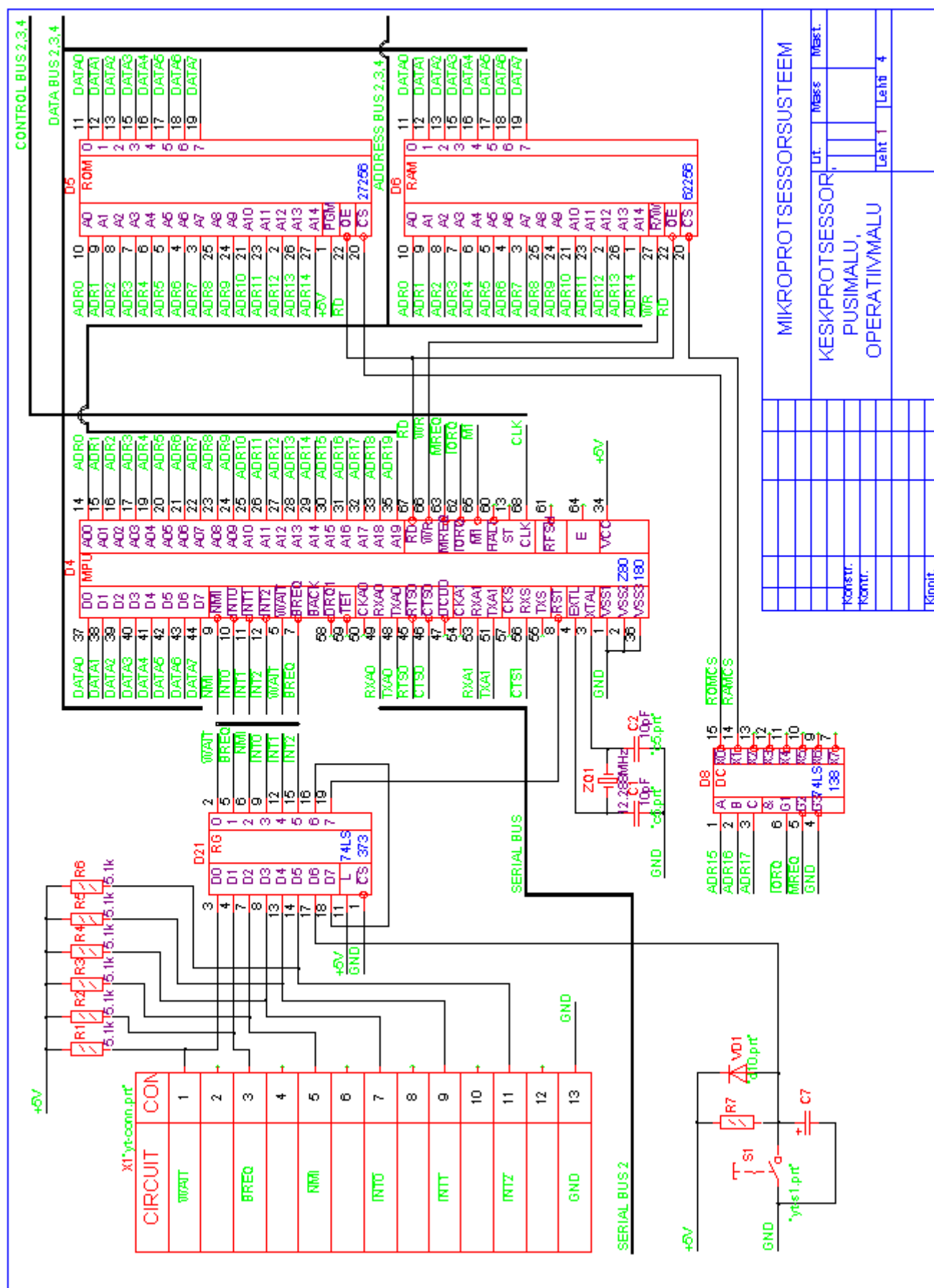
