



Opfriskursus digitale kennis

UBA MWV – 12 dec 2019



Inhoud deel 1

1. Analooq en digitaal

2. Digitale systemen

3. Getalsystemen

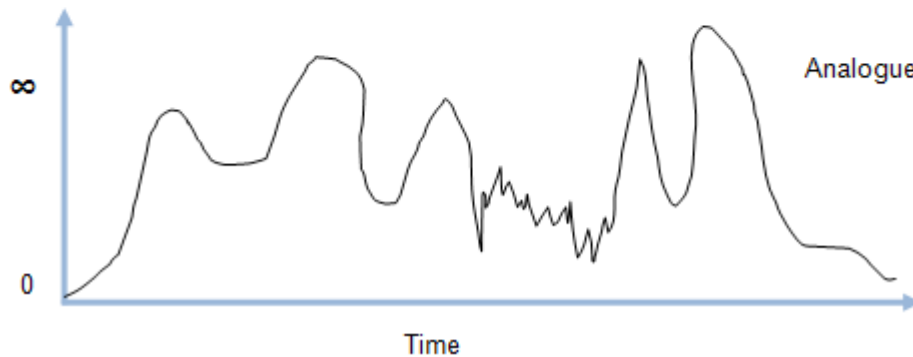
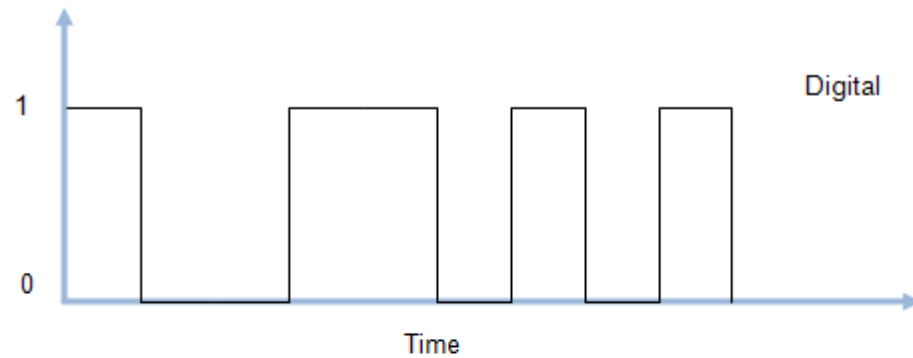
- a. decimaal
- b. binair
- c. hexadecimaal

4. Rekenen met binaire getallen

5. Binaire code-systemen

- a. BCD-kode
- b. Gray-kode
- c. ASCII-kode

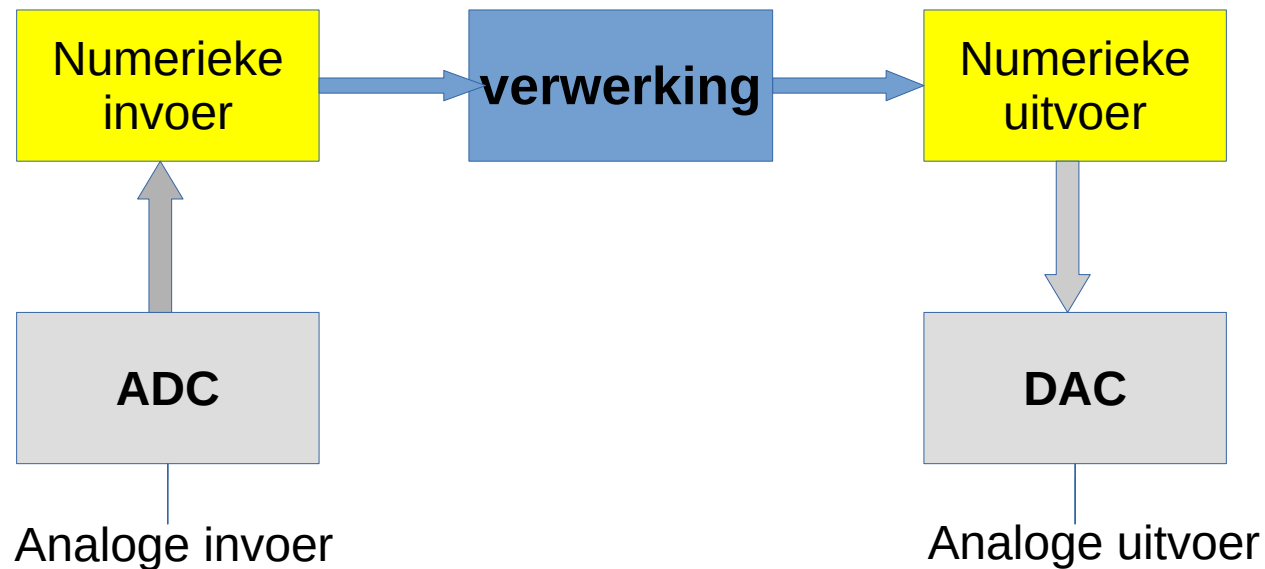
1. Analooq en digitaal



Een **analooq** signaal is continu in tijd en amplitude

Een **digitaal** signaal is discontinu in tijd en de amplitude heeft alleen discrete waarden

2. Digitale systemen



Elk digitaal systeem werkt met getallen!

3. Getalsystemen

a. decimaal getalsysteem

– grondtal=10 er zijn 10 cijfers: 0,1,2,3,4,5,6,7,8,9

– positie cijfer bepaalt de getalwaarde

$$2019 = 9 \times 1 + 1 \times 10 + 0 \times 100 + 2 \times 1000$$

$$2019 = 2 \times 10^3 + 0 \times 10^2 + 1 \times 10^1 + 9 \times 10^0$$

– dit geldt ook voor getallen kleiner dan 1:

$$1,75 = 1 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2}$$

– negatieve getallen worden voorafgegaan door '-' teken

3 Getalsystemen

b. BINAIR getalsysteem

- grondtal=2 er zijn 2 cijfers: 0,1
- positie cijfer bepaalt de getalwaarde

$$1011 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$1011 = 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 11 \text{ decimaal}$$

- dit geldt ook voor getallen kleiner dan 1:

$$11,01 = 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

$$11,01 = 2 + 1 + 0 + 0,25 = 3,25 \text{ decimaal}$$

- **negatieve** getallen worden (meestal) weergegeven met het 'two-complement' voorbeeld: **-17**

$$\begin{array}{ccccccc} - 1011 & \text{-----}> & 01011 & \text{-----}> & 10100 & \text{-----}> & \mathbf{10101} \\ \text{nul voorzetten} & & \text{inverteren} & & \text{1 bij tellen} & & \end{array}$$

Tabel1: Machten van 2

n bits:

Aantal combinaties
= 2^n

Grootste dec. Getal
= $2^n - 1$

aantal keer snijden	aantal lagen
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1 024
11	2 048
12	4 096
13	8 192
14	16 384
15	32 768
16	65 536
17	131 072
18	262 144
19	524 288
20	1 048 576
21	2 097 152
22	4 194 304
23	8 388 608
24	16 777 216
25	33 554 432
26	67 108 864
27	134 217 728
28	268 435 456
29	536 870 912
30	1 073 741 824
31	2 147 483 648
32	4 294 967 296

aantal keer snijden	aantal lagen
33	8 589 934 592
34	17 179 869 184
35	34 359 738 368
36	68 719 476 736
37	137 438 953 472
38	274 877 906 944
39	549 755 813 888
40	1 099 511 627 776
41	2 199 023 255 552
42	4 398 046 511 104
43	8 796 093 022 208
44	17 592 186 044 416
45	35 184 372 088 832
46	70 368 744 177 664
47	140 737 488 355 328
48	281 474 976 710 656
49	562 949 953 421 312
50	1 125 899 906 842 624
51	2 251 799 813 685 248
52	4 503 599 627 370 496
53	9 007 199 254 740 992
54	18 014 398 509 481 984
55	36 028 797 018 963 968
56	72 057 594 037 927 936
57	144 115 188 075 855 872
58	288 230 376 151 711 744
59	576 460 752 303 423 488
60	1 152 921 504 606 846 976
61	2 305 843 009 213 693 952
62	4 611 686 018 427 387 904
63	9 223 372 036 854 775 808
64	18 446 744 073 709 551 616

