1 Digitale techniek Digitale signalen

1.1 Voorstelling van informatie

1.1.1 Informatieverwerking

Overdracht van **informatie** kan plaats vinden onder verschillende vormen:

- gesprek tussen personen
- raadplegen van een website
- kijken naar een televisie-uitzending
- een microcontroller leest een sensorsignaal
- ...

Tijdens elke **informatie-overdracht** kunnen we steeds volgende stappen onderscheiden:

- het *opnemen* van de aangeboden informatie
- het *onderzoeken* van die informatie
- het *behandelen* van die informatie
- het *bewaren* van die informatie
- het *verspreiden* van die informatie

Dit proces van opname, onderzoek, behandeling, registratie en verspreiding zijn typische aspecten van de **dataverwerking** of **dataprocessing**. *Elektrische signalen* zijn uitermate geschikt als drager van informatie. Signaalverwerking komt dus ook neer op informatieverwerking.

1.1.2 Analoge en digitale informatie

Tegenwoordig slaat de reclamewereld ons ermee rond de oren: *digitale* smartphones, *digitale* MP3/MP4 muziekopnames, *digitale* animatiefilms, *digitale* spelconsoles, *digitale* fototoestellen, *digitale* snelheidsmeters,... Kortom, wil je het goed doen? Doe het dan *digitaal*! Wat bedoelen ze nu met 'digitaal' en 'analoog'?

Wanneer we over **analoge technieken** spreken, moet er een zekere *analogie* zijn tussen het ingangssignaal en het uitgangssignaal. Beschouw nu een bij uitstek analoog toestel: de goeie ouwe *platendraaier* (pick-up).

△ Fig.1.1 Terug van nooit echt weg geweest: afspelen van muziek op vinyl met een ouderwetse draaitafel... Bron: http://zoom.nl

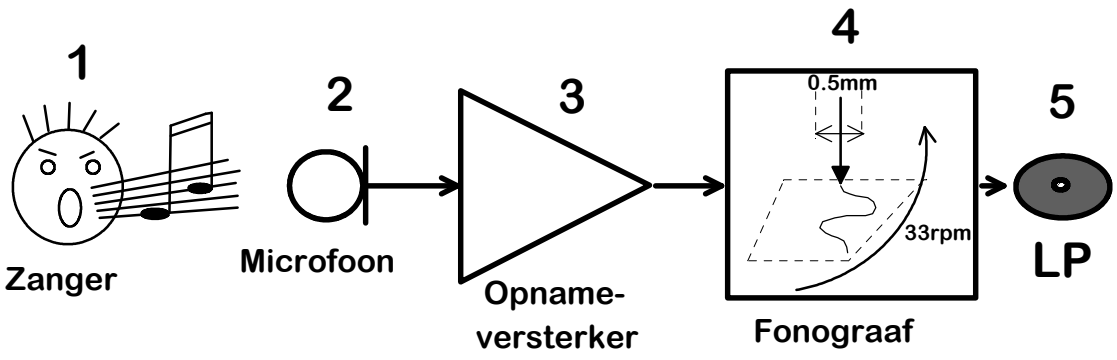
Een analoog systeem: de pick-up

Het 'vinyltijdperk'[1] ligt nog niet zo ver achter ons. Hoe werd de registratie van het mindere broertje van de CD toen gerealiseerd? Laten we een 'analogue recording session' eens stap voor stap bekijken. Tussen haakjes, sommige muziekliefhebbers blijven zweren bij vinyl. De muziek zou beter tot z'n recht komen en natuurlijker klinken... Het moet gezegd dat een nieuwe LP bij de eerste afspeelsessie met een professionele platenspeler nauwelijks te onderscheiden is van een CD!

[1] De eerste vinyl langspeelplaat (lp) werd officieel geïntroduceerd in 1948 in de Verenigde Staten en kon maximaal een halfuur muziek per kant weergeven.

Analoge registratie

Een zanger zingt (stap1, fig.1.2). Dit is een ANALOOG gegeven. Hij stoot lucht uit die zijn stembanden laten *trillen*. Via de mondholte worden deze trillingen versterkt. Deze trillingen beschrijven een golfpatroon eenmaal ze de lippen verlaten hebben. De luchtdruk-variaties planten zich voort doorheen het medium lucht. De trillingen van onze zanger worden opgevangen door een membraan in de *microfoon* (stap2). Hoe harder onze zanger zingt, hoe verder het microfoonmembraan uitwijkt. Hoe hoger hij zingt, hoe sneller dit uitwijken van dat membraan zal gebeuren. Een stijgende frequentie resulteert immers in meer trillingen per seconde. De uitwijkingen van dat microfoonmembraan doet een elektrische spanningssignaal ontstaan. Doordat het membraan een spoeltje op en neer doet gaan in een magnetisch veld, wordt er een spanning binnen in de microfoonspoel geïnduceerd. Hier hebben we te maken met een ANALOGIE: de spanning zal *evenredig* op en neer gaan met de toonhoogte en de amplitude zal toenemen met de geluidsdruk.



