

S  
I  
N  
C  
E  
H  
A  
R  
D  
W  
A  
R  
E  
S

JAN VERHOEVEN







# LOGICA

## 1.0. Inleiding

Onder logica worden in de computerkunde twee dingen verstaan. Enerzijds is er de wiskundige logica (de Boolese algebra) die veel gebruikt wordt voor de softwarekant. Anderzijds is er de hardware-logica : de logische bouwstenen of schakelingen.

## 1.1. Software logica

Iedereen die kan werken met "IF-THEN"-statements, heeft de basisprincipes van logisch denken in huis. Logica valt enigszins te omschrijven als :

Het trekken van één of meerdere conclusies uit een combinatie van basisgegevens.

Een bekend voorbeeld van logisch vergelijken en combineren is het volgende :

a) alle koeien zijn spleethoevige dieren

b) alle spleethoevige dieren zijn zoogdieren

Door vergelijken a) te substitueren in b) wordt verkregen :

c) alle koeien zijn zoogdieren.

## 1.2. Hardware logica.

De hardware logica is in principe identiek aan de software-logica, met dit verschil dat met behulp van een beperkt aantal basisgegevens, door onderlinge combinatie zoveel mogelijk nieuwe gegevens moeten worden gegenereerd. De basisgegevens zijn hier :

- a) de elementen van de "control-bus",
- b) de elementen van de "adress-bus".

In mindere mate wordt ook gebruik gemaakt van de acht elementen van de "databus".

Door combinatie van de verschillende "bus"-elementen worden nieuwe grootheden samengesteld, die nodig zijn om de processor gelijk te laten lopen (synchroniseren) met de randapparatuur (peripherals).

## 2. De bussen .

### 2.0. Wat is een "bus" ?

In de microprocessor-wereld (een microwereld) wordt onder een bus verstaan :

De verzameling van gelijkgezinde signalen die (soms) op en neer pendelen tussen processor en randapparatuur.

De "bus" verzorgt dus de communicatie tussen de processor en de rest van het systeem. Het begrip "bus" is tamelijk abstract. Het is gewoon een benaming zonder verder enige diepere betekenis.

In het algemeen worden drie soorten bussen onderscheiden :

- adress-bus
- data-bus
- control-bus.

Hierover meer in het volgende.

### 2.1. De control-bus.

De control-bus bevat de echte synchronisatiesignalen die  $\mu P$  (micro-processor) en buitenwereld gebruiken om met elkaar samen te werken. De control-bus bevat de volgende elementen :

| naam                         | afkorting     | in/uit |
|------------------------------|---------------|--------|
| memory-request               | <u>MREQ</u>   | uit    |
| input/output request         | <u>IORQ</u>   | uit    |
| bus-request                  | <u>BUSREQ</u> | in     |
| interrupt                    | <u>INT</u>    | in     |
| non maskable interrupt       | <u>NMI</u>    | in     |
| lezen                        | <u>RD</u>     | uit    |
| schrijven                    | <u>WR</u>     | uit    |
| 1 <sup>e</sup> machinecyclus | <u>M1</u>     | uit    |
| refresh                      | <u>RFSH</u>   | uit    |
| halt                         | <u>HALT</u>   | uit    |
| wachten                      | <u>WAIT</u>   | in     |
| reset                        | <u>RESET</u>  | in     |
| bus-acknowledge              | <u>BUSACK</u> | uit    |

Op het toepassen van de verschillende signalen wordt later terug gekomen.

## 2.2. De adresbus.

De adresbus bevat de informatie om aan te geven welke geheugencel of I/O-poort aangestuurd wordt.

Deze bus bevat 16 elementen, gemerkt  $A_0$  tot en met  $A_{15}$ . Wordt de adresbus gecombineerd met een actieve MREQ dan wordt een bepaalde geheugencel aangesproken. Afhankelijk van de rest van de control-bus wordt er dan gelezen, geschreven of ververst.

De adresbus is een éénrichtingsbus. Hij kan uitsluitend zenden (dus van de  $\mu P$  af).

## 2.3. De databus.

Zoals de naam al suggereert, bevat deze bus de zogenaamde datalijnen. Deze verbindingen tussen  $\mu P$  en buitenwereld zorgen voor de feitelijke communicatie tussen randapparatuur en processor.

