

SKRYPTY DLA SZKÓŁ WYŻSZYCH

POLITECHNIKA ŁÓDZKA

ANDRZEJ NAPIERAŁSKI

**KOMPUTEROWE PROJEKTOWANIE
UKŁADÓW ELEKTRONICZNYCH**

**BIBLIOTEKA
POLITECHNIKA ŁÓDZKA**

8387/8



200-008387-00-0

ŁÓDŹ 1990

NAKŁADEM POLITECHNIKI ŁÓDZKIEJ



Ze zbiorów Biblioteki Głównej Politechniki Łódzkiej

**Skrypt jest przeznaczony dla studentów studium dziennego
kierunku Elektronika**

Recenzent: prof. dr hab. Zdzisław Korzec



Redaktor: Danuta Jagiello-Gałęba

Redaktor techniczny: Ewa Guzek

Korekta: Anna Waldhauser

WYDAWNICTWO POLITECHNIKI ŁÓDZKIEJ

93-005 Łódź, ul. Wólczańska 219

**Nakład 300+40 egz. Ark. wyd. 14,78. Ark. druk. 14,75+3 wklejki.
Papier offset. kl. III 70 g. 61×86
Maszynopis wpłynął do Redakcji 27.XII.1985 r. Oddano do składu 28.V.1986 r.
Podpisano do druku 24.IX.1990 r. Druk ukończono w listopadzie 1990 r.
Zamówienie 101/86. Cena zł 5912,-
Wykonano w Zakładzie Poligraficznym PŁ, 93-005 Łódź, ul. Wólczańska 223**



SPIS TREŚCI

PRZEDMOWA.	7
1. WSTĘP.	8
2. PRZEGLĄD PROGRAMÓW KOMPUTEROWEJ ANALIZY UKŁADÓW ELEKTRONICZ- NYCH	10
3. IDENTYFIKACJA PARAMETRÓW MODELI PRZYRZĄDÓW PÓŁPRZEWODNIKOWYCH	16
3.1. Identyfikacja parametrów modelu diody półprzewodniko- wej	16
3.1.1. Modelowanie charakterystyki statycznej złącza. . .	18
3.1.2. Określenie parametrów rewersyjnych	21
3.1.3. Określenie pojemności dyfuzyjnej i złączonej diody.	22
3.1.4. Doświadczalna weryfikacja dokładności wyznaczo- nych parametrów modelu diody	26
3.2. Identyfikacja parametrów modelu diody Zenera.	27
3.3. Identyfikacja parametrów modelu tranzystora bipolar- nego.	33
3.4. Identyfikacja parametrów tranzystora złączonego FET . . .	39
4. OPIS JĘZYKA WEJŚCIOWEGO PROGRAMU NAP2.	41
4.1. Charakterystyka programu NAP2	41
4.1.1. Organizacja programu NAP2.	42
4.1.2. Zakres zastosowań programu NAP2.	46
4.1.3. Struktura danych wejściowych	47
4.2. Opis wprowadzenia danych do programu NAP2	49
4.3. Zdania opisu obwodu.	51
4.3.1. Elementy bierne	51
4.3.2. Źródła sterowane	54
4.3.3. Szeregowe dołączanie elementów	58
4.3.4. Równoległe dołączenie elementów.	59
4.3.5. Sprzężenia wzajemne	60
4.3.6. Instrukcja parametrowa	61
4.3.7. Instrukcja funkcyjna	62
4.3.8. Wartości funkcjonalne.	73



4.3.9. Sprzężenia nieliniowe	74
4.3.10. Nieliniowe elementy magazynujące energię.	75
4.3.11. Elementy magazynujące energię zmieniające się w czasie	76
4.4. Modele wbudowane w program i makromodele.	76
4.4.1. Dioda	77
4.4.2. Tranzystor bipolarny	81
4.4.3. Tranzystor złączowy JFET	93
4.4.4. Tranzystor MOSFET.	99
4.4.5. Bramki TTL	102
4.4.6. Bramki ECL	102
4.4.7. Wzmacniacz operacyjny.	102
4.4.8. Tyrystor	110
4.5. Modele deklarowane przez użytkownika.	117
4.6. Instrukcje wymiany i usuwanie elementów obwodu.	119
4.7. Zdania sterujące	121
4.7.1. Korzystanie z bibliotek modeli	121
4.7.2. Rozpoczęcie obliczeń	126
4.7.3. Drukowanie informacji dodatkowych.	126
4.7.4. Przechowywanie informacji o obwodzie w zbiorze po- mocniczym	127
4.7.5. Instrukcja *MODIFY i *TOLE	128
4.7.6. Zmiana parametru	131
4.7.7. Sterowanie analizą i wyjściem.	132
4.7.8. Optymalizacja.	141
4.7.9. Analiza Fouriera	142
4.7.10. Instrukcja *RESET	143
4.7.11. Instrukcja *RUN. Parametry sterujące przebie- giem analizy.	144
4.7.12. Ustawianie tytułu wydruku	148
4.7.13. Rozkaz *END.	148
5. OPIS JĘZYKA WEJŚCIOWEGO PROGRAMU PAUT2	149
5.1. Charakterystyka programu PAUT2.	149
5.1.1. Organizacja programu PAUT2	151
5.1.2. Zakres zastosowań programu PAUT2	152
5.2. Opis wprowadzania danych do programu PAUT2.	152
5.3. Zdania opisu obwodu i modele wbudowane	153



5.3.1. Elementy bierne	153
5.3.2. Źródła	159
5.3.3. Przełączniki	160
5.3.4. Diody i tyrystory	162
5.3.5. Transformatory	166
5.4. Funkcje	174
5.5. Zdania sterujące	177
5.5.1. Instrukcje wyboru analizy i wyjścia	177
5.5.2. Analiza Fouriera	180
5.5.3. Instrukcja X TIME	181
5.5.4. Instrukcja X PARA	184
5.5.5. Instrukcja X RUN. Parametry sterujące wydrukiem wy- ników pośrednich	186
5.5.6. Rozpoczęcie i zakończenie obliczeń	189
5.6. Dynamiczna rezerwacja pamięci	189
5.7. Uwagi praktyczne	191
5.8. Przykład zastosowania programu PAUT2	192
6. OPIS PROGRAMU MINUIT	196
6.1. Wprowadzenie	196
6.2. Karty danych	197
6.3. Karty rozkazów	201
6.3.1. Rozkaz CALL FCN	201
6.3.2. Rozkaz CONTOUR	202
6.3.3. Rozkaz COVARIANCE	202
6.3.4. Rozkaz END	203
6.3.5. Rozkaz END RETURN	203
6.3.6. Rozkaz EXIT	203
6.3.7. Rozkaz ERROR DEF	203
6.3.8. Rozkaz FIX	204
6.3.9. Rozkaz GRADIENT	204
6.3.10. Rozkaz HELP	205
6.3.11. Rozkaz HESSE	205
6.3.12. Rozkaz IMPROVE	205
6.3.13. Rozkaz LIMITS	205
6.3.14. Rozkaz MATOUT	206
6.3.15. Rozkaz MIGRAD	206
6.3.16. Rozkaz MINIMIZE	207
6.3.17. Rozkaz MINOS	207



6.3.18. Rozkaz PARAMETER.	208
6.3.19. Rozkaz PRINTOUT	208
6.3.20. Rozkaz PUNCH.	209
6.3.21. Rozkaz RELEASE.	209
6.3.22. Rozkaz RESTORE.	209
6.3.23. Rozkaz SCAN	209
6.3.24. Rozkaz SEEK	210
6.3.25. Rozkaz SET PARAM.	210
6.3.26. Rozkaz SIMPLEX.	210
6.3.27. Rozkaz STANDARD	211
6.4. Procedury dołączane przez użytkownika. Procedura FCN (NPAR, G, F, X, IFLAG).	211
6.5. Wydruk wyników	213
6.5.1. Wydruk kart danych.	213
6.5.2. Wydruk kart rozkazów.	214
6.5.3. Normalny poziom wydruku (PRINTOUT=1).	214
6.5.4. Wydruk macierzy kowariancji i macierzy korelacji błędu	215
6.5.5. Rysowanie poziomic.	215
6.6. Przykład użycia programu MINUIT.	215
7. WYKORZYSTANIE PROGRAMÓW NAP2, PAUT2 I MINUIT POD KONTROLĄ SY- STEMU OPERACYJNEGO GEORGE 3	220
7.1. Obliczenia z wykorzystaniem programu NAP2.	220
7.2. Obliczenia z wykorzystaniem programu PAUT2	228
7.3. Obliczenia z wykorzystaniem programu MINUIT.	228
LITERATURA	231



PRZEDMOWA

Zagadnienia omawiane w skrypcie są wykładane w ramach przedmiotu "Analiza i Projektowanie Komputerowe Układów Elektronicznych" na semestrze IV i V oraz obieralnego przedmiotu "Wybrane Zagadnienia Projektowania Komputerowego" na semestrze VII i VIII.

Materiał zawarty w skrypcie nie wyczerpuje oczywiście całości wykładów, lecz ma na celu jedynie ułatwienie słuchaczom posługiwania się pakietem programów wykorzystywanych podczas zajęć.

Programy przedstawione w niniejszym skrypcie mogą być również używane przez dyplomantów i doktorantów oraz pracowników Wydziału Elektrycznego PŁ.

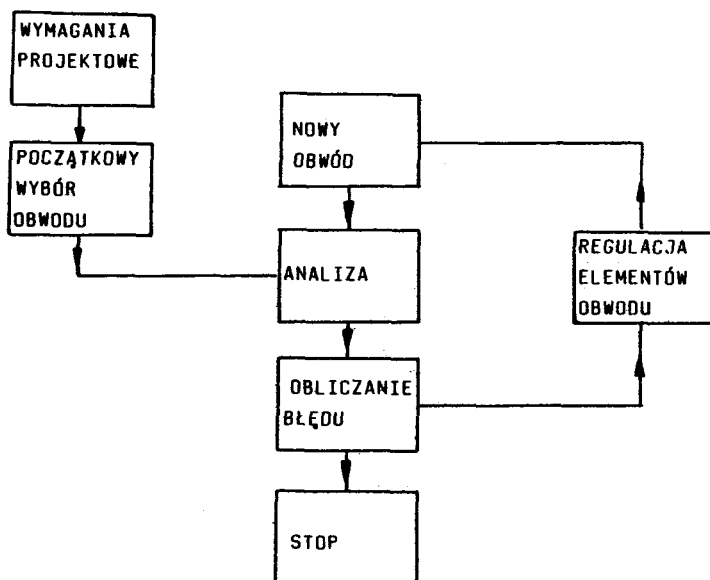
A u t o r



1.WSTĘP

Burzliwy rozwój technologii i co za tym idzie produkcji układów scalonych dużej i bardzo dużej skali integracji spowodował, że klasyczne metody projektowania układów elektronicznych stały się nieprzydatne.

Człowiek bez pomocy maszyny cyfrowej nie jest w stanie przeanalizować ogromnej ilości informacji dotyczących topologii i budowy układu scalonego i podjąć właściwej decyzji co do jego modyfikacji.



Rys. 1.1. Schemat blokowy projektowania układu elektronicznego przy pomocy maszyny cyfrowej

Typowy schemat blokowy procesu projektowania układu elektronicznego przedstawiono na rysunku 1.1.

Po początkowym wyborze obwodu, na ogół w procesie projektowania "ręcznego" lub też tylko częściowo wspomaganego maszyną cyfrową, zostaje on poddany wszechstronnej analizie za pomocą jednego z uniwersalnych programów analizy układów elektronicznych, np.: programu NAP2



opisanego w p. 4. Dalszy proces postępowania sprowadza się do obliczenia błędu pomiędzy otrzymanymi, a ustalonymi w wymaganiach projektowanych charakterystykami. Regulacja elementów obwodu lub wybór nowego obwodu może odbywać się w drodze ingerencji człowieka (projektanta) lub też może tego wyboru dokonywać maszyna cyfrowa wykorzystując pakiet programów do optymalizacji np. program MINUIT opisany w p. 6.

Wykonywanie analizy i regulacja elementów obwodu odbywa się tak długo, aż żądane wymagania projektowe zostaną spełnione z zadany**m** błędem.

Celem niniejszego skryptu jest przedstawienie pakietu programów wykorzystywanych w Insytucie Elektroniki Pł, umożliwiających przeprowadzenie analizy i opymalizacji parametrów dowolnego układu elektro-nicznego.



2. PRZEGLĄD PROGRAMÓW KOMPUTEROWEJ ANALIZY UKŁADÓW ELEKTRONICZNYCH

Rozwój programów komputerowej analizy układów elektronicznych rozpoczął się około 20 lat temu. Jednym z pierwszych programów był ECAP (Electronic Circuit Analysis Program) [69] opracowany przez IBM w 1965 roku i stosowany do dziś w wielu ośrodkach obliczeniowych na świecie.

W chwili obecnej do najczęściej używanych programów należą: SPICE [66], ASTAP i ADVICE. Programy te charakteryzują się użyciem najbardziej zaawansowanych metod numerycznych jak zmodyfikowana analiza węzłowa, technika macierzy rzadkich i niejawnych metod całkowania.

Do bardzo rozpowszechnionych szczególnie we Francji, programów należą również IMAG III oraz ASTEC 3 [72, 73], które dają duże możliwości deklarowania modeli i funkcji użytkownika.

Z najbardziej znanych programów analizy układów energetycznych i elektrycznych należy wymienić jeden z największych programów świata EMTP [70] (Electromagnetic Transient Program) liczący ponad 65000 kart Fortranu i ponad 300 procedur.

W niniejszym rozdziale będzie przedstawione porównanie możliwości programów PAUT2 i NAP2, używanych w Instytucie Elektroniki Pł., ze znanymi w świecie programami ANP3, ESACAP oraz SPICE [74]. Wszystkie te programy są napisane w języku FORTRAN.

PAUT2 jest przeznaczony do analizy układów tyrystorowych. NAP2 służy do analizy dowolnych układów nieliniowych dających się przedstawić w postaci zastępczego schematu elektrycznego. ANP3 jest przeznaczony do wyznaczania zer i biegunów układów liniowych w dziedzinie S lub Z . ESACAP jest przeznaczony do analizy dużych układów termicznych i dowolnych układów nieliniowych. SPICE może służyć do analizy układów scalonych dużej skali interakcji.

W tablicy 2.1 przedstawiono rozmiary wymienionych programów, programu MINUIT oraz nazwy komputerów, na których po raz pierwszy zostały one zastosowane.



T a b l i c a 2.1

Nazwa programu	Komputer, na którym go opracowano	Całkowita liczba kart Fortranu	Liczba kart Fortran po odjęciu kart komentarza	Liczba procedur
PAUT 2	Odra 1305	8068	≈ 7000	42
NAP 2	IMB-360	6762*	≈ 5800*	30
ANP 3	IMB-360	5000	3500	50
ESACAP	PDP-11/40	9000	?	100
SPICE 2	CDC-6400	16500	?	123
MINUIT	IMB-360	2546*	2546*	26

* - dotyczy wersji uruchomionej na m.c.Odra 1305

Możliwości wyżej wymienionych programów przedstawiono w tablicy 2.2 [74].

T a b l i c a 2.2

Możliwości programu	PAUT2	NAP2	ANP3	ESACAP	SPICE2
<u>Analiza DC</u>					
Wyznaczenie punktu pracy i charakterystyk przejściowych	+	+	-	+	+
Funkcje przenoszenia (OC)	-	+	-	-	+
Wrażliwość (DC)	-	+	-	-	+ -
Najgorszy przypadek (DC)	-	+	-	-	-
Optymalizacja punktu pracy i charakterystyk przejściowych (DC)	-	+	-	-	-
Analiza szumowa (DC, temperatura)	-	+	-	-	-
Analiza OC ze zmiennym krokiem	-	+	-	-	-



T a b l i c a 2.2 (cd.)

Możliwości programu	PAUT2	NAP2	ANP3	ESACAP	SPICE2
<u>Analiza AC</u>					
Wykresy Bode'go	-	+	+	+	+
Funkcje przenoszenia (AC)	+	+	+	+	+
Wrażliwość (AC)	-	-	+-	-	-
Częstotliwość zespolona (sigma + j* omega)	-	+	-	-	-
Zniekształcenia (AC)	-	-	-	-	+
Analiza szumowa (AC)	-	-	-	-	+
Bieguny i zera (AC)	-	-	+	+	-
Opóźnienie grupowe (AC)	-	-	+	+	-
<u>Analiza stanów przejściowych</u>					
Sygnały w funkcji czasu	+	+	+-	+	+
Parametry w funkcji czasu	+	+	-	+	-
Parametry w funkcji temperatury	-	+	-	+	+-
Parametry w funkcji sygnałów	+	+	-	+	+-
Analiza Fouriera	+	+	-	-	+
Wyznaczanie stanu usta- łego układów okresowych	+	-	-	+	-
Funkcje opóźnienia	-	-	-	+	-
Dowolne nieliniowości	+-	-	+	+	+-
Wbudowane złożone modele przysprządów półprzewodni- kowych	+-	+-	-	-	+
Wbudowany model idealne- go wzmacniacza operacyj- nego	-	+	+		-
Elementy termiczne	-	-	-	+	-



T a b l i c a 2.2 (cd.)

Możliwości programu	PAUT2	NAP2	ANP3	ESACAP	SPICE2
Automatyczne generowanie elementów	-	-	-	+	-
Linia długa, straty	-	+ -	-	+ -	+
Linia długa, dowolna symulacja	-	+ -	-	+ -	-
Automatyczne umieszczanie podobwodów	-	+	-	+	+
Biblioteka podobwodów	-	+	-	+ -	-
Definiowane przez użytkownika:					
- podprogramy	-	+	-	-	-
- funkcje (jak w FORTRANie)	-	-	-	+	-
- zmienne	-	+ -	-	+	+ -
- równania	-	+ -	-	+	+ -
Rozkaz *MODIFY	-	+	+	-	-
Rozkazy *SAVE i *LOAD	-	+	-	-	-
Rozkaz *HOLD do wykreślenia wielu wykresów tego samego przebiegu jednocześnie	-	+	-	-	-
Sterowanie przez użytkownika procesem obliczeń	+	+	-	-	+ -
Swobodnoformatowy język wejściowy	+	+	+ -	+ -	+ -

przy czym: + możliwe, - niemożliwe, + - możliwe z ograniczeniami.

W tablicy 2.3 przedstawiono liczbę parametrów opisujących modele wbudowane w programy PAUT2, NAP2 i SPICE2

Należy zaznaczyć, że w program NAP2 są wbudowane rozszerzone iniekcyjne modele Ebersa-Molla tranzystora biopolarnego i diody p-n, natomiast w programie SPICE 2 są wbudowane zmodyfikowane transportowe modele Gummel a-Poon a i stąd wynika tak znaczna różnica w ilości parametrów opisujących modele tych przyrządów.

W odróżnieniu od programu PAUT2, który posiada możliwość deklaracji a priori wielkości analizowanego obwodu, pozostałe programy mają z góry narzucone ograniczenia.



T a b l i c a 2.3

Liczba parametrów wbudowanego w pro- gram modelu	PAUT2	NAP2	SPICE2
Diody	6	7	14
tranzystora bipolar- nego	-	15	40
Tranzystora złączonego JFET	-	12	12
Tranzystora MOS (MOSFET)	-	-	50
Tyrystora	6	-	-
Przełącznika	2	-	-
Transformatora	6	-	-

W tablicy 2.4 przedstawiono maksymalne rozmiary obwodu, który może być analizowany przez poszczególne programy, z uwzględnieniem zastosowanego komputera i zajmowanej pamięci operacyjnej.

T a b l i c a 2.4

Program	Wersja	Wymagana pamięć	Maksymal- na liczba zmiennnych (węzłów)	Maksymal- na liczba współczyn- ników (ga- łęzi)
ANP3	IMB 3033	80 kbytes (over- lay)	40	100
NAP2	Odra 1305	59 kśłów (overlay)	50	200
NAP2	IBM 3033	250 kbytes (over- lay)	500	1500
ESACAP	DEC PDR- -11/40	?	60	350
ESACAP	VAX-11/780	?	> 350	> 10000
SPICE2	IBM 3033	400 kbytes (over- lay)	> 500	> 1500



Bardzo trudno jest zazwyczaj zbudować - dokładny i odzwierciedlający w pełni rzeczywistość - model, aż tak złożonego układu, na jaki pozwalają maksymalne ograniczenia tych programów. Na ogół postępuje się w ten sposób, że bardzo złożone układy dzieli się na mniejsze podobwoły, które analizuje się osobno. Następnie bazując na wykonanej analizie tych podobwołów buduje się dla nich makromodele, które uwzględniają tylko ich najbardziej istotne właściwości. Przy użyciu tak tworzonych makromodeli można następnie analizować bardzo złożone układy elektrotechniczne (VLSI), które normalnie przekraczałyby maksymalną pojemność programów z tablicy 2.4.

Na podstawie zamieszczonego porównania uniwersalnych programów analizy układów elektronicznych można stwierdzić, że trudno jest definitywnie powiedzieć, że któryś z nich jest zdecydowanie najlepszy. W zależności od typu analizowanego układu i żądanej analizy należy wybierać program, który będzie najbardziej odpowiedni w danym przypadku.



3. IDENTYFIKACJA PARAMETRÓW MODELI PRZYZRĄDÓW PÓŁPRZEWODNIKOWYCH

Użytkownik wykorzystując maszynę cyfrową do analizy i projektowania układów elektronicznych musi pamiętać, że wyniki analizy w dużym stopniu zależą od dokładności modeli przyrządów półprzewodnikowych zastosowanych w programie.

Prosty model przyrządu wymaga niewielu parametrów i pozwala na szybkie otrzymywanie wyników, ale może pomijać pewne zjawiska zasadnicze dla pracy obwodu lub być po prostu mało dokładny. Z drugiej strony wybór zbyt złożonego modelu zwiększa nakład pracy przy identyfikacji jego parametrów i wydłuża czas obliczeń. Dlatego wybór modelu przyrządu, odpowiedniego do wymagań stawianych obwodowi, jest jednym z podstawowych problemów, jakie napotyka użytkownik programów analizy komputerowej.

3.1. Identyfikacja parametrów modelu diody półprzewodnikowej

Równanie prądu dyfuzyjnego idealnego złącza półprzewodnikowego ma postać

$$I_D = I_0 \left(\exp \frac{qV}{kT} - 1 \right), \quad (3.1)$$

gdzie: k - stała Boltzmanna,

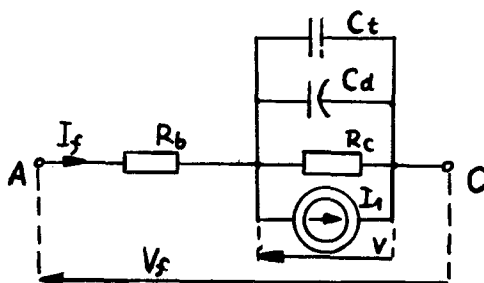
T - temperatura absolutna złącza,

$\frac{kT}{q} = V_T$ - tzw. potencjał termiczny złącza.

Wartość prądu I_0 zależy od parametrów technologicznych złącza. Modelem diody idealnej jest więc źródło prądowe opisane równaniem (3.1). W złączu rzeczywistym warunki idealnej dyfuzji są zakłócone przez generację i rekombinację nośników w obszarze zubożonym złącza, rezystancje kontaktów i struktury półprzewodnika oraz zjawiska związane z wystąpieniem dużych koncentracji nośników mniejszościowych porównywalnych z koncentracją domieszek przy dużych prądach złącza.



Te ostatnie zjawiska odgrywają decydującą rolę przy wysokim poziomie wstrzykiwania nośników. Sposób modelowania charakterystyk statycznych złącza p-n przy wysokich poziomach wstrzykiwania został przedstawiony w [43].



Rys. 3.1. Model Ebersa-Molla diody rzeczywistej

Ze względu na konieczność utrzymania stałej temperatury złącza pomiary będą wykonywane jedynie dla niezbyt wysokich prądów I_0 , dla których zjawiska wysokopoziomowe nie mają znaczenia. Model Ebersa-Molla przedstawiony na rysunku 3.1 jest dla takich prądów wystarczająco dokładny. Jest to model nieliniowy II-go rzędu uwzględniający magazynowanie ładunku w obszarze złącza oraz rezystancje kontaktów i materiału półprzewodnika. R_b reprezentuje rezystancje materiału półprzewodnika i kontaktów, R_c - uciążliwość złącza, C_t i C_d pojemność złączową i dyfuzyjną, a źródło prądowe I_1 prąd źródła idealnego złącza p-n opisanego równaniem

$$I_1 = I_S \left(\exp \frac{qV}{MkT} - 1 \right), \quad (3.2)$$

gdzie: I_S - prąd nasycenia złącza,

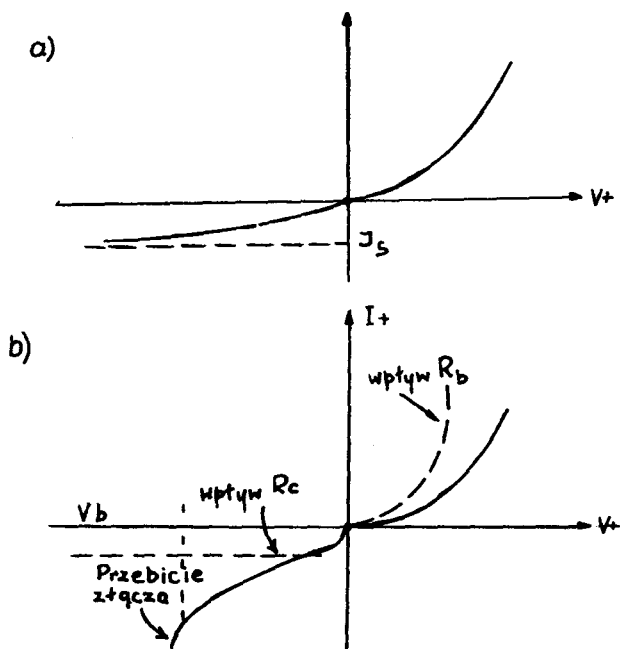
M - współczynnik emisji; dla złącza idealnego $M=1$,

a dla złącza rzeczywistego $M \in [1,2]$.

Na rysunku 3.2 przedstawiono wpływ parametrów R_b , M oraz R_c na kształt charakterystyki złącza. Poniżej napięcia V_b dla zwykłej diody następuje przebiecie i zniszczenie złącza.

W ogólnym przypadku wartości parametrów R_b , R_c , I_S , C_t i C_d nie są stałe, lecz zależą od punktu pracy i temperatury elementu.





Rys. 3.2. Charakterystyka statyczna: a) diody idealnej, b) diody rzeczywistej

3.1.1. Modelowanie charakterystyki statycznej złącza

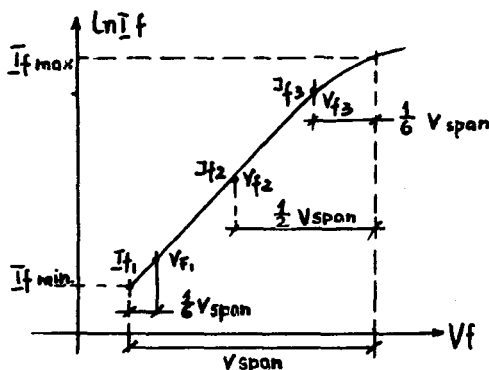
Model diody (rysunek 3.1) dla charakterystyki $I - V$ zdejmowanej w kierunku przewodzenia przedstawiają wzory

$$I_F = I_S \left(\exp \frac{qV}{MkT} - 1 \right) + \frac{V}{R_C} \quad (3.3)$$

$$V_F = V + I_F R_D, \quad (3.4)$$



składnik $\frac{V}{R_c}$ w (3.3) można pominąć, gdyż jego wartość jest znacząca jedynie przy polaryzacji w kierunku zaporowym. Stosując procedury dopasowania charakterystyk z zachowaniem minimum kwadratu błędu można obliczyć wartość współczynników M , I_S , R_b , (p. rozdz. 6). Wartości parametrów można obliczyć w sposób przybliżony na podstawie pomiarów charakterystyki w trzech punktach na krzywej w kierunku przewodzenia. Wybór punktów I_{F1} , I_{F2} , I_{F3} przedstawiono na rysunku 3.3.



Rys. 3.3. Wyznaczenie trzech punktów I_{F1} , I_{F2} , I_{F3} , na podstawie charakterystyki statycznej diody w kierunku przewodzenia

Dla tak dokładnych wartości prądu wykonuje się pomiar dokładnych wartości napięć metodą zerową. W wyniku pomiaru otrzymuje się wartości $(V_{F3} - V_{F1})$, $(V_{F2} - V_{F1})$, V_{F1} , które zostaną wykorzystane do obliczenia M , R_b , I_S . Dla punktów I_{F1} , V_{F1} , I_{F2} , V_{F2} , I_{F3} , V_{F3} równa (3.3) i (3.4) przedstawiają się następująco:

$$I_{F1} = I_S \left(e^{\frac{q}{MKT} (V_{F1} - I_{F1} R_b)} - 1 \right), \quad (3.5)$$

$$I_{F2} = I_S \left(e^{\frac{q}{MKT} (V_{F2} - I_{F2} R_b)} - 1 \right), \quad (3.6)$$



$$I_{F3} = I_S \left(e^{\frac{q}{MkT} (V_{F3} - I_{F3} R_b)} - 1 \right), \quad (3.7)$$

Wyrażenie $\frac{V}{R_c}$ zostało pominięte, ponieważ charakterystyka jest zdejmowana w kierunku przewodzenia.

We wzorach (3.5) i (3.7) można pominąć składnik I_S z niewielkim błędem; stąd

$$\frac{I_{F3}}{I_{F1}} = \frac{e^{\frac{q}{MkT} (V_{F3} - I_{F3} R_b)}}{e^{\frac{q}{MkT} (V_{F1} - I_{F1} R_b)}} = e^{\frac{q}{MkT} [V_{F3} - V_{F1} - R_b (I_{F3} - I_{F1})]},$$

$$\frac{MkT}{q} \ln \frac{I_{F3}}{I_{F1}} = V_{F3} - V_{F1} - R_b (I_{F3} - I_{F1}),$$

$$R_b = \frac{(V_{F3} - V_{F1}) - \frac{MkT}{q} \ln \frac{I_{F3}}{I_{F1}}}{I_{F3} - I_{F1}} \quad (3.8)$$

z równania (3.6)

$$I_S = I_2 / \exp \left\{ \frac{q}{MkT} (V_{F2} - I_{F2} R_b) \right\} \quad (3.9)$$

Po podstawieniu (3.8) i (3.9) do równania

$$\frac{I_{F2}}{I_{F1}} = e^{\frac{q}{MkT} [V_{F2} - V_{F2} - R_b (I_{F2} - I_{F1})]}$$



otrzymujemy

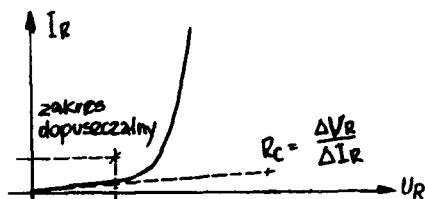
$$M = \frac{(V_{F3} - V_{F1}) \frac{I_{F2} - I_{F1}}{I_{F3} - I_{F1}} - (V_{F2} - V_{F1})}{\frac{kT}{q} \left[\frac{I_{F2} - I_{F1}}{I_{F3} - I_{F1}} \ln \frac{I_{F3}}{I_{F1}} - \ln \frac{I_{F2}}{I_{F1}} \right]} \quad (3.10)$$

Na podstawie pomierzonych współrzędnych punktów obliczamy kolejno M , R_D , I_S według równań (3.10), (3.8), i (3.9), przy czym

$$\frac{q}{k} = 1.161 \cdot 10^4 \frac{\text{deg}}{\text{V}}; \quad T = ^\circ\text{C} + 273,15.$$

3.1.2. Określenie parametrów rewersyjnych

Wartość złączowej omowej rezystancji upływu R_C można wyznaczyć z nachylenia charakterystyki V - I zdjętej w kierunku zaporowym, tak jak to przedstawiono na rysunku 3.4.



Rys. 3.4. Wyznaczenie rezystancji upływu R na podstawie charakterystyki diody w kierunku zaporowym

Dokładną wartość R_C można wyznaczyć z dopasowania charakterystyki V - I w kierunku zaporowym przy pomocy użycia metody optymalizacji, ale ponieważ R_C nie odgrywa dużej roli wystarcza jedynie jej wartość przybliżona. Poprawne działanie modeli zapewnia zazwyczaj każda wartość R_C większa od $1 \text{ M}\Omega$.



Wartość parametru R_b jest nieco inna dla kierunku przewodzenia i zaporowego, ponieważ interesują nas zazwyczaj parametry diody w kierunku przewodzenia, to przyjmujemy R_b jako stałą daną wzorem (3.8).

3.1.3. Określenie pojemności dyfuzyjnej i złączowej diody

Pojemność złączową obszaru zubożonego złącza zazwyczaj opisuje się wzorem

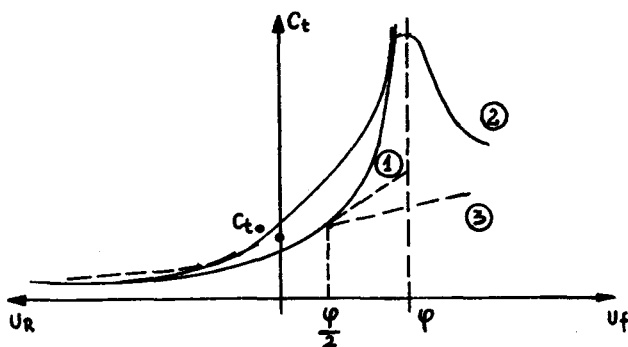
$$C_t = \frac{C_{t0}}{(\varphi - V)\delta}, \quad (3.11)$$

gdzie: C_{t0} - pojemność złączowa przy polaryzacji zerowej,

φ - stały potencjał dyfuzyjny złącza,

δ - stała zależna od technologii wykonania złącza, wartość stałej δ zależy od typu złącza i dla złącza skokowego wynosi 0,5.

Pojemność złączowa C_t modeluje przyrostowe zmiany ładunku zgromadzonego w warstwie ładunku przestrzennego złącza, związane z przyrostami napięcia złączowego. Zakładamy, że warstwa ładunku przestrzennego jest pozbawiona nośników. Można powiedzieć, że potencjał dyfuzyjny złącza jest napięciem na złączu, które zapewnia równowagę składowej dyfuzyjnej i dryftowej prądu. Zależność pojemności złączowej C_t od napięcia przedstawiono na rysunku 3.5.



Rys. 3.5. Zależność pojemności złączowej C_t od napięcia przyłożonego do diody



Z równania (3.11) wynika, że jeżeli $V \rightarrow \varphi$, to $C_t \rightarrow \infty$ (krzywa ① rys. 3.5). W rzeczywistości dla $V_F = \varphi$ pojemność C_t ma skończoną wartość, ponieważ założenie braku nośników w warstwie ładunku przestrzennego nie jest słuszne dla kierunku przewodzenia. Krzywa ② przedstawiona na rysunku 3.5 jest lepszym przybliżeniem, lecz wymaga wprowadzenia dodatkowego parametru b [75]

$$C_t = \frac{C_{to}}{(x^2 + b)} \delta / 2 \left(1 + \frac{\delta}{1 - \delta} \frac{b}{x^2 + b} \right), \quad (3.12)$$

gdzie $x = \frac{V - \varphi}{\varphi}$.

Zależność tę można również aproksymować prostą ③ (rysunek 3.5) dla napięć większych od $\varphi/2$

$$C_t(u) = 2^{\delta} C_{to} \left[2r \left(\frac{V}{\varphi} \right) + (1 - r) \right], \quad (3.13)$$

gdzie $V > \frac{\varphi}{2}$.

