[-november 2021-]

[-november 1971-]

[-historie intel-]

- Historie Intel:
  - Het bedrijf is in 1968 door Robert Noyce en Gordon Moore opgericht
  - Kwamen beide bij Fairchild Semiconductor vandaan
  - Santa Clara, Californië
  - Startkapitaal 2,5 miljoen dollar,

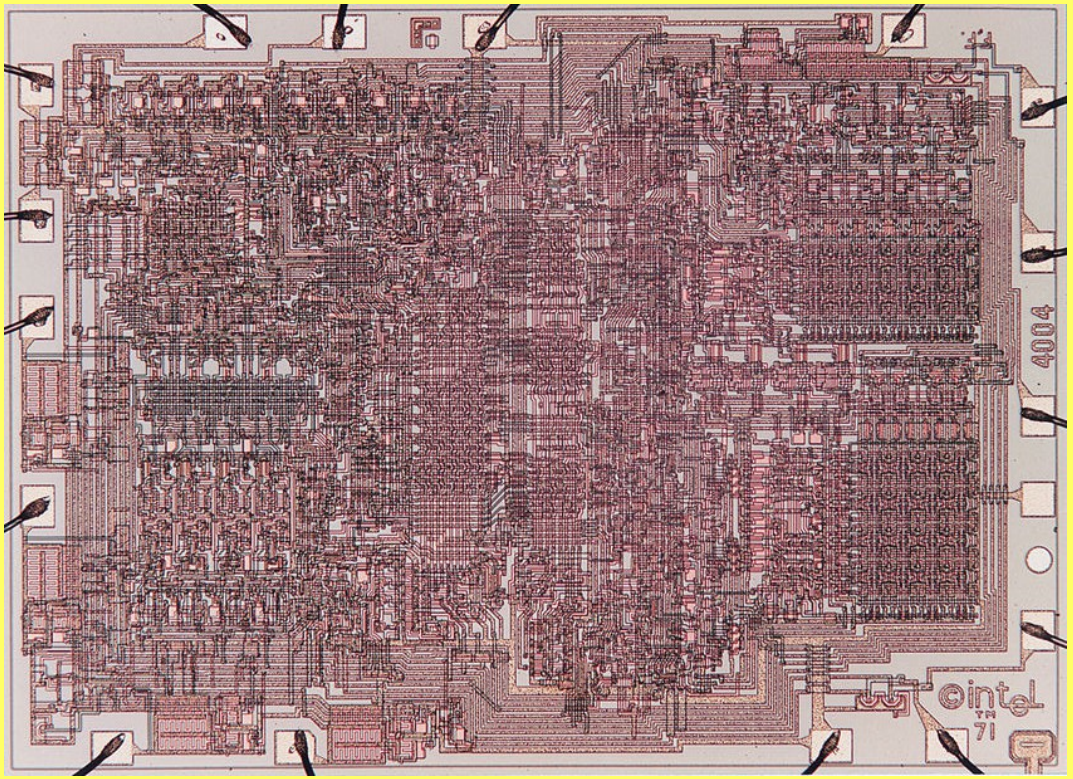
    in 2 dagen bijeengebracht.
  - De derde werknemer was Andrew Grove
  - DRAM 1103, 1 kilobit
  - Best verkopende RAM in 1969 - 1970

[-historie 4004-]

- Historie 4004:
  - 1969, Ted Hoff en Stan Mazor komen bij Intel
  - 1970, Federico Faggin komt bij Fairchild vandaan waar hij aan de SGS-MOS techniek gewerkt had
  - Busicom vraagt om 12 chips te ontwerpen
  - Masatoshi Shima
  - Voorjaar 1971 eerste 4004 afgeleverd
  - Zomer 1971, Busicom 141-FP
  - Intel koopt rechten terug
  - 15 November, advertentie in "Electronic news"

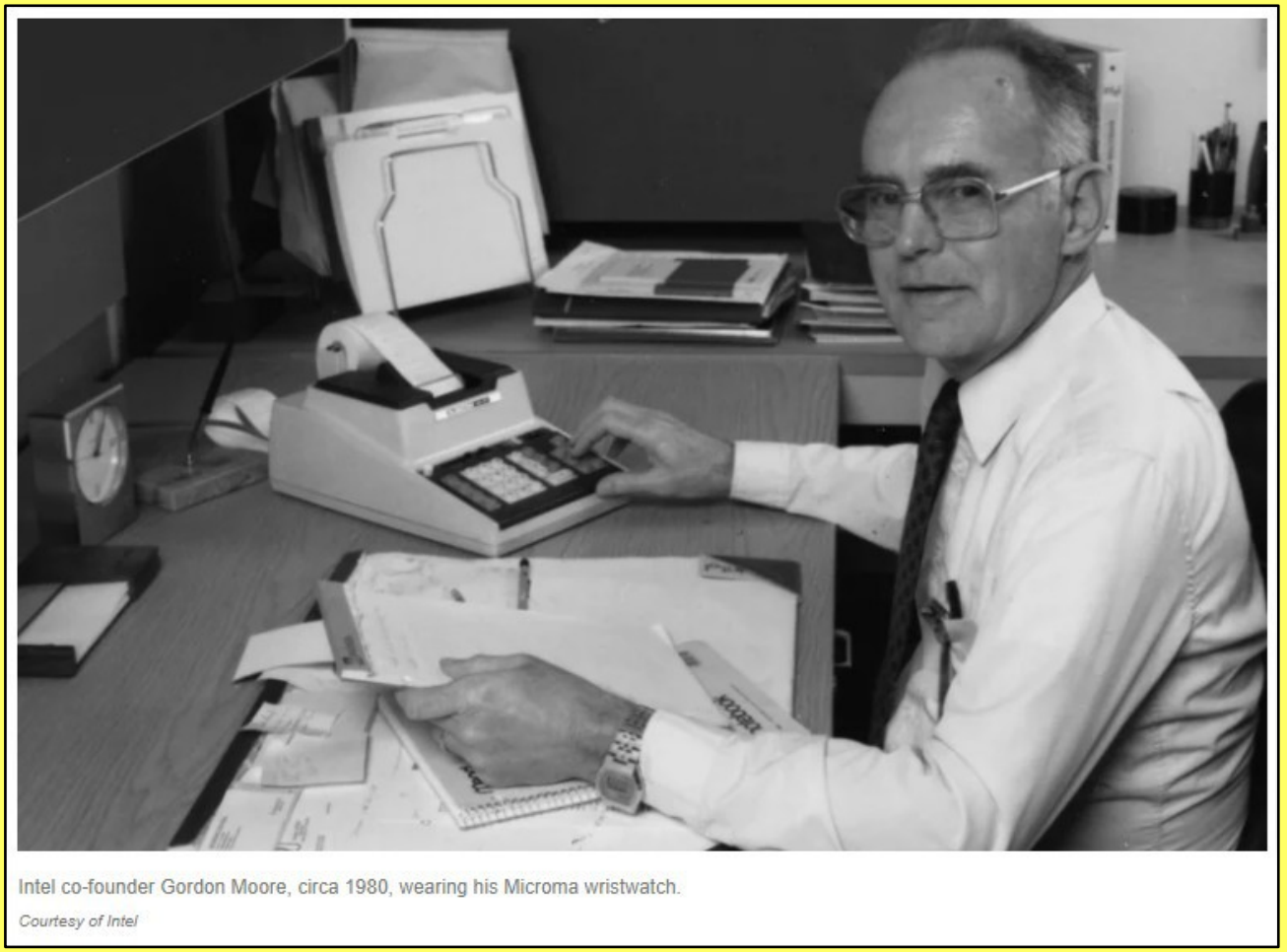
[-historie 4004-]

- Eigenschappen 4004:
  - Silicon Gate Technology (SGT) Metal Oxide Semiconductor
  - 2300 transistors
  - pMOS
  - 4 bits databus
  - 16 Pins IC
  - 740 kHz klok
  - 46 instructies
  - 4 K adres ruimte
  - 16 * 4 bits registers



[-uitstapje intel-]

- Neven aktiviteiten.
- Horloge



Intel co-founder Gordon Moore, circa 1980, wearing his Microma wristwatch.

