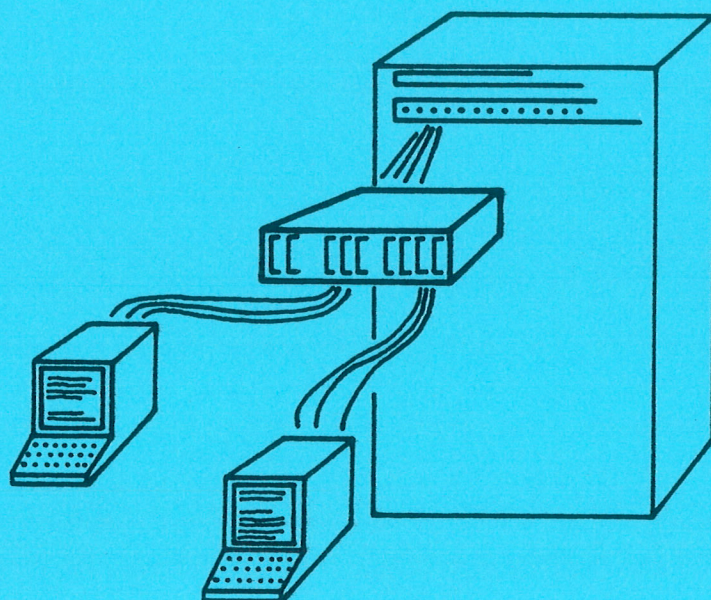


SEKTION VÄG- OCH VATTEN  
TEKNISKA HÖGSKOLAN I LUND

# VDC - VÄXEL

PER CHRISTIANSSON  
THORD LUNDGREN



SEPTEMBER 1981

## FÖRORD.

För att möjliggöra att ett större antal terminaler skulle kunna vara permanent anslutna till Nord-datorn vid Väg- och Vattens Datacentral, VDC, än de tillgängliga 20 terminalingångarna skissade författarna på en terminalväxel under våren 1981.

Under sommaren 1981 tillverkades en prototyp, Thord Lundgren, och skrevs ett styrprogram, Per Christiansson. Prototyp och styrprogram uttestades var för sig före slutlig hopkoppling gjordes.

Den i rapporten presenterade växeln med tillhörande programvara uppfyller de grundläggande krav som bör uppställas på en sådan enhet,

- kompakt utförande
- enkel funktion
- fysisk och funktionell utbyggbarhet

Arbetet är utfört vid Sektionen för Väg- och Vatten vid Tekniska Högskolan i Lund.

Lund i september

Per Christiansson  
Thord Lundgren

## INNEHÅLL

-----

1. Inledning
  
2. Växelns funktion
  - 2.1 Mot användaren
  - 2.2 Inkoppling till dator och terminal
  - 2.3 Styrprogram
  
3. Realtidsprogrammet VAXEL
  - 3.1 Programbeskrivning
  - 3.2 Tabell-fil
  - 3.3 Hantering av program
  
4. Växelns uppbyggnad
  - 4.1 Funktion
  - 4.2 Kommuikation med växel
  - 4.3 Layout

## APPENDIX

- A. RT-programmet vaxel
- B. Programmet VEM
- C. Tabellfilen vid VDC.
- D. Kretskortens funktion.

## 1. Inledning

Väg- och Vattens Datacentral finns beskriven i "Presentation av VDC" (Per Christiansson Januari 1981). Inom sektionen som är fysiskt samlad i en byggnad bedrivs undervisning och forskning. Datorn vid VDC användes både av teknologer och forskare via till anläggningen anslutna terminaler, 20 stycken. Antalet terminaler är emellertid större, 30-40 stycken. Några terminaler är anslutna till en så kallad ASK (terminalkoncentrator) vilken är direktansluten till Lunds Datacentral (Univac 1110/80). Nord-datorn vid VDC är även ansluten till LDC medförande att valfri VDC-ansluten terminal når LDC via ett kommando. Filer kan även flyttas i båda riktningar mellan VDC-datorn och LDC-datorn.

Vid ett par avdelningar finns manuella växlar (kopplingspaneler) med vars hjälp terminaler kan växlas till VDC.

Genom installationen av en ingångsväxel till VDC ökades utnyttjandegraden av befintliga terminalingångarna med en för användaren skenbar ökning av antalet ingångar och därmed tillgänglighet.

Till växeln kan endast anslutas terminaler med samma hastighet, för närvarande 2400 baud. De skrivande låghastighetsterminalerna växlas inte för närvarande.

Den i rapporten presenterade växeln kan givetvis anslutas till andra datorer än de av fabrikat Norsk Data. Exempelvis kan en mikrodator användas som styrdator. Det styrande programmet kan även utvidgas till att omfatta funktioner som hantering av terminaler med olika överföringshastighet och hantering av terminaler i kö för ledig dateringång.

## 2. Växelns funktion

### 2.1 Mot användaren.

=====

Användaren märker endast att hans/hennes terminal är växlad genom att meddelande om tilldelad nordingång eller om att ingen ingång finns tillgänglig erhålles, efter det att terminalens strömbrytare slagits till.

1. Terminalen slås på.

2. Om ledig ingång finns erhålles följande meddelande,

```
DU ÄR NU KOPPLAD MOT
NORDINGÅNG NUMMER
      NN           öåö
```

Efter meddelandet skrives ett par tecken ut som ej betyder något (användes för styra växeln).

Om ingen nordingång är ledig skrives följande meddelande ut,

```
TYVÄRR FINNS INGEN LEDIG NORDINGÅNG FÖR
NÄRVARANDE FÖRSÖK OM EN STUND IGEN
*****
```

Slå av terminalen och försök igen om en stund.

3. Efter avslutad körning loggar användaren ut och slår av terminalen.

Genom att ge kommandot VEM till systemet erhålles besked om aktuell status på de till växeln anslutna nordingångarna, exempelvis med följande utseende

LOG.NO	TERM.NO	PLACERING	
52	2	ELITE-9	
53	1	ELITE-8	FAST KOPPLAD
7	3	BT2-1	
48	LEDIG		
osv.			

LOG.NO överensstämmer med de nummer som erhålles vid kommando WHO eller TE-ST till systemet.

## 2.2 Inkoppling till dator och terminal.

=====

Växeln kan hantera upp till 62 terminaler och 15 nord-ingångar. Det elektriska gränssnittet är strömslinga (current loop) men kan modifieras till RS232.

Terminaler kan fritt kopplas ur och anslutas utan att störa växeln i drift. Om antalet tillgängliga nordingångar skall ändras måste vissa parametrar ändras i den så kallade tabellfilen. Varefter styrprogrammet måste återstartas, vilket ej påverkar de terminaler som är inkopplade till Norddatorn.

Växeln styrs via en in/ut port till styrdator. I beskriven installation är Norddatorn styrdator.

Växeln styres genom att information sändes om vilka terminal-ingångar som skall kopplas till vilka nordingångar. Som svar på ett visst styrtecken sänder växeln information om vilka terminaler som är påslagna. ( Se vidare kapitlet Kommunikation med växel).