De Z-80 is een acht-bits  $\mu P$ : hij werkt met verzamelingen van 8 bits tegelijk. Een verzameling van acht bits wordt ook wel byte genoemd. De elementen van de databus zijn genummerd  $D_0$  tot en met  $D_7$ .

De databus zorgt dus voor de gegevensstromen, en is in tegenstelling tot beide andere bussen bidirectioneel. In een volgende bijeenkomst wordt nader ingegaan op het samenspel van bussen en interne organisatie van de Z80.

## 3. Logische bouwstenen.

### 3.0. Poorten of "gates".

In de digitale electronica wordt altijd gesproken over poortschakelingen ofwel "gates".

Onder zo'n poort of "gate" wordt simpelweg een digitaal vergelijkingsnetwerk verstaan. Zo zijn er AND-gates, NAND-gates, OR-gates, NOR-poorten, enz.



### 3.1. Logische niveaus.

In de digitale electronica wordt gewerkt met 2 logische niveaus; de logische 0 en de logische 1. Onder een logische nul verstaat men een spanning in de buurt van de nul volt, terwijl een logische één wordt voorgesteld door een spanning in de buurt van de voedingsspanning. Hierover meer in paragraaf 3.2.

De logische niveau's kunnen ook worden gezien als "waarheden". In de positieve logica heet een logische één "waar" en een logische nul "onwaar" (respectievelijk "true" en "false").

Naast deze positieve logica is er ook nog de negatieve logica. Hier geldt het omgekeerde : logische nul komt overeen met waar.

Bij  $\mu P$ 's wordt hoofdzakelijk gewerkt met negatieve logica. Dit wordt dan aangegeven door een horizontale streep boven een symbool :  $\overline{MREQ}$  (true = actief = log. nul)  
ENABLE (true = actief = log. één).

### 3.2. Logische families.

Heden ten dage zijn er een drietal logische families :

- a) TTL (= Transistor-Transistor Logic)
- b) CMOS (= Complementary Metal Oxide Semiconductor)
- c) HCMOS (= High speed CMOS)

Deze worden in het navolgende beknopt besproken.

#### 3.2.1. TTL-serie.

TTL is een al wat oudere serie logische schakelingen bestaande uit enkele honderden verschillende IC's.

TTL heeft een voedingsspanning nodig van  $5\text{ V} \pm 5\%$ .

Ingangsspanningen tussen 0 en 0,8 Volt worden beschouwd als logische nul. Ingangsspanningen tussen 2,0 en 5,0 Volt wordt beschouwd als logische één.

De TTL-serie bestaat weer uit een 5-tal subserie's waarvan de LS-TTL en S-TTL de belangrijkste vormen.

Gewone TTL (gekenmerkt als 74XX serie) eist tamelijk veel stroom (ca 2 mA per ingang) en poortvertragingstijden in de orde van 20 ns (nano seconde =  $10^{-9}$  s).

De Schottky-TTL (=74Sxx) is ca 4x zo snel, doch neemt

ook 2 maal zoveel stroom op.

De Low-power-Schottky-TTL (= 74 LS XX) combineert de dubbele snelheid van TTL met sterk verminderde stroomopname (ca 0,4 mA/gate).

Voordelen van TTL zijn :

- hoge schakelsnelheid (tot 50 MHz)
- veel IC's beschikbaar.

Nadeel van TTL is de relatief grote stroomopname,

### 3.2.2. CMOS.

De CMOS serie ( typenummers 40XX en 45XX) bevat voor een deel dezelfde IC's als de TTL-reeks, en voor een deel bouwstenen die niet in TTL bekend zijn. Zie hiervoor de betreffende databoeken.

Karakteristiek voor de CMOS reeks is wel de zeer hoge ingangsimpedantie. Deze poorten bevatten speciale (complementaire ) ingangscircuits opgebouwd met FET's (Field Effect Transistors), wat resulteert in ingangsweerstanden in de buurt van enkele giga-ohms ( $10^9$  Ohm). De ingangsstroom bedraagt dan ook maar enkele nA.

CMOS uitgangen kunnen enkele mA leveren, dus het is wel duidelijk dat één uitgang hier vrijwel oneindig veel ingangen kan aansturen.