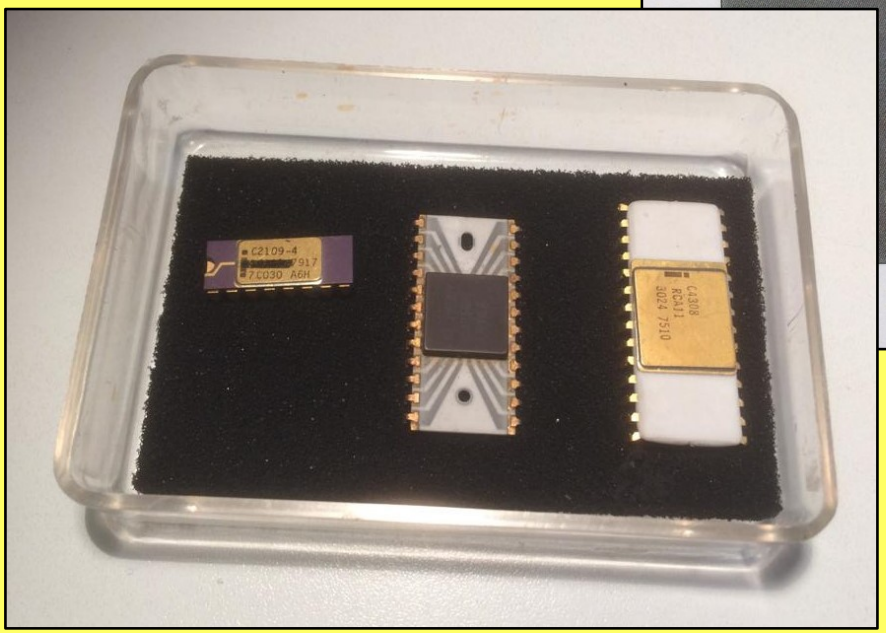
Courtesy of Intel

[-ik en intel-]

- Wat deed ik rond 1970
- Hoe kwam ik in contact met elektronica, computers en de Intel 4004

[-intel-]

- Jan Wubben

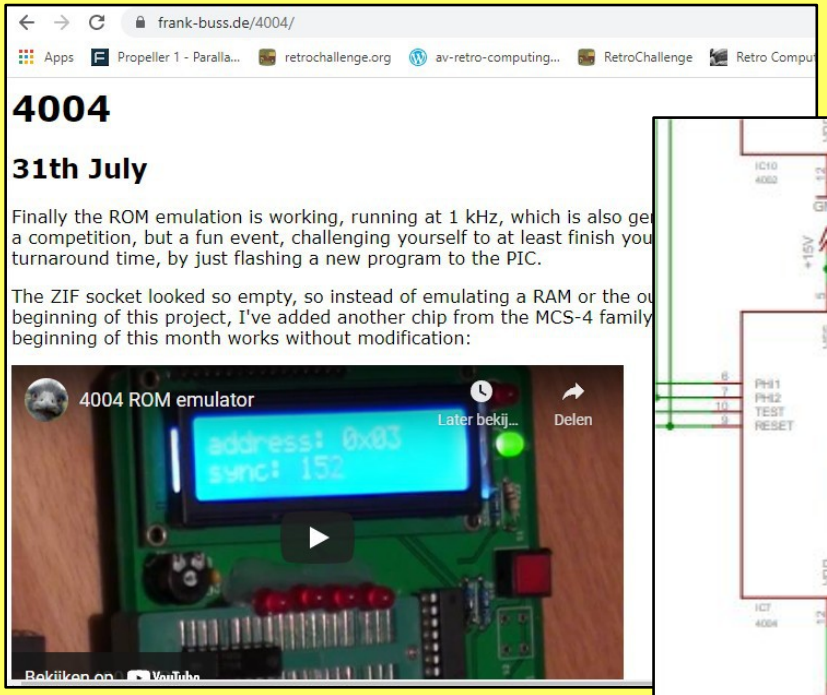


[-intel-]

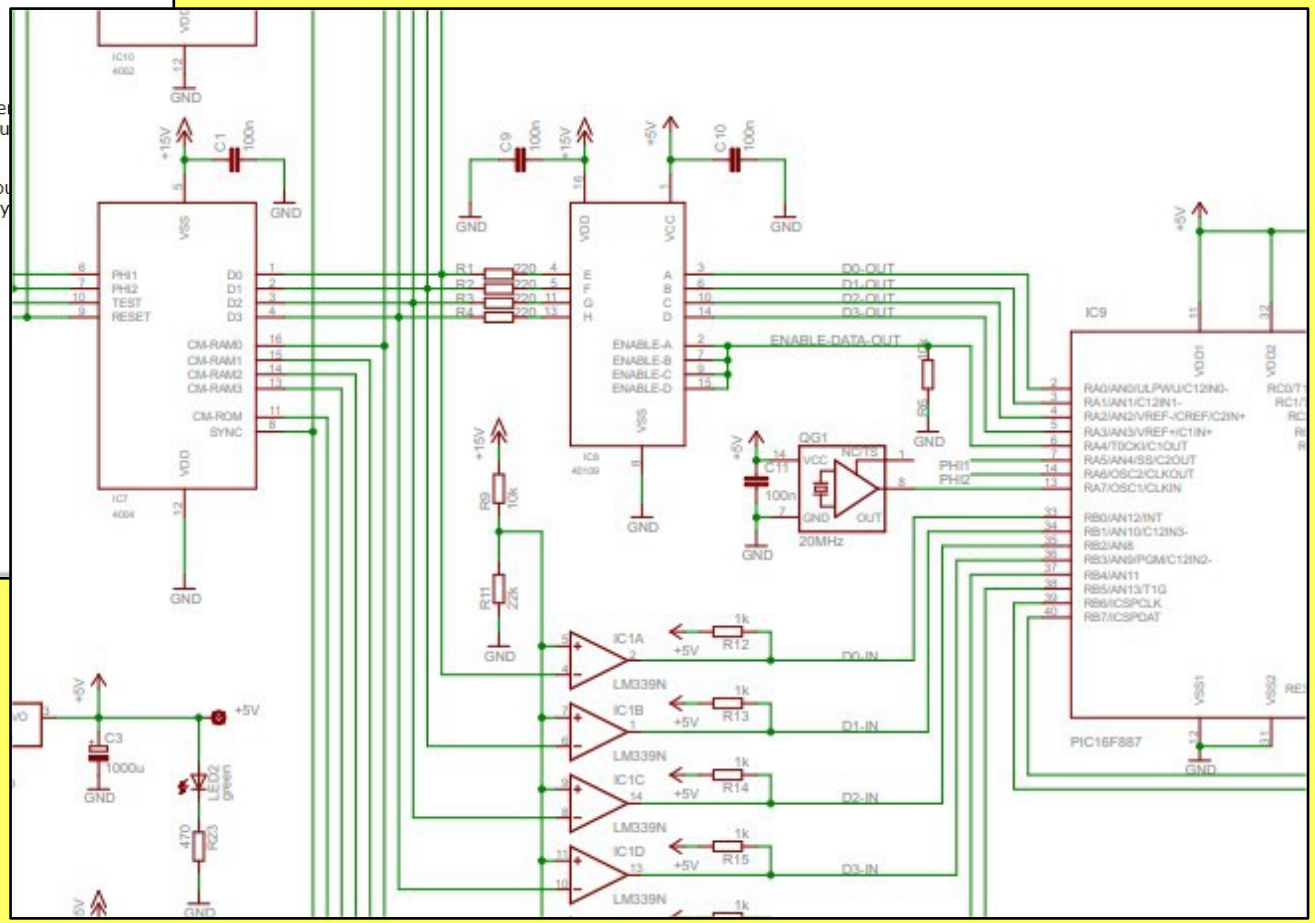
- Ruim 10 jaar in opslag
- Museum stuk
- Werkt hij nog ?
- Hoe van start ?
- Hij eet geen brood !
- ???

[-Idee intel-]