3. Getalsystemen

c. HEXADECIMAAL getalsysteem (h of \$ of 0x)

- grondtal=16 er zijn 16 **cijfers**: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- positie cijfer bepaalt de getalwaarde

$$3FA6 = 3 \times 16^3 + 15 \times 16^2 + 10 \times 16^1 + 6 \times 16^0$$

$$3FA6 = 3 \times 4096 + 15 \times 256 + 10 \times 16 + 6 \times 1 = 16294 \text{ decimaal}$$

- dit geldt ook voor getallen kleiner dan 1:

$$2F,3A = 2 \times 16^1 + 15 \times 16^0 + 3 \times 16^{-1} + 10 \times 16^{-2}$$

$$2F,3A = 32 + 15 + 3 \times 0,0265 + 10 \times 0,0039625 = 47,119125 \text{ decimaal}$$

- **negatieve getallen** worden van een '-' teken voorzien

DEC - HEX - BIN

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

4. Rekenen met binaire getallen

Rekenregels

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ en } 1 \text{ links bijtellen}$$

$$0 - 1 = 1 \text{ en links } 1 \text{ aftrekken}$$

Voorbeelden

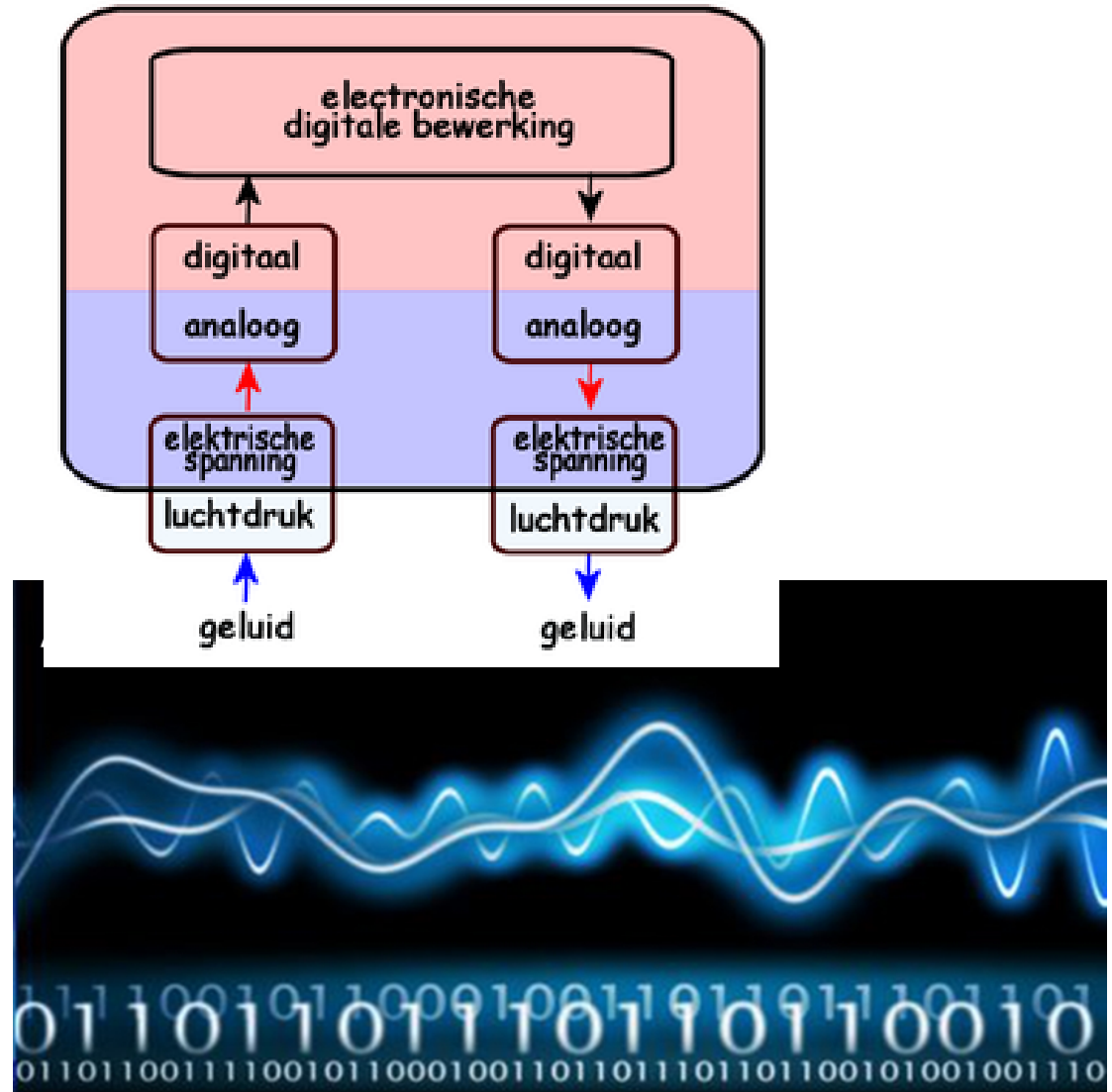
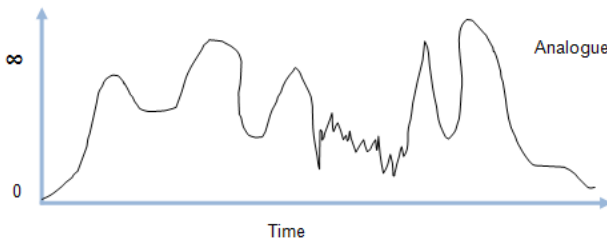
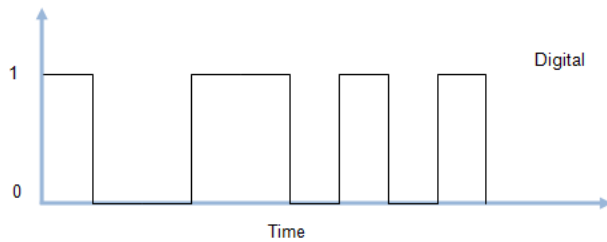
$$\begin{array}{r} 1011 \\ +1100 \\ \hline 10111 \end{array}$$

$$\begin{array}{r} 10111 \\ - 1101 \\ \hline 01010 \end{array}$$

$$\begin{array}{r} 10100 \\ \times 101 \\ \hline 10100 \\ 00000 \\ 10100 \\ \hline 1100100 \end{array}$$

Binair rekenen gebeurt op dezelfde manier als met decimale getallen

Verband binaire getallen en digitale signaalverwerking



5. Binaire codesystemen

BCD-code

- voor het coderen van arabische cijfers met een 4-bit code
- er zijn daarom alleen 10 ‘toegelaten’ codes!
- Is een gewogen code (8-4-2-1)
- wordt gebruikt o.a. voor displays

	MSB	BCD ₈₄₂₁		LSB
Decimal	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Voorbeeld:

De **cijfers** 183 in BCD-code:

0001.1000.0011

Opmerking: het **getal** (=aantal) 183

183 = B101010111

5. Binaire codesystemen

GRAY-code

- voor het coderen van 16 verschillende toestanden met 4 bits
- Is een niet-gewogen code-systeem
- de eerste 10 codes zijn BCD-kodes
- voor elke volgende code wijzigt slechts 1 bit in de code
- wordt gebruikt in bijvoorbeeld rotary encoders

Decimal Number	4 bit Binary Number <u>ABCD</u>	4 bit Gray Code <u>G₁G₂G₃G₄</u>
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000



5. Binaire code-systemen

ASCII-code

- binaire code voor letters, cijfers en controletekens
- ASCII = American Standard Code for Information Interchange
- bedoeld om alfanumerieke data tussen apparaten uit te wisselen
- oorspronkelijk 7 bit brede code (maximaal 128 tekens)
- veelal 8 bit breed (maximaal 256 tekens)
- er bestaan tegenwoordig diverse uitbreidingen: extended ascii, IBM (PC-) ascii, unicode...

