

Organisationsenhet AUTOTESTTEKNIK	Datum 1982-08-23	Reg nr PVA-R185:V6/11/12
Tjänsteställe/Handläggare 4340	Mottagare/Delgivning	
Ärende PVA-R181:V6/11/12 Speciella detaljinstruktioner för PVA. ATS 6/11/12		

ATS 6/11/12:
TESTPROGRAMMERING

INNEHÅLLSFÖRTECKNING

1. TESTPROGRAMMERING ALLMÄNT
2. NAMNSÄTTNING AV FILER
3. TESTMONITOR - /DIR, TEST, TECH
 - 3.1 Allmänt
 - 3.2 /DIR
 - 3.3 TEST
 - 3.4 TECH
4. TESTSEKVENSER
 - 4.1 Allmänt
 - 4.2 TSEQ01
 - 4.3 TSEQ02
5. BRANCHNING
 - 5.1 Allmänt
 - 5.2 Branchning - då branchprogrammet är skrivet i BASIC.
 - 5.3 Branchning - då branchprogrammet är skrivet i FORTRAN.
 - 5.4 Praktiska synpunkter
 - 5.5 Anrop av devicesubrutiner i FORTRAN
6. UTFORMNING AV BASIC - PROGRAM VID BRANCHNING
 - 6.1 Beskrivning av XXXXBO-fil för TSEQ01
 - 6.2 Utvärdering av testresultat
7. UTFORMNING AV BASIC - PROGRAM VID ENBART BASIC
 - 7.1 Beskrivning av XXXXBO-fil för TSEQ02
 - 7.2 Utvärdering av testresultat.

- 8. UTFORMNING AV BASIC - PROGRAM ALLMÄNT
- 8.1 Testidentifiering

- 9. OPERATÖRSKOMMUNIKATION
- 9.1 Allmänt
- 9.2 BASIC-satser för operatörskommunikation
- 9.3 Programmering av utskrifter
- 9.4 D1-filer

- 10. NEDKOPPLING AV TESTARE

- 11. TRAP-FUNKTIONER
- 11.1 Allmänt
- 11.2 Gröna knappen
- 11.3 KEPCO - kraftenheter
- 11.4 HP 6131 - likspänningsenheter
- 11.5 Korskopplingsfältet

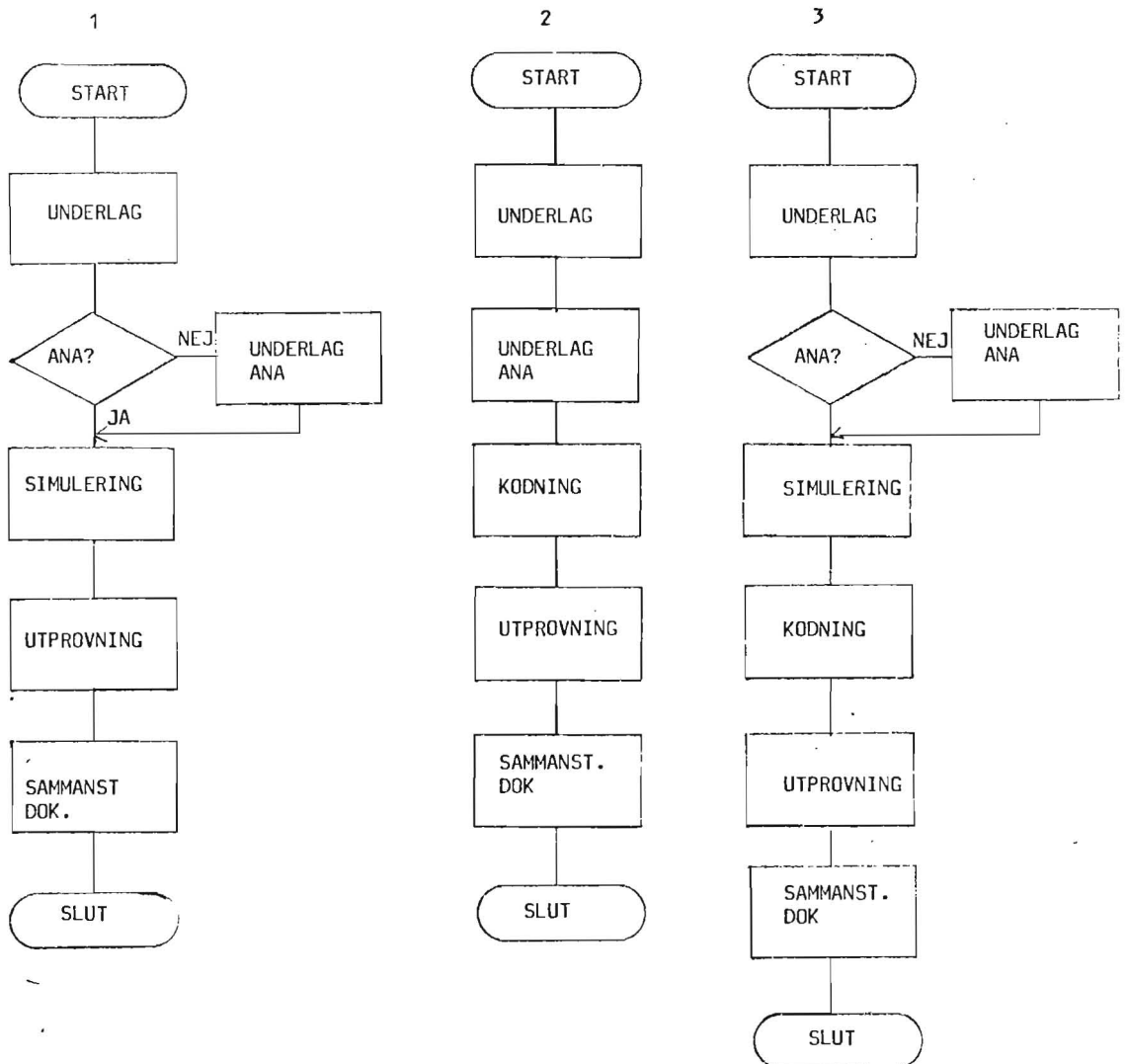
- 12. TESTAID - FASTRACE
- 12.1 ROMRD

1. TESTPROGRAMMERING ALLMÄNT

Testprogramframtagningen i ATS 6/11/12 kommer att omfatta tre olika typer av testobjekt.

1. Simulerade testobjekt utan branchning
2. Icke simulerade testobjekt
3. Simulerade testobjekt med branchning

Flödesschema för testprogramframtagning:



Kommentarer till flödesschema:

START

Denna punkt innebär att handläggaren tar kontakt med sue-koordinatören på 4330 för att få A0-nr och kostnadsram. FTPSUE (Färdiga testprogram SUE) är en databas där vissa uppgifter om sue:n skall föras in.

1.	F-nr beteckning	F6400-054523
2.	Ursprungsbeteckning	9116345-10-1
3.	Testprogrambeteckning	AN01
4.	Handläggare	SP
5.	ANA identitetsnummer	18
6.	A0-nr	4560-1875
7.	Klarrapportvecka	8212
8.	Skivbeteckning	3000
9.	Autotestare	ATS 6

Inläggning av data till FTPSUE sköts av sue-koordinatör på 4330. Utlistor från databasen distribueras till objekthandläggaren + verkstaden.

UNDERLAG

Här menas framtagning av underlag som behövs för det fortsatta arbetet t.ex kretsschema, komponentlista, komponentplaceringslista.

Här ingår även SIGLIST-arbetet, dvs kodningen av kretskort (gäller endast de simulerade korten).

SAMMANSTÄLLNING AV DOKUMENTATIONEN

Sammanställning av all dokumentation till en PRU-pärm och en TPD-pärm enligt PVA-R191:V6/11/12.

SLUT

4330 ombesörjer klarrapportering av Ao-nr, Fus-registrering.

Under hela programframtagningen så skall PVA-R186:V6/11/12:
Skivhantering samt rutiner för back-up följas.

2. NAMNSÄTTNING AV FILER

Samtliga filer som tas fram under utvecklingen av ett testprogram ska ges ett namn av enhetlig typ, bestående av sex tecken, där de olika teckenpositionerna har följande betydelse:

- - - - -
Pos 1 2 3 4 5 6

Pos 1-2: Förkortning på två bokstäver som anger i vilket elektro-
niksystem testobjektet ingår. (Framgår av FTSPUE).

Pos 3-4: Löpnummer som anger vilket testobjekt (sue) i respektive
system det gäller. (Framgår av FTSPUE)

Pos 5: Bokstav som anger filtyp.
Följande förkortningar ska användas.

S = Kretsbeskrivning för SIGLIST
N = NET som ska ingå i kretsbeskrivningen och tas
fram som separata filer.
P = Insignaler (pattern) för SIMULA.
C = "Crunch"-filer. Används för att i komprimerad
form spara resultatet efter körning av pro-
grammen SIMSET eller SIMULA i TESTAID.
A = Adapterinformation till PATDOC.
I = Indata för SETUP
O = Indata för PROB
R = "Crunch"-fil med resultatet från körning av
PATDOC.
U = Oppptäckbara fel. (ASCII-strängar).
D = OP-instruktioner
B,E = Basic-filer (PM)
F,G = Basic-filer (FS)
T = Avslutade testprogram efter FASTRACE.
W = Upstartningsprogram för Basic.

X,Y,Z = Branch-program i FORTRAN.

Siffror 0-9: För lagring av minnesinnehåll i ROM och PROM. Antal filer kan här bli så stort att pos 6 inte är tillräcklig för löpnummer.

Pos 6: Löpnummer för respektive filtyp.
Siffror och bokstäver kan användas.

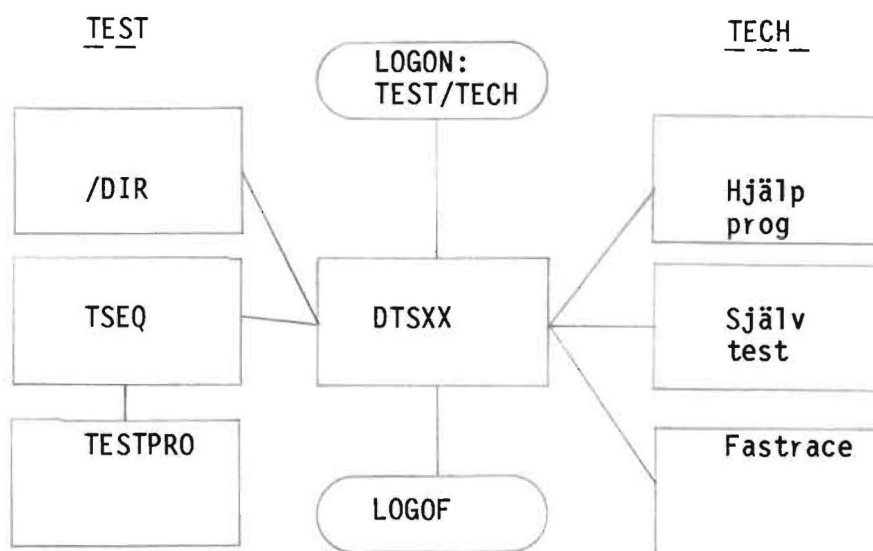
Om pos 5 är X, Y eller Z, dvs branch-program i FORTRAN, används pos 6 på följande sätt:

⌘ = källkod för FORTRAN-program,
% = relokerbar kod för FORTRAN-program.