CMOS schakelingen hebben een variabele voedingsspanning ; alles tussen ca 3 en 18 Volt voldoet. Resultaat hiervan is dat de logische niveau's niet meer vast liggen. Deze verschuiven met de voedingsspanning volgens :

- logische nul = 0-30 % van de voedingsspanning
- logische één = 70-100 % van voedingsspanning.

Grote nadeel van CMOS is wel de lage schakelsnelheid. Poortvertragingstijden in de orde van grootte van 200 ns zijn heel normaal; interfacen met een 2 MHz systeem wordt hierdoor al problematisch. De Sinclair machines zitten allen boven de 3 MHz dus...

### 3.2.3. HCMOS (High speed TTL-compatible CMOS)

Binnen enkele jaren (eind '84, begin '85) komt er een nieuwe serie logische bouwstenen op de markt, die de voordelen van CMOS en TTL combineert. Deze HCMOS reeks gaat een groot deel van de TTL-serie bevatten alsmede een aantal IC's uit de CMOS-reeks die niet in TTL voorkomen. Kenmerken van HCMOS zullen zijn :

- hoge schakelsnelheid ( minimaal 30 MHz)
- TTL logische niveau's
- grotere tolerantie voedingsspanning (  $\pm 10\%$  )
- CMOS ingangsweerstanden, gekoppeld aan hoge uitgangsströmen
- CMOS stroomverbruik.

Op het moment is er al wel een HCMOS (High speed CMOS) serie in beperkte mate op de markt, doch deze werkt met de CMOS logische niveau's. Hierdoor is deze serie wel snel genoeg om een  $\mu P$  bij te houden, doch door de afwijkende logische niveau's niet interessant, omdat zowel  $\mu P$  als randapparatuur TTL-niveau's verlangen. Het verschil tussen HCMOS en HCMOS is tevens dat HCMOS genoeg neemt met een voedingsspanning van 2 tot 6 Volt.

Dit is misschien toch nog wel interessant om te gebruiken in combinatie met LP SRAM's die zijn voorzien van een "battery-back-up"....

## 4. Soorten poortschakelingen.

### 4.0. Nomenclatuur.

Bij de beschrijving van poorten worden een aantal ingangen (gemarkt door de letters A, B, C, enz) door middel van een blok logica omgezet in een uitgang , welke gemerkt wordt door de letters vanaf P richting Z.

In zogenaamde "truth-tables" of waarheidstabellen worden de verschillende combinaties van ingangssignalen met het corresponderende uitgangssignaal overzichtelijk weergegeven. Hierbij wordt soms gebruik gemaakt van wat Boolese algebra volgens :

a)  $0 + 0 = 0$

b)  $1 + 0 = 1$

c)  $1 + 1 = 1$

d)  $0 \times 0 = 0$

e)  $1 \times 0 = 0$

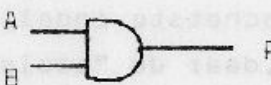
f)  $1 \times 1 = 1$

g)  $I = 0$



#### 4.1. Basispoorten.

##### 4.1.1. AND - gate.


symbool  P Boole :  $P = A \cdot B$

waarheidstabel:

Deze poort levert  
alleen dan een één op  
de uitgang, wanneer beide  
ingangen eveneens 1 zijn.

| A | B | P |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

##### 4.1.2. NOT - gate.

symbool  Q Boole :  $Q = \bar{A}$

tabel:

| A | Q |
|---|---|
| 1 | 0 |
| 0 | 1 |

De "NOT"-poort heet ook wel  
inverter.

##### 4.1.3. OR - gate.

 Z

Boole :  $Z = C + D$

tabel :

| C | D | Z |
|---|---|---|
| 1 | 1 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 0 | 0 |

De uitgang is uitsluitend  
nul wanneer beide  
ingangen ook nul zijn.

##### 4.1.4. EXOR - gate.

 Q

Boole :  $Q = A \cdot \bar{B} + \bar{A} \cdot B$

tabel :

| A | B | Q |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

De uitgang is alleen dan "hoog"  
wanneer beide ingangen ver-  
schillend niveau hebben.

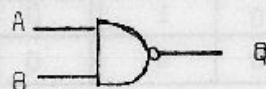
Deze poort wordt ook wel een  
"optel-poort" ge-noemd.

## 4.2. Afgeleide poorten.

### 4.2.0. Afgeleid ?

De in dit hoofdstuk geschetste indeling moet niet te absoluut gezien worden, daar de "basis"-poorten ook opgebouwd kunnen worden uit combinaties van "afgeleide poorten". Het is allemaal nogal relatief.