- Frank Buss: https://frank-buss.de/4004/



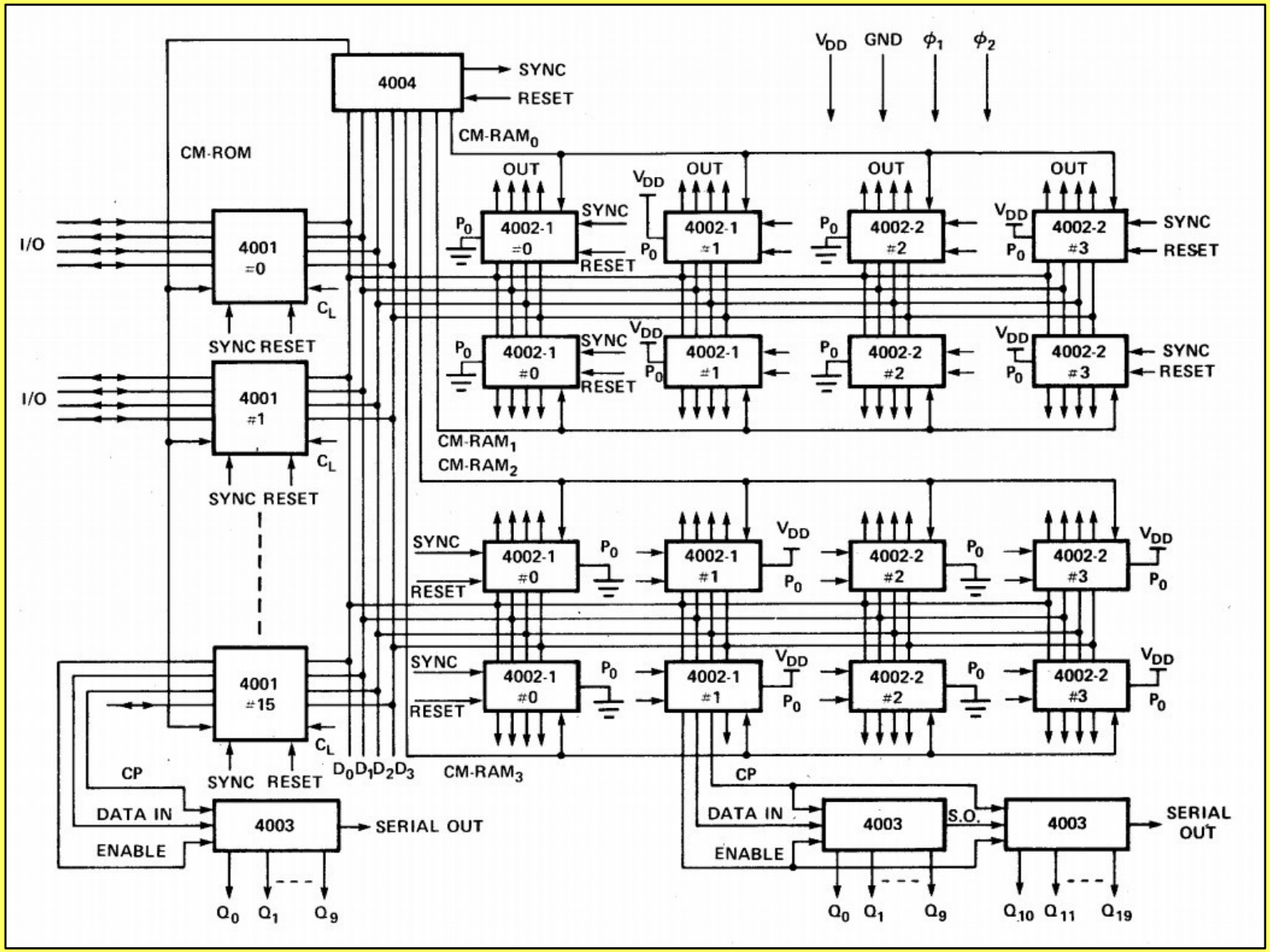
Retrochallenge 2012



[-MCS 4 intel-]

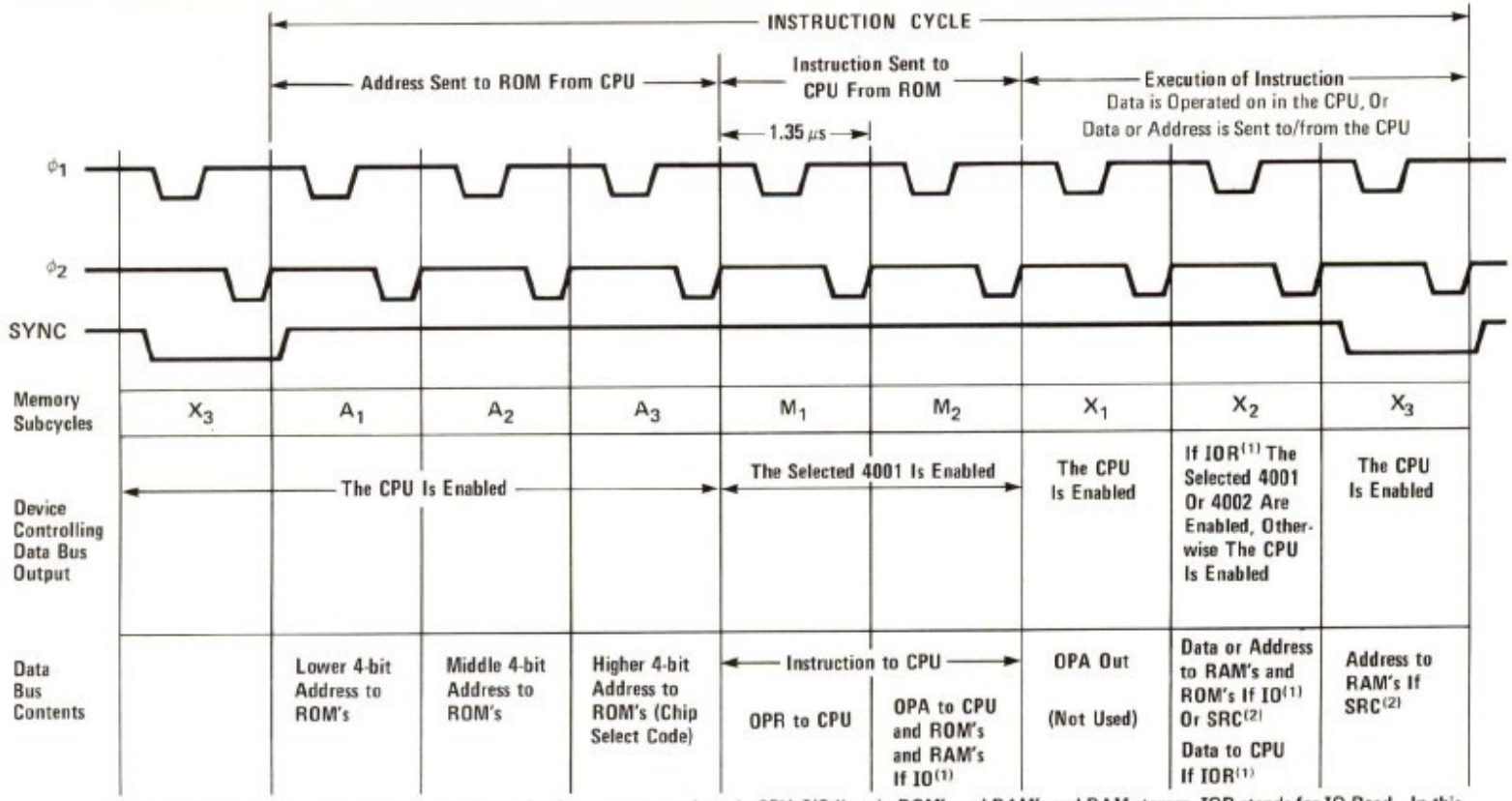
- Welke IC's en functie:
- 4001 – 256 byte ROM, 4 bit I/O
- 4002 – 320 bit RAM, 4 bit O
- 4003 – serieel in, 10 bit parallel uit, schuifregister
- 4004 – microprocessor, 4 bit, 740 kHz
- Latere uitbreiding:
- 4008 – adres uitbreiding tbv EPROM's
- 4009 – adres uitbreiding tbv I/O

[-MCS 4 intel-]


4004
SYNC
RESET
CM-ROM
CM-RAM0
VDD
GND
φ1
φ2
OUT
4001 #0
4001 #1
4001 #15
I/O
CL
SYNC RESET
4002-1 #0
4002-1 #1
4002-2 #2
4002-2 #3
P0
CM-RAM1
CM-RAM2
CM-RAM3
D0D1D2D3
CP
DATA IN
ENABLE
4003
SERIAL OUT
S.O.
Q0 Q1 Q9
Q10 Q11 Q19

[-MCS 4 intel-]

MCS-4 Operation



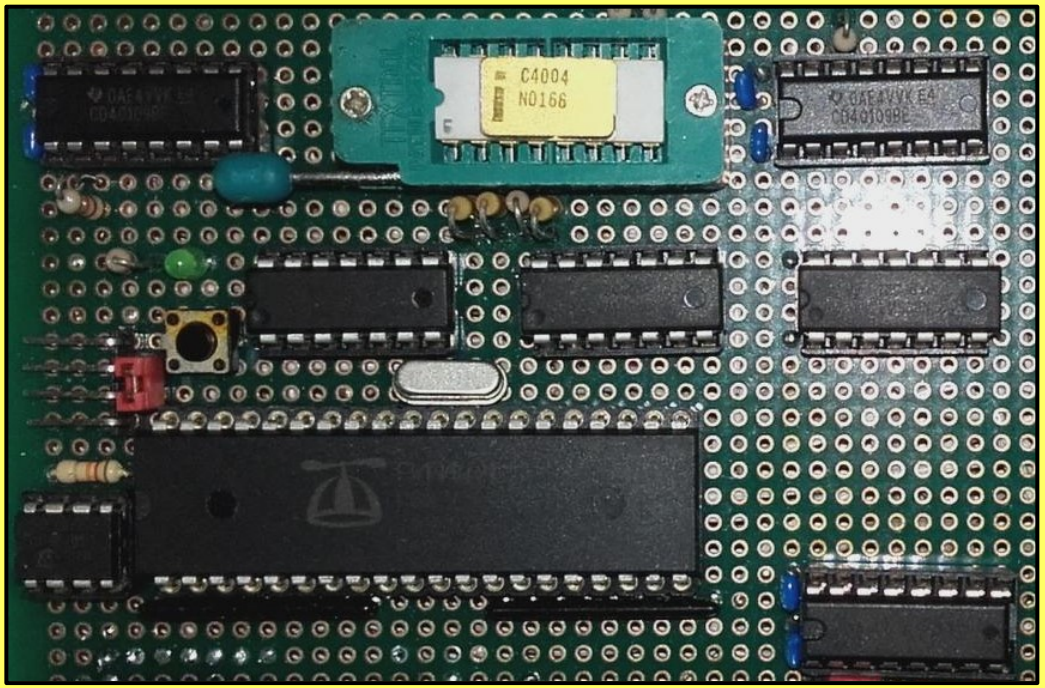
(1) IO instructions control the flow of information between accumulator in CPU, I/O lines in ROM's and RAM's and RAM storage. IOR stands for IO Read. In this case the CPU will receive data from RAM storage locations or I/O input lines of 4001's.

