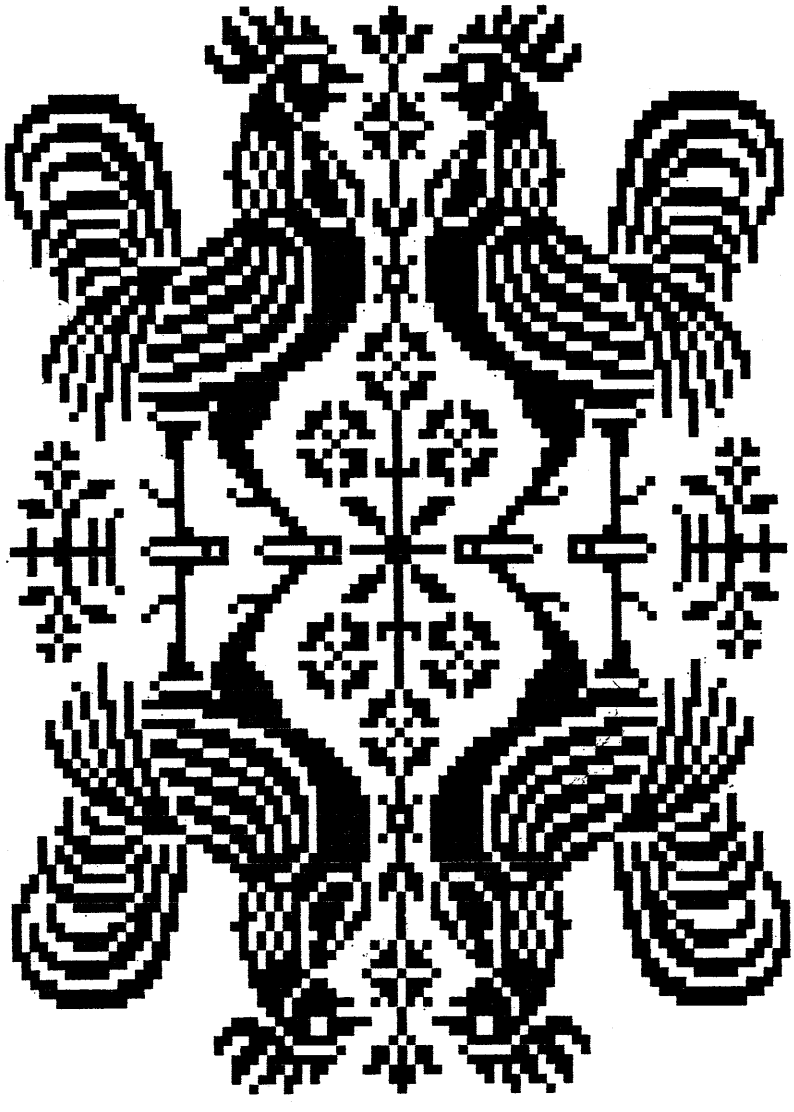
DAI NAMIC

Nr5 MEI-JUNI 1981



GEDRUKTE PERIODIEK verschijnt tweemaandelijks

Verantw. Uitgever : W.HERMANS HEIDE 98 3171 WESTMEERBEEK

COLOFON .

DAInamic verschijnt tweemaandelijks.
abonnementsprijs is inbegrepen in de jaarlijkse contributie:
750 Bfr 50 Gld 50 Dm
Bij toetreding worden de verschenen nummers van de jaargang toegezonden.
DAInamic redactie:
Dirk Bonné
Freddy De Raedt
Wilfried Hermans
Jules Meulenbergs
Jos Schepens
Roger Theeuws
Bruno Van Rompaey
Jef Verwimp
vormgeving :Ludo van Mechelen

U wordt lid door storting van de contributie op nr406-3016141-33 van KREDIETBANK WESTMEERBEEK,via bank-instelling of POSTGIRO.
Abonnement loopt van januari tot december.

U kan telefonisch contact nemen op nr 016/698623.

correspondentieadres:
DAInamic
Heide 98
3171 WESTMEERBEEK BELGIE

DAInamic verschijnt de eerste week van de pare maanden.
Bijdragen zijn steeds welkom.

4		3		2		1	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
A	40960	A	2560	A	160	A	10
B	45056	B	2816	B	176	B	11
C	49152	C	3072	C	192	C	12
D	53248	D	3328	D	208	D	13
E	54344	E	3584	E	224	E	14
F	61440	F	3840	F	240	F	15

belangrijke ASCII-waarden in DAIpc

functie/symbool	HEX	DEC
back-space	8	8
TAB	9	9
linefeed	A	10
clear screen	C	12
CURSOR UP	10	16
CURSOR DOWN	11	17
CURSOR LEFT	12	18
CURSOR RIGHT	13	19
space-bar	20	32
Ø	30	48
A	41	65
a	61	97
pijltje rechts	89	137
pijltje links	88	136
pijltje boven	5E	94
pijltje onder	8C	140
volle blok	FF	255
verticale lijn	A	10
horizontale lijn	B	11
6 hor.lijnen	1D	29

ASCII - HEX - ASCII CONVERSION TABLE

LSD	MSD	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	0000	NUL	DLE	SP	0	@	P	`	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENG	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	-	=	M	]	m	}
E	1110	SO	RS	.	>	N	↑	n	~
F	1111	SI	VS	/	?	O	←	o	DEL

DAInamic
DAI USERS CLUB

Westmeerbeek juni 81

Beste DAInamic-leden,

11 april : onze eerste bijeenkomst was een succes.
Meer dan 250 bezoekers hadden TONGELSBOS weten te vinden.
Daar hadden we niet op gerekend en 12 machines was dan ook nog te weinig. Op de vergadering hebben we de vertegenwoordiging van DAI onze suggesties kunnen overbrengen; hopelijk is dit terecht gekomen bij de beleidsmensen. Een technische documentatie met volledige schema's en een volledigen, uitgebreiden handboek is beslist noodzakelijk. Wat was er verder zoal te zien en te horen?
De speech synthesiser vertelde zijn verhaaltje, interface met IBM schrijfmachine, DAI aan gemodifieerde kleurenTV(peritel FR), onderwijsprogramma's, screen copies op TX-80 en diverse programma's.
In dit nummer vindt U een paar bijdragen die meedoen aan de wedstrijd DAI-application. Met 50 bladzijden hadden we nog ruimte te kort, zodat een aantal bijdragen verschoven zijn naar volgende nummers.
De beschrijving van MX-80 stellen we ook uit:mogelijk is een uitgebreider marktoverzicht wel op zijn plaats. Heeft U een andere printer dan MX-80 of TX-80 bezorg ons dan documentatie en een voorbeeld van de printkwaliteit. We verzamelen dit materiaal voor ons septembernummer. Het juli-augustus nummer krijgt hoofdzakelijk volgende inhoud: ledenbestand + uitvoerige bespreking van de softwarebibliotheek.

De redactie neemt vakantie van 15 augustus tot 15 september, zodat we rustig het nieuwe schooljaar kunnen voorbereiden.
Indien iemand (eindelijk) in het bezit komt van de floppy-drives willen we graag de bevindingen vernemen; wij zullen voorlopig de budgetaire keuze maken voor de digitale cassetterecorder.

Ondertussen vierden wij bescheiden het eenjarig bestaan van DAInamic...

Wij wensen U veel leesgenot en een zonnige vakantie

de redactie

BLADWIJZER

81	REMARK	Redaktiepraatje
82	BLADWIJZER	
83	APPLICATION	DAI aan de monitor H.Bakker
84	APPLICATION	"
85	APPLICATION	"
86	APPLICATION	"
87	APPLICATION	"
88	CATALOG	Programmabibliotheek nieuwe formule
89	LOOK	TEST OF MATHCHIP 9511 results
90	LOOK	"
91	LOOK	"
92	LOOK	"
93	LOOK	" +++ diagram
94	LOOK	diagram +++ tabel
95	LOOK	tabel
96	LIST	Het testprogramma +++ functie-plot
97	LIST	Het testprogramma +++ start/stop met interrupt 7
98	LOOK	Commentaar bij de test
99	LOOK	Commentaar bij de test
100	LIST	Ass listing van CLOCK for MATH TEST
101	LIST	CLOCK +++ SCREENCOPY on MX-80
102	LOOK	DATA SHEETS van 9511
103	LOOK	"
104	LOOK	"
105	LOOK	"
106	LOOK	"
107	LOOK	"
108	LIST	Paddelen met FGT : SCREENCOPY on MX (MODE 4)
109	LIST	Paddelen met FGT : programma
110	LOOK	EXACTE TIMER +++ TIME OUT +++ ANTWOORDTIJD
111	LIST	Programmavoorbeelden
112	READ	DATA SHEET N2235
113	TALK	Cassette recorder
114	TALK+CATALOG	PHILIPS MINI-DIGITAL-RECORDER
115	PEEK&POKE	PIN-OUT of 50-pens connector inside DAIpc (X-bus)
116	LOOK	Das gesetz der Grossen zahl A.Meystre
117	LIST	statistiek +++ random distributie +++ title
118	LIST	Ass listing van FASING KEYBOARD MUSIC
119	LIST	FASING KEYBOARD MUSIC +++ RANDOM ROUTINE
120	PEEK & POKE	Entrypoints:DOT,DRAW,FILL,SCRN
121	TEACH	Data Statements Generator (DIDACOM)
122	TEACH	Project:MOTORIEKE TRAINING(DIDACOM)
123	PEEK & POKE	POWER-ON INITIALISATIE (B.J. BOERRIGTER)
124	PEEK & POKE	"
125	PEEK & POKE	"
126	PEEK & POKE	"
127	PEEK & POKE	SERIAL RS 232 INTERFACE ON DAIpc
128	APPLICATION	CASSETTE CLIPPERS (W.de Leeuw v. Weenen,C.De Bont)
129	APPLICATION	CASSETTE CLIPPERS (KOP&DAIclub BEEK)
130	IN OTHER WORDS	

****

D E D A I A A N D E M O N I T O R

Om te kunnen zien wat zich allemaal afspeelt binnen uw DAI computer, dient deze aangesloten te worden op een televisie toestel. Hiertoe bevindt zich aan de achterzijde van uw DAI een uitgang, welke verbonden moet worden met de antenne ingang van een willekeurige TV.
Het beeld wat dan verschijnt op het scherm is prima, maar kan in veel gevallen met een vrij kleine ingreep verbeterd worden.
Het bij de DAI (en bij veel personal computers) toegepaste principe van signaaloverdracht naar een beeldscherm heeft voor- en nadelen. Als voordeel geldt de eenvoudige aansluiting op vrijwel ieder TV toestel.
Het principe gaat als volgt: De informatie bestemd voor het beeldscherm gaat niet rechtstreeks naar dat beeldscherm maar wordt in de DAI eerst gemengd met een hoogfrequente wisselspanning (= 'moduleren'). Hierdoor wordt het signaal geschikt voor de antenne ingang van een TV.
In de TV gebeurt dan eerst het omgekeerde: het hoogfrequente signaal wordt weggefilterd en de beeldscherminformatie blijft over ('de-moduleren'- zie fig. 1).

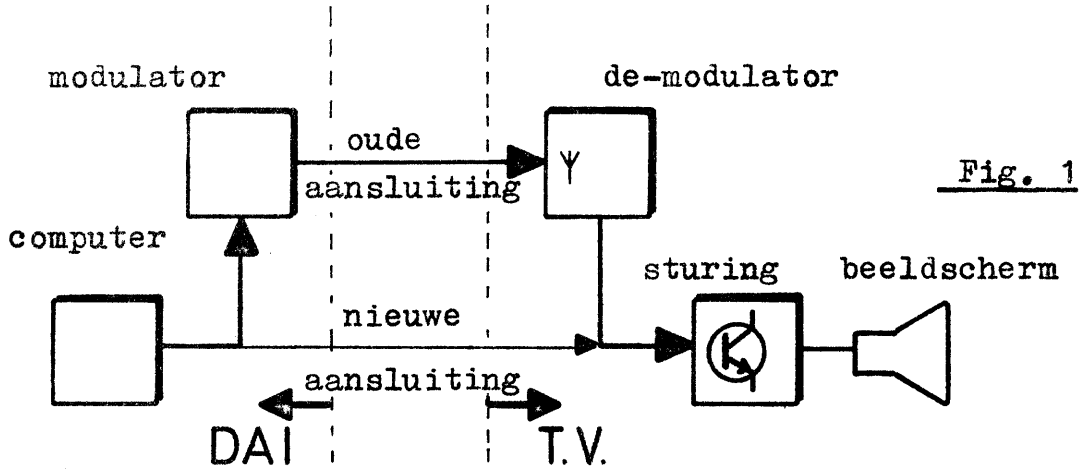


Fig. 1

Als gevolg van deze beide signaalconversies gaat de kwaliteit van het beeldsignaal achteruit, wat uiteindelijk merkbaar is in een iets minder scherp beeld.
Als we nu erin zouden slagen het beeldsignaal rechtstreeks van de DAI naar de beeldbuissturing van een TV te voeren, omzeilen we de beide signaalconversies, en verbetert het

beeld aanzienlijk .
Het probleem nu is: hoe doen we dat. Dit is inderdaad een probleem, want er zijn veel verschillende soorten TV toestellen. Bij b.v. kleuren- en oudere TV's met radiobuizen is het beter om NIET aan ombouw te beginnen.
Het eenvoudigste gaat het ombouwen bij moderne, draagbare zwart/wit toestellen. Belangrijk is het dat de TV een ingebouwde laagspanningsvoeding heeft, welke de 220 Volt voedingsspanning omzet in een lagere spanning en tegelijkertijd de 220 Volt galvanisch scheidt van de rest van de TV.
Verder is het noodzakelijk dat u beschikt over een schema van uw TV ('hoe zit wat waar aangesloten'),dat u dit schema kunt 'lezen' en dat u enige ervaring heeft met elektronica.

Kijk dan eerst op dit schema of de 220 Volt inderdaad omlaag getransformeerd wordt en dat de inwendige voedings(laag)-spanning galvanisch gescheiden is van de netspanning.
Is dit het geval, zoek dan de transistor op, die zorgt voor de intensiteit sturing van het beeldscherm. Meestal is deze als volgt aangesloten (zie pijl in Fig. 2):

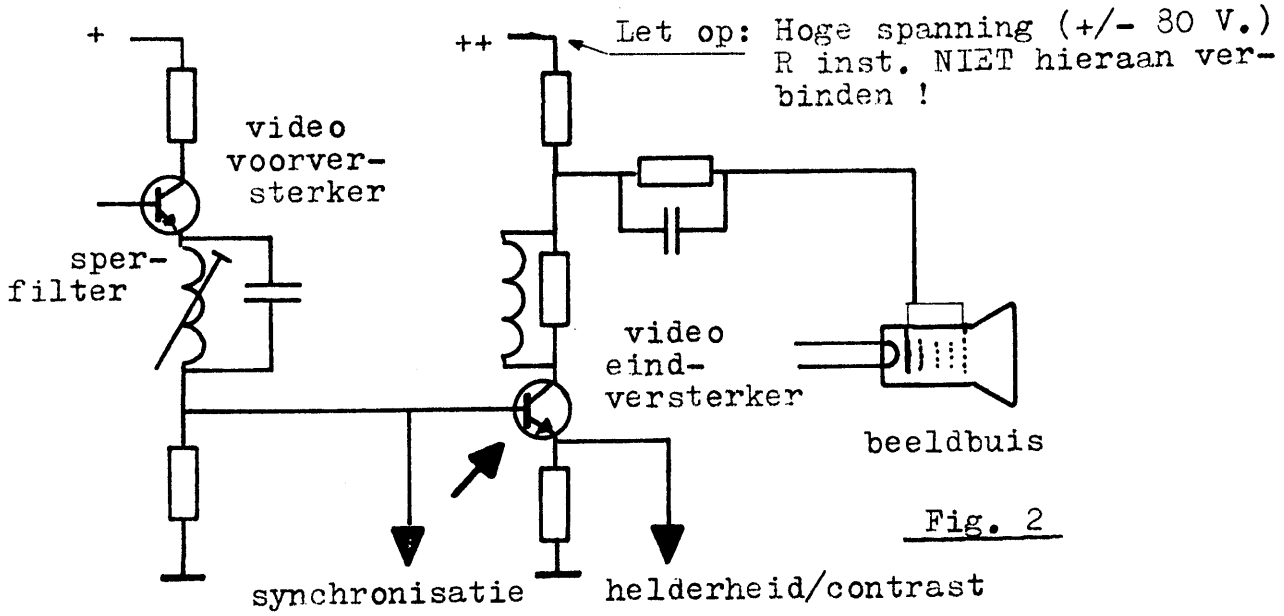
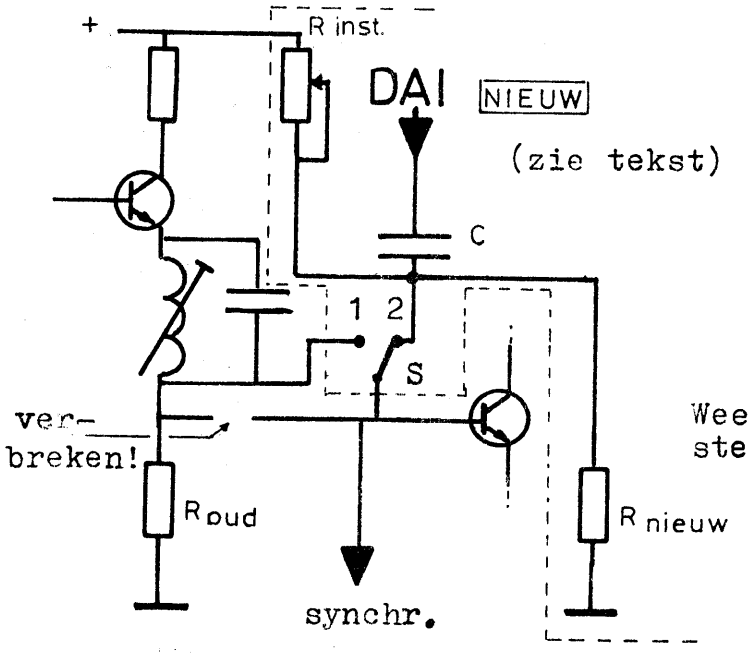


Fig. 2

Heeft u deze gevonden, dan is het karwei al voor een groot deel klaar. Nu volgt het ombouwen van de DAI en de TV.

Eerst de TV. Wijzig de TV schakeling zoals geschetst in fig. 3.



Schakelaar S:

Stand 1: Normaal TV
,, 2: Monitor

Fig. 3

Weerstand Rinst. eerst instellen op maximale waarde!

Monteer aan de achterzijde van de TV een schakelaar met 1 wiselkontakt. Verbreek de verbinding tussen de basis van de eindtransistor en het sperfilter zoals geschetst in Fig. 3 en verbind de basis met het moederkontakt van de schakelaar. Gebruik zo kort mogelijke verbindingsdraden Kies voor de nieuwe weerstand R nieuw een waarde ongeveer even groot als die van R oud. R inst moet ongeveer 4 x de waarde van R nieuw hebben. Draai de instelbare weerstand R inst na montage op zijn maximale waarde.
Kies voor de waarde van C voorlopig 2 microFarad (GEEN Elektrolytisch type).

Verbind kontakten 1 en 2 volgens Fig. 3.
De weerstand R inst moet ervoor zorgen dat de spanning op op kontakt 2 ongeveer hetzelfde is als op kontakt 1. Deze weerstand kan dus het beste afgeregeld worden bij ingeschakelde TV, maar zelfs voor ervaren hobby-isten is dit gevaarlijk gezien de aanwezigheid van hoge spanningen. Bij voorkeur niet doen dus. Probeer er achter te komen via uw schema wat de spanning ongeveer moet zijn op de basis van de video eindtransistor.
Is de spanning bekend, stel dan R inst zo in, dat de deling van de voedingsspanning door R inst en R nieuw de gewenste basisspanning oplevert.
Beter is echter: monteer R inst zo, dat deze met een schroevendraaier of iets dergelijks bereikbaar blijft bij een gesloten TV-kast.
De condensator wordt met een stukje coax kabel verbonden met een extra antenne ingangsbus. Vergeet niet de afscherming van dit stukje coax kabel te verbinden met de voedingsnul. Het beste kunt u de monitor ingang naast de bestaande antenne ingang monteren en de afscherming van de coax kabel verbinden met de afscherming van de oude en nieuwe antenne ingang.

Sluit de TV kast en schakel het apparaat in, nadat u de regelaar voor helderheid op uw toestel zo laag (donker) mogelijk heeft gedraaid. Schakel schakelaar S op 'MONITOR' en draai langzaam de helderheidsregelaar hoger (lichter) totdat de regelaar ongeveer halverwege staat.
Als u R inst al ingesteld had, dan moet nu ongeveer de normale helderheid op uw scherm zichtbaar zijn. Is dit niet het geval dan moet R inst bij te donker beeld iets kleiner gemaakt worden, bij te licht beeld iets groter.
Staat R inst nog steeds maximaal, draai dan R inst LANGZAAM in waarde terug tot de juiste helderheid is bereikt.
Blijft het beeld donker nadat R inst ongeveer tot 1/4 van zijn maximale waarde is teruggedraaid, vervang R inst dan door een andere die 1/3 is van de oude R inst waarde en monteer in serie hiermee een weerstand van 1/10 * R nieuw.
Begin dan weer met R inst maximaal.
Schakel nu vervolgens S over op 'NORMAAL TV' en daarbij mag de intensiteit niet erg veel veranderen.