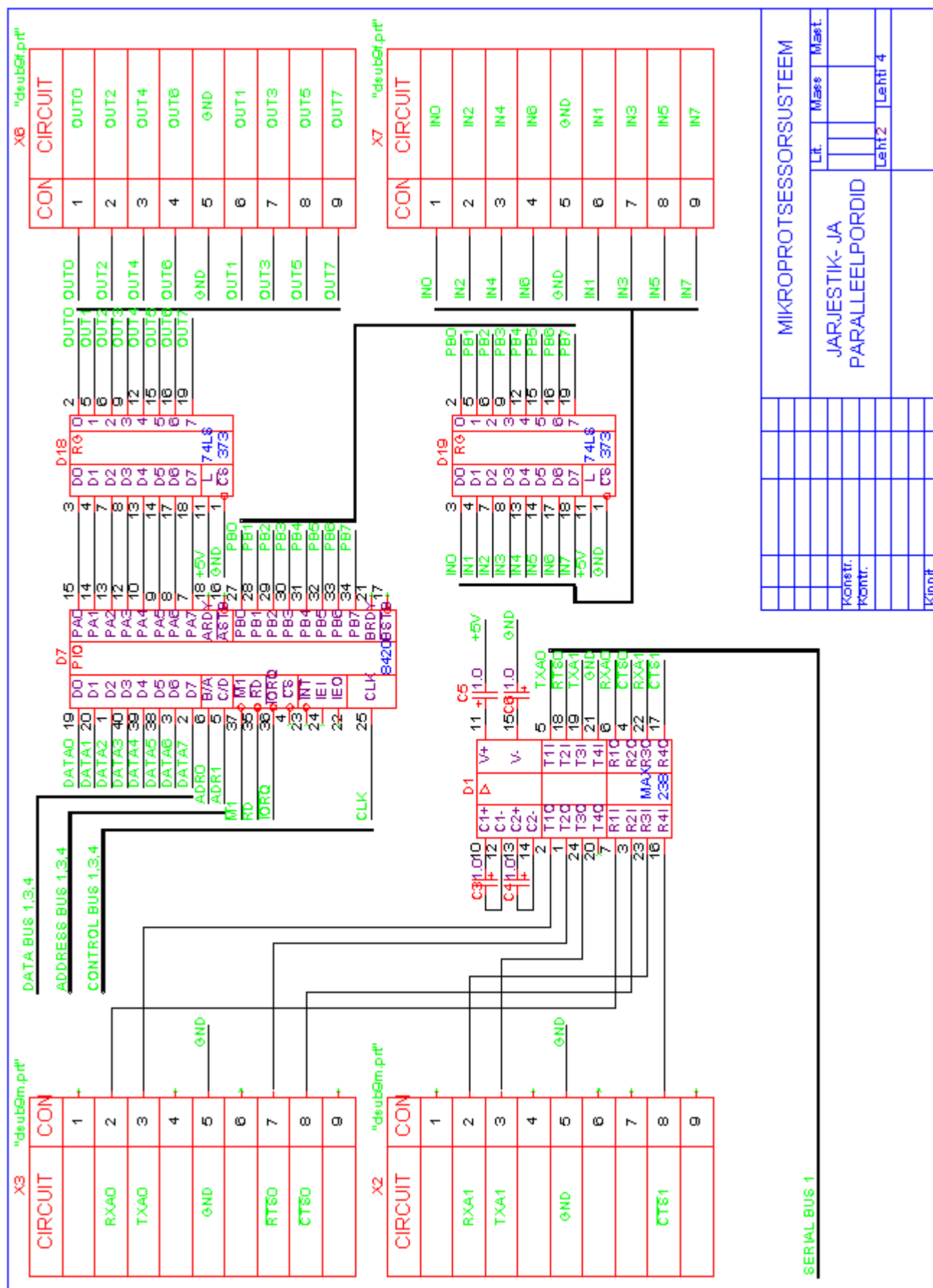