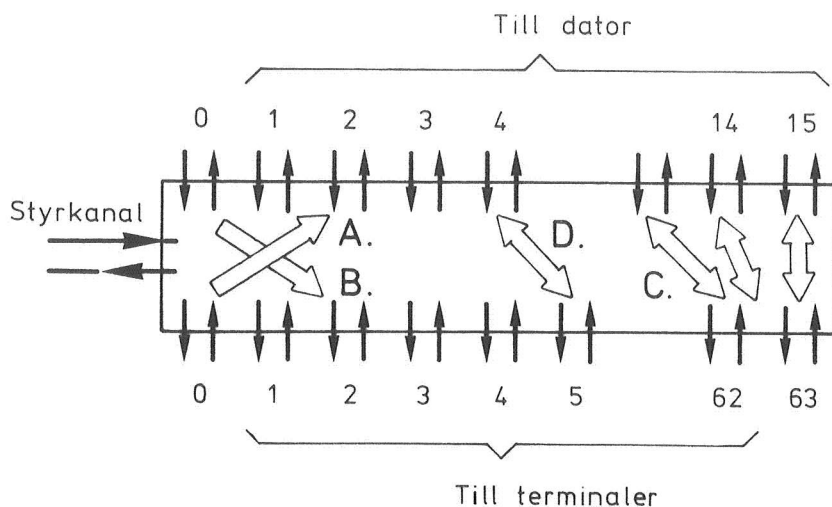
Växeln reagerar endast på tre olika styrtecken, vilket medför att tecken andra än dessa kan sändas via styrledningen utan att påverka växeln. Detta utnyttjas då meddelande sändes till terminal (A i figur 2-1) samt då utloggningssekvens sändes

till nordingång (B i figur 2-1).

Vid meddelande till terminal kopplas således nordingång 0 temporärt ihop med aktuell terminalingång (B) och då en utloggningssekvens skall sändas till en nordingång kopplas denna temporärt ihop med terminalingång 0 (A).

Då en nordingång ej användes kopplas den till terminalingång nummer 63 (C i figur 2-1). Flera nordingångar kan på detta sätt vara vilokopplade till terminalingång 63.

I figuren är även visad en normal hopkoppling av terminal nummer 5 med nordingång nummer 4 (D i figur 2-1).



Figur 2-1 Scematisk beskrivning av växel.

### 2.3 Styrprogram.

=====

Växeln styres av ett realtidsprogram som ligger i norddatorn (se Appendix A). Programmet läser av alla terminalingångarna var 3:e sekund. Om någon förändring skett sedan förra avläsningen, någon terminal har blivit på- eller avslagen, göres försök att koppla in terminalen eller kopplas förbindelsen ner i det senare fallet.

Om någon förändring skett i uppkopplingar uppdateras en speciell tabellfil VAXEL-TAB:SYMB.

Om ett strömavbrott inträffar sörjer en batteribackup för att inga koppel förändras (under max 1/2 timme).

Om återstart av datorsystemet av någon anledning måste göras startas realtidsprogrammet automatiskt och uppkoppling göres enligt informationen i tabellfilen. Terminaler som eventuellt slagits av under uppehållet kopplas givetvis ner efter nästa 3 sekunders intervall.

När en terminal slagits av sändes en utloggningsssekvens till associerad nordingång innan fysisk nerkoppling göres.



### 3. Realtidsprogrammet VAXEL

#### 3.1 Programbeskrivning.

=====

Programmet finns listat i Appendix A.

Programmet är skrivet i Fortran. Huvudprogrammet omfattar ungefär 40 rader och utgöres för övrigt av subrutiner, ungefär 180 rader.

Programmets flöde beskrives översiktligt enligt,

1. Reservera styrkanal (ID).
2. Läs tabellfilen.  
Gör uppkoppling enligt denna, CALL SEND.
3. Vänta i 3 sekunder (label 10)  
Har terminaler slagits på eller av, CALL COMPARE.
4. Ingen förändring, gå till 3.
5. Om terminalen är fast kopplad eller  
alla terminaler kontrollerade gå till 8.
6. En terminal har slagits av.  
Lagra information om att terminalen skall kopplas ner.  
Sänd utloggningssekvens.
7. En terminal har slagits på, CALL CUP.  
Sänd meddelande om tilldelad nordingång eller  
meddela att alla ingångar upptagna.  
Lagra information om att terminalen skall  
kopplas upp.
8. Gör fysisk uppkoppling. CALL SEND.  
Uppdatera tabellfilen.  
Gå till 3.

Programmet innehåller följande subrutiner,

SEND sänd uppkopplingsinformation till växeln.  
TERMST läs av terminalingångarna (på eller av).  
COMPARE kontrollera om terminaler slagits på eller av.  
UTLOG sänd utloggningssekvens till nordingång.  
SENDMESS sänd meddelande till terminal.  
WTAB läs tabellfilen.  
RTAB uppdatera tabellfilen.  
CUP koppla upp terminal.

Följande CALL-satser är specifika för norddatorer,

CALL RESRV(ID,0,1) reservera ID för inmatning  
CALL RESRV(ID,1,1) och utmatning  
CALL CIBUF(ID) clear input buffer  
CALL COBUF(ID) clear output buffer  
CALL HOLD(3,2) vänta i 3 sekunder  
CALL OUTCH(ID,RS) sänd ett tecken  
CH=INCH(ID) läs ett tecken  
IT=IAND(ITE,K2) logiskt OCH

### 3.2 Tabellfil.

=====

Kommentarer är i efterhand inlagda till höger om tabellen,

0	0	0	0	0	Nordingång 0 för meddelande
1	-1	52	2400	2	Nordingång 1 till växel
2	1	53	2400	1	Nordingång 2 fast mot term 1
3	-1	7	2400	3	Nordingång 3
4	-1	48	2400	63	Nordingång 4
5	-2	44	2400	63	Ej kopplad mot Nord
6	-2	45	2400	63	Ej kopplad mot Nord
7	-2	0	2400	63	.
8	-2	0	2400	63	.
9	-2	0	2400	63	.
10	-2	0	2400	63	.
11	-2	0	2400	63	.
12	-2	0	2400	63	.
13	-2	0	2400	63	.
14	-2	0	2400	63	.
15	-2	0	2400	63	.
0	-2	0	2400	-1	OUTLOGG Terminal 0
1	2	0	2400	-1	OELITE 9 Terminal 1
2	-1	0	2400	1	OELITE 8
3	-1	3	2400	3	OBT2-1
4	-1	0	2400	-1	OBS-LAB-1
5	-1	0	2400	-1	OBT2-2
6	-1	0	2400	-1	OBS-2
7	-2	0	2400	-1	OXXX Information ej
8	-2	0	2400	-1	OXXX införd om var
9	-2	0	2400	-1	OXXX terminalerna finns
.					

61	-2	0 2400	-1	OXXX
62	-2	0 2400	-1	OXXX
63	-2	0 2400	-1	OFRINORD

I tabellen ovan utgör de första 15 raderna växelns utgångar mot norddatorn och de följande 64 raderna (alla ej listade) växelns utgångar mot terminalerna (jämför även med figur 2-1). Nordingång 0 och terminalingång 0 användes internt i växeln för meddelande till terminal respektive utloggningssekvens till nordingång. Till terminalingång 63 kopplas de nordingångar som för tillfället ej användes.