Vervolgens de DAI. Verwijder de bovenste helft van kast door eerst de vier zwarte dopjes aan de zijkant te verwijderen.
LET OP: Als u dit doet binnen zes maanden na aankoop, vervalt de garantie op uw computer !
Links achter boven het moederbord bevindt zich de TV modulator op een klein printje.
Links achter op dit printje bevindt zich de eigenlijke modulator, die eruit ziet als een blikken doosje van 4x2,5 cm.
In de linkerzijde van dit doosje verdwijnen twee draadjes.
De voorste is de voeding, en de achterste (het dichsts bij de antenne uitgang) vervoert de beeldinformatie.
Dit draadje nu moet verbonden worden met de monitoringang die we net gemaakt hebben aan de TV (Fig. 4).

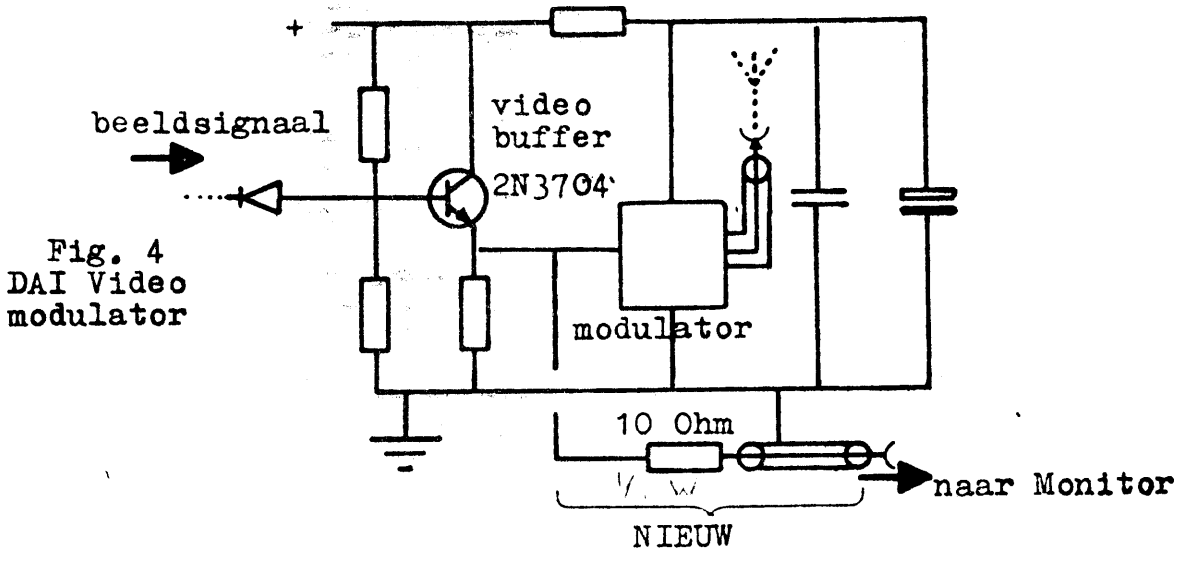


Fig. 4
DAI Video
modulator

Het beste kunnen we een stukje coax kabel solderen aan dit draadje via een weerstand van 10 Ohm en de afscherming van deze kabel verbinden met het blikken huis van de modulator. Voer deze coax kabel via de achterzijde of onderzijde naar buiten en knip de kabel op de gewenste lengte af.
Aan deze zijde monteren we dan een contrastekker (zoals die van de modulator) zodat de meegeleverde verbindingskabel tussen DAI en TV gebruikt kan blijven worden.

Dan nu het grote moment: Verbind de DAI met de monitor en schakel beide apparaten in. Nu moet het u bekende plaatje verschijnen.
Is dit niet het geval, dan zal waarschijnlijk de belasting van de TV op de DAI te groot zijn, of condensator C is te klein gekozen. In het laatste geval is dat te zien aan het niet egaal van 'kleur' (helderheid) zijn van het beeld. Geprobeerd kan worden of een grotere waarde van condensator C verbetering brengt, maar let bij het vervangen van C door een elektrolytisch type op de polariteit.
De gelijkspanning op de DAI uitgang is ongeveer 2,4 Volt en de spanning op de basis van de video eindtransistor in de TV moet u meten of berekenen aan de hand van de waarden R inst, R en de voedingsspanning in de TV.
Mocht een grotere capaciteit van C (tot ongeveer 100 micro Farad) geen oplossing brengen, dan kunt u nog proberen om de modulator in de DAI uit te schakelen door het signaal draadje waarvan we de beeldinformatie aftakken door te knippen. De oorspronkelijke TV uitgang van de DAI werkt dan echter niet meer !

Wanneer ook dit niet helpt en u weet zeker dat alles goed is aangesloten, wijzig dan zowel R als R inst in de TV. Maak beide 2 x zo groot, regel R inst weer af zoals beschreven en probeer het opnieuw.
Als dit tenslotte niet wil werken, dan moet u gaan denken aan een extra buffer (aanpassings schakeling) en het zou te ver voeren om deze oplossing hier te bespreken..

Tot slot nog wat algemene opmerkingen:

- Begin alleen maar aan de ombouw wanneer u weet dat u de ombouw tot een goed einde kunt brengen.
- Let vooral tijdens het aanbrengen van de wijzigingen erop dat geen stukjes koperdraad achterblijven in de computer of de TV. Dit kan kortsluiting veroorzaken !
- Werk zeer zorgvuldig en kontroleer achteraf altijd of u de veranderingen juist heeft aangebracht.
- Door schrijver dezes is bijna 1 jaar geleden een TV merk Audio Sonic type TC 3107 tot monitor omgebouwd en dit werkt nog steeds tot volle tevredenheid.
  De punten in Mode 5 en 6 zijn scherp te onderscheiden...
  Opgemerkt moet verder worden dat deze ervaring is opgedaan met een 'Zwart/Wit' computer (Teleac uitvoering) met 32 kByte uitbreiding. Ervaringen van mede hobbyisten met 'kleur' machines zijn welkom.
- Last but not least: Door bovengenoemde wijziging verdwijnt het geluid van de TV in de stand Monitor.

** S U C C E S **

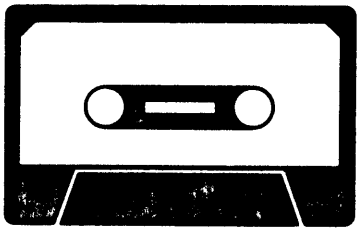
Hans Bakker

CATALOG

PROGRAMMABIBLIOTHEEK : NIEUWE FORMULE

Voortaan zullen de programma's alleen nog verkrijgbaar zijn op verzamelcassettes.Op deze manier kunnen we efficienter werken en de programma's voordeliger aanbieden.
We vermelden nu allen de titels,uitvoerige omschrijving in ons vakantienummer 6.De programma's in BASIC zijn steeds met listing: bij iedere collectie krijgt U dus een boekwerkje.

GAMES COLLECTION 1 35gld/500 Bfr



YATHZEE	SIP
AWARI	SIP
SUBMARINE	BAKKER
KANONSPEL	BAKKER
OTHELLO	SIP
STARTREK	BAKKER
REACTIETEST	V.COOTEN
LUNAR LANDING	BAKKER

VIER OP EEN RIJ(tegen de computer) BAKKER

GAMES COLLECTION 2 35 Gld/500 Bfr

INPAKKEN	DRUIJFF
BREAKOUT	SIP
AMAZING	
KIM CACHE	SIP
*MASTERMIND	SMIT
GEITENSPEL	
SPACE INVADERS	PHILLIPART
HANNIBAL 2000	
TOWERS OF HANOI+SOLUTION	

GAMES COLLECTION 3 35 Gld/500Bfr

BACKGAMMON	BAKKER
BARRICADE	DRUIJFF
SLANG	V.D.WORP
HAP MAAR	DRUIJFF
INVASION	DRUIJFF
ROBOTS	SIP
TRAFFIC TEST	
LIFE (ml+BASIC)	V.ECK

FGT APPLICATIONS 1 84 Gld/1250 Bf

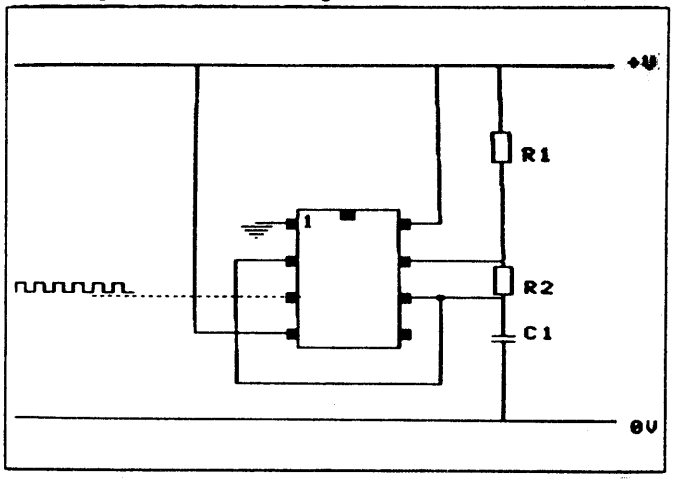
ALGORITHM OF HORNER	
555 QUICK DESIGN	
FGT TABLE pictures	
MATH COMPETITION	
TV-TENNIS	DRUIJFF
SUPERWURM	WASSERMANN
MASTERMIND	SMIT
CLOCK TRAINING	
FGT-PADDLES	

FGT PACKAGE

84 Gld
1250 Bfr

OBJ 29B-900
DEMO FGT
TABLE CREATOR
WORG GAME
29B-BFF pictures + minuscules
DEMO
29B-AFF shadow characters
DEMO
29B-1150 DAI GRAPHIC CHARACTER SET
DEMO
29B-AAO Greece alfabet
DEMO
29B-AAF Trigisch alfabet (SF)
+++ russian alfabet
+++ MORSE alfabet
+++ ???

Teaching electronics — design of 555 timer circuit.



TIMING OF MATH-FUNCTIONS(WITH AMD9511

```
>>>>>> LOOP-TIME :        0 MIN        6SEC         400MSEC

>>>>>> LOOP-TIME :        0 MIN        6SEC         400MSEC

 0
256   RETURN
EXECUTION TIME WITH       : 0 MIN      6 SEC        400 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      6 SEC        400 MSEC
CALCULATION TIME WITH     : 0.0 MSEC
CALCULATION TIME WITHOUT  : 0.0 MSEC
-------------------------------------------------------------------------

  1
257   X=J:RETURN
EXECUTION TIME WITH       : 0 MIN      7 SEC        560 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      7 SEC        560 MSEC
CALCULATION TIME WITH     : 1160.0 MSEC
CALCULATION TIME WITHOUT  : 1160.0 MSEC
-------------------------------------------------------------------------

  2
258   X%=J:RETURN
EXECUTION TIME WITH       : 0 MIN      10 SEC       600 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      11 SEC       660 MSEC
CALCULATION TIME WITH     : 4200.0 MSEC
CALCULATION TIME WITHOUT  : 5260.0 MSEC
-------------------------------------------------------------------------

  3
259   X=I1%:RETURN
EXECUTION TIME WITH       : 0 MIN      10 SEC       760 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      12 SEC       360 MSEC
CALCULATION TIME WITH     : 4360.0 MSEC
CALCULATION TIME WITHOUT  : 5960.0 MSEC
-------------------------------------------------------------------------

  4
260   X%=1:RETURN
EXECUTION TIME WITH       : 0 MIN      7 SEC        880 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      7 SEC        880 MSEC
CALCULATION TIME WITH     : 1480.0 MSEC
CALCULATION TIME WITHOUT  : 1480.0 MSEC
-------------------------------------------------------------------------

  5
261   X=1.0:RETURN
EXECUTION TIME WITH       : 0 MIN      8 SEC        20 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      8 SEC        20 MSEC
CALCULATION TIME WITH     : 1620.0 MSEC
CALCULATION TIME WITHOUT  : 1620.0 MSEC
-------------------------------------------------------------------------

  6
262   X%=I1%:RETURN
EXECUTION TIME WITH       : 0 MIN      8 SEC        280 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      8 SEC        280 MSEC
CALCULATION TIME WITH     : 1880.0 MSEC
CALCULATION TIME WITHOUT  : 1880.0 MSEC
-------------------------------------------------------------------------
```

```
  7
263   I3%=I1%+I2%:RETURN
EXECUTION TIME WITH       : 0 MIN     12 SEC      80 MSEC
EXECUTION TIME WITHOUT    : 0 MIN     12 SEC     140 MSEC
CALCULATION TIME WITH     : 5680.0 MSEC
CALCULATION TIME WITHOUT  : 5740.0 MSEC
-------------------------------------------------------------------------------
  8
264   I3%=I1%-I2%:RETURN
EXECUTION TIME WITH       : 0 MIN     12 SEC     220 MSEC
EXECUTION TIME WITHOUT    : 0 MIN     12 SEC     300 MSEC
CALCULATION TIME WITH     : 5820.0 MSEC
CALCULATION TIME WITHOUT  : 5900.0 MSEC
-------------------------------------------------------------------------------
  9
265   I3%=I1%/I2%:RETURN
EXECUTION TIME WITH       : 0 MIN     12 SEC     340 MSEC
EXECUTION TIME WITHOUT    : 0 MIN     14 SEC     280 MSEC
CALCULATION TIME WITH     : 5940.0 MSEC
CALCULATION TIME WITHOUT  : 7880.0 MSEC
-------------------------------------------------------------------------------
  10
266   I3%=I1%*1000:RETURN
EXECUTION TIME WITH       : 0 MIN     12 SEC     360 MSEC
EXECUTION TIME WITHOUT    : 0 MIN     14 SEC     680 MSEC
CALCULATION TIME WITH     : 5960.0 MSEC
CALCULATION TIME WITHOUT  : 8280.0 MSEC
-------------------------------------------------------------------------------
  11
267   F=J+J:RETURN
EXECUTION TIME WITH       : 0 MIN     12 SEC     660 MSEC
EXECUTION TIME WITHOUT    : 0 MIN     13 SEC     760 MSEC
CALCULATION TIME WITH     : 6260.0 MSEC
CALCULATION TIME WITHOUT  : 7360.0 MSEC
-------------------------------------------------------------------------------
  12
268   F=J-J:RETURN
EXECUTION TIME WITH       : 0 MIN     12 SEC     800 MSEC
EXECUTION TIME WITHOUT    : 0 MIN     13 SEC     620 MSEC
CALCULATION TIME WITH     : 6400.0 MSEC
CALCULATION TIME WITHOUT  : 7220.0 MSEC
-------------------------------------------------------------------------------
  13
269   F=J/J:RETURN
EXECUTION TIME WITH       : 0 MIN     12 SEC     960 MSEC
EXECUTION TIME WITHOUT    : 0 MIN     20 SEC     980 MSEC
CALCULATION TIME WITH     : 6560.0 MSEC
CALCULATION TIME WITHOUT  : 14580.0 MSEC
-------------------------------------------------------------------------------
  14
270   F=J*J:RETURN
EXECUTION TIME WITH       : 0 MIN     13 SEC      80 MSEC
EXECUTION TIME WITHOUT    : 0 MIN     16 SEC     400 MSEC
CALCULATION TIME WITH     : 6680.0 MSEC
CALCULATION TIME WITHOUT  : 10000.0 MSEC
-------------------------------------------------------------------------------
  15
271   F=J^2.0:RETURN
EXECUTION TIME WITH       : 0 MIN     21 SEC     720 MSEC
EXECUTION TIME WITHOUT    : 2 MIN     51 SEC     160 MSEC
CALCULATION TIME WITH     : 15320.0 MSEC
CALCULATION TIME WITHOUT  : 1.6476E5 MSEC
-------------------------------------------------------------------------------
```

```
  16
272   F=SQR(J):RETURN
EXECUTION TIME WITH       : 0 MIN     12 SEC      700 MSEC
EXECUTION TIME WITHOUT    : 0 MIN     40 SEC      760 MSEC
CALCULATION TIME WITH     : 6300.0 MSEC
CALCULATION TIME WITHOUT  : 34360.0 MSEC
-------------------------------------------------------------------------------
  17
273   F=J^0.5:RETURN
EXECUTION TIME WITH       : 0 MIN     21 SEC      960 MSEC
EXECUTION TIME WITHOUT    : 2 MIN     52 SEC      620 MSEC
CALCULATION TIME WITH     : 15560.0 MSEC
CALCULATION TIME WITHOUT  : 1.6622E5 MSEC
-------------------------------------------------------------------------------
  18
274   F=SIN(J):RETURN
EXECUTION TIME WITH       : 0 MIN     16 SEC      380 MSEC
EXECUTION TIME WITHOUT    : 1 MIN     34 SEC      240 MSEC
CALCULATION TIME WITH     : 9980.0 MSEC
CALCULATION TIME WITHOUT  : 87840.0 MSEC
-------------------------------------------------------------------------------
  19
275   F=COS(J):RETURN
EXECUTION TIME WITH       : 0 MIN     17 SEC      0 MSEC
EXECUTION TIME WITHOUT    : 1 MIN     35 SEC      760 MSEC
CALCULATION TIME WITH     : 10600.0 MSEC
CALCULATION TIME WITHOUT  : 89360.0 MSEC
-------------------------------------------------------------------------------
  20
276   F=TAN(J):RETURN
EXECUTION TIME WITH       : 0 MIN     17 SEC      980 MSEC
EXECUTION TIME WITHOUT    : 3 MIN     9 SEC       520 MSEC
CALCULATION TIME WITH     : 11580.0 MSEC
CALCULATION TIME WITHOUT  : 1.8312E5 MSEC
-------------------------------------------------------------------------------
  21
277   F=ASIN(F2):RETURN
EXECUTION TIME WITH       : 0 MIN     19 SEC      120 MSEC
EXECUTION TIME WITHOUT    : 2 MIN     16 SEC      40 MSEC
CALCULATION TIME WITH     : 12720.0 MSEC
CALCULATION TIME WITHOUT  : 1.2964E5 MSEC
-------------------------------------------------------------------------------
  22
278   F=ACOS(F2):RETURN
EXECUTION TIME WITH       : 0 MIN     19 SEC      340 MSEC
EXECUTION TIME WITHOUT    : 2 MIN     18 SEC      780 MSEC
CALCULATION TIME WITH     : 12940.0 MSEC
CALCULATION TIME WITHOUT  : 1.3238E5 MSEC
-------------------------------------------------------------------------------
  23
279   F=ATN(J):RETURN
EXECUTION TIME WITH       : 0 MIN     18 SEC      460 MSEC
EXECUTION TIME WITHOUT    : 1 MIN     20 SEC      440 MSEC
CALCULATION TIME WITH     : 12060.0 MSEC
CALCULATION TIME WITHOUT  : 74040.0 MSEC
-------------------------------------------------------------------------------
  24
280   F=ALOG(F2):RETURN
EXECUTION TIME WITH       : 0 MIN     18 SEC      100 MSEC
EXECUTION TIME WITHOUT    : 1 MIN     50 SEC      420 MSEC
CALCULATION TIME WITH     : 11700.0 MSEC
CALCULATION TIME WITHOUT  : 1.0402E5 MSEC
-------------------------------------------------------------------------------
```

```
  25
281   F=EXP(F2):RETURN
EXECUTION TIME WITH       : 0 MIN      17 SEC       400 MSEC
EXECUTION TIME WITHOUT    : 1 MIN      27 SEC       940 MSEC
CALCULATION TIME WITH     : 11000.0 MSEC
CALCULATION TIME WITHOUT  : 81540.0 MSEC
-------------------------------------------------------------------------------
  26
282   F=LOG(J):RETURN
EXECUTION TIME WITH       : 0 MIN      17 SEC       920 MSEC
EXECUTION TIME WITHOUT    : 1 MIN      28 SEC       740 MSEC
CALCULATION TIME WITH     : 11520.0 MSEC
CALCULATION TIME WITHOUT  : 82340.0 MSEC
-------------------------------------------------------------------------------
  27
283   F=LOGT(J):RETURN
EXECUTION TIME WITH       : 0 MIN      18 SEC       200 MSEC
EXECUTION TIME WITHOUT    : 1 MIN      34 SEC       460 MSEC
CALCULATION TIME WITH     : 11800.0 MSEC
CALCULATION TIME WITHOUT  : 88060.0 MSEC
-------------------------------------------------------------------------------
  28
284   F=FRAC(F1):RETURN
EXECUTION TIME WITH       : 0 MIN      15 SEC       480 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      19 SEC       660 MSEC
CALCULATION TIME WITH     : 9080.0 MSEC
CALCULATION TIME WITHOUT  : 13260.0 MSEC
-------------------------------------------------------------------------------
  29
285   F=INT(F1):RETURN
EXECUTION TIME WITH       : 0 MIN      15 SEC       880 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      16 SEC       440 MSEC
CALCULATION TIME WITH     : 9480.0 MSEC
CALCULATION TIME WITHOUT  : 10040.0 MSEC
-------------------------------------------------------------------------------
  30
286   F=SGN(J):RETURN
EXECUTION TIME WITH       : 0 MIN      16 SEC       660 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      17 SEC       20 MSEC
CALCULATION TIME WITH     : 10260.0 MSEC
CALCULATION TIME WITHOUT  : 10620.0 MSEC
-------------------------------------------------------------------------------
  31
287   F=ABS(J):RETURN
EXECUTION TIME WITH       : 0 MIN      14 SEC       960 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      15 SEC       240 MSEC
CALCULATION TIME WITH     : 8560.0 MSEC
CALCULATION TIME WITHOUT  : 8840.0 MSEC
-------------------------------------------------------------------------------
  32
288   F=FREQ(I1%):RETURN
EXECUTION TIME WITH       : 0 MIN      21 SEC       420 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      33 SEC       220 MSEC
CALCULATION TIME WITH     : 15020.0 MSEC
CALCULATION TIME WITHOUT  : 26820.0 MSEC
-------------------------------------------------------------------------------
  33
289   F=I1% IAND I2%:RETURN
EXECUTION TIME WITH       : 0 MIN      17 SEC       160 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      18 SEC       100 MSEC
CALCULATION TIME WITH     : 10760.0 MSEC
CALCULATION TIME WITHOUT  : 11700.0 MSEC
-------------------------------------------------------------------------------
```

```
  34
290   F=I1% SHL 3.0:RETURN
EXECUTION TIME WITH       : 0 MIN      20 SEC       760 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      23 SEC       20 MSEC
CALCULATION TIME WITH     : 14360.0 MSEC
CALCULATION TIME WITHOUT  : 16620.0 MSEC
-------------------------------------------------------------------------------
  35
291   F=I1% MOD 7.0:RETURN
EXECUTION TIME WITH       : 0 MIN      21 SEC       640 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      33 SEC       640 MSEC
CALCULATION TIME WITH     : 15240.0 MSEC
CALCULATION TIME WITHOUT  : 27240.0 MSEC
-------------------------------------------------------------------------------
  36
292   F=PI:RETURN
EXECUTION TIME WITH       : 0 MIN      14 SEC       360 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      14 SEC       500 MSEC
CALCULATION TIME WITH     : 7960.0 MSEC
CALCULATION TIME WITHOUT  : 8100.0 MSEC
-------------------------------------------------------------------------------
  37
293   RETURN
EXECUTION TIME WITH       : 0 MIN      11 SEC       640 MSEC
EXECUTION TIME WITHOUT    : 0 MIN      11 SEC       660 MSEC
CALCULATION TIME WITH     : 5240.0 MSEC
CALCULATION TIME WITHOUT  : 5260.0 MSEC
-------------------------------------------------------------------------------
WITHOUT
 0
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
```