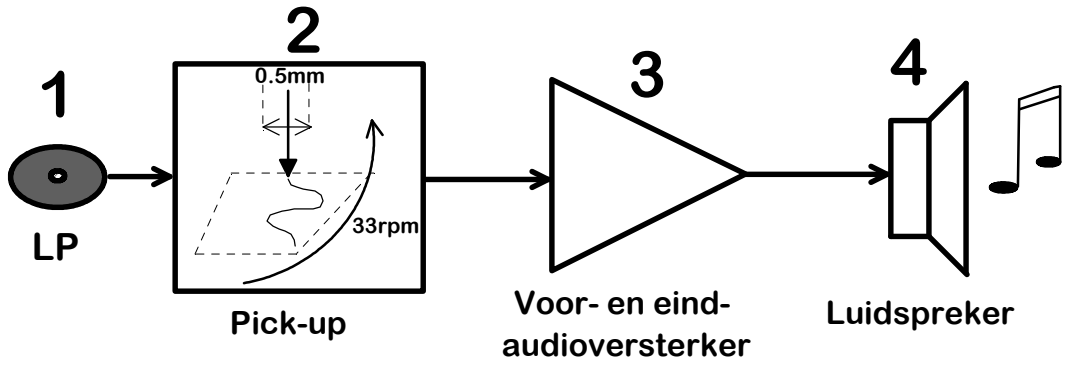
△ Fig.1.2 | Analoge registratie van de menselijke stem.
Bron: W.Van Wichelen

De *opnameversterker* (stap3) zal de analoge microfoonspanning de nodige zwaai geven zodat de stuurspanning kan dienen om de analoge informatie met een naald in de matrijs te kunnen krassen. Ook hier is er een ANALOGIE: het uitgangssignaal van de versterker zal dezelfde vorm hebben als het microfoonsignaal (ingang). Het is dus belangrijk dat de versterker het signaal *onvervormd* versterkt. De stuurspanning heeft dus dezelfde vorm als de zang van onze zanger. De beweging van de naald zal dus ook *dezelfde vorm* beschrijven. Terwijl de naald trilt op het ritme van de zang, zal ze een kras maken in een matrijs die ronddraait tegen ca. 33 toeren per minuut (stap4).

We hebben nu een plaat gemaakt. De matrijs is de *master* en we kunnen in principe nu onbeperkt kopieën maken in vinyl (stap5). Het is uiteraard de bedoeling dat we het originele signaal (zang) terug kunnen recupereren. Dit herwinnen van de zang gebeurt natuurlijk tijdens het afspelen van het vinyl met een platenspeler.

Analoge weergave

De naald zet de *mechanische* trillingen om in *elektrische* trillingen (fig. 1.3, stap1,2). De spanning die de platenspeler aflevert bezit dezelfde vorm als het microfoonsignaal, en dus een mooie reproductie van de zang. Het signaal dat de pick-upnaald produceert is van grootte-orde enkele millivolts. Dit signaal bezit niet het vermogen om *rechtstreeks* een luidspreker aan te sturen. Daarom versterken we het muzieksignaal tot een meer werkbaar signaal (stap3). Dit versterken moet liefst zonder vervorming gebeuren, anders kunnen we het origineel nooit terugwinnen. Het uitgangssignaal van de audioversterker levert voldoende energie om via de luidsprekerspoel een membraan in beweging te zetten (stap4) en alzo muziek te doen weerklinken. Dit bewegen gebeurt op het ritme en de sterkte van het originele muzieksignaal. We zien weerom dat er bij elke stap ANALOGIE bestaat.



△ Fig.1.3 Analoge weergave van de menselijke stem.
Bron: W.Van Wichelen

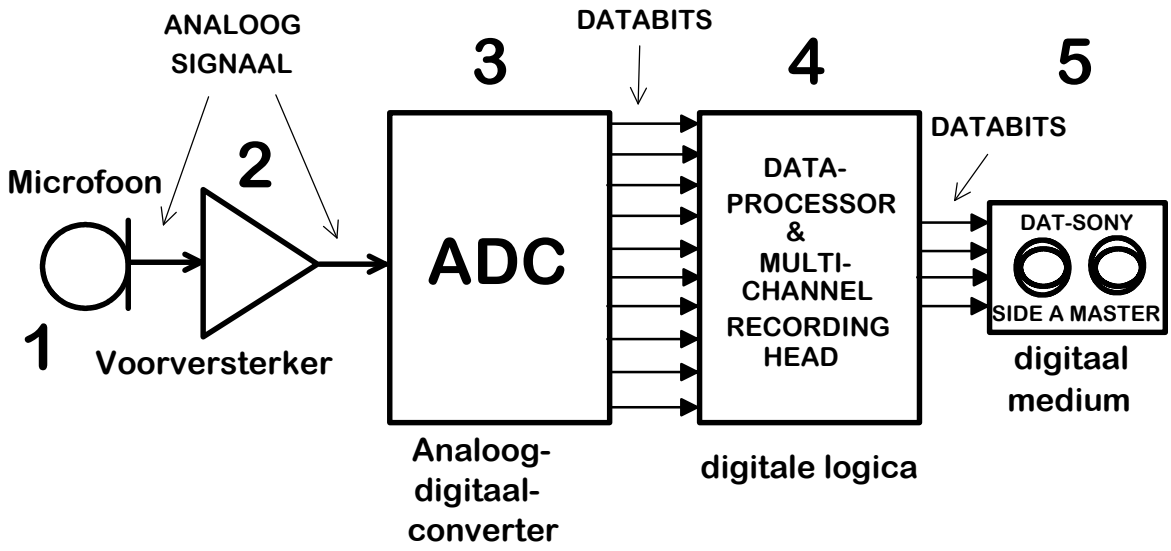
Definitie

We spreken van **ANALOGE INFORMATIE** wanneer:

- er een *analogie* bestaat tussen de fysische grootheid en het elektrische signaal;
- dit elektrische signaal *continu* is: als er geen signaal aanwezig is, is er geen informatie;
- bij *vervorming* (ruis, overspraak,...) van dit signaal ook de informatie *wijzigt*.