Tabel 2 (Oorspronkelijke) ASCII-codetabel

0-31
controletekens

48-57
Cijfers

65-90
Hoofdletters

97-122
kleine letters

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F		127	7F	□



Inhoud deel 2

1. Bits en bytes
2. Bouw en werking van computersystemen
3. micro**processor**, micro**computer**, micro**controller**
4. Logische bewerkingen
NOT, AND, OR, EXOR, schuiven
5. Opcode(s)
6. Programmeren en programmeertalen
broncode en objectcode
assembleren en linken

1. What 's in a word... Over bits en bytes

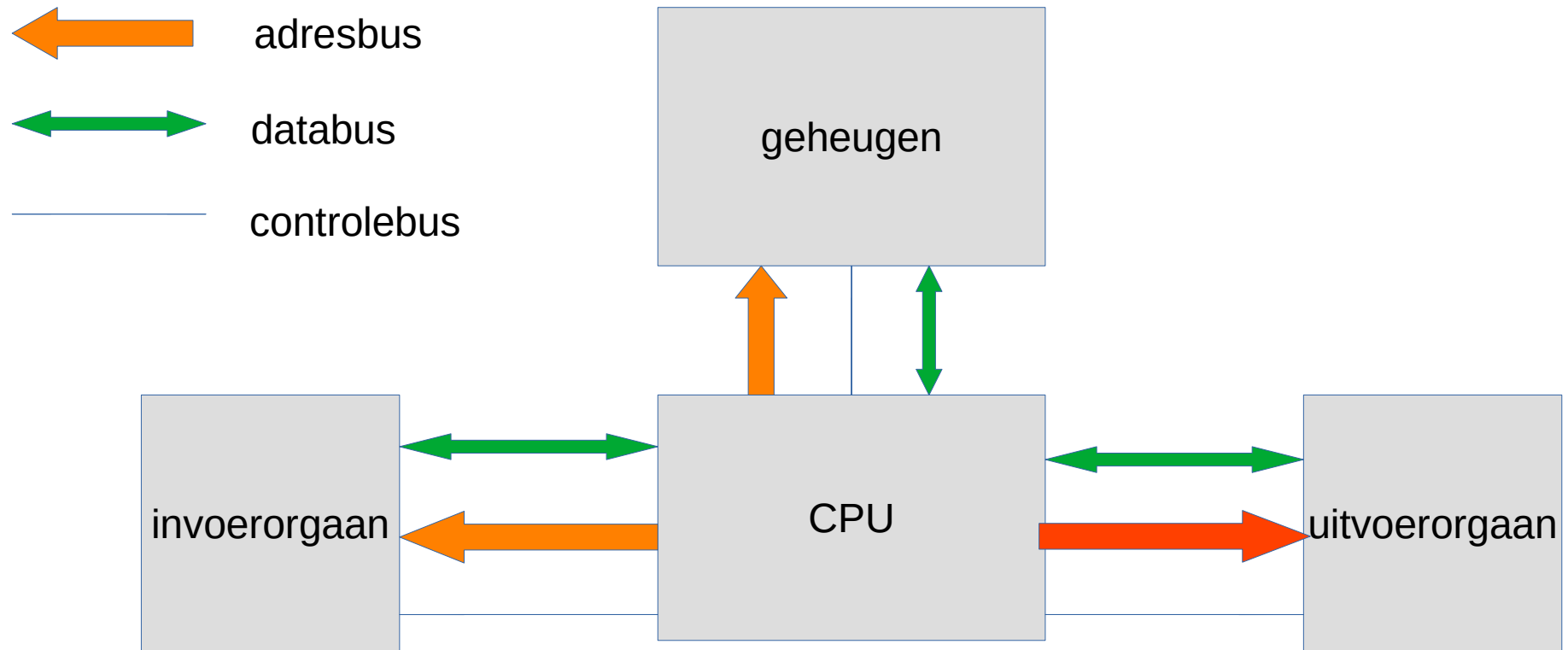
BIT 1 binair cijfer (0 of 1)

NIBBLE binair getal uit 4 bits **1011**

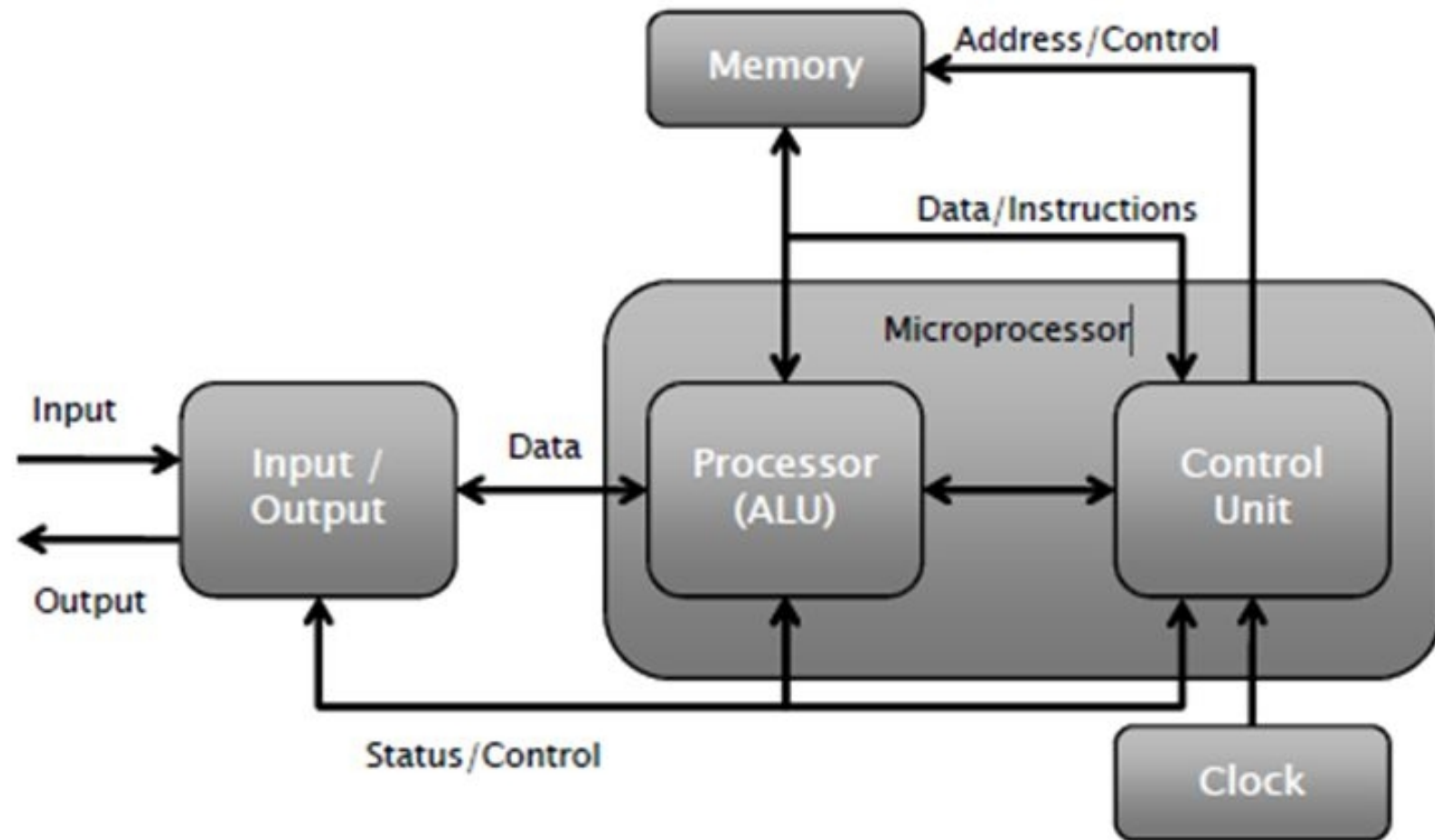
BYTE binair getal uit 8 bits (2 nibbles)
0101 1101