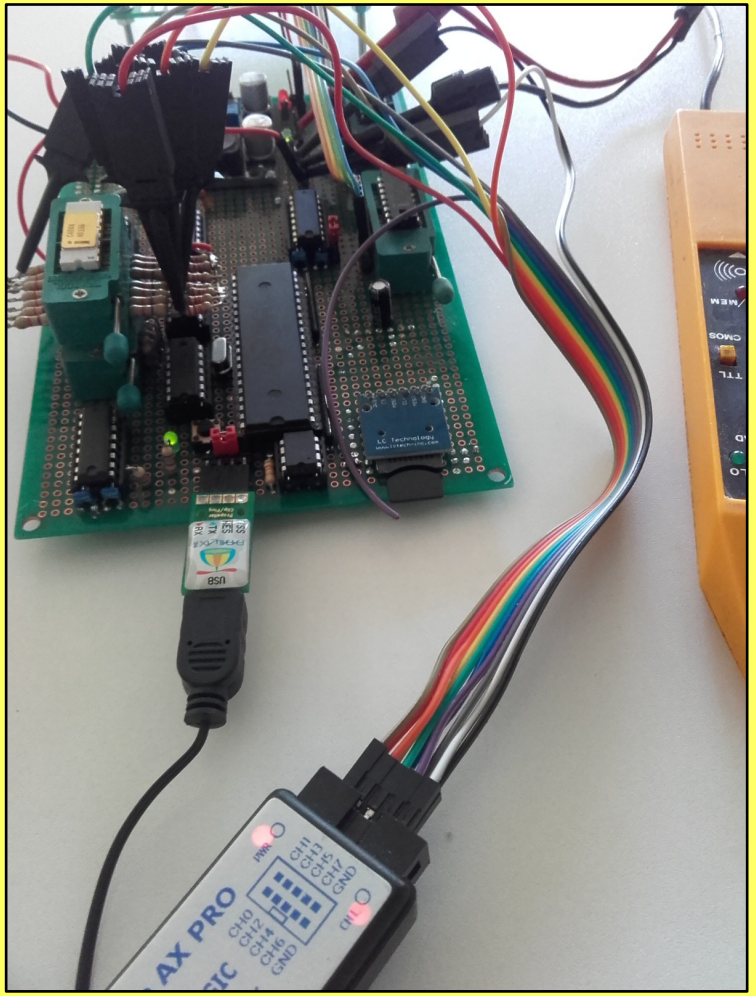
(2) The SRC instruction designates the chip number and address for a following IO instruction.

Figure 2. MCS-4 Basic Instruction Cycle

[-Mijn C4004 1-]

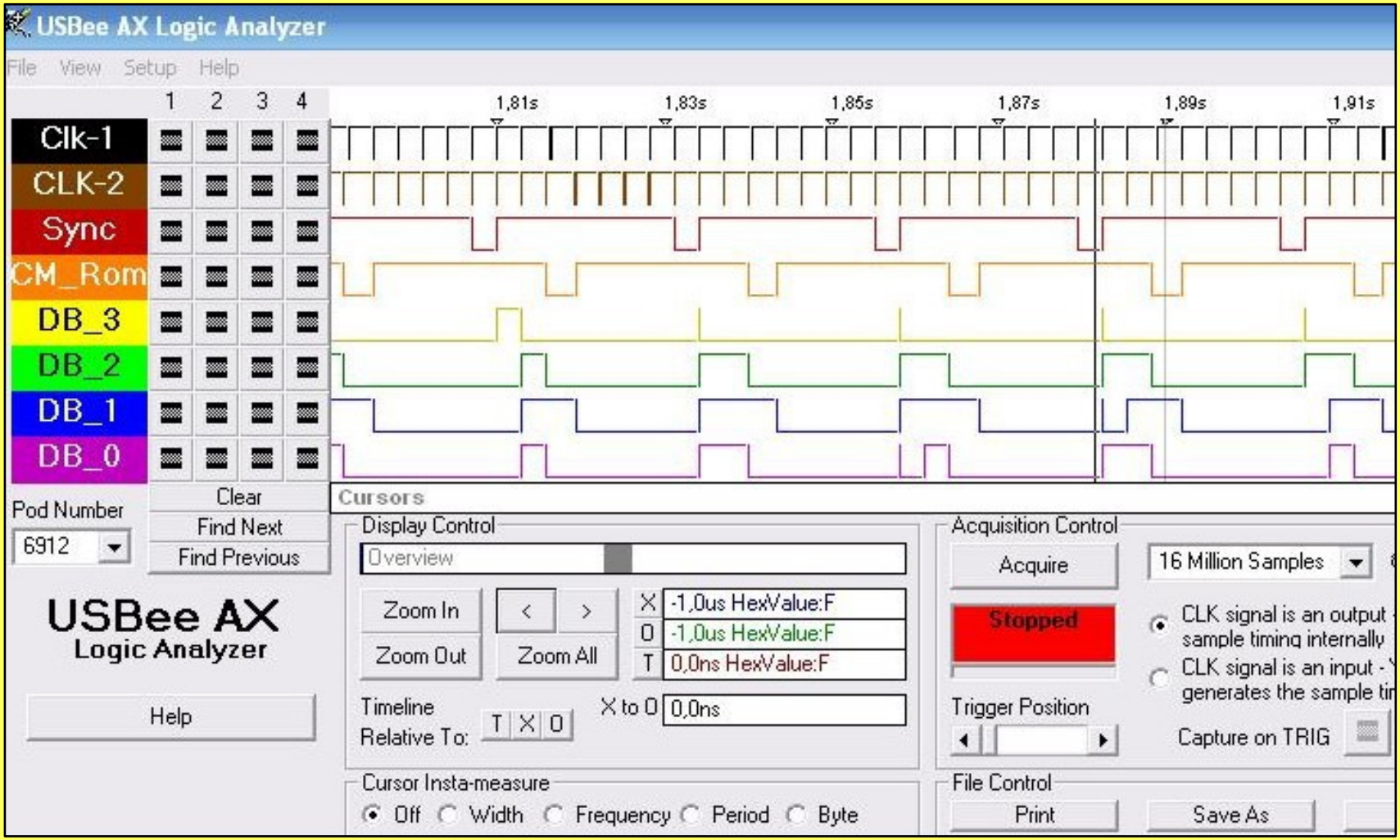
- Propellerchip als ROM en klok.





[-Mijn C4004 2-]

- Klok --> Sync resultaat



[-Mijn C4004 3-]

- NOP, NOP, NOP, NOP, Jmp naar 0

[Those instructions preceded by an asterisk (*) are 2 word instructions that occupy 2 successive locations in ROM]

MACHINE INSTRUCTIONS (Logic 1 = Low Voltage = Negative Voltage; Logic 0 = High Voltage = Ground)

MNEMONIC	OPR $D_3 D_2 D_1 D_0$	OPA $D_3 D_2 D_1 D_0$	DESCRIPTION OF OPERATION
NOP	0 0 0 0	0 0 0 0	No operation.
*JCN	0 0 0 1 $A_2 A_2 A_2 A_2$	$C_1 C_2 C_3 C_4$ $A_1 A_1 A_1 A_1$	Jump to ROM address $A_2 A_2 A_2 A_2$, $A_1 A_1 A_1 A_1$ (within the same ROM that contains this JCN instruction) if condition $C_1 C_2 C_3 C_4$ (1) is true, otherwise skip (go to the next instruction in sequence).
*FIM	0 0 1 0 $D_2 D_2 D_2 D_2$	R R R 0 $D_1 D_1 D_1 D_1$	Fetch immediate (direct) from ROM Data D_2, D_1 to index register pair location RRR. (2)
SRC	0 0 1 0	R R R 1	Send register control. Send the address (contents of index register pair RRR) to ROM and RAM at X_2 and X_3 time in the Instruction Cycle.
FIN	0 0 1 1	R R R 0	Fetch indirect from ROM. Send contents of index register pair location 0 out as an address. Data fetched is placed into register pair location RRR.
JIN	0 0 1 1	R R R 1	Jump indirect. Send contents of register pair RRR out as an address at A_1 and A_2 time in the Instruction Cycle.
*JUN	0 1 0 0 $A_2 A_2 A_2 A_2$	$A_3 A_3 A_3 A_3$ $A_1 A_1 A_1 A_1$	Jump unconditional to ROM address A_3, A_2, A_1.
*JMS	0 1 0 1 $A_2 A_2 A_2 A_2$	$A_3 A_3 A_3 A_3$ $A_1 A_1 A_1 A_1$	Jump to subroutine ROM address A_3, A_2, A_1, save old address. (Up 1 level in stack.)
INC	0 1 1 0	R R R R	Increment contents of register RRRR. (3)
*ISZ	0 1 1 1 $A_2 A_2 A_2 A_2$	R R R R $A_1 A_1 A_1 A_1$	Increment contents of register RRRR. Go to ROM address A_2, A_1 (within the same ROM that contains this ISZ instruction) if result $\neq 0$, otherwise skip (go to the next instruction in sequence).
ADD	1 0 0 0	R R R R	Add contents of register RRRR to accumulator with