#### 4.2.1. NAND - gate.

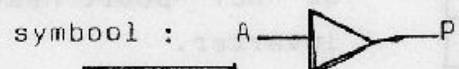


tabel :

| A | B | Q |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

$$\text{Boole : } Q = \overline{A \cdot B}$$

#### 4.2.2. Buffer -gate.

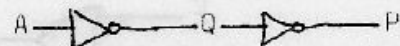


$$\text{Boole } P = A$$

tabel :

| A | P |
|---|---|
| 1 | 1 |
| 0 | 0 |

alternatieve opbouw :



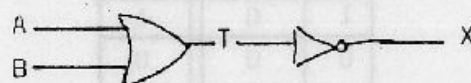
#### 4.2.3. NOR -gate.

symbol :



$$\text{Boole : } X = \overline{A + B}$$

alternatief :



#### 4.2.4. EXNOR - gate.

symbol :



$$\text{Boole : } Q = A \cdot \overline{B} + B \cdot \overline{A}$$

### 4.3. Enkele handigheidjes.

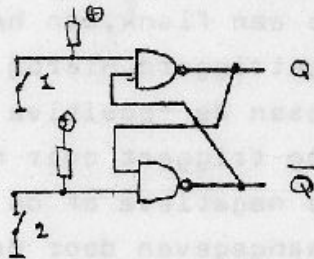




## 5. Complexe eenheden.

### 5.1. Flip-flop's

Met behulp van de genoemde "basis"-gates kunnen complexe netwerken worden gebouwd. Een zo'n schakeling is de flip-flop (FF). Dit is een soort bistabiel relais :



In drukken van drukknop 1 levert de "flip" op, terwijl de andere knop voor de "flop" zorgt. Er bestaat een 3-tal soorten FF's. Deze zijn :

- Data FF
- RS FF
- JK flipflop.

fig. 5.1.1.

Bovenstaande fig. 5.1.1. is een voorbeeld van een RS (Reset, Set) flipflop.

De D-flipflop is de bij uP's meest toegepaste FF. In bijlage I is een datasheet opgenomen. De Reset ingang van een FF wordt ook wel eens de "clear" ingang genoemd.

De data-FF zorgt ervoor dat bij iedere clock-puls het signaal wat op dat moment op de D-ingang staat, wordt doorgestuurd naar de Q-uitgang.

### 5.2. Triggering.

De vraag die bij de D-flipflop opdoemt is nu : "wanneer wordt de data exàct doorgegeven?". Ofwel : hoe schakelt een FF, op de logische niveau's of ergens anders op ?

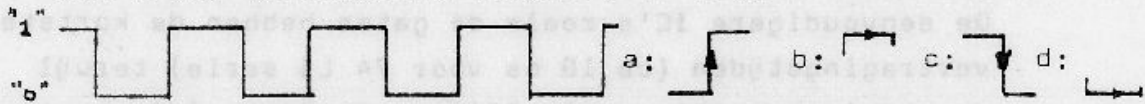


fig. 5.2.1.

fig. 5.2.2.

Een kloksignaal (fig. 5.2.1.) kan ontleed' worden in 4 delen (fig. 5.2.2.) :

- a) de opgaande flank (leading of rising edge) ofwel de 0-1 overgang.
- b) het logische "1" niveau
- c) de neergaande flank (falling edge, negative of trailing edge) ofwel de 1-0 overgang.
- d) het logische "0"-niveau

Hieruit blijkt dus dat de FF kan aanspreken (= triggeren) op twee verschillende wijzen :

- a) de flank van het signaal (fig. a en c )
- b) het logische niveau ( fig. b en c ).

Wanneer een schakeling triggert op een flank, dan heet deze "edge-triggered" ofwel flankgetriggerd. Hierbij wordt dan nog verschil gemaakt tussen de "positive edge" en de "negative edge". De schakeling triggert over het algemeen slechts op één flank ; de negatieve of de positieve. In de databoeken wordt dit aangegeven door de betreffende ingangen te merken met een van de pijltjes zoals weergegeven in figuren 5.2.2. a) & c). Sommige schakelingen triggeren op logische niveau's. Een voorbeeld hiervan vormen de RS-FF's.