Ponieważ pojemność złączowa jest dominującym składnikiem przy polaryzacji zaporowej, przyjmujemy dla jej symulacji równanie (3.11). Pojemność dyfuzyjna jest związana z ładunkiem nośników ruchomych. Jest ona funkcją prądu złącza. Ładunek zgromadzony w pojemności dyfuzyjnej można przedstawić jako

$$Q_{\text{stored}} = I_d \tau, \quad (3.14)$$

gdzie τ - czas przelotu.

Czas przelotu można wyznaczyć na podstawie oscylogramu charakterystyki przełączenia diody lub też z pomierzonej charakterystyki pojemności dyfuzyjnej w funkcji C_d w funkcji prądu przewodzenia diody. Pojemności złączowa i dyfuzyjna mierzone w kierunku przewodzenia bocznią się wzajemnie i trudno jest zinterpretować ich wzajemny udział w mierzonej pojemności całkowitej.

Ze względu na to, że pojemność dyfuzyjna ma decydujące znaczenie przy polaryzacji w kierunku przewodzenia, a pojemność złączowa przy polaryzacji w kierunku zaporowym, wygodnie jest przyjąć, że wartość pojemności zmienia się od wartości pojemności złączowej przy polaryzacji zaporowej do pojemności dyfuzyjnej przy polaryzacji w kierunku prze-



wodzenia. Płynne przejście, od jednego rodzaju pojemności do drugiego, przy zmianie polaryzacji zapewnia odpowiednio dobrana i zmodyfikowana wartość C_0 i zależność pojemności dyfuzyjnej od prądu złącza [42, 43]. Powyższe założenie upraszcza w istotny sposób wyznaczanie parametrów opisujących pojemność, a nie ma zasadniczego wpływu na dokładność modelu.

Bezpośredni pomiar zależności pojemności dyfuzyjnej od prądu diody można wykonać np. za pomocą mostka do pomiaru parametrów admitancyjnych. Parametry C_0 , φ oraz σ można w sposób przybliżony (dokładnie należałoby stosować minimalizację odpowiednio sformułowanej funkcji celu) wyznaczyć następująco: z ch-ki $C=f(V)$ wybieramy dwa punkty (C_1, V_{R1}) , (C_2, V_{R2}) , oraz dokonujemy pomiaru pojemności C_{j0} przy braku polaryzacji. Dla tych dwóch punktów możemy zapisać równania

$$C_1 = \frac{C_{j0}}{\left(1 - \frac{V_1}{\varphi}\right)^\sigma}, \quad (3.15)$$

$$C_2 = \frac{C_{j0}}{\left(1 - \frac{V_2}{\varphi}\right)^\sigma}, \quad (3.16)$$

Przekształcając równanie (3.15) do postaci

$$\frac{C_{j0}}{C_1} = \left(1 - \frac{V_1}{\varphi}\right)^\sigma \quad (3.17)$$

i logarytmując je stronami możemy wyznaczyć stałą σ

$$\sigma = \frac{\ln \frac{C_{j0}}{C_1}}{\ln \left(1 - \frac{V_1}{\varphi}\right)}. \quad (3.18)$$

Logarytmując stronami równania (3.15) i (3.16) i dzieląc je przez siebie otrzymujemy



$$\frac{\ln \frac{C_{j0}}{C_1}}{\ln \frac{C_{j0}}{C_2}} = \frac{\ln \left(1 - \frac{V_1}{\varphi}\right)}{\ln \left(1 - \frac{V_2}{\varphi}\right)}, \quad (3.19)$$

tak więc

$$\ln \frac{C_{j0}}{C_1} \ln \left(1 - \frac{V_2}{\varphi}\right) = \ln \frac{C_{j0}}{C_2} \ln \left(1 - \frac{V_1}{\varphi}\right), \quad (3.20)$$

i

$$\left(1 - \frac{V_2}{\varphi}\right)^B = \left(1 - \frac{V_1}{\varphi}\right)^A, \quad (3.21)$$

gdzie: $A = \ln \frac{C_{j0}}{C_2}$,

$B = \ln \frac{C_{j0}}{C_1}$.

Możemy teraz napisać, że

$$f(\varphi) = \left(1 - \frac{V_2}{\varphi}\right)^B - \left(1 - \frac{V_1}{\varphi}\right)^A = 0 \quad (3.22)$$

i do rozwiązania tego nieliniowego równania z jedną niewiadomą zastosować szybkobieżny algorytm Newtona-Raphsona [13], który dla równania o postaci

$$f(x) = 0 \quad (3.23)$$

można przedstawić w następujący sposób:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad (3.24)$$

gdzie: x_n - wartość pierwiastka równania w n-tym kroku,

x_{n+1} - wartość pierwiastka równania w n+1-szym kroku.



Model diody wykorzystywany w programie NAP2 zostanie przedstawiony w p. 4.4.1.

Należy pamiętać, że jeżeli pomiary parametrów diody odbywały się w temperaturze różnej od 25°C , to należy przeliczyć parametry I_S , oraz V_T na tę temperaturę (równania podano w p.4.4.1).

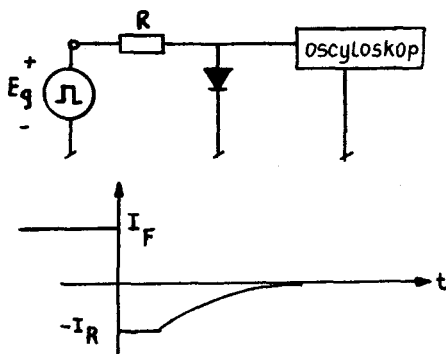
Wartość maksymalnej konduktacji dynamicznej diody G_S , występującej w modelu NAP2, można wyznaczyć na podstawie równania

$$G_S = \frac{I_{\text{dmax}} + I_S}{V_T}, \quad (3.25)$$

gdzie I_{dmax} - maksymalny prąd diody.

3.1.4. Doświadczalna weryfikacja dokładności wyznaczonych parametrów modelu diody

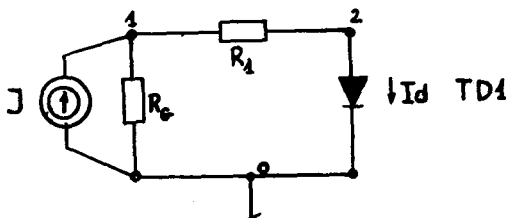
W celu sprawdzenia dokładności identyfikacji parametrów przyjętego modelu diody należy przeprowadzić komputerową analizę przełączania diody w układzie przedstawionym na rysunku 3.6 i porównać z przebiegiem doświadczalnym otrzymanym na oscyloskopie.



Rys. 3.6. Układ do doświadczalnej weryfikacji dokładności wyznaczonych parametrów diody



Rysunek 3.7 przedstawia schemat układu z rysunku 3.6, dostosowany do analizy za pomocą programu NAP2, a rysunek 3.8 - przykładowy program w języku NAP2 służący do symulacji procesu przełączania diody.



Rys. 3.7. Schemat układu do symulacji procesu przełączania diody przy pomocy programu NAP2

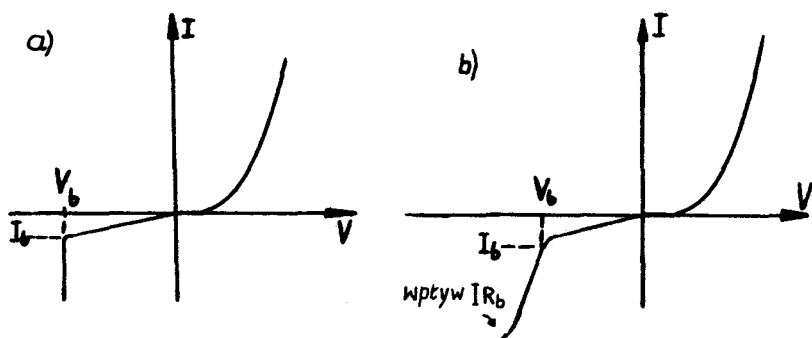
```
* CIRCUIT
: DANE DIODY
BA100 /DIODE/ IS 1E-14 VT 0.025 TT 8NS
CJ 6PF FI 0.7 GA 0.3
: DANE ZECI
TD1 2 0 BA100
R1 1 2 1K
PULSE /TAB 2/ 0 1 0.1US 1 0.1US -1 10US -1
RG 0 1 50 J= 0.5* PULSE (TIME)
* TIME 0 3.5US
* DCTR * PLOT(100) ITD1
* RUN
* END
```

Rys. 3.8. Przykładowy program do symulacji charakterystyki przełączania diody

3.2. Identyfikacja parametrów modelu diody Zenera

Model diody Zenera jest oparty na modelu Ebersa-Molla zwykłej diody, której charakterystyka w kierunku przewodzenia ma taki sam przebieg jak diody Zenera [32].





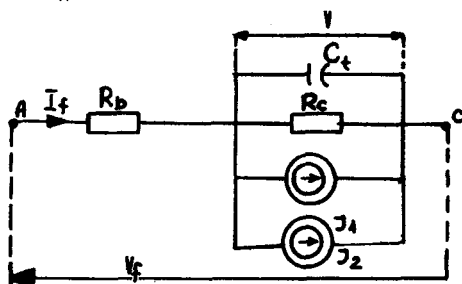
Rys. 3.9. Charakterystyka statyczna diody Zenera: a) idealnej, b) rzeczywistej

Na rysunku 3.9 przedstawiono charakterystyki statyczne idealnej (rysunek 3.9a) i rzeczywistej diody Zenera (rysunek 3.9b). Dla napięć $V > V_b$ dioda ta zachowuje się jak zwykła dioda. Gdy napięcie wsteczne osiąga V_b , dioda zaczyna stabilizować napięcie. W rzeczywistości element posiada skończoną rezystancję i charakterystyka odchyła się jak na rysunku 3.9b. Do modelu Ebersa-Molla (rysunek 3.1) dla zwykłej diody należy wprowadzić takie dodatkowe źródło prądowej, aby jego prąd dla polaryzacji dodatniej ≈ 0 , a przy zaporowej osiągnął wartość

$$I_2 = -e^A (V_b - B'V), \quad (3.26)$$

gdzie: $B' = B + 1$, $|B| < 0,1$;

$A = 10+100$; V_n - napięcie Zenera.



Rys. 3.10. Model diody Zenera

Kompletny model diody Zenera przedstawiono na rysunku 3.10. Uwzględnia on fakt, że przy normalnym dla pracy diody Zenera kierunku polaryzacji zaporowym pojemność dyfuzyjna $C_d \approx 0$. Pojemność C_t jest sumą pojemności złączeniowej i pojemności obudowy diody. Charakterystyki statyczne diody opisują następujące równania:

$$I_f = I_1 + I_2 + \frac{V}{R_c}, \quad (3.27)$$

$$V_f = V + I_f R_b. \quad (3.28)$$

Gdy $V > 0$, $I_2 \approx 0$ i $I_f \approx I_1$ (składnik $\frac{V}{R_c}$ jest do pominięcia w kierunku przewodzenia).

Gdy $V \ll 0$, $I_1 \approx 0$. Gdy $V < 0$ i $|V| < |V_b|/2$, $I_2 \approx 0$ i prąd diody $\approx$ prądowi upływu diody $\frac{V}{R_c}$.

Gdy $V \rightarrow V_b$, I_2 staje się składnikiem dominującym. Wartość parametru B określa, kiedy prąd zaczyna się odchyłać od wartości $\frac{V}{R_c}$ (gdy $|B| \searrow$, to "kolano" charakterystyki się wyostreza. Wartość A określa jak szybko prąd wzrasta, gdy osiągnięto już napięcie V_b ($A \nearrow$, to poprawia się stabilizacja napięcia).

W praktyce charakterystyka diody w kierunku zaporowym ma nieco inny charakter niż na to wskazuje równanie (3.26) i jest przedstawiona na rysunku 3.11.

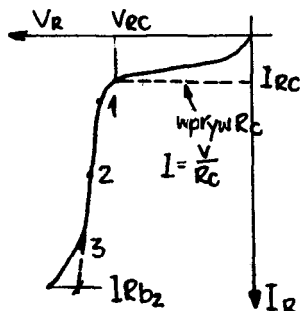
Z jej przebiegu wynika, że źródło prądowe I_2 powinno być przedstawione w postaci sumy dwóch składników

$$I_2 = -e^{A(V_b - B'V)} - e^{CV_b - DV}, \quad (3.29)$$

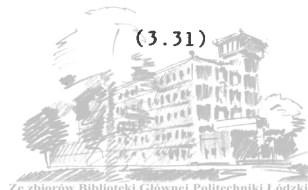
Równania opisujące diodę będą wówczas postaci

$$I_f = I_S \left(e^{\frac{qV}{MkT}} - 1 \right) - e^{A(V_b - B'V)} - e^{CV_b - DV} + \frac{V}{R_c}, \quad (3.30)$$

$$V_f = V + I_f R_b \quad (3.31)$$



Rys. 3.11. Dokładna charakterystyka statyczna diody Zenera w kierunku zaporowym



Należy pamiętać, że parametr R_b ma inną wartość w kierunku przewodzenia, a inną - w kierunku zaporowym, wobec czego lepiej używać dwóch wartości R_{B1} i R_{B2} . Wartość parametrów I_S , M , R_{B1} można otrzymać z charakterystyk w kierunku przewodzenia (analogicznie jak w p. 3.1). Parametr R_C otrzymujemy z wykresu $I_R = f(V_R)$ dla części charakterystyki spełniającej warunek $|V| < 0,5 |V_b|$.

Wartość parametrów A , B , R_{B2} lub A , B , C , D , R_{B2} najlepiej otrzymać z dopasowania charakterystyki $I_R = f(V_R)$ według wzoru (3.27), I_2 wyraża się wzorem (3.26) lub (3.29) z zachowaniem minimum kwadratu błędu. Do przybliżonego wyznaczenia parametrów A , B , R_{B2} wyznaczamy współrzędne trzech punktów na charakterystyce przedstawionej na rysunku 3.11.

$$I_R = f(V_R), \quad (I_{R1}, V_{R1}), \quad (I_{R2}, V_{R2}), \quad (I_{R3}, V_{R3}).$$

Otrzymujemy równania:

$$I_1 = -e^{AV_b - A'(V_{R1} - I_{R1} R_{B2})}, \quad (3.32)$$

$$I_2 = -e^{AV_b - A'(V_{R2} - I_{R2} R_{B2})}, \quad (3.33)$$

$$I_3 = -e^{AV_b - A'(V_{R3} - I_{R3} R_{B2})}, \quad (3.34)$$

przy czym:

$$I_1 = I_{R1} - I_{RC}, \quad (3.35)$$

$$I_2 = I_{R2} - I_{RC}, \quad (3.36)$$

$$I_3 = I_{R3} - I_{RC}, \quad (3.37)$$

z których otrzymujemy szukane parametry

$$R_{B2} = \frac{-V_{R2} + V_{R1} + \frac{1}{A} \ln \frac{I_1}{I_2}}{-I_{R2} + I_{R1}} \quad (3.38)$$

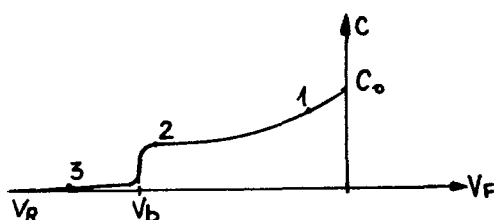


$$A' = \frac{\ln \frac{I_1}{I_2} (I_{R3} - I_{R1}) - \ln \frac{I_1}{I_3} (I_{R3} - I_{R1})}{(V_{R3} - V_{R1}) (I_{R2} - I_{R1}) - (V_{R2} - V_{R1}) (I_{R3} - I_{R1})}, \quad (3.39)$$

$$B' = \frac{A'}{A}, \quad (3.40)$$

przy czym

$$A = \frac{A' (V_{R3} - I_{R3} R_{B2}) + \ln (-I_3)}{V_b}. \quad (3.41)$$



Rys. 3.12. Charakterystyka pojemności złączowej diody Zenera

Przebieg pojemności złączowej diody Zenera przedstawiono na rysunku 3.12. Dla napięć bardziej dodatnich niż V_b przebieg tej charakterystyki jest podobny do $C_t = f(V_R)$ dla zwykłej diody i można opisać go wzorem

$$C_t = \frac{C_0}{(1 - \frac{V}{V_b})^{\delta}}, \quad (3.42)$$

Gdy $|V| > |V_b|$, to pojemność diody maleje do pojemności obudowy C_h . Należy pamiętać, że $C = C_t + C_h$.

Z przebiegu tej charakterystyki można znacznie dokładniej określić wartość napięcia Zenera V_b niż charakterystyki statycznej i wartości napięcia V_b określamy właśnie na podstawie tej charakterystyki.

Aby oddać skokowy charakter zmian pojemności złączowej aproksymuje się charakterystykę $C = f(V_R)$ wzorem

$$C_t = \frac{C_0}{(1 - \frac{V}{V_b})^{\delta}} \cdot \frac{1}{e^{K(V_b - V)} + 1}, \quad (3.43)$$



gdzie: oprócz znanych parametrów C_0, φ oraz γ występuje tu dodatkowy parametr K ; dla $V > V_b - e^{K(V_b - V)} \approx 0$ i C_t wyraża się podobnym wzorem jak dla normalnej diody; gdy $|V| > |V_b|$ i $V < 0$, $e^{K(V_b - V)} \rightarrow \infty$ i $C_t \rightarrow 0$.

Wartości C_0, φ, r', K odpowiadające charakterystyce $C = f(V_R)$ najwygodniej znaleźć dopasowując charakterystykę z zachowaniem minimum kwadratu błędu. Przybliżone obliczenia na podstawie trzech punktów na charakterystyce $C = f(V_R)$ (rys. 3.12.) wygląda następująco:

$$C_1 = \frac{C_0}{\left(1 - \frac{V_1}{\varphi}\right)^\gamma}, \quad (3.44)$$

(dla punktu pierwszego pomijamy wpływ parametru K)

$$C_2 = \frac{C_0}{\left(1 - \frac{V_2}{\varphi}\right)^\gamma} \cdot \frac{1}{2}, \quad \text{bo } (V_b - V_2) \approx 0, \quad (3.45)$$

$$C_3 = \frac{C_0}{\left(1 - \frac{V_3}{\varphi}\right)^\gamma} \cdot \frac{e^{K(V_b - V_3)}}{e^{K(V_b - V_3)} + 1} \quad (3.46)$$

Parametr C_0 odczytujemy bezpośrednio z pomiarów (pojemność przy polaryzacji zerowej złącza).

Analogicznie do punktu 3.1.3 otrzymujemy nieliniowe równanie w postaci

$$f(\varphi) = \left(1 - \frac{V_1}{\varphi}\right)^A - \left(1 - \frac{V_2}{\varphi}\right)^B, \quad (3.47)$$

gdzie $A = \ln \frac{C_0}{2C_2}$ $B = \ln \frac{C_0}{2C_1}$, które rozwiązujemy szybkozbieżnym algorytmem Newtona- Raphsona

$$\varphi_{n+1} = \varphi_n - \frac{1}{f'(\varphi_n)} F(\varphi_n), \quad (3.48)$$



$$f(\varphi) = \left(1 - \frac{V_1}{\varphi}\right)^A - \left(1 - \frac{V_2}{\varphi}\right)^B, \quad (3.49)$$

$$\frac{df(\varphi)}{d\varphi} = \frac{1}{\varphi^2} \left[AV_1 \left(1 - \frac{V_1}{\varphi}\right)^{A-1} - BV_2 \left(1 - \frac{V_2}{\varphi}\right)^{B-1} \right]; \quad (3.50)$$

następnie wyznaczamy

$$\gamma = \frac{\ln \frac{C_0}{C_1}}{\ln \left(1 - \frac{V_1}{\varphi}\right)} \quad (3.51)$$

oraz

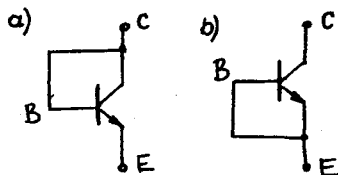
$$K = \left[\ln \frac{C_0}{C_3} - \gamma \ln \left(1 - \frac{V_3}{\varphi}\right) \right] \frac{1}{V_b - V_3}. \quad (3.52)$$

3.3. Identyfikacja parametrów w modelu tranzystora bipolarnego

Nieliniowy iniekcyjny model Ebersa-Molla tranzystora bipolarnego wbudowany w program NAP2 jest opisany w p. 4.4.2. W celu identyfikacji jego parametrów dla parametrów statycznych złącza można posłużyć się metodami opisanymi w p. 3.1.

Warunkiem poprawnej identyfikacji jest zapewnienie zerowej polaryzacji nie mierzonego złącza poprzez jego zwarcie tak jak to przedstawiono na rysunku 3.13.

W wyniku pomiarów charakterystyk statycznych złącza tranzystora można wyznaczyć parametry V_{TE} , I_{SE} oraz G_S



Rys. 3.13. Połączenie elektrod tranzystora przy identyfikacji parametrów statycznych złącza a) złącza baza-emiter, b) złącza baza-kolektor



dla złącza emiterowego, a także V_{TC} oraz I_{SC} dla złącza kolektorowego, a następnie obliczyć współczynniki n_i oraz n_v

$$n_i = \frac{I_{SC}}{I_{SE}} = NI, \quad (3.53)$$

$$n_v = \frac{V_{TC}}{V_{TE}} = NV, \quad (3.54)$$

gdzie: $V_{TE} = V_T$ - potencjał termiczny złącza B-E,

$I_{SE} = I_S$ - prąd nasycenia złącza B-E,

G_S - maksymalna przewodność złącza przyjmowana jako jednokowa dla obu złącz,

n_i, n_v - parametry charakteryzujące złącza B-C.

Identyfikacja parametrów opisujących pojemności złączowe tranzystora C_{Ej} i C_{Cj} nie różni się od pomiarów pojemności złączowej diody (z zachowaniem połączenia elektrod tranzystora, przedstawionych na rysunku 3.13).

W wyniku pomiarów otrzymujemy cztery parametry C_{EO}, C_{CO}, φ oraz δ opisujące przebieg pojemności złączowych tranzystora. Ze względu na konieczność przyjęcia w modelu jednakowej wartości potencjału złącza φ oraz wykładnika δ prawidłowa identyfikacja powinna polegać na minimalizacji sumy kwadratów odchyłeń charakterystyk teoretycznych obu złącz od wartości pomiarowych, tak jak to przedstawiono w równaniu

$$f_v = \sum_{i=1}^n \sum_{j=1}^2 \left[\left(\frac{C_{EO}}{\left(1 - \frac{V_{BE_i}}{\varphi}\right)^{\delta}} - C_{Ei} \right)^2 + \left(\frac{C_{CO}}{\left(1 - \frac{V_{BC_i}}{\varphi}\right)^{\delta}} - C_{Cj} \right)^2 \right], \quad (3.55)$$

gdzie: f_v - funkcja celu, wyznaczenie jej minimum pozwala na znalezienie właściwej wartości parametrów C_{EO}, C_{CO}, φ oraz δ

C_{Ei} - wartość pojemności złączowej złącza baza-emiter w i-tym punkcie pomiarowym,

V_{BE_i} - napięcie baza-emiter w i-tym punkcie pomiarowym,

C_{Cj} - wartość pojemności złączowej złącza baza-kolektor w j-tym punkcie pomiarowym,



V_{BCj} - napięcie baza- kolektor w j-tym punkcie pomiarowym,
 n - liczba pomiarów dla każdego ze złącz.

Model tranzystora wbudowany w program NAP2 zakłada stałe wartości współczynników wzmocnienia prądowego $\mathcal{L}_F$ i $\mathcal{L}_R$. Wartości te można znaleźć po wyznaczeniu za pomocą charakterografu współczynników wzmocnienia prądowego w układzie wspólnego emitera $\beta_F = f(I_C)$ oraz $\beta_R = f(I_C)$, a następnie obliczenia $\mathcal{L}_F$ z zależności

$$\mathcal{L}_F = \frac{\beta_F}{\beta_F + 1}, \quad (3.56)$$

$$\mathcal{L}_R = \frac{\beta_R}{\beta_R + 1}. \quad (3.57)$$

Następnie należy przyjąć wartości $\mathcal{L}_F$ i $\mathcal{L}_R$ dla określonego prądu lub też wartość uśrednioną w pewnym przedziale prądów. W celu dokładnej symulacji współczynników wzmocnienia prądowego $\mathcal{L}_F$ i $\mathcal{L}_R$ należy je zadeklarować jako funkcje prądu kolektora [3, 5, 43]

$$\mathcal{L} = \frac{1}{k_1 + k_2 I_C + k_3 I_C^4}, \quad (3.58)$$

gdzie k_1, k_2, k_3, k_4 , - parametry równania (3.58) zależne od typu tranzystora.

W celu dokładnej symulacji charakterystyk wyjściowych tranzystora bipolarnego w modelu opisanym w p. 4.4.2 wprowadzono symulację wypadkowej przewodności wyjściowej tranzystora G_O przy pomocy zależności

$$G_O = G_Z + n_G (G_{BE} + G_{BC}), \quad (3.59)$$

gdzie : G_Z -- konduktacja wyjściowa bez polaryzacji zewnętrznej,
 n_G - współczynnik proporcjonalności dla konduktacji wyjściowej,

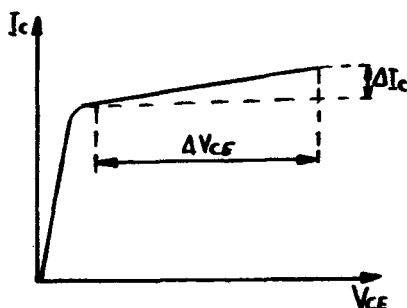
G_{BE}, G_{BC} - konduktacje złącz, baza-emiter i baza-kolektor w danym punkcie pracy.

Przewodność G_Z można pomierzyć przy pomocy mostka RLC przy zerowej polaryzacji tranzystora i rozwartym obwodzie bazy. Dla tranzystorów krzemowych najczęściej przyjmuje się wartość $G_Z = 0$.



Współczynnik proporcjonalności n_G można wyznaczyć na podstawie pomierzonej charakterystyki wyjściowej tranzystora w układzie wspólnego emitera (rys. 3.14.) dla wartości prądu emitera mniejszej od tej, dla której zaczyna się aproksymacja charakterystyki statycznej złącza emiterowego linią prostą ($I_E < I_{E_{max}}$). Po wyznaczeniu wartości przewodności wyjściowej G_{wy} z zależności (3.60) można wyznaczyć wartość współczynnika n_G

$$n_G = \frac{(g_{wy} - G_Z) V_I}{I_C} \quad (3.60)$$



Rys. 3.14. Pomiar przewodności wyjściowej tranzystora

Przy tym sposobie modelowania przewodności wyjściowej rośnie ona wraz ze wzrostem prądu, dopóki nie przekroczy on wartości $I_{E_{max}}$. Powyżej tej wartości prądu pozostaje ona stała i równa

$$G_o = n_G G_S, \quad (3.61)$$

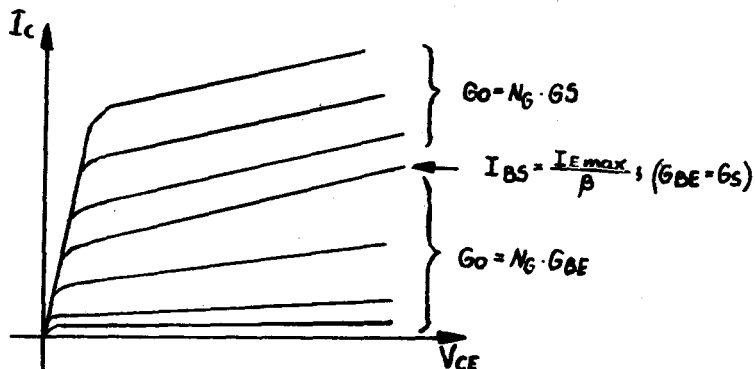
gdzie G_S - maksymalna konduktacja złącza baza-emiter.

Na rysunku 3.15 przedstawiono charakterystyki wyjściowe tranzystora otrzymane w wyniku zastosowania wbudowanego w program NAP2 modelu.

W celu wyznaczenia czasów przelotu τ_F i τ_R przez obszar bazy można posłużyć się pomiarami impulsowymi czasu narastania odpowiedzi prądowej tranzystora w układzie wspólnej bazy dla kierunku normalnego i inwersyjnego.



Dla współczesnych tranzystorów epiplanarnych wartość tego czasu jest rzędu ułamka nanosekundy i bezpośredni pomiar wymaga aparatury o szerokości pasma rzędu 100 GHz. Dlatego też częściej dokonuje się impulsowego pomiaru czasu życia nośników przez pomiar czasu odpowiedzi tranzystora w układzie wspólnego emitera, a następnie oblicza się czas życia. Wymagania dotyczące aparatury są wówczas znacznie mniejsze.



Rys. 3.15. Charakterystyki wyjściowe tranzystora otrzymane w wyniku wykorzystania wbudowanego w program NAP2 modelu

Najprostszą metodą wyznaczania czasów przelotu jest jednak wyznaczenie charakterystyki $f_T = f(I_C)$ dla połączenia normalnego i inwersyjnego tranzystora w układzie z rys. (3.16) korzystając z zależności

$$|\beta| = \left| \frac{I_C}{I_B} \right|, \quad (3.62)$$

gdzie:

$$I_C = \frac{V_C - U_3}{R_1}, \quad (3.63)$$

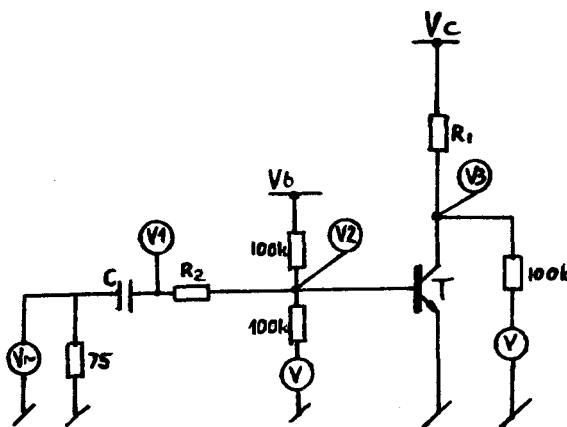
$$I_B = \frac{U_2 - U_1}{R_2}; \quad (3.64)$$

a następnie obliczając

$$f_t = |\beta(f)| f, \quad (3.65)$$

gdzie f - częstotliwość pomiarową (na ogół przyjmujemy $f \leq 1$ GHz).





Rys. 3.16. Układ do pomiaru charakterystyki
 $f_T = f(I_C)$

Następnie należy wyznaczyć pojemności złącz: emiterowego i kolektorowego spolaryzowanych w kierunku przewodzenia C_{Ej} (V_{BE}) i C_{Cj} (V_{BC}) i dla danej wartości prądu można wyznaczyć τ_F i τ_R w zależności

$$\tau_F = \frac{1}{2\pi f_T} - \frac{V_T}{I_C} C_{je}(V_{BE}), \quad (3.66)$$

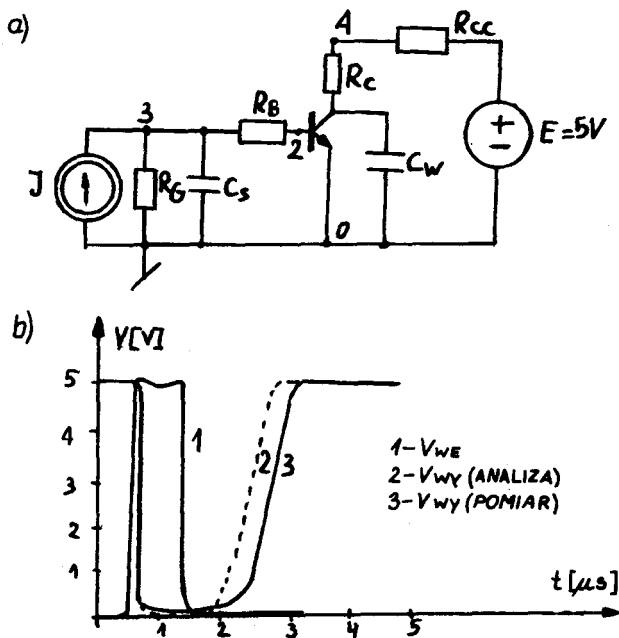
$$\tau_R = \frac{1}{2\pi f_T'} - \frac{V_T}{I_C} C_{jc}(V_{BC}), \quad (3.67)$$

gdzie f_T' , I_C' - wartości wyznaczone dla połączenia inwersyjnego tranzystora.

W celu sprawdzenia dokładności identyfikacji parametrów modelu tranzystora należy porównać wyznaczony komputerowo przebieg odpowiedzi tranzystora pracującego w układzie inwertera (rys.3.17a) z przebiegiem otrzymanym doświadczalnie (rys. 3.17b).

Należy pamiętać o przeliczeniu parametru I_S oraz V_T na temperaturę 25°C.





Rys. 3.17. Układ inwertera: a) schemat, b) przykładowe przebiegi

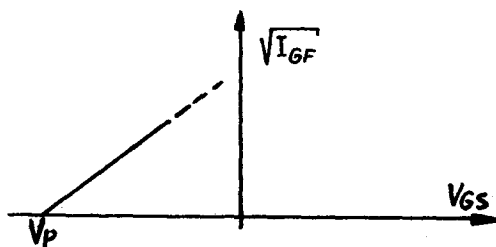
3.4. Identyfikacja parametrów tranzystora złączowego FET

Model tranzystora FET wbudowany w program NAP2 (p. 4.4.3) nie uwzględnia wpływu temperatury na wartość parametru modelu.

Ze względu na identyczny opis pojemności złączowych jak dla tranzystora bipolarnego, metoda identyfikacji parametrów C_S , C_D , φ oraz δ jest taka sama. Podobnie wyznaczenie parametrów G_Z i n_G modelujących wyjściową przewodność tranzystora jest identyczne jak dla tranzystora bipolarnego.

W celu wyznaczenia parametru transkonduktacji β , napięcia progowego V_p i napięcia górnego ograniczenia V_U należy pomierzyć jego charakterystykę przejściową (rys. 4.29), a następnie wykreślić ją w postaci $\sqrt{I_{GF}} = f(V_{GS})$ tak, jak to pokazano na rysunku 3.18.





Rys. 3.18. Metoda wyznaczania wartości napięcia progowego V_p

W miejscu przecięcia tej charakterystyki z osią V_{GS} otrzyma się wartość V_p , ponieważ dla zakresu kwadratowego

$$I_{GF} = \beta (V_{GS} - V_p)^2; \quad (3.62)$$

$$\sqrt{I_{GF}} = \sqrt{\beta} (V_{GS} - V_p). \quad (3.63)$$

Na podstawie rysunku 3.18 można również wyznaczyć wartość parametru transkonduktacji β .



4. OPIS JĘZYKA WEJŚCIOWEGO PROGRAMU NAP2

4.1. Charakterystyka programu NAP2

Program NAP2 VERSION PL84 jest zmodyfikowaną wersją programu NAP2 opracowanego w Danii przez Thomasa Rubnera-Petersona [61, 62], na maszynie cyfrową IBM 370/165. Program ten został napisany w języku Fortran IV i od chwili powstania jest ciągle w trakcie modyfikacji i prze-róbek zarówno w Danii, jak też w innych ośrodkach, w których jest używany. Aktualna wersja programu używanego w Instytucie Elektroniki Politechniki Łódzkiej powstała w wyniku kilkuletnich doświadczeń z wykorzystaniem tego programu do prowadzenia zajęć dydaktycznych z "Analizy i Projektowania Komputerowego Układów Elektronicznych". Oprócz wersji podstawowej PL84 zajmującej w pojedynczej precyzji 51K słów pamięci operacyjnej maszyny cyfrowej Odra 1305, istnieje też wersja MINI tego programu zajmująca tylko 31K słów, lecz posiadająca ograniczenia polegające na możliwości korzystania tylko z jednej biblioteki oraz braku możliwości wykonania optymalizacji i analizy AC. Wersja ta jest natomiast bardzo użyteczna w przypadku obliczeń pod kontrolą systemu George 3, gdy wymagania pojemności pamięci zajmowanej przez pojedynczego użytkownika są często bardzo krytyczne.