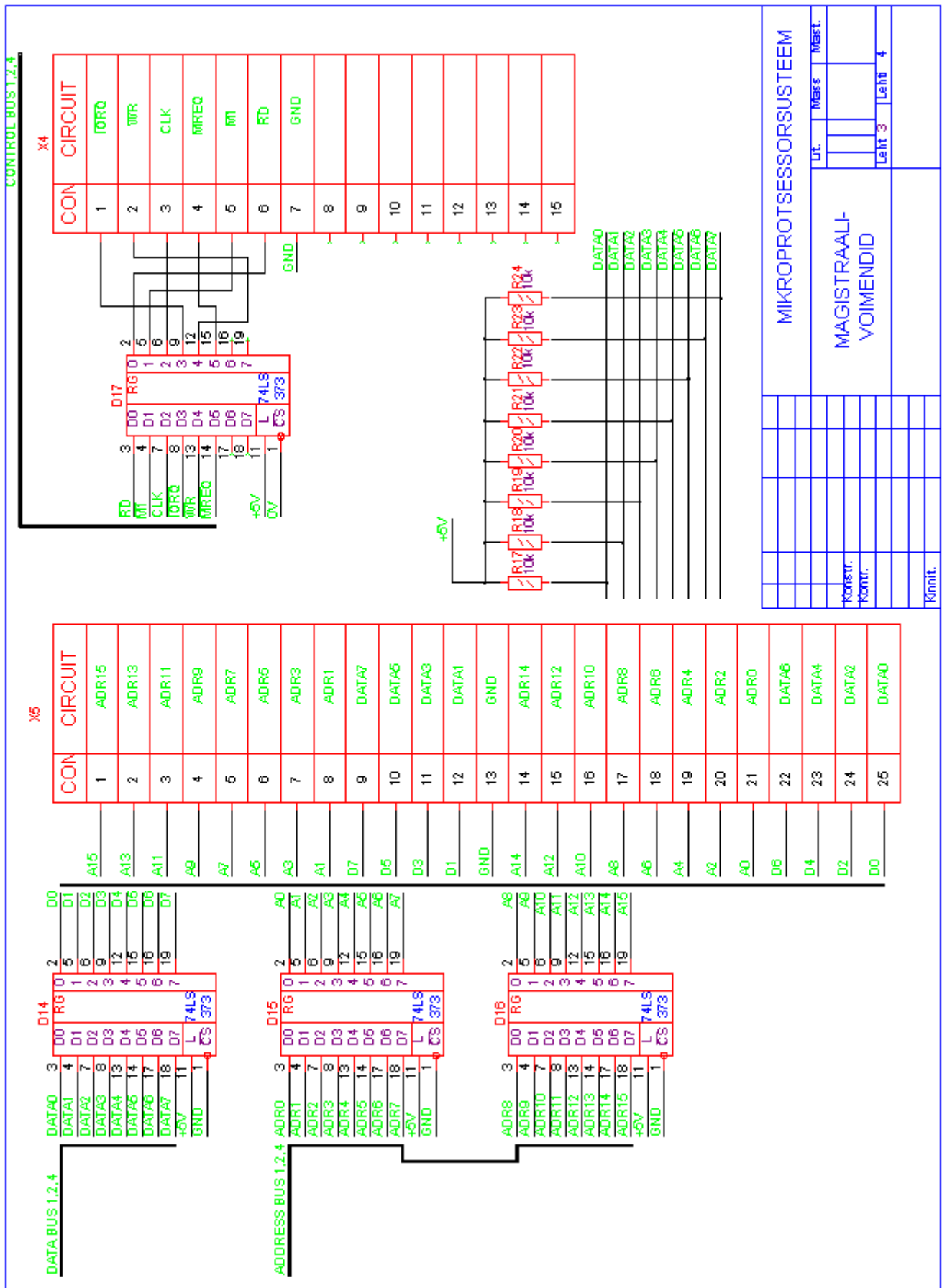
LISA 1

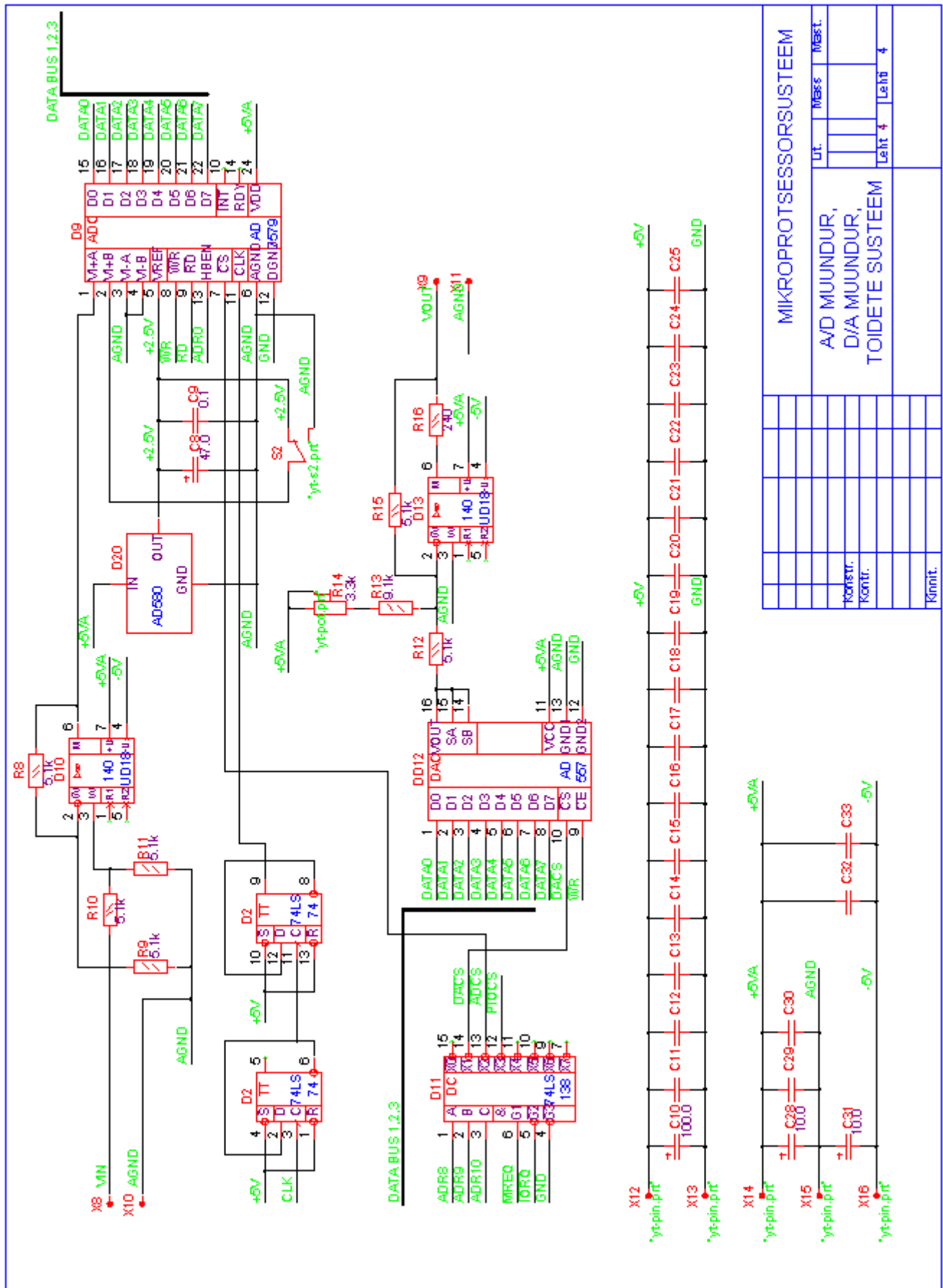
LKZ180 PÕHIMÖTTESKEEM





Heiki Kasemägi bakalaureusetöö: lisad



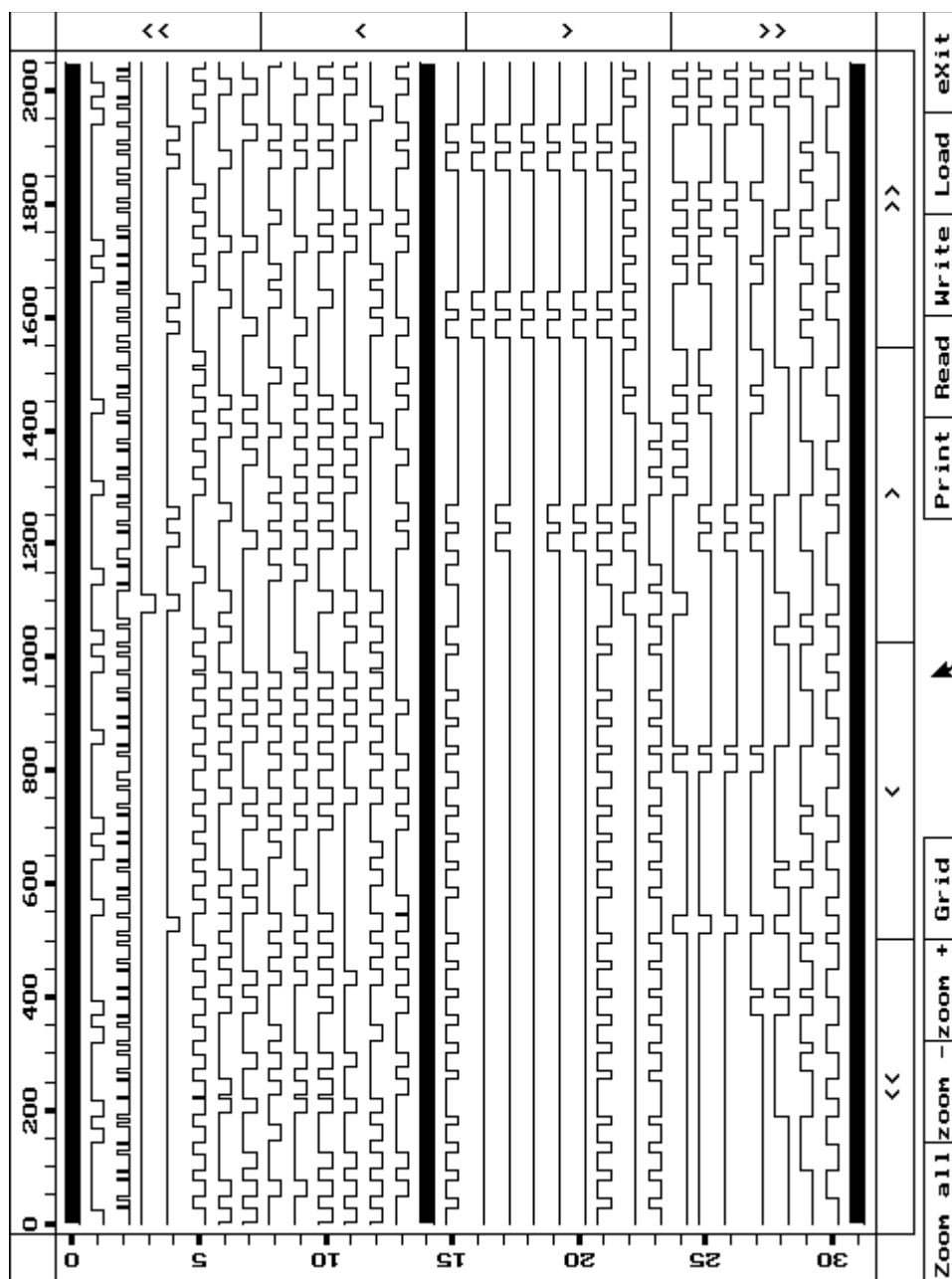


--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

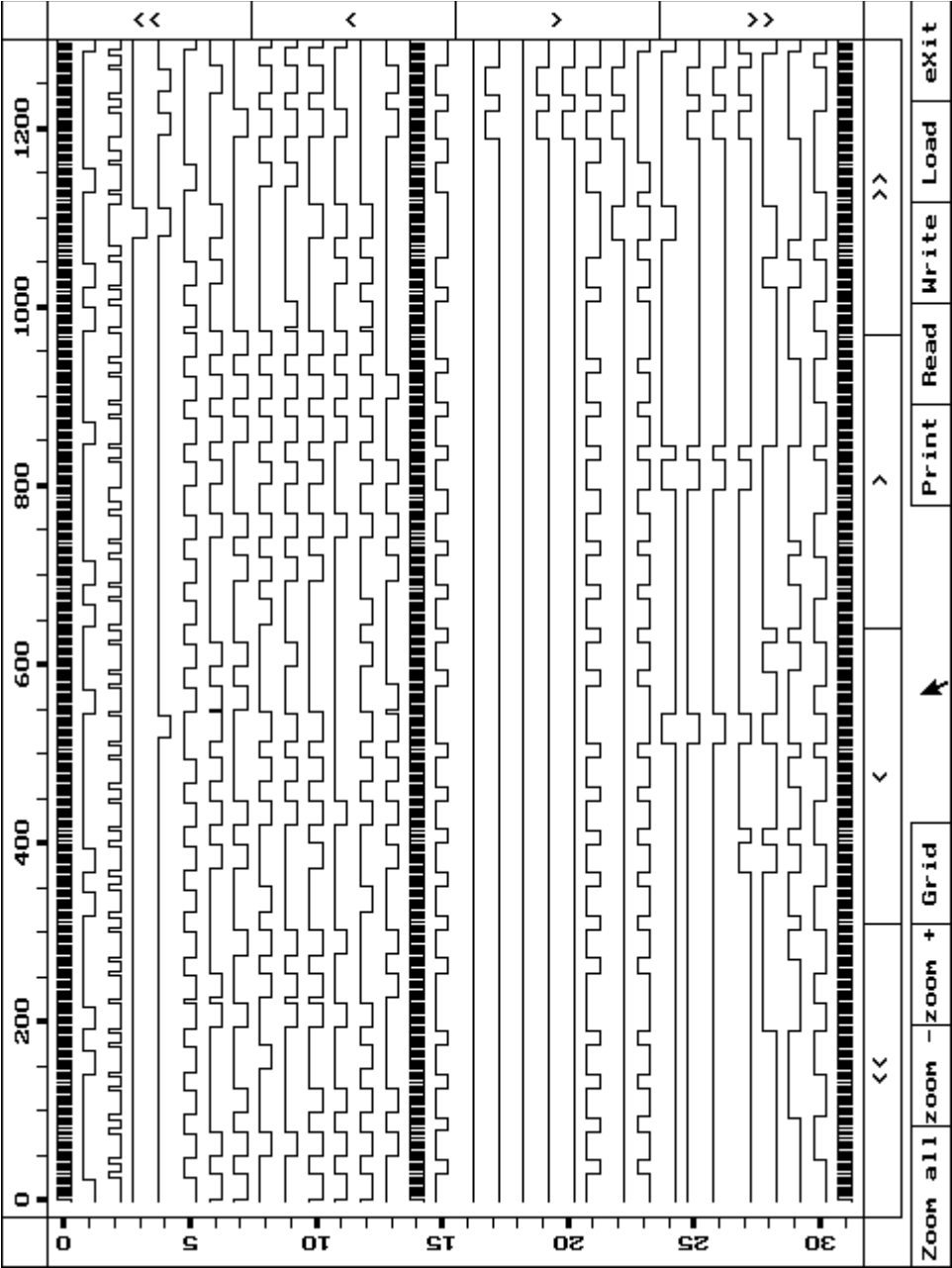
LISA 2

LOOGIKAANALÜSAATORI AJADIAGRAMMID

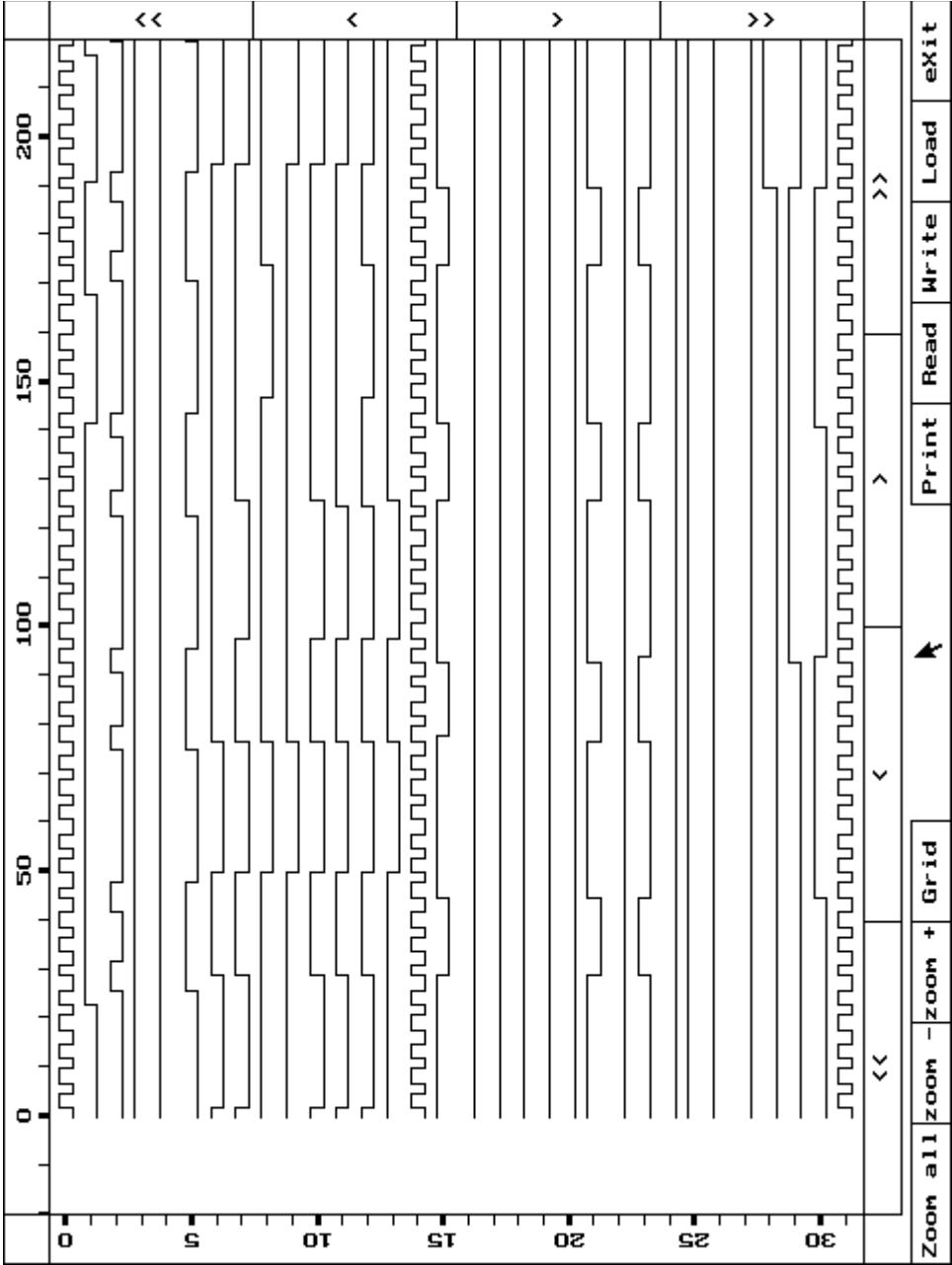
Heiki Kasemägi bakalaureusetöö: lisad



Ajadiagramm 1: Loogika-analüsaatori kõik 2048 lugemit ja 32 andmekanalit



Ajadiagramm 2: Signaalid siinidel ühe programmi täitmise ajal



Ajadiagramm 3: Eelmisel diagrammil oleva programmi esimesed käsud

LISA 3

MONITORPROGRAMMI ALAMPROGRAMMID

Heiki Kasemägi bakalaureusetöö: lisad

```
include lkz180.inc
;#####
;Autor: Toivo Vajakas
;#####
    aseq
    org    1200h
_cp_bc_hl:    ;unsigned compare
    ld     a,b    ;if BC<HL then Carry:=1 else Carry:=0
    cp     h    ;if BC=HL then Zero:=1 else Zero:=0
    ret     nz
    ld     a,c
    cp     l
    ret

;-----
;hl-de    contains 32-bit unsigned integer -- dividend
;bc    16-bit divisor
; returns de=result of divison
;    hl=remainder
;-----
_div16:    ld     a,l
    sub    c
    ld     a,h