### 5.3. Poortvertragingen.

Bekend is dus nu wannéér een IC schakelt, echter hoe lang duurt het dan nog tot de betreffende uitgang klaar is met zijn logische "aktie"?

Het is onlogisch om te veronderstellen dat deze tijd nul seconden duurt. Dit zou inhouden dat de elektronen in de halfgeleiders met een oneindige snelheid zouden bewegen.

In de praktijk blijkt dat deze vertragingstijd (ofwel in het engels "propagation-delay") eindig is en afhankelijk van zowel het soort IC als van de logische familie.

De eenvoudigere IC's zoals de gates hebben de kortste vertragingstijden (ca 10 ns voor 74 LS serie) terwijl de complexere IC's zoals FF's en decoders (die bestaan uit combinaties van "simpele" poorten) meestal vertragingstijden hebben die een veelvoud hiervan zijn.

### 5.4. Deler IC's.

Eén van de toepassingen van flipflops vormen de deler-IC's. Deze dienen om de frequentie van kloksignalen te



reduceren. Dus om van een 14 MHz klok een 7 MHz klok te maken.

Stel een kloksignaal zoals afgebeeld in fig. 5.4.1.a) is op de CP (ClockPulse) ingang van een D-FF aangesloten, zoals in fig. 5.4.2. weergegeven.

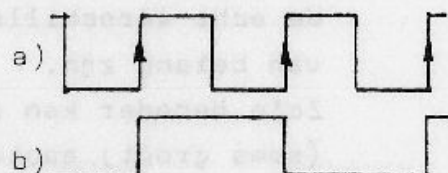


fig. 5.4.1.

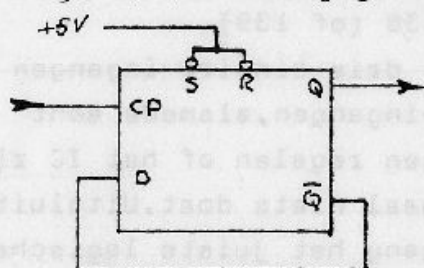


fig. 5.4.2.

De Q invers uitgang wordt met de data-ingang doorverbonden. Resultaat is dat iedere stijgende flank het logische niveau wisselt. De dalende flank wordt genegeerd.

En omdat het kloksignaal slechts één stijgende flank per periode

bevat, wisselt de Q uitgang dus 1 maal per periode van niveau; het kloksignaal wisselt echter tweemaal per periode van niveau. Het resultaat hiervan is dus dat één periode van de Q uitgang evenlang duurt als twee periodes van het ingangskloksignaal. De D-FF wordt hier dus gebruikt als (twee) deler. Door meerdere FF's achter en naast elkaar te schakelen wordt het mogelijk om vrijwel iedereillekeurige deelfactor in te stellen.

## 5.5. Decoders.

Wie enigszins bekend is met binaire wiskunde weet dat met drie "bits" acht verschillende toestanden kunnen worden gegenereerd. Dit kan met behulp van drie dubbeltjes gecontroleerd worden. Hierbij wordt de "kop" kant het logische nulniveau genoemd en de andere kant de logische één.

Stel nu er staan drie logische lijnen ter beschikking die in binair aangeven welke uit een serie van acht lampen (bij voorbeeld LED's) aan moet zijn.

Het probleem wat hier gedefinieerd is heet "decoderen". De drie gecodeerde logische lijnen worden ontrafeld tot de acht verschillende toestanden die voor de gebruiker van belang zijn.

Zo'n decoder kan opgebouwd worden met behulp van een (soms groot) aantal logische poorten.

Het is echter ook mogelijk om een kant en klaar decoder IC te kopen, zoals de 74 LS 138 (of 139).

Zo'n LS 138 (bijlage 2) heeft drie binaire ingangen gekoppeld aan drie "enable"-ingangen, alsmede acht uitgangen. De "enable"-ingangen regelen of het IC zijn werk verricht, of dat het geheel niets doet. Uitsluitend wanneer op iedere enable ingang het juiste logische niveau aanwezig is, worden de ingangen gedecodeerd. Wanneer niet ge-enabled (dus gedisabled) zetten de 138 en 139 logische enen op alle uitgangen. Wanneer ze wel ge-enabled zijn, staan op alle niet werkzame uitgangen logische enen, terwijl op de ene, exclusieve, uitgang die door de stand van de drie bits weergegeven wordt, een logische nul staat; negatieve logica.