Program NAP2 pozwala na wprowadzenie danych w formacie swobodnym, to znaczy nie jest istotne miejsce danej w konkretnej kolumnie (są dwa wyjątki), lecz każda instrukcja jest kolejno rozpoznawana zgodnie z semantyką języka. Zdania (statements) swobodnoformatowego języka NAP2 można podzielić na zdania opisu obwodu oraz na zdania sterujące. Jako znak separacji jest wykorzystywana spacja lub przecinek. Każde zdanie jest wprowadzane na oddzielnym rekordzie (karcie perforowanej), przy czym można wykorzystać kolumny od 1 do 72 włącznie. W przypadku, gdy długość rekordu jest zbyt krótka, użycie znaku przeniesienia > pozwala na zapisanie dalszej części zdania na następnej karcie.

Znacznikiem końca zdania jest średnik ";", koniec rekordu lub też znak komentarza ":". Każda informacja, która wystąpi po znaku ":" jest traktowana jako komentarz i zostaje pominięta przez procedurę analizującą dane wyjściowe. Jeżeli dwukropek wystąpi na początku, to wówczas cały rekord jest traktowany jako komentarz.



W programie NAP2 VERSION PL84 występują następujące ograniczenia:

- 1) liczba węzłów łącznie z liczbą pierwotnych zmiennych prądowych ≤ 50 ;
- 2) liczba elementów obwodu + liczba diod + $6 \times$ liczba tranzystorów bipolarnych + $6 \times$ liczba tranzystorów polowych + liczba opcji wyjściowych ≤ 195 ;
- 3) liczba podparametrów na listach funkcji i typów elementów półprzewodnikowych + $2 \times$ liczba wartości funkcjonalnych + $2 \times$ liczba diod + $4 \times$ liczba tranzystorów bipolarnych + $8 \times$ liczba tranzystorów polowych + $2 \times$ liczba sprzężeń nieliniowych ≤ 276 .

Podstawowe elementy półprzewodnikowe, takie jak diody, tranzystory polowe są wbudowane w program w postaci gotowych modeli. Modele i makromodele dla danych typów przyrządów półprzewodnikowych mogą być tworzone przez użytkownika zarówno ze standardowych wyrażeń języka NAP2 [52, 54], jak również poprzez dołączenie na etapie kompilacji własnych procedur napisanych przez użytkownika w języku FORTAN 1900 [23, 46, 47, 48, 44, 51, 56].

4.1.1. Organizacja programu NAP2

Program NAP2 VERSION PL84 jest skompilowany na dysku 000005, wraz z programem PAUT2. W celu zmniejszenia wielkości pamięci operacyjnej zajmowanej przez program jest on zorganizowany jako program nakładany. Jednostka stała, zawierająca segment MASTER i segmenty standardowe, znajduje się stale w trakcie wykonywania programu w określonym obszarze pamięci operacyjnej. Natomiast jednostki nakładane są przyporządkowane określonemu obszarowi pamięci. W przypadku wywołania segmentu zawartego w dowolnej jednostce nakładanej system sprawdza, czy jednostka ta znajduje się w pamięci operacyjnej. Jeżeli nie, zostaje przeniesiona ze zbioru zewnętrznego i umieszczona w przyporządkowanym jej obszarze pamięci.

W podobny sposób, w przypadku wykonania instrukcji RETURN, system sprawdza w miarę potrzeby jednostkę nakładaną zawierającą segment, do którego następuje powrót [56].

Program nakładany jest przechowywany w zbiorze dyskowym o nazwie PROGRAM NAPS. Oprócz tego zbioru do wykonania i użytkowania programu NAP2 są niezbędne jeszcze zbiory PERT, WWY, MAA, LIBRARY i CATALOG.



T a b l i c a 4.1

Programowy nr urządzenia	Typ urządzenia	Komentarz
5	CR	Czytnik kart wykorzystywany do wprowadzania danych wejściowych
6	LP	Drukarka wierszowa służąca do wyprowadzenia wyników
2	ED1	Zbiór dyskowy PERT wykorzystywany przy obliczeniach metodą perturbacji
3	ED2	Zbiór dyskowy WWY dla wyników wyprowadzonych w trybie "off-line"
11	ED3/DIRECT	Zbiór LIBRARY o długości rekordów 128 słów, służący do zakładania bibliotek LIB1 do LIB6
12	ED4/DIRECT	Zbiór CATALOG o długości rekordów 512 słów, służący do sterowania pracą bibliotek
20	ED5	Zbiór MAA służący do zapamiętywania informacji o analizowanym obwodzie (instrukcje SAVE, LOAD)
OFFL	LP	Opcjonalna dodatkowo drukarka do wyprowadzenia wyników w trybie "off-line" numer tego urządzenia jest deklarowany w zdaniu * RUN

Na rysunku 4.1. przedstawiono organizację dysku 000005 oraz wielkość pamięci zajmowanej przez poszczególne zbiory (1 Blok = 128 słów). Każdy pakiet fabrycznie nowy musi zostać zainicjowany. Podczas inicjowania pakiet otrzymuje numer seryjny, który służy następnie do jego identyfikacji; na pakiecie zostaje zarezerwowany obszar zajmujący cztery pierwsze cylindry (0-3). W obszarze tym znajduje się między innymi pole sterujące systemem (PSS), gdzie będą umieszczone informacje o zakładanych zbiorach.

PROGRAM NAPS	(3200 BKTS)
PERT	(800 BKTS)
WWY	(5120 BKTS)
MAA	(640 BKTS)
LIBRARY	(960 BKTS)
CATALOG	(60 BKTS)
PROGRAM PAUT	(1680 BKTS)

Rys. 4.1. Organizacja dysku 000005



Do zakładania zbiorów stosuje się program #XPJC. Należy zaznaczyć, że zbiory LIBRARY oraz CATALOG są zbiorami algorytmizowanymi (losowymi) o stałej długości rekordów. Na rysunku 4.2 przedstawiono przykład danych służących do zakładania zbiorów na pakiecie 000005 przy pomocy programu #XPJC.

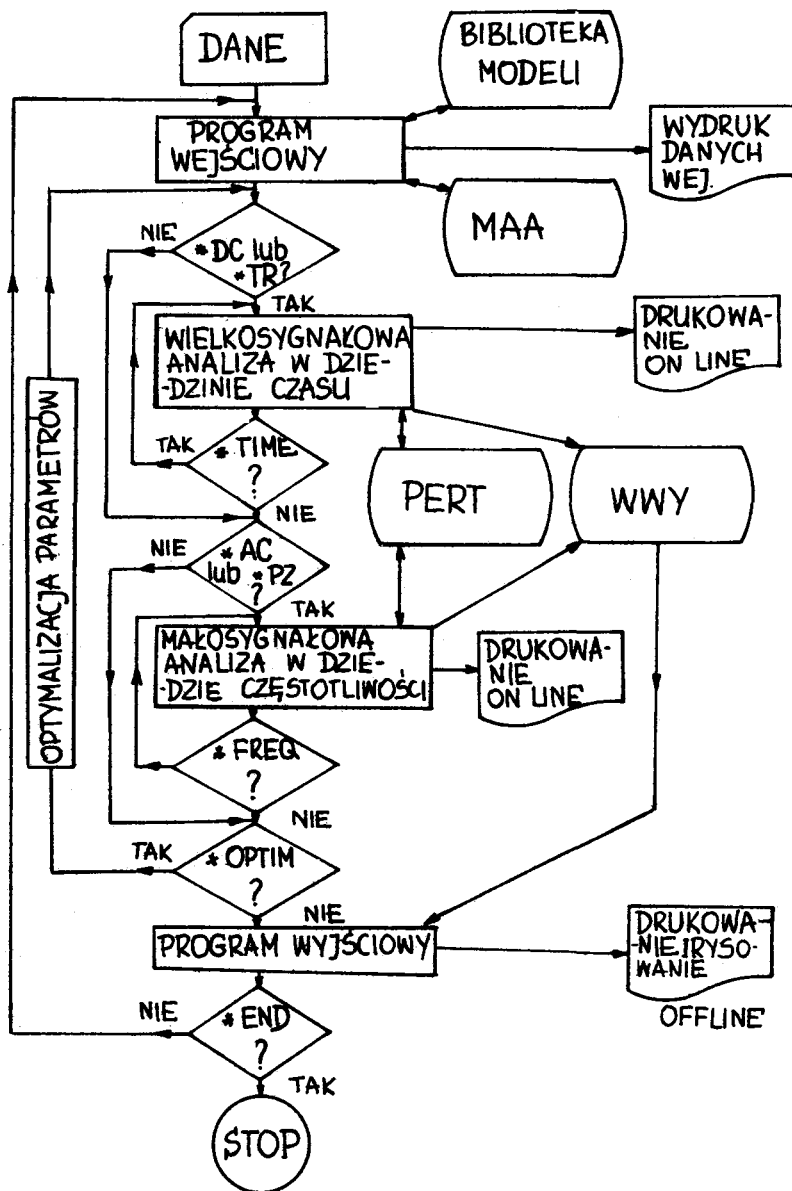
```
XPJC1,1,000005,PROGRAM NAPS,0,4095,3,1,
XPJC4,S(10,49,0,79)*
XPJC1,1,000005,PERT,0,4095,3,1,
XPJC4,S(50,59,0,79)*
XPJC1,1,000005,WWY,0,4095,3,1,
XPJC4,S(60,123,0,79)*
XPJC1,1,000005,MAA,0,4095,3,1,
XPJC4,S(124,131,0,79)*
XPJC1,1,000005,LIBRARY,0,4095,3,1,
XPJC2,F,R,20,0,0,0,2,
XPJC4,S(132,143,0,79)*
XPJC1,1,000005,CATALOG,0,4095,3,4,
XPJC2,F,R,500,0,0,0,2,
XPJC4,S(144,146,0,79)*
XPJC1,1,000005,PROGRAM PAUT,0,4095,3,1,
XPJC4,S(147,167,0,79)*
*****
```

Rys. 4.2. Dane do programu XPJC służące do zakładania zbiorów na dysku 000005.

Program, ażeby mógł poprawnie działać korzysta z urządzeń wejścia- wyjścia przedstawionych w tablicy 4.1. Należy podkreślić, że urządzenia o numerze 5 i 6 są niezbędne (obligatoryjne) i są wykorzystywane przy każdym przebiegu programu. Pozostałe urządzenia są opcjonalne i na przykład przy braku odwołań do bibliotek zbiory 11 i 12 nie będą wykorzystywane.

Uproszczony schemat blokowy programu NAP2 przedstawiono na rysunku 4.3.





Rys. 4.3. Schemat blokowy programu NAP2



4.1.2: Zakres zastosowań programu NAP2

Program NAP2 jest uniwersalnym programem analizy nieliniowych układów elektronicznych i umożliwia wykonywanie następujących obliczeń:

- 1) w zakresie analizy stałoprądowej *DC
 - a) wyznaczenie stanu ustalonego dla danych wartości parametrów,
 - b) wyznaczanie stanu ustalonego dla różnych automatycznie zmienianych wartości parametrów, przy czym krok może być liniowy lub też logarytmiczny,
 - c) analizę wielkosygnałową,
 - d) określenie funkcji przenoszenia,
 - e) wyznaczenie wrażliwości,
 - f) analizę najgorszego przypadku (*WORST),
 - g) analizę szumową (*NOISE),
 - h) analizę temperaturową,
 - i) optymalizację punktu pracy,
 - j) optymalizację charakterystyk przejściowych dla prądu stałego;
- 2) w zakresie analizy przejściowej *TR
 - a) wyznaczanie odpowiedzi czasowych dla dużych sygnałów,
 - b) wyznaczenie odpowiedzi układu na skok jednostkowy,
 - c) wyznaczanie wrażliwości.
 - d) analizę Fouriera (*F),
 - e) analizę temperaturową;
- 3) w zakresie analizy w dziedzinie częstotliwości *AC
 - a) wyznaczanie odpowiedzi dla zmieniających liniowo lub logarytmicznie wartości częstotliwości,
 - b) obliczanie funkcji przenoszenia,
 - c) obliczenie wrażliwości.

Analiza dla prądu stałego jest inicjowana instrukcją sterującą *DC. W tej analizie wszystkie pojemności są traktowane jako przerwy w obwodzie, a indukcyjność jako zwarcia. Formułowanie równań odbywa się przy pomocy zmodyfikowanej metody potencjałów węzłowych [21], tak więc zmiennymi pierwotnymi są napięcia węzłów i prądy w impedancjach. W celu zmniejszenia liczby równań opisujących układ, rezystory niezerowe są modelowane przez przewodności, których prądy nie są zmiennymi pierwotnymi. Wstawienie do nazwy rezystora przedrostka RR zamiast R powoduje, że jego prąd będzie odpowiedzią pierwotną.



Po ułożeniu równań są one rozwiązywane metodą dekompozycji macierzy rzadkiej (LU). Jeżeli obwód jest nieliniowy to stosuje się podstawową metodą Newtona-Raphsona. Po znalezieniu punktu pracy metodą analizy stałoprądowej są tworzone linearyzowane modele dla małych sygnałów, które następnie mogą być wykorzystywane do analizy częstotliwościowej.

Analiza stanów przejściowych jest inicjowana instrukcją sterującą *TR lub też, gdy jest wymagane automatyczne określenie warunków początkowych, instrukcją *DCTR. Używając instrukcji *TR, która powoduje, że wektor warunków początkowych jest zerowy, można ustalić odpowiednie warunki początkowe przy pomocy *MODIFY.

Do całkowania równań różniczkowych jest użyta metoda Geara ze zmiennym rzędem i krokiem [13].

Przy obliczaniu wrażliwości w analizie dla prądu stałego są one obliczane metodą dołączonego układu małosygnałowego [12, 13, 22, 30], co zapewnia szybkie obliczenie jednej odpowiedzi obwodu względem wartości wszystkich elementów. Metoda ta staje się zawodna w analizie w dziedzinie czasu, dlatego też w analizie stanów nieustalonych wrażliwości są obliczane bezpośrednio z równań różnicowych utworzonych przez algorytm całkowania. Stwarza to duże wymagania pojemności maszyny i tylko mała liczba wrażliwości sieci może być obliczona podczas odpowiedzi czasowej. Do obliczeń wrażliwości wielkosygnałowych jest stosowana metoda perturbacji parametru.

4.1.3. Struktura danych wejściowych

Dane wejściowe wprowadza się na kartach perforowanych lub też, przy pracy pod systemem GEORGE 3, symuluje się plik kart danych za pomocą odpowiedniego zbioru dyskowego. Program może analizować kolejno szereg zestawów danych tak, jak to przedstawiono w poniższym przykładzie.



Zapamiętywanie modeli i makromodeli w bibliotekach

Pakiet 1	<pre> *CIRCUIT Zadania opisu obwodu i przywołujące modele z bibliotek Instrukcje sterujące analizą i wyjściem *RUN Wymiana, usuwanie lub nowe instrukcje sterujące *RUN ⋮ </pre>
Pakiet 2	<pre> *CIRCUIT ⋮ *RUN </pre>
Pakiet 3	<pre> *CIRCUIT ⋮ *RUN *END </pre>

Przed rozpoczęciem właściwych obliczeń należy zapamiętać w bibliotece modele czy też instrukcje opisujące obwód, mające być później wykorzystywane.

Rozkaz ***CIRCUIT** zeruje wewnętrzną bazę danych i przygotowuje system do nowego opisu obwodu. Po wpisaniu sieci elektrycznej obwodu żąda się jego analizy za pomocą odpowiednich instrukcji sterujących (mających na początku gwiazdkę). Rozpoczęcie obliczeń jest inicjowane instrukcją ***RUN**, która przekazuje sterowanie do pakietu programu analizy. Obliczeń danego obwodu można dokonać wielokrotnie poprzez wymianę i usuwanie jego elementów lub też żądając nowych analiz poprzez wymianę instrukcji sterujących.

Następny pakiet rozpoczyna się od nowej karty ***CIRCUIT**. Ilość pakietów jest ograniczona tylko i wyłącznie dostępnym czasem komputera. Rozkaz ***END** kończy obliczenia i przekazuje sterowanie z powrotem do systemu operacyjnego instalacji.



4.2. Opis wprowadzenia danych do programu NAP2

W celu ułatwienia opisu języka wprowadzimy następujące reguły:

- 1) duże litery oznaczają wyrażenie zastrzeżone, takie jak przedrostki lub nazwy rozkazów,
- 2) małe litery oznaczają informację dostarczoną przez użytkownika,
- 3) nawias klamrowy { } oznacza, że dokładnie jeden z elementów umieszczonych w tym nawiasie jest obligatoryjny,
- 4) nawias kwadratowy [] oznacza, że jego zawartość jest opcjonalna, tzn. może lecz nie musi wystąpić,
- 5) trzy kropki ... wskazują, że jest dozwolona dowolna liczba poprzednich specyfikacji,
- 6) $x \ x \ x \ x$, $y \ y \ y \ y$, $z \ z \ z \ z$ - oznacza dowolną maksymalnie 8-znakową nazwę,
- 7) znak $\square$ oznacza spację (puste miejsce)

W programie jest stosowany układ SI, a przedrostek w nazwach elementów i odpowiadające im jednostki są następujące:

- rezystencja	R	OHM
- przewodność	G	MHO
- indukcyjność	L	HENR
- pojemność	C	FARAD
- napięcie	V	VOLT
- natężenie prądu	I	AMPER
- czas	TIME	SEKONDA
- częstotliwość	FREQ	HZ

W celu ułatwienia opisu danych wprowadzono następujące współczynniki skali:

T - 10^{12} , G - 10^9 , MEG - 10^6 , K - 10^3 , M - 10^{-3} , U - 10^{-6} , N - 10^{-9} , P - 10^{-12} , PHI - Φ ; $E^{\pm nn}$ - $10^{\pm nn}$.

Liczba może być zapisana w postaci całkowitej (z kropką dziesiętną) i wykładniczej (ze znakiem lub bez). Przy przyjętej konwencji - następujące wyrażenia:

250 2.5E2 0.25K 25E3 25E-5MEG 25E7M

oznaczają wszystkie tę samą liczbą.

Węzły obwodu mogą być numerowane w sposób dowolny od 0 do 999, przy czym zero przyjmuje się za węzeł masy i musi on być zdefiniowany.

25E7M



Program umożliwia analizę układów liniowych i nieliniowych zawierających rezystory, kondensatory, cewki źródła napięciowe i prądowe, niezależne lub sterowane prądem, napięciem czy też pochodną prądu lub napięcia, indukcyjności wzajemne, tranzystory bipolarne lub polowe, diody i modele użytkownika. Każdy z tych elementów jest identyfikowany przez maksymalnie 8-znakową nazwę. Typ elementu jest określony poprzez pierwszą literę.

Przykłady

*	- Rozkaz	*CIRCUIT, *MODIFY
R, RR	- Rezystancja	R1, RWE, RWY
G	- Przewodność	GO, G1, GWY
L	- Indukcyjność	L1, LOBC
C	- Pojemność	C1, CB
V	- Sterowane źródło napięciowe	VOAMP, VRM
I	- Sterowane źródło prądowe	IBETA, TRM, IRG
M	- Sprzężenie wzajemne	M12, MRL
T	- Wbudowany model tranzystora lub diody	TR1, TD2
Q	- Odwołanie do modelu użytkownika	QMODEL, Q1

Do opisu nieliniowości i parametrów modeli wbudowanych służą:

Przykłady

- parametry	X1, PARAM
- funkcje	FSIN, EXP, F1
- listy parametrów	BC107, N3055

Nazwy parametru, funkcji lub listy parametrów nie mogą mieć przedrostka V, I, W lub PW, gdyż przedrostki te są wykorzystywane przy odwołaniu się do odpowiedzi obwodowych:

Przykłady

V	- potencjał węzła lub napięcie na elemencie	V1, V10, VR1, VG10
I	- prąd przepływający przez element	IR1, IIBETA
PW	- moc wydzielana w elemencie	PWR1
W	- wartość współczynnika sprzężenia lub elementu	WIBETA, WVRM, WR1



Odwołanie się można wykonać tylko do uprzednio wprowadzonych instrukcji. Oprócz odwołania się do odpowiedzi obwodu można odwołać się też do wartości nazwanej:

Przykłady

- nazwa elementu	R1, C1, L1
- nazwa elementu, nazwa źródła	RIN.E, R1.MA
- nazwa parametru	X1, TEMP
- nazwa funkcji, nazwa podparametru	FSIN.A F1.B
- nazwa listy, nazwa podparametru	BC107.AF, N3055.AR

Należy zauważyć, że napięcia węzłów i straty mocy w elementach mogą być użyte tylko w listach wyjściowych, podczas gdy odpowiedzi napięciowa lub prądowa elementu mogą być również użyte jako argumenty w wartościach funkcjonalnych lub sprzężeniach liniowych i nieliniowych.

W dalszej części opisu będą często używane pojęcia pierwotnej odpowiedzi obwodowej, pod którymi rozumiemy napięcie węzłowe lub prąd elementu wprowadzonego za pomocą żyratora, tzn. prąd cewki, źródła napięciowego, rezystora o wartości zerowej lub z przedrostkiem RR w nazwie. Funkcje przenoszenia i wrażliwości można obliczyć tylko na podstawie odpowiedzi pierwotnych.

4.3. Zdania opisu obwodu

4.3.1. Elementy bierne

W programie NAP2 mogą występować wszystkie typy elementów biernych. Zdanie opisujące elementy typu R i G jest postaci

$\left\{ \begin{array}{l} Rxxx \\ RRxx \\ Gxxx \end{array} \right\}$	nr węzła 1 nr węzła 2 wartość [lista źródeł]
--	--

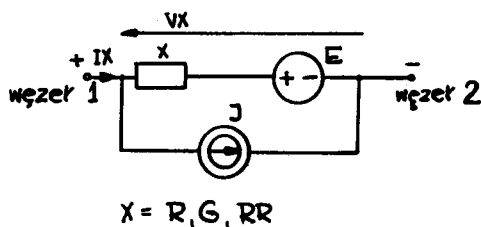
przy czym:

- wartość: stała lub funkcjonalna (p.4.3.8)

- lista źródeł:
$$\left\{ \begin{array}{ll} [E \text{ wartość}] & [J \text{ wartość}] \\ [RE \text{ wartość}] & [IM \text{ wartość}] \\ [MA \text{ wartość}] & [PH \text{ wartość}] \end{array} \right\}$$



Kierunki odniesienia pokazano na rysunku 4.4. Przyjęcie takiej konwencji powoduje, że pobór mocy przez źródło będzie ujemny. Należy zwrócić uwagę, że odpowiedzi prądowe i napięciowe elementu ze źródłami niezależnymi, są prądem czy też napięciem całej gałęzi.



Rys. 4.4. Kierunki odniesienia dla elementów biernych typu R lub G

Niezależne źródła w dziedzinie czasu są oznaczone przez E dla źródła napięciowego i przez J dla źródła prądowego. W przypadku analizy w dziedzinie częstotliwości są one ignorowane. Przy korzystaniu ze źródła w dziedzinie częstotliwości możemy ustalić jego wartość zarówno poprzez podanie amplitudy i fazy (MA i PH) jak też poprzez podanie części rzeczywistej i urojonej (RE i IM). Źródło prądu przemiennego jest źródłem napięciowym jeżeli element X jest rezystancją lub też źródłem prądowym jeżeli jest on konduktancją. W przypadku analizy w dziedzinie czasu źródła prądu przemiennego są ignorowane.

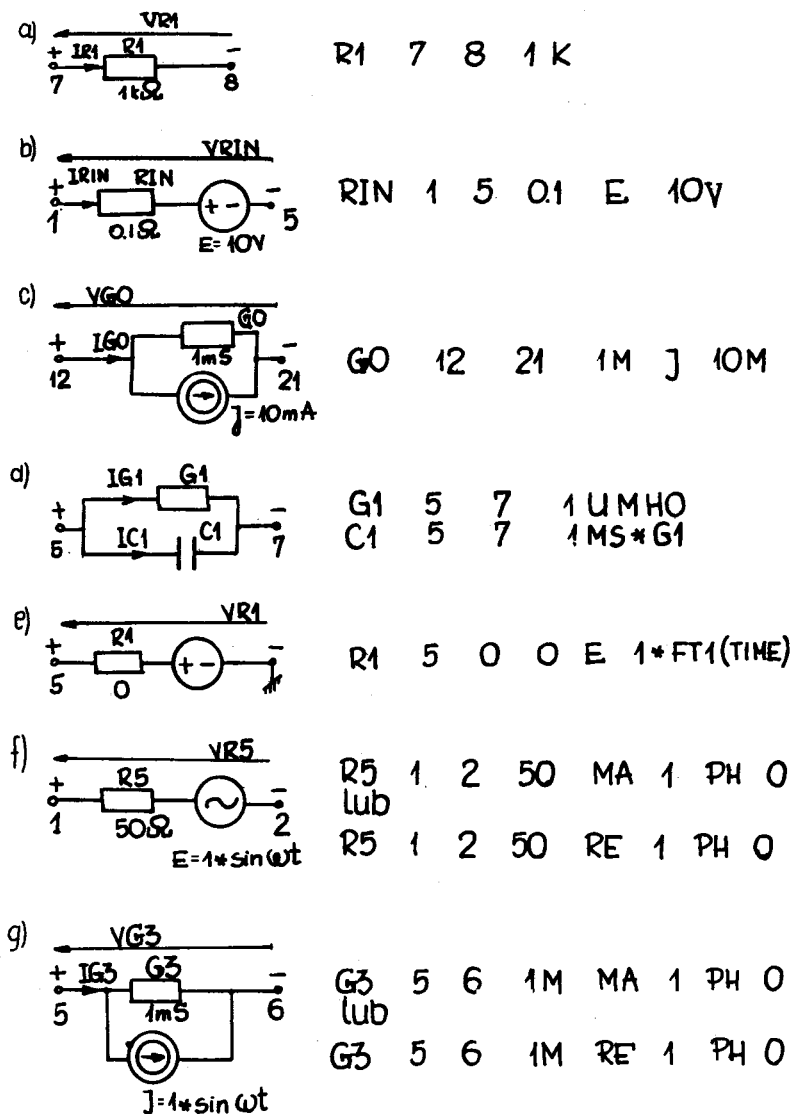
Na rysunku 4.5 przedstawiono szereg przykładów ilustrujących sposób wprowadzenia elementów biernych typu R , G do programu NAP2.

Na rysunku 4. 5a rezystancja $R1$ o wartości $1\text{ k}\Omega$ jest włączona pomiędzy węzeł 7 i 8, na rysunku 4.5b niezależne źródło napięciowe prądu stałego o wartości 10 V jest połączone w szereg z rezystancją o wartości $0,1\text{ }\Omega$

Na rysunku 4.5c niezależne źródło prądowe jest dołączone równolegle do konduktacji $G0 = 1\text{ mS}$.

Na rysunku 4.5d pojemność $C1$ jest funkcją konduktacji $G1$. Stała czasowa gałęzi $C1-G1$ jest zawsze równa 1 mS niezależnie od zmian wartości $G1$ ($G1 \neq 0$).





Rys. 4.5. Przykłady ilustrujące sposób wprowadzania elementów biernych typu R i G do programu NAP2



Na rysunku 4.5e przedstawiono idealne źródło napięciowe (ponieważ $R_l=0$), będące funkcją czasu. Zależność funkcyjna F_{l1} musi być zdefiniowana uprzednio za pomocą instrukcji funkcyjnej.

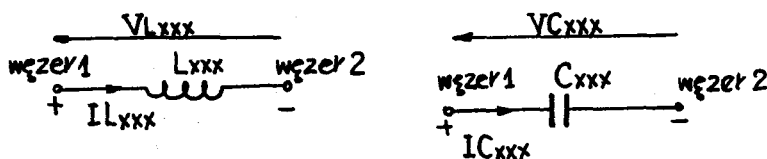
Rysunki 4.5f i 4.5g przedstawiają źródła napięciowe i prądowe prądu przemiennego opisane przez amplitudę i kąt fazowy w stopniach bądź też przez część rzeczywistą i urojoną.

Zdanie opisujące elementy L i C jest postaci

$\begin{Bmatrix} L_{xxx} \\ C_{xxx} \end{Bmatrix}$	nr węzła 1	Nr węzła 2	wartość
--	------------	------------	---------

gdzie: wartość - stała lub funkcjonalna (patrz p.4.3.8); kierunki odniesienia pokazano na rysunku 4.6.

Sposób wprowadzenia elementów L, C jest podobny jak R i G, lecz nie wolno deklarować wraz z nimi źródeł.



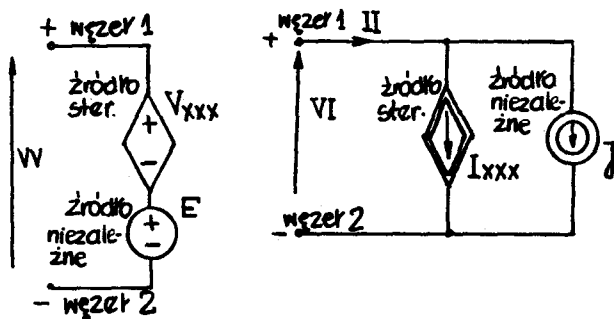
Rys. 4.6. Kierunki odniesienia dla elementów biernych typu L i C

4.3.2. Źródła sterowane

Program NAP2 dopuszcza wszystkie cztery typy źródeł sterowanych [13] oraz również źródła, które są sterowane przez pochodną prądu i napięcia.

Na rysunku 4.7 przedstawiono kierunki odniesienia dla źródeł sterowanych.





Rys. 4.7. Kierunki odniesienia dla źródeł sterowanych

Zdanie opisujące źródła sterowane jest postaci

$\begin{Bmatrix} V_{xxx} \\ I_{xxx} \end{Bmatrix}$	nr węzła 1	nr węzła 2	wartość	$\begin{Bmatrix} V \\ I \\ SV \\ SI \end{Bmatrix}$	nazwa elementu [lista źródeł]
--	------------	------------	---------	--	----------------------------------

gdzie: wartość

sprężenie

- stała lub funkcjonalna (patrz p.4.2.3)

- liniowe lub nieliniowe typu:

V - sterowane przez napięcie

I - sterowane przez prąd elementu

SV - sterowane przez dV/dt

SI - sterowane przez dI/dt

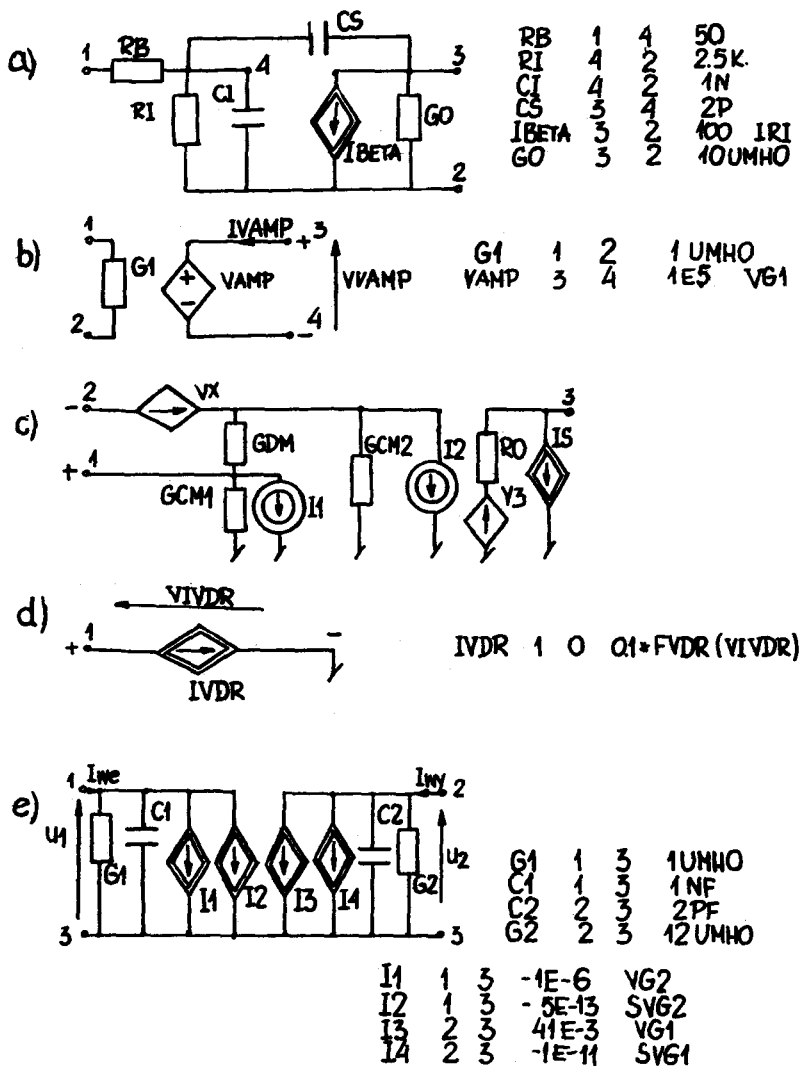
jeśli jest dokonywana analiza w dziedzinie częstotliwości, wówczas SV lub SI oznacza, że współczynnik wzmocnienia jest zespolony i równa się $j\omega$ *wartość;

lista źródeł - taka sama jak przy elementach biernych;

nie jest dozwolona przy sprężeniu typu S.

Na podkreślenie zasługuje fakt, że źródła napięciowe sterowane swoim własnym prądem i źródła prądowe sterowane swym własnym napięciem są traktowane jako elementy bierne i mogą być użyte jako elementy sterujące.





Na rysunku 4.8 przedstawiono szereg przykładów ilustrujących sposób wprowadzania źródeł sterowanych. Rysunek 4.8a przedstawia model typu hybryd $\bar{X}$ tranzystora dla małych sygnałów. IBETA jest źródłem prądowym sterowanym przez prąd opornika RI. Zamiast tego można również użyć źródła prądowego sterowanego napięciem na tym oporniku

IBETA 3 2 40MMHO VRI

Wartość opornika RI jest $2,5 \text{ k}\Omega$, tak więc oba wyrażenia są sobie równoważne.

Rysunek 4.8b przedstawia wzmacniacz napięciowy o wzmacnieniu równym 1E5 i konduktacji wejściowej równej $1 \mu\text{S}$. Rysunek 4.8c przedstawia makromodel stałoprądowy M20A wzmacniacza operacyjnego $\mu\text{A} 741$ [46, 48]. Źródło napięciowe VX jest sterowane poprzez funkcję FX (opis funkcji w p. 4.3.7) napięciem na konduktacji GCM2. Źródło napięciowe V3 dołączone szeregowo do opornika RO jest sterowane poprzez funkcję FE3 napięciem na konduktacji GDM. Źródło prądowe IS służące do ograniczenia prądu wyjściowego jest sterowane prądem rezystancji RO poprzez funkcję FIS.

Na rysunku 4.8d przedstawiono źródło prądowe, które jest sterowane swoim własnym napięciem. Nieliniowa charakterystyka musi być uprzednio zdefiniowana poprzez funkcję FVDR. Ostatni przykład (rysunek 4.8e) symuluje czwórnik opisany następującymi równaniami [71]:

$$I_{we} = (10^{-3} + j\omega 10^{-9}) U_1 + (-10^{-6} - j\omega 5 \cdot 10^{-13}) U_2,$$

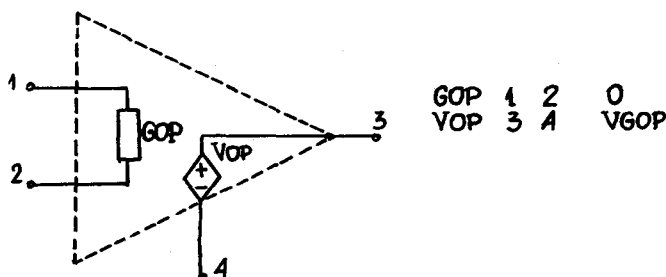
$$I_{wy} = (41 \cdot 10^{-3} - j\omega 10^{-11}) U_1 + (1,2 \cdot 10^{-5} + j\omega 2 \cdot 10^{-12}) U_2.$$

W celu zasymulowania idealnego wzmacniacza operacyjnego program NAP2 umożliwia realizację źródła napięciowego o nieskończeniu dużym wzmacnieniu. Deklaracja tego źródła jest postaci

V_{xxx}	nr węzła 1	nr węzła 2	$\begin{Bmatrix} V \\ I \end{Bmatrix}$	nazwa elementu
-----------	------------	------------	--	----------------

Na rysunku 4.9 przedstawiono idealny wzmacniacz operacyjny o nieskończeniu dużym wzmacnieniu, nieskończenie dużej impedancji wejściowej i zerowej impedancji wyjściowej.