Exempel: Ett insignalmonster som används vid simulering av testprogrammet för adressavkodare, PS46, betecknas PS01P2.

3. TESTMONITOR - /DIR, TEST, TECH3.1 Allmänt

Testmonitorn DTSXX är det program som tillsammans med vissa andra filer administrerar test, självttest och vissa delar av programutprovningen mm. Se blockschema fig 1.



Blockschema DTSXX

Fig 1

Det finns två olika sätt att logga på för att starta testmonitorn, TEST för produktion och TECH för självttest eller utprovning av testprogram (simulerade).

3.2 /DIR

/DIR är en datafil där testmonitorn hämtar information om hur de olika testobjekten skall testas, eller provas ut.

/DIR ska vara placerad på LU 15.

För varje testobjekt skall det i /DIR finnas två rader information enligt följande format.

A Format för testsekvens TSEQ01

*

FXXXX -XXXXXX,Y

TSEQ01,XXXXT1,LU,XXXX,XXXXD1,P1,ID,P2,P3

*

FXXXX-XXXXX = Beteckning på testobjektet, normalt F-nummer.

Y = Modifieringsläge, ruta som är kryssad på F-nummerskylten.

TSEQ01 = Testsekvens för simulerade testobjekt.
Se separat beskrivning i kapitel 4

XXXXT1 = Testdatafil för simulerade testobjekt. Ska alltid sluta på T1.

LU = LU-nummer på skivarian där samtliga filer för ett testobjekt är placerade.

XXXXX = Branchprogram i FORTRAN eller uppstartningsprogram för BASIC-tester.

XXXXD1 = Fil för operatörsinstruktioner, visas på bildskärmen.

P1 = Parameter för antal testdatafiler XXXXTN.

ID = Adapterkod för HP9412A i ATS-11/12.

P2 = 1= Stop efter "power on".

P3 = 1= Branchning vid digital felsökning.

Exempel: F1234-123456,2
TSEQ01::-2, EP01T1,-15,,EP01D1

B. Format för testsekvens TSEQ02

*

FXXXX - XXXXXX,Y

TSEQ02, XXXXW, LU, ID, P1, P2, P3, P4, P5

*

FXXXX-XXXXXX = Beteckning på testobjektet, normalt F-nummer.

Y = Modifieringsläge, ruta som är kryssad på F-nummerskylten.

TSEQ02 = Testsekvens för icke simulerade testobjekt .
Se separat beskrivning kapitel 4.

XXXXW = Upstartningsprogram för basic-tester.

LU = LU-nummer på skivarian där samtliga filer för ett testobjekt är placerade.

ID = Adapterkod för HP 9412A i ATS-11/12.

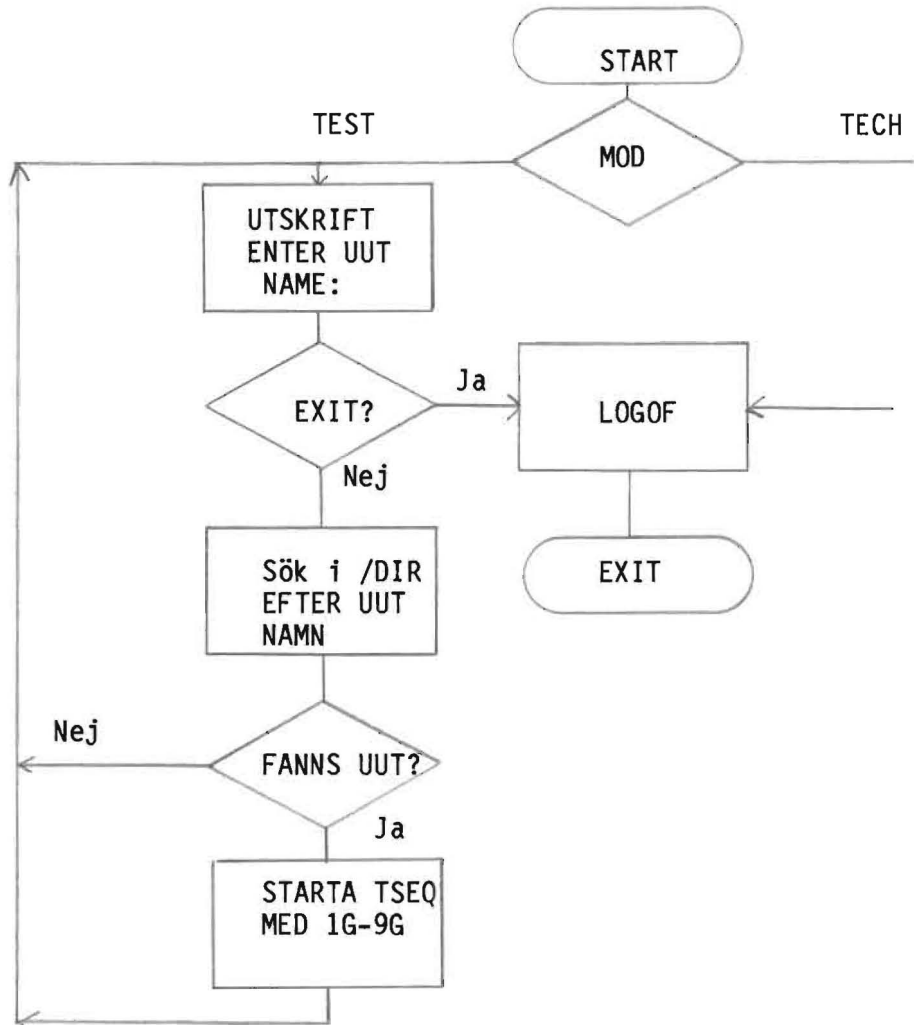
P1-P5 = Parametrar för överföring av data till test-program

Exempel: F1234-123456,2
TSEQ02::-2, EP01W,-16, 1234

3.3

TEST

Vid påloggning som TEST startas testmonitorn (DTSXX) upp i TEST-mod, vilken är den mod som skall användas vid test och felsökning i samband med produktion.



Flödesschema TEST-mod

DTSXX

Fig 2

Efter påloggning som TEST får operatören frågan "ENTER UUT NAME:" Operatören svarar med testobjektets beteckning, motsvarande rad-nr 1 av informationen i /DIR.

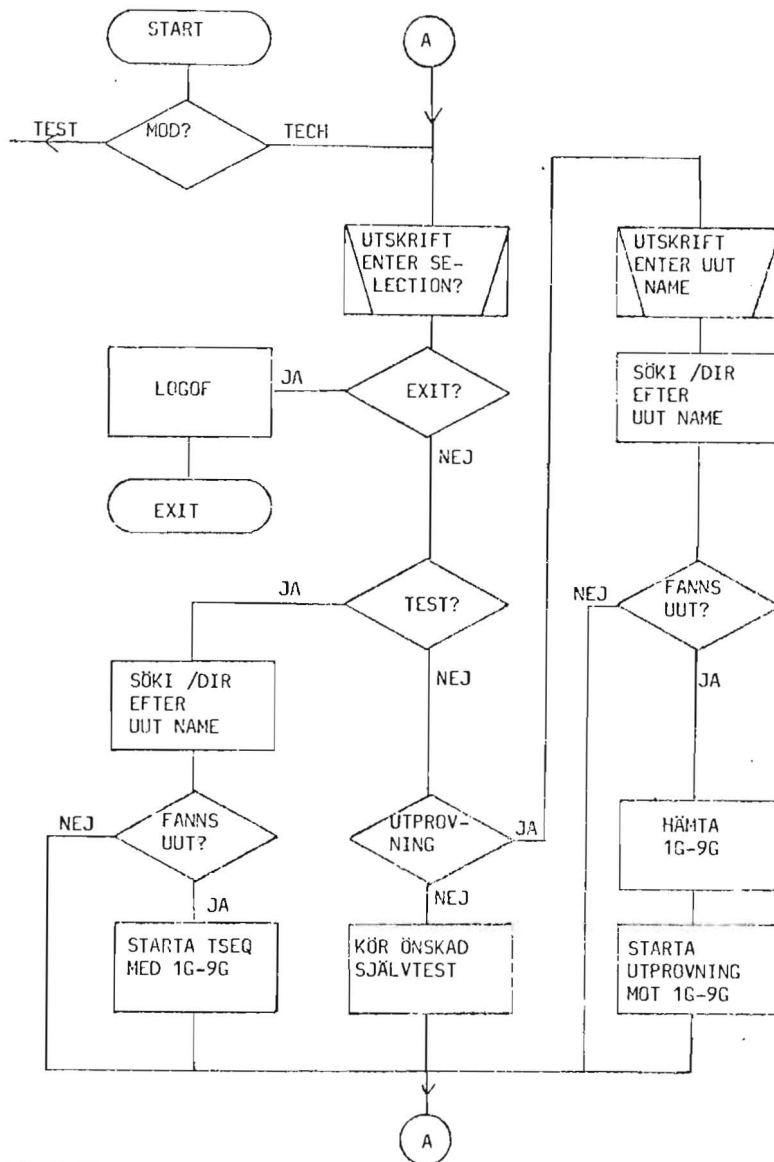
Testmonitorn söker därefter i /DIR efter motsvarande beteckning. Informationen i rad nr 2 i /DIR plockas upp och angiven test-sekvens (TSEQXX) startas upp mot övriga parametrar i rad nr 2.

Efter slutförd test återgår testmonitorn till "ENTER UUT NAME:".

3.4

TECH

Vid påloggning som TECH startas testmonitorn (DTSXX) upp i TECH-mod. TECH-moden används i huvudsak vid programutprovning av simulerade testobjekt och körning av självttest. För att programutprovning skall kunna ske under TECH-mod måste testobjektet vara inlagt i /DIR se kapitel 3.2.



Flödesschema TECH-mod
DTSXX
Fig 3

TECH-modens funktion framgår i stort sett av fig 3. Efter påloggning som TECH får operatören frågan "ENTER SELECTION", vid svar? eller ?? kommer utskriften enligt fig 4 på bildskärmen.

```

COMMAND  DESCRIPTION
-----
SET      HARDWARE SETUP PROGRAM
PRO      GUIDED PROBE OPERATION - NO TEST CHANGES ALLOWED
PROSAV   GUIDED PROBE OPERATION - TEST CHANGES PERMITTED
#PON     STROBE POWER SUPPLIES ON
#POFF    STROBE POWER SUPPLIES OFF
#SFT     9571A SYSTEM HARDWARE FUNCTIONAL TEST
#SPT     9571A SYSTEM HARDWARE PERFORMANCE TEST (CALIBRATION)
#AST     ANALOG SELF TESTS
#SAST    SHORT ANALOG SELF TEST
IC2      U9500-60863 SFT ASSEMBLY TEST PROGRAM
#IC2SFT  TESTAID-III SOFTWARE FUNCTIONAL TEST
#ED      RTE EDITOR
EXIT     RETURN TO LOGON
-----
OPERATOR MAY ENTER UUT NAME, COMMAND, OR ?COMMAND
TYPE ?COMMAND FOR ADDITIONAL INFORMATION
-----

```

Utskrift ? eller ??

Fig 4.

SET startar upp Fastrace-programmet SETUP vilket används vid programutprovningen för konfigurering och inställning av DTU och kraftenheter. För mer information se Fastrace manualen SETUP.