Tabellens del 1 lagras i fältet N i programmet VAXEL och tabellens del 2 i fälten T och SPEC.

De 15 första raderna beskriver dateringångar (kolumnvis),

kol 1.	kol 2.	kol 3.	kol 4.	kol 5.
Nord- ingång	Fast status	Nord- port	Hast. (baud)	Variabel status
-----	-----	-----	-----	-----
nr. 0- 15	-3=kass -2=ej inkopp- lad	logiskt ND-num- mer	använ- des ej 2400	0
	1-62 = fast kop- plad mot terminal nr			kopplad mot term. 1-62
	-1=anslu- sluten			63=Nordingång tillgänglig

De följande 64 raderna beskriver de olika terminalingångarna (kolumnvis),

kol 1.	kol 2.	kol 3.	kol 4.	kol 5.	kol 6.	kol 7.
Term. nr.	Fast Status	primär ND-in-gång	hast. baud	Vari-abel Status	Kö-plats	Pla-cering

-----	-----	-----	-----	-----	-----	-----
0-63	1-15= fast kopp-lad	1-15 0 för övrigt	anv. ej 2400	-1=ej påslagen	anv. ej 0	text

kopp-lad mot ND-port 1-15.

0=påslagen men ej kopplad, (ingen ledig ingång).

Om en terminal önskas fast ansluten till en nordingång måste tre parametrar ges speciella värden, kolumn 2 och 5 i tabellens del 1 samt kolumn 2 i tabellens del 2.

Terminalens placering anges i kolumn 7 i tabellens del 2 (fältet SPEC i programmet VAXEL).

Med nordingång avses numret på den port i växeln till vilken doringången är ansluten. Operativsystemets numrering av ingångarna återfinnes i kolumn 3 i del 1 av tabellen.

Det är möjligt att göra ändringar i tabellen även då växeln är i drift, se nästa kapitel.

För att öka sannolikheten för att en terminal skall kopplas till en angiven primär nordringång (kol 3, tabelldel 2) bör sådana datorringångar kopplas in i slutet av tillgängliga nordringångar mot växeln (högt nummer i kol 1 tabelldel 1). (Se även subrutinen CUP i styrprogrammet VAXEL).

### 3.3 Hantering av program.

=====

Realtidsprogrammet kompileras och laddas på segmentfil med följande kommandosekvens, vilken lämpligtvis lagras i en MODE-fil (kommandofil),

```
@FTN
COM VAXEL-RT,0,VAXEL-RT
EX
@RT-LO
CLEAR-SEG 200
Y
NEW-SEG 200,,,,,
SET-SEGM-FILE 2
NREE-LO VAXEL-RT,,,,,
END-LO
EXIT
```

Om man vid återstart av norddatorn (MASTER CLEAR, LOAD) exekverar ett batchjob bör detta innehålla ovanstående sekvens, dock ej kompileringsdelen (de tre första raderna), följt av kommandot RT VAXEL, för att en automatisk återstart av växelprogrammet skall erhållas.

Det är möjligt att göra ändringar i konfiguration under drift. RT-programmet stoppas med ABORT VAXEL. Med hjälp av QED (editorn) göres ändringar i tabellfilen VAXEL-TAB:S. Programmet startas därefter med kommandot RT VAXEL. Följande

sekvens erhålles således,

```
ABORT VAXEL
QED VA-TAB
D1      (ta bort första radens LF)
MTO(0)
gör ändringar
W VA-TAB
EX
RT VAXEL
```

Det är lämpligt att lagra en tabellfil, filen VA-YTAB:S nedan, med endast läsåtkomst, innehållande information om grundkonfigurationen, vilken kan användas om man av någon anledning förstör VAXEL-TAB. Observera att första raden i VAXEL-YTAB skall vara blank.

En frisk version av VAXEL-TAB skapas på följande sätt,

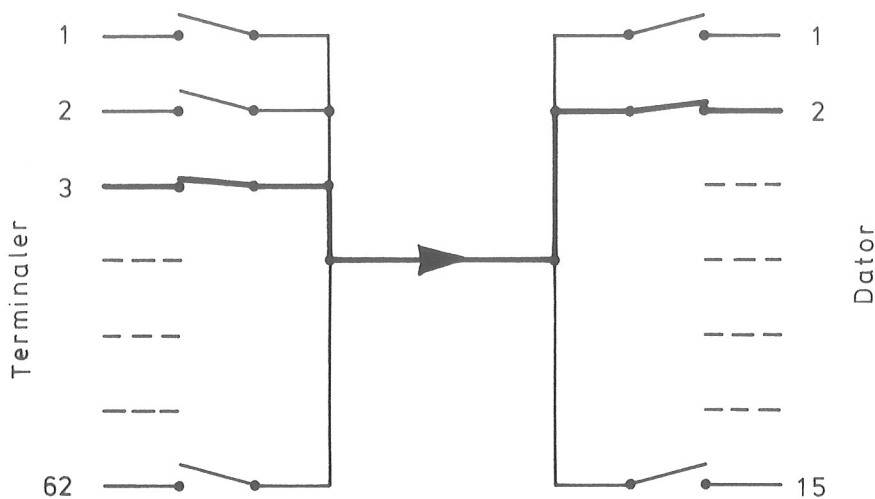
```
QED VA-YTAB
MTO(0)
gör ev. ändringar
W VA-TAB      skriv tillbaks på arbetsfilen.
EX
```

#### 4. Växelns uppbyggnad.

##### 4.1 Funktion.

=====

Växeln arbetar efter principen tidsmultiplex d.v.s. att terminalingångarna avsöks och kopplas samman med aktuell datorutgång under en kort tidsperiod (1.3 micro sek/koppel). Under mellantiden sörjer en kondensator för att ingen förändring av signalen sker på utgångarna från växeln. Växeln utför 768000 koppel/sek detta innebär, vid överföringshastigheten 2400 bit/sek (mellan terminal och dator), att 20 koppel utföres för varje bit. Växeln har testats och fungerar väl med överföringshastighet upp till 4800 bit/sek men bör modifieras något för att fungera med överföringshastigheten 9600 bit/sek.



Figur 4-1 Schematisk skiss av signalöverföring i en riktning via växel, från terminal till dator.



Växeln strömförsörjes från det inbyggda kraftaggregatet som lämnar 5 volt 0.2 ampere till logiken och +/-12 volt 2 ampere till strömslingorna. Ett laddningsbart batteri försörjer logiken vid kortare strömavbrott (30min) så att inga koppel förändras oavsiktligt.

#### 4.2 Kommunikation med växel.

=====

Växelns interna räknare nollställs genom att styrtecknet RS (reset), 124 (174 oktalt=ö) sändes till växeln. (RS återfinnes i programmet VAXEL liksom SB och TS).

Ett koppel sättes upp genom att först styrtecknet SB (send byte), 123 (173 oktalt=å) sändes och därefter terminalnummer för nordingång 0, terminalnummer för nordingång nummer 1 osv. (maximalt 16 tecken), kolumn 5 i tabellens del 1 (kolumn 5 i fältet N i programmet VAXEL). Sekvensen bör avslutas med resettecknet RS. Se även subrutinen SEND i VAXEL.