Rys. 4.9. Idealny wzmacniacz operacyjny

4.3.3. Szeregowe dołączanie elementów

Elementy typu R, L i V mogą być dołączone szeregowo do już zadeklarowanego elementu typu R lub L. Często pozwala to na przyspieszenie obliczeń, gdyż zmniejsza się liczba węzłów, a co za tym idzie liczba równań układu. Postacie deklaracji są następujące:

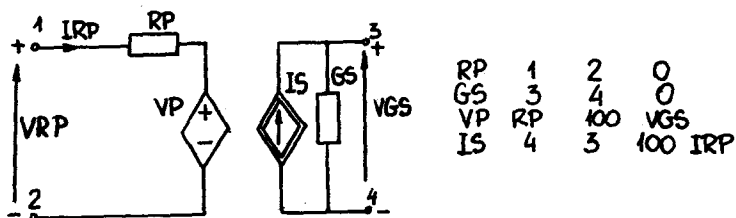
$$R_{xxx} \begin{Bmatrix} R_{yyy} \\ L_{yyy} \end{Bmatrix} \text{ wartość [lista źródeł]}$$

$$L_{xxx} \begin{Bmatrix} R_{yyy} \\ L_{yyy} \end{Bmatrix} \text{ wartość}$$

$$V_{xxx} \begin{Bmatrix} R_{yyy} \\ L_{yyy} \end{Bmatrix} \text{ wartość [sprężenie]}$$

Na rysunku 4.10 przedstawiono idealny transformator, którego napięcie pierwotne jest 100 razy większe od napięcia wtórnego, a prąd wtórny 100 razy większy od prądu pierwotnego. Źródło napięciowe VP jest tu dołączone szeregowo do opornika RP. Odpowiedź napięciowa tego opornika będzie teraz przyjmowana jako napięcie całej gałęzi, co pokazano na rysunku 4.10. Szeregowe dołączenie elementu pokazano również na rysunku 4.8c, gdzie sterowane źródło napięciowe V3 jest dołączone szeregowo do opornika R0.





Rys. 4.10. Idealny transformator

4.3.4. Równoległe dołączanie elementów

Elementy typu G, C oraz I mogą być dołączone równolegle do elementu typu G lub C. Zwiększa to liczbę układu, ponieważ prąd dopływający do gałęzi staje się odpowiedzią pierwotną. Postacie deklaracji są następujące:

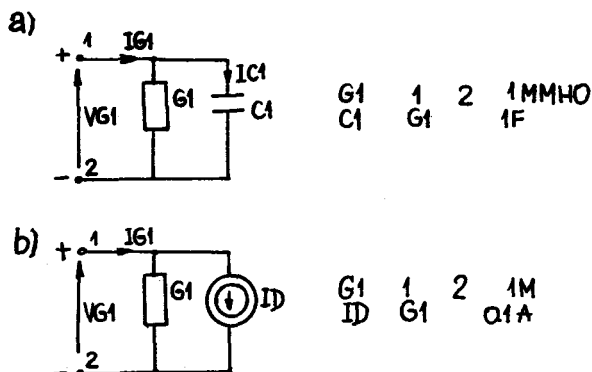
$$G_{xxx} \begin{Bmatrix} G_{yyy} \\ C_{yyy} \end{Bmatrix} \text{ wartość } [\text{lista źródeł}]$$

$$C_{xxx} \begin{Bmatrix} G_{yyy} \\ C_{yyy} \end{Bmatrix} \text{ wartość}$$

$$I_{xxx} \begin{Bmatrix} G_{yyy} \\ C_{yyy} \end{Bmatrix} \text{ wartość } [\text{sprężanie}]$$

Na rysunku 4.11 pokazano sposoby równoległego dołączania elementów. Odpowiedź prądowa gałęzi G1 jest traktowana jako prąd całkowity gałęzi.

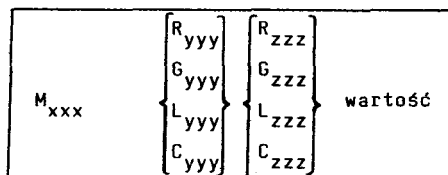




Rys. 4.11. Równoległe dołączanie elementów:
a) kondensatora, b) źródła prądu

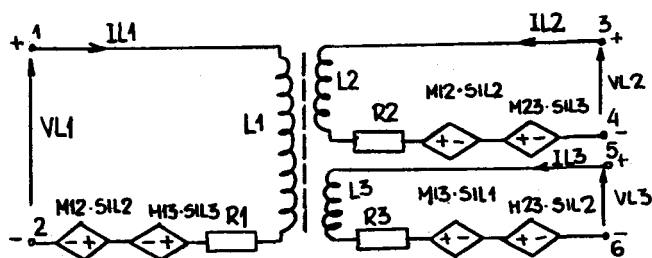
4.3.5. Sprzężenia wzajemne

Sprzężenie wzajemne może być utworzone pomiędzy dowolnymi dwoma elementami biernymi dołączonymi do węzłów. Instrukcja ma następującą postać:



Typ sprzężenia wzajemnego jest zależny od stosowanych elementów. Na rysunku 4.12 pokazano model nieidealnego transformatora z dwoma uzwojeniami wtórnymi i silnym sprzężeniem, otrzymany przy pomocy stosowania indukcyjności wzajemnych. Rezystancje uzwojeń $R1$, $R2$, i $R3$ są połączone szeregowo z indukcyjnościami.





L1	1	2	.1 HENRY
R1	L1		.1
L2	3	4	6.25 MH
R2	L2		.1
L3	5	6	6.25 MH
R3	L3		.1
M12	L1	L2	24.75MH : K12 = 0.99
M13	L1	L3	24.75MH : K13 = 0.99
M23	L2	L3	6.1875MH : K23 = 0.99

Rys...4.12. Model nieidealnego transformatora

4.3.6. Instrukcja parametrowa

Za pomocą instrukcji parametrowej mogą być zdefiniowane pojedyncze wartości danych lub funkcje. Postać instrukcji jest następująca:

nazwa parametru = wartość

gdzie: nazwa parametru jest dowolna, lecz nie zaczynająca się od liter V, I, W lub PW; jeżeli nazwa nie zaczyna się również od M, R, L, C, G, I, to można pominąć znak =.



Pięć parametrów zostało wbudowanych na stałe i nie mogą one być definiowane przez użytkownika. Są to:

TIME - czas, zmienna niezależna w analizie stanu nieustalonego,
 FREQ - częstotliwość (część urojona pulsacji zespolonej podzielona przez 2π),

SIGMA - część rzeczywista pulsacji zespolonej,

CONTROL - parametr użyty do automatycznego zakończenia analizy; analiza zostaje zakończona gdy wartość parametru CONTROL zmienia się z wartości nieujemnej na ujemną,

TEMP - temperatura, standardowa wartość programowa jest 25°C .

Wszystkie parametry wbudowane (poza TEMP) mają standardową wartość programową równą zeru. Można je zmieniać za pomocą instrukcji *MODIFY. Poniżej podano kilka przykładów instrukcji parametrycznych:

a) .TEMP = 100: ZMIANA TEMPERATURY NA 100 DEG

b) MOD = 1 * ABS (IRZ): FUNKCJA ABS MUSI BYĆ UPRZEDNIO ZDEFINIO-
 WANA

c) X1 = 1 * IR5 * VR5: OBLICZENIE MOCY TRACONEJ W OPORNIKU R5

4.3.7. Instrukcja funkcyjna

Instrukcja funkcyjna służy do zdefiniowania wyrażeń funkcyjnych dla wartości funkcjonalnych lub sprzężeń nieliniowych. Jest ona postaci:

nazwa funkcji / typ funkcji / lista podparametrów

gdzie: nazwa funkcji-może być dowolna o ile pierwszą literą nie jest I; V, W, L, PW; typ funkcji-dozwolone są następujące typy "L", ABS, EXP, LOG, SIN, TAN, ATAN; TAB1, TAB2, SUB; lista parametrów-składa się z trzech list - listy 1, listy 2 i listy 3.

Lista 1 jest postaci następującej:

A. wartość B wartość C wartość D wartość E wartość

Parametry zawarte na tej liście służą do obliczenia funkcji o następującej postaci:



$$f(X) = A + B \left[\text{ftyp} \left(\frac{X - C}{D} \right) \right]^E, \quad (4.1)$$

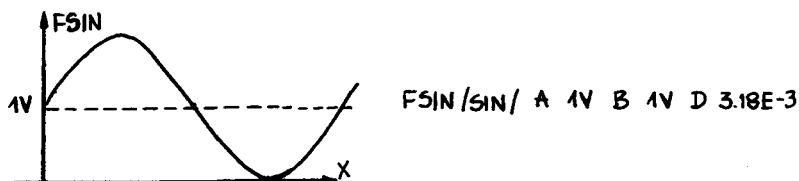
gdzie: X - stała * argument zależności funkcjonalnej lub sprzężenia nieliniowego, odwołującego się do zależności funkcjonalnej;

ftyp - jedna ze standardowych funkcji FORTRANU 1900: ABS, EXP, LOG, SIN, TAN, ATAN.

Wartości parametrów listy 1 wbudowane do programu są następujące:

$$A = 0, \quad B = 1, \quad C = 0, \quad D = 1, \quad E = 1.$$

Jako przykład użycia instrukcji funkcyjnej z wykorzystaniem listy 1 pokazano na rysunku 4.13, funkcję FSIN, która jest funkcją sinusoidalną o amplitudzie 1 V i składowej stałej 1 V. Częstotliwość wynosi 50 Hz, ponieważ $D = 0.02/2\pi$.



Rys. 4.13. Przykład wykorzystania listy 1 do opisu zależności funkcyjnej

Lista 2 jest w postaci następującej:

P wartość	L wartość	U wartość	SL wartość	SU wartość
SM wartość				

Precyzuje ona ograniczenia na funkcję $f(X)$. Są one zawarte w tablicy 4.2



Parametr	Komentarz	Standardowa wartość programowa
P	Okres - jeżeli $P > 0$, to $f(x)$ będzie funkcją okresową o okresie równym P	0
L	Dolna granica zmiennej X, jeżeli $X < L$, to jest stosowana aproksymacja liniowa $f(x) = f(L) + f'(L) (X-L)$	- 1E74
U	Górna granica zmiennej X, jeżeli $X > U$, to jest stosowana aproksymacja liniowa $f(x) = f(U) + f'(U) (X-U)$	- 1E74
SL	Nachylenie przy dolnej granicy, jeżeli $X < L$ i $SL \neq 0$, to $f(x) = f(L) + SL (X-L)$	0
SU	Nachylenie przy górnej granicy, jeżeli $X > U$ i $SU \neq 0$, to $f(x) = f(U) + SU (X-U)$	0
SM	Minimalne dopuszczalne nachylenie charakterystyk nieliniowych, jeżeli $f'(x) < SM$, to $f'(x) = SM$	- 1E74

Jako przykład użycia instrukcji funkcyjnej z wykorzystaniem listy 2, pokazano na rysunku 4.14 funkcję FVDR, która może symulować charakterystykę warystora. Dla $X > 6$ lub $X < -6$ jest stosowana aproksymacja liniowa. Jeżeli chcielibyśmy uzyskać nachylenie różne od pochodnej w punkcie ograniczenia, należałoby zadeklarować odpowiednie różne od zera wartości SU i SL.

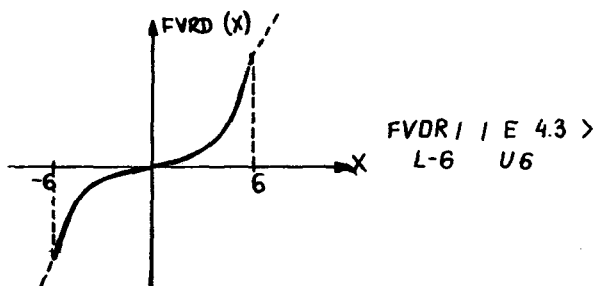
Lista 3 jest postaci następującej:

Wartość	Wartość	...
---------	---------	-----

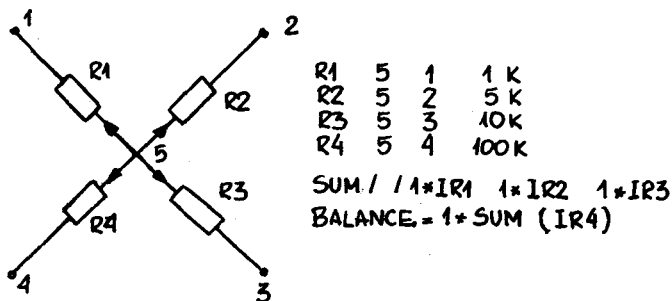
jest to po prostu nienazwany szereg wartości, które dodają się do funkcji $f(X)$

$$f(x) = A + B \times \left[\text{ftyp} \frac{X-C}{D} \right]^E + \text{wartość} + \text{wartość} + \dots \quad (4.2)$$





Rys. 4.14. Przykład wykorzystania listy 2 do opisu zależności funkcyjnej



Rys. 4.15. Przykład wykorzystania listy 3 do opisu zależności funkcyjnej

Na rysunku 4.15 pokazano przykład użycia listy 3 do określania funkcji w postaci

$$\text{SUM}(X) = X + \text{IR1} + \text{IR2} + \text{IR3},$$

jeżeli teraz wywołamy tę funkcję przy obliczeniu parametru BALANCE, to będzie on sprawdzał niezrównowagę prądową dla węzła 5, gdyż po podstawieniu w miejsce X prądu rezystora IR4 otrzymamy

$$\text{BALANCE} = \text{IR4} + \text{IR1} + \text{IR2} + \text{IR3},$$

oprócz funkcji standardowych, występujących w języku FORTRAN 1900, istnieją trzy funkcje specjalne typu TAB1, TAB2 oraz SUB. Dla funkcji



TAB1 i TAB2 wartości na liście 3 mają specjalne znaczenie i zostaną omówione osobno. Funkcja typu TAB1 jest funkcją tabelaryczną o równo odległych wartościach na osi X. Postać instrukcji tej funkcji jest następująca:

nazwa funkcji / TAB1 / C wartość D wartość y1 y2 ...

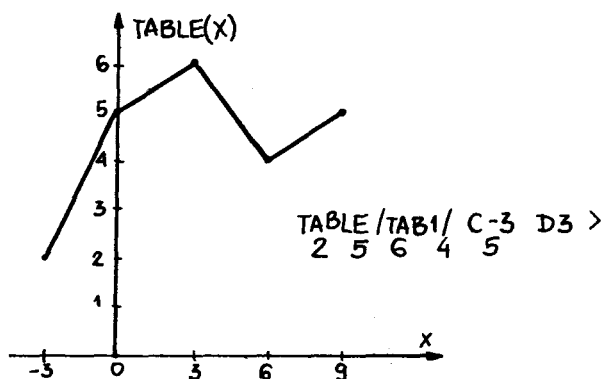
gdzie:

$C = X_{\min}$,

D - krok ΔX przyrostu na osi X

y1, y2... - wartości funkcji $f(x)$ w kolejnych punktach $X_{\min}$,
 $X_{\min} + \Delta X$, $X_{\min} + 2 \Delta X$...

Na rysunku 4.16 podano przykład użycia funkcji TAB1 do opisu zależności nieliniowej. Minimalna wartość na osi X jest równa -3, natomiast krok = 3.



Rys. 4.16. Przykład opisu nieliniowości przy pomocy funkcji TAB1

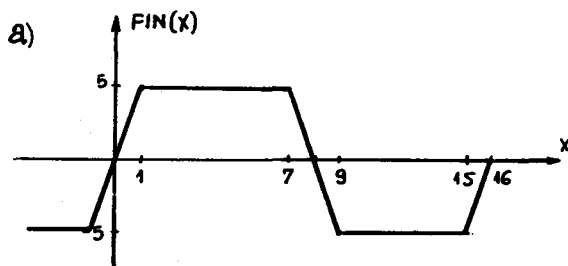
Funkcja typu TAB2 jest również funkcją tabelaryczną, przy czym jest ona definiowana przez podanie kolejno par wartości x, y. Postać instrukcji dla tej funkcji jest następująca:

nazwa funkcji / TAB2 / x_1 y_1 x_2 y_2 ...

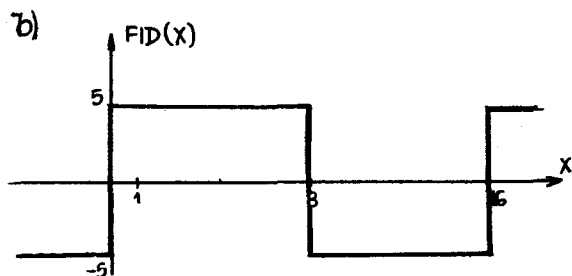
Na rysunku 4.17 pokazano przykłady użycia funkcji TAB2 do opisu okresowych zależności nieliniowych. Podanie parametru P = 16 mówi



o tym, że funkcja jest okresowa o okresie równym 16. Konieczne jest określenie tylko jednego okresu. Na rysunku 4.17b pokazano funkcję prostokątną o idealnie stromych zboczach. Deklaracja tej funkcji jest możliwa poprzez podwójne podanie wartości rzędnej y dla tej samej wartości odciętej x .



FIN/TAB2/ P 16 0 0 1 5 7 5 9 -5 15 -5 16 0



FID/TAB2/ P 16 0 -5 0 5 8 5 8 -5 16 -5 16 5

Rys. 4.17. Przykłady: a), b) opisu nieliniowości przy pomocy funkcji TAB2

Zupełnie innego typu funkcją jest funkcja typu SUB. Pozwala ona użytkownikowi na wprowadzenie funkcji opisanej procedurą w FORTRANIE 1900. Procedura SUB musi być dołączona do programu na etapie kompilacji. Ma ona następującą strukturę:



```

SUBROUTINE SUB (A, X, Y, D)
REAL A(1), X, Y, D
(opis podający sposób wyznaczania)
Y = F(X) oraz D = F'(X)
RETURN
END

```

Tablica A zawiera wielkości niezbędne do wyznaczenia Y i D, jak na przykład współczynniki, dodatkowe zmienne niezależne itd. D jest wykorzystywane wówczas, gdy funkcja jest użyta do opisu nieliniowości. W pozostałych przypadkach D nie musi być określone.

W wersji programu NAP2 VERSION PL84 skomplikowanych jest dziesięć podprogramów pozwalających na obliczanie funkcji hiperbolicznych, aproksymację wielomianową (obliczanie wartości wielomianów), na dokładne modelowanie tranzystorów MOS i ich pojemności oraz makromodelowanie bramek TTL i ECL. Postać procedury SUB sterującej wywołaniem odpowiednich procedur obliczeniowych jest następująca:

```

SUBROUTINE SUB(P, X, Y, D)
REAL P(20), X, Y, D
EXTERNAL FHYP, POLY, MOS, MOSCAP, FINP, FLOG, FCAP1, FCAP2,
FOUT, FINPE, FTHYR
INTEGER ISW
ISW = P(1) + 0.1
IF(ISW.EQ.1) CALL FHYP (P, X, Y, #8999)
IF (ISW.EQ.2) CALL POLY(P, X, Y, #8999)
IF(ISW.EQ.3) CALL MOS (P, X, Y, #8999)
IF(ISW.EQ.4) CALL MOSCAP(P,X, Y, #8999)
IF(ISW.EQ.5) CALL FINP (P,X,Y, #8999)
IF(ISW.EQ.6) CALL FLOG (P, X, Y, #8999)
IF(ISW.EQ.7) CALL FCAP1 (P, X, Y, #8999)
IF(ISW.EQ.8) CALL FCAP2 (P, X, Y, #8999)
IF(ISW.EQ.9) CALL FOUT (P,X, Y, #8999)
IF(ISW.EQ.10) CALL FINPE(P, X, Y, #8999)
IF(ISW.EQ.11) CALL FTHYR(P, X, Y, #8999)
WRITE(6,10)
10 FORMAT('0xxx ERROR NAP2, ONLY 11 EXTERNAL
SUBROUTINES AVAILABLE')
Y = 0
D = 0
STOP
999 RETURN
END

```



Parametrami sterującymi wywołanie odpowiedniej procedury jest P(1). Pozostałe parametry zależą od typu wywołanego modelu i zostaną opisane poniżej.

P(1) = 1 oznacza, że wywołana będzie procedura FHYP. Pozostałe parametry mają wówczas znaczenie następujące:

- P(2) - 1 oznacza obliczenie $\tanh$
- 2 oznacza obliczenie $\cosh$
- 3 oznacza obliczenie $\sinh$

P(3) - a

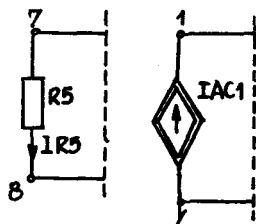
P(4) - b

P(5) - c

przy czym wyrażenie obliczające funkcję hiperboliczną jest postaci

$$Y = a X + b \text{ ifyp } (c X) . \quad (4.3)$$

Na rysunku 4.18 podano przykład użycia i wywołania funkcji FHYP.



$$\begin{array}{rcl}
 R5 & 7 & 8 \\
 \text{TANH / SUB /} & 1 & 1 \\
 IAC1 & 0 & 1 \\
 \text{czyli} & & \\
 IAC1 = 0.5 * IR5 + 0.3 * \tanh(2 * IR5) & & \\
 \quad \quad \quad \uparrow \quad \quad \quad \uparrow \quad \quad \quad \uparrow & & \\
 \quad \quad \quad a \quad \quad \quad b \quad \quad \quad c
 \end{array}$$

Rys. 4. 18. Przykład użycia procedury SUB do obliczania funkcji hiperbolicznej

P(1) = 2 oznacza, że będzie wywołana procedura POLY obliczająca wartość wielomianu. Pozostałe parametry mają wówczas znaczenie następujące:



$P(2)$ - rząd wielomianu
 $P(3)$ }
 $P(4)$ } - kolejne czynniki wielomianu:
 $\vdots$ } $P(3) X^{P(2)} + P(4) X^{P(2)-1} + \dots$

$P(1) = 3$ oznacza wywołanie procedury MOS służącej do modelowania charakterystyk tranzystora MOS [24]. Pozostałe parametry mają wówczas następujące znaczenie:

$P(2)$ - nie używamy,
 $P(3)$ - typ kanału: $P(3) = 0$ kanał typu N,
 $P(3) = 1$ kanał typu P,
 $P(4)$ - napięcie U_{GS} (argument),
 $P(5)$ - napięcie U_{BS} (argument),
 $P(6)$ - współczynnik wzmocnienia B (Parametr),
 $P(7)$ - napięcie progowe U_T (parametr),
 $P(8)$ - współczynnik redukcji ruchliwości (parametr),
 $P(9)$ - współczynnik modulacji długości kanału K_M (parametr),
 $P(10)$ - współczynnik nachylenia (body factor) K_B (parametr),
 $P(11)$ - potencjał wbudowany dren - podłoże 0,
 $P(12)$ - maksymalna wartość napięcia U_{DS} w pierwszej ćwiartce charakterystyk,
 X - napięcie U_{DS} (argument),
 Y - prąd drenu I_{DS} ,
 D - pochodna prądu drenu względem U_{DS} .

Przykłady wykorzystania procedury MOS i pozostałych procedur zostaną podane w p.4.4.4, dotyczącym modeli użytkownika.

$P(1) = 4$ oznacza wywołanie procedury MOSCAP służącej do obliczenia nieliniowych pojemności C_{GS} , C_{GB} i C_{GD} . Pozostałe parametry mają wówczas następujące znaczenie:

$P(2), P(3)$ - wskazują, która pojemność może być obliczana:
 jeżeli $P(3) = 1$, jest obliczana pojemność C_{GB} ,
 jeżeli $P(2) = 0$ i $P(3) = 0$, to jest obliczana pojemność C_{GS} ,
 jeżeli $P(2) = 1$ i $P(3) = 0$, to jest obliczana pierwsza pojemność C_{GD} ,
 $P(4)$ - napięcie U_{DS} ,
 $P(5)$ - pojemność dielektryczna C_{OX} ,
 $P(6)$ - napięcie progowe U_T ,
 $P(7)$ - ϕ , bulk - channel, "diode" potential (2 potencjały Fermiego podłoża),



$P(i + 13)$ - DVOLT

X - $X = 0 \rightarrow$ TTL NAND
 $X = 1 \rightarrow$ TTL NOR
 $X = 2 \rightarrow$ ECL NOR
 $X = -2 \rightarrow$ ECL OR
 $X = 10 \rightarrow$ funkcja MIN
 $X = 11 \rightarrow$ funkcja MAX

Zadeklarowanie parametrów $P(i+4)$ - $P(i+9)$ jako równych zero spowoduje przyjęcia przez nie wartości wbudowanych, podanych w nawiasach. Przyjęcie jako zerowych wartości parametrów $P(i+10)$ lub $P(i+12)$, spowoduje, że zależności temperaturowe lub też napięcia zasilania nie będą symulowane.

Argument X powoduje wybranie odpowiedniego typu bramki $X = 10$ i $X = 11$ jest zarezerwowane do wyboru minimalnej lub maksymalnej wartości sygnału z wszystkich sygnałów wejściowych.

Do opisu właściwości dynamicznych bramek służą procedury FCAP1 i FCAP2:

$P(1) = 7$ - oznacza wywołanie procedury FCAP1 służącej do dokładnej symulacji właściwości dynamicznych bramki,

$P(2)$ - napięcie RT.E

$P(3)$ - TRD

$P(4)$ - TRISE

$P(5)$ - TFD

$P(6)$ - TFALL

$P(7)$ - $P(12)$ - ciąg wartości OTR1N(4.31), ORT3N(0.129), ATEMP (3.75 m), DTEMP, AVOLT (1.01), DVOLT - VCT -

X - VCT - argument.

Jeżeli tak dokładna symulacja wartości dynamicznych, na jaką pozwala procedura FCAP1, nie jest potrzebna, to można posłużyć się procedurą FCAP2, która symuluje właściwości dynamiczne bramki w sposób uproszczony i - co za tym idzie - znacznie szybszy;

$P(1) = 8$ - wywołanie procedury FCAP2

$P(2)$ - napięcie RT.E

$P(3)$ - tp2H

$P(4)$ - tpHL

$P(5)$ - $P(6)$ - parametry OTR1N (4.31), ORT3N (0,129)

X - VCT - argument



Do symulacji stopnia wyjściowego bramki TTL służy procedura FOUT:

P(1) = 9 - wywołanie procedury FOUT

$P(2) = \begin{cases} 1 & \text{- obliczenie rezystencji wyjściowej RO} \\ 2 & \text{- obliczanie prądu IO} \end{cases}$

P(3), P(4) - ROL(40), ROH(77) - minimalna i maksymalna wartość rezystencji wyjściowej

P(5)-P(13) - OTR1N(4.81), OTR3N(0.129), MAYION(16.2 m), MTEMP(0), MVOLT(4m), ATEMP(3.75 m), DTEMP, AVOLT(1.01), DVOLT

P(14) - RT.E jeżeli P(2) = 2 i obliczany jest prąd IO

X - argument funkcji:

X = RT.E dla obliczeń RO

X = IRO dla obliczeń IO.

Makromodelowanie bramek ECL jest znacznie prostsze niż TTL i do ich modelowania napisano procedurę FINPE, która symuluje stopień wejściowy; stopień przejściowy jest symulowany przy pomocy procedury FLOG z deklaracją odpowiedniej wartości argumentu X:

P(1) = 10 - wywoływanie procedury FINPE

P(2) - P(11) - ciąg wartości DVOLT, DTEMP, AIN (1M), ATN (1V), ATEMPA (0.6 M), AVOLTA (0.13), AIIN (-1.43), AT2N(-1,2), OI2N (16 M), OI3N(21 M) do opisu charakterystyki wejściowej ECL.

Procedura FTHYR służąca do opisu modelu tyrystora jest jeszcze w trakcie testów i nie zostanie tu omówiona. Dokładny opis korzystania z procedur służących do symulacji tranzystora MOS i układów logicznych zostanie podany w p. 4.4; dotyczącym modeli użytkownika.

W celu oszczędniejszego wykorzystania pamięci maszyny cyfrowej nie zawsze jest celowe kompilowanie jednocześnie wszystkich procedur. Procedury dotyczące makromodeli bramek TTL i ECL są aktualnie skompilowane na dysku 000017 w programie NAP2 VERSION DIGITAL i korzystanie z nich jest możliwe tylko po otrzymaniu zgody prowadzącego zajęcia.

4.3.8. Wartości funkcjonalne

Pod pojęciem wartości funkcjonalnej (w programie NAP2 każda wartość może być funkcjonalna lub stała) rozumiemy wyrażenie o postaci

$\text{stała} \left\{ \begin{array}{l} * \\ / \end{array} \right\}$	nazwa funkcji ([stała *] argument)
---	------------------------------------



gdzie : pierwsza stała - musi być bezwzględnie określona,
 stała w nawiasie - opcjonalna,
 nazwa funkcji, - nazwa uprzednio zdefiniowanej instrukcji
 funkcyjnej określającej zależność funkcyj-
 ną,
 argument - wartości nazwane lub odpowiedzi obwodu.

Należy zauważyć, że wszystkie wartości mogą być funkcjami innych wartości lub funkcjami odpowiedzi obwodu w dziedzinie czasu dla poszczególnych elementów. Zmiana pewnej wartości spowoduje odpowiednią zmianę w wielkościach zależnych od niej. Ułatwia to opis nawet najbardziej skomplikowanych zależności nieliniowych.

W przypadku prostych operacji mnożenia lub dzielenia można je zdefiniować bez odniesienia funkcyjnego. Należy wówczas określić dwa argumenty

stała	$[*argument]$	$\left\{ \frac{*}{/} \right\}$	argument
-------	---------------	--------------------------------	----------

Przykłady

a) $SUPP = 1 * R0.E$

b) $FTV / - 25$
 $DTEMP = 1 * FTV(TEMP)$

c) $A = 1 * SUP * DTEMP$
 $FSIN/SIN/$

d) $TGY \ 0 \ 1 \ 0 \ J \ 1 * FSIN(TIME)$

Przykład a) i c) dotyczy prostych operacji podstawienia i mnożenia. W przykładzie b) i d) następuje odwołanie do uprzednio zdefiniowanej zależności funkcyjnej.

4.3.9. Sprzężenia nieliniowe

Elementy nieliniowe są definiowane charakterystykami, które dają zależność pomiędzy dwoma odpowiedziami obwodowymi. Można je modelować za pomocą elementów typu V lub I ze sprzężeniem nieliniowym. Zastępuje ono wartość i sprzężenie opisane w p. 4.3.2. Postać instrukcji jest następująca:

stała	$\left\{ \frac{*}{/} \right\}$	nazwa funkcji (	$[stała *]$	sprzężenie)
-------	--------------------------------	-----------------	---------------	--------------



gdzie :

- pierwsza stała - musi być podana,
- stała w nawiasach - opcjonalna,
- nazwa funkcji - nazwa uprzednio zdefiniowanej funkcji, opisującej charakterystykę elementu nieliniowego,
- sprzężenie - są dopuszczalne dwa typy sprzężeń,
- V nazwa elementu - element jest sterowany przez napięcie elementu,
- I nazwa elementu - element jest sterowany przez prąd elementu.

Należy podkreślić, że sprzężenia typu SI oraz SV nie mogą być używane w sprzężeniach nieliniowych, tak więc nieliniowe elementy pamiętające typu L i C nie mogą być wprowadzane w powyższy sposób. Można je wprowadzić na przykład stosując zależną funkcyjnie od napięcia lub prądu wartość C lub L (można tak również oczywiście wprowadzać też elementy R i G). Podejście to jest jednak znacznie gorsze, ponieważ sprzężenia nieliniowe są w każdej iteracji oszacowane do 1-go rzędu metodą ekstrapolacji Taylora, dając w wyniku małosygnałowy model elementu nieliniowego, który może być użyty do analizy dla prądu przemiennego. Natomiast wartości funkcyjne nie są ekstrapolowane, lecz tylko aktualizowane w każdej iteracji. Tak więc wartości elementów zależnych funkcyjnie od odpowiedzi będą dawały tylko wtedy poprawne modele dla prądu przemiennego, jeśli są rzeczywiście wartościami różniczkowymi.

Jako przykład można podać nieliniowy element symulowany przez źródło prądowe sterowane swoim własnym napięciem poprzez funkcję FVDR, która opisuje jego charakterystykę

IVDR 1 0 01 *FVDR (VIVDR)

4.3.10. Nieliniowe elementy magazynujące energię

Do elementów magazynujących energię zaliczamy: kondensatory, cewki i źródła prądowe lub napięciowe sterowane parametrem czasowym odpowiedzi obwodu (sprzężenie typu S). W przypadku, gdy elementy te są nieliniowe należy zdefiniować je jako dynamiczne lub różniczkowe kondensatory, cewki lub transmitacje. Na przykład dla funkcji ładunku $q(V)$ prąd kondensatora wynosi



$$i = \frac{dq(V)}{dt} = \frac{dq(V)}{dV} \frac{dV}{dt} \doteq C_d(V) \frac{dV}{dt}$$

i element nieliniowy, który ma być modelowany jest kondensatorem różniczkowym $C_d(V)$. W tym przypadku można określić funkcijną zależność wartości kondensatora i z rozwiązania nieliniowego został uzyskany poprawny odpowiednik dla prądu przemiennego AC, ponieważ wartość funkcjonalna jest już wartością różniczkową.

4.3.11. Elementy magazynujące energię zmieniające się w czasie

W celu poprawnego modelowania elementu magazynującego energię zmiennego w czasie, program NAP2 wymaga wprowadzenia przez użytkownika dodatkowego elementu. Na przykład prąd kondensatora zmieniającego się w czasie jest równy

$$i = \frac{d}{dt} [C(t) V(t)] = \frac{dC(t)}{dt} V(t) + C(t) \frac{dV(t)}{dt} = G(t) V(t) + C(t) \frac{dV(t)}{dt}, \quad (4.4)$$

tak więc oprócz kondensatora $C(t)$ należy wprowadzić jeszcze konduktancję

$$G(t) = \frac{dC(t)}{dt}. \quad (4.5)$$

4.4. Modele wbudowane w program i makromodele

Program NAP2 posiada wbudowane modele diody, tranzystora bipolarnego i tranzystora FET. Instrukcja pozwalająca na wywołanie modelu wbudowanego jest postaci

Txxx nr węzła 1 nr węzła 2 [nr węzła 3] nazwa listy

przy czym nazwa listy jest odwołaniem do uprzednio określonej listy parametrów, w której jest również określony typ elementu



nazwa listy / typ elementu / [lista podparametrów]
--

nazwa listy może być dowolna, lecz nie może się zaczynać od liter V, I, W lub PW.

W przypadku braku deklaracji listy podparametrów zostaną przyjęte ich wartości wbudowane w program

Dopuszczalne są następujące typy elementów:

DIODE - dioda złączowa PN,

NPN - bipolarny tranzystor złączowy NPN,

PNP - bipolarny tranzystor złączowy PNP,

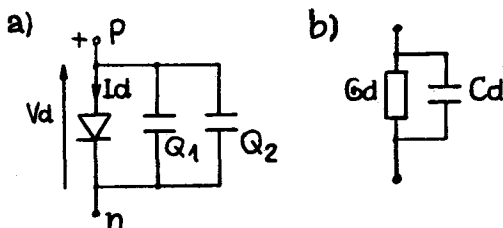
NJFET - tranzystor złączowy polowy z kanałem typu N,

PJFET - tranzystor złączowy polowy z kanałem typu P.