PRO startar upp Fastrace-programmet PROB vilket används vid programutprovningen för bland annat verifiering och dokumentering av testprogrammen. För mer information se Fastrace-manualen PROB.

PROSAV fungerar på samma sätt som PRO men tillåter ändringar av testdatafilen (XXXXT1). För mer information se Fastrace-manualen PROBE.

#PON slår till kraftmatningen.

#POF slår av kraftmatningen.

#SFT startar en interaktiv mode för körning av självttesterna för DTU:n.

#SPT startar en interaktiv mode för körning av kalibreringsprogrammen för DTU:n.

#AST startar analog självttest.

#SAST startar en kort version av den analoga självttesten.

IC2 kör testprogrammet för självttestadaptorn HP 09500-60863.

#IC2SFT är en självttest av samtliga Testaid-programmen.

#ED startar up editorn (ED)

EXIT eller EX avslutar testmonitorn.

? + kommando t.ex ? SET, ?EX osv ger en utskrift på bildskärmen som ytterligare förklarar kommandots funktion.

Man kan även köra produktionstest under TECH-mod genom att ange testobjektets beteckning (F-nummer).

Testen utförs då på samma sätt som vid körning under TEST-mode. Efter avslutad test återgår testmonitorn till "ENTER SELECTION"

4. TESTSEKVENSER4.1 Allmänt

En testsekvens är en kommandofil som styr program och dataflödet vid test. Två stycken generella testsekvenser finns i ATS-6/11/12.

TSEQ01 är en generell testsekvens för simulerade testprogram. TSEQ01 kan användas för såväl ren digitaltest som hybridtest (branchning).

TSEQ02 är en generell testsekvens för testprogram skrivna i BASIC. Dessa två generella testsekvenser är placerade på systemskivan LU #2.

4.2 TSEQ01

TSEQ01:s funktion framgår i stort sett av det förenklade block-schemat i fig 1.

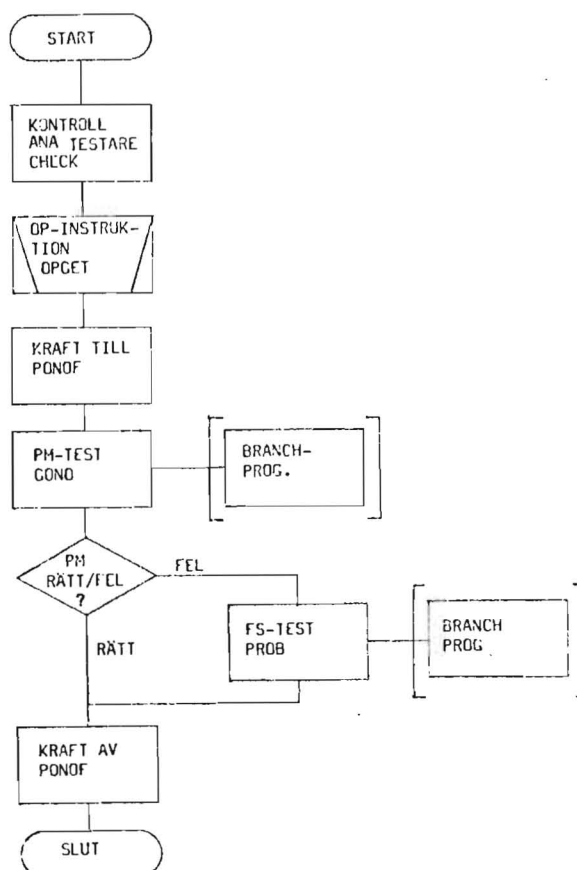


Fig 1.

Först kontrolleras ANA-kod, DTU-konfiguration och att matnings-
spänningen är tillslagen (IDANA,CHECK).

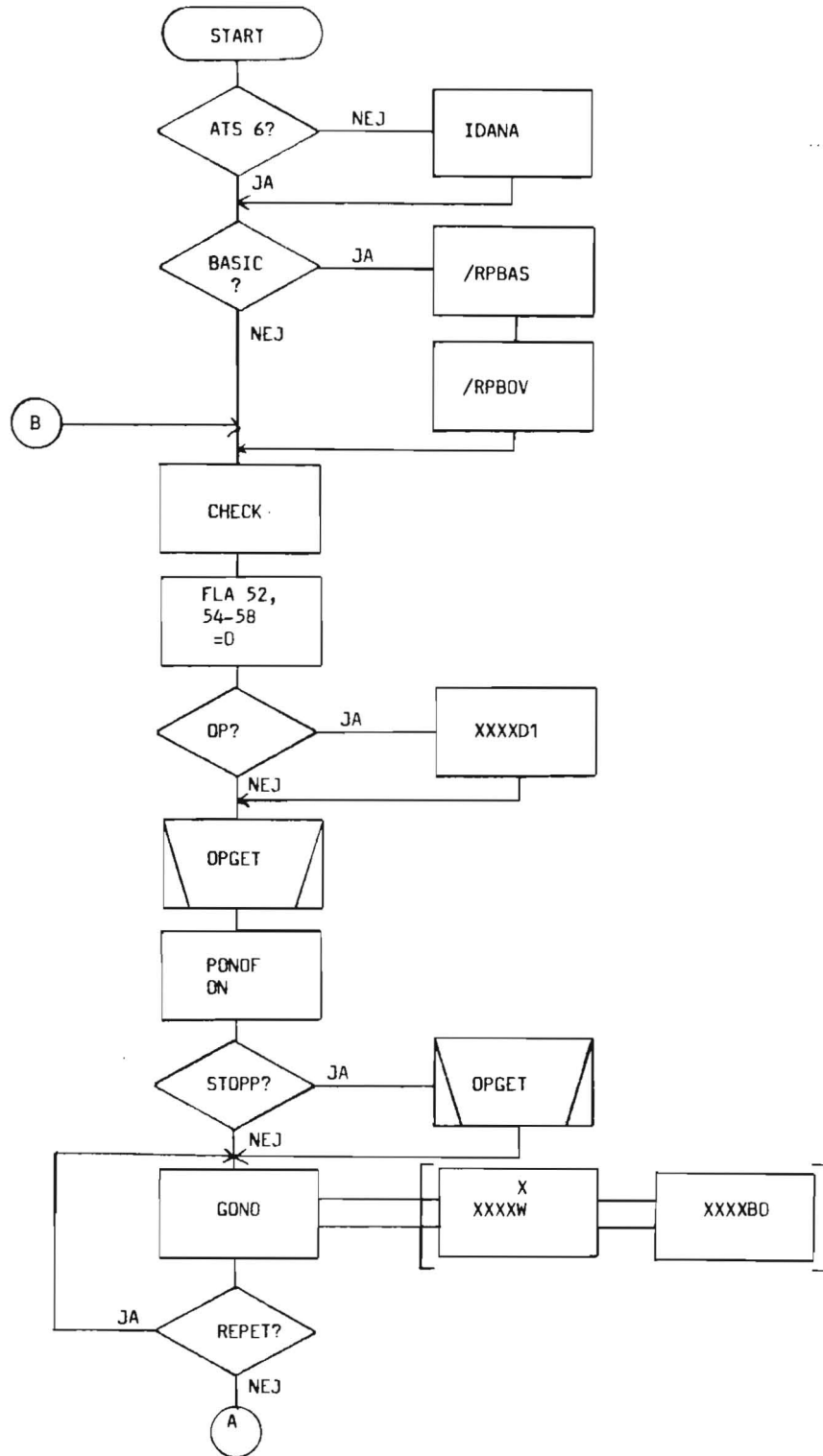
Därefter skrivs en generell operatörsinstruktion, "INSURT UUT",
ut eller den egna XXXXD1-filen visas och programmet går i halt
(OPGET).

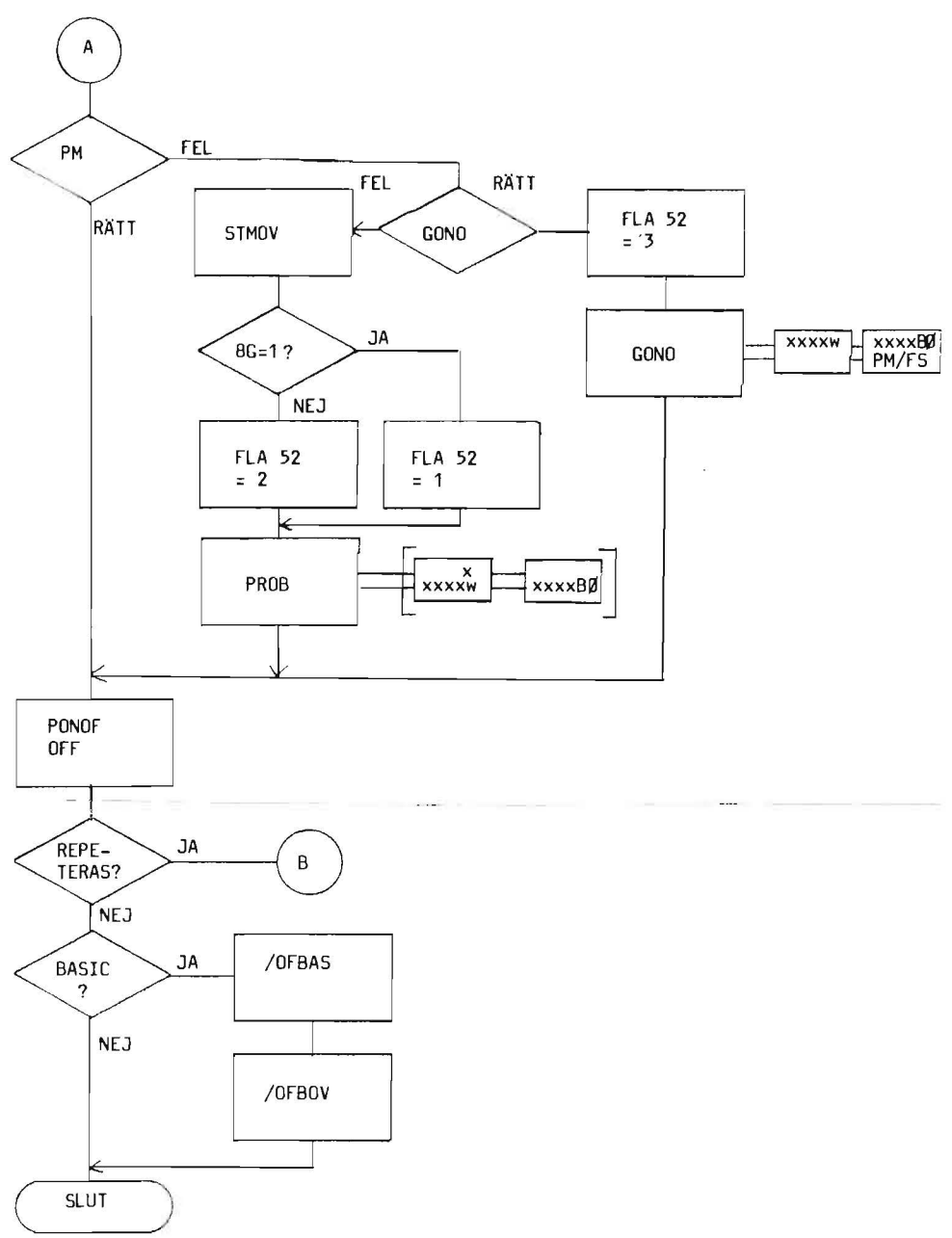
Kraften slås till (PONOF) och PM-testen utförs (GONO, BRANCH-prog).
Resultatet från PM-testen utvärderas. Om PM-testen gått rätt slås
kraften av (PONOF) och testen avslutas.