    sbc    b
    ret    nc    ; return with no carry if overflow
    ld     a,b    ;bc:=-bc
    cpl
    ld     b,a
    ld     a,c
    cpl
    ld     c,a
    inc    bc
    call   loop
loop:    ld     a,d
    ld     d,e
    ld     e,8
loop1:    add    hl,hl
    jp     c,over
    add    a
    jp     nc,subtr
    inc    hl
subtr:    push    hl
    add    hl,bc
    jp     c,ok
    pop    hl
    dec    e
    jp     nz,loop1
    ld     e,a
    scf
    ret

ok:    inc    sp
    inc    sp
    inc    a
    dec    e
    jp     nz,loop1
    ld     e,a
    scf
    ret
```

Heiki Kasemägi bakalaureusetöö: lisad

```
over:  adc    a,a
       jp     nc,oversub
       inc    hl
oversub: add    hl,bc
       dec    e
       jp     nz,loop1
       ld     e,a
       scf
       ret
correction:
       ld     de,0    ;initial value
       ret     z      ;if "-"
       dec    hl
       ret
```

```
;-----
_mult16:    ;16bit by 16bit unsigned integer multiply
            ;operands in BC and DE, result in DEHL
            ;less then 781 clocks          Journal 'Electronics' 6/83
;-----
       ld     a,e
       call   bcmult
       push   hl
       push   af
       ld     a,d