4.4.1. Dioda

Model diody wbudowany w program NAP2 pozwala zarówno na symulację jej właściwości stałoprądowych, jak też i dynamicznych

Na rysunku 4.19a jest przedstawiony wielkosygnałowy model diody, słuszny w całym zakresie prądów i napięć. Model z rysunku 4.1b, jest modelem małosygnałowym diody, słusznym tylko w danym punkcie pracy, dla którego zostały obliczone jego parametry.



Rys. 4.19. Model diody wbudowany w program NAP2: a) wielkosygnałowy, b) małosygnałowy

Poniżej przedstawiono równania opisujące wielko- i małosygnałowy model diody..



Model wielkosygnałowy	Model małosygnałowy
$I_d = I_s \left(\exp \frac{V_d}{V_T} - 1 \right)$	$G_d = \begin{cases} \frac{I_d + I_s}{V_T} & \text{dla } V_d > 0 \text{ i } G_d \leq G_s \\ G_s & \text{dla } V_d > 0 \text{ i } G_d > G_s \\ \frac{I_s}{V_T} & \text{dla } V_d < 0 \end{cases}$
$Q_1 = C_j \int_0^{V_d} \frac{dV}{\left(1 - \frac{V}{\phi}\right)^{1/2}}$	$C_d = \gamma_T G_d + \frac{C_j}{\left(1 - \frac{V_d}{\phi}\right)^{1/2}}$
$Q_2 = \tau_T I_d$	

gdzie:

$$V_T = V_{T0} \frac{T}{T_0}, \quad (4.6)$$

$$I_s = I_{S0} \left(\frac{T}{298} \right)^3 \exp \left(\frac{E_q}{V_{T0}} - \frac{E_q}{V_T} \right), \quad (4.7)$$

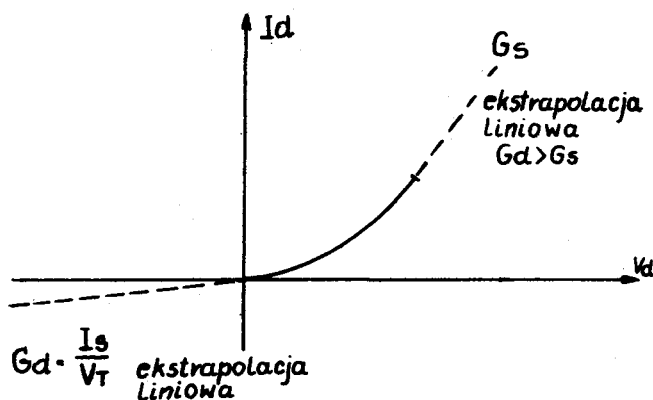
$$T_0 = 25^\circ\text{C},$$

$$T = 273^\circ + \text{TEMP},$$

$$E_q = 1,1 \text{ V}.$$

Charakterystykę statyczną diody przedstawiono na rysunku 4.20. Konduktacja G_s jest maksymalną konduktacją dynamiczną i musi być wyznaczana w zależności od typu diody. Parametry statyczne opisujące model diody zależą od temperatury. Potencjał termiczny V_{T0} odpowiada temperaturze 25°C , a właściwa wartość V_T jest funkcją temperatury. Prąd nasycenia I_{S0} jest również wartością określoną dla temperatury pokojowej i dla krzemu zmienia się z zależnością przedstawioną powyżej. $E_q = 1,1 \text{ V}$ jest napięciem przerwy energetycznej dla krzemu. Model diody opisuje 7 parametrów podanych w tabelicy 4.3.





Rys. 4.20. Charakterystyka statyczna diody

T a b l i c a 4.3

Symbol we wzorach	Nazwa w NAP2	Opis	Standard programowy
I_S	IS	- prąd nasycenia	IE-13
V_T	VT	- nkT/q , n-wsp. emisji	0.025
τ_T	TT	- czas przelotu	0
C_j	CJ	- pojemność statyczna złącza bez polaryzacji zewnętrznej	0
ψ	FI	- potencjał złącza	1
δ	GA	- wykładnik potęgi w równaniu pojemności diody	0.5
G_S	GS	- maksymalna konduktancja dynamiczna	1

W przypadku braku deklaracji danego podparametru jest przyjmowana jego wartość taka, jaka jest wbudowana w program (standard programowy).

Instrukcja odwołania do diody jest postaci

T_{xxx} węzeł anody węzeł katody nazwa listy

przy czym nazwa listy jest odwołaniem do uprzednio określonej listy parametrów, która dla diody jest postaci

nazwa listy / DIODE / I_S wartość V_T wartość τ_T wartość C_j
wartość ψ wartość δ wartość G_S wartość



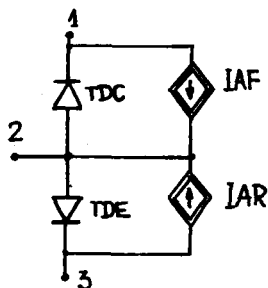
W celu odwołania się do elementów modelu diody, ich prądów i napięć, można posłużyć się następującymi odwołaniami:

- T_{xxx} - przewodność diody G_d ,
- IT_{xxx} - prąd diody (bez prądu pojemności) I_{Gd} ,
- VT_{xxx} - napięcie na diodzie V_{Gd} ,
- $T_{xxx} \text{ } t_{CD}$ - prąd pojemności diody I_{Cd} .

W tablicy 4.4 podano parametry wybranych diod dla programu NAP [60]

T a b l i c a 4.4

Parametr Typ diody	IS [A]	VT [mV]	GS [S]	TT [ns]	CJ [pF]	FI [V]	GA	Pojemność oprawki C_0 [pF]
BAVP 17	4.07E-9	57	1	5.4	1.5	1	.33	.5
BAVP 18	1.68E-8	51.2	1	30	4.5	1	.33	.7
BA 182	1.97E-12	37.13	.5693	30,6	1.8	.8	.5	.7
BAY 95	1.03E-15	22.4	.2956	3	.5	1	.5	.7
BAP 795	3.28E-15	23.9	.338	1.5	.2	1	.5	.5



Rys.4.21 Przykład odwołania się od modelu diody

Jako przykład wykorzystania modelu diody rozpatrzmy układ z rysunku 4.21, który symuluje model injekcyjny tranzystora bipolarnego.

Opis tego układu w języku programu NAP? będzie postaci:

```

DE / DIODE / IS 1E-14 VT 32 MV
DC / DIODE / IS 2E-14 VT 48 MV
TDC 2 1 DC
TDE 2 3 DE
IAF 1 2 .998 ITDE
IAR 3 2 .6 ITDC

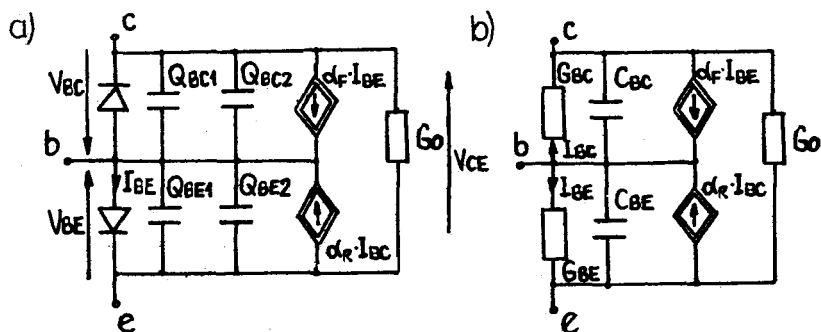
```



4.4.2. Tranzystor bipolarny

W program NAP2 jest wbudowany nieliniowy injekcyjny model Ebera-Malla [5] uzupełniony nieliniowymi pojemnościami, które uwzględniają efekty gromadzenia ładunku. Dodatkowo dla polepszenia symulacji właściwości statycznych tranzystora, wprowadzono funkcjonalną zależność konduktacji wyjściowej.

Na rysunku 4.22a jest przedstawiony wielkosygnałowy model tranzystora bipolarnego, słuszny w całym zakresie zmian napięć i prądów. Model z rysunku 4.22b jest modelem małosygnałowym tranzystora, słusznym tylko w danym punkcie pracy.



Rys. 4.22. Model tranzystora bipolarnego wbudowany w program NAP2 (npn): a) wielkosygnałowy, b) małosygnałowy

Poniżej przedstawiono równania opisujące wielko- i małosygnałowy model tranzystora.

Model wielkosygnałowy	Model małosygnałowy
$I_{BE} = I_S \left(\exp \frac{V_{BE}}{V_T} - 1 \right)$	$G_{BE} = \begin{cases} \frac{I_d + I_s}{V_T} & \text{dla } V_{BE} > 0 \\ iG_d < G_s \\ G_s & \text{dla } V_{BE} > 0 \text{ i } G_d > G_s \\ \frac{I_s}{V_T} & \text{dla } V_{BE} < 0 \end{cases}$



Model wielkosygnałowy	Model małosygnałowy
$I_{BC} = n_I I_S \left(\exp \frac{V_{BC}}{n_V V_T} - 1 \right)$	$G_{BC} = \begin{cases} \frac{I_d + I_s}{V_T} & \text{dla } V_{BC} \geq 0 \\ & \text{i } G_d \leq G_s \\ G_s & \text{dla } V_{BC} \geq 0 \text{ i } G_d > G_s \\ \frac{I_s}{n_V V_T} & \text{dla } V_{BC} < 0 \end{cases}$
$\left. \begin{aligned} Q_{BE1} &= C_{ej} \int_0^{V_{BE}} \frac{dv}{\left(1 - \frac{v}{\Psi}\right)^{\delta}} \\ Q_{BE2} &= \mathcal{L}_F \tau_F I_{BE} \end{aligned} \right\}$	$C_{BE} = \frac{C_{ej}}{\left(1 - \frac{V_{BE}}{\Psi}\right)^{\delta}} + \mathcal{L}_F \tau_F G_{BE}$
$\left. \begin{aligned} Q_{BC1} &= C_{cj} \int_0^{V_{BC}} \frac{dv}{\left(1 - \frac{v}{\Psi}\right)^{\delta}} \\ Q_{BC2} &= \mathcal{L}_R \tau_R I_{BC} \end{aligned} \right\}$	$C_{BC} = \frac{C_{cj}}{\left(1 - \frac{V_{BC}}{\Psi}\right)^{\delta}} + \mathcal{L}_R \tau_R G_{BC}$
	$G_O = G_Z + n_G (G_{BE} + G_{BC})$

Model tranzystora jest opisany przez 15 parametrów podanych w tablicy 4.5.

T a b l i c a 4.5

Symbol we wzorach	Nazwa w NAP2	Opis	Standard programowy
I_S	IS	Prąd nasycenia złącza baza - emiter	1E-13
V_T	VT	$n k T / q$, n -stała emisji dla złącza baza-emiter	0.025
n_I	NI	$n_I I_S$ jest prądem nasycenia złącza baza-kolektor	1
n_V	NV	$n_V V_T$ jest potencjałem termicznym diody baza-kolektor	1



T a b l i c a 4.5 (cd.)

Symbol we wzorach	Nazwa w NAP2	Opis	Standard programowy
α_F L_F	TF	Czas przelotu w kierunku prze- wodzenia w układzie wspólnej bazy	0
α_R L_R	TR	Czas przelotu w kierunku za- porowym w układzie wspólnej bazy	0
C_{ej}	CE	Pojemność baza-emiter bez po- laryzacji zewnętrznej	0
C_{cj}	CC	Pojemność baza-kolektor bez polaryzacji zewnętrznej	0
φ	FI	Potencjał złącza	1
γ	GA	Wykładnik w równaniu pojemno- ści	0.5
G_Z	GZ	Konduktacja wyjściowa bez po- laryzacji zewnętrznej	0
n_G	NG	Współczynnik proporcjonalno- ści dla konduktacji wyjścio- wej	0
β_F	AF	Wzmocnienie prądowe w kierun- ku przewodzenia w układzie wspólnej bazy	0.99
β_R	AR	Wzmocnienie prądowe w kierun- ku zaporowym w układzie wspólnej bazy	0.5
G_S	GS	Maksymalna konduktacja diody	1

Zależności temperaturowe są takie same dla obu złącz tranzystora
jak dla pojedynczej diody (p. 4.4.1).

Instrukcja odwołania do tranzystora jest w postaci

T_{xxx} nr węzła C Nr węzła B nr węzła E nazwa listy [α_F]

przy czym nazwa listy jest odwołaniem do uprzednio określonej listy
parametrów, która dla tranzystora jest postaci

nazwa listy / {NPN / PNP} / AF wartość AR wartość IS wartość NI wartość
VT wartość NV wartość TF wartość TR wartość CE wartość CC wartość FI
wartość GA wartość GZ wartość NG wartość GS wartość



Należy podkreślić, że wartości parametrów modelu są stałe lub funkcjonalne, tak więc umożliwia to tworzenie znacznie dokładniejszych modeli tranzystora [3, 4, 5]. Współczynnik wzmocnienia prądowego AF może - jako jedyny parametr - być określony lokalnie w instrukcji odwołania do tranzystora. Umożliwia to zróżnicowanie tranzystora tego samego typu pod względem wartości współczynnika wzmocnienia prądowego, co jest bardzo przydatne w praktyce.

W celu odwołania się do elementów modelu tranzystora, ich prądów i napięć można posłużyć się tablicą 4.6.

T a b l i c a 4.6

Odwołanie	Komentarz
T_{xxx}	- konduktacja kolektor-emiter G_o
IT_{xxx}	- prąd konduktacji kolektor-emiter I_{Go} (nie mylić z prądem emitera lub kolektora!)
VT_{xxx}	- napięcie kolektor-emiter V_{Go}
$T_{xxx} \cdot t_{GBE}$	- konduktacja diody baza-emiter G_{BE}
$IT_{xxx} \cdot t_{GBE}$	- prąd diody baza-emiter I_{GBE} (bez prądu pojemności)
$VT_{xxx} \cdot t_{GBE}$	- napięcie baza-emiter V_{BE}
$T_{xxx} \cdot t_{GBC}$	- konduktacja diody baza-kolektor G_{BC}
$IT_{xxx} \cdot t_{GBC}$	- prąd diody baza-kolektor I_{GBC} (bez prądu pojemności)
$VT_{xxx} \cdot t_{GBC}$	- napięcie baza-kolektor V_{BC}
$T_{xxx} \cdot t_{CBE}$	- pojemność baza-emiter C_{BE}
$IT_{xxx} \cdot t_{CBE}$	- prąd pojemności baza-kolektor I_{CBC}
$T_{xxx} \cdot t_{AP}$	- współczynnik $\mathcal{L}_F$ wzmocnienia prądowego w kierunku przewodzenia
$IT_{xxx} \cdot t_{AF}$	- prąd źródła AF = $\mathcal{L}_F I_{Be}$
$T_{xxx} \cdot t_{AR}$	- współczynnik $\mathcal{L}_R$ wzmocnienia prądowego w kierunku inwersyjnym
$IT_{xxx} \cdot t_{AR}$	- prąd źródła AR = $\mathcal{L}_R I_{BC}$

W tablicy 4.7 zamieszczono parametry modeli wybranych tranzystorów bipolarnych dla programu NAP2 [61].



T a b l i c a 4.7

Para- metr Typ transy- stora	IS [IE-13A]	VT [mV]	NI -	NV -	IF [NS]	TR [NS]	CE [pF]	CC [pF]	GA -	FI [V]	AF -	AR -	NG [IE-4]	GS [s]	GZ [s]
BD 135	0.8486	26.52	16.28	4.13	.55	116	60	20	.37	.76	.9924	.9091	3	3.1785	0
BD 136	0.71	28.76	14.16	4.14	.49	120	117	60	.39	.8	.9815	.833	3	3.3516	0
BD 255	7.28	26.52	4.0	1.057	.23	60	452	168	.4	.8	.995	.9615	5.4	3.9567	0
BD 354	6.1	26.3	2.04	1.0826	1.476	101	413	115	.37	.8	.9914	.927	1.9	4.6	0
BC 238A	1.197	26.52	21.8	1.156	.75	330	19.4	41.1	.45	.93	.9945	.829	2.52	.5057	0
BC 238B	1.8	26.52	6.52	1.1	.99	258	16.8	41.8	.55	.95	.9967	.908	4.84	.521	0
BC 239	0.292	26.3	18	1.14	.156	126	9.4	7.2	.44	.9	.9967	.911	2.63	.539	0
BC 308B	0.374	26.74	3.37	1.045	.583	97.8	9	13	.52	1.05	.9965	.85	8.02	.5263	0
BF 214	0.32	28.7	13.89	1.288	.204	45	2	1	.33	.8	.995	.85	5.47	.134	0
BF 180	0.067	28.76	7180.	1.972	.262	259	.6	.5	.5	1	.9938	.43	3.9	.167	0
KF 622	0.144	27.2	114.7	1.23	.224	226	10.1	6.85	.4	.8	.987	.8888	6.2	1.2436	0
BF 194	0.031	27.09	457	1.284	.31	97.8	3	1.5	.225	.86	.9914	.927	2.78	.35	0
BF 195	0.0344	27.19	444.4	1.284	.191	102	3.4	1.5	.247	.8	.9836	.7297	1.63	.348	0
UL 1111	0.047	28.17	2245.	1.32	.112	73.4	2.5	1.8	.4	.8	.991	.555	2.6	.11	0
85XP93	0.155	27.4	2.084	1.5	.22	5.7	5.1	3.3	.3	.8	.9892	.2	6.85	.66	0
BCP 107	0.783	25.6	4050.	1.54	.49	114	8.6	7.2	.43	.86	.99723	.75	2.5	.5	0
BCP 177	1.08	26.1	63.61	1.21	.52	78	11	12	.38	.8	.9924	.7143	116	.5	0
2N 3055	1783.	28.7	66.45	1.076	10	1000	1500	1500	.4	1	.9917	.923	1.15	3.68	0
BC 211	0.482	26.52	17.6	1.123	1	100	65	21	.4	.8	.9876	.8	1.32	2.16	0
BC 319	0.443	26.09	8.0	1.083	1.3	120	100	60	.4	.8	.9875	.9	6.5	2.48	0

Jako przykład rozpatrzmy fragment układu, w którym dwa tranzystory wykorzystują tę samą literę parametrów XY:

XY/NPN $\S$ IS 1E-11 VT 36MV AF .997 AR .7

TR1 8 9 10 XY

TR2 15 16 18 XY .996

POWER1 = 1 * VTR1 * ITR1.LGBE

POWER2 = 2 * VTR2 * ITR2.LGBE

Tranzystor TR2 ma zadeklarowany lokalnie współczynnik wzmocnienia prądowego AF = .996 i różni się od tranzystora TR1 tylko i wyłącznie wartością tego współczynnika. Parametry POWER1 i POWER2 określają pobór mocy przez poszczególne tranzystory jako iloczyn prądu emitera i napięcia na tranzystorze.

W przypadku, gdy chcemy określić w przybliżeniu parametry tranzystora, którego nie ma w tabelicy 4.7, możemy tego dokonać w przybliżeniu na podstawie danych znalezionych w katalogu.

Założmy, że dla tranzystorów BC 107 i BC 177 znaleźliśmy w katalogu dla punktu pracy, określonego jako $I_C = 2$ mA i $V_{CE} = 5$ V, dane zawarte w tabelicy 4.8.

T a b l i c a 4.8

Komentarz	Parametr	BC107	BC177
Pojemność złącza baza-emiter przy zerowej polaryzacji zewnętrznej	CE	12 pF	24 pF
Pojemność złącza baza-kolektor przy zerowej polaryzacji zewnętrznej	CC	5 pF	10 pF
Dla wybranego punktu pracy $I = 2$ mA; $V_{CE} = 5$ V mamy następujące parametry:			
- napięcie baza-emiter V_{BE}	V_{BE}	620 mV	650 mV
- częstotliwość graniczna	f_T	200 MHz	100 MHz
- parametry h dla układu wspólnego emitera i małych sygnałów (wzmocnienie prądowe h_{FE} dla dużych sygnałów)	$\begin{cases} h_{ie}(h_{11}) \\ h_{re}(h_{12}) \\ h_{FE}(h_{21}) \\ h_{oe}(h_{22}) \end{cases}$	$\begin{cases} 2.7 \text{ k}\Omega \\ 1.5 \text{ E-4} \\ 180 \\ 18 \text{ E-65} \end{cases}$	$\begin{cases} 2.5 \text{ k}\Omega \\ 2.3 \text{ E-4} \\ 140 \\ 38 \text{ E-65} \end{cases}$



T a b l i c a 4.8 (cd)

Komentarz	Parametr	BC107	BC177
W obszarze nasycenia dla $I_C = 10 \text{ mA}$ i $I_B = 0.5 \text{ mA}$ mamy:			
- napięcie nasycenia kolektor- -emiter	$V_{CE \text{ sat}}$	90 mV	78 mV
- napięcie nasycenia baza- -emiter	$V_{BE \text{ sat}}$	700 mV	700 mV

Wszystkie dane zawarte w tablicy 4.8 są słuszne w danym punkcie pracy modelu małosygnałowego. Wykorzystując więc model małosygnałowy z rysunku 4.22b możemy napisać

$$C_{BE} = \frac{I_{BE}}{VT}; \quad G_{BC} = \frac{I_{BC}}{V_{CT}};$$

$$C_{BE} = \mathcal{L}_F \tau_F G_{BE} + \frac{C_{Ej}}{\left(1 - \frac{V_{BE}}{\varphi}\right)^{\gamma}};$$

$$C_{BC} = \mathcal{L}_R \tau_R G_{BC} + \frac{C_{Cj}}{\left(1 - \frac{V_{CB}}{\varphi}\right)^{\gamma}};$$

$$G_O = G_Z + N_G (G_{BE} + G_{BC});$$

teraz wyrażamy te parametry poprzez parametry he

$$h_{ie} = \frac{1}{(1 - \mathcal{L}_F) G_{BE}}$$

$$h_{re} = \frac{(1 - \mathcal{L}_R) G_{BC}}{(1 - \mathcal{L}_F) G_{BE}}$$

$$h_{FE} = \frac{\mathcal{L}_F}{1 - \mathcal{L}_F}$$

$$G_{BE} = \frac{h_{FE}}{h_{ie}}$$

$$G_{BC} = \frac{1}{1 - \mathcal{L}_R} \frac{h_{re}}{h_{ie}}$$

$$\mathcal{L}_F = \frac{h_{FE}}{1 + h_{FE}}$$



$$h_{oe} = G_o + \frac{1 - \mathcal{L}}{1 - \mathcal{L}_F} G_{BC}$$

$$G_o = \left(\frac{h_{oe} h_{ie}}{h_{FE}} - h_{re} \right) G_{BE}$$

Następnie możemy przejść do oszacowania parametrów dla dużych sygnałów, jako najlepsze dopasowanie w punkcie pracy $I_C = 2 \text{ mA}$, $I_{CE} = 5 \text{ V}$, $V_{BE} = 620 \text{ mV}$ dla BC 107 i V_{BE} dla BC 177:

$AF = \frac{h_{FE}}{1 + h_{FE}}$	BC 107	BC 177
AR (szacunkowo)	0.9945	0.993
$VT = I_C \frac{h_{ie}}{h_{FE}}$	0.5	0.5
$I_S = I_C \exp \left(\frac{-V_{BE}}{V_T} \right)$	30 mV	36 mV
$NG = \left(\frac{h_{oe} h_{ie}}{h_{FE}} - h_{re} \right)$	2E-12	3E-11
$TF = \mathcal{L}_F \mathcal{L}_F \approx \frac{1}{2\pi f_T}$	1.2 E-4	4.5 E-4
TR = 0.5TF (szacunkowo)	0.8 NS	1.6 NS
GA (szacunkowo)	0.4 NS	0.8 NS
	0.3	0.3

W celu obliczenia pozostałych parametrów modeli, takich jak V_{CT} i I_{CS} zauważamy, że w modelu tranzystora NAP2 przyjęto, że konduktacja diody kolektora spolaryzowanego zaporowo G_{BC} jest stała i równa I_{CS}/V_{CT} .

Daje to jedno równanie wiążące prąd nasycenia diody kolektora i potencjału V_{CT} .

W obszarze nasycenia $I_C/I_{BSat} = 20 \ll h_{FE}$, więc możemy napisać

$$I_{CS} = \frac{1}{1 - \mathcal{L}_R} I_{BSat} \exp \left(\frac{-V_{BEsat} + V_{CEsat}}{V_{CT}} \right), \quad (4.8)$$

$$G_{BC} = \frac{I_{CS}}{V_{CT}} = \frac{1}{1 - \mathcal{L}_R} \frac{h_{re}}{h_{ie}}. \quad (4.9)$$



Z rozwiązania układu równań (4.8) i (4.9) można obliczyć I_{CS} i V_{CT} , a następnie NV oraz NI

	BC 107	BC 177
V_{CT}	55 mV	56 mV
I_{CS}	6E-9	1E-8
$NV = V_{CT}/V_T$	1.63	1.56
$NI = I_{CS}/I_S$	3000.	330

Parametry tranzystora znalezione powyższą metodą stanowią najlepsze dopasowanie modelu w spodziewanym punkcie pracy. Jego dokładność będzie malała w przypadku dużych odchyłeń od znamionowego punktu pracy.

Parametry tranzystorów zebrane w tablicy 4.7 zostały obliczone na podstawie dopasowania zmierzonej i obliczonej odpowiedzi tranzystora dla dużych sygnałów. Zastosowanie tutaj procedury minimalizacji błędu metodą najmniejszych kwadratów dla dużej grupy tranzystorów danego typu jest metodą najlepszą, lecz bardzo czasochłonną. Uproszczona metoda znajdowania parametrów modelu daje w większości przypadków bardzo dobre rezultaty i jest godna polecenia w przypadku, gdy nie znamy obliczanych dokładną metodą parametrów modelu.

Deklaracje powyższych tranzystorów będą postaci:

BC 107/NPN/ AF .9945 VT 30MV IS 2E-12 NI 3E-3 >

NV 1.63 NG 1.2E-4 TF .8NS TR 40NS CE 12PF

CC 5PF GA .3

BC 177/PNP/ AF .993 VT 36MV IS 3E-11 NI 330 NV 1.56 >

NG 4.5E-4 TF 1.6NS TR 80NS CE 24PF CC 10PF GA .3

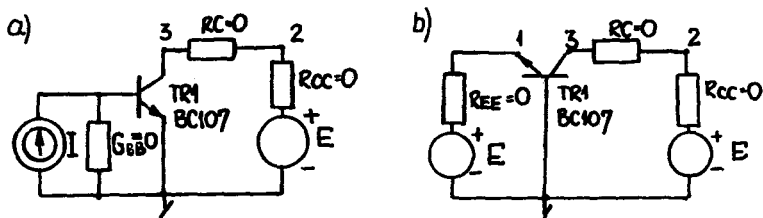
T1 8 9 10 BC107

T2 15 16 18 BC177

Parametry nie podane w liście parametrów FI, AR, GS, i GZ przyjmują automatycznie swoje wartości wbudowane, podane w tablicy 4.5.

W celu sprawdzenia poprawności modelu można przeprowadzić test zachowania się modelu i porównać go z doświadczeniem. Na rysunku 4.23a przedstawiono obwód pomiarowy do otrzymania charakterystyk I_C , V_{CE} tranzystora a na rysunku 4.23b - obwód pomiarowy do wykreślenia zmiany pojemności złącz C_{BE} i C_{BC} w funkcji napięć na złączach tranzystora w kierunku zaporowym.





Rys. 4.23. Układ pomiarowy do: a) otrzymania charakterystyk statycznych I_C , V_{CE} tranzystora, b) wykreślenia zmian pojemności złącz tranzystora C_{BE} i C_{BC} w funkcji napięcia wstecznego

Na rysunku 4.24 przedstawiono zestaw kart wejściowych służących do wykonania powyższych testów, a na rysunku 4.25a - wyniki uzyskane przy pomocy programu NAP2.

Po wpisaniu topologii obwodu z rysunku 4.23a, instrukcja sterująca `*TIME RCC.E 0 1 V` wymusza zmianę napięcia V_{CE} od 0 do 1 V. Napięcie to będzie zmieniało się z automatycznie dobieranym krokiem, gdyż nie została określona wielkość tego kroku w instrukcji `*TIME` (p.4.7.6).

Rozkaz `*DC` oznacza wykonanie analizy stałoprądowej, a podrozkaz `*PLOT (50)` żąda wykreślenia w 50 punktach prądu I_{RC} w funkcji napięcia V_{CE} .

Parametr kontrolny `HOLD` w instrukcji `*RUN` oznacza, że wyniki mają być przechowywane w pamięci do momentu, gdy wystąpi instrukcja `*RUN`. Pozwala to na otrzymanie wydruku kilku przebiegów na jednym wykresie. W tym przypadku będą to przebiegi przy prądzie bazy równym 5 μA , 10 μA , 20 μA i 50 μA , tak jak pokazano na rysunku 4.25b.

Ze względu na automatyczny dobór kroku, punkty na wykresie będą zagęszczone, jeżeli zmiany odpowiedzi będą gwałtowne i rozrzedzone, gdy odpowiedzi będą zmieniać się łagodnie.

Po zmianie tytułu `*`: Będą wykonane obliczenia parametrów h_e dla małych sygnałów. Należy podkreślić, że `*`: musi wystąpić odpowiednio w 1 i 2 kolumnie i jest to jeden z dwóch wyjątków od swobodnoformatowego wprowadzania danych (p. 4.7.12).

Transmitacje `V1/JGBB`, `V1/ERCC`, `IRC/JGBB` oraz `IRC/ERCC` odpowiadają odpowiednio parametrom h_{ie} , h_{re} , h_{ge} oraz h_{oe} .

Należy zwrócić uwagę na różnicę pomiędzy `GBB.J`, co oznacza odwołanie do źródła prądowego bazy i `V1/JGBB`, co oznacza odpowiedź przejściową dla małych sygnałów, nie mającą żadnego związku z rzeczywistymi źródłami niezależnymi.



#LISTING OF :NAP2.DANEPRTRANZ(1/) PRODUCED ON 4JAN83 AT 02.18.04
 #OUTPUT BY LISTFILE IN ':NAP2.NAPIERAL' ON 16AUG84 AT 03.55.10 USING U8
 DOCUMENT DANEPRTRANZ

*CIRCUIT

```

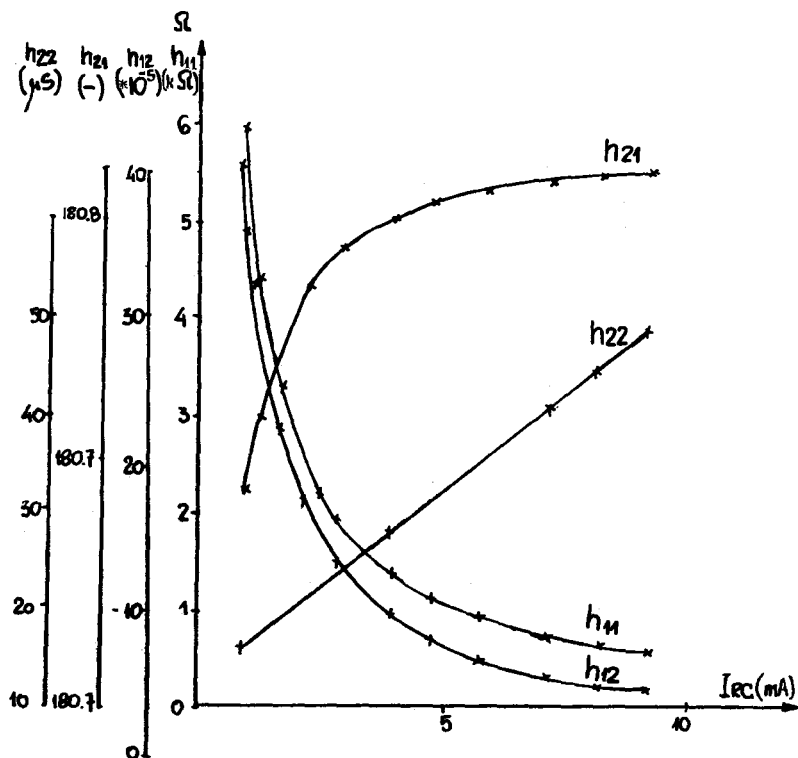
:-----:
:2345678901234567890123456789012345678901234567890123456789012
: PROGRAM TRANZYSTOR - PRZYKŁAD DLA STUDENTÓW
: WERSJA MINI NAPA
: 08 LUTY 1982R.
: A. NAPIERALSKI
: CHARAKTERYSTYKI IC,VCE TRANZYSTORA BC107
: LISTA PARAMETRÓW DLA BC107
BC107/NPN/ AF .9945 VT 30MV IS 2.2E-12 NI 3E3 NV 1.63 >
      NG 1.2E-4 TF .8NS TR .4NS CE 12PF CC 5PF GA .3
: WBUDOWANY MODEL TRANZYSTORA O WEZŁACH: COLL BASE EMIT
TR1 3 1 0 BC107      : ZRÓDŁO PRĄDOWE BAZY
GBB 0 1 0 J 5UA      : ZEROWA REZYSTANCJA KOLEKTORA
RC 2 3 0            : ZRÓDŁO NAPIĘCIOWE VCE
RCC 2 0 0 E 0V      : ZMIANA NAPIĘCIA VCE OD 0 DO 1 VOLTA
*TIME RCC,E 0 1V    : WYKRES IRC W FUNKCJI VCE
*DC *PLOT(50) IRC   : ZACHOWAJ W PAMIĘCI WARTOŚCI WYKRESU
*RUN HOLD           : MODYFIKACJA PRĄDU BAZY
*MODIFY GBB,J 10UA
*RUN HOLD
*MODIFY GBB,J 20UA
*RUN HOLD
*MODIFY GBB,J 50UA
*RUN               : WYDRUK WSZYSTKICH ZATRZYMANÝCH DO TEJ PORY
                   : WYNIKÓW
:
: PARAMETRY HE TRANZYSTORA BC107 W FUNKCJI PRĄDU KOLEKTORA PRZY VCE=1V
*TIME GBB,J 5UA 50UA : ZMIANA PRĄDU BAZY OD 5UA DO 50UA
*DC *PLOT(50 IRC) V1/JGBB/ERCC > : WYDRUK I WYKRES PARAMETRÓW HE W FUN
IRC/JGBB/ERCC       : KCJI IRC
*RUN
: POJEMNOŚCI DYNAMICZNE CBE I CBC TRANZYSTORA BC107
: W FUNKCJI ZMIAN NAPIĘCIA W KIERUNKU ZAPOROWYM
GBB- :WYMAZANIE GBB Z PAMIĘCI MASZYNY
TR1 3 0 1 BC107      : WYMIANA WEZŁÓW TRANZYSTORA TR1
REE 1 0 0 E 0       : WPROWADZENIE ZRÓDŁA
                   : NAPIĘCIOWEGO (DIODY EMITEROWEJ)
*TIME REE,E -.3V 5V > :ZMIANA WARTOŚCI ZRÓDŁA
RCC,E -.3V 5V       :NAPIĘCIOWYCH POLARYZUJĄCYCH DIODY
                   :EMITEROWA I KOLEKTOROWA W KIERUNKU ZAPOROWYM
*DC *PLOT(50) TR1,ECBE > :WYKRES POJEMNOŚCI DYNAMICZNYCH W FUNKCJI
TR1,ECBC           :NAPIĘCIA W KIERUNKU ZAPOROWYM
*RUN
*END

```

Rys. 4.24. Dane do wykonania testów tranzystora BC107



Prąd bazy jest zmienny od $5\mu\text{A}$ do $50\mu\text{A}$, a parametry h_e są wykreślane w funkcji prądu kolektora IRC. Wyniki obliczeń pokazano na rysunku 4.26.



