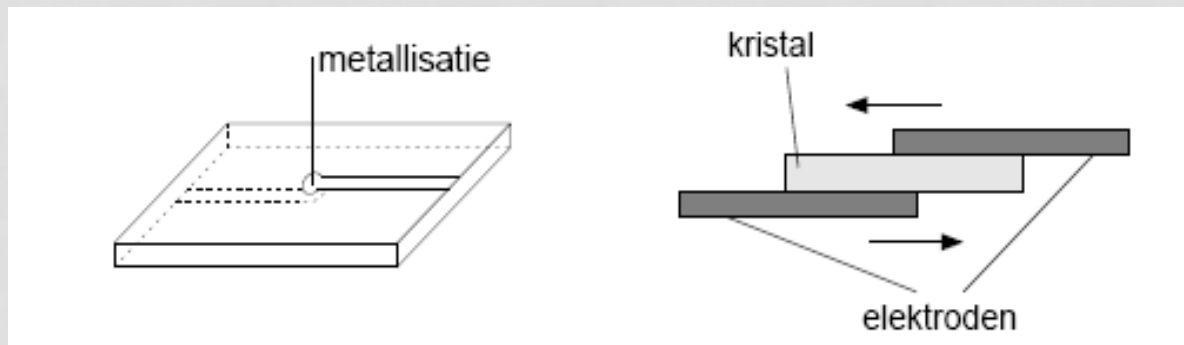


FIGURE B.7.2 The inputs to a combinational logic block come from a state element, and the outputs are written into a state element. The clock edge determines when the contents of the state elements are updated.



- **kwartskristal** : dun plaatje kwarts waarvan de onder- en bovenzijde voorzien zijn van een metallisatie
- bij een elektrische spanning zal het kristal elastisch vervormen
- piëzo-elektrisch effect : mechanische trillingen in het kristal veroorzaken een elektrisch signaal en omgekeerd
- ander typen oscillatoren : RC en keramisch



EEN (UP/DOWN) COUNTER



OPGAVE

- systeemklok heeft een frequentie van 1 MHz
- counter telt tot 2000
- wat is de periode van de puls die de counter afgeeft ?