       call   bcmue0
       ld     d,a
       pop    af
       add    a,h
       ld     h,l
       ld     l,e
       ld     e,a
       jp     nc,nc1
       inc    d
nc1:  pop     bc
       add    hl,bc
       ret    nc
       inc    de
       ret
bcmult:                                ;AHL=A*BC
       ld     e,0
bcmue0:  ld     h,e
       ld     l,e
       add    a,a
       jp     nc,zzz0
       add    hl,bc
       adc    e
zzz0:  add    hl,hl
       adc    a
       jp     nc,z1
       add    hl,bc
       adc    e
z1:    add    hl,hl
       adc    a
       jp     nc,z2
       add    hl,bc
       adc    e
z2:    add    hl,hl
       adc    a
       jp     nc,z3
       add    hl,bc
       adc    e
z3:    add    hl,hl
       adc    a
       jp     nc,z4
       add    hl,bc
       adc    e
```


Heiki Kasemägi bakalaureusetöö: lisad

```
z4:  add    hl,hl
      adc    a
      jp     nc,z5
      add    hl,bc
      adc    e

z5:  add    hl,hl
      adc    a
      jp     nc,z6
      add    hl,bc
      adc    e

z6:  add    hl,hl
      adc    a
      ret    nc
      add    hl,bc
      adc    e
      ret
```

```
;-----
; convert 16bit value into unsigned decimal 5-digit ASCII