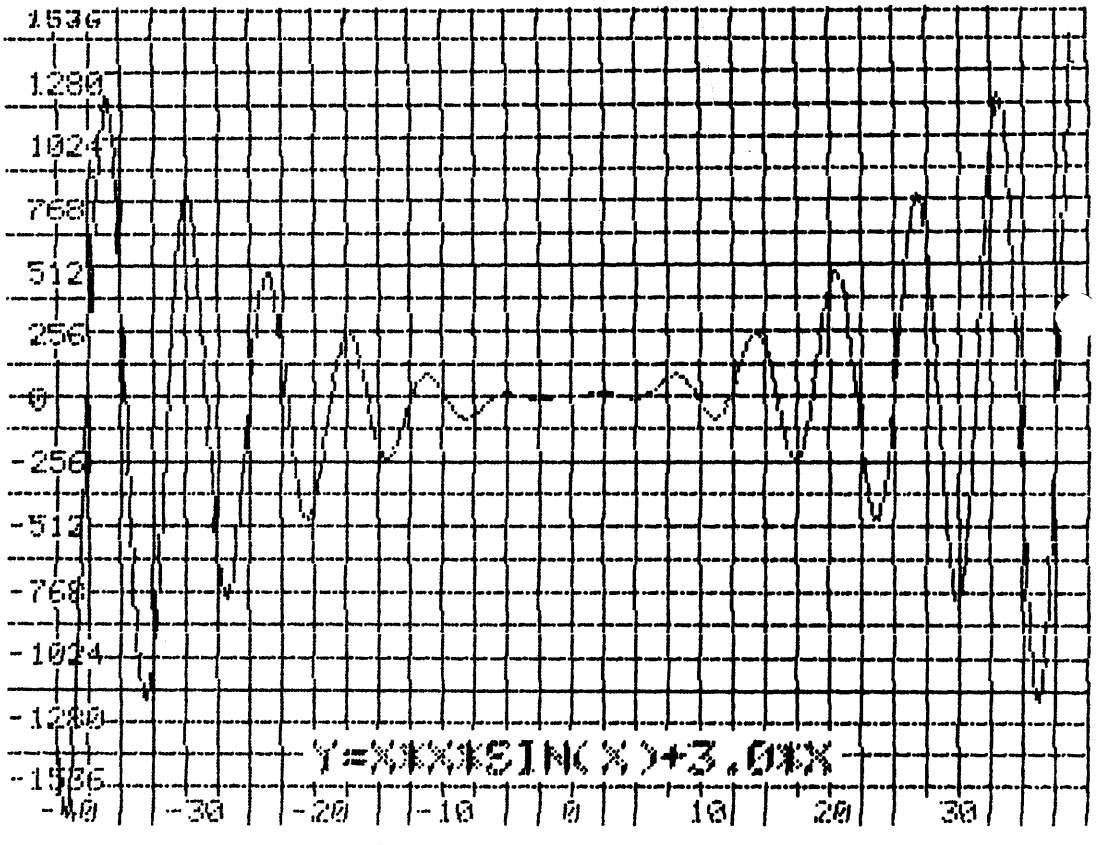
CALCULATION TIMES WITH CORRECTION:

N	TIME WITH	% WITH	TIME WITHOUT	% WITHOUT	RATIO (%)
0	0.0	0.0	0.0	0.0	100
1	1018.38	7.717	1017.84	0.564	100
2	3916.76	29.682	4975.67	2.76	127
3	3935.13	29.821	5533.51	3.069	141
4	913.513	6.922	911.351	0.505	100
5	911.891	6.91	909.189	0.504	100
6	1030.27	7.807	1027.03	0.569	100
7	4688.65	35.531	4744.86	2.631	101
8	4687.03	35.519	4762.7	2.641	102
9	4665.4	35.355	6600.54	3.661	141
10	4543.78	34.433	6858.38	3.804	151

N	TIME WITH	% WITH	TIME WITHOUT	% WITHOUT	RATIO (%)
11	4702.16	35.634	5796.21	3.215	123
12	4700.54	35.621	5514.05	3.058	117
13	4718.92	35.761	12731.9	7.062	270
14	4697.3	35.597	8009.73	4.443	171
15	13195.7	99.999	1.62628E5	90.209	1232
16	4034.05	30.571	32085.4	17.797	795
17	13152.4	99.672	1.63803E5	90.862	1245
18	7430.81	56.312	85281.1	47.305	1148
19	7909.19	59.937	86658.9	48.069	1096
20	8747.56	66.291	1.80277E5	99.999	2061
21	9745.94	73.857	1.26655E5	70.255	1300
22	9824.32	74.451	1.29252E5	71.696	1316
23	8802.7	66.708	70770.3	39.256	804
24	8301.08	62.907	1.00608E5	55.807	1212
25	7459.46	56.529	77985.9	43.259	1045
26	7837.84	59.397	78643.8	43.623	1003
27	7976.22	60.445	84221.6	46.717	1056
28	5114.59	38.759	9279.46	5.147	181
29	5372.97	40.717	5917.3	3.282	110
30	6011.35	45.555	6355.13	3.525	106
31	4169.73	31.599	4432.97	2.458	106
32	10488.1	79.481	22270.8	12.353	212
33	6086.49	46.124	7008.65	3.887	115
34	9544.86	72.333	11786.5	6.537	123
35	10283.2	77.928	22264.3	12.35	217
36	2861.62	21.686	2982.16	1.654	104
37	0.0	0.0	0.0	0.0	100

*

```
20   PRINT :POKE #FF06,#E:PRINT "TIMING OF MATH-FUNCTIONS (WITH AMD 9511)"
22   PRINT :POKE #94C,0:POKE #A32,0
25   FOR N=0.0 TO 79.0:PRINT CHR$(166);:NEXT:PRINT
30   CLEAR 1000:DIM TA(38.0,2.0),M1%(2.0),S1%(2.0),MS1%(2.0),MAX(2.0):I1%=75648:I2%=958473:F1=867.565:F2=0.96857
100  FOR N=0.0 TO 37.0
110  POKE #94C,N:POKE #A32,N
120  W=0.0
123  IF W=1.0 THEN POKE #D4,#7B
124  IF W=0 THEN POKE #D4,0
150  FOR I%=0 TO 4:POKE #3B9+I%,0:NEXT
160  CALLM #340
170  FOR J=1.0 TO 2000.0
175  GOSUB 293
180  NEXT J
190  CALLM #350
200  IF N=0 GOTO 4000
202  M1%(W)=PEEK(#3BD)*10+PEEK(#3BC):S1%(W)=PEEK(#3BB)*10+PEEK(#3BA):MS1%(W)=PEEK(#3B9)*20
203  TA(N,W)=1000.0*((M1%(W)-M%)*60.0+(S1%(W)-S%))+MS1%(W)-MS%
204  W=W+1.0:IF W<2.0 THEN 123
205  NNN%=INT(N):POKE #FF06,#E:PRINT NNN%
210  LIST 293
230  PRINT "EXECUTION TIME WITH      :";M1%(1.0);" MIN",S1%(1.0);" SEC",MS1%(1.0);" MSEC"
232  PRINT "EXECUTION TIME WITHOUT   :";M1%(0.0);" MIN",S1%(0.0);" SEC",MS1%(0.0);" MSEC"
235  PRINT "CALCULATION TIME WITH    :";1000.0*((M1%(1.0)-M%)*60.0+(S1%(1.0)-S%))+MS1%(1.0)-MS%;" MSEC"
237  PRINT "CALCULATION TIME WITHOUT :";1000.0*((M1%(0.0)-M%)*60.0+(S1%(0.0)-S%))+MS1%(0.0)-MS%;" MSEC"
245  FOR NN=0.0 TO 79.0:PRINT "-";:NEXT
247  PRINT
250  NEXT N
251  GOTO 5000
252  REM ************************************************
256  RETURN
257  X=J:RETURN
258  X%=J:RETURN
259  X=I1%:RETURN
260  X%=1:RETURN
261  X=1.0:RETURN
262  X%=I1%:RETURN
263  I3%=I1%+I2%:RETURN
264  I3%=I1%-I2%:RETURN
265  I3%=I1%/I2%:RETURN
266  I3%=I1%*1000:RETURN
267  F=J+J:RETURN
268  F=J-J:RETURN
269  F=J/J:RETURN
270  F=J*J:RETURN
271  F=J^2.0:RETURN
272  F=SQR(J):RETURN
273  F=J^0.5:RETURN
274  F=SIN(J):RETURN
275  F=COS(J):RETURN
276  F=TAN(J):RETURN
277  F=ASIN(F2):RETURN
278  F=ACOS(F2):RETURN
279  F=ATN(J):RETURN
280  F=ALOG(F2):RETURN
281  F=EXP(F2):RETURN
282  F=LOG(J):RETURN
283  F=LOGT(J):RETURN
284  F=FRAC(F1):RETURN
285  F=INT(F1):RETURN
286  F=SGN(J):RETURN
287  F=ABS(J):RETURN
288  F=FREQ(I1%):RETURN
289  F=I1% IAND I2%:RETURN
290  F=I1% SHL 3.0:RETURN
291  F=I1% MOD 7.0:RETURN
292  F=PI:RETURN
293  RETURN
```



uit het programma "FUNCTIE-PLOT" van H.BAKKER

```
3999  REM ##############################################
4000  MZ=PEEK(#3BD)#10+PEEK(#3BC):SZ=PEEK(#3BB)#10+PEEK(#3BA):MSZ=PEEK(#3B9)#20
4010  PRINT ">>>>>> LOOP-TIME :",MZ;" MIN",SZ;"SEC",MSZ;"MSEC":PRINT
4020  GOTO 202
5000  W=0.0
5004  KK=(TA(37.0,W)-TA(0.0,W))/37.0:TAA=TA(0.0,W)
5005  FOR NZ=0 TO 37:TA(NZ,W)=TA(NZ,W)-(KK#NZ+TAA):NEXT
5007  MAX(W)=TA(0.0,W):FOR NZ=0 TO 37:IF TA(NZ,W)>MAX(W) THEN MAX(W)=TA(NZ,W)
5010  NEXT:A$="WITH":IF W=0.0 THEN A$=A$+"OUT"
5015  PRINT :POKE #FF06,#E:PRINT A$:PRINT
5020  K=76.0/MAX(W)
5030  FOR NZ=0 TO 37
5040  IF NZ<10 THEN PRINT " ";
5042  IF NZ<100.0 THEN PRINT " ";
5044  PRINT NZ;
5050  FOR JJ=0.0 TO INT(TA(NZ,W)#K):PRINT CHR$(166);:NEXT:PRINT
5060  NEXT NZ:W=W+1.0:IF W<2.0 THEN 5004
5070  PRINT :POKE #FF06,#E:PRINT "CALCULATION TIMES WITH CORRECTION:":PRINT
5072  PRINT "=============================================================="
5075  PRINT "!  N    TIME       %        TIME       %       RATIO !"
5076  PRINT "!       WITH      WITH     WITHOUT    WITHOUT     (%)  !"
5077  PRINT "=============================================================="
5080  PRINT :FOR NZ=0 TO 37
5081  PROC1=100.0/MAX(1.0):PROC0=100.0/MAX(0.0)
5082  TW=(INT(TA(NZ,1.0)#1000.0))/1000.0:PRW=(INT(TA(NZ,1.0)#PROC1#1000.0))/1000.0
5083  TN=(INT(TA(NZ,0.0)#1000.0))/1000.0:PRN=(INT(TA(NZ,0.0)#PROC0#1000.0))/1000.0
5087  VERHZ=INT((TA(NZ,0.0)+1E-5)/(TA(NZ,1.0)+1E-5)#100.0+0.5)
5088  IF NZ<10.0 THEN PRINT " ";
5089  IF NZ<100.0 THEN PRINT " ";
5090  PRINT NZ;TAB(7);TW;TAB(18);PRW;TAB(29);TN;TAB(40);PRN;TAB(51);VERHZ
5095  PRINT
5100  NEXT
```

START/STOP VIA INTERRUPT 7

```
PAGE 01

001                     *deze  routine verzorgt START/STOP
002                     *van  BASIC & machinetaal programm
003                     *besturimg met EVENT 1 (ANI 32)
004                     *           of EVENT 2 (ANI 16)
005                     *zet interrupt 7 op #300 door:
006                     *V7D9A9-300  (=D9A9 vervangen door  300)
007                             ORG   :300
008 0300 F3                     DI
009 0301 F5                     PUSH  PSW
010 0302 3A00FD         TEST    LDA   :FD00      input port
011 0305 E620                   ANI   32         maskeer bit 5 (of 4)
012 0307 C20203                 JNZ   TEST       indien ingedrukt:loop TEST
013 030A F1                     POP   PSW
014 030B FB                     EI
015 030C C3A9D9                 JMP   :D9A9      vevolg interrupt 7 procedure
016 030F                        END

****************************
* S Y M B O L   T A B L E *
****************************

TEST    0302
```

VERGELIJKENDE STUDIE OVER DE MATHCHIP AMD 9511

Door zijn grote verscheidenheid aan mathematische ofte wiskundige functies is DAIpc al een grote uitblinker.
We kunnen ons echter de vraag stellen hoe het met de uitvoeringstijden van de verschillende functies zit. Een ander belangrijk probleem voor vele personal-computers-gebruikers stelt het al dan niet aankopen van de oh zo dure AMD 9511 "MATHCHIP".
Krijgen we waar voor ons geld? Is de relatief grote som voor de aankoop van deze mathchip wel verantwoord voor een of andere toepassing? Over deze en andere gelijkaardige vragen zullen we proberen wat meer duidelijkheid te brengen.
Daarom heb ik een BASIC-programma geschreven dat de looptijden van de verschillende BASIC-functies en sommige BASIC-statements nagaat. De tijden worden achtereenvolgens gemeten met en zonder MATHCHIP.
nb: Indien men een MATHCHIP in het toestel heeft kan men die disabelen of inactief maken door POKE éD4,0.
actief door POKE éD4,é7B.

Toelichting bij het programma:

30 Initialisatie van de veranderlijken
Ik laat U nu raden wat lijn 22 juist doet.
100 HOOFD-FOR-NEXT lus.Er worden 37 functies getest.
110 wat doen deze POKES hier ?
120 W=1: met MATHCHIP W=0: zonder
124 zet MATHCHIP aan of uit.
160 reset de chronometer en start hem (CALLMé340)
190 voer de functie 2000 maal uit en stop chrono.
203 bereken de tijden en sla deze op in een tabel (TA).
247 druk de resultaten af voor elke functie.
251 indien de 37 functies werden getimed ga dan naar 5000.
293 hier bevinden zich de te testen functies.
4000 deze routine wordt aangeroepen voor de 0-functie ttz een RETURN zonder meer. De bedoeling is dat de tijd benodigd voor het uitvoeren van de lus wordt afgetrokken van de BRUTO-tijd.Doch er volgen complicaties !!
5000 deze routine gaat alle tijden statistisch vergelijken zodat een zinvolle vergelijking mogelijk wordt.
5050 hier wordt een BARGRAPH afgedrukt.
5070 Nu de complicaties...
functie 37 is ook een RETURN en wel om de volgende reden:
De computer heeft wat tijd nodig om een bepaalde subroutine te vinden en hoe verder de subroutine in het geheugen zit, hoe langer deze zoektijd duurt.
Ik heb verondersteld dat de zoektijd lineair toeneemt.
Dit lijkt me een redelijke veronderstelling vermits de verschillende subroutines ongeveer even lang zijn.
Er wordt daarom geinterpoleerd tussen de 2 RETURN-tijden deze tijd wordt dan afgetrokken van de looptijd.
Dit geeft dan TIME WITH en TIME without in de tabel.
Vervolgens worden deze tijden procentueel vergeleken met de traagste functie met en zonder MATHCHIP.
Om te besluiten wordt de onderlinge verhouding berekend.

VERGELIJKENDE STUDIE MATHCHIP

U vraagt zich misschien nog steeds af wat lijnen 22 en 110 doen?
En hoe komt het dat elke lijn apart gelist wordt?
En waarom geen ON GOSUB maar GOSUB met een lijnnummer?
Het antwoord is zeer eenvoudig: Het programma verandert zichzelf.
Na enig zoekwerk (lees PEEK&POKEwerk) werden de bytes gevonden die de geliste regelnummer en de subroutine waarnaar gesprongen wordt bepalen. Aldus is het mogelijk deze tijdens de loop van programma aan te passen.Misschien mogen we hier wel spreken van "SELF-MODIFYING-CODE".
Het werkt in ieder geval prima en is volgens mij de beste manier om een zo constant mogelijke looptime te creeren.
Een belangrijk gevolg van deze manier van werken is echter dat het een uiterst delicate zaak wordt om het programma te wijzigen.
Tevens wil ik de collega's die het programma wensen in te tikken waarschuwen:In ieder geval SAVEN voor RUN !!
Ook niet vergeten de HEAPPOINTERS te veranderen!
Het machinetaalprogramma start op é300 en loopt tot é3B9.
Op te merken zijn de twee routines op é 340 en é 350 om de chrono te starten en te stoppen.Een dergelijke methode is ook nodig om de real-time clock uit DAInamic N3 -waarvan de chrono is afgeleid-, te starten en te stoppen.

INTERPRETATIE VAN DE RESULTATEN

1. De real-time clock heeft een resolutie van 20 msec.
   We moeten dus alle resultaten beschouwen met een fout die tot 20 msec kan oplopen.

2. De tijd die de real-time clock elke 20 msec inneemt is erg kort,dit kunnen we in de berekeningen verwaarlozen.

3. Feitelijk zou moeten aangetoond worden dat het zoeken naar een lijnnummer lineair verloopt.

4. Er zijn nog vele andere functies die men zou kunnen testen.
   Een machtsverheffing neemt niet steeds evenveel tijd in beslag.
   (vergelijk test 15 en 17)

5.

> De BARGRAPHS zijn niet op dezelfde schaal getekend !!
> De prestaties MET MATHCHIP zijn 13,6 maal VERGROOT !