Om PM-testen gått fel startas FS-testen upp (PROB, BRANCH-prog).
Efter avslutad felsökning slås kraften av (PONOF) och testen av-
slutas.

För mer detaljerad information se blockschema och programlista.

BLOCKSCHEMA TSEQ01





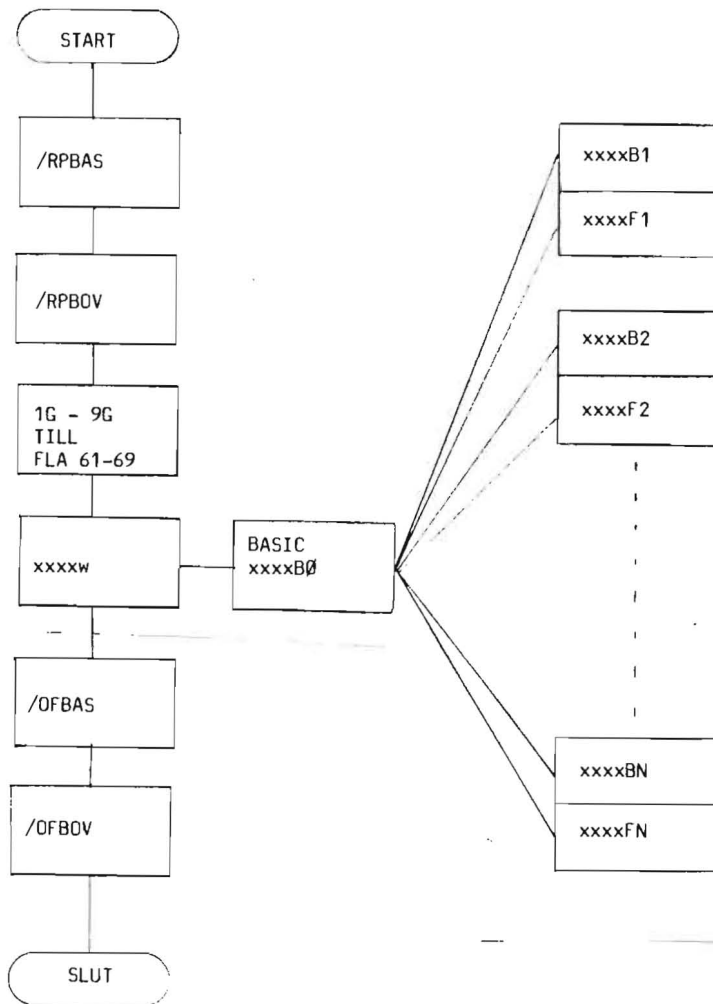

```

0059 :DP,
0060 :IF,2P,NE,0,7
0061 :* CALCULATE NEW TEST #.
0062 :IF,-17P,EQ,1,4
0063 :CA,-18:P,-18P,+,1
0064 :CA,-17:P,-17P,-,1
0065 :CA,-33:P,-33P,+,1
0066 :IF,0,EQ,0,-15
0067 :CA,-33:P,-33P,-,-19P,+,1
0068 :* CHECK IF DIGITAL FAILUR.
0069 :IF,2P,NE,U,6
0070 :* CHECK IF BRANCH.
0071 :IF,-28P,EQ,0,24
0072 :* CHECK IF ANALOG FAILUR.
0073 :RU,SLSSG,0,51,3P
0074 :IF,3P,EQ,1,16
0075 :IF,0,EQ,0,20
0076 :DP,[1 BEGIN DIGITAL FAULT ISOLATION.
0077 :DP,
0078 :RU,STMOV,1G::2G,VPSFD1::-3,*L
0079 :IF,1P,EQ,0,1
0080 ::
0081 :* CHECK IF BRANCH.
0082 :IF,-28P,EQ,0,7
0083 :* CHECK IF STIM.
0084 :IF,-7P,NE,1,3
0085 :* SET FLAG FOR STIM.
0086 :RU,SLSSG,3,52,1
0087 :IF,0,NE,0,2
0088 :* SET FLAG FOR RETURN.
0089 :RU,SLSSG,3,52,2
0090 :RU,PROB,VPSFD1::-3,30,32,0,BACK10::-2,3G::2G,*PAR:33:0
0091 :IF,0,EQ,0,4
0092 :DP,[1 BEGIN ANALOG FAULT ISOLATION.
0093 :DP,
0094 :RU,SLSSG,3,52,3
0095 :RU,GJNO,1G::2G,30,3G::2G,*PAR:33:0
0096 :RU,PONOF,30,33,0
0097 :IF,1P,EQ,0,1
0098 ::
0099 :DP,
0100 :DP,[ EEEEEEE[C&S$B POWER SUPPLIES ARE OFF. [C&S$
0101 :DP,
0102 :DP,
0103 :DP,ENTER OPTION:
0104 :DP,[ 1 - TEST SAME UUT.
0105 :RU,OPGET,,[ 2 - SELECT NEW UUT (ALSO EXIT)?
0106 :IF,1P,EQ,1,-80
0107 :IF,1P,NE,2,-6
0108 :DP,
0109 :* CHECK IF BASIC
0110 :IF,-25P,NE,22304,3
0111 :* OFF BASIC AND OVERLAYS.
0112 :TR,/OFBAS::-2
0113 :TR,/OFBOV::-2
0114 ::

```

4.3 TSEQ02

TSEQ02 tillsammans med BASIC-testen XXXXB0 är det som administrerar körningen av testprogram skrivna i BASIC. (Se fig 2). TSEQ02 ställer upp förutsättningarna för BASIC, lagrar information från /DIR i FLA-cellerna 61-69 och kör uppstartningsprogrammet (XXXXW) för BASIC. Efter avslutad test tar man bort BASIC ur systemet och avslutar testsekvensen.



Blockschema TSEQ02

Fig 2

PROGRAMLISTA TSEQ02

```

0001  :*
0002  :*   TSEQ02 BASIC TEST ATS-11/12
0003  :*
0004  :*   1985-02-13 UL/JM
0005  :*
0006  :TR,/RPBAS::-2
0007  :TR,/RPBOV::-2
0008  :*   SPAR 1E-9G I FLA 61-69
0009  :RU,SLSSG,3,61,1G
0010  :RU,SLSSG,3,62,2G
0011  :RU,SLSSG,3,63,3G
0012  :RU,SLSSG,3,64,4G
0013  :RU,SLSSG,3,65,5G
0014  :RU,SLSSG,3,66,6G
0015  :RU,SLSSG,3,67,7G
0016  :RU,SLSSG,3,68,8G
0017  :RU,SLSSG,3,69,9G
0018  :*   RU XXXXW
0019  :RU,1G::2G
0020  :*   OFF BASIC OCH OVERLAYER.
0021  :TR,/DFBAS::-2
0022  :TR,/DFBOV::-2
0023  ::

```

5. BRANCHNING - UTHOPP FRÅN FASTRACE5.1 Allmänt

Från FASTRACE-programmen GONO och PROB har man möjlighet att hoppa till ett program ("brancha") för att styra testaren via device-subrutiner.

Dessa program ("branchprogram") skrivs i BASIC eller FORTRAN. Vid simuleringen anger man vid vilka testmönster man skall "brancha" ("branchpunkter"). Antalet "branchpunkter" kan vara 1-100, men bör av skäl som redovisas senare hållas så lågt som möjligt.

5.2 Branchning - då branchprogrammet är skrivet i BASIC

Generellt gäller att programmet som man branchar till måste vara skrivet i FORTRAN. I de fall då branchprogrammet är skrivet i BASIC innebär det att man måste införa ett startprogram skrivet i FORTRAN som man branchar till. Startprogrammet startar i sin tur upp i BASIC-programmet.

Startprogrammet kan skapas automatiskt genom att exekvera transferfilen *SKAPA.

Exempel

TR,*SKAPA

ANGE NAMN PÅ BASIC PROGRAM TILL VILKET ETT STARTPROGRAM SKALL SKAPAS. ANGE OCKSÅ CRN DAR STARTPROGRAMMET SKALL SPARAS.

ANVAND FOLJANDE FORMAT

: ,EP01B0,15

:: ,EP01B0,15

Ett FORTRAN program kommer nu att skapas, kompileras och laddas på angiven CRN.

Namnet på FORTRAN programmet skapas ur namnet på BASIC programmet och blir enligt följande.

EP01B0

EP01W

Branchning till program skrivna i BASIC är tidskrävande och man måste räkna med att en branchning tar c:a 5 sek.

Det är därför nödvändigt att antalet branchningar är så få som möjligt.

Om en testmönsterfil har en branchpunkt kan antalet branchningar beräknas enligt följande.

GONO	5 st
BACKTRACE	5 st
PROBE punkter	10 st
Verifiering	<u>5 st</u>
Summa:	25 st

Om testmönsterfilen har 4 branchpunkter blir antalet branchningar 4 ggr så många dvs 100 st.

5.3 Branchning - då branchprogrammet är skrivet i FORTRAN

Här behövs inget startprogram. Samtliga testprogram är nu skrivna i FORTRAN.

Tiderna för branchning blir åtminstone 10 ggr snabbare i det här fallet.

Nackdelen är att programmering tar längre tid och att utprovningen blir svårare.

Detta beror bl.a på att i FORTRAN måste programmet kompileras och laddas, medan man i BASIC redan vid inskrivningen får en syntaxkontroll.

Vid utprovningen kan man i BASIC enklare lägga in brytpunkter och utskrifter och på så vis snabbare avgöra var felet uppträder.

För laddning av branch-program finns en transferfil framtagen,
*LADDA.

Anrop:

::*LADDA,P1,P2,P3,P4

P1 = Programnamn

P2 = Lu eller CRN där den relokerbara modulen finns.

P3 = Lu eller CRN där det laddade programmet ska placeras.

P4 = List-lu

EX. ::*LADDA,EP01X,-48,-48,6

::*LADDA,PS05Y,PS,PS,1

Om man anger ?? som första parameter till *LADDA, fås en beskrivning av anropet på bildskärmen.

EX. ::*LADDA,??

5.4

Praktiska synpunkter

Om man väljer att skriva sina branchprogram i BASIC, som är önskvärt, måste man hålla nere antalet branchningar.

Även i de fall då branchprogrammet är skrivet i FORTRAN bör man ha så få branchpunkter som möjligt. I dessa fall är det lämpligt att först skriva programmet i BASIC. Programmet utprovas och då man anser att utprovningen är klar översätts det till FORTRAN.

5.5. Anrop av devicesubrutiner i FORTRAN

När devicesubrutinerna anropas från FORTRAN anropas de med andra namn än i BASIC.

Dessutom tillkommer en eller 2 extra parametrar.

De två extra parametrarna är LU nummer och felfält.

Sambandet mellan BASIC anrop och motsvarande FORTRAN anrop framgår av nedanstående tabell.