; in: bc=highest digit address. Always 5 digits with leading 0s
; in: hl=value
; no registers affected
; uses 14 bytes of stack (incl ret addr)
;-----
```

```
utoa:
      push   bc
      push   af
      ld     a,'0'
      ld     (bc),a
      inc    bc
      ld     (bc),a
      inc    bc
      ld     (bc),a
      inc    bc
      ld     (bc),a
      inc    bc
      ld     (bc),a
      ld     a,10
      call   itoa
      pop    af
      pop    bc
      ret
```

```
_printx: ld a,16
          jp   printf
_printd:  ld   a,10
          jp   printf
_printb: ld a,2
          jp   printf
```

```
;-----
;printf:      ;send to ASC1 ASCII number
;in:hl=value,A=base
;-----
```

```
printf:
      ld     bc,prnbuf+prnbufen-1
      push   af
      ld     a,0
      ld     (bc),a      ;ASCIIZ terminator
      pop    af
      dec    bc
      call   itoa
      ld     h,b
      ld     l,c
      call   puts
      ret
```

Heiki Kasemägi bakalaureusetöö: lisad

```
;-----
; convert 16bit value into ASCII
; in: bc=lowest digit address, fills to left. Leading 0/blank not written
; in: hl=value
; in: a= 2,8,10 or 16 numeric system base
; out: bc=1st digit address