Rys. 4.26. Parametry h_e tranzystora BC107 otrzymane w wyniku testowania modelu przybliżonego

Po modyfikacji układu pomiarowego z przedstawionego na rysunku 4.23a na układ z rysunku 4.23b, poprzez usunięcie gałęzi GBB, zmianę węzłów tranzystora TR1 i wprowadzeniu źródła REE.E, dokonano wykreślenia wartości pojemności złącz tranzystora BC107 w funkcji napięcia wstecznego. Jest ono zmieniane dla obu złącz jednocześnie poprzez jednoczesną zmianę napięć źródeł REE.E i RCC.E. Odwołanie do pojemności tranzystora następuje zgodnie z tablicą 4.6 poprzez podanie przedrostka μ i wpisaniu najpierw nazwy tranzystora np. TR1. μCBE otrzymane wyniki przedstawiono na rysunku 4.27



#LISTING OF :NAP2.WYNIKTRP(1/) PRODUCED ON 20AUG84 AT 14.00.29
#OUTPUT BY LISTFILE IN ':NAP2.A' ON 21AUG84 AT 08.30.43 USING U8
DOCUMENT WYNIKTRP
***** NAP2 VERSION MINI DATE 08/02/82 TIME 17/57/16 *****

:20

*CIRCUIT

:2345678901234567890123456789012345678901234567890123456789012
: PROGRAM TRANZYSTOR - PRZYKŁAD DLA STUDENTOW
: WERSJA MINI NAPA
: 08 LUTY 1982R.

: A. NAPIERAŁSKI
: CHARAKTERYSTYKI IC,VCE TRANZYSTORA BC107
: LISTA PARAMETROW DLA BC107
BC107/NPN/ AF .9945 VT 30MV IS 2.2E-12 NI 3E3 NV 1.63 >
NG 1.2E-4 TF .8NS TR .4NS CE 12PF CC 5PF GA .3
: WBUDOWANY MODEL TRANZYSTORA O WEZŁACH: COLL BASE EMIT

TR1 3 1 0 BC10

GBB 0 1 0 J 5UA

RC 2 3 0

RCC 2 0 0 E OV

*TIME RCC.E 0 1V

*DC *PLOT(50) IRC

*RUN HOLD

INPUT LIST CPU SECONDS

: ZRODŁO PRĄDOWE BAZY
: ZEROWA REZYSTANCJA KOLEKTORA
: ZRODŁO NAPIĘCIOWE VCE
: ZMIANA NAPIĘCIA VCE OD 0 DO 1 VOLTA
: WYKRES IRC W FUNKCJI VCE
: ZACHOWAJ W PAMIĘCI WARTOŚCI WYKRESU
38.00

*MODIFY GBB.J 10UA

*RUN HOLD

INPUT LIST CPU SECONDS

: MODYFIKACJA PRĄDU BAZY
57.00

*MODIFY GBB.J 20UA

*RUN HOLD

INPUT LIST CPU SECONDS

111.00

*MODIFY GBB.J 50UA

*RUN

: WYDRUK WSZYSTKICH ZATRZYMANÝCH DO TEJ PORY



DATE 08/02/82
TIME 17/57/16

1	VN	1/JGBB
2	VN	1/ERCC
3	I RC	/JGBB
4	I RC	/ERCC

1	6.0E 01	-1.3323E-14	1.2000E 03	2.4000E 03	3.6000E 03	4.8000E 03	6.
2	5.0E-06	-5.0000E-05	5.0000E-05	1.5000E-04	2.5000E-04	3.5000E-04	4.5
3	8.0E-04	1.8074E 02	1.8075E 02	1.8077E 02	1.8078E 02	1.8080E 02	1.
4	4.0E-07	1.2000E-05	2.0000E-05	2.8000E-05	3.6000E-05	4.4000E-05	5.

[illegible]

X AXIS



X	I RC				
1	VN	1/JGBB			
2	VN	1/ERCC			
3	I RC	/JGBB			
4	I RC	/ERCC			
X					
	1	2	3	4	
9.127395E-04	5.965006E 03	4.025464E-04	1.807446E 02	1.598870E-05	
9.127476E-04	5.965006E 03	4.025464E-04	1.807446E 02	1.598870E-05	
9.127803E-04	5.964740E 03	4.025284E-04	1.807446E 02	1.598886E-05	
9.128457E-04	5.964313E 03	4.024996E-04	1.807446E 02	1.598912E-05	
9.129763E-04	5.963460E 03	4.024421E-04	1.807446E 02	1.598965E-05	
9.132376E-04	5.961754E 03	4.023269E-04	1.807446E 02	1.599070E-05	
9.137603E-04	5.958344E 03	4.020968E-04	1.807447E 02	1.599279E-05	
9.145899E-04	5.952942E 03	4.017323E-04	1.807447E 02	1.599612E-05	
9.159069E-04	5.944387E 03	4.011549E-04	1.807448E 02	1.600140E-05	
9.1799 4E-04	5.930851E 03	4.002415E-04	1.807450E 02	1.600979E-05	
9.213160E-04	5.909498E 03	3.988005E-04	1.807453E 02	1.602310E-05	
9.265839E-04	5.875913E 03	3.965340E-04	1.807457E 02	1.604423E-05	
9.349462E-04	5.823377E 03	3.929886E-04	1.807463E 02	1.607777E-05	
9.482205E-04	5.741897E 03	3.874899E-04	1.807473E 02	1.613101E-05	
9.669934E-04	5.630438E 03	3.799682E-04	1.807487E 02	1.620632E-05	
9.935425E-04	5.480025E 03	3.698177E-04	1.807506E 02	1.631281E-05	
1.031089E-03	5.280514E 03	3.563537E-04	1.807530E 02	1.646340E-05	
1.080633E-03	5.038521E 03	3.400229E-04	1.807560E 02	1.666205E-05	
1.146007E-03	4.751202E 03	3.206333E-04	1.807596E 02	1.692415E-05	
1.232271E-03	4.418752E 03	2.981980E-04	1.807637E 02	1.726992E-05	
1.340960E-03	4.060514E 03	2.740224E-04	1.807681E 02	1.770584E-05	
1.477902E-03	3.684414E 03	2.486415E-04	1.807727E 02	1.825465E-05	
1.650443E-03	3.299310E 03	2.26528E-04	1.807775E 02	1.894617E-05	
1.860778E-03	2.926431E 03	1.974892E-04	1.807821E 02	1.978909E-05	
2.117186E-03	2.572069E 03	1.735752E-04	1.807864E 02	2.081656E-05	
2.429758E-03	2.241231E 03	1.51248 E-04	1.807905E 02	2.206901E-05	
2.810796E-03	1.937436E 03	1.307472E-04	1.807943E 02	2.359572E-05	
3.290883E-03	1.654825E 03	1.116753E-04	1.807978E 02	2.551917E-05	
3.895766E-03	1.397907E 03	9.433 27E-05	1.808009E 02	2.794256E-05	
4.693928E-03	1.160221E 03	7.829711E-05	1.808039E 02	3.114020E-05	
5.747124E-03	9.476148E 02	6.394946E-05	1.808065E 02	3.535949E-05	
7.078857E-03	7.693497E 02	5.191930E-05	1.808087E 02	4.069458E-05	
8.080141E-03	6.740163E 02	4.548576E-05	1.808099E 02	4.470581E-05	
9.081431E-03	5.997040E 02	4.047082E-05	1.808108E 02	4.871705E-05	

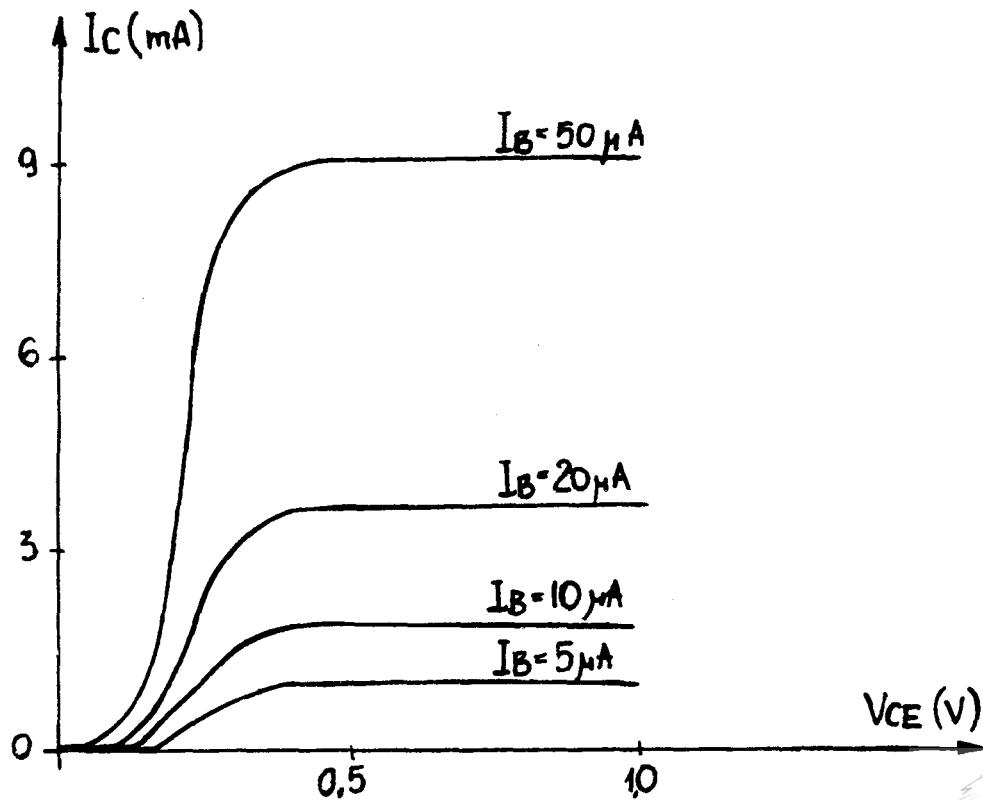
INPUT LIST CPU SECONDS 156.00

```

: POJEMNOSCI DYNAMICZNE CBE I CBC TRANZYSTORA BC107
: W FUNKCJI ZMIAN NAPIECIA W KIERUNKU ZAPOROWYM
GBB-      :WYMAZANIE GBB Z PAMIĘCI MASZYNY
.TR1 3 0 1 BC107      : WYMIANA WEZŁÓW TRANZYSTORA TR1
REE 1 0 0 E 0      : WPROWADZENIE ŹRÓDŁA
      : NAPIECIOWEGO (DIODY EMITEROWEJ)
*TIME REE.E -.3V 5V > :ZMIANA WARTOŚCI ŹRÓDEŁ
RCC.E -.3V 5V      :NAPIECIOWYCH POLARYZUJĄCYCH DIODY
      :EMITEROWA I KOLEKTOROWA W KIERUNKU ZAPOROWYM
*DC *PLOT(50) TR1.£CBE > :WYKRES POJEMNOSCI DYNAMICZNYCH W FUNKCJI
TR1.£CBC      :NAPIECIA W KIERUNKU ZAPOROWYM
* UN

```

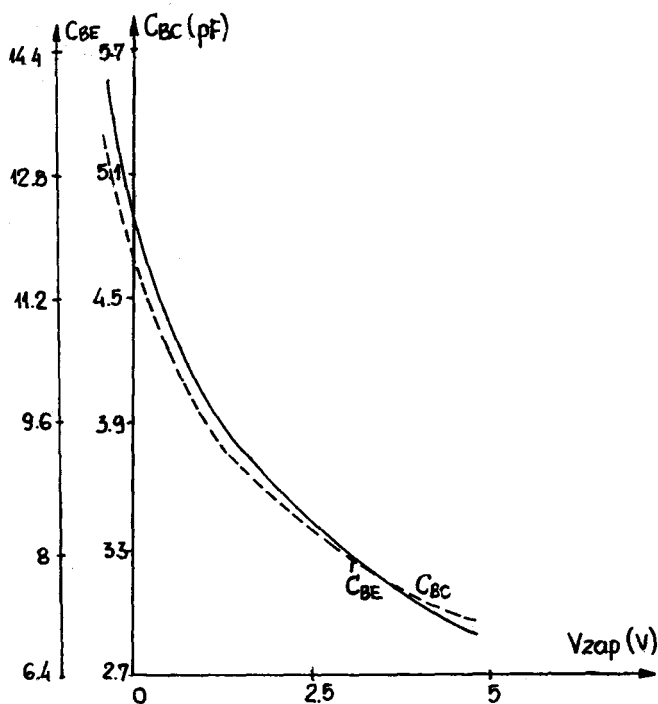




Rys. 4.25b. Testowanie tranzystora BC107 - charakterystyki statyczne



Rezystancje omowe obszarów kolektora, bazy i emitera można włączyć do istniejących elementów układu lub też do tranzystora. Dokładność modelu tranzystora bipolarnego stosowanego w programie NAP2 jest dla większości zastosowań w pełni zadowalająca.



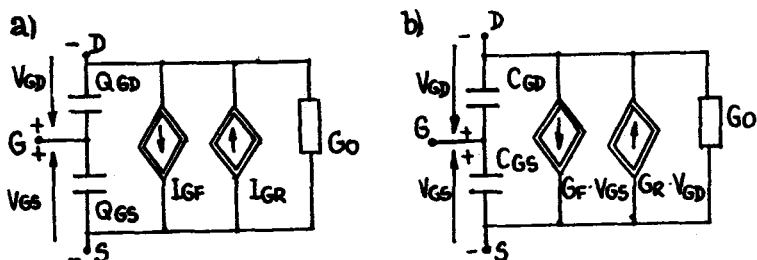
Rys. 4.27. Pojemności złącz tranzystora BC107 otrzymane w wyniku testowania modelu przybliżonego

4.4.3. Tranzystor złączowy JFET

W program NAP2 jest wbudowany model złączowego tranzystora poleowego JFET [42, 61] z kwadratowymi charakterystykami, zarówno dla pracy normalnej jak i inwersyjnej.



Na rysunku 4.28a jest przedstawiony model wielkosygnałowy, a na rysunku 4.28b - model małosygnałowy tranzystora JFET z kanałem typu N.



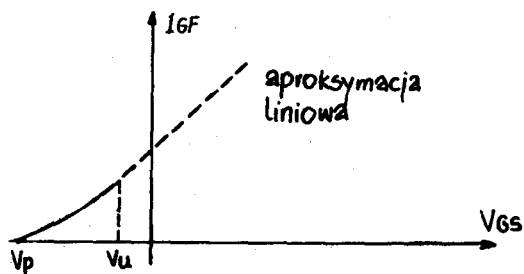
Rys. 4.28 Model tranzystora JFET wbudowany w program NAP2 (NJFET): a) wielkosygnałowy, b) małosygnałowy

Efekty gromadzenia ładunku są modelowane nieliniowymi pojemnościami warstwy zubożonej dla każdego złącza bramki. Konduktancja wyjściowa tranzystora JFET jest proporcjonalna do jego transkonduktancji.

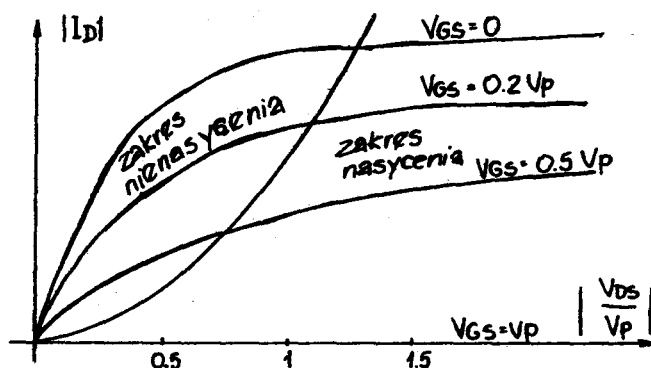
Poniżej przedstawiono równanie opisujące wielko- i małosygnałowy model tranzystora.

Model wielkosygnałowy	Model małosygnałowy
$V_{GS} \leq V_P \rightarrow I_{GF} = 0$ $V_P < V_{GS} \leq V_u \rightarrow I_{GE} = \beta (V_{GS} - V_P)^2$ $V_{GS} > V_u \rightarrow I_{GF} = \beta (V_u - V_P)^2 + G_F (V_{GS} - V_u)$	$G_F = 0$ $G_F = 2\beta (V_{GS} - V_P)$ $G_F = 2\beta (V_u - V_P)$
$V_{GD} < V_P \rightarrow I_{GR} = 0$ $V_P < V_{GD} < V_u \rightarrow I_{GR} = \beta (V_{GD} - V_P)^2$ $V_{GD} > V_u \rightarrow I_{GR} = \beta (V_u - V_P)^2 + G_R (V_{GD} - V_u)$	$G_R = 0$ $G_R = 2\beta (V_{GD} - V_P)$ $G_R = 2\beta (V_u - V_P)$
$Q_{GS} = C_S \int_0^{V_{GS}} \frac{dV}{(1 - \frac{V}{\varphi})^\gamma}$ $Q_{GD} = C_D \int_0^{V_{GD}} \frac{dV}{(1 - \frac{V}{\varphi})^\gamma}$	$G_O = G_Z + n_G (G_F + G_R)$ $C_{GS} = \frac{C_S}{(1 - \frac{V_{GS}}{\varphi})^\gamma}$ $C_{GD} = \frac{C_D}{(1 - \frac{V_{GD}}{\varphi})^\gamma}$

V_p jest napięciem odcięcia (ujemnym dla tranzystora NJFET), a V_u napięciem granicznym, powyżej którego kwadratowe charakterystyki są aproksymowane liniowo. Charakterystykę przejściową z zaznaczonymi napięciami V_u i V_p dla tranzystora NJFET przedstawiono na rysunku 4.29, a charakterystykę wyjściową na rysunku 4.30.



Rys. 4.29. Charakterystyka przejściowa tranzystora NJFET



Rys. 4.30. Charakterystyki wyjściowe tranzystora JFET

Model tranzystora jest opisany przez 9 parametrów, podanych w tabelicy 4.9.



T a b l i c a 4.9

Symbol we wzorach	Nazwa w NAP2	Opis	Standard programowy
β	BE	Parametr transkonduktancji	1E-3
V_P	VP	Napięcie odcięcia (pinch-off voltage)	-3
V_U	VU	Górne ograniczenie charakterystyki	0
C_S	CS	Pojemność bramka-źródło przy zerowej polaryzacji zewnętrznej	0
C_d	CD	Pojemność bramka-dren przy zerowej polaryzacji zewnętrznej	0
φ	FI	Potencjał złącza bramki	1
δ	GA	Wykładnik w równaniu pojemności	0.5
G_Z	GZ	Konduktancja zerowa dren-źródło przy zerowej polaryzacji zewnętrznej	0
n_G	NG	Współczynnik proporcjonalności dla konduktancji wyjściowej	0

Zależności temperaturowe tranzystora nie są uwzględnione w modelu. W przypadku, gdy jest znana przewidywana temperatura pracy można tak zmodyfikować wartości parametrów tranzystora, ażeby model był słuszny w założonej temperaturze.

Instrukcja odwołania do tranzystora JFET jest postaci

T_{xxx}	nr węzła d	nr węzła g	nr węzła s	nazwa listy
-----------	------------	------------	------------	-------------

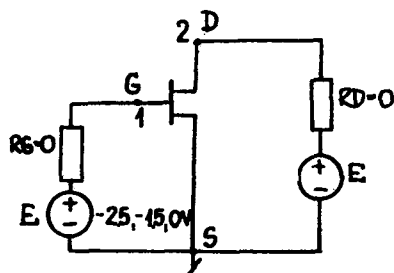
gdzie nazwa listy jest odwołaniem do uprzednio określonej listy parametrów, która dla tranzystora JFET jest postaci

nazwa listy	$\left\{ \begin{array}{l} \text{NJFET} \\ \text{PJFET} \end{array} \right\}$	BE wartość	VP wartość	VU wartość	>
CS wartość	CD wartość	FI wartość	GA wartość	GZ wartość	>
NG wartość					

gdzie: NJFET - tranzystor polowy złączowy JFET z kanałem typu n,
PJFET - tranzystor polowy złączowy JFET z kanałem typu p.



W celu wykreślenia charakterystyk wyjściowych tranzystora można posłużyć się układem z rysunku 4.31. Po znalezieniu parametrów opisujących model tranzystora BF 245, program służący do wykreślenia jego



Rys. 4.31. Układ pomiarowy do otrzymania charakterystyk wyjściowych tranzystora JFET

Charakterystyki jest przedstawiony na rysunku 4.32. Po wczytaniu topologii obwodu wprowadzono parametr JD w celu odwrócenia kierunku prądu IRD, który jest ujemny ze względu na to, że opornik RD jest zadeklarowany razem ze źródłem. Ten sam efekt można osiągnąć wprowadzając dodatkowy zerowy opornik, tak jak w przykładzie 4.23 i odwołując się do jego prądu.

Instrukcja *TIME zmienia napięcie V_{DS} od 0 do 10 V. Wartość napięcia V_{GS} zadeklarowana początkowo na -2.5 V jest kolejno zmieniana instrukcją *MODIFY na -1.5 V i na 0 V. Otrzymane w ten sposób charakterystyki wyjściowe tranzystora przedstawiono na rysunku 4.33

*CIRCUIT

```
BF245/NJFET/ BE 5.212E-4 VP -4.2V VU -1.05V CS 5.2PF CD 4.5PF >
FI 0.644V GA 0.2178 GZ 1.85E-5 NG 9.96E-3
```

```
TJF1 2 1 0 BF245
```

```
RG 1 0 0 E -2.5V
```

```
RD 2 0 0 E 0V
```

```
JD=-1*IRD
```

```
*TIME RD.E 0 10V
```

```
*DC *PLOT(50) JD
```

```
*RUN HOLD
```

```
*MODIFY RG.E -1.5V
```

```
*RUN HOLD
```

```
*MODIFY RG.E 0V
```

```
*RUN
```

```
*END
```

```
****
```

Rys. 4.32. Dane do numerycznego wyznaczania charakterystyk wyjściowych tranzystora NJFET BF 245



NAP2 DC

DATE 16/08/84
TIME 16/35/20

X RD E
1 JD

X	1
0.000000E 00	9.983125E-12
1.000000E-05	1.809228E-08
2.000000E-05	3.617431E-08
4.000000E-05	7.233888E-08
8.000000E-05	1.446662E-07
1.434960E-04	2.594726E-07
2.442897E-04	4.417109E-07
4.04289 E-04	7.309725E-07
6.582739E-04	1.190092E-06
1.061449E-03	1.918760E-06
1. 01449E-03	3.075096E-06
2.71 385E-03	4.909781E-06
4.330084E-03	7.619928E-06
6.890084E-03	1.243388E-05
1.095383E-02	1.974390E-05
1.740463E-02	3.131212E-05
2.764463E-02	4.958557E-05
4.389961E-02	7.836616E-05
6.970280E-02	1.234812E-04
1.106628E-01	1.936575E-04
1.756827E-01	3.014280E-04
2.788955E-01	4.633630E-04
4.427355E-01	6.973871E-04
7.028153E-01	1.010841E-03
1.115666E 00	1.362176E-03
1.771026E 00	1.554661E-03
1. 93093E 00	1.555264E-03
1.814231E 00	1.555842E-03
1.856506E 00	1.556997E-03
1.941058E 00	1.559307E-03
2.110161E 00	1.563928E-03
2.448367E 00	1.573170E-03
3.124780E 00	1.591653E-03
4.477606E 00	1.628618E-03
7.183257E 00	1.702550E-03
1.000000E 01	1.779518E-03



NAP2 DC

DATE 16/08/84
TIME 16/35/20

X RD E
2 JD

X	2
0.000000E 00	6.011192E-12
1.000000E-05	2.861613E-08
5.000000E-05	1.430562E-07
1.300000E-04	3.719332E-07
2.569921E-04	7.352304E-07
4.585795E-04	1.311896E-06
7.785795E-04	2.227213E-06
1.286548E-03	3.679963E-06
2.092897E-03	5.985506E-06
3.372897E-03	9.643916E-06
5.404771E-03	1.544775E-05
8.630168E-03	2.465182E-05
1.375017E-02	3.923988E-05
2.187766E-02	6.234032E-05
3.477925E-02	9.886715E-05
5.302486E-02	1.502248E-04
8.198795E-02	2.310301E-04
1.279640E-01	3.574870E-04
2.009464E-01	5.536548E-04
3.167988E-01	8.535358E-04
5.007030E-01	1.300549E-03
7.926327E-01	1.937018E-03
1.256042E 00	2.763095E-03
1.991659E 00	3.610122E-03
2.112293E 00	3.694654E-03
2.303789E 00	3.797377E-03
2.60 768E 00	3.881157E-03
2.669318E 00	3.886278E-03
2.767021E 00	3.889521E-03
2.870300E 00	3.892879E-03
3.076857E 00	3.899595E-03
3.489971E 00	3.913028E-03
4.316199E 00	3.939894E-03
5.968655E 00	3.993625E-03
7.984327E 00	4.059167E-03
1.000000E 01	4.124709E-03
1.000000E 01	4.124709E-03

NAP2 DC

DATE 16/08/84
TIME 16/35/20

X RD E
3 JD

X	3
0.000000E 00	0.000000E 00
1.000000E-05	3.334764E-08
5.000000E-05	1.667927E-07
1.300000E-04	4.342280E-07
2.900000E-04	9.686621E-07
6.100000E-04	2.037531E-06
1.250000E-03	4.175269E-06
2.530000E-03	8.450744E-06
5.090000E-03	1.700169E-05
1.021000E-02	3.410360E-05
2.045000E-02	6.830740E-05
4.093000E-02	1.367150E-04
8.189000E-02	2.735302E-04
1.638100E-01	5.471606E-04
3.276500E-01	1.094421E-03
6.553300E-01	2.188943E-03
7.980367E-01	2.665614E-03
1.083450E 00	3.618185E-03
1.157273E 00	3.858897E-03
1.238996E 00	4.118680E-03
1.346830E 00	4.450706E-03
1.518007E 00	4.952623E-03
1.789733E 00	5.686017E-03
2.221071E 00	6.690575E-03
2.905778E 00	7.882963E-03
3.992682E 00	8.762157E-03
4.056429E 00	8.775111E-03
4.157620E 00	8.786887E-03
4.318252E 00	8.793383E-03
4.472787E 00	8.799611E-03
4.781858E 00	8.812067E-03
5.399999E 00	8.836980E-03
6.636281E 00	8.886806E-03
8.318140E 00	8.954590E-03
1.000000E 01	9.022373E-03



INPUT LIST CPU SECONDS 44.00

*END



4.4.4. Tranzystor MOSFET

W przypadku, gdy nie zależy nam na dużej dokładności symulacji, można zastosować wbudowany model tranzystora JFET do modelowania właściwości tranzystora MOS (tranzystor polowy z izolowaną bramką typu MOS = metal-oxide-semiconductor). W zależności od typu tranzystora musi być właściwie zadeklarowana wartość napięcia odcięcia V_p i ograniczenia górnego V_u . Efekty gromadzenia ładunku są najlepiej modelowane przez pojemności liniowe; dla tranzystora MOSFET wartość δ powinna być zadeklarowana jako równa zero.

Jako przykład rozpatrzmy fragment układu, w którym dwa tranzystory MOS są modelowane za pomocą modelu tranzystora JFET z kanałem typu n, o odpowiednio dobranych wartościach parametrów z listy parametrów NMOS:

NMOS/NJFET/	VP	2V	VU	6V	CS	5PF	CD	2PF	GA	0
TM1	8	9	10	NMOS						
TM2	15	16	18	NMOS						

W przypadku, gdy nie wystarcza nam prosty model wbudowany w program, lecz chcemy użyć pełnego nieliniowego modelu tranzystora MOS, można odwołać się do procedury MOS i MOSCAP, zgodnie z opisem przedstawionym w p. 4.3.7.

Do opisu charakterystyki tranzystora MOS jest stosowane w tej procedurze następujące wyrażenie [24]:

$$I_D = \frac{\beta}{1 + \beta(V_{GS} - V_T)} \frac{(V_{GS} - V_T)^2}{2} \frac{1}{1 - K_M \sqrt{V_{DS}}} \tanh x \times \left(\mathcal{L} \frac{V_{DS}}{V_{GS} - V_T} \right), \quad (4.10)$$

gdzie: $\mathcal{L} = 2$ dla tranzystora z kanałem typu p,

$\mathcal{L} = 3$ dla tranzystora z kanałem typu n.

W celu symulacji właściwości dynamicznych tranzystora MOS/FET wprowadzono trzy nieliniowe pojemności C_{GS} , C_{GB} , i C_{GD} będące odpowiednio funkcjami napięć V_{GS} i V_{DS} . Obliczane są one poprzez wywołanie procedury MOSCAP [23, 24]. Nie wchodząc w dalsze szczegóły opisu modelu (patrz [24]), rozpatrzmy jako przykład wykreślania charakterystyk statycznych tranzystora MOS z kanałem typu N, wchodzącego w skład układu MC1400 7UB [24, 55].



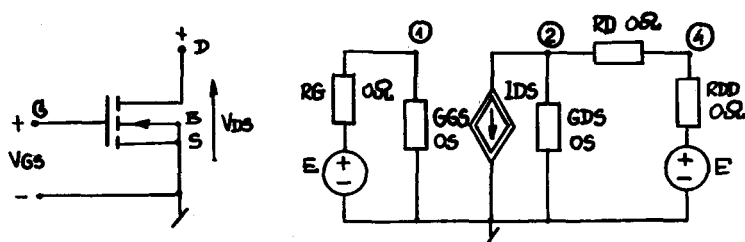
Wyniki identyfikacji parametrów stałoprądowego modelu tranzystora MOS dla tego układu przedstawiono w tablicy 4.11.

T a b l i c a 4.11

Kanał p	Kanał n	Jednostka
$\beta = 1.04$	$\beta = 1.57$	mA/V^2
$V_T = -0.8$	$V_T = 1.5$	V
$\Theta = 0.116$	$\Theta = 0.073$	V^{-1}
$K_M = 0.076$	$K_M = 0.0$	$\text{V}^{-1/2}$

Wartości tych parametrów otrzymano poprzez użycie procedury optymalizacyjnej [24, 23], która wyznacza parametry modelu w ten sposób, ażeby najlepiej symulował układ rzeczywisty.

Układ pomiarowy do otrzymania charakterystyk statycznych tranzystora MOS z kanałem typu n przedstawiono na rysunku 4.34.



Rys. 4.34 Układ pomiarowy do otrzymania charakterystyk wyjściowych tranzystora MOS z kanałem typu n

Program służący do wykreślenia charakterystyk statycznych tego tranzystora przedstawiono na rysunku 4.35. Po wczytaniu topologii obwodu i wprowadzeniu procedury MOS poprzez funkcję SUB instrukcja sterująca * TIME żąda zmiany napięcia RDD.E od -10 do 20 V w celu otrzymania obu ćwiartek charakterystyk. Wyniki symulacji komputerowej przedstawiono na rysunku 4.36.

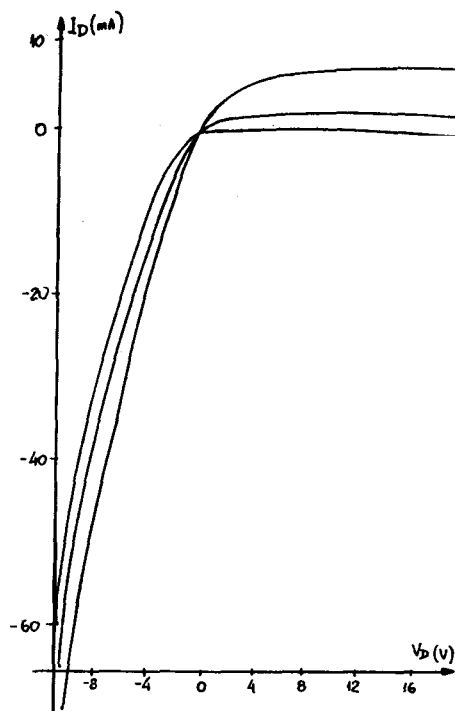


```

*CIRCUIT
RG 1 0 0 E 2V          : GATE BATTERY
GGS 1 0 0              :GATE SOURCE VOLTAGE
RD 4 2 0              :DRAIN LOAD RESISTOR
RDD 4 0 0 E -10V       : DRAIN BATTERY
GDS 2 0 0              :DRAIN-SOURCE VOLTAGE
MOS/SUB/ 3 0 0 1*VGGS 0 1.16M 1.4 0.116 0.075 0 0 0
IDS 2 0 1*MOS(VDS)
*TIME RDD.E -10 20V
*DC VNALL *PLOT(40 VGDS) IRD *PLOT(40 VGDS -3 3) IRD -5M 5M
*RUN HOLD
*MODIFY RG.E =3V
*RUN HOLD
*MODIFY RG.E=5V
*RUN
*END
****

```

Rys. 4.35. Dane do numerycznego wyznaczania charakterystyk wyjściowych tranzystora MOS



Rys. 4.36. Obliczone numerycznie charakterystyki wyjściowe tranzystora MOS



4.4.5. Bramki TTL

Do symulacji właściwości bramek TTL zostały wbudowane w program NAP2 procedury FINP, FLOG, FCAP1 i FCAP2, do których można odwołać się zgodnie z opisem podanym w p. 4.3.7. Dokładny opis makromodeli bramek TTL wykracza poza ramy niniejszego skryptu i można go znaleźć w pracy [49].

4.4.6. Bramki ECL

Do symulacji właściwości bramek ECL zostały wbudowane procedury FLOG (wykorzystywana również do modelowania bramek TTL) oraz FINPE, do której można się odwołać zgodnie z opisem podanym w p. 4.3.7. Dokładny opis makromodeli bramek ECL wykracza poza ramy niniejszego skryptu i można znaleźć w pracy [49].

4.4.7. Wzmacniacz operacyjny

Program NAP2 nie posiada wbudowanego modelu wzmacniacza operacyjnego. W związku z bardzo szerokim zastosowaniem tego typu przyrządu został opracowany zestaw makromodeli [48, 51, 64], które pozwalają na szybką i dokładną analizę układów zawierających wzmacniacze operacyjne.

W celu maksymalnego wykorzystania możliwości programu należy uwzględnić wszystkie jego ograniczenia, takie jak np. maksymalna liczba elementów, węzłów, funkcji, sterowanych i nieliniowych źródeł itd. Ze względu na te ograniczenia opracowano 6 makromodeli, których właściwości i ograniczenia przedstawiono w tabelicy 4.12 [51].

T a b l i c a 4.12

Kryterium porównawcze	Makromodel					
	M60A	M50A	M40A	M30A	M20A	M10A
Liczba elementów	16	16	13	9	9	7
Liczba stopni	3	3	3	2	2	3
Liczba węzłów	6	6	6	4	4	4
Liczba źródeł sterowanych	5	5	4	3	3	2
Liczba źródeł nieliniowych	3	3	0	2	2	0
Maksymalna liczba makromodeli w jednym programie	2	5	5	3	6	12

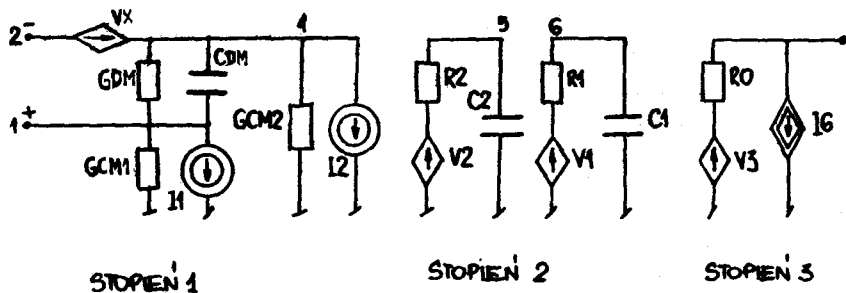


T a b l i c a 4.12(cd.)