Een digitaal systeem: A/D-omzetting

Muziek is per definitie een analoog gegeven. Het analoge microfoonsignaal wordt onvervormd versterkt tot een niveau dat bruikbaar is voor de **analoog/digitaal converter** (stap1,2). Deze A/D-omzetter zorgt voor het *digitaliseren* van ons ingangssignaal (stap3) en zet het *continue* (analoge) signaal om in *getallen*. Deze 'getallen' worden in de techniek *digits* genoemd. Dit omzetten gebeurt natuurlijk volgens een bepaald patroon, vastgelegd in een zg. 'conversietabel'. In digitale systemen werkt men met een **binair** in plaats van met een decimaal stelsel. De digits kunnen slechts de waarde '1' of '0' aannemen. Een combinatie van deze digits levert een binair getal op dat een maat is voor ons analoge ingangssignaal. Om de conversie mogelijk te maken 'proeft' de A/D-converter op *vaste* tijdsstippen het continue ingangssignaal. Dit 'proeven' heet in de literatuur **bemonsteren** of **sampelen**.



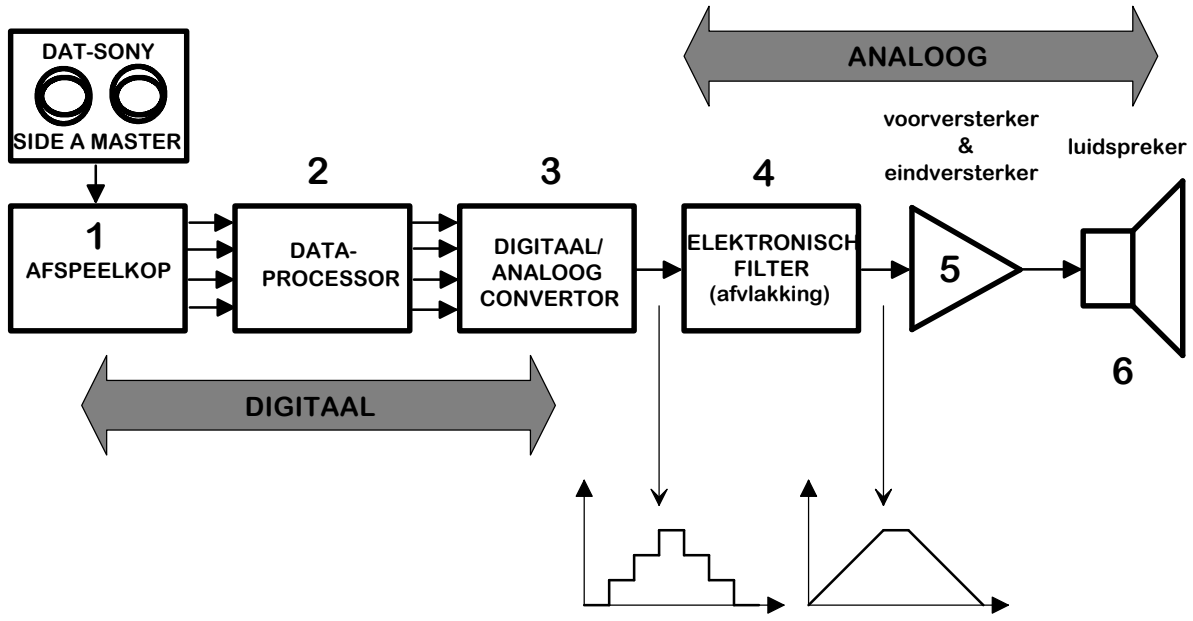
△ Fig.1.4 | Digitaliseren van muziek met een DAT-recorder.
Bron: W.Van Wichelen

Het analoge (continue) signaal is nu omgevormd tot een **datastroom** van *enen* en *nullen*. Dit is op een gestandaardiseerde manier gebeurd, zodat we de originele informatie later terug kunnen winnen. Het is de bedoeling deze digitale informatiestroom op een magnetische tape te krijgen (stap4). Daar zorgt een **processor** voor: hij herschikt op een bepaalde (wiskundige) wijze deze informatie, zodat ze via meerkanaals opneemkoppen op de DAT-tape[2] terecht komt (stap5).

[2] De digital audio tape (DAT) is een cassette-systeem dat werd ontwikkeld door Sony in 1987 voor professionele geluidsopnamen. Tegenwoordig wordt het nog toegepast als back-up medium in computersystemen.

Reproductie via D/A-omzetting

Een multisporenkop leest de digitale data af (fig.1.5, stap1) en levert deze dan (parallel of serieel) af aan een dataprocessor. De dataprocessor ordent en schikt de data zodat deze klaar is om verwerkt te worden door de D/A-converter. Deze omzetter vertaalt de aangeboden digitale informatiestroom naar een analoge stapvormige signaalspanning (stap3). Een laagdoorlaatfilter verwijdert de stijle overgangen zodat het herwonnen signaal bij benadering gelijk is aan het originele opgenomen analoge signaal (stap4). Het herwonnen signaal wordt versterkt en aangeboden aan een luidspreker en weldra wordt de kamer gevuld met (ruisvrije) analoge klank (stap5,6). De kwaliteit van deze operatie hangt af van het aantal bits per sample en de samplefrequentie: streven naar kwaliteit heeft dus grote datastromen tot gevolg. Een courante samplefrequentie is 44,1kHz. Meestal digitaliseren we muziek met 16 bits A/D-convertoren. Een standaard CD duurt ca. 74 minuten wanneer de volle capaciteit van 650 MB wordt benut[3].



△ Fig.1.5 | Principe van DA-conversie.
Bron: W.Van Wichelen

[3] In 1980 legde Sony en Philips de standaard van de compact disc (CD) vast. Er bestaat een mythe rondom de specificaties: Sony stond erop dat de speelduur 74 minuten moest worden, opdat ook de negende symfonie van Beethoven op een enkele CD zou passen.