[-Mijn C4004 3-]

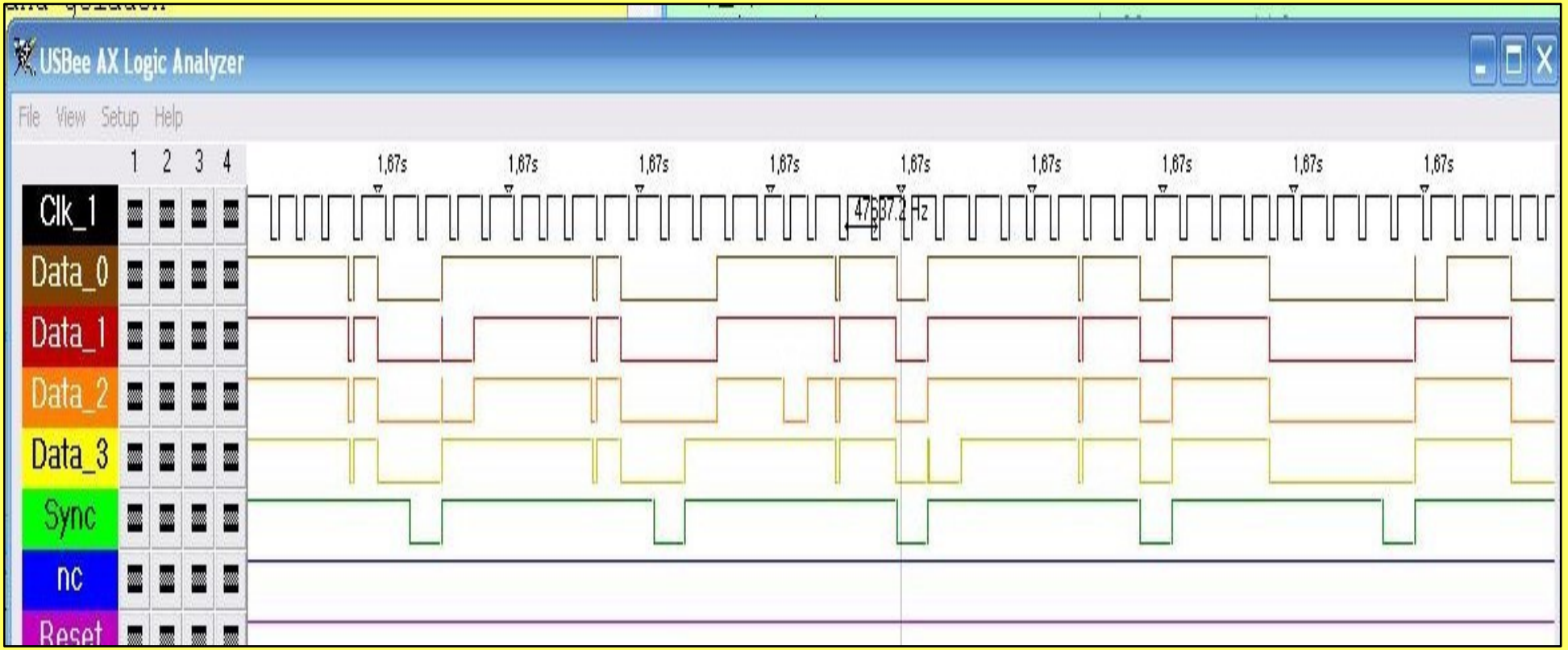
- NOP, NOP, NOP,
- NOP, Jmp naar 0

source code:

```
NOP
NOP
NOP
NOP
NOP
JUN $0000
```

object code:

```
00 00 00 00 00 40 00
```

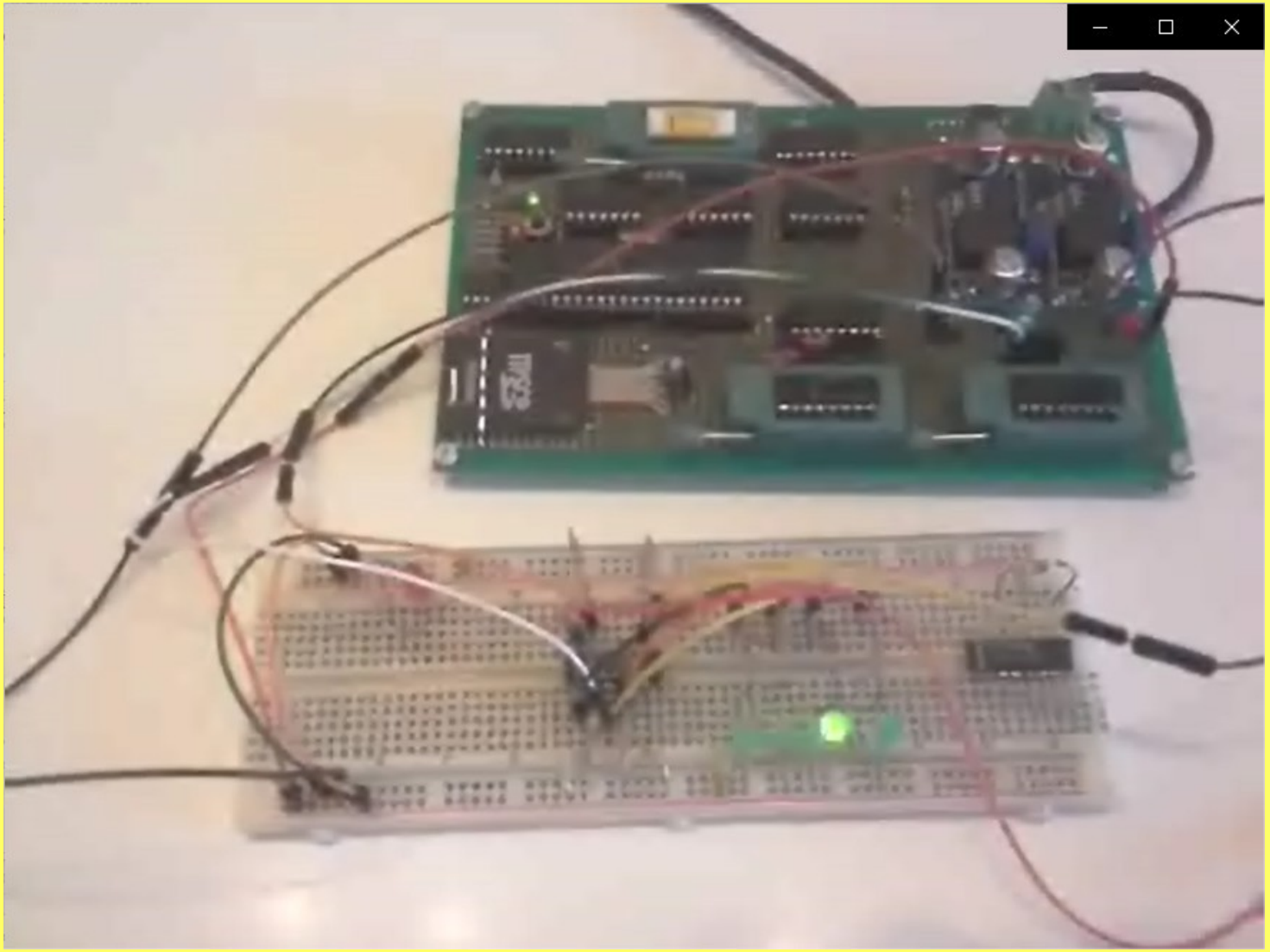


[-Mijn C4004 4-]