#### 5.6. Tri-state uitgangen.

Wanneer meerdere uitgangen van IC's op een ingang samen komen is het wenselijk dat deze uitgangen elkaar niet kunnen beïnvloeden. Dit is mogelijk gemaakt door naast de beide logische niveau's nog een derde toestand te creëren. Deze toestand noemt men de derde toestand (of in het engels three state of 3S). Hierbij wordt de uit-

gang van het IC gewijzigd.

De normale TTL uitgang (fig. 5.6.1.) bevat één transistor en evenveel weerstanden. Bij een logische één op de uitgang wordt de transistor niet opengestuurd. Bij een logische nul wel.

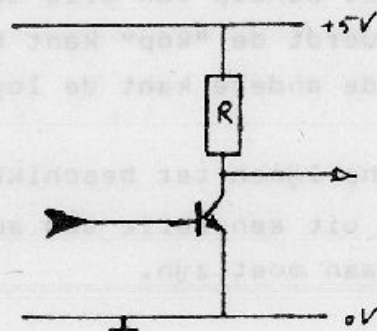
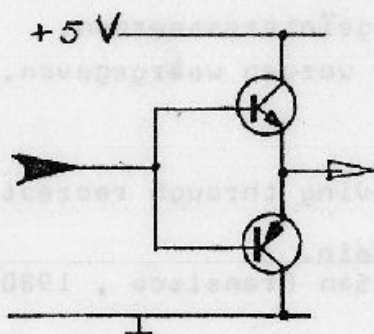


fig 5.6.1.



Door nu de uitgang te voorzien van twee complementaire transistoren (fig. 5.6.2.) kan ook nog een derde toestand ingebouwd worden.



Bij een logische één wordt de bovenste transistor opengestuurd. Bij een logische nul wordt de onderste opengezet.

Voor het creëren van de derde toestand worden beide transistors "dichtgeknepen". De uitgang "drijft" dan met de rest van de signalen mee, zonder er zelf invloed op uit

fig. 5.6.2.

te oefenen. Deze derde toestand wordt daarom ook wel eens de "float" genoemd.

De derde toestand wordt meestal gekozen wanneer niet alle enable ingangen geselecteerd zijn.

### 5.7. Voeding.

De voedingsspanning voor LS TTL bedraagt zoals reeds vermeld 5 Volt  $\pm$  5 %. Deze 5 V kan met een goede spanningsregelaar (zie ook Hardware deel 1) op vrij eenvoudige wijze aangemaakt worden.

Het aansluiten van de IC's zelf lijkt ook erg eenvoudig. In feite is het dit ook, doch men moet wel weten waar op te letten.

In de eerste plaats dienen de voedingslijnen zo dik mogelijk te zijn; tevens moeten de voedingslijnen van alle aanwezige IC's zoveel mogelijk onderling met elkaar gekoppeld worden (een soort matrix-netwerk).

Op de tweede plaats dient men per 2 à 4 TTL IC's een ont-koppel condensator aan te brengen. Deze moet dan een waarde hebben van ergens tussen de 22 en 100 nF (nano-farad).

Het is aan te bevelen om zoveel mogelijk verschillende waarden toe te passen; dus niet allemaal van één waarde.

### Literatuur-overzicht.

In het onderstaande enkele boekwerken die interessant kunne (cq zullen) zijn voor de geïnteresseerden.  
De boeken zullen per onderwerp worden weergegeven.

#### - Logica (software) :

= Mathematics : Problem solving through recreational mathematics.

Bonnie Averbach & Orin Chein.

W.H. Freeman and company, San Fransisco , 1980

#### - Logica (hardware) :

+ COS/MOS Integrated Circuits

RCA solid state devices , 1980

Inelco BV

Turfstekerstraat 63

1431 GD Aalsmeer.

+ TTL data book.

Fairchild Semiconductor , 1978

+ High-Speed CMOS Data Book

Fairchild Semiconductor , 1983

Fairchild Semiconductor

Ruysdaalbaan 35

5613 DX Eindhoven

+ CMOS data book.

Fairchild Semiconductor , 1977

+ Microprocessors and microcomputers.

R.J. Tucci and L.P. Laskowsky.