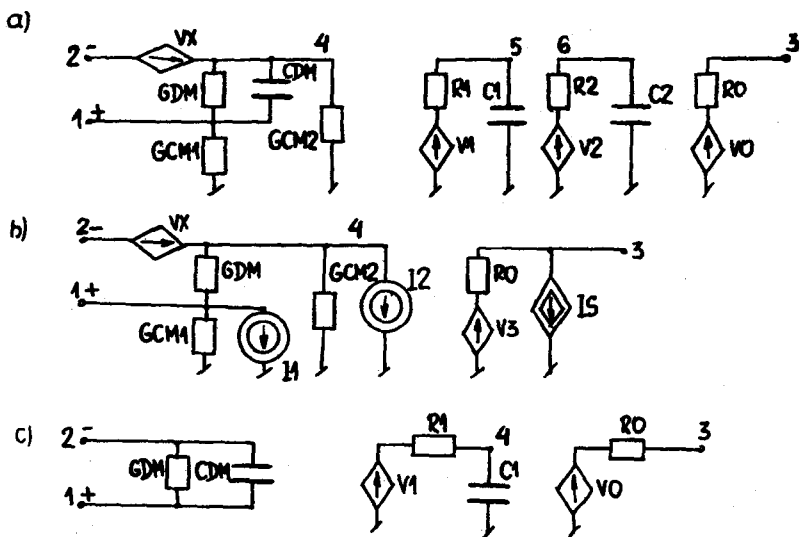
Kryterium porównawcze	Makromodel					
	M60A	M50A	M40A	M30A	M20A	M10A
<u>Symulowanie własności</u>						
Wzmocnienie z otwartą pętlą ADM	+	+	+	+	+	+
Współczynnik tłumienia sygnału współbieżnego CMRR	+	+	+	+	+	-
Wejściowe napięcie niezrównoważenia V _{OFS}	+	+	-	+	+	-
Dryft temperaturowy i napięciowy wejściowego napięcia niezrównoważenia	+	+	-	+	-	-
Wejściowe prądy polaryzujące	+	+	-	+	+	-
Dryft temperaturowy i napięciowy wejściowych prądów polaryzujących	+	-	-	+	-	-
Wejściowy prąd niezrównoważenia	+	+	-	+	+	-
Dryft temperaturowy wejściowego prądu niezrównoważenia	+	-	-	+	-	-
Wejściowa rezystancja różnicowa	+	+	+	+	+	+
Wejściowa pojemność różnicowa	+	+	+	-	-	+
Rezystancje dla sygnału współbieżnego	+	+	+	+	+	-
Rezystancja wyjściowa	+	+	+	+	+	+
Charakterystyka częstotliwościowa: 1 - jedno, 2 - biegunowa	2	2	2	-	-	1
Nasycenie stopnia wejściowego	+	+	-	-	-	-
Ograniczenie szybkości zmian na- pięcia wyjściowego (slew rate)	+	+	-	-	-	-
Ograniczenie napięcia wyjściowego	+	+	-	+	+	-
Ograniczenie prądu wyjściowego	+	+	-	+	+	-

Nie wchodząc w szczegóły opisu i budowy makromodeli (można je znaleźć w pracach [46, 48, 51]) na rysunku 4.37 przedstawiono schemat makromodeli M60A i M50A, a na rysunku 4.38a,b,c - schematy makromodeli M40A, M30A, M20A i M10A. Topologia makromodeli M60A i M50A oraz M30A i M20A jest identyczna, lecz różnią się one uwzględnieniem (M60A i M30A) lub też pominięciem (M50A i M20A) wpływu temperatury i napięcia zasilania na parametry makromodelu.





Rys. 4.37. Schemat makromodelu wzmacniacza operacyjnego M60 A i M50A



Rys. 4.38. Modele wzmacniaczy operacyjnych: a) makromodel M40A do analizy w dziedzinie częstotliwości b) makromodel M30A do analizy stało-prądowej, c) podstawowy makromodel M10A do analizy układów zawierających dużą ilość wzmacniaczy operacyjnych

Na rysunkach 4.39 do 4.44 przedstawiono opis poszczególnych makromodeli przy pomocy instrukcji programu NAP2 dla wzmacniacza μA 741. W tablicy 4.13 podano parametry tego wzmacniacza.



```

: NONLINEAR LARGE SIGNAL M60A MACROMODEL
: *****
SUPP=15
PTV/ / -25
DTEMP =1*PTV(TEMP)
DVSUPP=0.6*PTV(1.6666667*SUPP)
:STAGE1
: *****
GDM 1 4 .5E-6
CDM 1 4 1.4PF
YB1=110NA
YB2=90NA
ALI1=-1.33N
ALI2=-1.18N
BETAI=.58N
FI1/ / 1*ALI1*DTEMP 1*BETAI*DVSUPP
FI2/ / 1*ALI2*DTEMP 1*BETAI*DVSUPP
GCM1 1 0 .5E-9 J 1*FI1(YB1)
GCM2 4 0 .5E-9 J 1*FI2(YB2)
VOFS=.3MV
ALVOF=40U
BETAUF=2U
FVO/ / 1*ALVOF*DTEMP 1*BETAUF*DVSUPP
VOF=1*FVO(VOFS)
CMMR=3.16E5
AC=.5/CMMR
FX/ / B 1*AC 1*AC*VGCM1 1*VOF
VX 4 2 1*FX(VGCM2)
:STAGE2
: *****
ADM=2E5
P1=9000V
P2=7000V
FAS/TAB2/ -1E70 -1*P2 -1*P2 -1*P2 1*P1 1*P1 1E70 1*P1
R2 5 0 1K
C2 5 0 79.6PF
V2 R2 1*ADM VGDM
R1 6 0 119.36K
C1 6 0 133UF
V1 R1 1*FAS(VC2)
:STAGE3
: *****
PZ1=0.08*FTV(12.5*SUPP)
PZ2=0.08*FTV(12.5*SUPP)
FE3/TAB2/ -1E70 -1*PZ2 -1*PZ2 -1*PZ2 1*PZ1 1*PZ1 1E70 1*PZ1
R0 3 0 75
V3 R0 1*FE3(VC1)
PIS=25MA
FIS/TAB2/ -1000*PIS -999*PIS -1*PIS 0 1*PIS 0 1000*PIS 999*PIS
IS 3 0 1*FIS(ISO)

```

Rys. 4.39. Opis makromodelu M60A w języku NAP2



```

:NONLINEAR LARGE SIGNAL M50A MACROMODEL
: STAGE1
GDM 1 4 .5E-6
CDM 1 4 1.4PF
GCM1 1 0 .5E-9 J 110NA
GCM2 4 0 .5E-9 J 90NA
FX/ / B 1.5823E-6 1.5823E-6*VGCM1 .3M
VX 4 2 1*FX(VGCM2)
:STAGE2
FAS/ TAB2/ -1E70 -7000 -7000 -7000 9000 9000 1E70 9000
R2 5 0 1K
C2 5 0 79.6PF
V2 R2 2E5 VGDM
R1 6 0 119.67K
C1 6 0 .133UF
V1 R1 1*FAS(VC2)
:STAGE3
FB3/TAB2/ -1E70 -13 -13 -13 13 13 1E70 13
RO 3 0 75
V3 RO 1*FB3(VC1)
FIS/TAB2/ -25 -24.975 -0.025 0 0.025 0 25 24.975
IS 3 0 1*FIS(IRO)

```

Rys. 4.40. Opis makromodelu M50A w języku NAP2

```

:M40A - MACROMODEL FOR FREQUENCY-DOMAIN ANALYSIS
:STAGE1
GDM 1 4 .5E-6
CDM 1 4 1.4PF
GCM1 1 0 .5E-9
GCM2 4 0 .5E-9
PX/ / B 1.5823E-6 1.5823E-6*VGCM1
VX 4 2 1*PX(VGCM2)
:STAGE2
R2 5 0 1K
C2 5 0 79.6PF
V2 R2 2E5 VGDM
R1 6 0 119.67K
C1 6 0 .133UF
V1 R1 1 VC2
:STAGE3
RO 3 0 75
VO RO 1 VC1

```

Rys. 4.41. Opis makromodelu M40 A w języku NAP2



```

;NONLINEAR D.C M30A MACROMODEL
TEMP=25
SUPP=15
FTV/ / -25
DTEMP=1*FTV(TEMP)
DVSUPP=0.6*FTV(1.6666667*SUPP)
;STAGE1
GDM 1 4 .5E-6
YB1=110NA
YB2=90NA
FI1/ / -1.33E-9*DTEMP .58E-9*DVSUPP
FI2/ / -1.18E-9*DTEMP .58E-9*DVSUPP
GCM1 1 0 .5E-9 J 1*FI1(YB1)
GCM2 4 0 .5E-9 J 1*FI2(YB2)
UOFS=.3MV
FUO/ / 40E-6*DTEMP 2E-6*DVSUPP
UOF=1*FUO(UOFS)
FX/ / B 1.5823E-6 1.5823E-6*VGCM1 1*UOF
VX 4 2 1*FX(VGCM2)
;STAGE2
PZ1=0.08*FTV(12.5*SUPP)
PZ2=0.08*FTV(12.5*SUPP)
FE3/TAB2/ -1E70 -1*PZ2 -1*PZ2 -1*PZ2 1*PZ1 1*PZ1 1E70 1*PZ1
RO 3 0 75
V3 RO 1*FE3(2E5*VGDM)
FIS/TAB2/ -25 -24.975 -0.025 0 0.025 0 25 24.975
IS 3 0 1*FIS(IRO)

```

Rys. 4.42. Opis makromodelu M30A w języku NAP2

```

#LISTING OF :NAP2.M20A(2/) PRODUCED ON 16AUG84 AT 03.47.00
#OUTPUT BY LISTFILE IN ':NAP2.NAPIERAL' ON 16AUG84 AT 03.48.11 USING UB
DOCUMENT M20A
;NONLINEAR D.C. M20A MACROMODEL
;STAGE1
GDM 1 4 .5E-6
GCM1 1 0 .5E-9 J 110NA
GCM2 4 0 .5E-9 J 90NA
FX/ / B 1.5823E-6 1.5823E-6*VGCM1 .3E-3
VX 4 2 1*FX(VGCM2)
;STAGE2
FE3/TAB2/ -1E70 -13 -13 -13 13 13 1E70 13
RO 3 0 75
V3 RO 1*FE3(2E5*VGDM)
FIS/TAB2/ -25 -24.975 -0.025 0 0.025 0 25 24.975
IS 3 0 1*FIS(IRO)

```

Rys. 4.43. Opis makromodelu M30A w języku NAP2



```

:M10A - MACROMODEL FOR CIRCUITS WITH MANY O.A.
:STAGE1
GDM 1 2 .5E-6
CDM 1 2 1.4PF
:STAGE2
R1 4 0 119.67K
V1 R1 2E5 VGDM
C1 4 0 .133UF
RO 3 0 75
VO RO 1 VC1

```

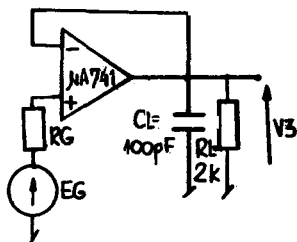
Rys. 4.44. Opis makromodelu M10A w języku NAP2

T a b l i c a 4.13

Opis	Element lub parametr	Wartość dla uA 741
Wzmocnienie z otwartą pętlą	ADM	2.E5
Wejściowa konduktancja różnicowa	GDM	5.E-6
Wejściowa pojemność różnicowa	CDM	1.4 pF
Zastępcze wejściowe	$\left\{ \begin{array}{l} IB1' = I_{B1} + 0.5 I_{OF} \\ IB2' = I_{B2} - 0.5 I_{OF} \end{array} \right.$	$\left\{ \begin{array}{l} IB1' = 110 \text{ nA} \\ IB2' = 90 \text{ nA} \end{array} \right.$
prądy polaryzujące:		
$ALI1 = I_{B1} + \frac{I_{OF}}{2}$	ALI1	-1.33 nA/°C
$ALI2 = I_{B2} - \frac{I_{OF}}{2}$	ALI2	-1,18 nA/°C
Konduktancja dla sygnału } współbieżnego	$\left\{ \begin{array}{l} GCM1 \\ GCM2 \end{array} \right.$	$\left\{ \begin{array}{l} .5 \text{ E-9} \\ .5 \text{ E-9} \end{array} \right.$
Wejściowe napięcie niezrównoważenia	UOFS	.3 mV
Dryft temperaturowy wejściowego napięcia niezrównoważenia	$\mathcal{L} \text{ VOF}$	40 $\mu\text{V}/^{\circ}\text{C}$
Dryft napięciowy wejściowego napięcia niezrównoważenia	$\rho \text{ VOF}$	2 $\mu\text{V}/\text{V}$
Współczynnik tłumienia sygnału współbieżnego	CMRR	3.16 E5
Elementy do symulacji bieguny } niskoczęstotliwościowego	$\left\{ \begin{array}{l} R1 \\ C1 \end{array} \right.$	$\left\{ \begin{array}{l} 119.36 \text{ k}\Omega \\ 133 \text{ }\mu\text{F} \end{array} \right.$
Elementy do symulacji bieguny } wysokoczęstotliwościowego	$\left\{ \begin{array}{l} R2 \\ C2 \end{array} \right.$	$\left\{ \begin{array}{l} 1 \text{ k}\Omega \\ 79.6 \text{ pF} \end{array} \right.$
Parametry graniczne } źródła napięciowego V1	$\left\{ \begin{array}{l} P1 \\ P2 \end{array} \right.$	$\left\{ \begin{array}{l} 9000 \text{ V} \\ 7000 \text{ V} \end{array} \right.$
Parametry graniczne } napięcia wyjściowego	$\left\{ \begin{array}{l} PZ1 \\ PZ2 \end{array} \right.$	$\left\{ \begin{array}{l} 13 \text{ V} \\ 13 \text{ V} \end{array} \right.$
Parametr ograniczający maksymalny prąd wyjściowy	PIS	25 nA
Dryft napięciowy wejściowych prądów polaryzujących I_{B1} i I_{B2}	βI	.58 nA/V

Wartości wszystkich parametrów podanych w tabelicy 4.13 można znaleźć bezpośrednio z katalogu, bądź też łatwo otrzymać przy pomocy standardowych układów pomiarowych [51, 64].

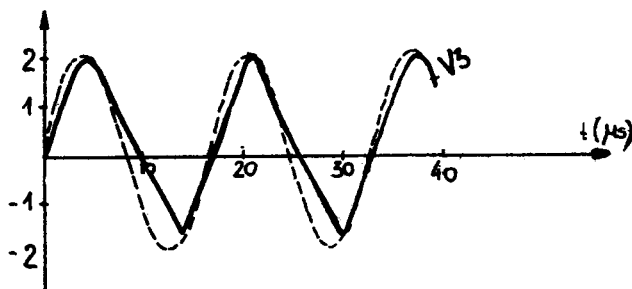
Na rysunku 4.45 przedstawiono układ wtórniaka emiterowego, w którym na wejście podano sygnał o amplitudzie 2V i częstotliwości 60 kHz. Karty programu do analizy tego układu przedstawiono na rysunku 4.46. Makromodel został wywołany z biblioteki 6 (p. 4.7.1), w której są zmagazynowane wszystkie makromodely wzmacniacza operacyjnego.



Rys. 4.45. Układ wtórniaka emiterowego.

```
* CIRCUIT
* LIST 9
QM 1=1 2=2 3=3 0=0
* LIB5 M50A
Q*
R2 3 2 0
RL 3 0 2K
GL 3 0 100P
FG/SIN/ B 2 D 2.65250E-6
RG 1 0 600 E 1* FG (TIME)
* TIME 0 50U
* DCTR * PLOT (100) V1 *Y V3
* RUN STEP .5U MINSTEP .1U
* END
```

Rys. 4.46. Program do analizy układu z rys. 4.45



Rys. 4.47. Wyniki symulacji komputerowej układu z rys. 4.45

Wyniki analizy układu przedstawiono na rysunku 4.47. Jak widać

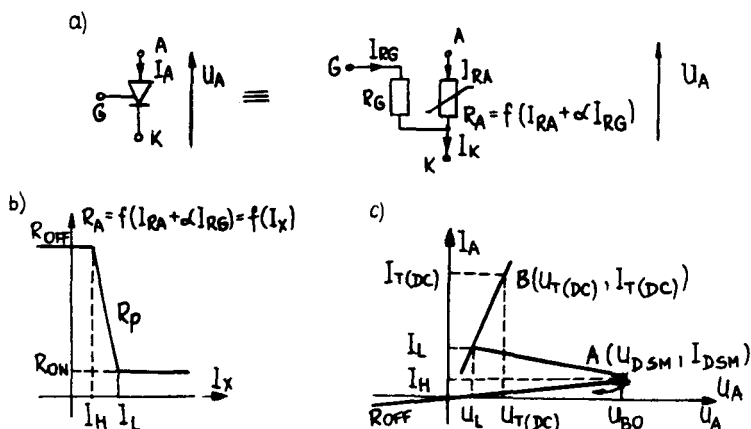


z otrzymanych rezultatów, użyty makromodel M50A pozwala na symulację zjawiska Slew Rate. Wyboru najodpowiedniejszego makromodelu do danego typu analizy można łatwo dokonać na podstawie tablicy 4.12. Dokładny opis makromodeli wzmacniacza operacyjnego można znaleźć w pracach [48, 51, 64].

4.4.8. Tyrystorki

Program NAP2 nie posiada wbudowanego modelu tyrystora. W celu symulacji układów tyrystorowych najwygodniej jest posłużyć się makromodelem tego przyrządu [43, 50, 52, 54].

W przypadku, gdy właściwości dynamiczne przyrządu nie są istotne z punktu widzenia ich wpływu na działanie układu można posłużyć się makromodelem RR, RRE lub też RLE [44].



Rys. 4.48. Model tyrystora: a) makromodel dwurezystancyjny, b) zależność nieliniowej rezystencji R_A od prądu I_x , c) symulowana charakterystyka statyczna

Na rysunku 4.48 przedstawiono makromodel RR tyrystora. Parametry tego makromodelu mogą być wyznaczone bezpośrednio z katalogu na podstawie następujących równań [44]:



- punkt B (rys. 4.48c)

$$R_{ON} = U_{T(DC)} / I_{T(DC)}, \quad (4.11)$$

- punkt A (rys. 4.48c)

$$R_{OFF} = U_{DSM} / I_{DSM}, \quad (4.12)$$

$$R_G = U_{GT} / I_{GT}, \quad (4.13)$$

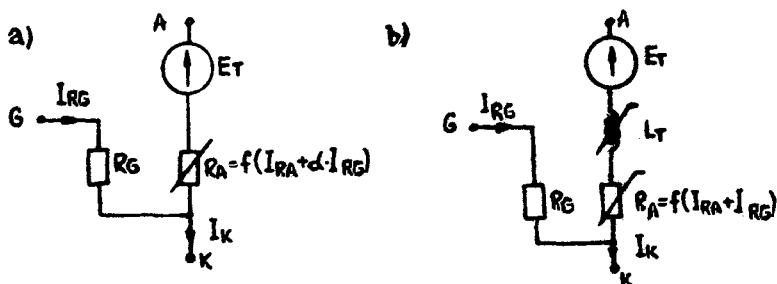
$$\alpha = I_{DSM} / I_{GT}, \quad (4.14)$$

gdzie: R_{ON} - rezystancja tyrystora, gdy jest on załączony (stan przewodzenia),

R_{OFF} - rezystancja tyrystora, gdy jest on wyłączony (stan blokowania lub zaporowy),

α - współczynnik prądu bramki,

R_G - rezystancja zastępcza obwodu bramki.



Rys. 4.49. Makromodel tyrystora: a) typu RRE, b) typu RLE

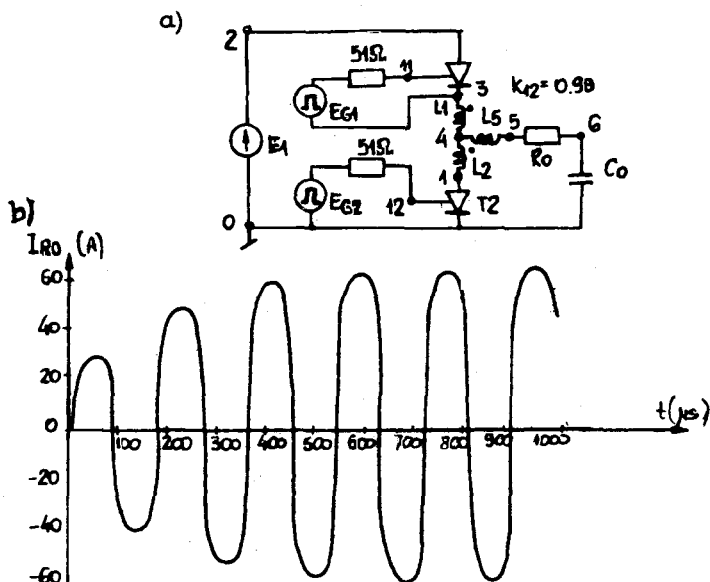
Dwa pozostałe makromodele tyrystora, służące do symulacji jego właściwości statycznych, przedstawiono na rysunku 4.49.

Źródło napięciowe E_T oraz indukcyjność L_T przyjmują wartości zgodnie z poniższymi zależnościami:



$$E_T = \begin{cases} 0 & \text{dla } V_{AK} \leq 0 \\ V_{T0} & \text{dla } V_{AK} > 0 \end{cases} \quad (4.15)$$

$$L_T = \begin{cases} L & \text{dla } V_{AK} > 0 \\ K1.L & \text{dla } V_{AK} \leq 0 \end{cases} \quad (4.16)$$



Rys. 4.50. Układ falownika szeregowego: a) schemat, b) wyniki symulacji komputerowej

Zastosowanie źródła napięciowego E_T pozwala na dokładniejszą symulację charakterystyki tyrystora dla stanu przewodzenia. Różne wartości indukcyjności L_T dla stanu przewodzenia i zaporowego pozwalają na ominięcie pewnych problemów związanych z całkowaniem numerycznym [44, 50].

Na rysunku 4.50a przedstawiono układ falownika szeregowego, a na



rysunku 4.50b przebiegi uzyskane w wyniku symulacji komputerowej przy zastosowaniu makromodelu RR [37, 44]. Dane służące do wykonania tego programu przedstawiono na rysunku 4.51.

```
*LIB2 NEW
*LIB2 TO0804R+
:MODEL DWUREZYSTANCYJNY TYRYSTORA TO0804R
: G=1 A=2 K=0
FRA/TAB2/ -100 50K 10M 50K 10.1M 26M 100 26M
RG 1 3 30
RO 3 0 0
RA 2 3 0.1*FRA(IRO)
>
*CIRCUIT
:-----
: DANE DO FALOWNIKA ZMODYFIKOWANE 26 CZERWIEC 1981R. A.NAPIERALSKI
TYRYSTOR TO0-80-4
: 09.10.1981
: PROGRAM SSFSZR1
: SYMULACJA STARTU FALOWNIKA SZEREGOWEGO PRZY UZICIU
: REZYSTANCYJNEGO MODELU TYRYSTORA
  RZ 20 0 1E-9 E=24V
  RO 5 6 0.126
  CO 6 0 68U
  LS 4 5 1U
  L1 3 4 27U
  L2 4 1 27U
  M12 L1 L2 26.46U : K12=0.98
  FIRG1/TAB2/ P 0.4M 0 1 10U 1 11U 0 0.399M 0 0.4M 1
  FIRG2/TAB2/ P 0.4M 0 0 0.199M 0 0.2M 1 0.21M 1 0.211M 0 0.4M 0
  RZT1 11 3 51 E=10.0*FIRG1(TIME)
  RZT2 12 0 51 E=10.0*FIRG2(TIME)
Q1 20=2 11=1 3=0
*LIB2 TO0804R
Q*
Q2 1=2 12=1 Q=0
*LIB2 TO0804R
Q*
  *TIME 0 100US
  *TR *PLOT(100) V6 V1 IQ1.RA IQ1.RG IQ2.RA IRO
  *RUN MAXSTEP 10U
  *END
:
****
```

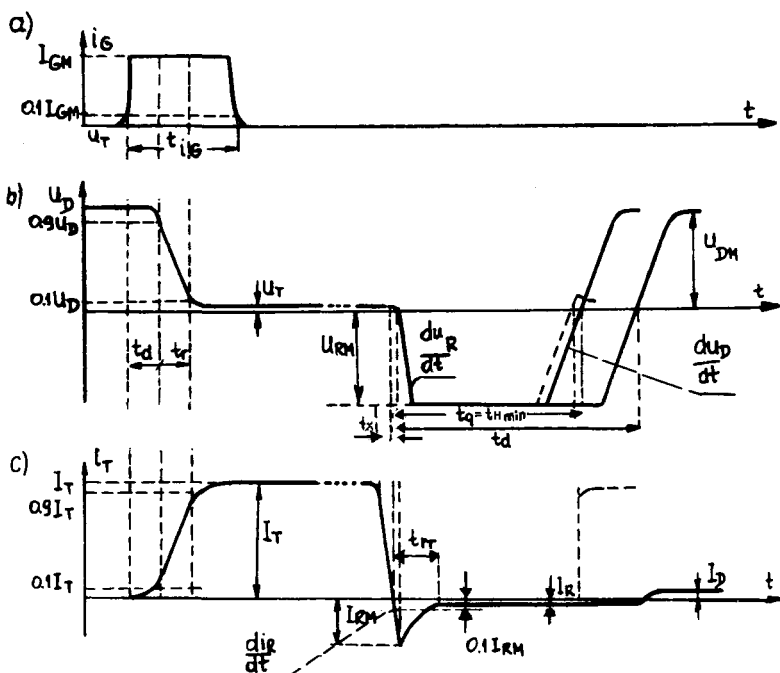
Rys. 4.51. Dane do symulacji startu falownika szeregowego przy pomocy programu NAP2

W przypadku, gdy właściwości dynamiczne przyrządu (patrz rysunek 4.52) muszą być wzięte pod uwagę, należy użyć makromodelu DR (diodowo-



-rezystancyjnego) tyrystora [44, 51]. Struktura tego makromodelu jest przedstawiona na rysunku 4.53. Wszystkie parametry tego modelu mogą być łatwo wyznaczone na podstawie katalogu. R_{OFF} można wyznaczyć bezpośrednio z równania (4.12), a pozostałe parametry z poniższych równań:

$$R_{ON} = \frac{U_{T(DC)} - V_T \ln \frac{I_{T(DC)}}{I_S}}{I_{T(DC)}}, \quad (4.17)$$



Rys. 4.52. Definicja parametrów dynamicznych tyrystora: a) impuls bramki, b) napięcie V_{AK} , c) prąd I_{AIC}

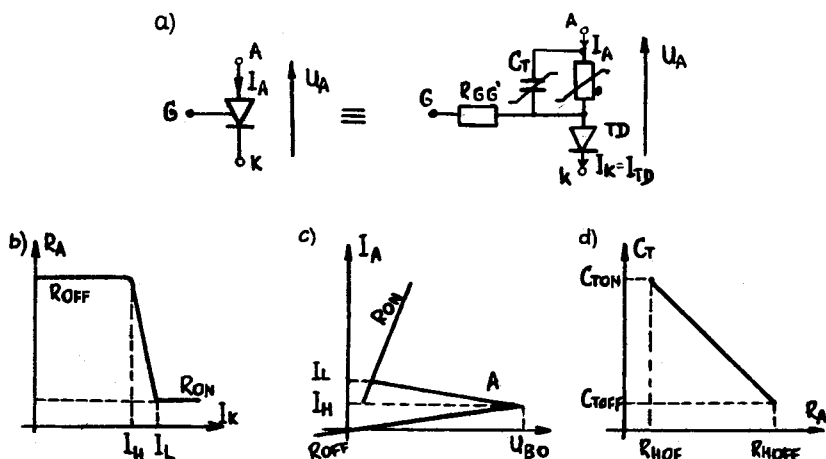


$$R_{GG'} = \frac{U_{T(DC)} - V_T \ln \frac{I_{GT}}{I_S}}{I_{GT}}, \quad (4.18)$$

$$C_{TOFF} = \frac{I_{OSM}}{\frac{dU}{dt} / \text{kryt.}}, \quad (4.19)$$

$$C_{TON} = \frac{t_r}{R_{ON} \ln 9}, \quad (4.20)$$

$$TT \approx t_x \frac{I_{RM} - I_L}{I_{RM} - I_-} \quad (\text{p.rys.4.52}). \quad (4.21)$$



Rys. 4.53. Diodowo-rezystancyjny (DR) makromodel tyrystora: a) schemat, b) zależność nieliniowej rezystancji RA od prądu I_A , c) symulowana charakterystyka statyczna tyrystora, d) zależność pojemności C_T od wartości rezystancji RA gdzie: C_T - pojemność do symulacji czasu załączania i efektu dV/dt . $R_{GG'}$ - zastępcza rezystancja bramki, TT - czas przelotu przez diodę (do symulacji ładunku przejściowego przy wyłączaniu)

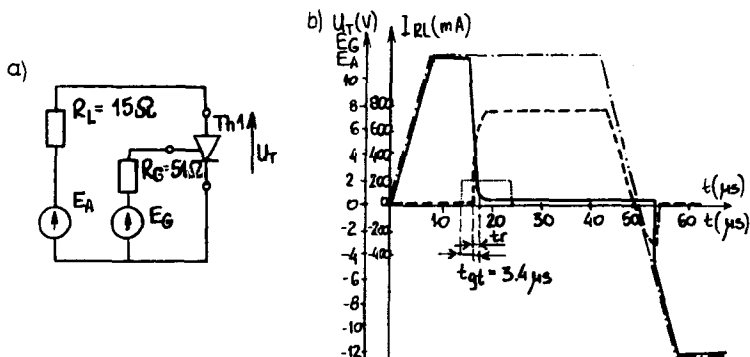


W celu dokładniejszego określenia czasu przelotu T_T , służącego do symulacji ładunku przejściowego przy wyłączaniu tyrystora, zamiast równania (4.21) można posłużyć się programem TTAL, który oblicza właściwą wartość T_T na podstawie danych eksperymentalnych [45].

W tablicy 4.14 podano parametry makromodelu tyrystora DR dla tyrystorów typu BT P2/100 i BT P7/300 [44].

Tablica 4.14

Parametry makromodelu DR	R_{OFF} [k Ω]	R_{ON} [m Ω]	$R_{GG'}$ [Ω]	C_{TOFF} [pF]	C_{TON} [μ F]	T_T [μ s]	T_S [nA]	V_T [mV]
BTP2/100	80	620	11	30	2.2	5.2	100	45
BTP7/300	51	57	13	70	6.1	4.2	100	45



Rys. 4.54. Analiza stanów przejściowych tyrystora: a) układ pomiarowy, b) wyniki symulacji komputerowej

Na rysunku 4.54a przedstawiono układ pomiarowy do testowania poprawności symulacji właściwości dynamicznych tyrystora przez makromodel DR. Wyniki symulacji komputerowej przedstawiono na rysunku 4.54b. Potwierdzają one w pełni przydatność tego makromodelu do symulacji dynamicznych właściwości tyrystora. Opis ładowania makromodelu do biblioteki 2 przedstawiono na rysunku 4.55.



Należy podkreślić, że wykonywanie obliczeń - z automatycznie dobieranym krokiem - może prowadzić w przypadku analizy układu tyrystorowego (z użyciem dowolnego z zaprezentowanych makromodeli) do niezauważenia impulsu bramkowego, gdy krok całkowania numerycznego zostanie powiększony na tyle, że "przeskoczy" impuls bramki. Najprostszym

```
*LIB2 T00804D+
:MODEL DIODOWOREZYSTANCYJNY TYRYSTORA T00804D
: G=1 A=2 K=0
      RGG 1 3 23.8
FCT/TAB2/ 8M 70U 50K 33.3P
FRT/TAB2/ -100 50K 10M 50K 10.1M 8M 100 8M
D1/DIODE/ IS 100M VT 45M TT 10U GS 100
TD1 3 0 D1
RT 2 3 1*FRT(ITD1)
CT 2 3 1*FCT(RT)
>
```

Rys. 4.55. Opis ładowania makromodelu diodowo-rezystancyjnego do biblioteki

wyjściem z tego typu kłopotu jest zadeklarowanie parametru MAXSTEP (maksymalna długość kroku) nie większego niż nominalna szerokość impulsu bramki [44, 45].

4.5. Modele deklarowane przez użytkownika

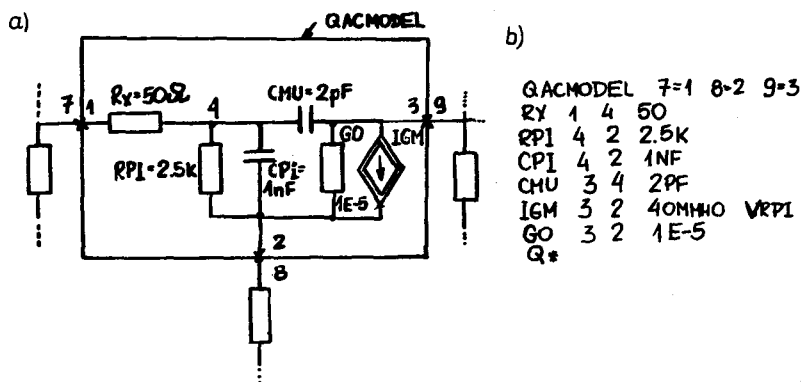
W programie NAP2 istnieje możliwość deklarowania własnych modeli przez użytkownika w celu ich późniejszego wielokrotnego wykorzystywania. Dany model jest opisany przez zbiór instrukcji programu NAP2 i stanowi pewien układ elektroniczny. W celu umieszczenia tego modelu w analizowanym obwodzie należy pewne jego węzły połączyć z pewnymi węzłami obwodu głównego. Te węzły będziemy w dalszym ciągu nazywać węzłami zewnętrznymi (modelu lub obwodu), w odróżnieniu od pozostałych węzłów, które nie będą podlegały połączeniu i będą zwane węzłami wewnętrznymi.

Deklaracja modelu użytkownika jest postaci

```
Qxxx nr węzła zew. obwodu = nr węzła zew. modelu ...
zdania opisujące model użytkownika
Q*
```



Na rysunku 4.56a przedstawiono małosygnalowy model tranzystora typu hybryd-phi. Węzły zewnętrzne wspólne dla modelu i obwodu oznaczone są gniazdkami. W związku z tym, że węzły te należą zarówno do modelu, jak o do obwodu, to mają one podwójne numery. Sposób deklaracji tego modelu przedstawiono na rys. 4.56b. Węzły zewnętrzne 1, 2 i 3 tego modelu są dołączone do węzłów 7, 8 i 9 obwodu głównego. Karta Q* oznacza zakończenie wpisywania modelu użytkownika i musi być bezwzględnie użyta.



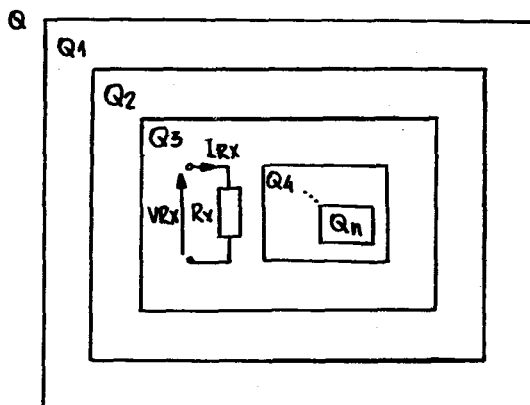
Rys. 4.56. Model użytkownika: a) QACMODEL hybryd-phi tranzystora, b) deklaracja tego modelu

Należy podkreślić, że istnieje możliwość umieszczania modeli wewnątrz innych modeli tak, jak to przedstawiono na rysunku 4.57. Symbol Q oznacza obwód główny, do którego można się odwołać z wnętrza modelu, i który może być interpretowany jako najbardziej zewnętrzny model o nazwie Q.