Definitie

We spreken van **DIGITALE INFORMATIE** wanneer:
- er slechts 2 niveau's mogelijk zijn: *hoog* (1) en *laag* (0);
- de informatie *niet* vervat zit in de vorm, amplitude en frequentie van het elektrische signaal maar in de opeenvolging van nullen en enen;
- de informatie voorgesteld wordt m.b.v. digits (*binaire getallen*).

1.1.3 Voordelen van digitale informatie

In vergelijking met analoge informatie bezit het gebruik van digitale informatie niet te miskennen voordelen.

- Digitale signalen zijn *onbeperkt* te **reproduceren**: uit ervaring weet je dat je zeer gemakkelijk digitale kopieën kan maken van een origineel bestand.
- Analoge signalen zijn *moeilijk* te **reproduceren**: elke analoge kopie gaat gepaard met kwaliteitsverlies.
- Digitale signalen bezitten een *gedefinieerde* **nauwkeurigheid**: vervorming van het digitale signaal - binnen bepaalde grenzen - leidt niet tot kwaliteitsverlies.
- Digitale signalen kunnen *eenvoudig* **opgeslagen** worden: in de begindagen van de homecomputer duurde het minuten om enkele honderden kilobytes op te slaan, tegenwoordig versluisen we in enkele seconden meerdere gigabytes naar opslagservers in de 'cloud'!

△ Fig.1.6 Opslag van digitale informatie op harde schijven gebeurt door magnetisatie en wordt nog veel toegepast. Bron:https://pxhere.com

1.2 Talstelsels

In essentie is elke computer een *rekenmachine*. Het is dus belangrijk dat je als ICT-technicus over weg kan met de gebruikelijke **talstelsels** waar computersystemen mee rekenen.

1.2.1 Het decimaal talstelsel

Het *decimale stelsel* is zeker het meest gebruikte talstelsel om getallen voor te stellen. Hierin worden 10 symbolen 0, 1, 2, 3, 4, 5, 6, 7, 8 en 9 gebruikt die ondubbelzinnig de cijferwaarde weergeven van een getal in decimale voorstelling. Een elektronisch toestel dat bewerkingen uitvoert in dit getallensysteem, zou over elementen moeten beschikken met 10 van elkaar duidelijk te onderscheiden toestanden, die elk één van de 10 decimale cijfers voorstelt. Dergelijke elementen - hoewel technisch realiseerbaar - vereisen uiterst complexe schakelingen.

Laten we eerst ons vertrouwde decimale getallensysteem nog eens te bekijken. Het aantal cijfersymbolen noemt men het *grondtal*. Een decimaal getal wordt voorgesteld door een aantal symbolen achter elkaar te plaatsen. De plaats van een bepaald symbool in de cijfercombinatie is maatgevend voor haar belangrijkheid, ook *gewicht* genoemd.

10^3	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}	← gewicht
1	**9**	**7**	**3**	**2**	**0**	**6**	← symbool
1000	900	70	3	0,2	0,00	0,006	← cijferwaarde

△ Fig.1.7 Opbouw van een decimaal getal.
Bron: W.Van Wichelen

Om een decimaal getal weer te geven, plaatsen we de coëfficiënten tussen haakjes en daarbuiten de index 10: $(1973{,}206)_{10}$

1.2.2 Het binair talstelsel

Hedendaagse computersystemen bevatten sequentiële circuits die slechts **twee** schakelwaarden kennen. De berekeneningen gebeuren dus op een **binaire** manier. Binaire cijfers worden intern omgezet in twee spanningsniveau's die voldoende ver uit elkaar liggen. Zo ontstaat een robuuste schakelomgeving zodat computers betrouwbare berekeningen kunnen maken.

Bij binaire getallen is het *grondtal* 2, d.w.z. dat in een binair getal een cijfer slechts 2 verschillende waarden kan aannemen nl. '0' en '1'. In de digitale techniek kunnen deze 2 cijfers voorgesteld worden door twee sterk van elkaar te onderscheiden spannings-niveaus, zoals 0 volt = '0' en +5 volt = '1'. Ook in het binaire talstelsel vertegenwoordigt de *plaats* van het binaire cijfer een zeker *gewicht* dat gebaseerd is op een bepaalde macht van 2.

LSB en MSB

- Het meest rechtse cijfer is de *minst belangrijke bit.* Het wordt aangeduid met **'LSB'**, de afkorting van *Least Significant Bit*.
- De meest linkse cijfer is de *meest belangrijke bit* en wordt aangeduid met **'MSB'** van *Most Significant Bit*.

2^6	2^5	2^4	2^3	2^2	2^1	2^0	← gewicht
1	**0**	**0**	**1**	**0**	**0**	**1**	← symbool
64	32	16	8	4	2	1	← cijferwaarde

△ Fig.1.8 Opbouw van een binair getal.
Bron: W.Van Wichelen

In figuur 1.8 ontdekken we dat het decimale getal: 64 + 8 + 1 = $(73)_{10}$ overeenkomt met het binaire getal $(1001001)_2$.

Bits en bytes

De combinatie van nullen en enen wordt ook een binair *woord* genoemd. Bestaat dit binair woord uit **8 bits**, dan spreken we van een woordlengte van **1 byte**[4]. Acht bits bieden 2^8 = **256** combinatiemogelijkheden.