Om tecknet TS (term status), 123 (175 oktalt=ä) sändes till växeln svarar denna med att sända tillbaka 16 tecken där de 4 minst sigifikanta bits i varje tecken markerar om de 4 terminalingångarna (n+3, n+2, n+1, n) är på- eller avslagna (1 resp. 0). Se även subrutinen TERMST i VAXEL.

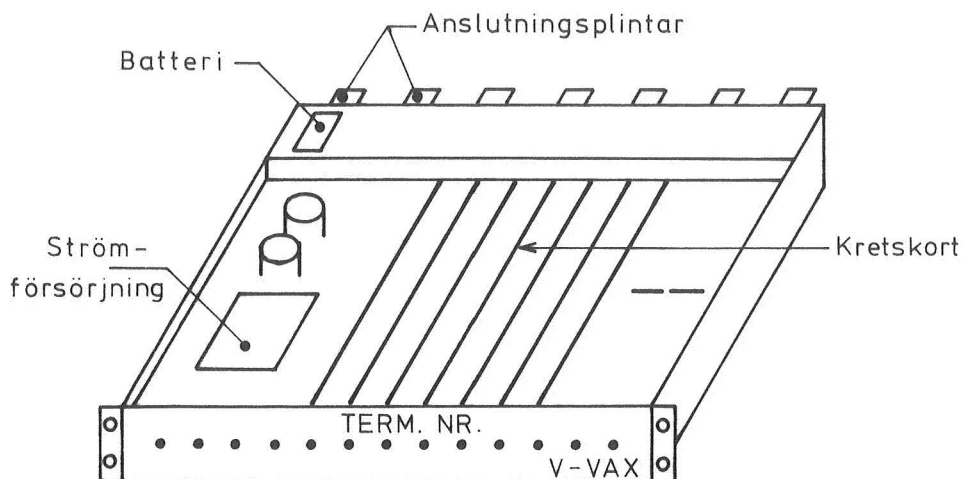
### 4.3 Layout.

=====

Växeln är uppbyggd i en standard 19 tums kortram med höjden 132 mm och djupet 340 mm. Kretskortens storlek är av litet europakortsformat 100\*160 mm. Apparaten är modulärt uppbyggd, vilket möjliggör valfri bestyckning av antalet in- och utgångar, dock max 62 terminalingångar och 15 datorutgångar.

Vid fullt utbyggd apparat ingår följande kort:

1 st V-UART	Kommunikationskort
1 st V-MEM	Minneskort
4 st V-VAX	Datorkort
16 st V-VAX	Terminalkort



Figur 4-2 Växelns uppbyggnad.

APPENDIX

=====

- A. RT-programmet VAXEL
- B. Programmet VEM
- C. Tabellfilen vid VDC.
- D. Kretskortens funktion.

A. RT-programmet vaxel.

=====

PROGRAM VAXEL,30

```
C *****
C * RT PROGRAMMET VAXEL. PER CHRISTIANSSON 16.8.1981      *
C *****
C * PROGRAMMET ANROPAR TABELLFILEN VAXEL-TAB:SYMB, SOM    *
C * INNEHÅLLER INFORMATION OM 16 NORDINGÅNGAR OCH 64     *
C * TERMINALINGÅNGAR, VILKA ÄR ANSLUTNA TILL VÄXELN.     *
C * DÅ EN TERMINAL SLÅS PÅ REGISTRERAS DETTA AV PROGR.   *
C * SOM CHECKAR VÄXELN VAR 3:E SEKUND. OM LEDIG NORDINGÅNG*
C * FINNS KOPPLAS TERMINALEN TILL DENNA, DOCK I FÖRSTA  *
C * HAND TILL EV. ANGIVEN PRIMÄR NORDINGÅNG ( ENLIGT    *
C * FILEN VAXEL-TAB:SYMB). DÅ EN TERMINAL SLÅS AV      *
C * FRIGÖRES NORDINGÅNGEN, SÅVIDA DEN EJ ÄR MARKERAD SOM *
C * FAST I VAXEL-TAB:SYMB. INNAN FYSISK FRIKOPPLING AV  *
C * NORDINGÅNGEN GÖRES SÄNDES FÖR SÄKERHETS SKULL EN UT- *
C * LOGGNINGSSEKVENNS IN PÅ NORDINGÅNGEN (GÄLLER EJ FASTA *
C * ANSLUTNINGAR).                                     *
C * GENOM ATT GE KOMMANDOT VEM ERHÅLLES INFORMATION OM  *
C * STATUS PÅ TILL VÄXELN ANSLUTNA NORDINGÅNGAR (GÖRES  *
C * AV SPECIELLT PROGRAM). VID INLOGGNINGEN ERHÅLLES   *
C * DESSUTOM BESKED OM VILKEN NORDINGÅNG SOM VÄXELN TILL- *
C * DELAT TERMINALEN.                                   *
C * NORDINGÅNGARNAS NUMMER ÄR DESAMMA SOM DE SOM LISTAS *
C * MED KOMMANDOT WHO ELLER TERMINAL-STATUS.           *
C * RT-PROGRAMMET VAXEL KRÄVER UNGEFÄR 10KILOORD UTRYMME. *
C *
C * VÄXELN MED TILLHÖRANDE PROGRAM FINNS BESKRIVNA I   *
C * RAPPORTEN "VDC-VÄXEL" AV PER CHRISTIANSSON OCH THORD *
C * LUNDGREN.                                           *
C *****
```

CHARACTER MESS(3)\*50, SPEC(0:63)\*10

```

        INTEGER RS,SB,TS,T63,QUE1
        INTEGER TOFN(0:63),DIFF(0:63)
        INTEGER N(0:15,1:5),T(0:63,1:6)

C
C      RS=Ö,SB=Å,TS=Ä
        RS=174B
        SB=175B
        TS=173B
        T63=63
        QUE1=0
        ID=71B
        CALL RESRV(ID,0,1)
        CALL RESRV(ID,1,1)
C ***** OPEN SATS I.S.F RESRV OM EJ RT KÖRNING
C      OPEN(70B,FILE='IO70',ACCESS='RW')
        CALL CIBUF(ID)
        CALL COBUF(ID)
C ***** LÄS TABELL OCH GÖR UPPKOPPLING ENLIGT DENNA
        CALL RTAB(N,T,SPEC)
        CALL SEND(N,RS,SB,ID)
C ***** KOLLA OM TERMINALER BLIVIT AV- ELLER PÅKOPPLADE
10      CALL HOLD(3,2)
        CALL COMPARE(DIFF,T,RS,TS,ID,TOFN)
C ***** INGEN ÅTGÄRD OM DIFF=0 ELLER FAST KOPPLING
        DO 100 I1=1,62
        IF (DIFF(I1).NE.0.AND.T(I1,2).LE.0) GOTO 200
100     CONTINUE
        GOTO 10

C
C ***** NÅGON TERMINAL HAR BLIVIT AV- ELLER PÅKOPPLAD
C
200     DO 300 I2=I1,62
C
C ***** OM DIFF=0 ELLER FAST UPPKOPPLING GÅ TILL 300