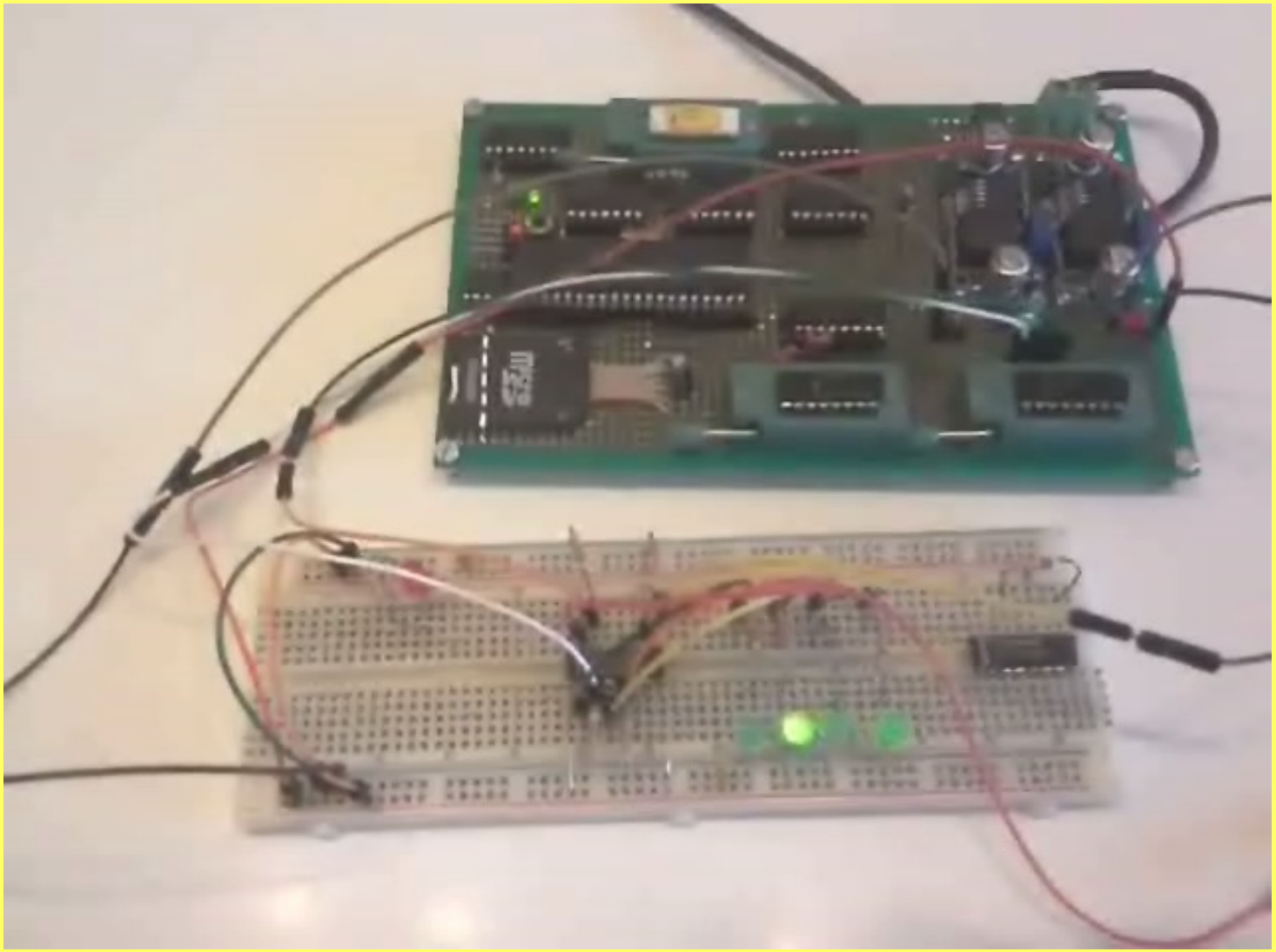
- RAM als uitgang

```
; Toggel output op RAM

START
        FIM R2R3, $40    ; RAM uitgang adres

BEGIN
        SRC R2R3         ; laad RAM adres
        LDM 0            ; laad accu met waarde
        WMP              ; zet op RAM-IO
        LDM 15           ; laad accu met waarde
        WMP              ; zet op RAM-IO
        JUN BEGIN
```

[-Mijn C4004 4-]



[-Mijn C4004 4-]



[-Mijn C4004 5-]

- ROM als ingang --> naar RAM

```
;       Lees ROM, schrijf naar RAM

START
        FIM R0R1, $10    ; ROM_1 adres
        FIM R2R3, $40    ; RAM   adres

BEGIN
        CLB                     ; Clear accu
        SRC R0R1                ;  R0R1 naar ROM/RAM
        RDR                     ; Lees input ROM in accu
        NOP                     ; Niets
        NOP
        SRC R2R3                ; R2R3 naar ROM/RAM
        WMP                     ; Schrijf accu naar RAM
        NOP
        NOP
        NOP
        JUN BEGIN               ; Spring naar begin
```

[-Mijn C4004 5-]

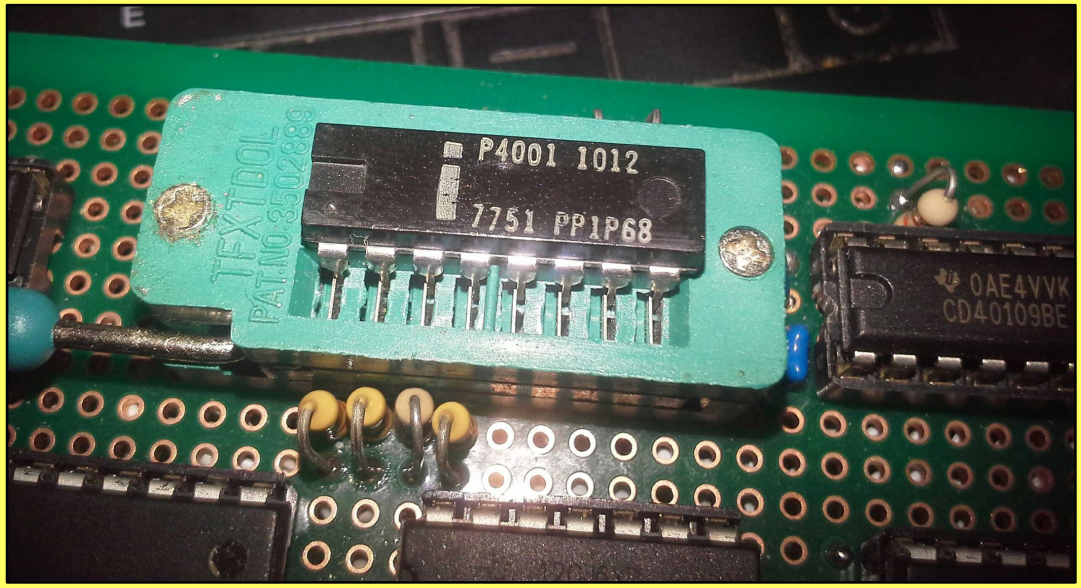
[-Mijn C4004 5-]

[-Mijn C4004 5-]

- ROM als ingang,

  problemen:

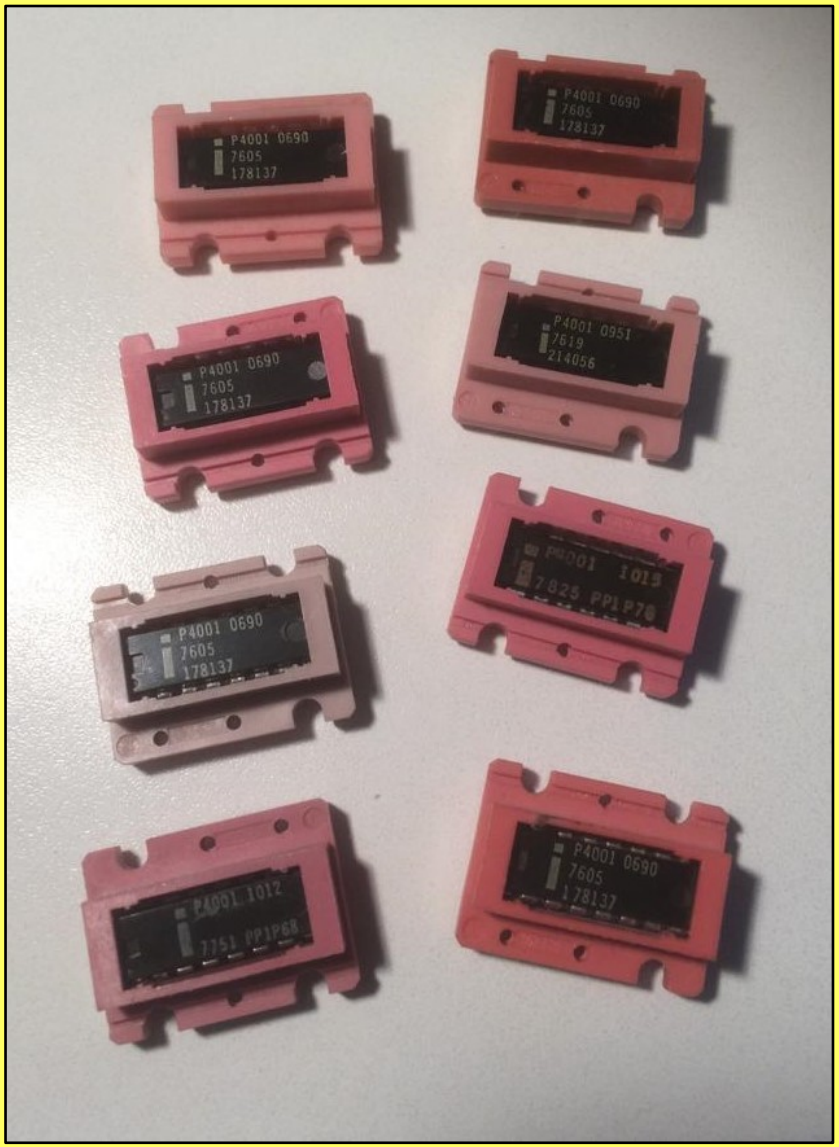
  - Adressering !!!!