Anrop i BASIC	Anrop i FORTRAN
DINIT ()	XNIT(LU,IERR,)
DFD ()	XDLY(LU,IERR,)
DSREF ()	XTREF(LU,IERR,)
DAREF ()	XWSET(LU,IERR,)
DPREF ()	XPREF(LU,IERR,)
DPSLW ()	XPSWL(LU,IERR,)
DDUT ()	XTDRV(IERR,)
DDIN ()	XTCMP(IERR,)
DSPH ()	XETHI(IERR,)
DSPL ()	XETLO(IERR,)
DTEST ()	XTEST(IERR,)
DPU ()	XPSUP(LU,IERR,)
DSTAT ()	XSTAT(LU,IERR,)
DKON ()	XCONF(LU,IERR,)
DPULS ()	XPULS(LU,IERR,)
DINP ()	XINIT(LU,IERR,)
DTO ()	XTUTO(LU,IERR,)
DSTR ()	XSTR(LU,IERR,)
ANAID ()	XWAID(LU,IERR,)
RELTE ()	XWTST(LU,IERR,)

Anrop i BASIC	Anrop i FORTRAN
RELID ()	XWCID(LU,IERR,)
RELK ()	XWMAP(LU,IERR,)
RELST ()	XODSW(LU,IERR,)
MKANS ()	XCANC(LU,IERR,)
MGDEF ()	XCAND(LU,IERR,)
KOAX ()	XHFSW(LU,IERR,)
DMVAL ()	XWTCH(LU,IERR, ,0)
DRELST ()	XELAY(LU,IERR,)
KLS ()	XLS(LU,IERR,)
KLSU ()	XLSU(LU,IERR,)
LSWRP ()	XWRP(LU,IERR,)
FLA ()	XLA(IERR,)
DUTV ()	XUTV(IERR,)

Parametrarna till respektive devicesubrutin beskrivs ingående i kapitlet subrutinanrop (kap. 4).

Felfältet IERR är ett heltalsfält som skall dimensioneras till 5. LU-numret skall inte "hårdkodas" i devicesubrutinanropen, utan istället skall nedanstående metod användas.

Bestäm session LU för aktuell terminal

ISTN=ISN(IDMY)

Där:

IDMY= En dummy variabel för att tillfredsställa FORTRAN kompilatorn.

ISTN= Erhållen session LU. Noll om den inte hittas.
Noll innebär att programmet inte är i session.

Bestäm LU nummer för enhet.

LUN = LUDV(ISTN, IDVTP, IUNIT)

Där:

ISTN = Terminalens session LU, erhållen från ISN

IDVTP = Device type nummer

IUNIT = Enhetsnummer krävs om det finns flera enheter av samma device type i gruppen (clustret).
Kan hoppas över, default värdet är 1

LUN = Logisk enhet. Om noll eller ett negativt värde erhålles har något fel inträffat.

Enheter som ingår i suctestsystemen och är anropsbara från FORTRAN program har följande device types

Device type Nr	Benämning
1	Bildskärm (2640,2644,2645,2648)
3	Digitaltestenhet (DTU)
11	Switchregisterpanel
31	Switch Controller (HP 9411)
453	KEPCO 488-8

6. UTFORMNING AV BASIC-PROGRAM VID BRANCHNING6.1 Beskrivning av XXXXBO-fil för TSEQ01

För simulerade testobjekt ska TSEQ01 användas. TSEQ-filen startar upp FASTRACE-programmen GONO och PROB. GONO och PROB startar upp XXXXW-filen, som i sin tur startar upp branch-programmet XXXXBO. Endast ett branch-program kan finnas för ett testobjekt. Man måste därför ha ett antal programväxlar i XXXXBO.

Kriterier för programväxling:

- o Ett antal branchpunkter kan finnas i testprogrammet. Om olika saker ska göras i de olika branch-punkterna måste man växla p.g.a branchpunkt. Device-subrutinen DSTAT används för att läsa av vid vilken branch-punkt (testmönster) man befinner sig.
- o När TSEQ01 utförs kan man vid flera tillfällen komma till branch-programmet XXXXBO.
 - o Vid körning av GONO. Vid detta tillfälle ska endast PM-delen av branch-programmet utföras. I TSEQ01 sätts då FLA-cell 52 till 0. I XXXXBO måste FLA-cell 52 läsas av, och om den är 0 ska endast PM utföras.
 - o Vid körning av PROB för digital felsökning. Vid detta tillfälle ska branch-programmet endast utföras om det innehåller någon form av stimulering, initiering e d, som fordras för den digitala felsökningen. Däremot ska inte mätningar utföras. I TSEQ01 sätts FLA-cell 52 till 1 om branch-programmet innehåller stimulering. I XXXXBO måste FLA-cell 52 läsas av, och om den är 1 ska endast stimuleringsdelen utföras.

Om branch-programmet inte innehåller någon stimulering, sätts FLA-cell 52 till 2 i TSEQ01. Vid detta tillfälle läses FLA-cell 52 av redan i XXXXW-programmet, och om den är 2 utförs inte XXXXB0.

- o Vid körning av GONO för analog felsökning. Vid detta tillfälle ska endast felsökningsdelen av branch-programmet utföras. Felsökningen kan läggas i XXXXB0, men man kan också tänka sig att länka till separata felsöknings-tester. I TSEQ01 sätts FLA-cell 52 till 3 om analog felsökning ska utföras. I XXXXB0 måste FLA-cell 52 läsas av och om den är 3 ska endast felsökningsdelen utföras.

Sammanfattning:

FLA 52 = 0: Analog PM
FLA 52 = 1: Stimulering digital felsökning
FLA 52 = 2: XXXXB0 ska ej utföras vid digital felsökning.
FLA 52 = 3: Analog felsökning.

- o I TSEQ01 nollställs system-FLA-cellerna 54-58. Dessa kan sedan användas för programväxling i XXXXB0 efter behov.

6.2 Utvärdering av testresultat

För utvärdering av testresultat i branch-program finns en speciell subrutin framtagen, DUTV.

Anrop:

DUTV (mätvärde, övre gräns, undre gräns, måttenhet, svarsvariabel).

Subrutinen utför följande:

- o Jämför mätvärdet mot hög och låg gräns
- o Sätter svarsvariabeln till
0 vid rätt
1 vid fel
- o Sätter FLA-cell 51 till 1 vid fel. (FLA-cell 51 läses av i TSEQ01 för att avgöra om analog PM gått rätt eller fel).
- o Läser av testnummer från DTU.
- o Ger utskrift på bildskärmen.

Utskriften sker i följande format:

```
XXXXX RÄTT/FEL MV=YY SORT HG=ZZ SORT LG=UU SORT
```

För närmare information om användning av måttenheter, se kap 7.2.3.

7. UTFORMNING AV BASIC - PROGRAM VID ENBART BASIC

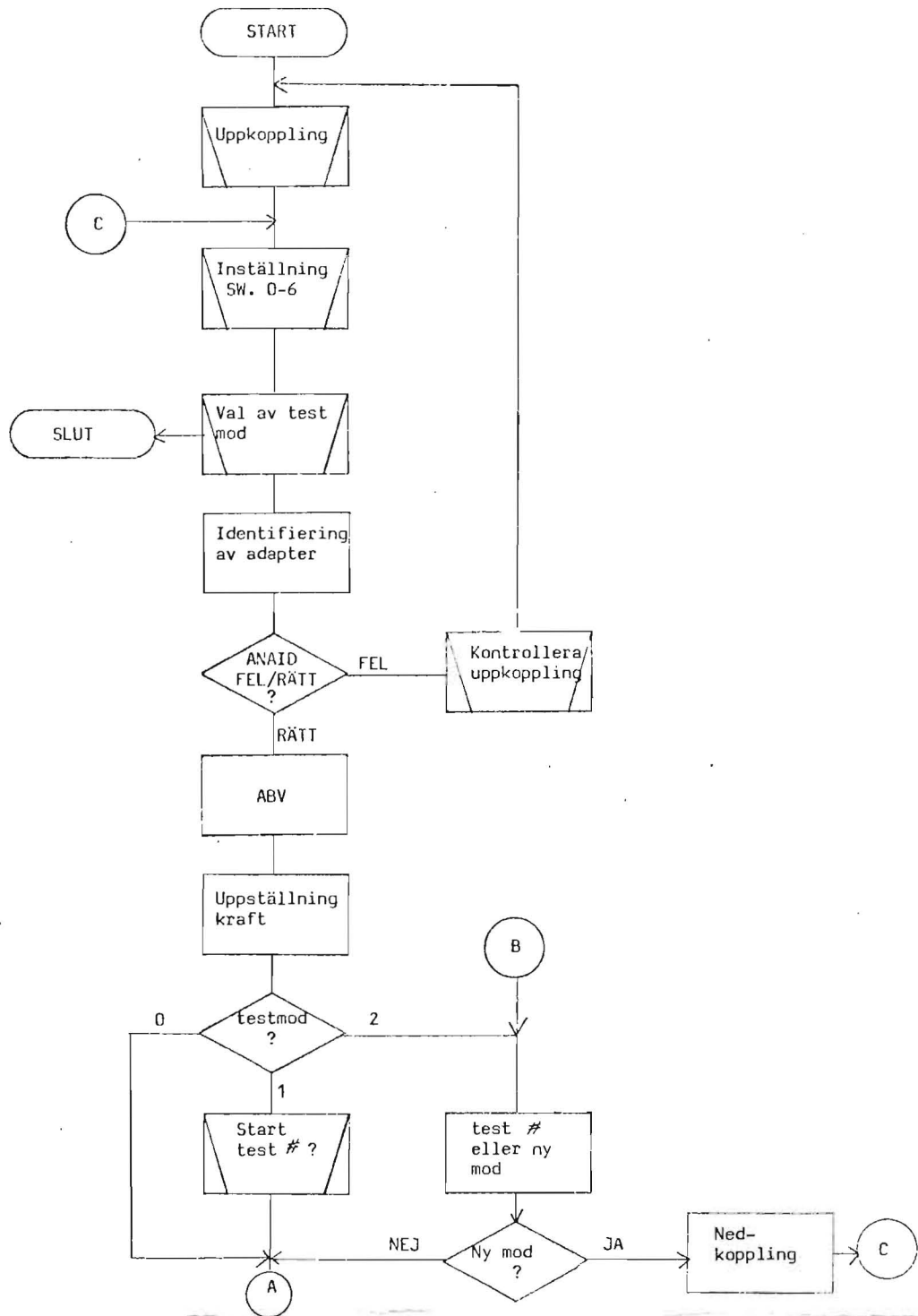
7.1 Beskrivning av XXXXBO-fil för TSEQ02