```

```

C
      IF (DIFF(I2).EQ.0.OR.T(I2,2).GE.1) GOTO 300
C
C ***** OM DIFF=-1, KOPPLA NER TERMINALEN (=1 KOPPLA UPP)
C
      IF (DIFF(I2).EQ.1) GOTO 250
C ----- KOPPLA NER I TABELLEN -----
      CALL UTLOG(I2,N,T,RS,SB,ID)
      N(T(I2,5),5)=63
      T(I2,5)=-1
      GOTO 300
250      CONTINUE
C
C ***** KOPPLA UPP TERMINAL
      CALL CUP(I2,QUE1,N,T,RS,SB,T63,ID)
300      CONTINUE
C
C ***** GÖR FYSISK UPPKOPPLING
C
      CALL SEND(N,RS,SB,ID)
C ----- SKRIV UT N OCH T I TABFILEN
      CALL WTAB(N,T,SPEC)
      GOTO 10
      END
C

```

```

C ***** SUBROUTINBLOCKET Börjar *****
      SUBROUTINE SEND(N,RS,SB,ID)
      INTEGER N(0:15,1:5)
      INTEGER RS,SB,ID
C SÄND RS (RESET COUNTER) TILL VAXEL
C SÄND SB (SEND(N,RS,SB,ID) BYTE) TILL VAXEL
C DÄREFTER TERM. NR. MOT NORDINGÅNG 0, N(0,5)
C
C           .....
C           MOT NORDINGÅNG 15,N(15,5)
C TERMINAL MED NUMMER 63 ANVÄNDES EJ
C TERMINAL MED NUMMER 0 ANVÄNDES FÖR UTLOGGNINGSMEDD.
C NORDINGÅNG 0 N(0,3) ÄR KOPPLAD TILL IO-PORTEN
C OCH ANVÄNDES FÖR ATT SÄNDA MEDDELANDE TILL TERMINALER.
C - EFTER VARJE RS GÖRES ETT LITET DELAY 5*.020 SEK. VILKET
C - VÄXELN KRÄVER FÖR RESET.
C *****
      CALL COBUF(ID)
      CALL OUTCH(ID,RS)
      CALL HOLD(5,1)
      CALL OUTCH(ID,SB)
      DO 10 I1=0,15
      CALL OUTCH(ID,N(I1,5))
10      CONTINUE
      CALL OUTCH(ID,RS)
      CALL HOLD(5,1)
      RETURN
      END

```

```

C *****
      SUBROUTINE TERMST(TOFN,RS,TS,ID)
C SUBROUTINEN CHECKAR TERMINALINGÅNGARNA 0-63
C OM TERMINAL N ÄR PÅSLAGEN BLIR TOFN(N)=1
C                               AVSLAGEN BLIR TOFN(N)=0
C
C *****
      INTEGER TOFN(0:63),TE(1:17),K(4)
      INTEGER RS,TS,ID
      CALL CIBUF(ID)
      CALL OUTCH(ID,RS)
      CALL HOLD(5,1)
      CALL OUTCH(ID,TS)
C ---- LÄS EN SLASKBYTE FRÅN VÄXELN TE(17)
      DO 10 I1=1,17
      TE(I1)=INCH(ID)
10      CONTINUE
      CALL OUTCH(ID,RS)
      CALL HOLD(5,1)
C --- KONVERTERA,          T3,T2,T1,T0
C                          T7,T6,T5,T4
C                          .....0,1..
C                          T63,T62,T61,T60

      K4=17B
      K(1)=1
      K(2)=2
      K(3)=4
      K(4)=10B
      DO 30 I1=1,16
      ITE=IAND(TE(I1),K4)
      DO 20 I2=1,4
      IT=IAND(ITE,K(I2))
      I3=(I1-1)*4+I2-1
      TOFN(I3)=0

```



```
IF (IT.GT.0) TOFN(I3)=1
20 CONTINUE
30 CONTINUE
RETURN
END
```

C \*\*\*\*\*

          SUBROUTINE COMPARE(DIFF,T,RS,TS,ID,TOFN)

C LÄS AV TERMINALSTATUS OCH JÄMFÖR MED T(I1,5)

C OM TERMINAL N BLIVIT AVSLAGEN SÄTTES DIFF(N)=-1

C OM TERMINALSTATUS OFÖRÄNDRAD SÄTTES DIFF(N)=0

C OM TERMINAL N BLIVIT PÅSLAGEN SÄTTES DIFF(N)=1

C

          INTEGER TOFN(0:63),DIFF(0:63)

          INTEGER T(0:63,1:6)

          INTEGER RS,TS,ID

          CALL TERMST(TOFN,RS,TS,ID)

          DO 40 I1=0,63

          DIFF(I1)=0

          IF (TOFN(I1).EQ.0.AND.T(I1,5).GE.0) GOTO 20

          IF (TOFN(I1).EQ.1.AND.T(I1,5).LT.0) GOTO 30

          GOTO 40

20          DIFF(I1)=-1

          GOTO 40

30          DIFF(I1)=1

40          CONTINUE

          RETURN

          END

C \*\*\*\*\*

          SUBROUTINE UTLOG(NT,N,T,RS,SB,ID)

C LOGGA UT TERMINAL NUMMER NT

C SPARA TERMINALENS NORDINGÅNG I ND

C KOPPLA NORDINGÅNGEN MOT TERMINAL 0, OCH SÄND

C UTLOGGNINGSTEXTEN

C ÅTERSTÄLL ADRESSEN

C

          INTEGER N(0:15,1:5),T(0:63,1:6),ICH(1:10)

          INTEGER NT,RS,SB,ID

C \*\*\* :=72B E=105B X=130B ESC=33B L=114B

C \*\*\* O=117B G=107B CR=15B

          ICH(1)=15B

          ICH(2)=72B

          ICH(3)=105B

          ICH(4)=130B

          ICH(5)=15B

          ICH(6)=33B

          ICH(7)=114B

          ICH(8)=117B

          ICH(9)=107B

          ICH(10)=15B

          ND=T(NT,5)

          N(ND,5)=0

          CALL SEND(N,RS,SB,ID)

C --- SKICKA UTLOGGNINGSSEKVENNS

          CALL OUTCH(ID,RS)

          CALL HOLD(5,1)

C

          DO 10 I1=1,2

          DO 9 I2=1,5

          CALL OUTCH(ID,ICH(I2))