;-----
itoe: ld    (base),a
      push  af
      push  de
      push  hl
      ex    de,hl
cvloop: push  bc
      ld    hl,0
      ld    a,(base)
      ld    c,a
      ld    b,0
      call  div16
      pop   bc
      ld    a,1
      add   a,'0'

      cp    '9'+1
      jp    c,digiok
      add   a,'A'-'9'-1
digiok:
      ld    (bc),a
      dec   bc
      ld    a,e    ;test de=0 ?
      or    d
      jp    nz,cvloop
      pop   hl
      pop   de
      inc   bc
      pop   af
      ret

_scanfd: ld A,10
        call scanf1
        ret

_scanfx: ld A,16
        call scanf1
        ret

scanf1:
;-----
;convert string from ASCII input ASCII to integer
; in: A=base (2,8,10 or 16)
; leading whitespace allowed. Leading '+' or '-' sign allowed
; overflow not tested.
; out: HL=value
;-----
      ex    de,hl    ;DE=digit ptr
      ld    (base),a
      ld    hl,0      ;HL=value=0
      ld    a,0
      ld    (sign),a  ;sign=0

wsloop: ;wait for whitespace
      call  getcheu
      cp    ' '
      jp    z,wsloop
      cp    TAB
      jp    z,wsloop
      cp    '+'
      jp    z,digitloop
      cp    '-'
      jp    nz,gotdigit
      ld    a,1
      ld    (sign),a
```