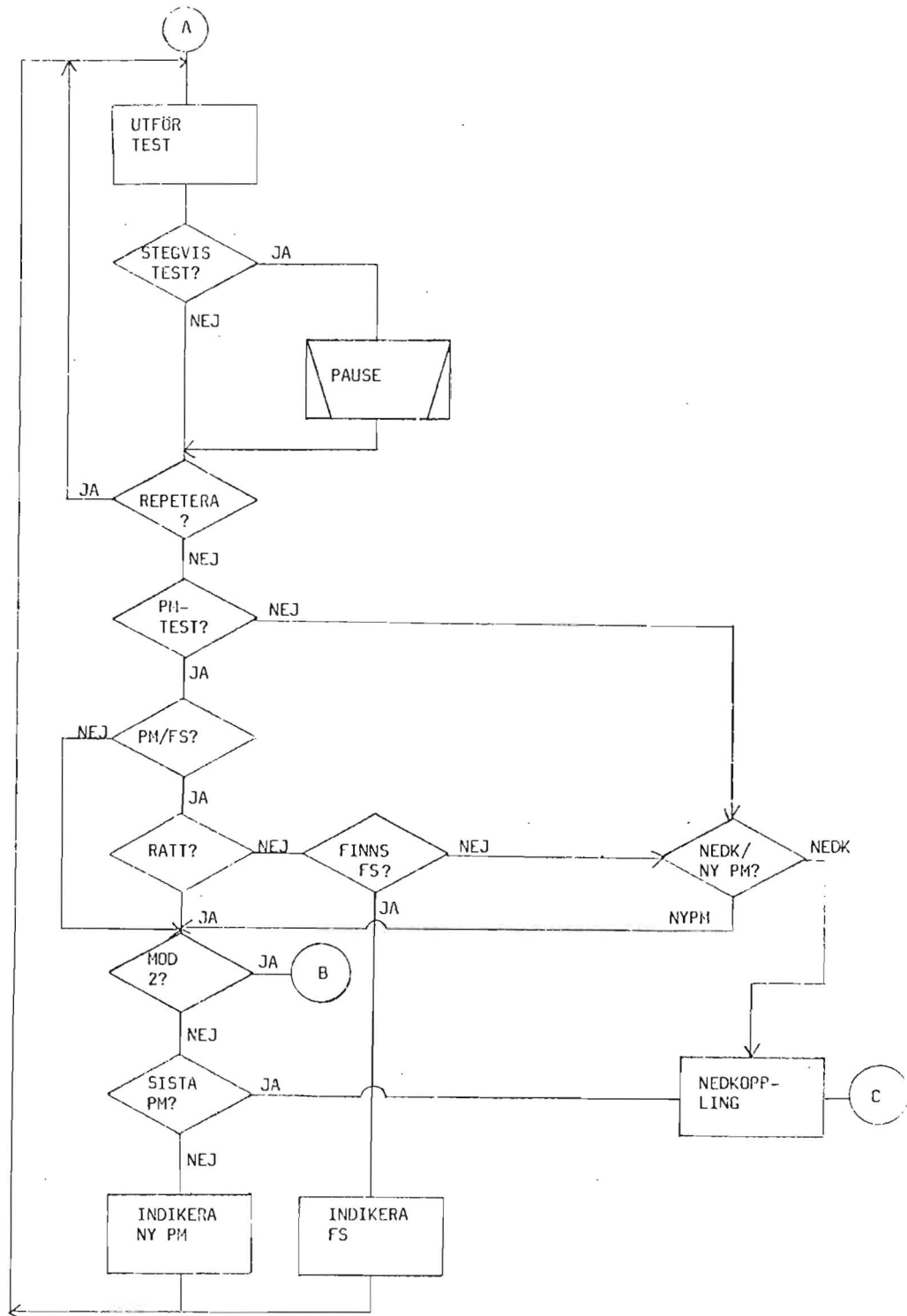
För icke-simulerade testobjekt ska TSEQ02 användas. TSEQ02 startar upp XXXXW-filen, som i sin tur startar upp XXXXBO-filen.

XXXXBO-filen administrerar flödet av de egentliga testerna i testprogrammet. XXXXBO-filen begränsar antalet PM-tester i PM-kedjan till 36 tester. Till varje PM-test kan en FS-test höra, som utförs då PM-testen gått fel.

Eftersom XXXXBO-filen administrerar flödet mellan testerna, ska testerna vara helt fristående och ej länka till varandra.

FLÖDESPLAN FÖR XXXXBO FÖR TSEQ02





Först i XXXXB0-filen ges en operatörsinstruktion för anslutning av testobjektet, och därefter ska operatören ställa in switch 0-6 på switchregisterpanelen, samt välja testmod.

Tre olika testmoder finns:

Mod 0 innebär att testerna utförs sekventiellt med start från första PM-test, dvs hela testprogrammet utförs.

Mod 1 innebär att testerna utförs sekventiellt, men att operatören får välja med vilken PM-test i flödet som han vill starta.

Mod 2 innebär att PM-testerna utförs en och en, där operatören hela tiden väljer vilken test han vill utföra.

Detta innebär att varje PM-test måste vara självförsörjande och gå att utföra separat.

Mod 3 innebär att testningen avslutas och man hoppar tillbaka till TSEQ02.

Efter val av testmod sker adapteridentifiering, och den avlästa koden jämförs med den adapterkod som angivits i /DIR.

Eventuella övriga begynnelsevillkor ska sedan utföras, och därefter ställs kraften upp.

För mod 0 startas sedan första PM-test, medan för mod 1-2 operatören väljer en test som sedan startas.

Då testningen utförts enligt vald mod, sker nedkoppling och därefter kan operatören välja ny testmod, eller avsluta testningen.

I samband med val av testmod ställs switch 0-6 på switchregisterpanelen in. Switch nr 4-6:s inställning påverkar testflödet och läses därför av i XXXXB0-filen. Switch 4 anger om felsökning önskas vid fel, eller om man istället önskar att PM ska fortsätta. Switch 5 anger om en test ska repeteras och switch 6 om man önskar pause mellan varje test.

För att man i XXXXB0-filen ska kunna veta, om en test har gått rätt eller fel, och därmed avgöra vilken nästa test som ska utföras är, lagras UTV information om testresultatet i FLA-cell 60. Denna avläses sedan i i XXXXB0-filen.

PROGRAMLISTA XXXXBO FÖR TSEQ02

```

1  REM V11/12-XXXXBO
2  REM 84-01-30
3  REM GENERELL XXXXBO-FIL FOR TSEQ02
9  REM -----
10 DIM A$(255),B$(255),C$(255)
11 COM DE255]
20 REM M = TESTMOD
30 REM M1 = ANTAL PM-TESTER
40 REM T = ORDNINGSNUMMER FOR TEST SOM SKA UTFORAS
50 REM T1=0: PM-TEST, T1=1: FS-TEST
60 REM A$ = STRANGVARIABEL MED PM-TESTER
70 REM B$ = STRANGVARIABEL MED FS-TESTER
80 LET V=0
90 CALL FLA(1,V,52) FAIL: GOTO 7000
100 REM INLASNING AV DATA
110 READ M1
120 LET Y=INT(M1/8)
125   FOR I=1 TO Y
130     READ A$[(I-1)*56+1,I*56]
135   NEXT I
140 IF M1/8-INT(M1/8)#0 READ A$[Y*56+1,7*M1]
145   FOR I=1 TO Y
150     READ B$[(I-1)*56+1,I*56]
155   NEXT I
160 IF M1/8-INT(M1/8)#0 READ B$[Y*56+1,7*M1]
199 REM -----
200 REM OP. INSTR. FOR ANSLUTNING AV TESTOBJEKT
210 PRINT
220 PRINT "*** ANSLUT ANA NR 1 TILL ATS 12."
230 PRINT "*** ANSLUT TESTOBJEKT TILL ANA."
240 PRINT "*** ANSLUT KABLAGE FRAN TESTOBJEKTETS"
250 PRINT "      KORTRYGGSKONTAKT TILL ANA."
270 PRINT
390 PAUSE
399 REM -----
400 PRINT
410 PRINT "<<< INSTALLNING AV SWITCREGISTERPANEL OCH VAL AV TESTMOD >>>"
420 PRINT
430 PRINT "*** STALL IN SWITCH 0 - 6 PA SWITCREGISTERPANEL ***"
440 PRINT
450 PRINT "*** VALJ TESTMOD ***"
460 PRINT
470 PRINT "      0 = SEKVENTIELL TEST, START FRAN FORSTA PM-TEST."
480 PRINT "      1 = SEKVENTIELL TEST, VALJ FORSTA PM-TEST."
490 PRINT "      2 = ENSTAKA TEST, VALJ PM-TEST."
500 PRINT "      3 = AVSLUTNING TEST."
510 PRINT
520 INPUT M
530 REM MOD 3
540 IF M=3 GOTO 9000
550 REM MOD 0-2
599 REM -----

```

```

600 REM IDENTIFIERING AV ADAPTER
610 CALL FLA(0,X,63) FAIL: GOTO 7000
620 CALL RELST(-1,1,1,1) FAIL: GOTO 7000
630 CALL ANAID(K) FAIL: GOTO 7000
640 IF K=X GOTO 710
650 REM FEL VID ADAPTERIDENTIFIERING
660 PRINT
670 PRINT "*** FEL VID ADAPTERIDENTIFIERING."
680 PRINT "*** KONTROLLERA UPPKOPPLING."
690 PRINT
700 GOTO 200
710 REM IDENTIFIERING AV KORT
799 REM -----
800 REM ALLMANNA BEGYNNELSEVILLKOR FOR TESTPROGRAMMET
1199 REM -----
1200 REM UPPSTALLNING AV KRAFT
1210 CALL KLS(1,5,2) FAIL: GOTO 7000
1220 CALL KLS(2,15,1) FAIL: GOTO 7000
1899 REM -----
1900 REM INITIERING
1910 REM INDIKERA PM-TEST
1920 LET T1=0
1930 LET C$=A$
1999 REM -----
2000 IF M=0 LET T=1
2010 IF M=0 GOTO 3000
2020 PRINT
2030 IF M=1 PRINT "*** VALJ FORSTA PM-TEST."
2040 IF M=2 PRINT "*** VALJ PM-TEST ELLER NY TESTMOD."
2050 PRINT
2060 IF M=2 PRINT " 0 = NY TESTMOD."
2070 PRINT " 1 = AMPLITUD OCH FREKVENNS OSCILLATOR
2080 PRINT " 2 = OFFSET INGANGSFORSTARKARE
2090 PRINT " 3 = KONTROLL TRIGGPULS
2100 PRINT
2800 INPUT T
2810 REM NY TESTMOD
2820 IF T=0 GOTO 7000
2830 PRINT
2840 IF T>M1 PRINT "PM-TEST "T"FINNS EJ."
2850 IF T>M1 GOTO 2020
2999 REM -----
3000 REM STARTA TEST
3010 INVOKE C$[7*(T-1)+1,7*T-1]
3020 REM STEGVIS TEST
3030 IF LSWRP(5)=1 PAUSE
3040 REM REPETERA
3050 IF LSWRP(5)=1 GOTO 3000
3060 REM PM-TEST ELLER FS-TEST?
3070 IF T1=1 GOTO 6500
3080 REM PM/FS-KORNING ELLER PM-KORNING?
3090 IF LSWRP(4)=1 GOTO 4050
4000 REM PM/FS-KORNING
4010 REM PM RATT ELLER FEL?
4020 CALL FLA(0,X,60) FAIL: GOTO 7000
4030 IF X#0 GOTO 6000

```

XXXXB1."
XXXXB2."
XXXXB3."