6. De RND-functies werden niet opgenomen in de test omdat ze werkelijk teveel tijd in beslag nemen.

RND(0)	1 min en 10 sec
RND(J)	48 sec
RND(0)XJ	1 min en 15 sec

BESLUIT

Of de AMD 9511 zijn geld waard is zal ieder voor zich moeten uitmaken.Het staat echter als een paal boven water dat de MATHCHIP het rekenwerk enorm versnelt.Ik durf geen prijs op te geven omdat deze nogal durft veranderen.Vraag wel de 9511 A,de oudere versie had -ik ben er niet zeker van- een maskfout.
Tot binnen twee maand en ..... LET THEM TICK

Jos Schepens

```
PAGE 01     :CLOCK FOR MATH TEST

002                     *339 = UNITS OF 20 MSEC
003                     *33A = UNITS OF SEC
004                     *33B = TENS OF SEC
005                     *33C = UNITS OF MIN
006                     *33D = TENS OF MIN
007                           ORG   :300
008 0300 E5                   PUSH  H
009 0301 D5                   PUSH  D
010 0302 F5                   PUSH  PSW
011 0303 213903               LXI   H,:339      20 MSEC
012 0306 1E32                 MVI   E,:32
013 0308 34                   INR   M
014 0309 7B                   MOV   A,E
015 030A BE                   CMP   M
016 030B C23103               JNZ   EXIT1
017 030E C5                   PUSH  B
018 030F 060A                 MVI   B,:A
019 0311 0E06                 MVI   C,:6
020 0313 1600                 MVI   D,:0
021 0315 72                   MOV   M,D
022 0316 23                   INX   H           SEC
023 0317 34                   INR   M
024 0318 78                   MOV   A,B
025 0319 BE                   CMP   M
026 031A C23003               JNZ   EXIT2
027 031D 72                   MOV   M,D
028 031E 23                   INX   H           TENS OF SEC
029 031F 34                   INR   M
030 0320 79                   MOV   A,C
031 0321 BE                   CMP   M
032 0322 C23003               JNZ   EXIT2
033 0325 72                   MOV   M,D
034 0326 23                   INX   H           MINUTES
035 0327 34                   INR   M
036 0328 78                   MOV   A,B
037 0329 BE                   CMP   M
038 032A C23003               JNZ   EXIT2
039 032D 72                   MOV   M,D
040 032E 23                   INX   H           TENS MIN
041 032F 34                   INR   M
042 0330 C1           EXIT2   POP   B
043 0331 F1           EXIT1   POP   PSW
044 0332 D1                   POP   D
045 0333 E1                   POP   H
046 0334 C3A9D9               JMP   :D9A9       CONT INTERRUPT PROC
047                           ORG   :340        START TIMER
048 0340 F3                   DI
049 0341 F5                   PUSH  PSW
050 0342 3E00                 MVI   A,:0        CHANGE VECTOR
051 0344 327000               STA   :70         TO OUR ROUTINE
052 0347 3E03                 MVI   A,:3
053 0349 327100               STA   :71
054 034C F1                   POP   PSW
```

```
PAGE 02     :CLOCK FOR MATH TEST

055 034D FB                     EI
056 034E C9                     RET
057                             ORG    :350        STOP TIMER
058 0350 F3                     DI
059 0351 F5                     PUSH   PSW
060 0352 3EA9                   MVI    A,:A9       OLD VECTOR
061 0354 327000                 STA    :70
062 0357 3ED9                   MVI    A,:D9
063 0359 327100                 STA    :71
064 035C F1                     POP    PSW
065 035D FB                     EI
066 035E C9                     RET
067 035F              END       END

****************************
* S Y M B O L   T A B L E *
****************************

END     035F    EXIT1   0331    EXIT2   0330


    ]#P.
0300 E5 D5 F5 21 39 03 1E 32 34 7B BE C2 31 03 C5 06
0310 0A 0E 06 16 00 72 23 34 78 BE C2 30 03 72 23 34
0320 79 BE C2 30 03 72 23 34 78 BE C2 30 03 72 23 34
0330 C1 F1 D1 E1 C3 A9 D9

0340 F3 F5 3E 00 32 70 00 3E 03 32 71 00 F1 FB C9

0350 F3 F5 3E A9 32 70 00 3E D9 32 71 00 F1 FB C9

    ]


5       FOR I=0 TO 4:POKE #339+I,0:NEXT
10      INPUT "START";S$
20      CALLM #340:REM START
30      A=GETC:IF A=0 THEN 30
40      CALLM #350:REM STOP
50      PRINT PEEK(#33D)*10+PEEK(#33C),PEEK(#33B)*10+PEEK(#33A),PEEK(#339)*20
55      REM #3BD-#3BA IN THE TEST PROGRAM
60      GOTO 5
*


    63000 REM SCREEN COPY ON EPSON MX-80  (MODES 1,2,3,4)
    63002 POKE #131,1
    63005 INPUT " WELKE BACKGROUNDKLEUR ";BG:PRINT :POKE #131,0
    63007 PRINT CHR$(27);"E"
    63008 FOR Y=YMAX-3 TO 0 STEP -3
    63010 FOR X=0.0 TO XMAX-2 STEP 2
    63020 V=0
    63030 IF SCRN(X,Y+2)<>BG THEN V=V+1
    63040 IF SCRN(X+1,Y+2)<>BG THEN V=V+2
    63050 IF SCRN(X,Y+1)<>BG THEN V=V+4
    63060 IF SCRN(X+1,Y+1)<>BG THEN V=V+8
    63070 IF SCRN(X,Y)<>BG THEN V=V+16
    63080 IF SCRN(X+1,Y)<>BG THEN V=V+32
    63090 PRINT CHR$(160+V);:NEXT:PRINT :NEXT
    *
```

Am9511

Arithmetic Processor

DISTINCTIVE CHARACTERISTICS

- Fixed point 16 and 32 bit operations
- Floating point 32 bit operations
- Binary data formats
- Add, Subtract, Multiply and Divide
- Trigonometric and inverse trigonometric functions
- Square roots, logarithms, exponentiation
- Float to fixed and fixed to float conversions
- Stack-oriented operand storage
- DMA or programmed I/O data transfers
- End signal simplifies concurrent processing
- General purpose 8-bit data bus interface
- Standard 24 pin package
- +12 volt and +5 volt power supplies
- Advanced N-channel silicon gate MOS technology
- 100% MIL-STD-883 reliability assurance testing

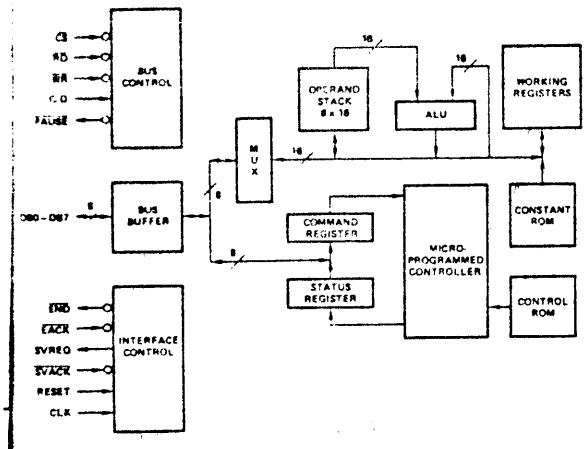
GENERAL DESCRIPTION

The Am9511 Arithmetic Processing Unit (APU) is a monolithic MOS/LSI device that provides high performance fixed and floating point arithmetic and a variety of floating point trigonometric and mathematical operations. It may be used to enhance the computational capability of a wide variety of processor-oriented systems.

All transfers, including operand, result, status and command information, take place over an 8-bit bidirectional data bus. Operands are pushed onto an internal stack and a command is issued to perform operations on the data in the stack. Results are then available to be retrieved from the stack, or additional commands may be entered.

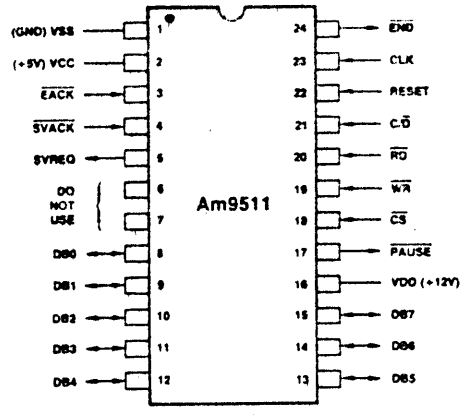
Transfers to and from the APU may be handled by the associated processor using conventional programmed I/O, or may be handled by a direct memory access controller for improved performance. Upon completion of each command, the APU issues an end of execution signal that may be used as an interrupt by the CPU to help coordinate program execution.

BLOCK DIAGRAM



MOS-046

CONNECTION DIAGRAM
Top View



Pin 1 is marked for orientation.

MOS-047

ORDERING INFORMATION

Package Type	Ambient Temperature	Clock Frequency		
		2MHz	3MHz	4MHz
Hermetic DIP	0°C ≤ T_A ≤ +70°C	Am9511DC	Am9511-1DC	Am9511-4DC
	−55°C ≤ T_A ≤ +125°C	Am9511DM	Am9511-1DM	

INTERFACE SIGNAL DESCRIPTION

VCC: +5 Volt power supply

VDD: +12 Volt power supply

VSS: Ground

CLK (Clock, Input)

An external timing source should be applied to the CLK pin. The Clock input may be asynchronous to the Read and Write control signals.

RESET (Reset, Input)

The active high Reset signal provides initialization for the chip. Reset terminates any operation in progress, clears the status register and places the Am9511 into the idle state. Stack contents are not affected by Reset. The Reset should be active for at least 5 clock periods following stable supply voltages and stable clock input. There is no internal power-on reset.

$\overline{CS}$ (Chip Select, Input)

$\overline{CS}$ is an active low input signal which conditions the read and write signals and thus enables communication with the data bus.

C/$\overline{D}$ (Command/Data, Input)

In conjunction with the $\overline{RD}$ and $\overline{WR}$ signals, the C/$\overline{D}$ control line establishes the type of transfers that are to be performed on the data bus.

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	Function
0	1	0	Enter data byte into stack
0	0	1	Read data byte from stack
1	1	0	Enter command
1	0	1	Read status

$\overline{RD}$ (Read, Input)

The active low Read signal is conditioned by $\overline{CS}$ and indicates that information is to be transferred from internal locations to the data bus. $\overline{RD}$ and $\overline{WR}$ are mutually exclusive.

$\overline{WR}$ (Write, Input)

The active low Write signal is conditioned by $\overline{CS}$ and indicates that information is to be transferred from the data bus into internal locations. $\overline{RD}$ and $\overline{WR}$ are mutually exclusive.

$\overline{EACK}$ (End Acknowledge, Input)

This active low input clears the end of execution output signal ($\overline{END}$). If $\overline{EACK}$ is tied low, the $\overline{END}$ output will be a pulse that is less than one clock period wide.

$\overline{SVACK}$ (Service Acknowledge, Input)

This active low input clears the service request output (SVREQ).

$\overline{END}$ (End Execution, Output)

This active low, open-drain output indicates that execution of the previously entered command is complete. It can be used as an interrupt request and is cleared by $\overline{EACK}$, RESET or any read or write access to the Am9511.

SVREQ (Service Request, Output)

This active high output signal indicates that command execution is complete and that post execution service was requested in the previous command byte. It is cleared by $\overline{SVACK}$, by RESET, or by the end of a subsequent command that does not request service.

$\overline{PAUSE}$ (Pause, Output)

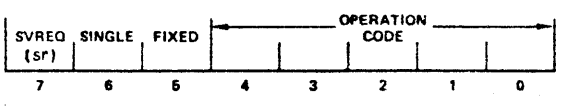
This active low output indicates that the Am9511 has not yet completed its information transfer with the host (or DMA) over the data bus. Whenever a data read or a status read operation is requested, $\overline{PAUSE}$ goes low. It returns high only after the data bus contains valid output data. When an existing command is still in the process of execution, and a data write, data read or command write is requested, then $\overline{PAUSE}$ goes low for the remaining duration of the existing command plus any time needed for initiating a data read. In both cases, the host should neither change any information to the Am9511, nor (in the case of data read or status read) attempt to capture data from the Am9511 DB outputs until $\overline{PAUSE}$ has returned high. (See "Pause Operation" section on page 5).

DB0-DB7 (Bidirectional Data Bus, I/O)

These eight bidirectional lines provide for transfer of commands, status and data between the Am9511 and the CPU. The Am9511 will drive the data bus only when $\overline{CS}$ and $\overline{RD}$ are low.

COMMAND STRUCTURE

Each command entered into the Am9511 consists of a single 8-bit byte having the format illustrated below:



Bits 0-4 select the operation to be performed as shown in the table. Bits 5-6 select the data format for the operation. If bit 5 is a 1, a fixed point data format is specified. If bit 5 is a 0, floating point format is specified. Bit 6 selects the precision of the data to be operated on by fixed point commands (if bit 5 = 0, bit 6 must be 0). If bit 6 is a 1, single-precision (16-bit) operands are indicated; if bit 6 is a 0, double-precision (32-bit) operands are indicated. Results are undefined for all illegal combinations of bits in the command byte. Bit 7 indicates whether a service request is to be issued after the command is executed. If bit 7 is a 1, the service request output (SVREQ) will go high at the conclusion of the command and will remain high until reset by a low level on the service acknowledge pin ($\overline{SVACK}$) or until completion of execution of a succeeding command where bit 7 is 0. Each command issued to the Am9511 requests post execution service based upon the state of bit 7 in the command byte. When bit 7 is a 0, SVREQ remains low.

COMMAND SUMMARY

Command Code								Command Mnemonic	Command Description
7	6	5	4	3	2	1	0		
FIXED POINT 16 BIT									
sr	1	1	0	1	1	0	0	SADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	1	1	0	1	1	0	1	SSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	1	1	0	1	1	1	0	SMUL	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.
sr	1	1	1	0	1	1	0	SMUU	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.
sr	1	1	0	1	1	1	1	SDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
FIXED POINT 32 BIT									
sr	0	1	0	1	1	0	0	DADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	0	1	0	1	1	0	1	DSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	0	1	0	1	1	1	0	DMUL	Multiply NOS by TOS. Lower half of result to NOS. Pop Stack.
sr	0	1	1	0	1	1	0	DMUU	Multiply NOS by TOS. Upper half of result to NOS. Pop Stack.
sr	0	1	0	1	1	1	1	DDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
FLOATING POINT 32 BIT									
sr	0	0	1	0	0	0	0	FADD	Add TOS to NOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	0	1	FSUB	Subtract TOS from NOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	1	0	FMUL	Multiply NOS by TOS. Result to NOS. Pop Stack.
sr	0	0	1	0	0	1	1	FDIV	Divide NOS by TOS. Result to NOS. Pop Stack.
DERIVED FLOATING POINT FUNCTIONS									
sr	0	0	0	0	0	0	1	SQRT	Square Root of TOS. Result in TOS.
sr	0	0	0	0	0	1	0	SIN	Sine of TOS. Result in TOS.
sr	0	0	0	0	0	1	1	COS	Cosine of TOS. Result in TOS.
sr	0	0	0	0	1	0	0	TAN	Tangent of TOS. Result in TOS.
sr	0	0	0	0	1	0	1	ASIN	Inverse Sine of TOS. Result in TOS.
sr	0	0	0	0	1	1	0	ACOS	Inverse Cosine of TOS. Result in TOS.
sr	0	0	0	0	1	1	1	ATAN	Inverse Tangent of TOS. Result in TOS.
sr	0	0	0	1	0	0	0	LOG	Common Logarithm (base 10) of TOS. Result in TOS.
sr	0	0	0	1	0	0	1	LN	Natural Logarithm (base e) of TOS. Result in TOS.
sr	0	0	0	1	0	1	0	EXP	Exponential (e^x) of TOS. Result in TOS.
sr	0	0	0	1	0	1	1	PWR	NOS raised to the power in TOS. Result in NOS. Pop Stack.
DATA MANIPULATION COMMANDS									
sr	0	0	0	0	0	0	0	NOP	No Operation
sr	0	0	1	1	1	1	1	FIXS	Convert TOS from floating point to 16-bit fixed point format.
sr	0	0	1	1	1	1	0	FIXD	Convert TOS from floating point to 32-bit fixed point format.
sr	0	0	1	1	1	0	1	FLTS	Convert TOS from 16-bit fixed point to floating point format.
sr	0	0	1	1	1	0	0	FLTD	Convert TOS from 32-bit fixed point to floating point format.
sr	1	1	1	0	1	0	0	CHSS	Change sign of 16-bit fixed point operand on TOS.
sr	0	1	1	0	1	0	0	CHSD	Change sign of 32-bit fixed point operand on TOS.
sr	0	0	1	0	1	0	1	CHSF	Change sign of floating point operand on TOS.
sr	1	1	1	0	1	1	1	PTOS	Push 16-bit fixed point operand on TOS to NOS (Copy)
sr	0	1	1	0	1	1	1	PTOD	Push 32-bit fixed point operand on TOS to NOS. (Copy)
sr	0	0	1	0	1	1	1	PTOF	Push floating point operand on TOS to NOS. (Copy)
sr	1	1	1	1	0	0	0	POPS	Pop 16-bit fixed point operand from TOS. NOS becomes TOS.
sr	0	1	1	1	0	0	0	POPD	Pop 32-bit fixed point operand from TOS. NOS becomes TOS.
sr	0	0	1	1	0	0	0	POPF	Pop floating point operand from TOS. NOS becomes TOS.
sr	1	1	1	1	0	0	1	XCHS	Exchange 16-bit fixed point operands TOS and NOS.
sr	0	1	1	1	0	0	1	XCHD	Exchange 32-bit fixed point operands TOS and NOS.
sr	0	0	1	1	0	0	1	XCHF	Exchange floating point operands TOS and NOS.
sr	0	0	1	1	0	1	0	PUPI	Push floating point constant "π" onto TOS. Previous TOS becomes NOS.

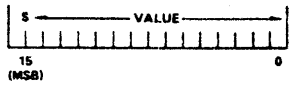
NOTES:

1. TOS means Top of Stack. NOS means Next on Stack.
2. AMD Application Brief "Algorithm Details for the Am9511 APU" provides detailed descriptions of each command function, including data ranges, accuracies, stack configurations, etc.
3. Many commands destroy one stack location (bottom of stack) during development of the result. The derived functions may destroy several stack locations. See Application Brief for details.
4. The trigonometric functions handle angles in radians, not degrees.
5. No remainder is available for the fixed-point divide functions.
6. Results will be undefined for any combination of command coding bits not specified in this table.

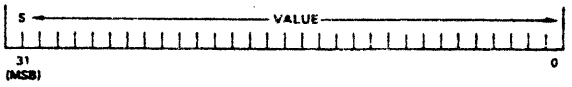
DATA FORMATS

The Am9511 Arithmetic Processing Unit handles operands in both fixed point and floating point formats. Fixed point operands may be represented in either single (16-bit operands) or double precision (32-bit operands), and are always represented as binary, two's complement values.

16-BIT FIXED POINT FORMAT



32-BIT FIXED POINT FORMAT



The sign (positive or negative) of the operand is located in the most significant bit (MSB). Positive values are represented by a sign bit of zero (S = 0). Negative values are represented by the two's complement of the corresponding positive value with a sign bit equal to 1 (S = 1). The range of values that may be accommodated by each of these formats is −32,768 to +32,767 for single precision and −2,147,483,648 to +2,147,483,647 for double precision.

Floating point binary values are represented in a format that permits arithmetic to be performed in a fashion analogous to operations with decimal values expressed in scientific notation.

$$(5.83 \times 10^{2})(8.16 \times 10^{1}) = (4.75728 \times 10^{4})$$

In the decimal system, data may be expressed as values between 0 and 10 times 10 raised to a power that effectively shifts the implied decimal point right or left the number of places necessary to express the result in conventional form (e.g., 47,572.8). The value-portion of the data is called the mantissa. The exponent may be either negative or positive.

The concept of floating point notation has both a gain and a loss associated with it. The gain is the ability to represent the significant digits of data with values spanning a large dynamic range limited only by the capacity of the exponent field. For example, in decimal notation if the exponent field is two digits wide, and the mantissa is five digits, a range of values (positive or negative) from 1.0000×10^{-99} to $9.9999 \times 10^{+99}$ can be accommodated. The loss is that only the significant digits of the value can be represented. Thus there is no distinction in this representation between the values 123451 and 123452, for example, since each would be expressed as: 1.2345×10^{5}. The sixth digit has been discarded. In most applications where the dynamic range of values to be represented is large, the loss of significance, and hence accuracy of results, is a minor consideration. For greater precision a fixed point format could be chosen, although with a loss of potential dynamic range.