Heiki Kasemägi bakalaureusetöö: lisad

```
digitloop:
    call    getcheu
gotdigit:
    sub     '0'
    jp      m,err
    cp      10
    jp      c,noadjust
    cp      'A'-'0'
    jp      c,err
    sub     'A'-'9'-1
noadjust:
    ld      b,a
    ld      a,(base)
    cp      b        ;if base<=digitvalue then error
    jp      c,err
    jp      z,err
    push    bc
    ld      b,h        ;bc=oldvalue
    ld      c,l
    call    bcmult      ; ahl := a*bc = base*oldvalue
    pop     bc
    ld      a,b        ;A=new digit
    add     a,1
    ld      l,a
    ld      a,0
    adc     a,h
    ld      h,a
    jp      digitloop
err:    ret

_delay:    ;delay microsec
    push    af
    push    bc
bcloop:
    dec     bc
    ld      a,b
    or      c
    jp      nz,bcloop
    pop     bc
    pop     af
    ret

;----- INPUT FROM CONSOLE -----
_kbhit:    ;test for console keyboard status
    ; if no char then A=0 ZF=1
    ; else A<>0, ZF=0
    in0     a,(stat1)
    and     rxrdy
    ret

_getch:    ;return A=byte from serial input, wait for char, no echo
    ; out: A=char
    call    kbhit
    jp      z,getch
    in0     a,(rdrl)
    ret

_getche:   ; return A=byte from console, wait for char, echo back
    call    getch
    call    putch
    ret

_toupper: ;converts char in A into uppercase
    cp      'a'
    ret     c
    cp      'z'+1
    ret     nc
    sub     'a'-'A'
    ret

_getcheu:  call    getche
    call    toupper
    ret
```

Heiki Kasemägi bakalaureusetöö: lisad

```
_putch:      ;wait until transmitter ready, send out 1 byte
             ;in: A=byte. No registers affected
             push bc
             push af
putc1loop:
             ld     bc,stat1
             in     a,(c)
             and    txrdy
             jp     z,putc1loop
             pop    af
             ld     bc,tdr1
             out    (c),a
             pop    bc
             ret
```

```
_puts:      ;in: hl=string address. sends data out.
             push   hl
puts1loop:
             ld     a,(hl)
             and    a
             jp     z,puts1end
             call   _putch
             inc    hl
             jp     puts1loop
puts1end:
             pop    hl
             ret
```

```
jpbase      equ    1000h
             org    jpbase+0h
             jp     _putch
             org    jpbase+8h
             jp     _puts
             org    jpbase+10h
             jp     _printd
             org    jpbase+18h
             jp     _printx
             org    jpbase+20h
             jp     _printb

             org    jpbase+28h
             jp     _getch
             org    jpbase+30h
             jp     _getche
             org    jpbase+38h
             jp     _getcheu
             org    jpbase+40h
             jp     _kbhit
             org    jpbase+48h
             jp     _scanf
             org    jpbase+50h
             jp     _scanfx

             org    jpbase+58h
             jp     _mult16
             org    jpbase+60h
             jp     _div16
             org    jpbase+68h
             jp     _cp_bc_hl
             org    jpbase+70h
             jp     _delay
             org    jpbase+78h
             jp     _toupper

             org    0f800h

base        ds     1
sign        ds     1
prnbuf      equ    16
prnbuf      ds     prnbuf
```

Heiki Kasemägi bakalaureusetöö: lisad

end

```
;standardalamprogrammid
jpbase      equ    01000h

putch equ    jpbase+0h
puts  equ    jpbase+8h
printd equ    jpbase+10h
printx equ    jpbase+18h
printb equ    jpbase+20h

getch equ    jpbase+28h
getche equ    jpbase+30h
getcheu equ    jpbase+38h
kbhit equ    jpbase+40h
scanf equ    jpbase+48h
scanfx equ    jpbase+50h

mult16 equ    jpbase+58h
div16 equ    jpbase+60h
cp_bc_hl equ    jpbase+68h
delay equ    jpbase+70h
toupper equ    jpbase+78h
```