Nog veel voorkomende woordlengtes in de digitale techniek zijn:

- 2 bytes = 16 bits
- 4 bytes = 32 bits
- 8 bytes = 64 bits
- 1 kilobyte = 1024 bytes = 1024 x 8 bits
- 1 megabyte = 1024 x 1024 bytes
- 1 nible = 4 bits

[4] Het woord 'byte' is een aanpassing van het woord 'bite' (hapje) om verwarring met bit te voorkomen. Het is bedacht in 1956 tijdens de ontwikkeling van de IBM Stretch-computer.

1.2.3 Het hexadecimaal talstelsel

Een veel gebruikt talstelsel in de computertechniek is het zestientallig of *hexadecimale* talstelsel. Het bezit **16** cijfersymbolen. De coëfficiënten zijn: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E en F. Waarbij A een 'tien' voorstelt, B een 'elf', C een 'twaalf', D een 'dertien', E een 'veertien' en F een 'vijftien' voorstelt. Tellen in het hexadecimaal rekenstelsel gebeurt als volgt: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13,..., 19, 1A, 1B,..., 1F,20, 21, 22,...

Algemeen kan een hexadecimaal getal voorgesteld worden als een som van machten van 16. Dit talstelsel wordt veel toegepast om binaire getallen verkort weer te geven. Het hexadecimale stelsel wordt vrijwel alleen gebruikt door degenen die op laag niveau met computers werken. Dit is een gevolg van de binaire werking van de computer en de opbouw en werking van de geheugens in de computer. Praktisch kent het zijn nut bij microprocessors om o.a. de geheugenlocaties aan te duiden en data in te schrijven. Door de toepassing van hogere programmeertalen is kennis van het hexadecimale stelsel meestal geen directe noodzaak meer.

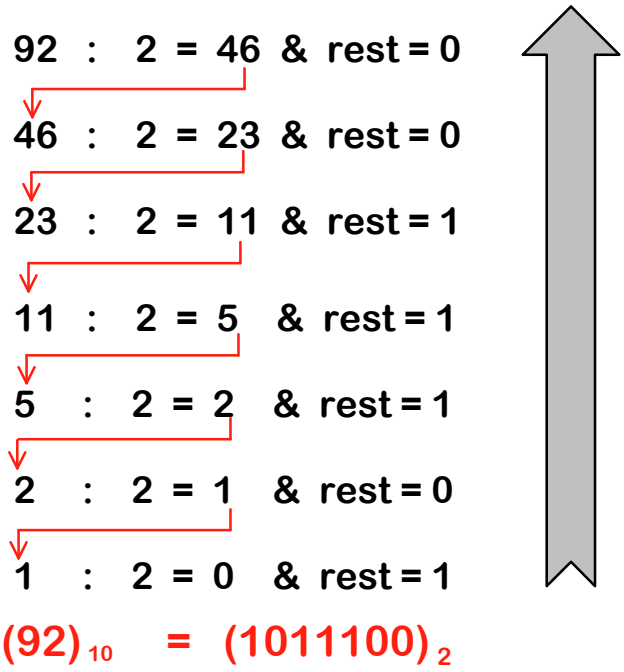
16^2	16^1	16^0	← gewicht
2	**B**	**F**	← symbool
256	16	1	← cijferwaarde

△ Fig.1.9 Opbouw van een hexadecimaal getal.
Bron: W.Van Wichelen

In figuur 1.9 ontdekken we dat het decimale getal: 2 x 256 + 11 x 16 + 15 x 1 = $(703)_{10}$ overeenkomt met het hexadecimale getal $(2BF)_{16}$.

1.2.4 Omzetting van decimaal naar binair

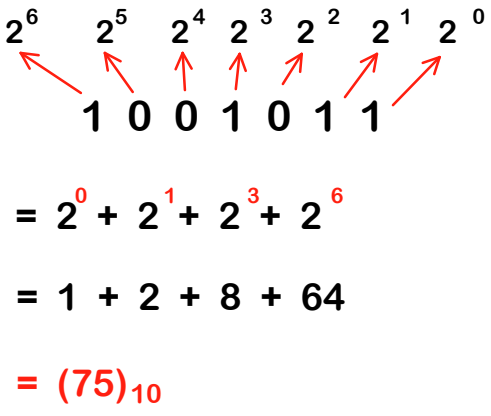
Figuur 1.10 laat een manier zien om het decimale getal $(92)_{10}$ om te zetten naar de binaire voorstelling:



△ Fig.1.10 | Omzetting van decimaal naar binair.
Bron: W.Van Wichelen

1.2.5 Omzetting van binair naar decimaal

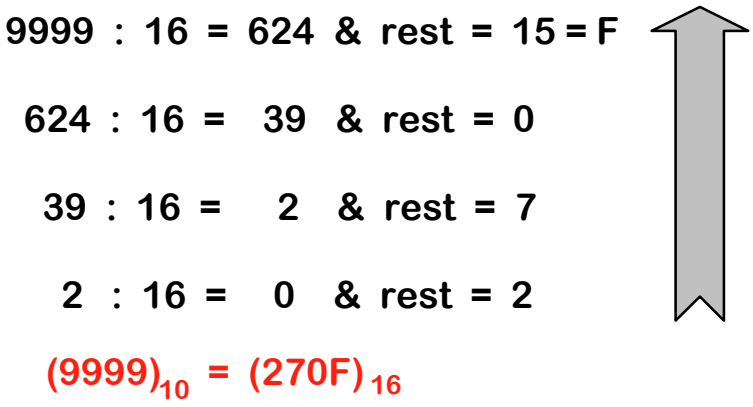
Figuur 1.11 toont hoe je het binaire getal $(1001011)_2$ omzet naar de overeenkomstige decimale waarde. Je telt gewoon de 'aanwezige' machten van 2 op:



△ Fig.1.11 | Omzetting van binair naar decimaal.
Bron: W.Van Wichelen

1.2.6 Omzetting van decimaal naar hexadecimaal

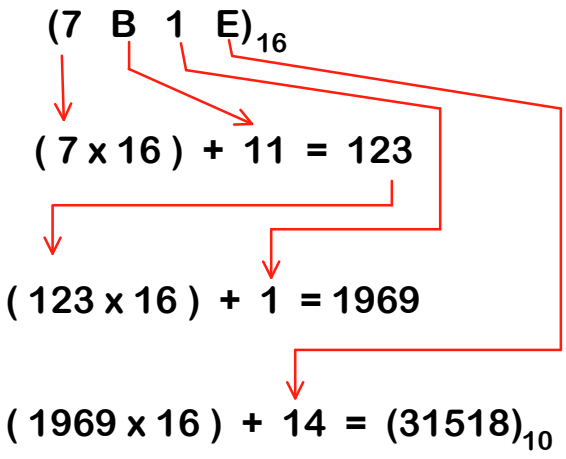
Figuur 1.12 toont hoe je het decimale getal $(9999)_{10}$ omzet naar de overeenkomstige hexadecimale waarde:



△ Fig.1.12 | Omzetting van decimaal naar hexadecimaal.
Bron: W.Van Wichelen

1.2.7 Omzetting van hexadecimaal naar decimaal

Figuur 1.13 toont een manier om hoe het hexadecimale getal $(7B1E)_{16}$ om te vormen naar de overeenkomstige decimale waarde:



△ Fig.1.13 | Omzetting van hexadecimaal naar decimaal.
Bron: W.Van Wichelen

1.2.8 Omzetting van binair naar hexadecimaal

In figuur 1.14 zetten we het binaire getal $(0010110011100101)_2$ om naar zijn hexadecimale tegenhanger. We verdelen het binaire getal in groepjes van 4 bits en vervangen de code door het hexadecimale symbool:

0010	1100	1110	0101
$(2)_{10}$	$(12)_{10}$	$(14)_{10}$	$(5)_{10}$
$(2)_{16}$	$(C)_{16}$	$(E)_{16}$	$(5)_{16}$
→	$(2CE5)_{16}$		

△ Fig.1.14 Omzetting van binair naar hexadecimaal.
Bron: W.Van Wichelen

1.2.9 Omzetting van hexadecimaal naar binair

Figuur 1.15 toont hoe je het hexadecimale getal $(FF16)_{16}$ omzet naar zijn binaire voorstelling:

F	F	1	6
$(15)_{10}$	$(15)_{10}$	$(1)_{10}$	$(6)_{10}$
$(1111)_2$	$(1111)_2$	$(0001)_2$	$(0110)_2$
→	$(1111111100010110)_2$		

△ Fig.1.14 Omzetting van binair naar hexadecimaal.
Bron: W.Van Wichelen

1.2.10 Negatieve binaire getallen

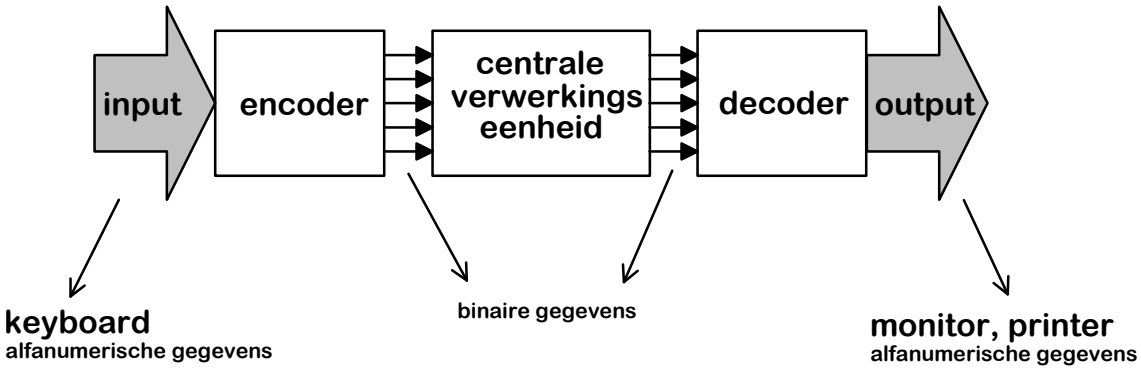
Opdat binaire computers *rekenkundige* bewerkingen zouden kunnen uitvoeren, is het noodzakelijk om een manier te vinden om *negatieve* getallen voor te stellen. We doen dit door een *extra* **tekenbit** toe te voegen aan het binaire getal.

Tekenbit

- De tekenbit plaatsen we uiterst *links* zodat het digitale systeem hem altijd zal herkennen.
- Tekenbit = 0 voor POSITIEVE getalen
  Voorbeeld: $(\mathbf{0}\ 1010)_2 = (\mathbf{+}10)_{10}$
- Tekenbit = 1 voor NEGATIEVE getallen
  Voorbeeld: $(\mathbf{1}\ 1010)_2 = (\mathbf{-}10)_{10}$

1.3 Digitale codes

Digitale computersystemen begrijpen enkel de taal van nullen en enen. Wij, mensen, verstaan enkel decimale getallen en **alfanumerische gegevens**. Alfanumerische gegevens zijn alle letter- en cijfertekens, maar ook de lees- en controletekens die je terug kan vinden op het toetsenbord van je computer. Daarom zal digitale elektronica de decimale en alfanumerische gegevens moeten omzetten in binaire informatie: dit noemen wij het *encoderen* of eenvoudigweg *coderen* van die informatie. Verderop bespreken we twee voorbeelden van digitale coderingen: de *BCD-code* en de *ASCII-code*.