The Am9511 is a binary arithmetic processor and requires that floating point data be represented by a fractional mantissa value between .5 and 1 multiplied by 2 raised to an appropriate power. This is expressed as follows:

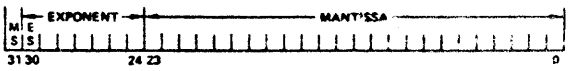
$$\text{value} = \text{mantissa} \times 2^{\text{exponent}}$$

For example, the value 100.5 expressed in this form is 0.11001001×2^{7}. The decimal equivalent of this value may be computed by summing the components (powers of two) of the mantissa and then multiplying by the exponent as shown below:

$$\begin{aligned}\text{value} &= (2^{-1} + 2^{-2} + 2^{-5} + 2^{-8}) \times 2^{7} \\ &= (0.5 + 0.25 + 0.03125 + 0.00290625) \times 128 \\ &= 0.78515625 \times 128 \\ &= 100.5\end{aligned}$$

FLOATING POINT FORMAT

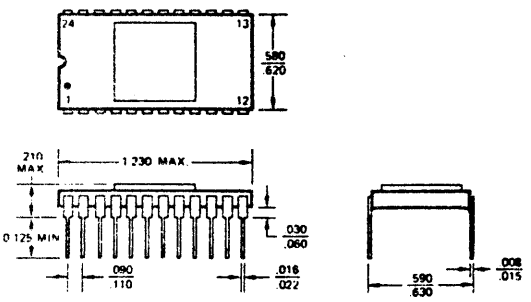
The format for floating point values in the Am9511 is given below. The mantissa is expressed as a 24-bit (fractional) value; the exponent is expressed as an unbiased two's complement 7-bit value having a range of −64 to +63. The most significant bit is the sign of the mantissa (0 = positive, 1 = negative), for a total of 32 bits. The binary point is assumed to be to the left of the most significant mantissa bit (bit 23). All floating point data values must be normalized. Bit 23 must be equal to 1, except for the value zero, which is represented by all zeros.



The range of values that can be represented in this format is $\pm(2.7 \times 10^{-20}$ to $9.2 \times 10^{18})$ and zero.

PHYSICAL DIMENSIONS
Dual-In-Line

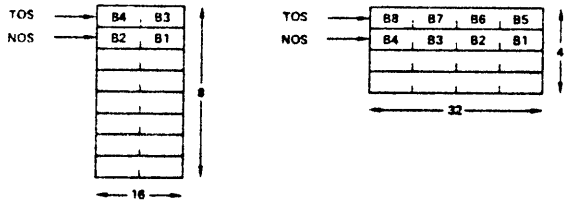
24-Pin Side-Brazed



FUNCTIONAL DESCRIPTION

Stack Control

The user interface to the Am9511 includes access to an 8 level 16-bit wide data stack. Since single precision fixed point operands are 16 bits in length, eight such values may be maintained in the stack. When using double precision fixed point or floating point formats four values may be stored. The stack in these two configurations can be visualized as shown below:



Data are written onto the stack, eight bits at a time, in the order shown (B1, B2, B3, . . .). Data are removed from the stack in reverse byte order (B8, B7, B6. . . .). Data should be transferred into or out of the stack in multiples of the number of bytes appropriate to the chosen data format.

Data Entry

Data entry is accomplished by bringing the chip select ($\overline{CS}$), the command/data line (C/$\overline{D}$), and $\overline{WR}$ low, as shown in the timing diagram. The entry of each new data word "pushes down" the previously entered data and places the new byte on the top of stack (TOS). Data on the bottom of the stack prior to a stack entry are lost.

Data Removal

Data are removed from the stack in the Am9511 by bringing chip select ($\overline{CS}$), command/data (C/$\overline{D}$), and $\overline{RD}$ low as shown in the timing diagram. The removal of each data word redefines TOS so that the next successive byte to be removed becomes TOS. Data removed from the stack rotates to the bottom of the stack.

Command Entry

After the appropriate number of bytes of data have been entered onto the stack, a command may be issued to perform an operation on that data. Commands which require two operands for execution (e.g., add) operate on the TOS and NOS values. Single operand commands operate only on the TOS.

Commands are issued to the Am9511 by bringing the chip select ($\overline{CS}$) line low, command/data (C/$\overline{D}$) line high, and $\overline{WR}$ line low as indicated by the timing diagram. After a command is issued, the CPU can continue execution of its program concurrently with the Am9511 command execution.

Command Completion

The Am9511 signals the completion of each command execution by lowering the End Execution line ($\overline{END}$). Simultaneously, the busy bit in the status register is cleared and the Service Request bit of the command register is checked. If it is a "1" the service request output level (SVREQ) is raised. $\overline{END}$ is cleared on receipt of an active low End Acknowledge ($\overline{EACK}$) pulse. Similarly, the service request line is cleared by recognition of an active low Service Acknowledge ($\overline{SVACK}$) pulse.

Pause Operation

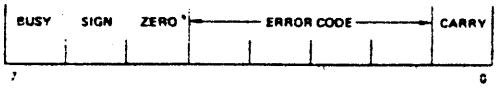
An active low Pause ($\overline{PAUSE}$) is provided. This line is high in its quiescent state and is pulled low by the Am9511 under the following conditions:

1. A previously initiated operation is in progress (device busy) and Command Entry has been attempted. In this case, the $\overline{PAUSE}$ line will be pulled low and remain low until completion of the current command execution. It will then go high, permitting entry of the new command.
2. A previously initiated operation is in progress and stack access has been attempted. In this case, the $\overline{PAUSE}$ line will be pulled low, will remain in that state until execution is complete, and will then be raised to permit completion of the stack access.
3. The Am9511 is not busy, and data removal has been requested. $\overline{PAUSE}$ will be pulled low for the length of time necessary to transfer the byte from the top of stack to the interface latch, and will then go high, indicating availability of the data.
4. The Am9511 is not busy, and a data entry has been requested. $\overline{PAUSE}$ will be pulled low for the length of time required to ascertain if the preceding data byte, if any has been written to the stack. If so $\overline{PAUSE}$ will immediately go high. If not, $\overline{PAUSE}$ will remain low until the interface latch is free and will then go high.
5. When a status read has been requested, $\overline{PAUSE}$ will be pulled low for the length of time necessary to transfer the status to the interface latch, and will then be raised to permit completion of the status read. Status may be read whether or not the Am9511 is busy.

When $\overline{PAUSE}$ goes low, the APU expects the bus and bus control signals present at the time to remain stable until $\overline{PAUSE}$ goes high.

Device Status

Device status is provided by means of an internal status register whose format is shown below:



BUSY: Indicates that Am9511 is currently executing a command (1 = Busy).

SIGN: Indicates that the value on the top of stack is negative (1 = Negative).

ZERO: Indicates that the value on the top of stack is zero (1 = Value is zero).

ERROR CODE: This field contains an indication of the validity of the result of the last operation. The error codes are:

0000 – No error
1000 – Divide by zero
0100 – Square root or log of negative number
1100 – Argument of inverse sine, cosine, or e^x too large
XX10 – Underflow
XX01 – Overflow

CARRY: Previous operation resulted in carry or borrow from most significant bit. (1 = Carry/Borrow, 0 = No Carry/No Borrow)

If the BUSY bit in the status register is a one, the other status bits are not defined; if zero, indicating not busy, the operation is complete and the other status bits are defined as given above.

Read Status

The Am9511 status register can be read by the CPU at any time (whether an operation is in progress or not) by bringing the chip select ($\overline{CS}$) low, the command-data line (C/$\overline{D}$) high, and lowering $\overline{RD}$. The status register is then gated onto the data bus and may be input by the CPU.

EXECUTION TIMES

Timing for execution of the Am9511 command set is shown in the table below. Speeds are given in terms of clock cycles and should be multiplied by the clock period being used to arrive at time values. Where substantial variation of execution times is possible, the minimum and maximum values are shown; otherwise, typical values are given. Variations are data dependent. Some boundary conditions that will cause shorter execution times are not taken into account. The listing is in alphabetical order by mnemonic.

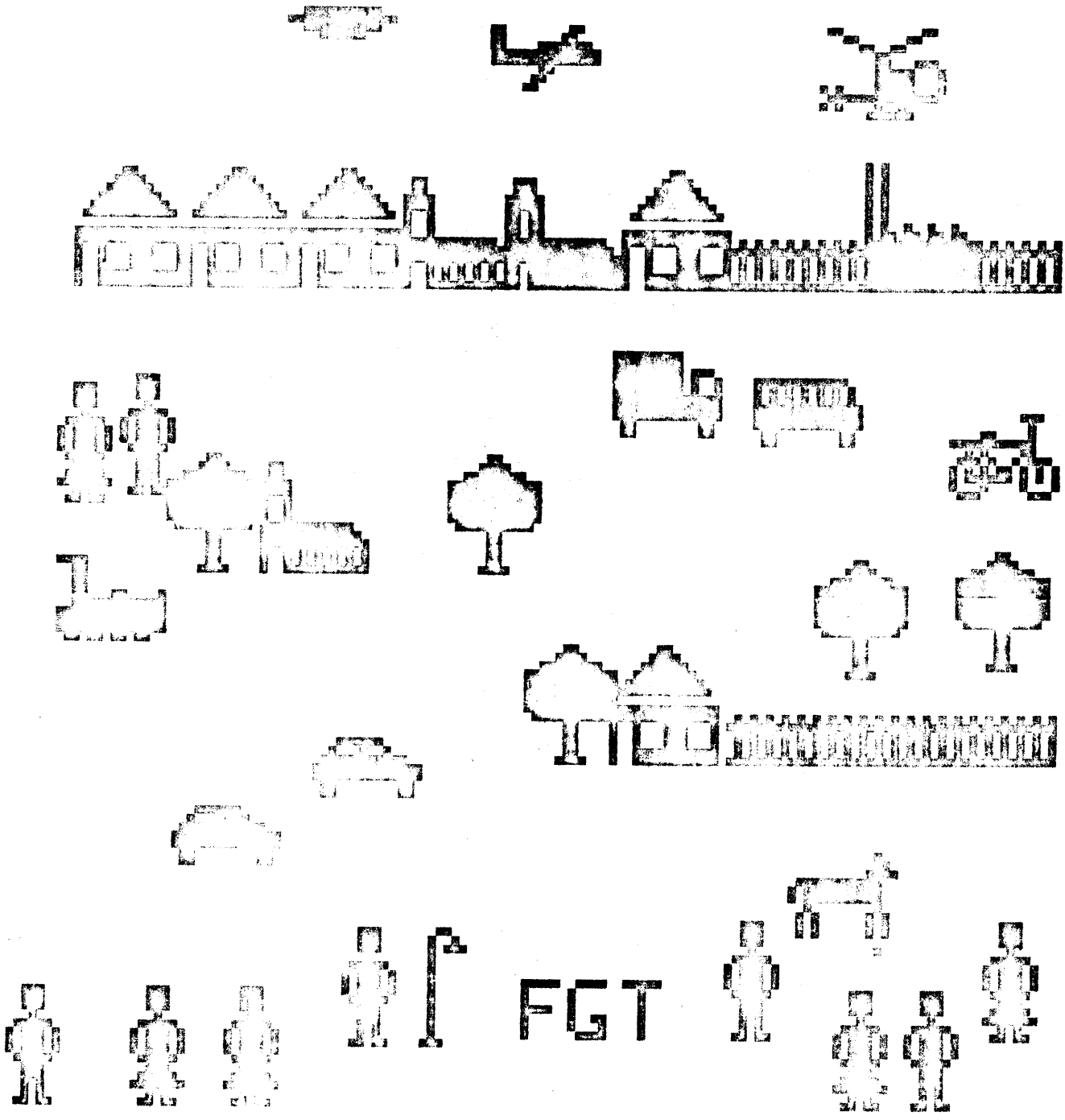
Total execution times may require allowances for operand transfer into the APU, command execution, and result retrieval from the APU. Except for command execution, these times will be heavily influenced by the nature of the data, the control interface used, the speed of memory, the CPU used, the priority allotted to DMA and Interrupt operations, the size and number of operands to be transferred, and the use of chained calculations, etc.

COMMAND EXECUTION TIMES

Command Mnemonic	Clock Cycles	Command Mnemonic	Clock Cycles
ACOS	6304-8284	LOG	4474-7132
ASIN	6230-7938	LN	4298-6956
ATAN	4992-6536	NOP	4
CHSD	26-28	POPD	12
CHSF	16-20	POPF	12
CHSS	22-24	POPS	10
COS	3840-4878	PTOD	20
DADD	20-22	PTOF	20
DDIV	196-210	PTOS	16
DMUL	194-210	PUPI	16
DMUU	182-218	PWR	8294-12034
DSUB	38-40	SADD	16-18
EXP	3794-4878	SDIV	84-94
FADD	54-368	SIN	3796-4808
FDIV	154-184	SMUL	84-94
FIXD	90-336	SMUU	80-98
FIXS	90-214	SQRT	782-870
FLTD	56-342	SSUB	30-32
FLTS	52-156	TAN	4894-5886
FMUL	146-168	XCHD	26
FSUB	70-370	XCHF	26
		XCHS	18

As mentioned, the above clock cycle execution times can be converted to μsec by multiplying by the clock period used. Several examples (minimums) are shown below:

Command Description	Am9511 (2MHz)	Am9511-1 (3MHz)	Am9511-4 (4MHz)
32-Bit Floating-Point Cosine (COS)	1920μsec	1280μsec	960μsec
32-Bit Floating-Point e^x (EXP)	1897μsec	1265μsec	949μsec
32-Bit Floating Point Multiply (FMUL)	73μsec	49μsec	37μsec
16-Bit Fixed-Point Multiply, Lower (SMUL)	42μsec	28μsec	21μsec
32 Bit Floating-Point Add (FADD)	27μsec	18μsec	14μsec
16-Bit Fixed-Point Add (SADD)	8μsec	5μsec	4μsec


FGT

PADDELEN MET FGT

T-1000

```
100  MODE 0:PRINT CHR$(12)
105   PRINT "PADDELEN MET FGT ...":FOR X=1 TO 59:PRINT CHR$(255);:NEXT
107   PRINT :PRINT
110   LIST 280-380
120   G=GETC:IF G=0 THEN 120
130   GOTO 500
280   REM "HUIS    a   VLIEGTUIG  l"
290   REM "KERK    b   HELIKOPTER m"
300   REM "HEK     c   U.F.O      n"
310   REM "BOOM    d   VOGEL      o"
320   REM "MAN     e   FIETS      p"
330   REM "VROUW   f   HOND       q"
340   REM "AUTO    g   LADDER     r"
350   REM "TREIN   h   VIS        s"
360   REM "CAMION  i   LANTAARN   t"
370   REM "AUTOBUS j   POPULIER   u"
380   REM "FABRIEK k   HALVE MAAN v"
400   REM ***SUBROUTINE FGT
410   SOUND 1 0 15 0 FREQ(440):POKE #2F2,X MOD 256:POKE #2F3,X/256:POKE #2F4,Y
420   POKE #2F5,SP:POKE #2F6,ID
430   C=FC*#40+CC:F=FF*#80+PF*#40+ZF*#20+VF*#10+DF:POKE #2F0,C:POKE #2F1,F
440   CALLM #300,A$:SOUND OFF
450   RETURN
500   MODE 4:COLOR6 0 15 15 15
505   X=10:Y=10:CC=15:PF=0:DF=0:VF=0:ZF=0:G=ASC("A")
510   A=PDL(1):B=PDL(2)
520   C=SCRN(A,B):DOT A,B 15:WAIT TIME 3:DOT A,B C
525   REM kies nieuwe tekening met keyboard
530   G=GETC:IF G<>0 THEN IF G<20 THEN 550:VF=0:ZF=0:A$=CHR$(G):FILL 0,0 15,15 0:X=0:Y=0:GOSUB 400
532   REM plaats nieuwe tekening met event
535   IF PEEK(#FD00) IAND 48<>0 THEN X=A:Y=B:GOSUB 400
540   GOTO 510
550   REM aanpassing van ZF en VF met CURSOR^toetsen
560   IF G=16 THEN VF=0
570   IF G=17 THEN VF=1
580   IF G=18 THEN ZF=1
590   IF G=19 THEN ZF=0
595   IF G=9 THEN FILL A,B A+15,B+15 0:REM wis met TAB
600   GOTO 510
1000  REM Bij OFF SCREEN : MODE 4: RUN 510
```

EXACTE TIMER +++ TIME OUT +++ ANTWOORDTIJD +++ REACTIEMETING

Adressen é1BE (LOW) en é1BF(HIGH) bevatten een timerteller.
Deze constructie (eerst LOW BYTE dan HIGH BYTE) is typisch voor 8080 microprocessor.We kunnen duidelijk de splitsing HIGH/LOW vaststellen als we van een getal de HEX& vragen.

```
vb:     PRINT HEX&(65535) :éFFFF    HB= FF    LB=FF
        PRINT HEX&(255)   :éFF      HB= Ø     LB=FF
        PRINT HEX&(1000)  :é3E8     HB= 3     LB=E8
```

De benadering via HEX& is echter niet de eenvoudigste manier om de beide delen van het getal weg te POKEN.
De LOW BYTE kunnen we ook bekomen door volgende constructies:

```
GETAL MOD é100
GETAL MOD 256
GETAL IAND 255
```

De HIGH BYTE komt te voorschijn op de volgende manieren:

```
GETAL SHR 8
INT(GETAL/256)
```

Op interruptbasis wordt elke 20 milliseconden deze timer met 1 verminderd, totdat hij de waarde Ø bereikt.(é1BE en é1BF beiden Ø)
Deze Ø-inhoud blijft behouden totdat een nieuwe waarde in de teller wordt geplaatst.
Doel van deze teller is de generatie van tijdseenheden voor de WAIT TIME instructie van BASIC.
In het programma wordt de waarde eerst vermenigvuldigd met 50 zodat we de vertraging kunnen opgeven in seconden.
Deze waarde wordt dan verdeeld over de twee teller-bytes volgens bovenstaande beschrijving.
Het programma gaat nu in een loop totdat de beide adressen Ø bevatten.
De uitvoeringstijden van het programma zijn niet belangrijk omdat de timing constant gebeurt door de interrupt-werking.
OPMERKING: Tijdens een actieve cassetteroutine (dus als de cursor niet meer flikkert) worden alle interrupts genegeerd en gaat het decrementeren van de teller niet verder.
Meteen is ook duidelijk waarom de maximum WAIT TIME waarde 65535 is: dit is de maximale waarde die we kwijt kunnen over 2 bytes.
Bijgaande programma's maken duidelijk hoe we deze timer-techniek in BASIC-programma's kunnen gebruiken.

```
5     REM TIME OUT BEWAKING MET INTERRUPT-TELLER
10    TIMEMAX=5:REM IN SEK
20    T=TIMEMAX*50:POKE #1BE,T MOD #100:POKE #1BF,T SHR 8:REM T IN x20 MS
30    G=GETC
40    IF PEEK(#1BE)=0 AND PEEK(#1BF)=0 THEN 100:REM TEST
50    IF G=#D GOTO 80:REM car ret,nieuwe string,restart
60    IF G<>0 THEN PRINT CHR$(G);
70    GOTO 30
80    PRINT :GOTO 10
100   SOUND 1 0 15 0 FREQ(1000):WAIT TIME 5:SOUND OFF :PRINT "  TIME OUT"
110   GOTO 10
```

```
5     REM BEPALING VAN RESPONSTIJD
10    T1=#FFFF:REM MAXIMUMTIJD IN EENHEDEN VAN 20 MS (21 MIN)
20    POKE #1BE,#FF:POKE #1BF,#FF
30    INPUT A$
40    T2=PEEK(#1BE) IOR (PEEK(#1BF) SHL 8):REM NIEUWE STAND
50    IF T2=0 THEN PRINT " OUT OF TIME ....":GOTO 20
60    PRINT " U GEBRUIKTE ";(T1-T2)/50.0;" SEC VOOR HET ANTWOORD"
70    GOTO 10
```

```
10    REM REAL TIME CLOCK (maximaal [ 65535/50 ] seconden = 21 minuten)
20    POKE #1BE,#FF:POKE #1BF,#FF:REM start clock.
30    REM De verlopen tijd sinds start clock is steeds :
40    PRINT (#FFFF-(PEEK(#1BF)*256)-PEEK(#1BE))/50
50    GOTO 40
60    REM Na 21 minuten is de tijd niet meer significant !!!
70    REM U kan steeds opnieuw starten door lijn 20 uit te voeren.
80    REM GEEN WAIT TIME uitvoeren ondertussen !!!
90    REM U kan WAIT TIME simuleren door een dummy lus:
100   REM vb: FOR X=1 TO 10000:NEXT
```

READ

N 2235

SPECIFICATIES

- Mono cassetterecorder die op batterijen en op netstroom werkt, waardoor het mogelijk is hem onder alle omstandigheden te gebruiken.
- Toetsbedieningen voor de verschillende mogelijkheden : pauze, stop, weergave, spoelen, terugspoelen en opnemen.
- Automatische « FULL AUTO SHUT OFF », met stroomonderbreking en uitschakeling van de bedieningen op het einde van de cassette.
- Een « REVIEW-CUE » systeem vergemakkelijkt het zoeken of het snel beluisteren van een opname (in de « start » stand) d.m.v. de « spoel » of « terugspoel »-toetsen.
- De opnamen kunnen gemakkelijk teruggevonden worden dank zij een driecijferige teller met nulstelling.
- Door het gebruik van een extra microfoon is MIXING tijdens de opname mogelijk.
- Het toestel kan dienst doen als een vaste microversterker d.m.v. een extra microfoon (PUBLIC ADDRESS).
- Elke opname zal ongetwijfeld geslaagd zijn dank zij de automatische controle van het opnameniveau.
- Aanduiding van de opnamestand d.m.v. LED diode.
- Dank zij een beveiligde POST FADING knop kan men tijdens de weergave de ongewenste gedeelten uitwissen.
- De elektronisch gestabliseerde motorsnelheid verzekert een jengelvrije beluistering en het apparaat kan hierdoor zowel door een netstroom van 50 als van 60 Hz gevoed worden.
- Een PITCH knop voor de snelheidsregeling maakt het de gebruiker mogelijk het bandverloop ± 2 % te doen schommelen (erg gewaardeerd voor audiovisuele montages).
- Beveiligingssysteem waarmee men het toevallig uitwissen van vooraf opgenomen cassettes voorkomt.
- De opname/weergavekop van het « long-life »- type verzekert een duurzame kwaliteit van de opnamen en van het aflezen.
- Dank zij de toonregeling kan de geluidsweergave aan eenieders smaak aangepast worden.
- Een indicator met elektroluminescente LED-diode gaat branden wanneer de batterijen vervangen dienen te worden.
- Een inklapbaar handvat maakt een gemakkelijke verplaatsing mogelijk.