9          CONTINUE

10         CONTINUE

C

```
DO 20 I1=1,2  
DO 19 I2=5,10  
CALL HOLD(5,1)  
CALL OUTCH(ID,ICH(I2))
```

19 CONTINUE

20 CONTINUE

C

C ----- ÅTERSTÄLL ADRESSEN

```
      N(ND,5)=NT
```

```
      CALL SEND(N,RS,SB,ID)
```

```
      RETURN
```

```
      END
```

```

C *****
      SUBROUTINE SENDMESS(NT,MESS,RS,SB,T63,ID)
C SÄND MEDDELANDE TILL TERMINAL NT.
C 3 RADER OM VARDERA 50 TECKEN
      CHARACTER MESS(3)*50
      INTEGER NT,RS,SB,T63,ID
C
C --- SÄTT TERMINALADRESS I FÖRSTA BYTE (BARA 1 BYTE)
      CALL COBUF(ID)
      CALL OUTCH(ID,RS)
      CALL HOLD(5,1)
      CALL OUTCH(ID,SB)
      CALL OUTCH(ID,NT)
      CALL OUTCH(ID,RS)
      CALL HOLD(5,1)
C
C ---- SÄND TEXT (FÅR EJ INNEHÅLLA TS ELLER SB)
C
      DO 20 I1=1,3
      WRITE(ID,10)MESS(I1)
20    CONTINUE
10    FORMAT(1H ,/,A50)
C ---- SÄTT TERMINAL ADRESS 63 MOT NORDINGÅNG 0
C ---- FYRA TECKEN SYNS PÅ TERMINALEN EFTER MEDDELANDET.
C ---- DESSA TECKEN STÄLLER OM VÄXELN.
      CALL COBUF(ID)
      CALL OUTCH(ID,RS)
      CALL HOLD(5,1)
      CALL OUTCH(ID,SB)
      CALL OUTCH(ID,T63)
      CALL OUTCH(ID,RS)
      CALL HOLD(5,1)
      RETURN
      END

```

```

C *****
      SUBROUTINE WTAB(N,T,SPEC)
C SKRIV N,T OCH SPEC MATRISERNA PÅ TABFIL
C
      INTEGER N(0:15,1:5),T(0:63,1:6)
      CHARACTER SPEC(0:63)*10
10      OPEN(9,ERR=100,FILE='VAXEL-TAB:S',ACCESS='W')
100     IF (ERRCODE.NE.0) GOTO 10
C
      DO 300 I1=0,15
      WRITE(9,70)(N(I1,I2),I2=1,5)
300     CONTINUE
      DO 400 I1=0,63
      WRITE(9,80)(T(I1,I2),I2=1,6),SPEC(I1)
400     CONTINUE
70      FORMAT(5I5)
80      FORMAT(6I5,A10)
      CLOSE (9)
      RETURN
      END

```

```

C *****
      SUBROUTINE RTAB(N,T,SPEC)
C LÄS IN FÄLTEN N,T OCH SPEC FRÅN TABFIL
C LÄS FÖRSTA RADEN MED FORMAT 60 FÖR ATT SKIPPA ETT
C EXTRA LF PRODUCERAT AV WTAB.
C
      INTEGER N(0:15,1:5),T(0:63,1:6)
      CHARACTER SPEC(0:63)*10
10      OPEN(9,ERR=100,FILE='VAXEL-TAB:S',ACCESS='R')
100     IF (ERRCODE.NE.0) GOTO 10
C
      READ(9,60)(N(1,I2),I2=1,5)
      DO 300 I1=1,15
      READ(9,70)(N(I1,I2),I2=1,5)
300     CONTINUE
      DO 400 I1=0,63
      READ(9,80)(T(I1,I2),I2=1,6),SPEC(I1)
400     CONTINUE
60      FORMAT(1X,5I5)
70      FORMAT(5I5)
80      FORMAT(6I5,A10)
      CLOSE (9)
      RETURN
      END

```

```

C *****
      SUBROUTINE CUP(NT,IQUE,N,T,RS,SB,T63,ID)
C TERMINAL NT KOPPLAS UPP MOT PRIMÄR NORDINGÅNG OM
C SÅDAN ÄR ÅSATT, T(NT,3).
C I ANNAT FALL TILL FÖRSTA LEDIGA.
C DEN FYSISKA KOPPLINGEN SKER I HUVUDPROGRAMMET MED
C SUBROUTINANROPTET CALL SEND(N,RS,SB,ID).
C VARIABELN IQUE KAN ANVÄNDAS FÖR ATT BETECKNA ANATALET
C TERMINALER I KÖ.
C
      CHARACTER MESS(3)*50
      INTEGER N(0:15,1:5),T(0:63,1:6)
      INTEGER NT,IQUE,RS,SB,T63,ID
      I1=0
5      I1=I1+1
      IF (I1.LE.15) GOTO 10
C ---- INGA LEDIGA NORDINGÅNGAR
      MESS(1)=' TYVÄRR FINNS INGEN LEDIG NORDINGÅNG FÖR'
      MESS(2)=' NÄRVARANDE. FÖRSÖK OM EN STUND IGEN'
      MESS(3)=' -'
      CALL SENDMESS(NT,MESS,RS,SB,T63,ID)
      T(NT,5)=0
      GOTO 400
C
C ***** ÄR NORDINGÅNG I1 LEDIG OCH INKOPPLAD MOT NORD
C
10      IF (N(I1,5).NE.63.OR.N(I1,2).NE.-1) GOTO 5
C ---- FÖRSÖK KOPPLA MOT PRIMÄR NORDINGÅNG
      I2=T(NT,3)
      IF (I2.LT.1) GOTO 14
      IF (N(I2,5).NE.63.OR.N(I2,2).NE.-1) GOTO 14
      WRITE(MESS(3),12)N(I2,3)
12      FORMAT(I5,'      ')
      N(I2,5)=NT

```



```
      T(NT,5)=I2
      GOTO 300
C ----- KOPPLA MOT FÖRSTA LEDIGA NORDINGÅNG
14      N(I1,5)=NT
      T(NT,5)=I1
      WRITE(MESS(3),12)N(I1,3)
300     MESS(1)=' DU ÄR NU KOPPLAD MOT'
      MESS(2)=' NORDINGÅNG NUMMER'
      CALL SENDMESS(NT,MESS,RS,SB,T63,ID)
400     RETURN
      END
C
      EOF
```

B. Programmet VEM.

=====

```
PROGRAM VEM
C *****
C * PER CHRISTIANSSON 16.8.1981
C ***** PROGRAMMET LISTAR VÄXELSTATUS
C LÄS IN FÄLTEN N,T OCH SPEC FRÅN TABFIL
      INTEGER N,T
      DIMENSION N(0:15,1:5),T(0:63,1:6)
      CHARACTER SPEC(0:63)*10
10    OPEN(9,ERR=100,FILE='(RT)VAXEL-TAB:S',ACCESS='R')
100   IF (ERRCODE.NE.0) GOTO 10
C
60    FORMAT(1X,5I5)
70    FORMAT(5I5)
80    FORMAT(6I5,A10)
      READ(9,60)(N(1,I2),I2=1,5)
C --- FÖRSSTA RADEN I FILEN VAXEL-TAB BÖRJAR MED ETT LF
C --- PRODUCERAT AV SUBROUTINEN WTAB I RT-PROG. VAXEL
C --- DÄRFÖR INLÄSES DENNA SEPARAT.
      DO 103 I1=1,15
      READ(9,70)(N(I1,I2),I2=1,5)
103   CONTINUE
      DO 104 I1=0,63
      READ(9,80)(T(I1,I2),I2=1,6),SPEC(I1)
104   CONTINUE
      CLOSE (9)
C
C
      WRITE(1,110)
110   FORMAT(1X,*LOG.NO      *,*TERM.NO*,*  PLACERING*)
      DO 600 I1=1,15
      IF (N(I1,2).LT.-1) GOTO 600
```

```
IF (N(I1,2).GE.1) GOTO 200
IF (N(I1,5).GE.1.AND.N(I1,5).LE.62) GOTO 250
IF (N(I1,5).EQ.63) GOTO 300
STOP

C
200 WRITE(1,205)N(I1,3),N(I1,5),SPEC(N(I1,5))
205 FORMAT(1X,I6,I10,4X,A10,* FAST KOPPLING*)
GOTO 600

250 WRITE(1,255)N(I1,3),N(I1,5),SPEC(N(I1,5))
255 FORMAT(1X,I6,I10,4X,A10)
GOTO 600

300 WRITE(1,305)N(I1,3)
305 FORMAT(1X,I6,* LEDIG*)
600 CONTINUE
END
EOF
```

C. Tabellfilen vid VDC.

=====

Nedan är hela tabellfilen VAXEL-TAB:s listad sådan den såg ut i september 1981.