△ Fig.1.15 Omzetting van alfanumerische gegevens naar binaire informatie bij computersystemen. Bron: W.Van Wichelen

1.3.1 BCD-code

Een **BCD-code** (**B**inary **C**oded **D**ecimal) is een **binaire** code om decimale getallen op te slaan. Elk cijfer van het getal wordt gecodeerd door een groep van vier bits die een binaire voorstelling zijn van het betrokken cijfer. Vroeger werd BCD-codering veel gebruikt om de elektronica te vereenvoudigen bij bevoorbeeld de weergave van cijfers met zevensegments-displays. Tegenwoordig maakt computersoftware meestal gebruik van andere coderingen. De decimale weergave van getallen worden meestal rechtstreeks vanuit deze coderingen berekend. Figuur 1.16 toont hoe je het decimale getal $(73)_{10}$ omzet naar de respectievelijke BCD-code:

7	3
$(0111)_2$	$(0011)_2$
→	$(01110011)_{BCD}$

△ Fig.1.16 BCD-codering van een decimaal getal. Bron: W.Van Wichelen

1.3.2 ASCII-code

ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) is een standaard **alfanumerieke** 7-bits-tekencodering om Latijnse letters, cijfers, leestekens en stuurcodes te representeren. Een eerste editie werd gepubliceerd in 1963. De standaard ASCII-tabel bestaat uit de 94 zichtbare tekens (hoofdletters, kleine letters, cijfers, leestekens en enkele symbolen), de spatie, en 33 controle-tekens of stuurcodes[5]. De stuurcodes stellen geen zichtbare tekens voor en stammen nog af uit de tijd dat de uitvoer meestal niet op een beeldscherm werd getoond. Telkens wanneer we bijvoorbeeld een toets indrukken op het toetsenbord van een computer genereren we deze ASCII-code. Omdat de code gebruik maakt van 7 bits zijn er bijgevolg 2^7(=128) mogelijke coderingen. Deze 128 ASCII-codes zijn te zien in de tabel van figuur 1.17. De 94 zichtbare tekens en de spatie, en hun representaties als byte, zijn exact overgenomen in hedendaagse uitbreidingen zoals ANSI, ISO 8859-1 en UTF-8.

Low Order Bits		High Order Bits 0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7
0000	0	NUL	DLE	Space	0	@	P	`	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{
1100	C	FF	FS	,	<	L	\	l	|
1101	D	CR	GS	-	=	M	]	m	}
1110	E	SO	RS	.	>	N	^	n	~
1111	F	SI	US	/	?	O	_	o	DEL

△ Fig.1.17 De ASCII-karakterset.
Bron: http://vidarbhastudents.com

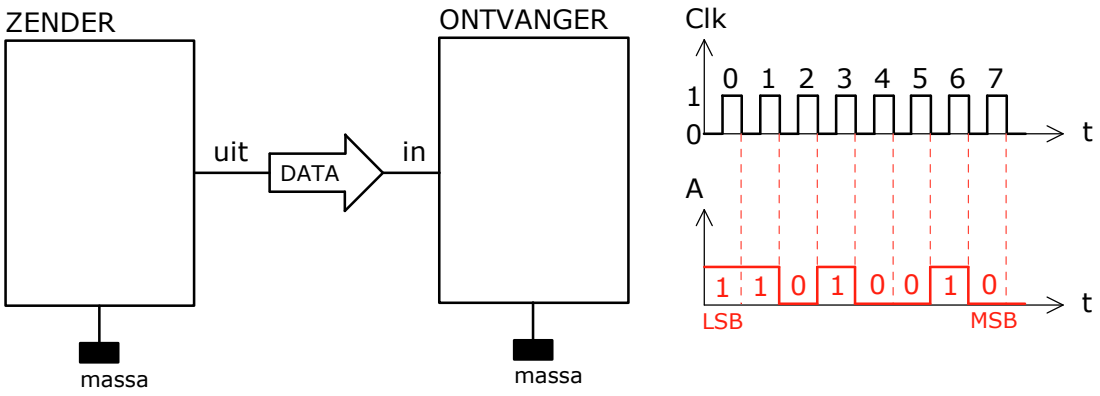
[5] De code 'DEL' bestaat binair uit 7 enen en werd gebruikt om foutieve invoer te herstellen bij ponsbanden. Als men per ongeluk het verkeerde teken had ingeponst, ponste men er eenvoudig een DEL (met overal gaten) overheen, zodat het verkeerde teken werd genegeerd.

1.4 Digitale transmissie

In de digitale wereld worden gegevens (data) verzonden door gebruik te maken van bits. Dit kan gebeuren op twee manieren: *serieel* of *parallel*. Het voornaamste verschil tussen seriële en parallelle transmissie is de manier waarop de gegevens worden verzonden. Bij seriële overbrenging is het *opeenvolgend*, terwijl dit bij parallelle overdracht *gelijktijdig* is.

1.4.1 Seriële transmissie

Bij **seriële transmissie** gebeurt de overdracht van een bit tussen zender en ontvanger op een *sequentiële* manier. Een zogenaamde **klok** legt het tempo van de overdracht op. We wensen bijvoorbeeld een databyte $(01001011)_2$ over te brengen via een serieel transmissiekanaal. Eerst wordt '0' verzonden en dan wordt '1' verzonden, opnieuw '0' enzovoort. In principe is bij deze transmissiemethode slechts één data lijn nodig. Vandaag is deze methode zeer populair omwille van belangrijke voordelen. Er wordt immers geen gebruik gemaakt van *parallelle* bits zodat er geen bijkomende synchronisatie nodig is. We kunnen de kloksnelheid tot een zeer hoog niveau opdrijven waardoor een grote *baudrate*[6] bereikt kan worden. Seriële transmissie is dan ook aangewezen om lange afstanden te overbruggen. Aangezien er geen nabijgelegen parallelle lijnen zijn, wordt het signaal niet beïnvloed door fenomenen zoals *crosstalk*[7] en *interferentie*. Hier heeft parallelle overbrenging wel last van.