TECHNISCHE KENMERKEN

- BATTERIJVOEDING
  9 V - 6 x 1,5 V - R 14
- NETVOEDING
  220 - 240 V 50/60 Hz
  110 - 127 V (Service oplossing)
- CASSETTESYSTEEM
  PHILIPS Compact cassetten (2 sporen)
- BANDVERLOOP
  4,76 cm/sec.
- SIGNAAL/RUISVERHOUDING
  beter dan 45 dB
- JENGEL, ZWEVEN
  minder dan 0,35 %
- UITGANGSVERMOGEN
  (± 1 dB; D = 10 %)
  1.500 mW sinus
  3.000 mW music
- LUIDSPREKER
  Ø 10 cm, 4 ohm
- AANDRIJFMOTOR
  gelijkstroom
- AANSLUITINGEN
  microfooningang - afstandsbediening START/STOP
  Lijningang en lijnuitgang
  extra luidspreker
- HALFGELEIDERS
  2 IC, 9 transistoren 4 dioden
- HUIS
  polystyreen
- AFMETINGEN
  L x H x D 200 x 67 x 290 mm

Onder voorbehoud van wijzigingen.

Code : 918.332

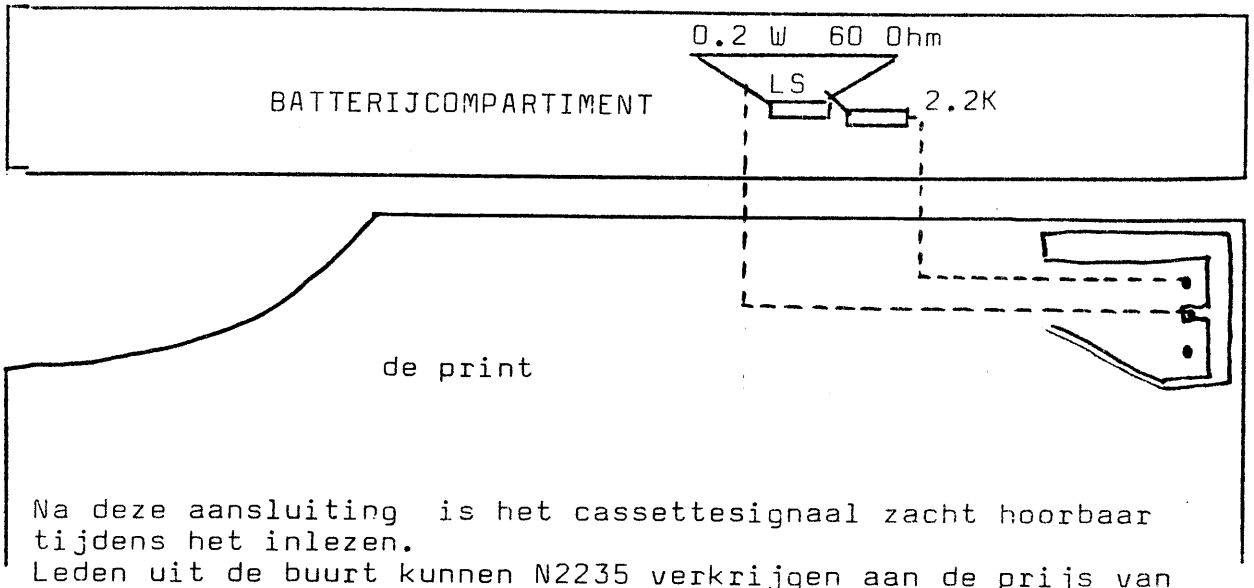
PHILIPS

Naamloze Vennootschap Philips
de Brouckèreplein, 2, Bus 1 - 1000 BRUSSEL
Tel. 02/219.18.00 - 219.30.00
H.R. Brussel 2488

PHILIPS

CASSETTE RECORDER

In onze eerste publicaties hadden wij PHILIPS N2219 voorgesteld als cassetterecorder voor onze DAIpc.Dit toestel werkt heel betrouwbaar en is bij vele leden in gebruik.
Nu heeft PHILIPS de productie van dit type beeindigd en moeten wij dus uitkijken naar een ander toestel.
Bij het doornemen van de informatiebladen blijkt dat N2235 als goede tweede uit de bus komt.
De fabrikant beweert dat N2235 haast electronisch identiek is met N2219, dus alle reden om voortaan N2235 te adviseren.
Proeven hebben uitgewezen dat alle opnames van N2219 zonder problemen met N2235 in te lezen zijn.
N2235 biedt zelfs nog een aantal voordelen en is voordeliger in prijs.In Belgie +/- 3300 Bfr.
Voordelen:FULL AUTO SHUT OFF,REVIEW-CUE:erg gemakkelijk om een programma op te zoeken,LONG LIVE koppen,PITCH-regeling(de snelheid bijregelen).
Wij vinden het prettig om tijdens het inlezen het signaal te kunnen beluisteren(normaal is dit volledig uitgeschakeld door de LSplug).
Hieronder vindt U een erg eenvoudige methode om dit bij N2235 te realiseren.U heeft nodig:1 miniluidspreker met diameter kleiner dan 5.5 cm,2 eindjes draad van +/- 15 cm en 1 weerstand van 2.2K .
We solderen de eindjes draad aan de LS en aan de aangeduide soldeerpunten en stoppen de luidspreker in het batterijcompartiment, eenvoudiger kan het niet.U maakt de vier schroeven oneraan los,neemt het deksel eraf en krijgt volgend panorama:



Na deze aansluiting is het cassettesignaal zacht hoorbaar tijdens het inlezen.
Leden uit de buurt kunnen N2235 verkrijgen aan de prijs van 2950 Bfr bij TV HANDEL TUBBAX Aarschotsebaan 7A 3140 RAMSEL tel: 016/698111.
Verdere informatie over N2235 vindt U op bijgaand infoblad.

De DAI cassette-interface is heel soepel wat betreft snelheidsvariaties: als de leader voorbij is kan je op N2235 de PITCH knop helemaal rechtsom draaien,wat ongeveer 50% tijdwinst betekent bij het inlezen. Het lukt zelfs bij FAST FORWIND met CUEING,maar dit vereist wel een zekere vingervaardigheid...

Mini-Digital Cassette Recorder
An alternative to disc for program & data storage

PHILIPS DIGITAL CASSETTE RECORDER INTERFACE

Begin juni 81 zal de DCR interface voor DAI klaar zijn.
We kunnen U al het volgende hierover vertellen:
De interface is ontworpen door DAInamicer H.WEGMAN.
De verdeling zal gebeuren door MEMOCOM BV tel 010-148284.(NEDERLAND)
Voorzichtige richtprijs:600 Gld ex BTW.
De DCR zal opereren op 6000 Baud, dus 10X cassettesnelheid.
De besturing gebeurt volledig vanaf het toetsenbord.
Het TOS(tape operating system)komt in EPROM op de X-bus binnenin de DAIpc en heeft geen adressen nodig van DAI-system.
Buiten de klassieke commando's:SAVE,LOAD,SAVEA,LOADA zijn nog voorhanden:REWIND,SKIP(eventueel gevolgd door nr) DELETE,CHECK, CAS-DCR,DCR-CAS en als extraatje herbergt TAB nu CHR$(12).
De besturing kan ook gebeuren vanuit BASIC-programma's zodat een DATAbestand op DCR erg realistisch wordt.
De decoder-buffer print biedt de mogelijkheid om 4 DCR's aan te sluiten,240K ON LINE,klinkt helemaal niet gek.
De prijs van het geheel zal waarschijnlijk oa afhankelijk zijn van het aantal gegadigden.Indien U interesse heeft, neem alvast contact met MEMOCOM.
Deze informatie bereikte ons telefonisch, we komen hier zeker nog uitgebreid op terug.

CATALOG

TOOLKIT COLLECTION 1 (source files incl) 84 Gld 1250 Bfr

RENUMBER OBJ 2F0 3FF
RENUMBER BASIC pr
NEW FORMAT LISTING 2EC 66F
DOCUMENTATION NFL
DATA STATEMENTS GENERATOR 300 86F
BASIC EXTENSIONS: C1E DFA (above FGT)
- LABEL GOTO
- LABEL GOSUB
- LABEL READ DATA
- LABEL CASE
- CHEKSUM
- BIT TEST

More COLLECTION tapes will be announced in NEWSLETTER 6.

PIN-OUT OF THE 50-PENS CONNECTOR INSIDE DAIpc

1	GROUND	SCREENING LINE
2	D0	DATA BIT 0
3	GROUND	SCREENING LINE
4	D1	DATA BIT 1
5	GROUND	SCREENING LINE
6	D2	DATA BIT 2
7	GROUND	SCREENING LINE
8	D3	DATA BIT 3
9	GROUND	SCREENING LINE
10	D4	DATA BIT 4
11	GROUND	SCREENING LINE
12	D5	DATA BIT 5
13	GROUND	SCREENING LINE
14	D6	DATA BIT 6
15	GROUND	SCREENING LINE
16	D7	DATA BIT 7
17	GROUND	SCREENING LINE
18	-	NO CONNECTION
19	GROUND	SCREENING LINE
20	MEMW	MEMORY WRITE STROBE
21	GROUND	SCREENING LINE
22	-	NO CONNECTION
23	A10	ADDRESS LINE 10
24	MEMR	MEMORY READ STROBE
25	A14	ADDRESS LINE 14
26	A11	11
27	A12	12
28	A13	13
29	A9	9
30	A15	15
31	A7	7
32	A8	8
33	A5	5
34	A6	6
35	A3	3
36	A4	4
37	A1	1
38	A2	2
39	A0	0
40	INTA	INTERRUPT ACKNOWLEDGE
41	CS LOW ROM	CHIP SELECT LOWER ROM
42	CS LB UPP ROM	CHIP SELECT LOWER BANK UPPER ROM
43	PSEUDE A12	A12 AFTER START-LOGIC
44	CS UB UPP ROM	CHIP SELECT UPPER BANK UPPER ROM
45	+5V	5 VOLT
46	+5V	5 VOLT
47	RAMOP	NOT RAM OPERATION
48	CK2	TTL LEVEL CLOCK (2MHz)
49	HOLD	HOLD REQUEST
50	SYNC	CPU SYNC SIGNAL

DAS GESETZ DER GROSSEN ZAHL A. MEYSTRE 3/81

Mit den Zufallszahlengeneratoren R = RND(0) und R = RND(1) laesst sich durch die Abfrage IF R> 0.5 der Muenzwurf (Kopf oder Zahl) simulieren. Bei einer homogenen Muenze (Schwerpunkt in der Mitte) wird beim mehrmaligen Werfen Kopf und Zahl gleich oft vorkommen, auch wenn es am Anfang anders scheint. Die Wahrscheinlichkeit von Kopf (oder von Zahl) ist p = 0.5.

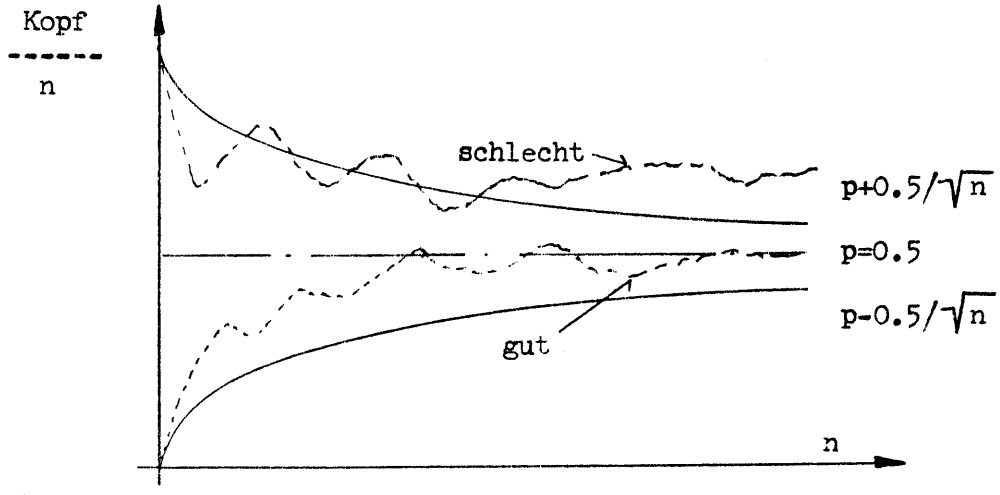
Das Gesetz der grossen Zahl besagt, dass wenn die Muenze unendlich oft geworfen wird, folgender Wert entsteht.

$$p = \frac{\text{Anzahl Kopf}}{\text{Anzahl Wuerfe}} = 0.5$$

Das vorliegende Programm simuliert ein Muenzwurfexperiment mit 320 Wuerfe und zeichnet laufend das erhaltene Ergebnis. Die Streuungsgrenzen um den erwarteten Wert sind als rote Flaeche dargestellt. Diese Grenzfunktion ist gegeben mit

$$S = p \pm \frac{\sqrt{p\,(1-p)}}{\sqrt{n}} \qquad \text{fuer } p=0.5 \qquad S = 0.5 \pm \frac{0.5}{\sqrt{n}}$$

wobei n = Anzahl Wuerfe
p = Wahrscheinlichkeit Kopf zu erhalten



Pendelt sich die Wahrscheinlichkeitskurve nicht innerhalb der Grenzkurve ein, so bedeutet dies, dass Kopf nicht gleich oft vorkommt wie Zahl, oder auf den Zufallsgenerator des DAI bezogen, innerhalb des Bereichs 0 bis 1 kommen Werte unterhalb 0.5 nicht gleich oft vor wie Werte oberhalb 0.5. Mit anderen Worten der Generator ist nicht echt. Bei meinem DAI trifft dies leider fuer RND(0) zu. RND(1) hingegen liefert ein gutes Resultat. Der gewuenschte Generator wird via Zeile 510 definiert.

Das stoerende Flackern des Cursors laesst sich entfernen mittels
POKE #75,#20

```
10    REM BEISPIEL AUS DER STATISTIEK      MEYSTRE 3/81
20    XOFF=7.0:YMIT=107.0:YOFF=7.0
30    ROT=3.0:GRUEN=5.0:GELB=14.0:COLORG GRUEN ROT GELB 0
40    MODE 6A:MODE 6A
50    PRINT CHR$(12):POKE #75,32:REM CURSOR BLANCO
60    PRINT "ZUFALLSEXPERIMENT MIT TREFFERWAHRSCHEINLICHKEIT p=0.5"
70    PRINT "        ANZALH PROBEN   n                              "
80    PRINT "        RELATIVE HAEUFIGKEIT r/n                       "
90    R=0.0:N=1.0:NSIG=1.0
100   FOR I=1.0 TO 10.0:GOSUB 400:NEXT
110   FOR I=1.0 TO 310.0:GOSUB 400:GOSUB 500:NEXT
120   FOR I=1.0 TO 10.0:GOSUB 500:NEXT
130   GOTO 130
400   REM GRENZKURVE
410   DRAW XOFF+NSIG,YMIT-100/SQR(NSIG) XOFF+NSIG,YMIT+100/SQR(NSIG) ROT
420   NSIG=NSIG+1.0:RETURN
500   REM MOMENTANWERT DE HAEUFIGKEIT
510   IF RND(1.0)>0.5 THEN R=R+200.0
520   DOT XOFF+N,YOFF+R/N GELB
530   CURSOR 35,2:PRINT N
540   CURSOR 35,1:PRINT (R/200.0)/N
550   N=N+1.0:RETURN
```

```
5     MODE 5
10    REM RANDOM DISTRIBUTIE
15    REM HET PROGRAMMA EINDIGT MET "OFF SCREEN",
16    REM ONDERTUSSEN KAN U WEDDENSCHAPPEN AFSLUITEN....
20    DIM R(15.0)
30    Q%=RND(1.0)*16.0:REM Q%=RND(16) FOR SOFTWARE RANDOM
40    R(Q%)=R(Q%)+1.0
50    DRAW Q%*20,R(Q%) Q%*20+5,R(Q%) Q%
60    GOTO 30
```

TITLE OF GAMES COLLECTION 1

```
10    MODE 0:PRINT CHR$(12.0);:COLORT 8 0 8 8
20    POKE #75,32
25    FOR Y=15.0 TO 6.0 STEP -1.0:READ A$
30    P=(Y-5.0)*2.0:E=#BFEF-#86*(24.0-P)-12.0:POKE #BFEE-(24-P)*#86,#C0+Y
40    FOR X=0.0 TO LEN(A$)-1.0:A=ASC(MID$(A$,X,1))
45    F=E-X*2.0:D=#B3E5-12.0-X*2.0
50    FOR Z=D TO F STEP #86:POKE Z,A:WAIT TIME 1:POKE Z,32
60    NEXT:POKE Z,A:NEXT:NEXT
100   DATA YATHZEE,AWARI,SUBMARINE,STARTREK,KANONSPEL,OTHELLO,LUNAR LANDING,REACTIETEST,VIER OP EEN RIJ
150   DATA "  DAInamic  Software "
```

LIST

```
PAGE 01     : FASING KEYBOARD MUSIC

002                     SCTRL   EQU   :FC06      8253 CONTROL
003                     SCH0    EQU   :FC00      CHANNEL 0
004                     SCH1    EQU   SCH0+2
005                     SCH2    EQU   SCH0+4
006                     TABLE   EQU   :9000
007                     VOL0    EQU   :FD04      VOLUME CH 0+1
008                     VOL2    EQU   VOL0+1     VOLUME CH 2(+NOISE)
009                             ORG   :300
010 0300 E5             IN      PUSH  H           SAVE REGISTERS
011 0301 C5                     PUSH  B
012 0302 D5                     PUSH  D
013 0303 F5                     PUSH  PSW
014 0304 010000                 LXI   B,:0
015 0307 3E30                   MVI   A,:30       CH0 MODE0 2BYTES
016 0309 3206FC                 STA   SCTRL
017 030C 3EFF                   MVI   A,:FF       VOL 0+1 MAX
018 030E 3204FD                 STA   VOL0
019 0311 3E0F                   MVI   A,:F        VOL 2
020 0313 3205FD                 STA   VOL2
021 0316 CDBBD6         GETC    CALL  :D6BB       GETC:ASCII IN A
022 0319 CA1603                 JZ    GETC        NO KEY
023 031C FE09                   CPI   9           TAB:OUT
024 031E CA5603                 JZ    OUT
025 0321 07                     RLC               A          A*2
026 0322 4F                     MOV   C,A         OFFSET VALUE IN C
027 0323 2106FC                 LXI   H,SCTRL
028 0326 3636                   MVI   M,:36       CONTROL BYTE FOR 8253
029 0328 210090                 LXI   H,TABLE
030 032B 09                     DAD   B
031 032C 7E                     MOV   A,M
032 032D 3200FC                 STA   SCH0        OSCILLATOR 0
033 0330 23                     INX   H
034 0331 7E                     MOV   A,M
035 0332 3200FC                 STA   SCH0
036 0335 210090                 LXI   H,TABLE
037 0338 09                     DAD   B
038 0339 7E                     MOV   A,M
039 033A 3C                     INR   A
040 033B 3C                     INR   A
041 033C 3202FC                 STA   SCH1        OSCILLATOR 1
042 033F 23                     INX   H
043 0340 7E                     MOV   A,M
044 0341 3202FC                 STA   SCH1
045 0344 210090                 LXI   H,TABLE
046 0347 09                     DAD   B
047 0348 7E                     MOV   A,M
048 0349 3C                     INR   A
049 034A 3C                     INR   A
050 034B 3204FC                 STA   SCH2        OSCILLATOR 2
051 034E 23                     INX   H
052 034F 7E                     MOV   A,M
053 0350 3204FC                 STA   SCH2
054 0353 C31603                 JMP   GETC
```

```
PAGE 02     : FASING KEYBOARD MUSIC

055 0356 F1          OUT     POP   PSW
056 0357 D1                  POP   D
057 0358 C1                  POP   B
058 0359 E1                  POP   H
059 035A C9                  RET
060 035B                     END

****************************
* S Y M B O L   T A B L E *
****************************

GETC    0316    IN      0300    OUT     0356    SCH0    FC00
SCH1    FC02    SCH2    FC04    SCTRL   FC06    TABLE   9000
VOL0    FD04    VOL2    FD05

#P.
0300 E5 C5 D5 F5 01 00 00 3E 30 32 06 FC 3E FF 32 04
0310 FD 3E 0F 32 05 FD CD BB D6 CA 16 03 FE 09 CA 56
0320 03 07 4F 21 06 FC 36 36 21 00 90 09 7E 32 00 FC
0330 23 7E 32 00 FC 21 00 90 09 7E 3C 3C 32 02 FC 23
0340 7E 32 02 FC 21 00 90 09 7E 3C 3C 32 04 FC 23 7E
0350 32 04 FC C3 16 03 F1 D1 C1 E1 C9
```

```
5     REM FASING KEYBOARD MUSIC:TABLE CEATOR
6     REM ----------------------------------
10    READ A$:IF A$="STOP" THEN END
20    A=ASC(A$)
25    PRINT A
30    READ F
40    FR=FREQ(F):REM OF FR=FREQ(F/2)...
50    POKE #9001+2*A,FR/256:POKE #9000+2*A,FR MOD 256
60    GOTO 10
100   DATA Z,528,X,595,C,660,V,704,B,792,N,880,M,990,",",1056
110   DATA S,561,D,627,G,748,H,836,J,935,L,1122,;,1254
115   DATA .,1188,/,1320
120   DATA STOP
```

SHORT RANDOM ROUTINE (NOISE)

```
PAGE 01

001                          ORG    :300
002                RANDOM    EQU    :2FE
003                BITS      EQU    :FD00
004 0300 E5                  PUSH   H
005 0301 F5                  PUSH   PSW
006 0302 3A00FD              LDA    BITS
007 0305 2AFE02              LHLD   RANDOM
008 0308 29                  DAD    H
009 0309 87                  ADD    A
010 030A F20E03              JP     NOSET
011 030D 23                  INX    H
012 030E 22FE02    NOSET     SHLD   RANDOM
013 0311 F1                  POP    PSW
014 0312 E1                  POP    H
015 0313 E9                  PCHL
016 0314           END       END

****************************
* S Y M B O L   T A B L E *
****************************

BITS    FD00    END     0314    NOSET   030E    RANDOM  02FE
```

SDOT

PURPOSE : DRAWS A DOT ON THE SCREEN
CALL : RST 5 + DATA 1E (actual call:E710)
ENTRY : C = Y
HL= X
A = COLOUR OF DOT
EXIT : CARRY UNSET:ALL REGISTERS PRESERVED EXCEPT F
CARRY SET :A CONTAINS ERROR CODE (colour not available, off screen)
OTHER REGISTERS (EXCEPT F) PRESERVED

SDRAW

PURPOSE : DRAWS A LINE ON THE SCREEN
CALL : RST 5 + DATA 21 (actual call:E71B)
ENTRY : B = Y1
DE= X1
C = Y2
HL= X2
A = COLOUR FOR LINE
EXIT : AS DOT
note: The package will draw the line starting at the left end, whichever order the parameters are given in.