0	0	0	0	0	
1	-1	52	2400	63	
2	1	53	2400	1	
3	-1	7	2400	10	
4	-1	48	2400	63	
5	-1	50	2400	3	
6	-1	54	2400	63	
7	-2	15	2400	63	
8	-1	42	2400	63	
9	-2	43	2400	63	
10	-2	0	2400	63	
11	-2	0	2400	63	
12	-2	0	2400	63	
13	-2	0	2400	63	
14	-2	0	2400	63	
15	-2	0	2400	63	
0	-2	0	2400	-1	OUTLOGG
1	2	0	2400	-1	OELITE 9
2	-1	0	2400	-1	OELITE 8
3	-1	5	2400	5	OBT2-1
4	-1	4	2400	-1	OBS-1-LAB
5	-1	0	2400	-1	OBT2-2
6	-1	0	2400	-1	OBS-2
7	-1	0	2400	-1	OGEOT-1
8	-1	0	2400	-1	OBT1-1
9	-1	0	2400	-1	OTT-1
10	-1	0	2400	3	OGEOD-1
11	-1	0	2400	-1	OBS-3

12	-2	0	2400	-1	OBT1-2-LAB
13	-2	0	2400	-1	OXXX
14	-2	0	2400	-1	OXXX
15	-2	0	2400	-1	OXXX
16	-2	0	2400	-1	OXXX
17	-2	0	2400	-1	OXXX
18	-2	0	2400	-1	OXXX
19	-2	0	2400	-1	OXXX
20	-2	0	2400	-1	OXXX
21	-2	0	2400	-1	OXXX
22	-2	0	2400	-1	OXXX
23	-2	0	2400	-1	OXXX
24	-2	0	2400	-1	OXXX
25	-2	0	2400	-1	OXXX
26	-2	0	2400	-1	OXXX
27	-2	0	2400	-1	OXXX
28	-2	0	2400	-1	OXXX
29	-2	0	2400	-1	OXXX
30	-2	0	2400	-1	OXXX
31	-2	0	2400	-1	OXXX
32	-2	0	2400	-1	OXXX
33	-2	0	2400	-1	OXXX
34	-2	0	2400	-1	OXXX
35	-2	0	2400	-1	OXXX
36	-2	0	2400	-1	OXXX
37	-2	0	2400	-1	OXXX
38	-2	0	2400	-1	OXXX
39	-2	0	2400	-1	OXXX
40	-2	0	2400	-1	OXXX
41	-2	0	2400	-1	OXXX
42	-2	0	2400	-1	OXXX
43	-2	0	2400	-1	OXXX
44	-2	0	2400	-1	OXXX
45	-2	0	2400	-1	OXXX

46	-2	0 2400	-1	OXXX
47	-2	0 2400	-1	OXXX
48	-2	0 2400	-1	OXXX
49	-2	0 2400	-1	OXXX
50	-2	0 2400	-1	OXXX
51	-2	0 2400	-1	OXXX
52	-2	0 2400	-1	OXXX
53	-2	0 2400	-1	OXXX
54	-2	0 2400	-1	OXXX
55	-2	0 2400	-1	OXXX
56	-2	0 2400	-1	OXXX
57	-2	0 2400	-1	OXXX
58	-2	0 2400	-1	OXXX
59	-2	0 2400	-1	OXXX
60	-2	0 2400	-1	OXXX
61	-2	0 2400	-1	OXXX
62	-2	0 2400	-1	OXXX
63	-2	0 2400	-1	OFRINORD

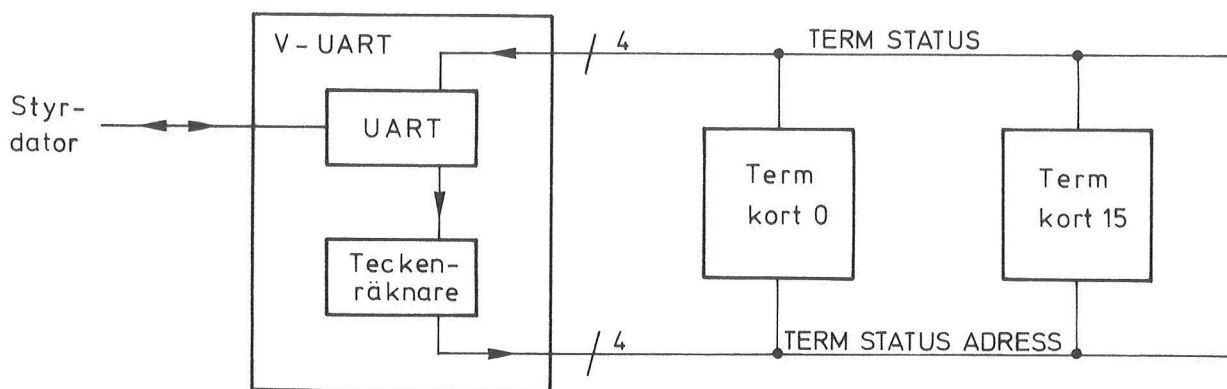
#### D. Kretskortens funktion.

=====

Växeln är uppbyggd av 3 st olika korttyper, korten är tillverkade i CMOS teknologi vilket medför låg strömförbrukning.

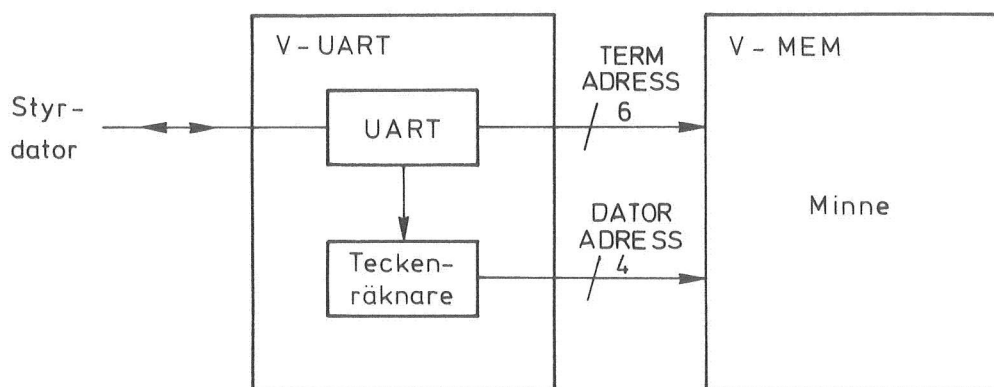
V-UART kortet är ett kommunikationskort som kommunicerar med styrdatorn via en seriell in/ut ledning (strömslinga, current loop CL). Kortet innehåller optokopplare för anslutning till styrledning, UART (Universell Asynkron Resiver Transmitter) för omvandling av 8 bit seriell data till 8 bit parallell data och 2 st teckenräknare för räkning av sända resp mottagna tecken. På kortet finns möjlighet att koda valfritt tecken för RS (reset), SB (send byte) och TS (term status).