△ Fig.1.18 Principe van seriële transmissie
Bron: W.Van Wichelen

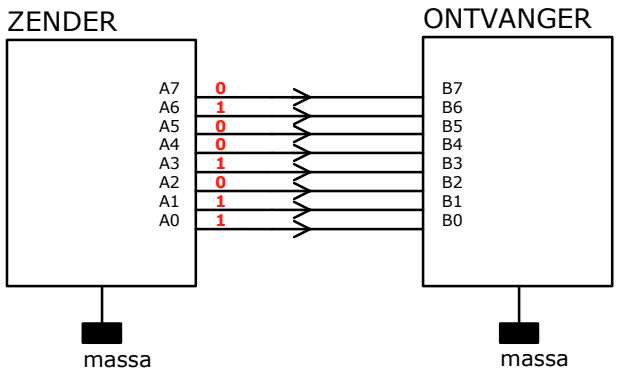
[6] Baudrate geeft het aantal signaalwisselingen per seconde aan op een datatransmissiekanaal. De eenheid 'baud' is afgeleid van Émile Baudot, de uitvinder van de baudotcode voor telegrafie.

[7] Crosstalk of overspraak is elektromagnetische interferentie tussen verschillende signalen. Wanneer bijvoorbeeld telefoonkabels enkele kilometers dicht tegen elkaar liggen zou het kunnen gebeuren dat er geluiden in de telefoon klinken die van anderen afkomstig zijn.

Een in het verleden veel gebruikte seriële transmissiemethode was de **RS-232** standaard en staat ook bekend als de *seriële poort*. De meest gebruikte seriële interface vandaag is **USB** (Universal Serial Bus). **Ethernet** passen we toe voor het aansluiten van netwerken en is ook een voorbeeld van seriële communicatie.

1.4.2 Parallelle transmissie

Bij **parallelle transmissie** gebeurt de overbrenging van parallelle databits **tegelijkertijd**. Stel dat we beschikken over een parallel transmissiesysteem (zie figuur 1.19) dat 8 bits per keer verzendt. Om dit te kunnen verwezenlijken hebben we 8 afzonderlijke lijnen nodig. Stel je voor dat we de databyte $(11010010)_2$ willen doorzenden. De eerste lijn (A0) stuurt '1', de tweede lijn (A1) stuurt '1', de derde lijn (A2) stuur '0', enzovoort. Elke lijn stuurt op hetzelfde tijdstip de overeenkomende bit. Het nadeel is dat we moeten beschikken over *meerdere* draden. Omdat er meer pennen moeten zijn, worden de poorten en sloten groter in vergelijking met seriële transmissie. Theoretisch zou parallelle overbrenging sneller moet zijn, omdat meerdere bits tegelijkertijd worden verzonden. In de praktijk blijkt dit echter niet zo te zijn. De reden hiervoor is dat *alle* parallelle gegevensbits aan de zijde van de ontvanger moeten ontvangen zijn voordat de volgende dataset kan worden verzonden. Door looptijdverschillen kan het gebeuren dat niet alle bits tegelijkertijd bij de ontvanger terecht komen. Er moet dan een *wachttijd* ingelast worden voor de nodige synchronisatie. Hierdoor kan de kloksnelheid niet zo hoog worden als bij seriële overbrenging. Een ander nadeel van parallelle overbrenging is dat de naburige draden problemen kunnen opleveren door interferentie. Om deze redenen wordt parallelle overbrenging gebruikt voor *korte* afstanden.



△ Fig.1.19 Principe van parallele transmissie
Bron: W.Van Wichelen

De meest bekende parallelle overbrenging is de printerpoort. Dit is de poort die ook wel de **parallelle poort** wordt genoemd en hij werd vroeger gebruikt om printers aan te sturen. In het verleden werden harde schijven en CD-ROM lezers aangesloten op de pc via het PATA-protocol (Parallel Advanced Technology Attachment). Momenteel zijn ze vervangen door seriële transmissietechnologie. De snelste bus in de computer, die de CPU met het RAM-geheugen verbindt, blijft wel een parallelle overdracht.

Samenvatting

- Bij seriële overdracht worden de gegevens per bit overgebracht. Bij parallelle overdracht worden meerdere bits tegelijkertijd verzonden.
- Seriële transmissie heeft slechts één draad nodig. Parallelle overbrenging vereist meerdere draden.
- De grootte van de seriële bussen is kleiner dan parallelle bussen, aangezien het aantal pennen minder is.
- Seriële transmissielijnen hebben geen last van interferentie en crosstalk problemen.
  Parallelle overdracht wordt wel geconfronteerd met dergelijke problemen door de nabijgelegen lijnen.
- Seriële overdracht kan sneller worden gemaakt door de kloksnelheid te verhogen. Om de volledige ontvangst van alle bits te synchroniseren moet bij parallelle overdracht de kloksnelheid langzamer worden gekozen.
- Seriële transmissielijnen kunnen gegevens over een lange afstand verzenden terwijl dit niet zo is bij parallelle transmissie.
- Vandaag is de meest gebruikte transmissietechniek de seriële overdracht.