8211
60.5008.3 utg 7

```

4040 REM PM-RATT
4050 REM OM MOD = 2 VALJER OPERATOR NY PM-TEST.
4060 IF M=2 GOTO 2020
4070 REM FINNS FLER PM-TESTER?
4080 IF T=M1 GOTO 7000
4090 LET T=T+1
5000 LET C$=A$
5010 GOTO 3000
5999 REM -----
6000 REM PM/FS-KORNING, PM-TEST HAR GATT FEL
6010 REM FINNS FS-TEST?
6020 IF B$[7*(T-1)+1,7*T-1]="XXXXXX" GOTO 6500
6030 REM FS FINNS
6040 LET C$=B$
6050 LET T1=1
6060 GOTO 3000
6499 REM -----
6500 REM FS FINNS EJ/ FS REDAN UTFORD
6510 PRINT
6520 PRINT "*** ONSKAS NEDKOPPLING (0)"
6530 PRINT "    ELLER FORTSATT PM (1)";
6540 INPUT X
6550 IF X=0 GOTO 7000
6560 REM FORTSATT PM
6570 LET T1=0
6580 LET C$=A$
6590 GOTO 4050
6999 REM -----
7000 REM NEDKOPPLING
7010 CALL KLS(0,0,0,0) FAIL: GOTO 7020
7020 CALL <VS(0,0) FAIL: GOTO 7030
7030 CALL NOLLST(0) FAIL: GOTO 7040
7040 WAIT (100)
7050 CALL RELST(3,1,1,1) FAIL: GOTO 7480
7480 PRINT
7490 PRINT "<<< NEDKOPPLING UTFORD >>>"
7500 REM VALJ NY TESTMOD
7510 GOTO 400
7999 REM -----
8000 REM DATA
8010 REM ANTAL PM-TESTER
8020 DATA 3
8030 REM PM-TESTER
8040 DATA "XXXXB1,XXXXB2,XXXXB3,"
8110 REM FS-TESTER
8200 DATA "XXXXF1,XXXXXX,XXXXF3,"
8999 REM -----
9000 REM TEST AVSLUTAS, TILLBAKA TILL TSEQ-FIL
9010 END

```


Följande rader i XXXXB0 måste ändras beroende på testobjekt.

1 - 3 : Testidentifiering

200 - 390 : Op.instruktion för anslutning av objektet.

710 - 790 : Ev. kortidentifiering

800 - 1190 : Ev. allmänna begynnelsevillkor

1200 - 1890 : Uppställning av kraft

2070 - 2790 : Lista över PM-tester.

7000 - 7470 : Nedkoppling

8020 : Antal PM-tester

8040 - 8100 : Datasträngar med namn på PM-tester separerade med komma. PM-testerna anges i den ordning de ska utföras. Max 36 tester. 8 tester anges på varje rad.

8200 - 8500: Datasträngar med namn på FS-tester separerade med komma.
Den FS-test som hör ihop med första PM-testen, ska komma först i datasträngen, den som hör ihop med andra PM-testen på andra plats osv. Om en PM-test saknar FS-test ska i datasträngen för FS-tester anges XXXXXX. 8 tester anges på varje rad.

De generella raderna i XXXXB0 ligger på Lu 3 med filnamn GENB0::3.

7.2 UTVÄRDERING AV TESTRESULTAT

7.2.1 Allmänt

Efter erhållande av ett testvärde måste detta utvärderas och resultatet dokumenteras d v s man måste kontrollera om det ligger inom tillåtna gränser samt skriva ut resultatet av utvärderingen på bildskärmen eller radskrivare.

Detta skall ske med UTV-makrot.

Innan testvärdet utvärderas kan det behöva avrundas till ett visst antal decimaler.

Med hjälp av INT-funktionen kan man avrunda tal. Avrundning till närmaste heltal göres enligt:

$$X = \text{INT}(X+.5)$$

$$\text{EX: INT}(2.9+.5)=\text{INT}(3.4)=3.$$

INT-funktionen kan också användas för avrundning till godtyckligt antal decimaler.

$\text{INT}(10^{\wedge}D * X + .5) / 10^{\wedge}D$ avrundar X korrekt till D decimaler.

7.2.2 UTV-makrot

Format:

UTV (testnamn, index, testvärde, hög gräns för testvärde, låg gräns, "sort", felvariabel).

UTV-makrot utför i normalfallet följande:

- o Utskrift av testnr och index. Index är ett tal som anger ordningsnummer för ett testvärde inom en test.

o Utskrift av testvärde och gränsvärden med sort.

o Felvariabeln (parameter 7) sätts:

Vid rätt = 0

Vid fel = 1

Felvariabeln måste sedan testas i testprogrammet för att olika åtgärder ska kunna vidtas vid rätt resp. fel.

o FLA-cell nr 60 sätts:

Vid rätt = 0

Vid fel = index för sista felaktiga testvärde inom en test.

FLA-cell nr 60 kan sedan testas i felsökningstesten, om olika åtgärder ska vidtas beroende på vilken mätning som gått fel i PM-testen.

UTV-makrots funktion kan styras av 5 switchar på switchregisterpanelen enligt följande:

Switch 0:	noll-ställd:*)	Utskriften erhålles på bildskärm
	ett-ställd:	Utskriften erhålles på bildskärm och radskrivare
Switch 1:	noll-ställd:	Utskrift av endast testresultat
	ett-ställd:	Utskrift av testresultat med gränsvärden
Switch 2:	noll-ställd:	Utskrift av endast felresultat
	ett-ställd:	Utskrift av samtliga resultat
Switch 3:	noll-ställd:	Programmet fortsätter efter utskrift av fel
	ett-ställd:	Programmet går till PAUSE efter utskrift av fel.

Switch 4: noll-ställd: PM/FS-körning.
Felvariabeln sätts som beskrivits
ovan

ett-ställd: Enbart PM-körning.
Vid PM-fel sätts variabeln till 0.
Detta för att programmet ska fort-
sätta med nästa PM trots det fel-
aktiga resultatet.
I övrigt enl. ovan.

I normalfallet är alltså switch 0, 1 och 2 ett-ställda och switch 3 och 4 noll-ställda.

Före körning av testen står switcharna i normalläget, ändring av detta måste ske genom order från testprogrammet till operatören via DISPLAY-satser.

*) En switch är noll-ställd när den står i nerläge.

Utskriften sker i följande format:

Testresultat med gränsvärden:

XXXXXX.X RÄTT/FEL MV=YY SORT HG=ZZ SORT LG=UU SORT

Endast testresultat:

XXXXXX.X RÄTT/FEL MV=YY SORT

Felaktiga resultat skrivs ut med rött.

7.2.3

ANVÄNDNING AV MÄTTENHETER

Måttenheten, som ska programmeras i UTV-makrot par 6, ska följa de normer som givits ut av Sveriges standardiseringskommision. Som en vägledning hur man i BASIC betecknar enheten och eventuella prefix ges här några exempel.

STORHET	ENHET	
<u>Benämning</u>	<u>Benämning</u>	<u>Basicbeteckning</u>
Elektrisk spänning	Volt	V
Elektrisk ström	Ampere	A
Frekvens	Hertz	Hz
Effekt	Watt	W
Tid	Sekund	S
Vinkel	Radian	R
Motstånd	Ohm	OHM
Kapacitans	Farad	F

TALFAKTOR	PREFIX	
<u>Benämning</u>	<u>Benämning</u>	<u>Basicbeteckning</u>
1E12	Tera	T
1E9	Giga	G
1E6	Mega	M
1E3	Kilo	K
1E2	Hekto	H
1E1	Deka	DA
1E-1	Deci	D
1E-2	Centi	C
1E-3	Milli	M
1E-6	Mikro	U
1E-9	Nano	N
1E-12	Piko	P

8. UTFORMNING AV BASIC - PROGRAM ALLMÄNT8.1 Testidentifiering

Varje test ska inledas med REM-instruktioner för identifiering av testen.

REM-instruktionerna ska innehålla underhållsnivå, testnamn, datum och testbenämning.

De inledande instruktionerna får då följande utseende.

1. REM underhållsnivå - testnamn
2. REM datum
3. REM testbenämning

Underhållsnivå: V6 = ATS 6
 V11 = ATS 11
 V12 = ATS 12

Exempel:

1. REM V6 - EP01B1
2. REM 82-08-31
3. REM TEST AV FÖRSTÄRKNING

9. OPERATÖRSKOMMUNIKATION9.1 Allmänt

Under ett testprograms gång erfordras att operatören gör vissa ingripanden. Programmeraren måste se till att operatören blir upplyst om vad han skall göra och när han skall göra det. Detta sker genom utskrift av meddelanden på bildskärmen.

Då operatören kör testprogrammet skall om så behövs en operatörsinstruktion finnas tillgänglig.

Den skall innehålla de instruktioner som ej är lämpade för utskrift på bildskärmen. I dessa fall refererar meddelandet på bildskärmen till var i operatörsinstruktionen åtgärden finns beskriven. (Meddelandet kan i vissa fall referera till andra PRU dokument).

9.2 Basic-satser för operatörskommunikation

Kommunikation med operatören skall normalt ske med följande satser:

PRINT Utskrift av text på bildskärmen eller radskrivare.

INPUT Inmatning av tal från bildskärmen.

PAUSE Ställer testprogrammet i PAUSE-mod.

9.2.1 PRINT-satsen

Används för utmatning av text från testprogrammet till bildskärmen. Då utskriften ger order till operatören skall texten föregås av 3 stjärnor (***) .

```
EX PRINT"*** STÄLL BRYTARE A1 I LÄGE 0"
```

Vid upplysningar till operatören skall texten omges av 3 vinkelparentester (.....).

```
EX PRINT"<<< NEDKOPPLING UTFÖRD >>> "
```

9.2.2 INPUT-satsen

Används för att tilldela variabler värden, som matas in från bildskärmen under pågående körning. Då programmet kommer till en INPUT-sats skrivs ett frågetecken ut på bildskärmen.

De värden, som skall tilldelas variablerna i INPUT-satsen skrivs in på bildskärmen, åtskilda av mellanslag, och inmatning sker genom att trycka på RETURN-knappen.

Om INPUT-satsen innehåller flera variabler än som skrivits in skrivs ytterligare ett frågetecken ut på bildskärmen och nya tal matas in av operatören.

Proceduren upprepas tills samtliga variabler i INPUT-satsen tilldelats värden.

9.2.3 PAUSE-satsen

Används för att ställa programmet i PAUSE-mod efter order (via PRINT "*** ") till operatören om ett ingripande.

Då programmet kommer till PAUSE-satsen skrivs texten PAUSE ut på bildskärmen och programmet stannar.

Då operatören utfört sin åtgärd startas programmet genom att skriva GO på bildskärmen.

9.3 Programmering av utskrifter

Då utskriften är lång och förekommer i flera tester skrivs den lämpligen som en test vilken inkallas vid behov via SUB-satser.

Om utskriften förekommer flera gånger i samma test skrivs den lämpligen som en SUB-rutin inom testen, vilken anropas via GOSUB-satsen.

9.4 D1-filer

Vid digitaltest har man möjlighet att ge operatörsinstruktioner i en XXXXD1-fil.

Den kommer att utföras vid utförandet av TSEQ01 före anslutning av testobjektet.

D1-filen ska vara en transferfil som ger utskrifter på bildskärmen

Exempel på D1-fil:

```
: DP
: DP,   *** STALL SWITCHARNA (KOMPLEMENTPLATS IC32)
: DP,   *** SWITCH 2,4,6,8 I LAGE OPEN
: DP
::
```

10

NEDKOPPLING AV TESTARE

Detta kapitel gäller ATS 11/12.

För att undvika skador på testutrustning och testobjekt, ska nedkoppling ske i följande ordning.

- 100 KLS (0,0,0,0)
- 110 KVS (0,0)
- 120 NOLLST (0)
- 130 WAIT (1000)
- 140 RELST (3,1,1,1)

Behovet av väntetid före RELST varierar beroende på belastning, men en väntetid på 1 sekund räcker i de flesta fallen.