SFILL

PURPOSE : FILLS A RECTANGULAR AREA ON SCREEN
CALL : RST 5 + DATA 24 (actual call:E818)
ENTRY : AS DRAW
EXIT : AS DRAW
notes: The middle of the rectangle is filled first,then the left and the right vertical strips.
Rectangle is filled in same order,whichever order parameters are given in.

SSCRN

PURPOSE : ASKS COLOUR OF A POINT ON SCREEN AND SIZE OF GRAPHICS SCREEN
CALL : RST 5 + DATA 27 (actual call: E8F4
ENTRY : C = Y, HL = X
EXIT : AS DOT EXCEPT THAT IF CARRY UNSET:
A contains colour, B,DE MAX coordinates of screen

didacom

D A T A S T A T E M E N T G E N E R A T O R DSG

De onderstaande subroutine is in de onderwijspraktijk ontstaan. Leerkrachten die niets van programmeertalen afweten, moet je niet in een Basic-programma laten struikelen, want dan schakelen ze nooit meer de computer in bij hun lessen. Vandaar dat de werkgroepleden van Didacom via een subroutine het maken van een eigen meerkeuzetoets voor de leerkracht aantrekkelijk hebben gemaakt, door de ingevoerde vragen en antwoordmogelijkheden automatisch naar een datagebied te laten wegzetten.

De datastatementgenerator DSG is het hart van het automatisch genererende multiple-choice programma van Didacom en werd voor DAInamics gerealiseerd door F. Peters in Hoorn.

Veel programmeerplezier.

```
    1 REM dsg
  100 CLEAR 2ØØ
  200 DIM X$(4Ø.Ø)
  300 READ X$
  400 INPUT "Hoeveel tekens ";X:PRINT
  500 INPUT "Welke tekens ";X$
29056 XA=PEEK(#124)+PEEK(#125)*256+5:XS$=""
29057 FOR XI=Ø.Ø TO X-LEN(X$):XS$=XS$+" ":NEXT XI
29058 X$=X$+XS$
29059 FOR XI=Ø.Ø TO X
29060 POKE XA, ASC(MID$(X$,XI,1)):XA=XA+1.Ø
29061 NEXT XI
29062 READ X$
32003 DATA
32004 DATA.........
33500 PRINT
33600 LIST 32003-32004
34000 GOTO 1
```

nvdr: Er kunnen niet meer tekens ingevoerd worden dan er punten voorzien zijn op DATAlijn 32004.
Bij invoer van meer tekens gaat het programma de mist in..
Dit kan voorkomen worden door op lijn 300 2X X$ te lezen en dan met LEN(X$) na te kijken hoeveel tekens mogen ingevoerd worden.

BESCHRIJVING HARD & SOFTWARE PROJECT : MOTORIEKE TRAINING

Doel : jonge kinderen met motorische storingen rijp maken om te gaan leren schrijven.

Probleem : motorisch gestoorde kinderen kunnen hun arm- en hand-motoriek niet zodanig beheersen dat ze met vrucht de schrijfles kunnen meedoen.

Huidige oplossingen: fysiotherapie,trainingskaarten voorbereidend schrijven,prikkaarten enz. Het nadeel van deze oefeningen is dat er niet direct gescoord kan worden om verbetering van het resultaat waar te nemen, behalve als er een testsituatie gecreëerd wordt.

Oplossing met computergebruik:

HARDWARE: Er wordt een kleine computer gebruikt die de mogelijkheden heeft de signalen van een lichtpen om te zetten naar binaire code.In een tafel wordt een kleurenmonitor onder een lichte hoek gemonteerd, omdat zo de beste schrijfhouding kan worden bereikt.De lichtpen is verbonden met de kleine computer, de monitor eveneens.

SOFTWARE:Het beeldscherm is in de achtergrond donker van kleur, terwijl er op de voorgrond golflijnen gegenereerd worden. Deze golflijnen zijn duidelijk en breed. Het kind heeft de opdracht met de lichtpen over deze golflijnen heen te gaan. Komt de pen in het donkere gebied, dan scoort de computer. De moeilijkheidsgraden van deze oefeningen kunnen varieren, doordat er meer golven per beeldscherm gegenereerd worden, doordat de breedte van de golflijn afneemt, doordat andere ingewikkelder vormen gegenereerd worden.

FINANCIERING: Er is een kleine computer op de school aanwezig, een DAI 48K, een lichtpensysteem is er niet.
Een kleurenmonitor zal moeten worden aangeschaft, daarvoor zal geld moeten gevraagd worden aan een of andere instantie, zo ook voor de AM 9511.

REALISATIE: Er zal een onderzoekje gedaan moeten worden naar een nuttig lichtpensysteem en zijn interfacemogelijkheden via DCE bus, RS232 of PADDLE-ingangen.
Indien nodig dan moet er extra hardware ontwikkeld worden.
Een MATHCHIP AM9511 kan zijn diensten bewijzen in de DAI, omdat die de wachttijden voor het kind bij het uitrekenen van de schrijfvormen duidelijk zal beperken.
Bij de softwarerealisatie dient een wiskundige de te realiseren schrijfvormen in formules samen te vatten, om een zo kort mogelijke berekeningstijd te verkrijgen.

CONTACTADRES voor dit project:
Inno Broekman
Avenbeek 98
2182 RZ HILLEGOM NEDERLAND

POWER - ON INITIALISATIE

In dit overzicht wordt het initialiseren van de DAI pC bij het inschakelen behandeld. Alle adressen en data staan in hex-notatie. De video-RAM adressen zijn gebaseerd op een 48-K machine.

- Zet stackpointer op F900.

- Laad FD06 met 30: ROM-bank 0, cassette motors uit.

- Laad TICC interrupt mask register (FFF8) met 04; alleen externe interrupts zijn dan toegestaan.

- Laad het TICC command register met 0C en direkt daarna met 0D:
  - TICC reset
  - selektie IN7 interrupt (20 ms. blanking puls van TV logica)
  - INTA signaal van CPU wordt geaccepteerd.

- Laad adressen 01C0 en 01C1. Deze worden gebruikt als 'timers' in resp. de RST7 en RST6 restart procedures.

- Laad Timer 3 (FFFB) en Timer 4 (FFFC) van de TICC 5501.

- Laad de adressen 0000 - 003F met de interrupt vector routines en de vector adressen. De laatste staan op 0062 - 0071.

- Controleer of de Math.chip aanwezig is. In adres 00D4 wordt 00 of 7B gezet, afhankelijk van het niet of wel aanwezig zijn van de AMD9511.
  De adressen 00D1/D0 en 00D3/D2 worden geladen met resp. C7F2 en DDE0.

- Controleer de RAM-capaciteit van de DAI pC.
  Dit wordt gedaan door de inhoud van de adressen 1000, 2000 enz. te lezen, de inhoud te inverteren en dan weer terug te laden. Daarna wordt weer gekeken wat er in de desbetreffende geheugenlocatie staat.
  Zo lang er RAM aanwezig is, gaat deze procedure op. Is er geen RAM meer (8K - 32K machines), of is het ROM bereikt (48K) werkt dit niet meer. Zo wordt het hoogste RAM adres bepaald, dat gebruikt wordt voor de opbouw van het video-RAM.

---- INITIERING VAN HET VIDEO-RAM:

- Elke routine die iets met het video-RAM te doen heeft, maakt gebruik van de RST5 routine. Hierin wordt geswitched naar ROM-bank 2. Het actuele adres wordt bepaald door de data die volgt op de RST5 instruktie:

bv:	xxx0	RST 5	Er wordt gesprongen naar
	xxx1	data: 03	2E003.

  Voor het opzetten van het video-RAM wordt gebruik gemaakt van RST5/00.

- Laad 0081/80 met BFFF en 0083/82 met BFEF.

- Zet 'screen mode' op mode 1 (009D = 00).

- Definieer cursor mode en ASCII-waarde voor cursor (0074, 0075).

- Bepaal de kleuren voor de COLORT registers (007C - 007F).

- Selecteer 'screen mode' 0.

- Initieer de kleuren voor de COLORG registers (009E - 00A1).

- Laad 00C5/C4 met CA01 en 00C7/C6 met CA25.
  Zet 'screen mode' op 10.

- Laad B350 in 02A6/A5.

- Laad 0084 - 0098 met adressen betreffende het video-RAM.

- Laad de kleurregisters BFF0 - BFFF met data.

- Nu wordt het hele video-RAM ingedeeld, en wel voor een leeg scherm in mode 0.
  De lijn controle bytes worden 7A, de kleur bytes 40.
  De karakter data- en kleur bytes worden 20 (spatie), resp. 00.

- In 0079/78 wordt het lijncontrole adres van de eerste regel opgeslagen (later van de op dat moment in gebruik zijnde regel) en in 007A het laatst bruikbare adres van deze lijn.

- Laad de kleurregisters B350 - B35F met data.
  Zowel het kleurregister BFF0 - BFFF als het kleurregister B350 - B35F worden gebruikt voor COLORT kleuren.

- Selecteer 'screen mode' 0.

- NU IS HET VIDEO-RAM OPGEBOUWD.

- Laad 0296 met 00.

→ Selecteer scherm en RS232 als output (0131), en keyboard en scherm als input (0135).

→ Zet het TICC communications rate register (FFF5) op 9600 baud met 1 stop bit.

---- <u>NU WORDT HET VIDEO RAM GEVULD MET "DAI PERSONAL COMPUTER".</u>

Hiervoor wordt RST5/03 gebruikt. Deze routine wordt i.h.a. altijd gebruikt om data naar het video RAM te brengen.

- Er wordt gesprongen naar regel 7 van het scherm d.m.v. 6x 'carriage return'

- Op de 7e regel van boven wordt nu 'DAI PERSONAL COMPUTER' in het video RAM geplaatst. Dit wordt gedaan in mode 0, in de hoogste resolutie, met 14 spaties tussen 'PERSONAL' en 'COMPUTER'.

- Nu wordt regel 7 geset voor medium resolutie (lijncontrole byte 5F. Dit geeft de bekende grotere letters, echter 'COMPUTER' staat nu buiten het scherm.
  Daartoe wordt nu tussen 'PERSONAL' en 'COMPUTER' een nieuw lijncontrole byte gecreëerd (5F + 40). Dit heeft tot resultaat dat 'COMPUTER' nu keurig onder het voorgaande komt te staan.
  De rest van het video_RAM wordt nu gereorganiseerd.

---- Het video gedeelte is nu klaar.

Nu moeten nog diverse 'pointers' en andere geheugenlokaties geïnitieerd worden, voordat er gewacht kan worden op een input.

- Initieer de Sound generator (FC06). Het volume wordt op 0 gezet (FD04/05).

- In de lokaties 01C2, 01D0, 01DE en 01EC wordt FF geladen.

- De inhoud van de ROM-adressen D7A4 - D7CA wordt naar de RAM-lokaties 02C5 - 02EB gebracht. Dit stukje programma wordt gebruikt voor de cassette-sturing. In het geval er van een floppy-disc gebruik gemaakt wordt, dient dit gedeelte gewijzigd te worden (zie handboek).

→ De DCE-bus (GIC) wordt geïnitieerd via FE01 en FE03. Hiervoor wordt RST1/0C gebruikt. Er wordt gekeken of de DCE-bus aktief is; voor de power-on reset nemen we aan van niet.

- De pointers voor het BASIC programma worden geladen.
  029B/C : start adres HEAP.
  029D/E : grootte HEAP.
  029F/A0: start adres tekst buffer.
  02A1/2 : start adres symbool tabel.
  02A3/4 : begin video RAM.

- Cassette poort 1 wordt geaktiveerd.

- De GETC routine wordt voorbereid:
  - ROM-pointer 02A7/8 wordt geladen met het begin adres van de tabel met ASCII-codes voor de karakters.
  - GETC wordt ge-enabled door 00 in 02B9.
  - 02BE/F en 02C0/1 worden geladen met 02BA. Dit is het adres waar de ASCII-code van het eerste karakter opgeslagen wordt.
  - 02C4 wordt met FF geladen (BREAK).

- De lokaties 0275 - 028F worden geladen met 00.

---- <u>NU WORDT GEWACHT TOT EEN TOETS GEDRUKT WORDT</u>.

Wanneer dit gebeurt, wordt de GETC routine uitgevoerd. Hiervoor wordt RST1/15 gebruikt.

- Resultaat van de GETC-routine:
  - De ASCII-code van de gedrukte toets staat in de accu en in 02BA. 02B9 is weer geladen met FF en (02BE) en (02C0) zijn met 1 verhoogd. (02C4) = 00.

- De kleuren van de COLORT registers worden veranderd en de registers zelf gewijzigd via RST5/06.

- Het scherm wordt schoongemaakt en de cursor wordt op BFE7 (links boven) gezet.

- Nu wordt " BASIC V1.Ø " op de eerste regel geschreven d.m.v. RST5/03. Daarna volgt een'carriage return'.

---- Nu kan de DAI pC voor BASIC klaargemaakt worden.

- Diverse RAM-lokaties worden met 00 geladen:
  0100, 0101, 0104, 0105, 0113, 0114, 0117, 0118, 0122.
  De stackpointer wordt weer op F900 gezet; deze waarde wordt ook geladen in 0128/27.

- Het TICC interrupt mask register (FFF8) wordt ge-update; eerst voor interrupts van TIMER 4 en external interrupts, daarna ook voor IN7 interrupts (TV page blanking signal).

- via RST5/0C, verschillende CPU registers worden aangepast.

- Nu wordt de 'prompt' (*) aan het begin van de 2e regel geplaatst via RST5/03, gevolgd door de cursor.

- Daarna komt het programma in een eideloze lus in een gedeelte van de GETC-routine. Deze lus kan alleen onderbroken worden door interrupts.

---- INTERRUPTS.

In dit stadium zijn er 2 mogelijke interrupts (behalve de externe die we verder buiten beschouwing laten):

- Wanneer timer 4 is afgelopen. Dan wordt RST6 aangeroepen.
- Via het TV 'page blanking signal' (elke 20 ms). Hier wordt RST7 gebruikt.

- RST7:
  - (01C0) wordt ge-decrement. Zolang het resultaat niet = 0, wordt terug gesprongen naar het hoofdprogramma.
    Is dit wel het geval, dan volgt RST5/12. Hierin laat men de cursor knipperen door elke keer als de RST7 routine dit uitvoerd afwisselend 5F of 20 voor de cursor te schrijven (balk of spatie).

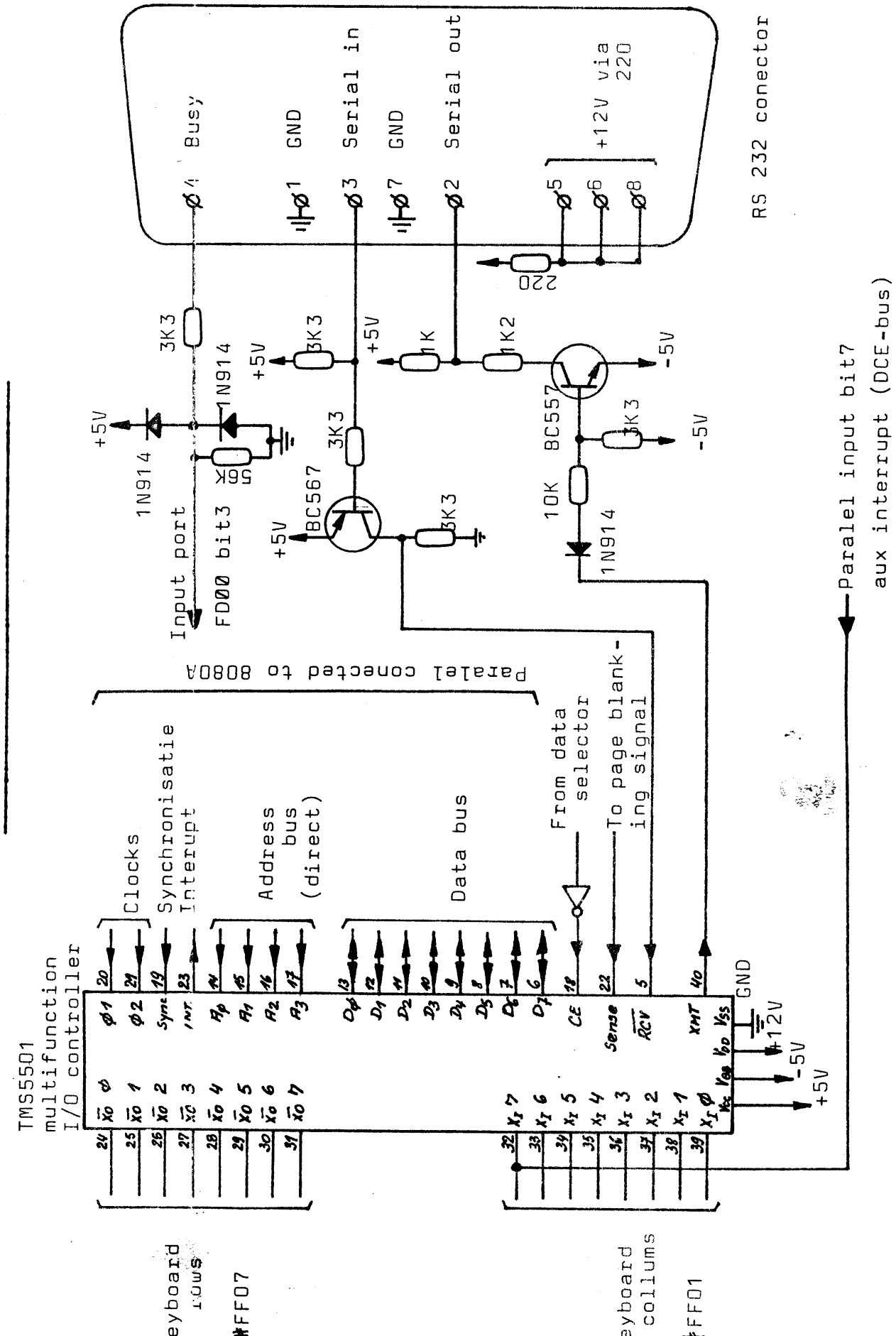
- RST6:
  - Timer 4 wordt herladen.
  - Wanneer (01C0) na decrementing niet = 1, dan wordt terug gesprongen naar het hoofdprogramma. Is dit wel het geval, dan wordt de GETC-routine aangeroepen en uitgevoerd.