1100	0000	0111	0000	0000
1101	0000	0111	0000	0000
1110	0000	0111	0000	0000
1111	0000	0111	0000	0000
0000	0000	1000	0100	1000
0001	0000	1000	0000	1101
0010	0000	1000	0000	0000
0011	0000	1000	0000	0000
0100	0000	1000	0100	1000
0101	0000	1000	0000	1101
0110	0000	1000	0000	0000
0111	0000	1000	0000	0000
1000	0000	1000	0100	1000
1001	0000	1000	0011	0000
1010	0000	1000	0000	0000
1011	0000	1000	0100	1000
1100	0000	1000	0011	0000
1101	0000	1000	1101	0001
1110	0000	1000	1111	1101
1111	0000	1000	0101	1000
0000	0000	1001	0000	0000

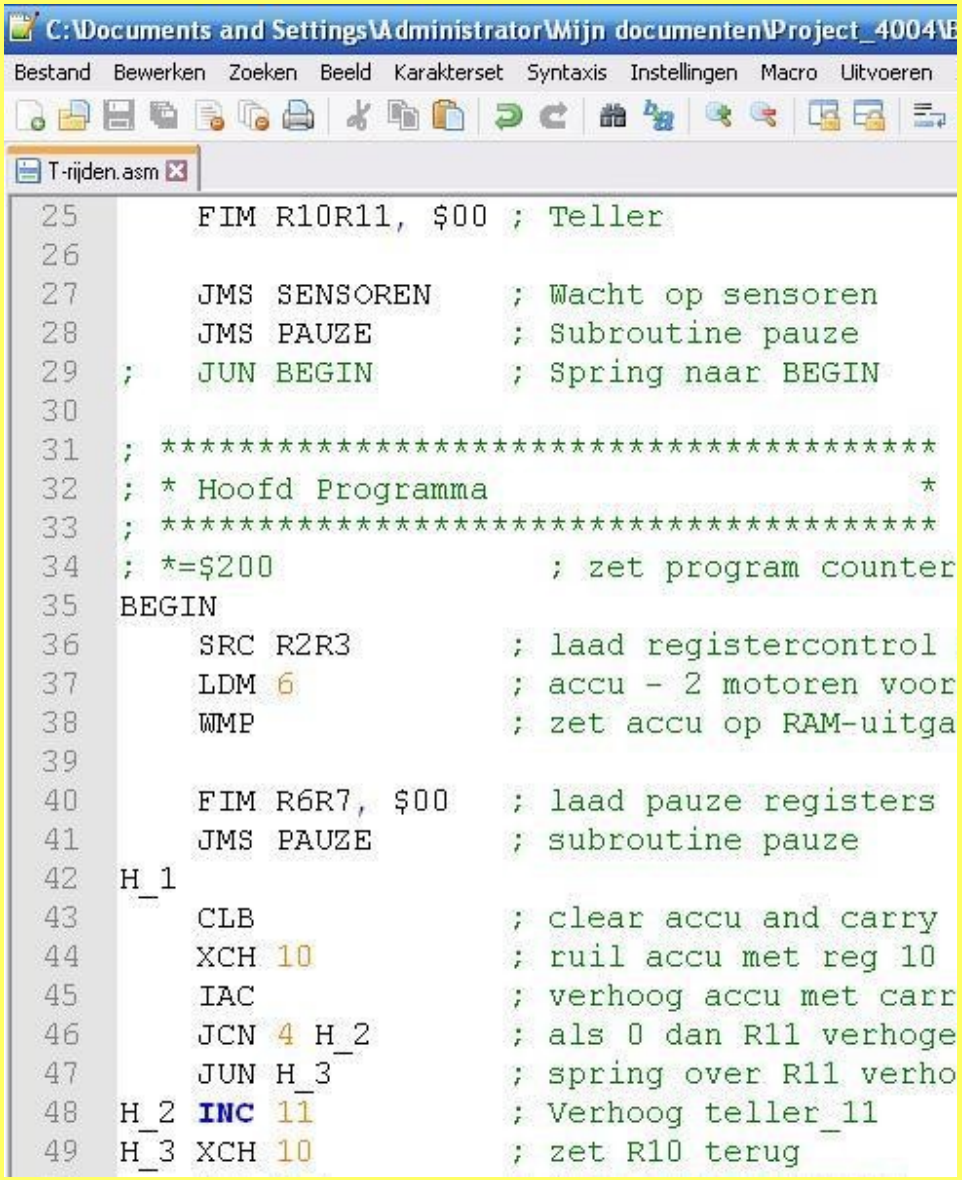
[-Mijn C4004 5-]

- 10 stuks gekocht
- NIET ROM-0
- En 4 ingangen
- Geen garantie !!



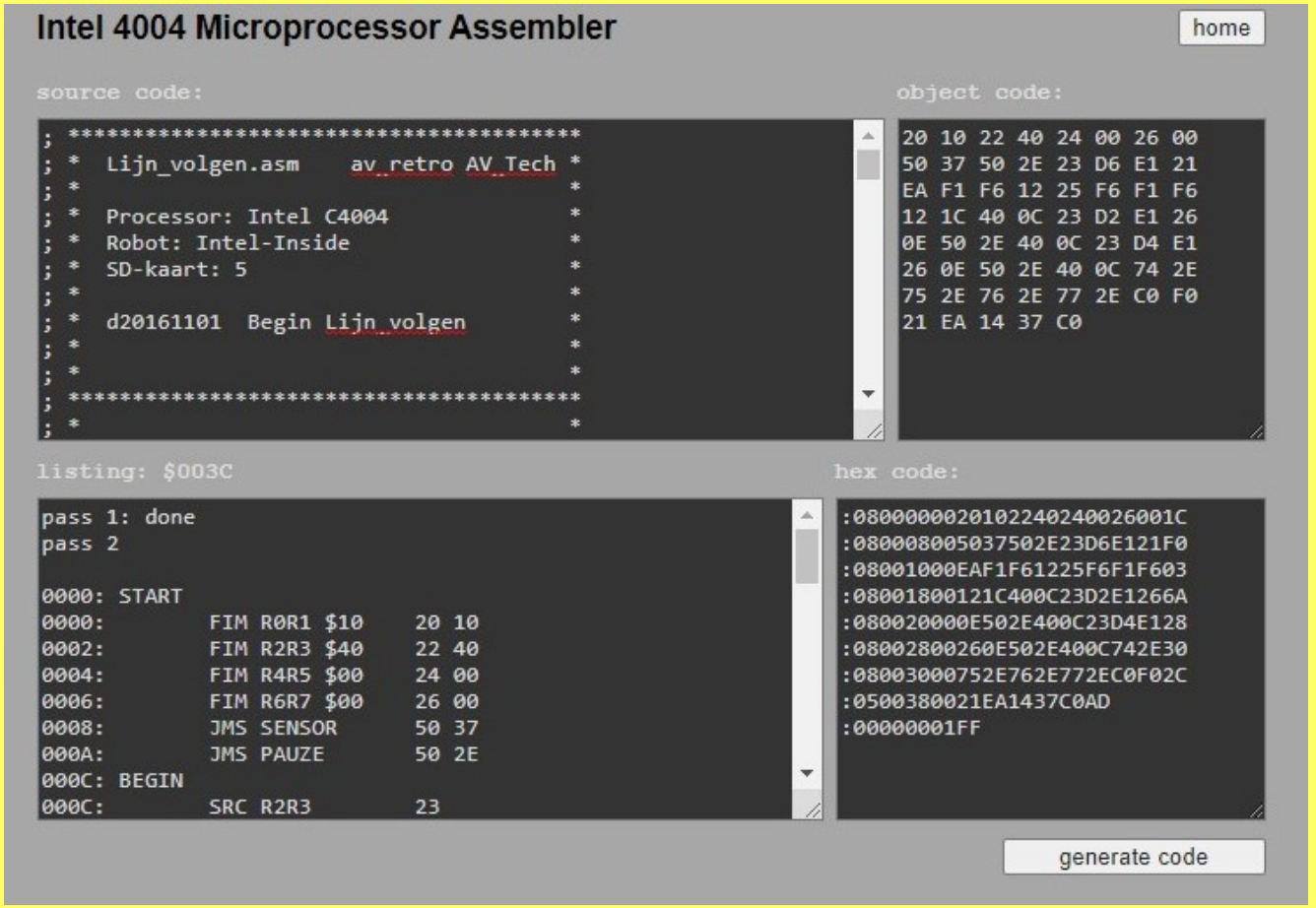
[-Mijn C4004 6-]