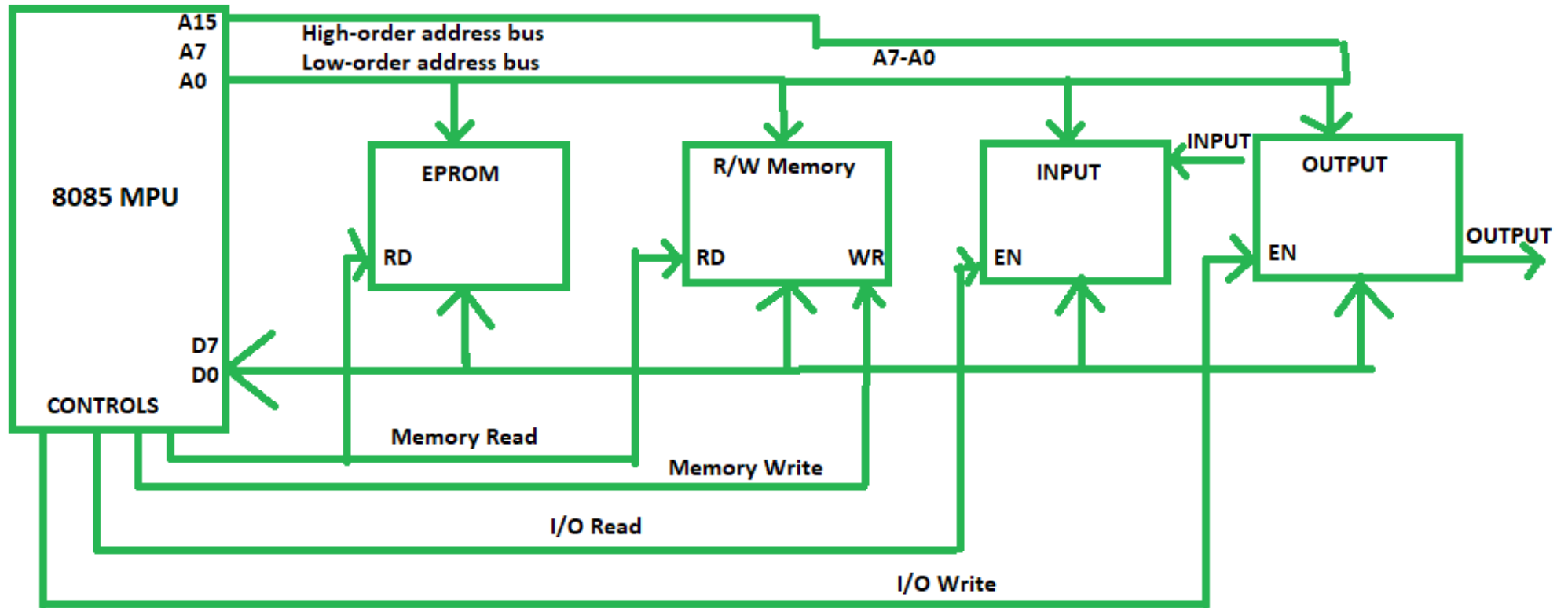
WORD binair getal uit 16 bits (2 bytes) (*meestal*)
10011100 10110101

2. Bouw en werking van een computer



Microcomputer





A Microcomputer System

.....
LDA
05H
BRA
07H
.....
ADD
06H

Programma (assembler

Adres inhoud

.....
0001 0010 10000110
0001 0011 00000101
0001 0100 00100000
0001 0101 00000111
0001 0110
.....
0001 1100
0001 1101 10001011
0001 1110 00000110



Werking van een CPU

- In het geheugen zijn instructies (=opcode) en data opgeslagen
- de programmateller (PC) start op adres 0000 0000
Daar moet een opcode staan.
- de ALU voert die code uit en verhoogt de programmateller
- de volgende opcode wordt in de ALU geladen en uitgevoerd
- de programmateller wordt verhoogd

enz..... enz... enz.....

Opmerking:

Instructies en data in hetzelfde geheugen: **Neumann-architectuur**

Instructies en data in gescheiden geheugens: **Havard-architectuur**

3. Microprocessor / Microcomputer / Microcontroller

Microprocessor:

- * is de CPU (CPU) van een computersysteem op één chip
- * uitgevoerd in verschillende bitbreedten: 4, 8, 16, 32... bit
- * in verschillende families (Z80, 8085, 8086, pentium,....) met eigen instructieset (RISC of CISC);

Microcomputer:

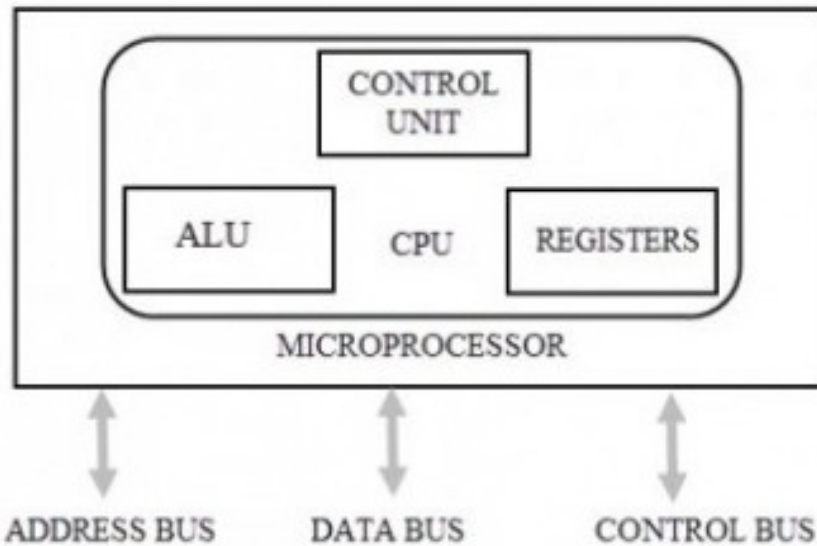
- * computer gebouwd rond een microprocessor
- * met discrete geheugenchips en discrete ondersteunende chips (systeembus, parallele en/of seriële I/O-schakelingen, interrupt-controller, enz...)
- * de samenstellende onderdelen zijn verbonden met een adres-, data- en controlebus.
- * inzetbaar in een groot aantal, niet vooraf gedefinieerde taken.



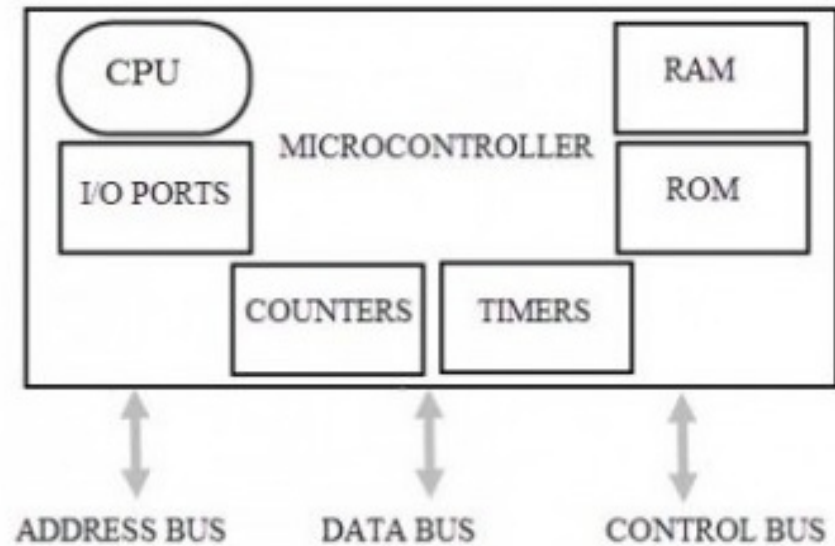
Microcontroller:

- * microcomputer met de samenstellende delen (CPU, geheugen, poorten, timers, ADC, enz. op één chip)
- * hoofdzakelijk gebruikt voor een specifieke applicatie in meet- en regeltoepassingen
- * doorgaans met beperkt het geheugen
- * busstructuur extern niet toegankelijk
- * communicatie loopt via programmeerbare poorten
- * meestal intern Harvard-architectuur
- * twee grote families: PIC en AVR (vroeger: 8051 en co)

Microprocessor



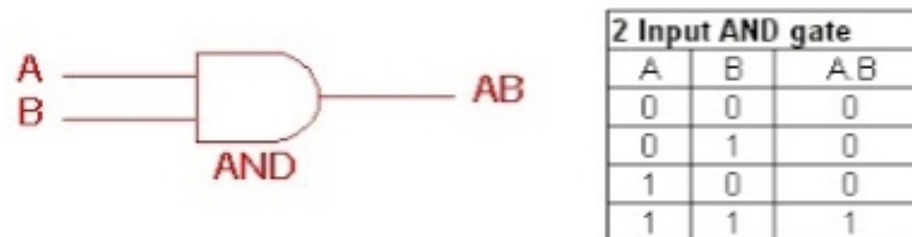
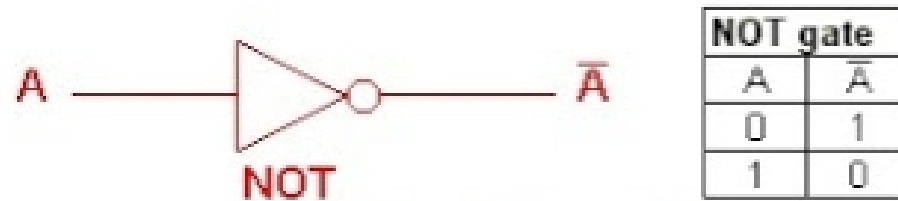
Microcontroller



Microprocessor vs Microcontroller by EEEPROJECT.COM

4. Logische bewerkingen

- A, B, C,... zijn binaire variabelen (waarden 0,1 of L,H)
- logische bewerkingen: NIET, EN, OF, EXOF (NOT, AND, OR, EXOR)
- resultaat logische bewerkingen weergegeven door **waarheidstabellen**





2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1



2 Input EXOR gate		
A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Logische bewerkingen op binaire woorden worden **bitgewijze** toegepast:



– Logische bewerkingen op binaire woorden worden **bitgewijze** toegepast:

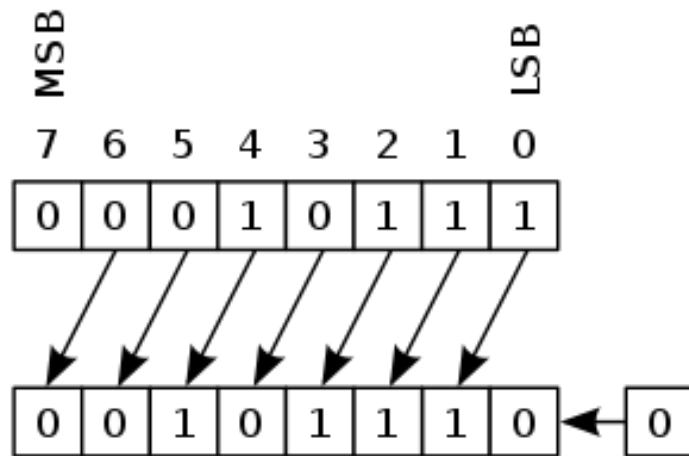
NOT: 10101110 wordt 01010001 (complementbewerking)

AND:
10101110 AND 01010010 wordt 00000010

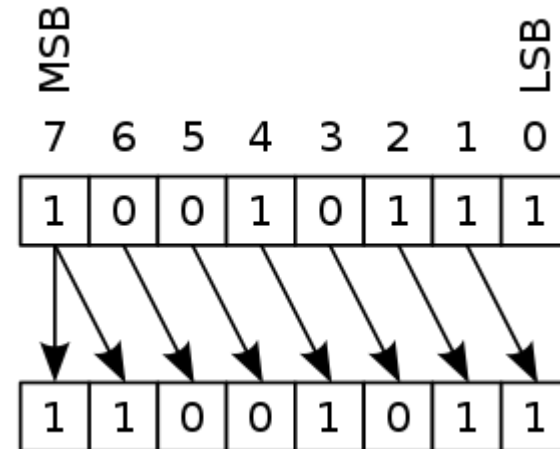
OR:
10101110 OR 01010010 wordt 11111110

EXOR:
10101110 EXOR 01010010 wordt 11111100

- schuifbewerkingen (shift of rotation)



LEFT shift is vermenigvuldigen met 2!



RIGHT shift is delen door 2 in 2-complement

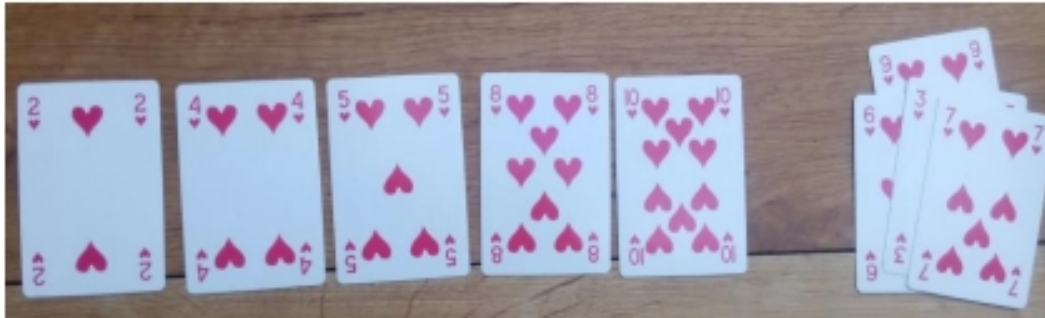
6. Programmeren en programmeertalen

Een programma is een groep opdrachten in een bepaalde volgorde om een wel omschreven taak uit te voeren