V-UART kortet reagerar endast på RS, SB och TS tecken, andra tecken stör ej kortet. RS nollställer teckenräknarna. När kortet mottagit TS sänds 16 tecken med information om vilka terminaler som är påkopplade, mindre än 16 tecken kan överföras genom att styrdatorn sänder RS när den mottagit lämpligt antal tecken (användbart om mindre antal terminalkort används).



Figur D-1 V-UART anslutning till terminalkorten.

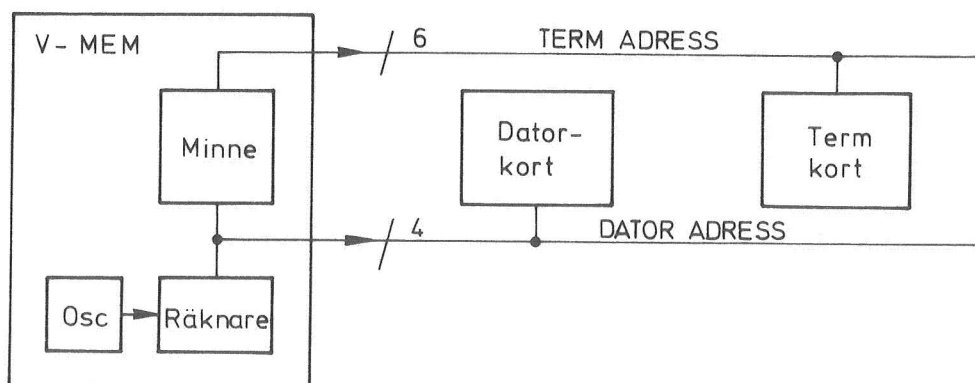
Efter att SB mottagits är kortet redo att ta emot upp till 16 tecken som anger terminalingångsadresser och teckenräknaren anger motsvarande datoringångsadresser. Dessa 16 terminal- och datoradresser överföres i parallellform till minneskortet V-MEM.



Figur D-2 V-UART kortets anslutning till V-MEM.

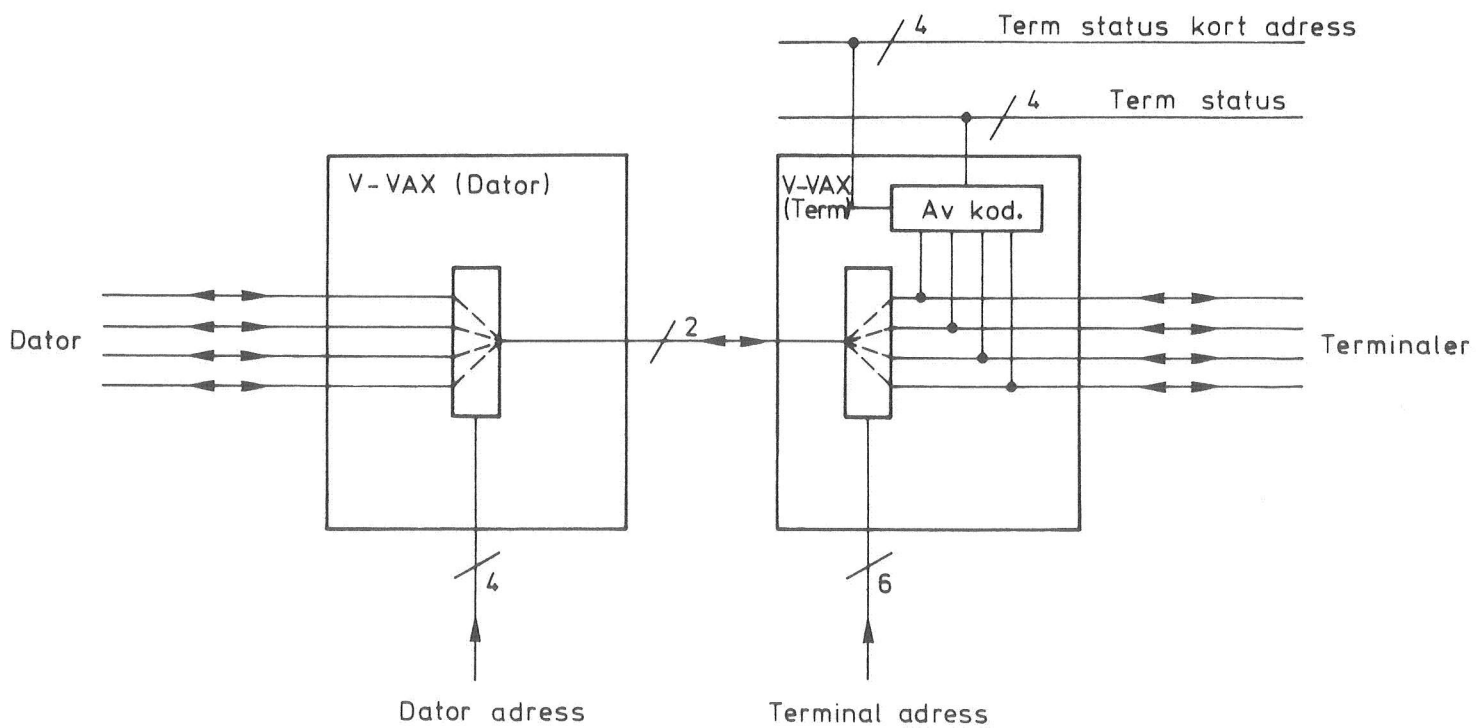
V-MEM kortet innehåller minne 16\*6 bit och multiplexoscillator. Avläsning av minnets 16 celler sker kontinuerligt i logisk ordning med en hastighet av 768000 avläsningar/sek. Adress och data läggs ut på en bussledning som terminal- och datorkort är anslutna till. Skrivning i minnet sker från V-UART kortet utan att stoppa pågående avläsning.





Figur D-3 V-MEM kortets anslutning till terminal respektive datorkort.

V-VAX kortets komponentbestyckning varierar något beroende på om det skall användas som terminal- eller datorkort. Till kortet kan 4 terminaler eller datorringångar anslutas. Kortet innehåller optokopplare för anslutning av terminaler med strömslinga, en krets som känner av om terminal är påkopplad (terminalens sändslinga sluten) samt analoga omkopplare för sammankoppling av terminal med aktuell datorringång. På kortet finns kortadressavkodare och 4-polig omkopplare som anger kortadress.



Figur D-4 Terminal-och datorkortens sammankoppling.