- Programma flow:
  - 1: Notepad, in assembler een programma maken



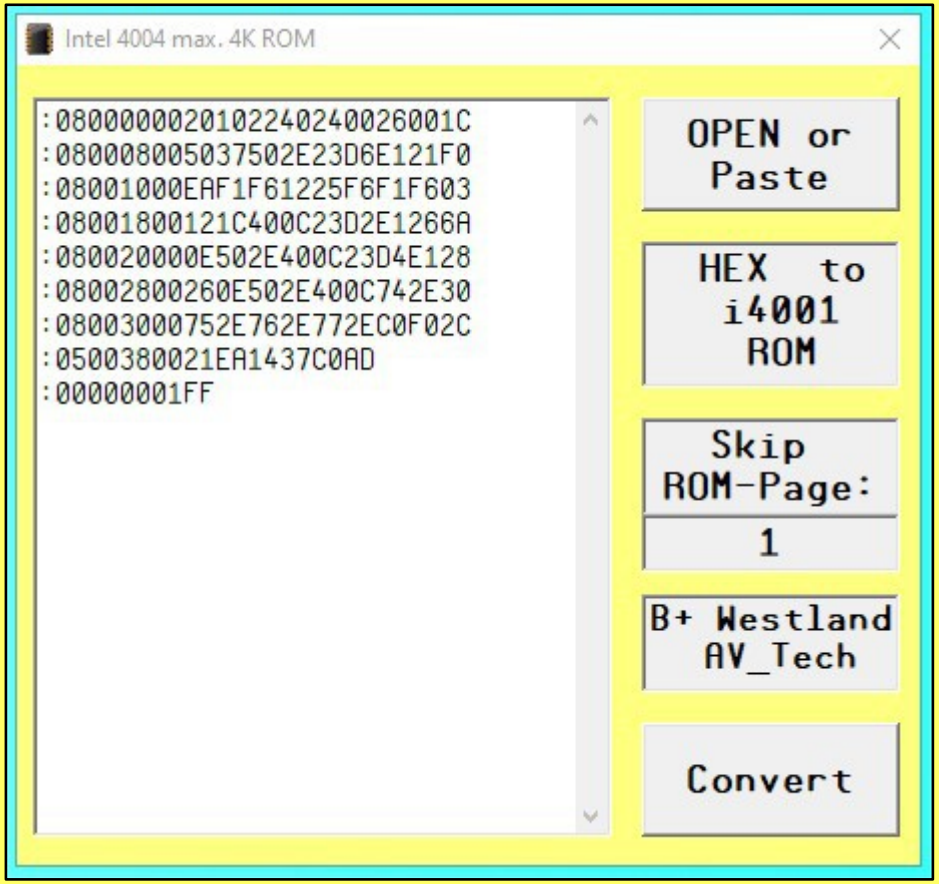
[-Mijn C4004 6-]

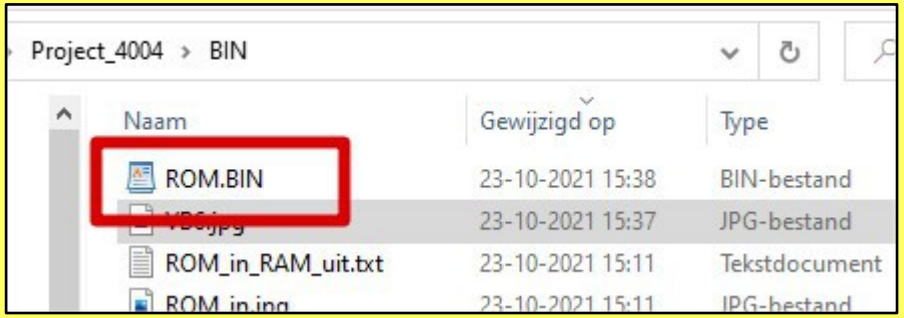
- Programma flow:
  - 2: Assembleren: e4004.szyc.org/asm.html



[-Mijn C4004 6-]

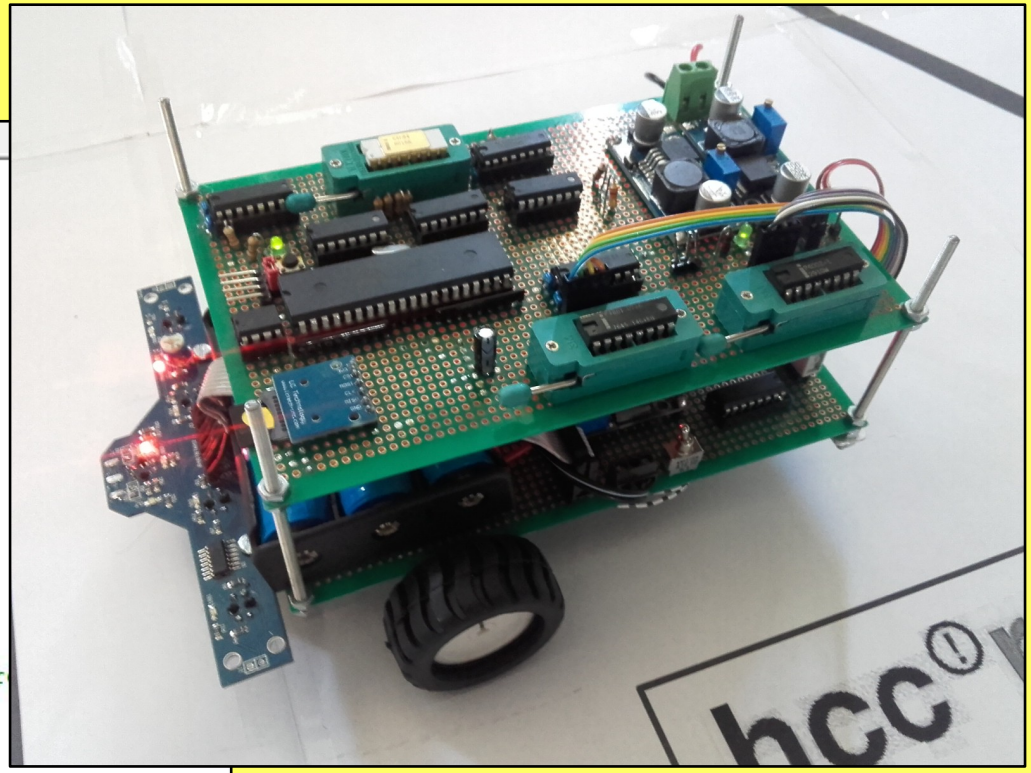
- Programma flow:
  - 3: Omzetten in binaire bestand via VB programma





[-Mijn C4004 7-]

- Resultaat: Lijnvolger.



Lijn_volgen.asm

```
16  ; ****************************************
17  ; * Initialisatie van Registers          *
18  ; ****************************************
19  START
20      FIM R0R1, $10    ; ROM-1 ingang adres
21      FIM R2R3, $40    ; RAM-1 uitgang adres
22      FIM R4R5, $00    ; PAUZE registers
23      FIM R6R7, $00    ; PAUZE registers
24
25      JMS SENSOREN     ; Wacht op sensoren
26      JMS PAUZE        ; Subroutine pauze
27
28  ; ****************************************
29  ; * Hoofd Programma                      *
30  ; ****************************************
31  BEGIN
32      SRC R2R3         ; laad registercontrol RAM-adr
33      LDM 6            ; accu - 2 motoren vooruit
34      WMP              ; zet accu op RAM-uitgang
35
36      SRC R0R1         ; laad ROM adres
37      RDR              ; Lees input ROM in accu
38      CLC              ; Clear carry
39      RAR              ; Rotate accu right door carry (1e bit)
40      JCN 2 H_LM       ; test op accu=0/c=1
41  ;                        ja -> jump -> lbl
42  ;                        2=carry, 4=accu
43      RAR              ; Rotate accu right door carry (2e bit)
44      CLC
45      RAR              ; Rotate accu right door carry (3e bit)
46      JCN 2 H_RM       ; test op accu=0/c=1
47      JUN BEGIN
```

[-Info intel-]

- Bron vermelding:
  - Intel - 4004 op internet
  - https://frank-buss.de/4004/
  - e4004.szyc.org/asm.html
  - https://avretro.wordpress.com/