---- Op deze manier kunnen nu inputs gemaakt worden.

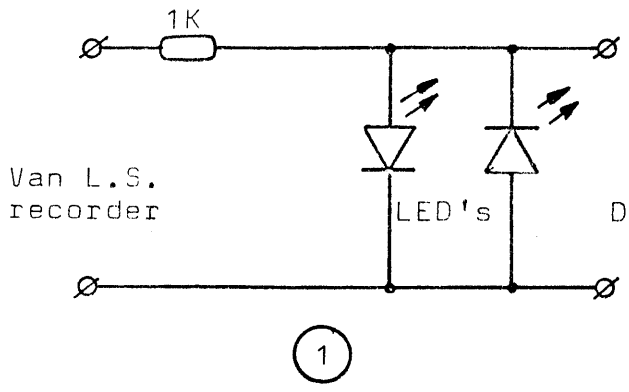
EINDE POWER - ON INITIALISATIE

De complete power-on initialisatie routine (ca. 70 pagina's gedetaileerd uitgewerkte software!!) kan besteld worden door overmaking van *f* 55.- op postgiro-rekening 1.492.579 t.n.v. B.J.Boerrigter, Fabritiusstraat 15, 6174 RG Sweikhuizen, Nederland. S.v.p. vermelden: 'DAI power-on software'.

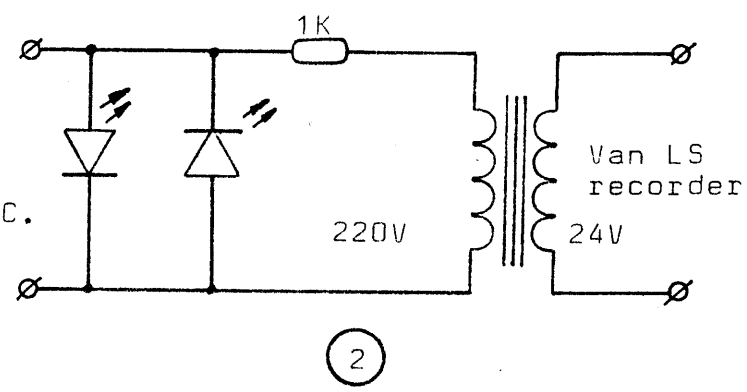
Een engelse versie van dit verhaal is eveneens beschikbaar tegen copiëer- en verzendkosten.


SERIAL RS232 INTERFACE ON DAI pc
RS 232 conector
Busy
GND
Serial in
GND
Serial out
+12V via 220
220
3K3
1N914
+5V
3K3
1K
1K2
-5V
BC557
3K3
-5V
10K
1N914
BC567
3K3
+5V
1N914
56K
Input port
FD00 bit3
Paralel input bit7
aux interrupt (DCE-bus)
Paralel conected to 8080A
Clocks
Synchronisatie
Interupt
Address bus (direct)
Data bus
From data selector
To page blanking signal
TMS5501
multifunction
I/O controller
GND
+12V
-5V
+5V
keyboard rows
#FF07
keyboard collums
#FF01

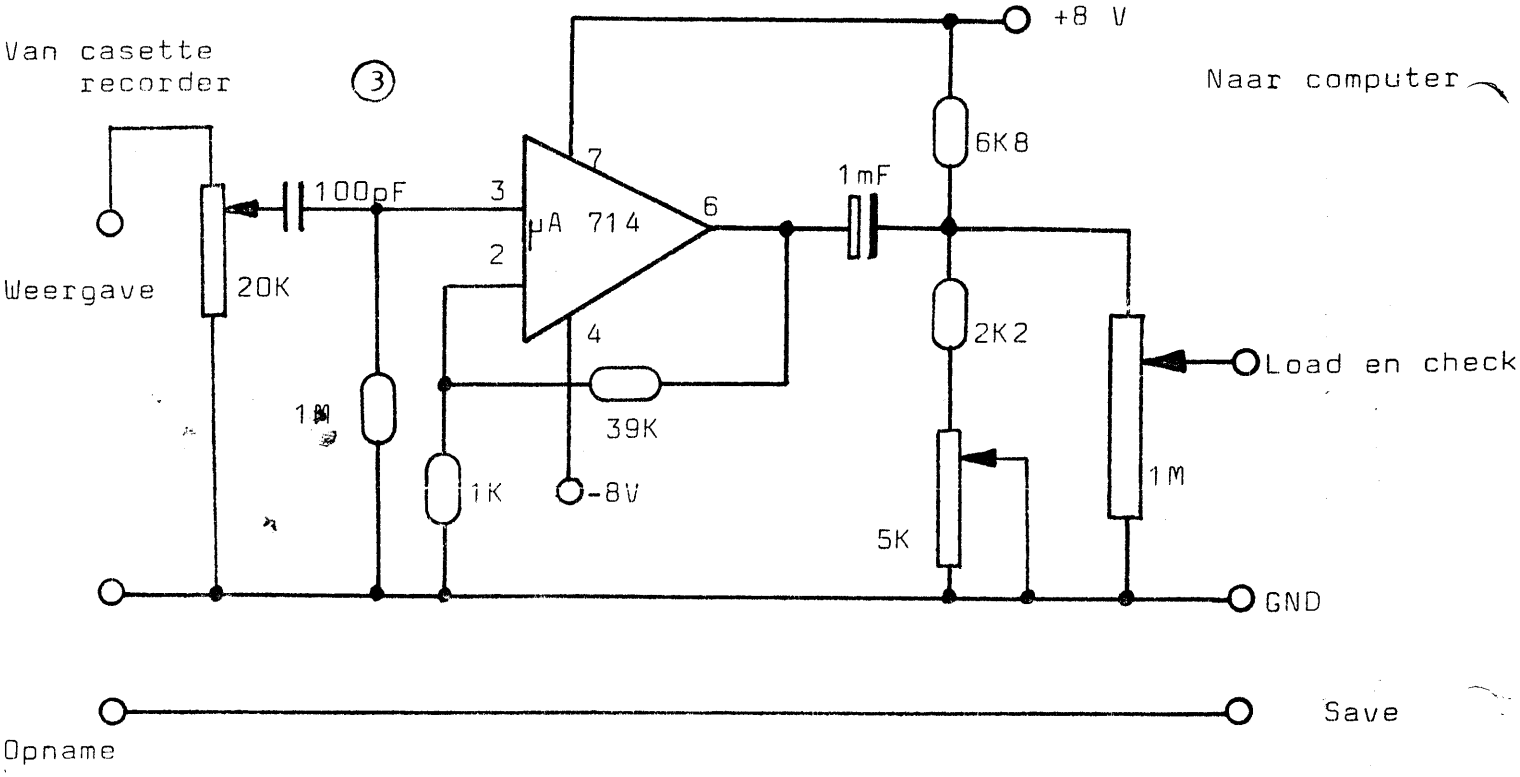
SIGNAL CLIPPERS/PROCESSORS for CASSETTE INTERFACE



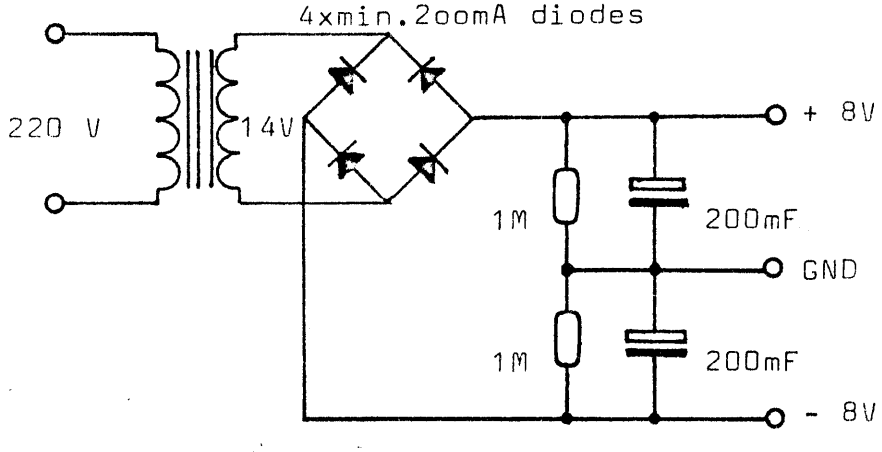
①

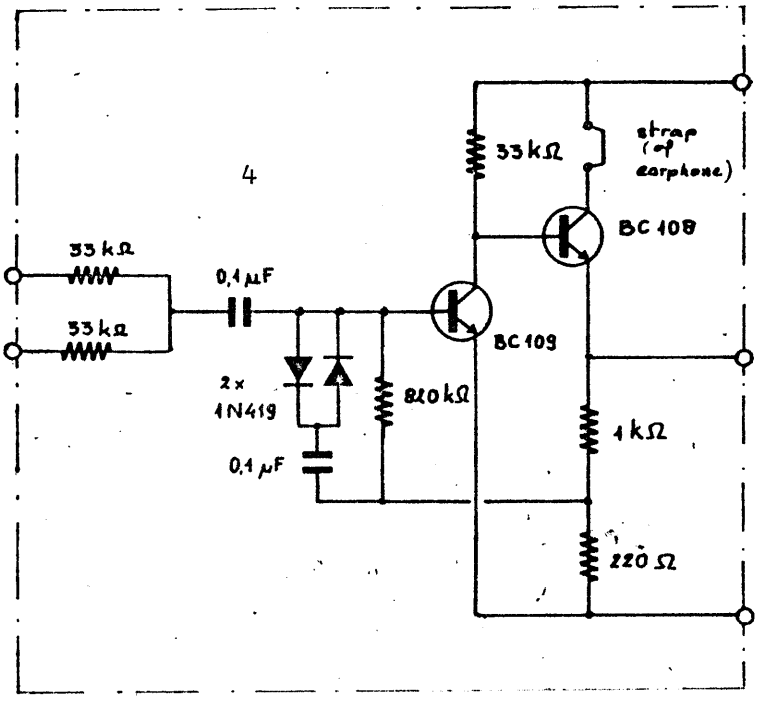


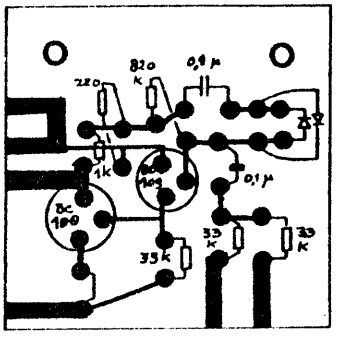
②

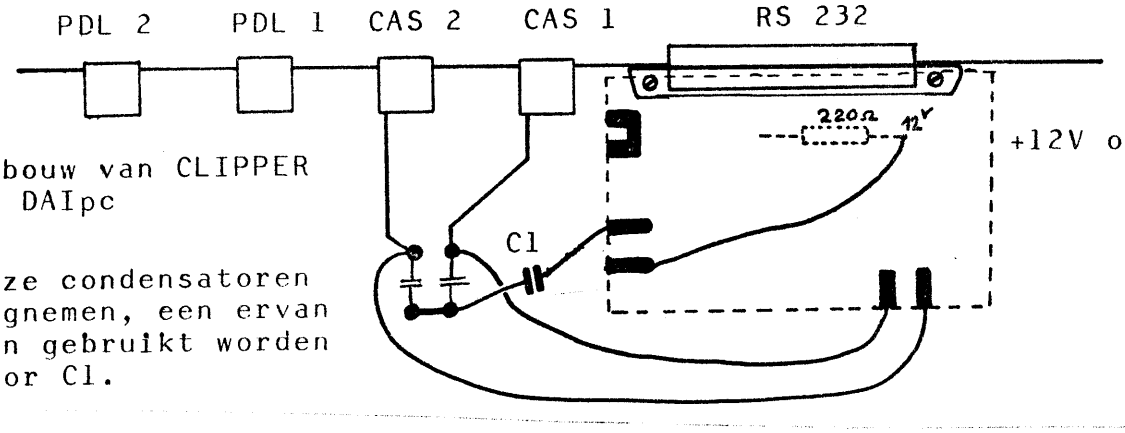


POWER SUPPLY for ③









Inbouw van CLIPPER in DAIpc

deze condensatoren wegnemen, een ervan kan gebruikt worden voor C1.

1 & 2 : een paar eenvoudige oplossingen van W.de leeuw van Weenen.

1. 2 LED's antiparallel : zeer efficient indien de LS-uitgang een fors signaal kan leveren.
2. indien het niveau van de LS-uitgang te laag is wordt de spanning hiermee opgetransformeerd.

3 : een ontwerp van C.De Bont om zijn cassettedeck te kunnen gebruiken met DAIpc. Ingang en uitgang zijn hier continu regelbaar.
714 is een verbeterde versie van 741,de meeste OPAMPS zullen hier voldoen.

4 : een realisatie van DAIclub EUROCONTROL BEEK.
dit is een actieve clipper die het uitgangssignaal constant houdt over een zeer ruim gebied van ingangssignalen.

Mocht U een van deze schakelingen proberen dan zijn wij en de ontwerpers uiteraard benieuwd om uw resultaten te vernemen.
Indien u over twee recorders beschikt kan U moeilijke signalen meestal inlezen door eerst een copie te maken,eventueel iets oversturen.

81 There were more than 250 visitors on our first meeting on 11 april, there were even some friends from Germany!
83 How to connect your DAIpc to a B/W monitor,this applies mostly to the 8K B/W machines sold with the TELEAC course on TV.
88 CATALOG: new formule for our library, most of the programs will be available on collection tapes, at lower prices.
89 MATH CHIP TEST :for some functions the gain in speed is very high!
97 The short 8080 routine runs on interrupt 7, you have command over execution of BASIC and machine language routines,also nice to use with LOOK.
99 Attention for the bargraphs:they are presented on a different scale: the graphs with MATHCHIP are enlarged 13.6 times.
100 You need this clock routine to run the MATHCHIP-TEST program.
101 Screen-copy for MX-80, up to resolution 3&4,by changing the values of Y and X on lines 63008 & 63010 you can make screen copies of a part of the screen.If you use the serial interface to the printer,you have to set the dip-switch for 8 bits!
102 DATA SHEETS OF 9511
108 screen copy with MX-80
109 FGT with paddles:these symbols are available on the new collectio tape of FGT.
110 addresses élBE and élBF are used for the time delays of WAIT TIME and the cursor. You can check this information to create exact time mesuraments,see program examples on p 111.
112 DATA SHEET of N2235,we suppose this type is available all over Europe.
113 An easy way to have audio control of the tape signal.
114 H.Wegman realised a tape operating system for PHILIPS DIGITAL CASSETTE RECORDER.The prorgam will be on the X-bus in EPROM. The system will be distributed by MEMOCOM Holland.
116 about random distribution
117 more random distribution +++ title program for collection tape 1.
118 FASING KEYBOARD MUSIC
119 short 8080 random routine, the noise signal is used as seed, so real random values !!!
120 ENTRY POINTS for DOT,DRAW,FILL,SCRN
121 A small DATA STATEMENTS GENERATOR from DIDACOM, this is an organisation covering the subject EDUCATION&COMPUTERS.
122 A report on a project from DIDACOM: training writing skills.
123 What happens inside DAIpc during POWER-ON ?
127 Schematics of RS 232 interface.
128 Signal processors for cassette interface : the choise is allyours.

DAInamic is 1 year old : time to take some holidays.
We promessed our wifes not to touch the keyboard from 15 aug to 15 sept.
In our next issue you will find the list of DAInamic members+ all about our library,we have even plans to offer you foto's of many programs ... we wish you sunny holidays !

DAI PERSONAL COMPUTER °°° DAI PERSONAL COMPUTER °°°DAI PERSONAL

0	zwart
1	blauw
2	d.rood
3	rood
4	paars
5	groen
6	d.bruin
7	l.bruin
8	grijs
9	blauw
10	oranje
11	rose
12	l.blauw
13	l.groen
14	geel
15	wit

alle adressen in HEXvorm!

29B-29C	start heap
29D-29E	size heap
29F-2A0	start text buffer
2A1-2A2	start symbol table
2A3-2A4	end of symbol table
2A5-2A6	bottom screen ram

131,0	output scrn+ RS232
131,1	screen only
131,2	edit buffer
135,2	read from edit buffer

75	cursor symbol
74	cursor mode
72-73	cursor position

MODE	XMAX	YMAX
1/2	71	64
3/4	159	129
5/6	335	255

40,28 cass motor 1 ON
40,18 cass motor 2 ON
40,30 1 and 2 OFF

MERGE
°CLEAR XXX
°LOAD"A"
°EDIT BREAK/BREAK
°LOAD"B"
°POKE 135,2

COLORG R1 R2 R3 R4
20 21 22 23
16 :R2*R1 R4*R3
17 :R1*R2 R3*R4 32K 7XXX
18 :R3*R1 R4*R2 12K 2XXX
19 :R1*R3 R2*R4 8K 1XXX

IMP INT *** IMP FPT
°IMP FPT
°CLEAR XXXX
°EDIT BREAK/BREAK
°IMP INT
°POKE 135,2

LIJN	CTRL	COLOR	LIJN	CTRL	COLOR
23	BFEF	BFEE	11	B9A7	B9A6
22	BF69	BF68	10	B921	B920
21	BEE3	BEE2	9	B89B	B89A
20	BE5D	BE5C	8	B815	B814
19	BDD7	BDD6	7	B78F	B78E
18	BD51	BD50	6	B709	B708
17	BCCB	BCCA	5	B683	B682
16	BC45	BC44	4	B5FD	B5FC
15	BBBF	BBBE	3	B577	B576
14	BB39	BB38	2	B4F1	B4F0
13	BAB3	BAB2	1	B46B	B46A
12	BA2D	BA2C	0	B3E5	B3E4

CTRL&COLOR BYTES IN A-MODE

MODE	CTRL	COLOR	LIJN
1A/2A	BAE7	BAE6	3
	BA61	BA60	2
	B9DB	B9DA	1
	B955	B954	0
3A/4A	ACD3	ACD2	3
	AC4D	AC4C	2
	ABC7	ABC6	1
	AB41	AB40	0
5A/6A	7557	7556	3
	74D1	74D0	2
	744B	744A	1
	73C5	73C4	0

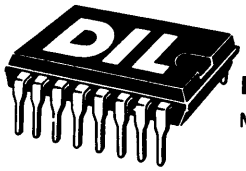
FD00 b2 page signal
b3 serial out rdy
b4 right paddle
b5 left paddle
b6 random data
b7 cass. input
FD01 Trigger paddle
FD04 0-3 volume ch.1(0)
4-7 volume ch.2(1)
FD05 0-3 volume ch.3(2)
4-7 volume noise
FD06 b0 cass.out
b1/2 paddle select
b3 paddle enable
b4 cass motor 1
b5 cass motor 2
b6/7 ROM BANK SWITCH

FF00 ser.inp.buff
FF01 b0-6 keyb.inp.
b7 in7 DCE
FF02 Interr.reg.
FF03 b1 frame error
b2 overrun error
b3 rec.buf.loaded
b4 trans.buf.empty
FF04 COMMAND REGISTER
FF05 BAUD RATE REGISTER
FF06 ser.out buf.
FF07 keyb.output
FF08 interr.mask reg.

TEST EVENT
PEEK(éFD00) IAND 32
PEEK(éFD00) IAND 16
PEEK(éFD00) IAND 48

FF09 TIMER 0
FF0A TIMER 1
FF0B TIMER 2
FF0C TIMER 3
FF0D TIMER 4
8253
CH 0 FC00/FC01
CH 1 FC02/FC03
CH 2 FC04/FC05
STATUS FC06/FC07

DAI-DEALERS *** COMPUTERVERENIGINGEN



D.I.L.-ELEKTRONIKA
Mijnsherenlaan 108, 3081 CH Rotterdam

ALLE DOE-HET-ZELF ELEKTRONIKA - TECHN. TIJDSCHRIFTEN EN -BOEKEN

LEGOTRONICS
Middenstraat 8
8800 ROESELARE BELGIE
tel. 051/207878

ORDIMAX
Rue de la Bonnefemme 11
4030 GRIVEGNEE BELGIE

MULTISOFT
Rue Bargue 25
75015 PARIS FRANCE
7838837

TELEC
Steenstraat 40
9711 GP GRONINGEN NEDERLAND

MSB R.NEDELA
MARTKSTRASSE 3
POSTFACH 1420
D7778 MARKDORF GERMANY

COMPAC
Plaats 25
2513 AD DEN HAAG NEDERLAND

HCC NEDERLAND hobby computer club
Prinsenhof 11
2641 RN PIJNACKER
NEDERLAND

DAI BRUSSEL
Raketstraat 60
1130 BRUSSEL BELGIE
02/2166010

HCC BELGIE
Borkelstraat 51
2120 SCHOTEN BELGIE
031/589674

DAI NEDERLAND
Van Vollenhovenstraat 15A
3016 BE ROTTERDAM NEDERLAND
010/361288

Stichting BASICned
Tolakkerweg 81
3739 JJ HOLL.RADING NEDERLAND

DIDACOM computers&onderwijs
p/a I.BROEKMAN AVENBEECK 98
2182 RZ HILLEGOM
2520/18032 NEDERLAND