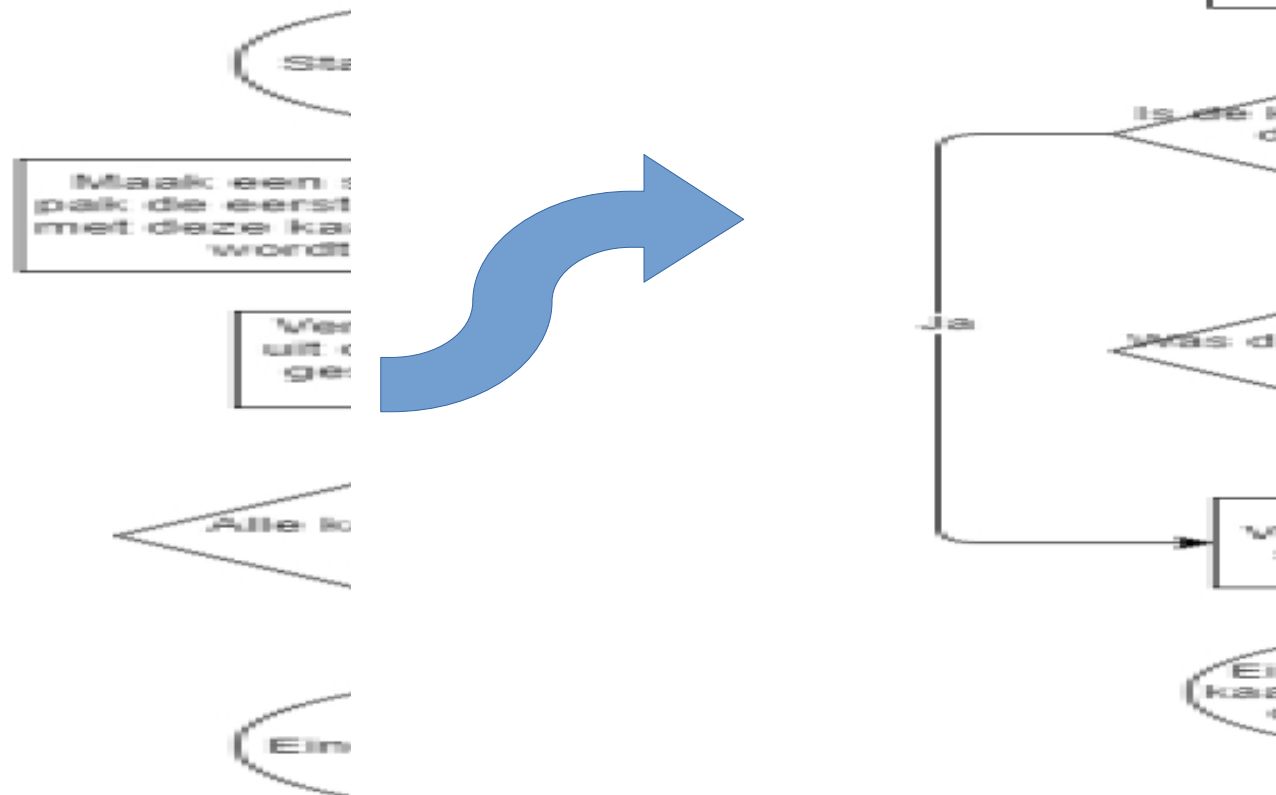
Maken van een computerprogramma

stap	nodig
1. formuleer het probleem	pen en papier en... creativiteit
2. formuleer een manier van oplossen (algoritme)	idem
3. schrijf een lijst van opdrachten in een program- meertaal (= broncode)	editor (platte ascii!) programmeertaal
4. compileer de broncode en link met een bibliotheek (=objectcode)	compiler ev. linker!
5. test de objectcode	geduld en volharding
6. los programmeerfouten op (=debuggen)	debugger

Probleem: sorteer kaarten in oplopende volgorde



Algoritme



Programmeertalen: de babylonische spraakverwarring

Er bestaan honderden programmeertalen:

1ste generatie: **machinetaal**: in binaire opcode programmeren

2-de generatie: **assembler**: in mnemonics direct in machinetaal omzetbaar

3-de generatie: procedurele talen zoals cobol, basic, pascal, C, C++

4-de generatie: talen met een hoger abstractieniveau zoals Progress, SQL

```
VAR
  ND, SC   : CARDINAL;
  KFL[10]  : (idle, ready);
  KDA[11]  : [0..9];
  DDA[13]  : (closed, open);
BEGIN
  LOOP
    SC := 0;
    FOR ND := 1 TO 3 DO
      REPEAT UNTIL KFL = ready;
      SC := SC * 10 + KDA;
      KFL := idle;
    END (* FOR *);
    IF SC = 321
      THEN DDA := open
    END (* IF *);
  END (* LOOP *)
END
```

Een IDE

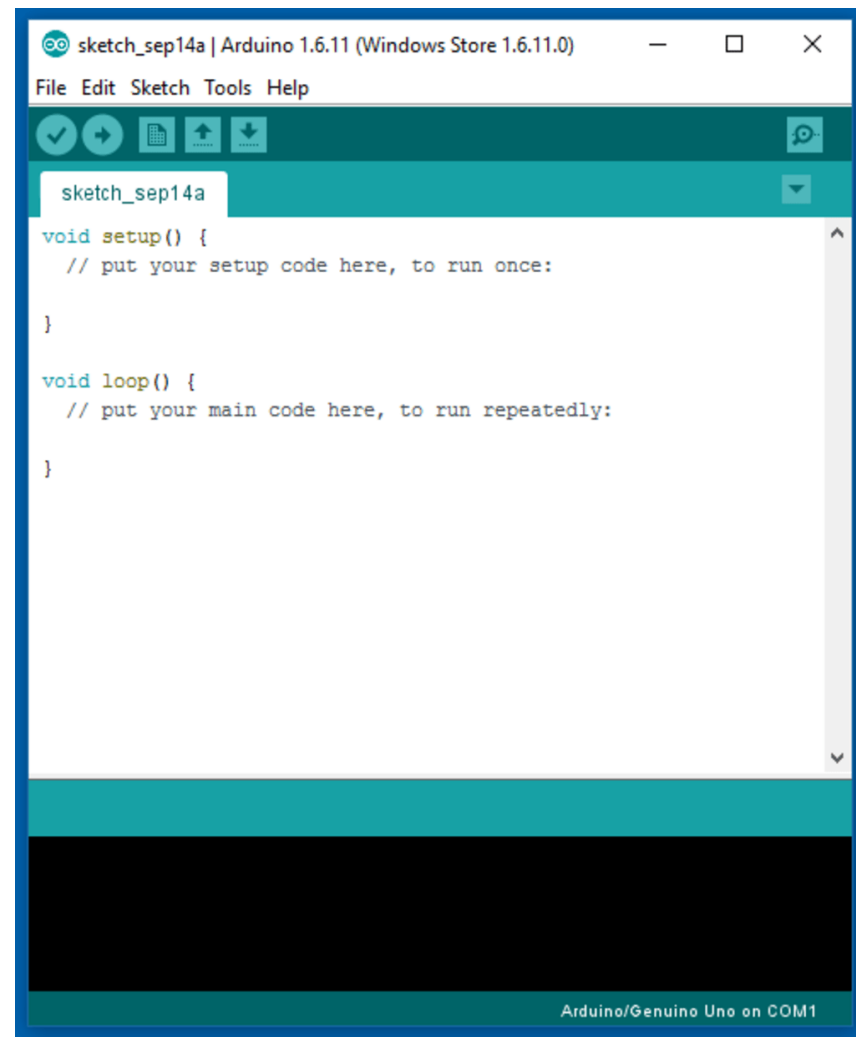
(Integrated development environment)

Een IDE bestaat minstens uit:

- * een editor
- * een compiler/linker
- * documentatie

eventueel uitgebreid met:

- * een debugger
- * grafische interface
- * hulpmiddelen zoals versiebeheer





Inhoud deel 3

1. A \rightarrow D en D \rightarrow A conversie

2. communicatie

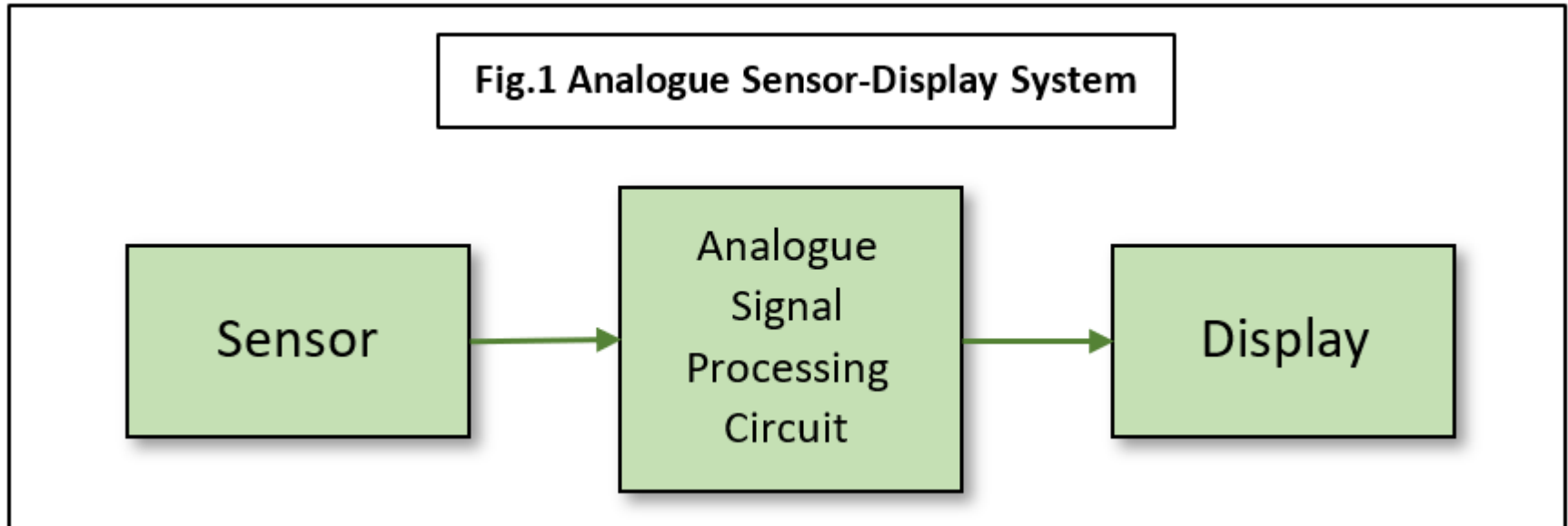
- a. parallel
- b. serieel

3. communicatiesystemen en protocollen

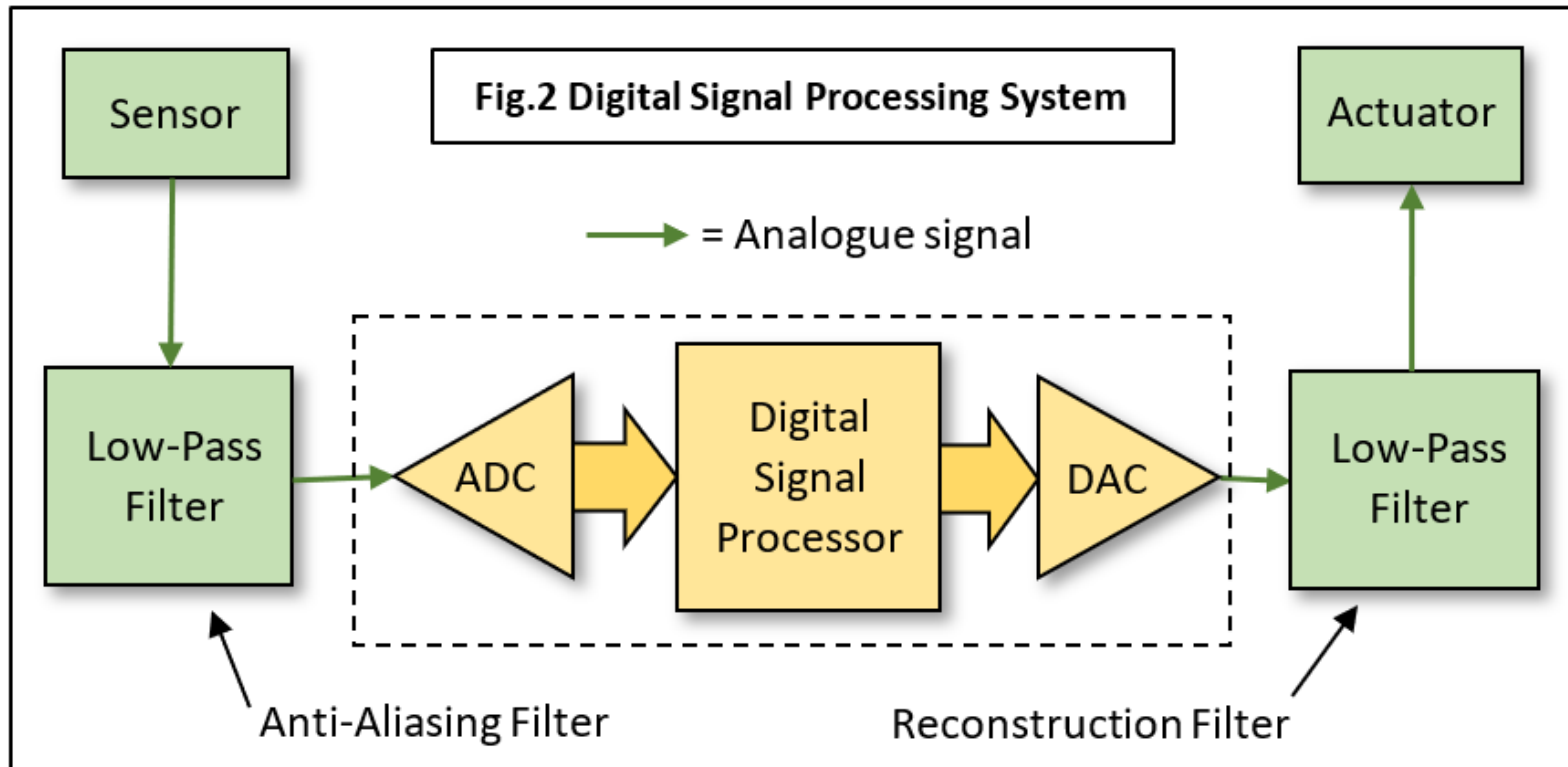
- a handshaking
- b RS-232
- c I2C



1. Analooq wordt digitaal (en omgekeerd)

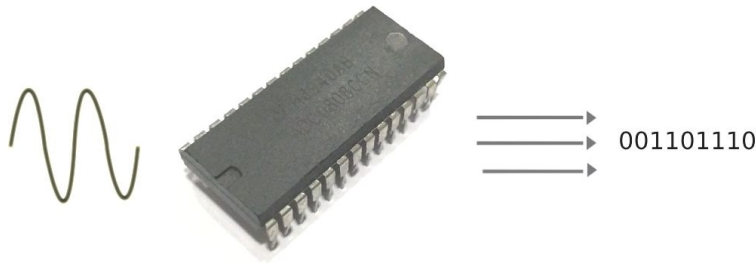


- In de ‘echte’ wereld veranderen alle fysische grootheden op continue wijze
- Een sensor zet de waarde van een fysische grootheid om in een overeenkomstige elektrische spanning (= analoog signaal)
- Na (eventuele) bewerking wordt dit weergegeven op een analoog ‘display’



- In een digitaal systeem wordt het signaal voor verwerking omgezet in een binaire getallen via **bemonstering**
- Na bewerking kunnen de binaire getallenterug omgezet worden in een analoog signaal om een **actuator** aan te sturen

Analog to Digital Converters



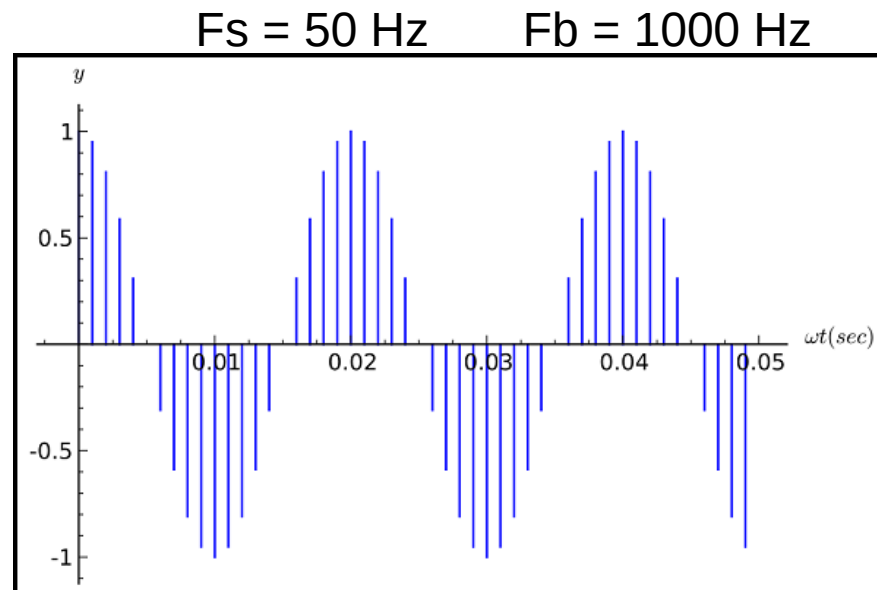
Bemonstering:

- bemonsterfrequentie
- resolutie: bitbreedte

Stelling van Nyquist:

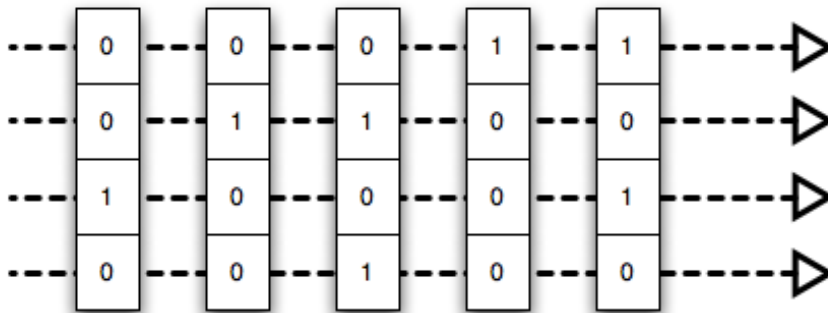
De bemonsterfrequentie moet minstens 2 x zo hoog zijn als de hoogste frequentie in het signaal

Daarom: laagdoorlaatfilter bij complexe signalen om 'aliassen' te vermijden



2. communicatie tussen digitale apparaten

a. PARALLELE



Groepen van bits worden **gelijktijdig** overgedragen (meerdere lijnen)

b. SERIEEL



Groepen van bits worden **bit per bit** over **1** lijn overgedragen

3. Communicatiesystemen en protocollen

a. Parallele communicatie

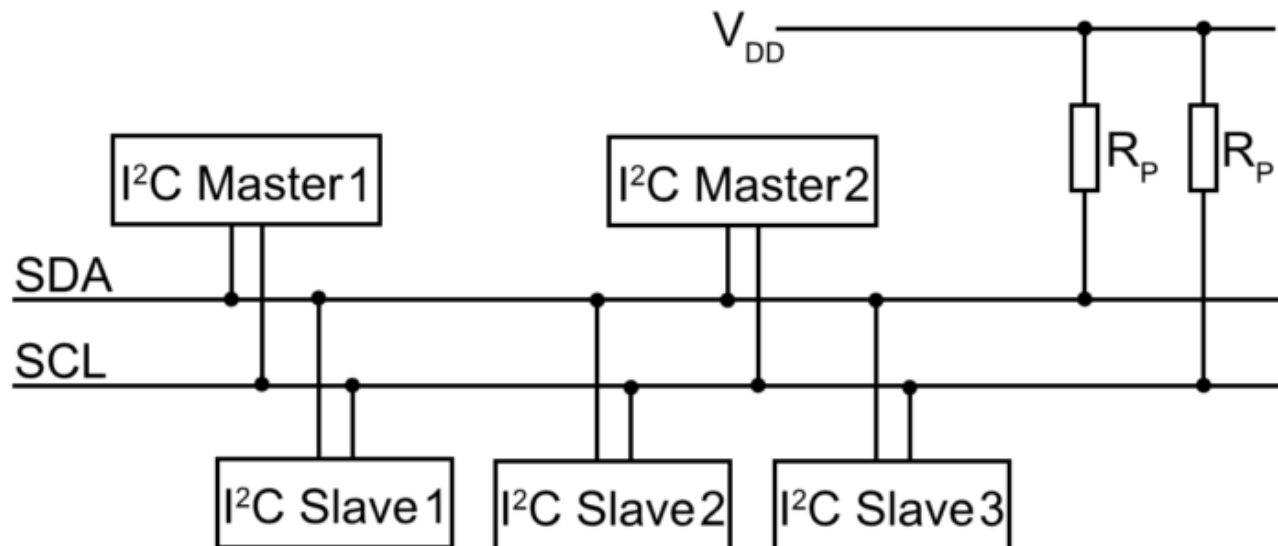
- * simplex, half- en full duplex
- * handschaking: softwarematig: XON en XOFF(CTRL-P/Q)
hardwarematig: ACK, BUSY, STROBE

b. Seriele communicatie (asynchroon)

- * simplex, half en full duplex; RXD en TXD
- * baudrate (bitbreedte/teken x aantal tekens/seconde)
- * start- en stopbits
- * spanningsniveau: RS-232 en TTL
- * RTS en CTS – DTR en DSR

c. synchrone seriële communicatie: I2C

BUS: 2 lijnen: SDA (data) en SCL (clock)
gebruikt een 7 bit adres
maximaal 128 adressen waarvan er 12 gereserveerd zijn
multi-master, multi-slave
pull-up-weerstanden zijn vereist!

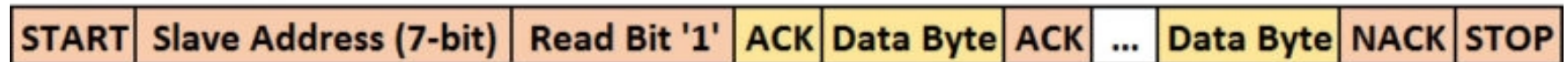


PROTOCOL:


Write data to slave



Read data from slave



 From Master to Slave

 From Slave to Master

ACK = Acknowledge (SDA Low)
NACK = Not Acknowledge (SDA High)

START en STOP

Terwijl SCL hoog is zijn start en stop gedefinieerd als stijgende/dalende flank van SDA

ADRES en R/W

Na START volgt het adres gevolgd door R/W-bit op de SDA-lijn

DATA Zenden (R/W=0):

Na ACK ('0') door slave op SDA-lijn zendt master een databyte
slave antwoordt na elke databyte met ACK ('0')

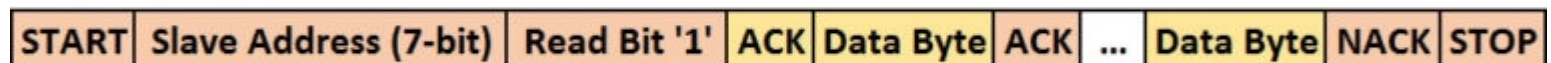
lezen (R/W=1)

Na ACK op uitgezonden adres leest master een databyte en stuurt een ACK. Als het de laatste byte is stuurt de master een NACK ('1') op de SDA-lijn


Write data to slave



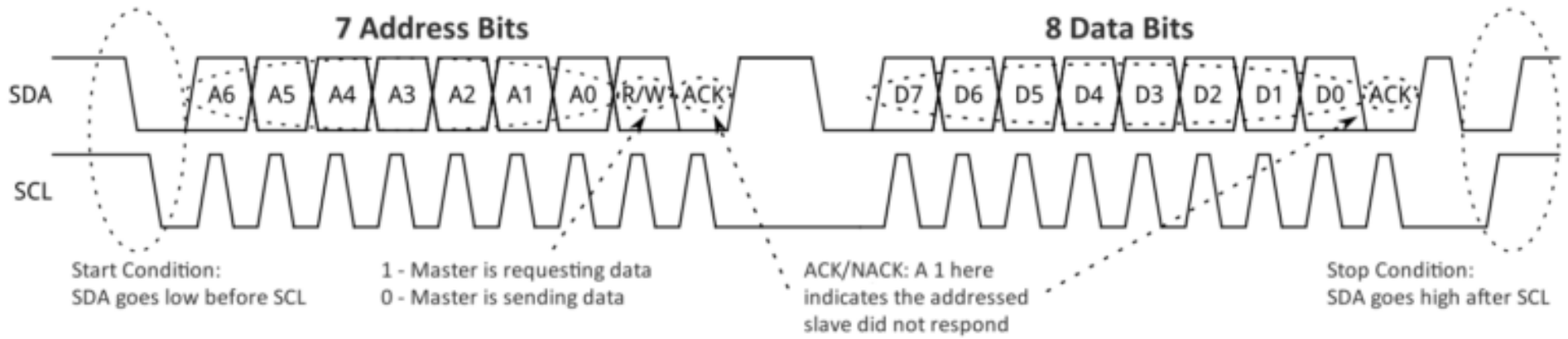
Read data from slave



 From Master to Slave

 From Slave to Master

ACK = Acknowledge (SDA Low)
NACK = Not Acknowledge (SDA High)





**En toen kwam er een varken met een lange
staart en het verhaal is uit**