För nedkoppling av ATS 11 och ATS 12 finns ett speciellt nedkopplingsprogram framtaget, NEDKO, som kan köras från FILE MANAGERN.

: RU, NEDKO

Detta kan användas under utprovningstiden, då man vill förvissa sig om att testaren är nedkopplad.

Programmet ger följande felkoder:

- FELKODER:
- 8 = FEL VID I/O MOT WTK 172 #1
 - 16 = FEL VID I/O MOT WTK 172 #2
 - 32 = FEL VID I/O MOT SOLARTRON 1172
 - 64 = FEL VID I/O MOT WTK 859 #1
 - 128 = FEL VID I/O MOT WTK 859 #2
 - 256 = FEL VID I/O MOT TRANSMAGNETIC
 - 512 = FEL VID I/O MOT DTU
 - 1024 = FEL VID I/O MOT DANA 1515
 - 2048 = FEL VID I/O MOT KEPCO
 - 4096 = FEL VID I/O MOT ELGAR
 - 8192 = FEL VID I/O MOT HP9411A

Vid fel vid I/O mot flera instrument adderas felkoderna.
Fel vid I/O mot LSE HP 6131C ger ingen felkod.

11. TRAP - FUNKTIONER11.1 Allmänt

TRAP förekommer endast i ATS 11/12. Det finns 4 saker i systemen som kan ge TRAP-avbrott nämligen.

- Gröna knappen
- KEPCO kraftenheterna
- HP 6131 likspänningenheter
- Korskopplingsfältet

Beträffande användningen av TRAP funktioner i BASIC program hänvisas till befintlig BASIC manual.

11.2 Gröna knappen

Den gröna knappen är placerad till höger om korskopplingsfältet. Den är knuten till TRAP 1.

Ingen särskild funktion är reserverad för denna knapp, utan det är upp till testprogrammeraren att utnyttja den på lämpligt sätt.

11.3 KEPCO - Kraftenheter

För KEPCO-aggregaten gäller att ett TRAP avbrott erhålles då programmerad strömgräns överskrids.

För KEPCO s del är följande TRAP celler aktuella.

Kraftenhet nr 1 är knuten till TRAP 2

-"	2	-"	-"	3
-"	3	-"	-"	4
-"	4	-"	-"	5
-"	5	-"	-"	6
-"	6	-"	-"	7
-"	7	-"	-"	8
-"	8	-"	-"	9

Endast ATS 12

Endast ATS 12

11.4 HP 6131 - Likspänningsenheter

För HP 6131 gäller att ett TRAP avbrott erhålles då programmerad strömgräns överskrids.

För HP 6131 får följande TRAP celler aktuella

HP 6131 Nr 1	är knuten till	TRAP 12	
-"-	2	-"-	TRAP 13
-"-	3	-"-	TRAP 14 Endast ATS 12

11.5 Korskopplingsfältet

Från korskopplingsfältet erhålles TRAP avbrott då testadaptern bortkopplas eller ansluts.

För att få avbrott krävs att en bygling görs i adaptern.

För ytterligare information se instruktion för adapterframtagning (PVA-R190:V11/12).

TRAP 15 är knuten till korskopplingsfältet.

12. TESTAID - FASTRACE

12.1 ROMRD

ALLMÄNT

ROMRD är ett program som i första hand används för att via DTU:n modellera ROM och avläsa ROM-innehåll, men det kan också användas för alla kombinatoriska integrerade kretsar vars pinn-antal ligger inom programmets ram.

Indata till programmet är information om kretsens pinnar.

Utdata från programmet är en fil med kretsbeskrivning för SIGLIST (*NET).

Listning ger minnesinnehåll i form av en oktal sanningstabell.

Maximal kretsstorlek är:

12 adressbitar
16 utdatabitar
6 aktiva "low enables"
6 aktiva "high enables"
5 kraftenheter
4096 ord * 16 bitar (65536 bitar)

ANROP

RU,ROMRD,CRT,INDATA,LISTNING,UTDATA

CRT

Här anges session-Lu för terminalen.
Det är alltid 1.

INDATAAllmänt

Indata till programmet ges i form av ett antal styrord, beskrivna nedan.

Dessa kan skrivas in direkt från terminalen (INDATA=1) och då kontrolleras varje rad vid inskrivningen. Vid fel ignoreras raden. Efter en grupp korrekta rader utförs programmet enligt indata.

Indata kan också ges från en fil (INDATA=filnamn). Då kontrolleras samtliga rader, och om inga fel upptäcks utförs programmet enligt indata. Om fel upptäcks avbryts programmet.

Styrorden *A, *D, *H, *L kan endast användas en gång. Orsaken är att alla pinnar tillhörande ett styrord måste ges på samma rad. På varje rad kan anges max 256 karaktärer. Om styrorden *T eller *P upprepas, kommer sist inmatade rad att användas. Styrordens ordning saknar betydelse. En pinne kan endast användas en gång.

Beskrivning av styrorden

Obligatorisk del av styrorden har strukits under.

*TITLE, ROMID::NN, KOMMENTAR

ROMID - Filnamn för *NET
NN - Kretsens fysiska antal pinnar 0<NN<41

*POWER, V1:A1, V2:A2, V3:A3, V4:A4, V5:A5

V1 - Spänning i Volt för kraftenhet 1.
A1 - Ström i Ampere för kraftenhet 1.
V2 - Spänning i Volt för kraftenhet 2.
A2 - Ström i Ampere för kraftenhet 2.
osv.

Spänning och ström ges som heltal.
Både spänning och ström måste anges.
Data för kraftenheterna måste ges i rätt ordning. Om kraftenhet 2 används, måste 0:0 anges för kraftenhet 1, osv.
Om *P utelämnas kommer kraftenheterna ej att strobos på.

*COMMENT (upp till 256 karaktärer)

Alla kommentarrader ignoreras.

*ADRESS, <PINN-LISTA>

<PINN-LISTA>=DTU-pinne(:ROM-pinne),...,DTU-pinne(:ROM-pinne)

DTU-PINNE = Anslutning till DTU (1-360)
ROM-PINNE = Fysisk pinne på kretsen ansluten till DTU-pinne (1-40). Om ROM-pinne utelämnas antar ROM-pinne samma värde som DTU-pinne. ROM-pinne kan då anta värden >40.

Mellan 3 och 12 adress-pinnar måste specificeras. Pinnarna specificeras från mest till minst signifikanta bit.

*DATAOUT, <PINN-LISTA>

Mellan 1 och 16 utdata-pinnar måste specificeras. Pinnarna specificeras från mest till minst signifikanta bit.

*HIGHENABLE, <PINN-LISTA>

Mellan 0 och 6 pinnar kan specificeras.
Dessa pinnar hålls höga under läsningen.

*LOWENABLE, <PINN-LISTA>

Mellan 0 och 6 pinnar kan specificeras.
Dessa pinnar hålls låga under läsningen.

*END

Avslutar indata till programmet. Även EOF eller CNTL D accepteras som *E.

ABORT

Avslutar programmet.
Från RTE-promten kan även kommandot BR användas.

LISTNING

I denna parameter anges var listning av minnesinnehåll önskas.
Lu-nummer eller filnamn anges.

Format för minnesinnehåll.

ADDR ROM DATA -> (MSB TO LSB FOR EACH WORD)

0000: 037 116 377 377 100 000 (16 WORDS)

0020: 111 325 065

OR

0000: 177777 153337 000543 (8 WORDS)

0010: 033333 111666

UTDATA

I denna parameter anges filnamn där kretsbeskrivning för SIGLIST (*NET) ska placeras.

FELKODER

Vid fel fås utskriften

ROMRD ERROR XX

XX är en felkod med följande betydelse:

- 1 - Felaktig syntax eller felaktig karaktär
- 2 - Felaktigt styrord
- 3 - Styrord redan använt
- 4 - >40 ROM-pinnar
- 5 - <3 adress-pinnar specificerade
- 6 - >12 adress-pinnar specificerade
- 7 - Inga utdata-pinnar specificerade
- 8 - >16 utdata-pinnar specificerade
- 9 - DTU-pinn-nummer >360
- 10 - DTU-pinn-nummer <1
- 11 - >5 krafterheter
- 12 - Felaktigt data för krafterhet
- 13 - >6 "high eller low enables" specificerade
- 14 - Felaktigt "keyword"
- 15 - Felaktigt Lu-nummer
- 16 - Fel vid filaccess (FMP-fel ges också)
- 17 - DTU-pinn-nummer redan använt
- 18 - ROM-pinn-nummer redan använt
- 19 - ROM-pinn-nummer <1
- 20 - ROM-pinn-nummer >40

EXEMPEL

INDATA

```
* ROM DEFINITION FOR INTEL 3624 TYPE ( 91073A MICROCODE PROM SET)
*T,U101-A:24
*A,53:23,31:1,32:2,33:3,34:4,35:5,36:6,37:7,38:8
*D,39:9,40:10,41:11,43:13,44:14,45:15,46:16,47:17
*H,48:18,49:19
*L,50:20,51:21
*P,5:1
*E
```

LISTNING

DATA FOR U101-A

```
0000: 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377
0020: 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377
0040: 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377
0060: 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377
0100: 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377 377
```

.
.

.

.

```
0700: 143 351 147 067 160 120 327 367 031 020 020 351 020 020 211 200
0720: 247 037 027 167 031 020 020 020 037 020 040 200 020 317 020 053
0740: 340 020 327 153 020 327 073 020 024 137 020 120 120 024 020 120
0760: 060 020 057 037 267 327 060 213 020 267 020 020 020 037 377 377
```

UTDATA

*NET U101-A (24)\$

*INPUTS

C(23, 01, 02, 03, 04, 05, 06, 07, 08)\$

C(18, 19)\$

C(20, 21)\$

*OUTPUTS

C(09, 10, 11, 13, 14, 15, 16, 17)\$

*GATES

C(CONN) 053.23 031.01 032.02 033.03 034.04 035.05 036.06

037.07 038.08 039.09 040.10 041.11 043.13 044.14

045.15 046.16 047.17 048.18 049.19 050.20 051.21 \$

D(DEC) ADD.01 053.10 031.09 032.08 033.07 034.06 035.05

036.04 037.03 038.02 \$

H(NAND) HEN.01 048.02 049.03 \$

L(OR) LEN.01 050.02 051.03 \$

D0039(OR) 039.01 R039.02 HEN.03 LEN.04 \$

R0039(ROM/

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

00000000 00000000 00010001 10000001 00000011 00100100 00100000 11110100

00101100 00001000 11111001 10000010 00000110 11000110 00000001 01001000

00000110 10000000 01101001 00000010 00001010 10000100 10001100 01000000

01000011 00010011 10000000 00010100 10100100 00000000 00001101 01000011

) R039.01 ADD.02 \$

D0040(OR) 040.01 R040.02 HEN.03 LEN.04 \$

R0040(ROM/

D0047(OR) 047.01 R047.02 HEN.03 LEN.04 \$

R0047(ROM/

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

11111111 11111111 11011001 10011111 01111011 00101101 00000101 00010010

11111000 01000110 00100011 00000110 01110000 11001100 10110010 01001110

01101001 10010001 10111000 00001010 01001001 01010011 10001110 10101110

11110011 10010010 11111000 10000101 00110110 01000000 00111101 01000111

) R047.01 ADD.02 \$

*END