Prentice-Hall Inc. , Englewood cliffs , New Jersey  
1982

+ Datasheet boek

Uitgeversmaatschappij Electuur bv

Beek , 1983

+ Digitale electronica voor beginners.

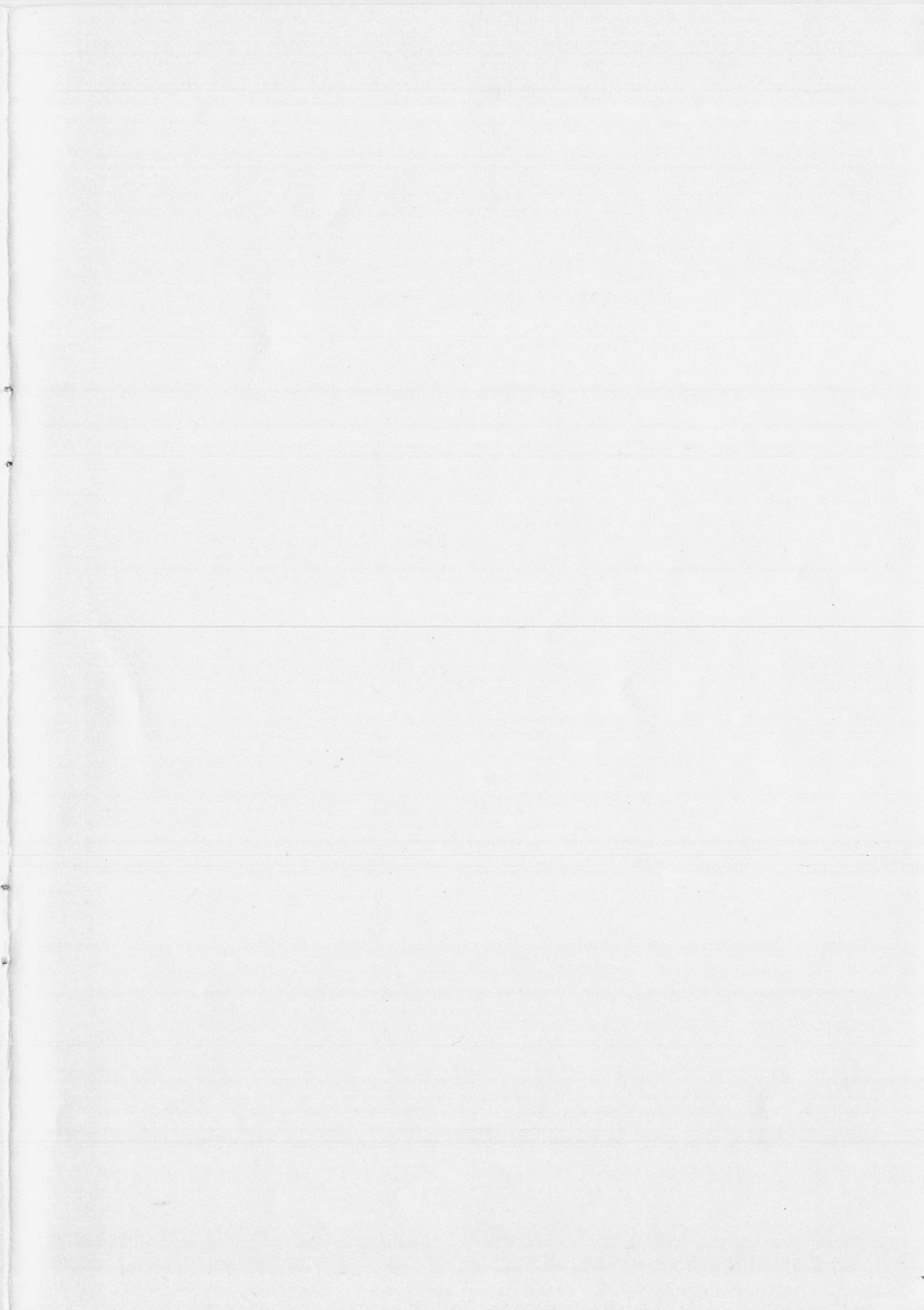
J. Kleemann.

Kluwer's electronica bibliotheek nr 3.

Deventer , 1975

+ Diverse tijdschriften waarvan Elektuur de belangrijkste is , was , en waarschijnlijk ook wel zal blijven.













hrc

GEBRUIKERS  
GROEP  
NEDERLAND