W celu odwołania się do parametru modelu będącego częścią innych modeli należy podać kolejne nazwy począwszy od modelu najbardziej zewnętrznego. Na przykład w celu odwołania się do wartości elementu RX wchodzącego w skład modelu Q₃ z rysunku 4.57 należy podać instrukcję postaci

Q1.Q2.Q3.RX





Rys. 4.57. Struktura modeli w programie NAP2

Odwołanie do napięcia i prądu tego elementu będzie natomiast postaci

VQ1.Q2.Q3.RX

IQ1.Q2.Q3.RX

Odwołanie z wnętrza dowolnego modelu do obwodu głównego następuje przez ustawienie litery Q przed nazwą

Q. RLOAD

VQ. RLOAD

IQ. RLOAD

4.6. Instrukcje wymiany i usuwania elementów obwodu

W celu ułatwienia użytkownikowi korzystania z programu istnieje możliwość powtórzenia danej analizy bez konieczności ponownego wprowadzenia całego pliku danych wejściowych ze zmodyfikowaną (lub nie) strukturą układu i wartościami jego elementów (p. 4.1.3).



Zdanie opisu obwodu można wymienić poprzez wstawienie kropki przed nazwą i podanie nowego opisu

• nazwa	nowy opis
---------	-----------

gdzie: nazwa - jest nazwą, od której zaczyna się zmienione zdanie

Przykład

```
RLOAD 25 0 1K
:
.RLOAD 26 30 10K
```

Po wykonaniu pewnych obliczeń z opornikiem RLOAD włączonym pomiędzy węzeł 25 a masę, o wartości 1K, został on przełączony pomiędzy węzeł 26 a 30 i jego wartość została zmieniona na 10K. Jeżeli chcemy w ogóle usunąć dany element z układu, to należy podać znak minus (-) po nazwie instrukcji

• nazwa -

Przykład

```
.RLOAD 25 5 1K
```

```
.RLOAD -
```

Opornik RLOAD został usunięty z układu.

Jeżeli nie chcemy usuwać, ani też dokonać przełączenia elementu w inne miejsce układu, lecz chcemy tylko zmienić jego wartość, to wówczas można użyć następującej instrukcji:

• nazwa = nowa wartość

Przykład

```
RLOAD = 2.2 KΩ
```

Opornik RLOAD zmienił swoją wartość na nową, równą 2.2 kΩ.

W przypadku, gdy chcemy dokonać zmiany pierwszej stałej w określeniu wartości, znacznie lepiej jest posłużyć się instrukcją



***MODIFY** (p. 4.7.5). Użycie tej instrukcji powoduje tylko zmodyfikowanie wartości w utworzonej macierzy, a nie powoduje układania od nowa równań układu. Oczywiście w przypadku nadawania elementowi nowej wartości funkcjonalnej należy posłużyć się koniecznie instrukcją wymiany (chyba, że zmieniamy tylko pierwszą stałą w opisie tej wartości).

Należy zauważyć, że nie można w ten sposób wymieniać zdań opisujących modele użytkownika (p. 4.5), gdyż są one jak gdyby zawarte w innym obwodzie (rysunek 4.57).

4.7. Zdania sterujące

Zdania sterujące służą do sterowania analizą i wyjściem z programu NAP2 oraz do sterowania współpracą i bibliotekami modeli. Wszystkie instrukcje sterujące rozpoczynają się od znaku gwiazdki (*) i mogą składać się tylko z pojedynczego rozkazu lub też zawierać listę parametrów. Należy zwrócić szczególną uwagę na właściwe stosowanie zdań sterujących, gdyż popełnienie błędu w ich użyciu często prowadzi do straty dużej ilości czasu komputera, a co za tym idzie do niepotrzebnych kosztów.

4.7.1. Korzystanie z bibliotek modeli

W bibliotekach modeli można zapamiętać dowolny zbiór instrukcji w celu ich późniejszego wielokrotnego wykorzystania. W programie NAP2 VERSION PL84 jest dostępnych sześć bibliotek, o kolejnych nazwach *LIB1, *LIB2, ... *LIB6.

Biblioteka *LIB1 jest zabezpieczona przed indywidualnym zapisem i użytkowanie jej jest możliwe dopiero po podaniu hasła.

W bibliotece *LIB2 są zawarte listy parametrów diod, tranzystorów bipolarnych i tranzystorów polowych złączowych JFET (p. tab. 4.4 i 4.7).

Biblioteka *LIB3 jest przeznaczona do wpisywania i zapamiętywania własnych modeli oraz makromodeli użytkowników i jest wykorzystywana przez studentów podczas zajęć projektowych z Analizy i Projektowania Komputerowego Układów Elektronicznych.

W bibliotece *LIB4 są zawarte modele i makromodele tyrystora (patrz p. 4.4.8).



W bibliotece *LIB5 są zawarte makromodele bramek TTL i ECL [49].

W bibliotece *LIB6 są zawarte makromodele wzmacniaczy operacyjnych (patrz p. 4.4.7).

W tablicy 4.15 przedstawiono sposób zorganizowania bibliotek dla programu NAP2.

T a l i c a 4.15

Nr biblioteki	Komentarz	Uwagi
*LIB1	Zabezpieczona przed indywidualnym użytkownikiem	-
*LIB2	Modele diod i tranzystorów	tylko do odczytu
*LIB3	Do korzystania przez indywidualnych użytkowników	można wpisywać i odczytywać
*LIB4	Makromodele tyrystorowe	tylko do odczytu
*LIB5	Makromodele bramek TTL i ECL	tylko do odczytu
*LIB6	Makromodele wzmacniaczy operacyjnych	tylko do odczytu

Ze względu na zawartość bibliotek *LIB2, *LIB4, *LIB5 i *LIB6 nie wolno do nich dodawać żadnych wpisów bez uzgodnienia z prowadzącym ani też kasować zawartych w nich zbiorów. Można natomiast wykorzystywać zawarte w nich modele i makromodele do uruchamiania własnych programów. Biblioteka *LIB3 jest ogólnie dostępna dla wszystkich użytkowników.

Poniżej zostaną podane podstawowe informacje dotyczące sposobu korzystania z biblioteki modeli.

1. Zapamiętywanie w bibliotece. Postać instrukcji służącej do zapamiętania w bibliotece własnego modelu użytkownika jest następująca:

* LIBn nazwa +
Instrukcje przeznaczone do przechowywania
> (koniecznie w kolumnie !)

gdzie: nazwa - dowolna, a znak (+) wskazuje, że zostanie ona umieszczona w katalogu *LIBn (n = 1, 2 ... 6) jako nazwa



członu, pod którą są przechowywane instrukcje; dla indywidualnych użytkowników jest dopuszczalne używanie tylko $n = 3$,

- > - symbol kontynuacji; umieszczony w kolumnie 1 wskazuje koniec instrukcji przeznaczonych do przechowywania, jest to drugi wyjątek od swobodnoformatowego języka wejściowego, gdyż znak kontynuacji musi być umieszczony bezwzględnie w kolumnie 1.

Należy podkreślić, że instrukcje zawarte w bibliotece zostają podane dopiero wtedy kontroli błędu, gdy są wprowadzone do zbioru obwodu. Wszelkie modyfikacje tych instrukcji są więc możliwe dopiero po wprowadzeniu do zbioru obwodu. Przy wprowadzeniu własnego modelu do biblioteki należy szczególną uwagę zwrócić na umieszczenie znaku kontynuacji ">" w kolumnie 1, jako sygnału końca wpisywania modelu. Brak tego znaku (lub umieszczenie go w innej kolumnie) spowoduje, że wszystkie wczytywane instrukcje będą ładowane do biblioteki i będzie konieczna interwencja operatora.

Przykłady

*LIB3 BC107 +

BC107/NPN/ IS 7.83E-12 VT 25.6MV NI 4050 NV 1.54 >

TF .49NS TR 114NS CE 8.6PF CC 7.2PF GA .43 >

FI .86V AF .99723 AR .75 NG 2.5E-4 GS .55

> : W KOLUMNIE 1

W bibliotece *LIB3 zostanie zapamiętana lista parametrów tranzystora BC 107

LIB3 HYBRIDPI +

RX 1 4 50

RPI 4 2 2.5K

CPI 4 2 1NF

CMV 3 4 2PF

IGM 3 2 40MMHO VRPI

GO 3 2 1E-5

> : W KOLUMNIE 1



W bibliotece *LIB3 zostanie zapamiętany model tranzystora dla małych sygnałów i będzie przechowywany pod nazwą HYBRIDPI.

Należy zaznaczyć, że przy wpisywaniu do biblioteki zbioru o danej nazwie, zbiór o takiej samej nazwie (o ile się znajduje w bibliotece) zostaje z niej automatycznie usunięty.

2. Kopiowanie z biblioteki. Postać instrukcji służącej do kopiowania z biblioteki jest następująca:

*LIBn	nazwa
-------	-------

gdzie nazwa - instrukcje przechowywane w danej (n) bibliotece pod tą nazwą są wpisywane do zbioru odwodu; w tym momencie następuje dopiero kontrola błędów.

Przykład 1

```
*LIB2  N3055
TR7      8    21    30    N3055
```

Z biblioteki 2 zostaje odczytana lista parametrów dla tranzystora 2N3055 (opuszczono 2 przed nazwą, bo nazwa listy parametrów musi zaczynać się od litery). Tranzystor TR7 wykorzystujący listę parametrów N3055, zawartą w bibliotece *LIB2, jest umieszczony pomiędzy węzłami 8 (kolektor), 21 (baza) i 30 (emiter).

Sposób wywołania podany w powyższym przykładzie umożliwia korzystanie z biblioteki modeli diod, tranzystorów i tranzystorów JFET zawartych w bibliotece *LIB2 i pozwala na znaczne ułatwienie obliczeń. W przypadku braku danego typu tranzystora w bibliotece LIB2, można policzyć jego parametry na podstawie katalogu (p. 4.4.2) i po wprowadzeniu go do biblioteki *LIB3 w identyczny sposób wykorzystywać jego listę parametrów.

Przykład 2

```
QACMODEL  7=1    8=2    9=3
*LIB3  HYBRIDPI
.RX = 20
Q*
```



Model HYBRIDPI zawarty w bibliotece 3 jest wprowadzony do zbioru obwodu i jego węzły zewnętrzne 1, 2, 3 są odpowiednio dołączone do zewnętrznych węzłów obwodu 7, 8, 9. Po wprowadzeniu tego modelu dokonano modyfikacji wartości opornika RX z 50Ω na 20Ω .

3. Usuwanie z biblioteki. W celu usunięcia modelu z biblioteki należy posłużyć się instrukcją o następującej postaci:

*LIBn nazwa -

gdzie znak (-) wskazuje, że nazwa modelu i wszystkie instrukcje należące do niego są usuwane z biblioteki.

Dla indywidualnego użytkownika jest dopuszczalne tylko $n = 3$.

4. Otwieranie (kasowanie), drukowanie i dziurkowanie zawartości biblioteki. W celu otwarcia biblioteki (użycia jej po raz pierwszy) należy użyć następującego zdania:

*LIBn NEW

Instrukcja ta przygotowuje zbiór danych do zapisania w bibliotece pod nazwą *LIBn ($n = 1, 2 \dots 6$). Instrukcja ta powoduje skasowanie (wyczyszczenie) całej zawartości biblioteki o numerze n, i jej użycie jest uzasadnione tylko w przypadku otwierania danej biblioteki po raz pierwszy. W żadnym przypadku nie jest dopuszczalne używanie tej instrukcji przez indywidualnego użytkownika bez porozumienia z prowadzącym zajęcia.

W celu wydrukowania lub wydziurkowania (na tasiemkę papierową) zawartości danej biblioteki należy użyć instrukcji o postaci

* LIBn $\left\{ \begin{array}{l} \text{PRINT} \\ \text{PUNCH} \end{array} \right\}$

gdzie: PRINT - dotyczy wydrukowania zawartości biblioteki *LIBn,

PUNCH - dotyczy dodatkowo wydziurkowania zawartości biblioteki *LIBn na tasiemce papierowej (wydruk następuje również jak przy instrukcji PRINT).



5. Kompresja zbioru w bibliotece. Jeżeli do danej biblioteki było wpisywane wiele zbiorów przy jednoczesnym kasowaniu starych, to na dysku w miejscu starych, usuniętych zbiorów są "puste miejsca". W celu zapełnienia tych wolnych miejsc należy zbiór poddać kompresji. Dokonuje się tego w następujący sposób:

* LIBn PRINT -

Zawartość biblioteki * LIBn zostanie wydrukowana oraz jednocześnie zbiory zostaną poddane kompresji.

4.7.2. Rozpoczęcie obliczeń

Po uruchomieniu programu NAP2 i ewentualnym zapamiętaniu modeli i makromodeli użytkownika w bibliotekach, przy pomocy instrukcji podanych w p. 4.7.1, należy użyć następującego zdania sterującego:

*CIRCUIT

Spowoduje ono wyzerowanie wewnętrznej bazy danych i przygotuje system do wprowadzenia nowego opisu obwodu. Jeśli krata ze zdaniem *CIRCUIT zostanie pominięta przy pierwszym wejściu do programu (p. 4.13), wówczas zdania opisu obwodu następujące po niej również zostaną pominięte. W przypadku pominięcia tej karty w następnym pliku danych, cały program następujący po tej karcie zostanie włączony do programu poprzednika (poprzedni plik danych), co na ogół powoduje wystąpienie błędów wykonania i przerwanie obliczeń.

4.7.3. Drukowanie informacji dodatkowych

W przypadku, gdy użytkownik chce otrzymać nieco więcej informacji o badanym układzie niż zapewnia to standardowe ustawienie parametrów, należy po zdaniu *CIRCUIT podać następujące zdanie sterujące:

* LIST n ...



gdzie n - liczba całkowita nie większa od 20 (dla n ujemnego listy wydruku podane w tablicy 4.16 nie będą wydrukowane).

W tablicy 4.16 podano znaczenie poszczególnych list wydruku i odpowiadające im wartości standardowe.

T a b l i c a 4.16

n	Lista wydruku	Wartość standardowa
1	Drukuj obraz kart wejściowych	tak (+1)
2	Drukuj obraz układu	nie (-2)
9	Drukuj statystyki przebiegu	nie (-9)
3÷8 10÷20	Wydruk wyników pośrednich	nie

Numerzy 3-8 i 10-20 są zarezerwowane dla testowania programu poprzez umożliwienie wydruku wyników pośrednich i w żadnym przypadku nie powinny być stosowane przez indywidualnego użytkownika.

Przykład

```
*LIST 2 9
  ⋮
*LIST -9
```

Pojawienie się pierwszego zdania spowoduje, że będzie drukowany obraz układu oraz statystyki przebiegu wykonywanego programu. Drugie zdanie sterujące odwoła drukowanie statystyk przebiegu.

4.7.4. Przechowywanie informacji o obwodzie
w zbiorze pomocniczym

W celu zapamiętania zbioru danych obwodu wraz ze zbiorem odpowiedzi (o ile zostały policzone) w pomocniczym zbiorze MAA (rysunek 4.1 i 4.3), należy posłużyć się następującym zdaniem:

*** SAVE**



Wywołanie informacji zapisanych w zbiorze MAA i wprowadzenie ich z powrotem do bazy danych odbywa się poprzez podanie następującego zdania:

*LOAD

Obie te instrukcje są bardzo użyteczne w tych przypadkach, gdy chcemy zapamiętać topologię obwodu i jego nominalne rozwiązanie w celu używania go w dalszych etapach analizy. Na przykład, po wykonaniu optymalizacji układu możemy zapamiętać optymalne rozwiązanie i wywoływać je wielokrotnie. W przypadku szeregu analiz (np. analiza najgorszego przypadku) rozwiązanie to jest likwidowane i gdyby nie możliwość uprzedniego zapamiętania go przy pomocy instrukcji *SAVE, to analizę należałoby wykonywać ponownie.

Należy pamiętać, że nie wolno używać instrukcji *LOAD bez uprzedniego wykonania instrukcji *SAVE, gdyż spowoduje to załadowanie układu (i jego rozwiązania) zapamiętanego przez poprzedniego użytkownika.

4.7.5. Instrukcja *MODIFY i *TOLE

Instrukcja *MODIFY służy do:

- 1) modyfikacji wartości elementów i parametrów, przy czym jeżeli wartość jest funkcjonalna, to w określeniu wartości zmieniona jest stała pierwsza,
- 2) ustalenia warunków początkowych dla napięć w węzłach i prądów pierwotnych,
- 3) definiowania klas tolerancji,
- 4) definiowania elementów skorelowanych.

Pierwsza postać instrukcji *MODIFY służy do wykonania p. 1 i p. 2, i jest następującej postaci:

$$*MODIFY \left\{ \begin{array}{l} VNALL \\ IALL \\ V \text{ nr węzła} \\ I \text{ nazwa elementu} \\ Nazwa \end{array} \right\} \left[\left[\begin{array}{c} * \\ / \end{array} \right] \right] \text{ stała } \dots$$


gdzie:

- VNALL - wszystkie napięcia węzłów,
- IALL - wszystkie prądy pierwotne,
- V nr węzła - napięcie węzła,
- I nazwa elementu - nazwa prądu pierwotnego,
- Nazwa - dowolna wartość nazwana,
- * - względna zmiana starej wartości - nowa
wartość = stara wartość * stała,
- / - względna zmiana starej wartości - nowa
wartość = stara wartość/stała.

Przykład 1

*MODIFY R5 = 4.7K R6 *1.05 R7/1.25 RE1.E 20V

Wartość opornika R5 zostanie zmieniona na 4.7 k Ω , R6 - zwiększona o 5%, R7 - zmniejszona o 20% ($X/1.25 = 0,8 X$), a wartość napięcia źródła napięciowego RE1.E zmieni się na 20 V.

Przykład 2

*MODIFY VNALL 0 IALL 0

Wszystkie napięcia węzłowe i prądy pierwotne zostaną zmodyfikowane na zero.

Druga postać instrukcji *MODIFY służy do wykonania p. 3 i p. 4 i jest następującej postaci:

*MODIFY $\left[\left[\text{nr klasy tolerancji} \right] +\text{tole} \right] -\text{tole}$ nazwa elementu
(nazwa elementu ...)

gdzie: nr klasy tolerancji - liczba całkowita od 1 do 10 oznaczająca klasę tolerancji,

- +tole - względna tolerancja dodatnia w danej klasie tolerancji,
- tole - względna tolerancja ujemna w danej klasie tolerancji, jeżeli nie podano, to: -tole = +tole,



nazwa elementu - nazwa dowolnego elementu lub źródła niezależnego,
 (nazwa elementu...) - elementy skorelowane z danym elementem.

Przykład 3

```
*MODIFY 1 .1 .2 R1 (R2 R3) R4
*MODIFY 2 .1 R10
*MODIFY 5 .001 .005 TR1.ŁAF
```

Rezystory R1 i R4 są przypisane do pierwszej klasy tolerancji z tolerancją +0.1 i -0.2, a zmiany rezystorów R2 i R3 są skorelowane ze zmianą wartości rezystora R1.

Druga instrukcja *MODIFY przypisuje opornik R10 do drugiej klasy tolerancji z tolerancją +0.1 -0.1 (opuszczenie -tole powoduje nadanie jednakowej tolerancji w górę i w dół).

Trzecia instrukcja *MODIFY przypisuje do 5-tej klasy tolerancji współczynnik wzmocnienia prądowego AF tranzystora TR1 z tolerancją 0.001 i -0.005.

W przypadku, gdy układ zawiera bardzo dużo elementów o tej samej tolerancji, to dla ułatwienia można użyć instrukcji *TOLE, która jest postaci:

*TOLE nr klasy tolerancji [+tole [-tole]]

gdzie: nr klasy tolerancji - liczba całkowita od 1 do 10 oznaczająca klasę tolerancji,

+tole - względna tolerancja dodatnia w danej klasie tolerancji,

-tole - względna tolerancja ujemna w danej klasie tolerancji, jeżeli nie podano, to: -tole = +tole.



Wszystkie elementy układu podane po rozkazie *TOLE przyjmują tolerancję zdefiniowaną tym rozkazem. Kolejny rozkaz *TOLE zmienia klasę tolerancji na nim zdefiniowaną. W celu zakończenia przypisywania klas tolerancji elementom należy podać rozkaz *TOLE 0.

Przykład

```
*CIRCUIT
*TOLE 1 .1 .2
R1 5 6 4.7K
R2 1 2 1K
C3 3 4 1PF
*TOLE 2 .1
R7 7 8 1K
R8 9 10 4.7K
L1 2 3 1K
*TOLE 0
C5 1 7 1P
C6 2 8 3P
R14 8 7 10K
:
:
```

Elementy R1, R2 i C3 mają pierwszą klasę tolerancji $\pm 10\%$, -20% , elementy R7, R8 i L1 są w drugiej klasie tolerancji $\pm 10\%$, natomiast elementy C5, C6 i R14 nie należą już do żadnej klasy tolerancji.

4.7.6. Zmiana parametru

W celu dokonania zmiany wartości dowolnego elementu lub parametru w pewnym przedziale służy instrukcja o następującej postaci:

* {TIME} [/nwydr] [nazwa 1] min max [krok] nazwa 2 od do ...
 * {FREQ}

gdzie: nwydr - określa częstotliwość wydruku jeżeli nwydr > 0 to wyniki są drukowane co nwydr kroków, jeżeli nwydr ≤ 0 to wyniki są drukowane co MAXSTEP (p. 4.7.7); standard programowy jest nwydr = 0,



nazwa 1 - nazwa wartości podlegającej zmianie; jeżeli jest ona pominęta, zmienia się czas - TIME lub częstotliwość - FREQ, w zależności od typu analizy,
 min - minimalna wartość zmienianej wielkości,
 max - maksymalna wartość zmienianej wielkości,
 krok - jeżeli krok > 0 , to są przyjmowane liniowe zmiany ze stałym krokiem, jeżeli krok = 0, to krok określany jest automatycznie (standard programowy), jeżeli krok < 0 , to przyjmowane są zmiany logarytmiczne z liczbą przedziałów równą $(-\text{krok})$,
 nazwa 2 - nazwa wielkości zmienianej jednocześnie ze zmianą nazwy 1, FREQ lub TIME,
 od - wartość początkowa nazwy 2,
 do - wartość końcowa nazwy 2.

Przykłady

```

*TIME 0 1US : ZMIANA CZASU OD 0 DO 1US Z AUTOMATYCZNIE
          : DOBIERANYM KROKIEM
*FREQ 1Hz 100KHz - 60 : LOGARYTMICZNA ZMIANA CZĘSTOTLIWOŚCI
          : od 1Hz do 100KHz z 60 PRZEDZIAŁAMI
*TIME R1.E 0 10V R2.E 0 - 10V : JEDNOCZESNA ZMIANA
          : NAPIĘĆ R1.E I R2.E.
  
```

4.7.7. Sterowanie analizą i wyjściem

Zdanie służące do określania rodzaju obliczeń i wyjść z programu jest następującej postaci:

$* \left\{ \begin{array}{c} \text{DC} \\ \text{DCTR} \\ \text{TR} \\ \text{AC} \\ \text{PZ} \end{array} \right\} \left[\text{lista 1} \right]$	$* \left\{ \begin{array}{c} \text{PLOT} \\ \text{P PLOT} \\ \text{PUNCH} \end{array} \right\}$	$* \text{PRINT} \left[\text{lista 2} \right] \dots \left[\text{lista 3} \right]$
---	--	--

gdzie: *DC - żądanie wykonania analizy dla prądu stałego DC (direct current),

*DCTR - żądanie wykonania analizy stanu nieustalonego z auto-



matycznym wyznaczeniem warunków początkowych (analiza DC zostanie przeprowadzona jako pierwsza w celu wyznaczenia tych warunków),

- * TR - żądanie przeprowadzenia tylko analizy stanu nieustalonego TR (transient response),
- * AC - żądanie analizy w dziedzinie częstotliwości (alternative current),
- * PZ - żądanie analitycznego lub iteracyjnego wyznaczenia biegunów i zer (poles and zeros).

Po każdej instrukcji * RUN zostaje wykonana jedna analiza w dziedzinie czasu lub też jedna analiza w dziedzinie częstotliwości. W jednym programie można więc wykonać cały szereg analiz różnego typu.

Pierwsza lista po żądaniu wykonania danego typu analizy - listą 1 - zawiera wyniki, które mają być wyprowadzane w trybie "on line", natomiast lista 2 oraz 3 zawierają wyniki, które mają być wyprowadzane w trybie "off line", czyli po ich uprzednim zapisaniu w pomocniczym zbiorze dyskowym WWY i opracowaniu.

Zanim przejdziemy do szczegółowego opisu listy 1, 2 i 3, poniżej przedstawimy dopuszczalne opcje listy wyjściowej.

Przykład

Wartości nazwane	R1 C5 BC107.AF R5.E
Odpowiedzi obwodu	V5 VR1 IC5 IRLOAD
Funkcje przenoszenia (transmitancje)	V1/ER1 IRLOAD/JG5
Funkcje wrażliwościowe	V1/R1 VR5/C2

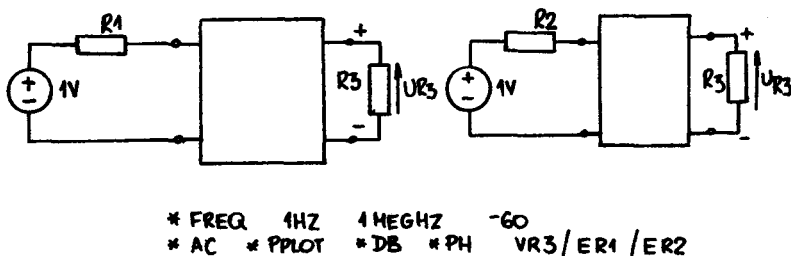
Nieco dokładniejszego omówienia wymaga sposób obliczania funkcji przenoszenia (transmitancji) oraz wrażliwości. Ogólna postać instrukcji do obliczania transmitancji jest następująca:

$$\left[\begin{array}{cc} \text{odpowiedź} & \left[\begin{array}{c} E \\ J \end{array} \right] \\ \text{pierwotna} & \end{array} \right] \left[\begin{array}{cc} \text{nazwa} & \left[\begin{array}{c} E \\ J \end{array} \right] \\ \text{elementu} & \end{array} \right] \left[\begin{array}{cc} \text{nazwa} & \left[\begin{array}{c} E \\ J \end{array} \right] \\ \text{elementu} & \end{array} \right] \dots$$

gdzie są dopuszczalne tylko elementy typu R lub G.



Obliczanie transmitancji jest oparte na modelu dla małych sygnałów poprzez wstawienie fikcyjnego źródła napięciowego lub prądowego (E lub J) w szereg lub równolegle do elementu. Wszystkie pozostałe źródła są ignorowane, a układ znajduje się w punkcie pracy obliczanym uprzednio podczas analizy DC. Na rysunku 4.58 przedstawiono przykład obliczania funkcji przenoszenia.



Rys. 4.58. Obliczanie funkcji przenoszenia (transmitancji)

Oporniki R1 i R2 definiują końcówki wejściowe, a opornik R3 - końcówki wyjściowe, pomiędzy którymi są obliczane dwie funkcje wzmocnienia napięciowego. Zarówno amplituda (w decybelach), jak i kąt fazowy (w stopniach) będą drukowane i wykreślone w funkcji częstotliwości (od 1 Hz do 1E6 Hz), z 60 punktami na osi częstotliwości w skali logarytmicznej.

Postać instrukcji do obliczania wrażliwości jest następująca:

odpowiedź	/	wartość	[	wartość	]	...
pierwotna		elementu		elementu		

gdzie wartość elementu jest wartością dowolnego elementu.

Obliczanie wrażliwości jest również oparte na modelu dla małych sygnałów.

Wrażliwość $S_{i,j}$ pierwotnej odpowiedzi X_i względem parametru obwodu Cj jest zdefiniowana następująco:

$$S_{i,j} = \frac{\partial X_i}{\partial C_j} C_j. \quad (4.22)$$

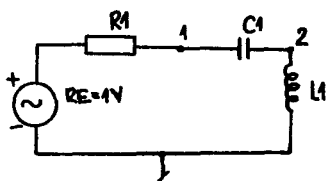


W przypadku, gdy zadeklarowano elementy skorelowane z danym elementem (p. 4.7.5), to ogólny wzór dla wrażliwości z uwzględnieniem korelacji będzie postaci

$$S_{i,j} = \frac{\partial X_i}{\partial C_j} C_j + \sum_{k=1}^n \frac{\partial X_i}{\partial C_k} C_k, \quad (4.23)$$

gdzie n - liczba elementów skorelowanych z elementem C_j .

Na rysunku 4.59 przedstawiono przykład obliczania wrażliwości potencjału V_2 , względem elementów R_1 , C_1 i L_1 , szeregowego połączenia RLC.



* AC * DB * PH V2 V2/R1/C1/L1

Rys. 4.59. Obliczanie wrażliwości

Na liście wyjściowej są jeszcze dopuszczalne następujące podroz-
kazy:

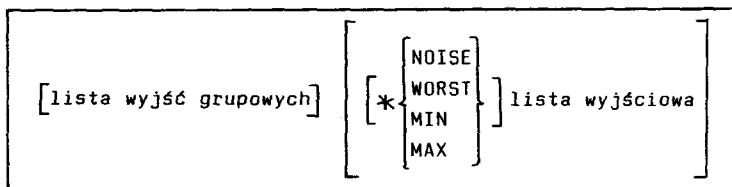
- * F - powoduje wyznaczenie współczynników transformaty Fourie-
ra okresowej odpowiedzi czasowej (p. 4.7.9),
- * Y - służą do identyfikacji wielkości, które będą miały wspól-
ną oś rzędnych y , muszą one być zawarte pomiędzy (*Y...
*Y) tymi podrozkazami,
- * RE, * IM - wyznaczenie wartości części rzeczywistej (* RE) lub uro-
jonej (* IM) wyników analizy AC,
- * MA, * DB - wyznaczenie amplitudy (* MA) lub amplitudy w decybelach
(* DB) wyników analizy AC,
- * PH, * RAD - wyznaczenie fazy w stopniach (* PH) lub radianach (* RAD)
wyników analizy AC.



Podrozказы te odnoszą się do najbliższej nazwy umieszczonej za nimi. Dopuszczalne jest umieszczenie przed daną nazwą kilku tych podrozkazów, np. (*DB *PH VR5) lub też (*RE *IM VR5).

Teraz możemy opisać kolejno 3 listy z instrukcji sterowania analizą i wyjściem.

Lista 1 - postać listy 1 jest następująca:



gdzie: dopuszczalne opcje listy wyjściowej zostały opisane powyżej.

Lista wyjść grupowych określa następujące grupy odpowiedzi:

VNALL - wszystkie napięcia węzłów,

VALL - wszystkie napięcia na elementach,

IALL - wszystkie prądy elementów,

WALL - wszystkie wartości elementów,

PWALL - wszystkie straty mocy w elementach,

ALL - wszystkie napięcia, prądy, wartości i straty mocy w elementach,

$\begin{Bmatrix} \text{VNALL} \\ \text{VALL} \\ \text{IALL} \end{Bmatrix} / \begin{Bmatrix} \text{E} \\ \text{J} \end{Bmatrix}$ nazwa elementu - funkcje przenoszenia od elementu do wszystkich napięć węzłów, pierwotnych napięć na elementach lub wszystkich prądów pierwotnych elementów,

odpowieź / $\begin{Bmatrix} \text{EALL} \\ \text{JALL} \end{Bmatrix}$ pierwotna - funkcje przenoszenia od wszystkich elementów typu R i G do odpowiedzi pierwotnej,

$\begin{Bmatrix} \text{VNALL} \\ \text{VALL} \\ \text{IALL} \end{Bmatrix} /$ nazwa elementu - wrażliwość wszystkich napięć węzłów, pierwotnych napięć na elementach lub pierwotnych prądów względem wartości elementu,

odpowieź / WALL pierwotna - wrażliwość odpowiedzi pierwotnej względem wszystkich wartości elementów.

Rozkazy występujące w liście 1 mają następujące znaczenia:

*NOISE - żądanie wykonania analizy szumowej,

*MIN *MAX *WORST - podrozказы żądające wykonania analizy najgorszego przypadku.



Współczynnik szumów jest szacowany poprzez obliczenie charakterystyk przenoszenia od wszystkich elementów R i G do odpowiedzi pierwotnej X_i

$$N_i = \sqrt{4kT \Delta f \sum_j \frac{\partial x_i^2}{\partial E_j^2} |R_j| + \sum_k \frac{\partial x_i^2}{\partial J_k^2} |G_k|}, \quad (4.24)$$

gdzie: N_i - średniokwadratowa wartość napięcia szumów,
 $\Delta f = 1 \text{ Hz}$,

$\frac{\partial x_i}{\partial E_j}$ - transmitancje od elementów R,

$\frac{\partial x_i}{\partial J_k}$ - transmitancje od elementów G,

$4kT = 1.6E20$ dla $TEMP = 25^\circ C$.

Analizę szumową można wykonać tylko podczas analizy dla prądu stałego DC.

Przykład

*DC *NOISE V1

*RSS - powoduje obliczenie średniokwadratowej wartości wrażliwości (wartości skutecznej).

Wartość średniej kwadratowej wrażliwości RSS (Root Sum Square) jest obliczana na podstawie następującej zależności:

$$RSS_i = \sqrt{\sum_{j=1}^n (S_{i,j} \Delta \text{tol}_j)^2}, \quad (4.25)$$

gdzie: Δtol_j - względny przedział tolerancji parametru obwodu C_j ,

$S_{i,j}$ - wrażliwość z uwzględnieniem korelacji,

n - liczba elementów skorelowanych.

Wartość wrażliwości średniokwadratowej RSS jest ściśle związana ze standardową dewiacją i może być miarą zmiany oczekiwanej badanej odpowiedzi obwodu.

Zakładając, że liczba przypadkowo zmieniających się parametrów obwodu jest na tyle duża, że rozkład wartości badanej odpowiedzi jest



zbliżony do normalnego, to główna część odpowiedzi obwodu będzie miała wartość odległą od nominalnej nie więcej niż o połowę wartości RSS z prawdopodobieństwem 90%.

W celu wykonania analizy najgorszego przypadku należy posłużyć się jednym z trzech rozkazów: rozkaz *MIN określa tylko minimalną wartość odpowiedzi, *MAX tylko maksymalną, natomiast rozkaz *WORST zarówno minimalną jak i maksymalną i nominalną wartość odpowiedzi.

Do przewidywania najgorszych przypadków odpowiedzi pierwotnych jest wykorzystywany znak wrażliwości. Parametry, z określonymi (przy pomocy instrukcji *MODIFY) tolerancjami, są następnie przyjmowane jako wartości skrajne, równe granicom tolerancji. Program sprawdza, czy wrażliwości zmieniły znak w rozwiązaniu przy najgorszym przypadku. Zmiana znaku oznacza, że najgorszy przypadek leży pomiędzy granicami tolerancji. Rozwiązanie nie jest wówczas najgorszym przypadkiem. Pewności jednak, że rozwiązanie jest rzeczywiście najgorszym przypadkiem, nie mamy nawet wtedy, gdy nie wystąpi zmiana znaku wrażliwości, gdyż mógł on się zmienić więcej niż raz w przedziale tolerancji. Przy każdej z tych analiz jest obliczana również wartość RSS.

Przykład 1

*DC ALL VNALL VRLOAD/WALL

W tym przykładzie instrukcja ALL spowoduje wydrukowanie wartości wszystkich napięć, prądów i strat mocy w elementach oraz ich wartości. Instrukcja VNALL spowoduje dodatkowo wydrukowanie wszystkich napięć węzłowych, a instrukcja VRLOAD/WALL wrażliwości napięcia na oporniku RLOAD względem wszystkich wartości elementów.

Przykład 2

*MODIFY 1 .1 R1 R2 R3

*MODIFY 2 .001 .005 TR1.tAF TR2.tAF

*DC *WORST V20

W przykładzie 2, po ustawieniu instrukcjami *MODIFY dwóch klas tolerancji, zostanie przeprowadzona analiza najgorszego przypadku napięcia V20 (np. napięcia wyjściowego układu). Dla każdego najgorszego przypadku będzie również obliczona średnia kwadratowa wrażliwości.



Lista 2

Lista 2 występuje po następujących podrozkazach wyjściowych:

- *PRINT - żądanie wydruku tabelarycznego,
- *PLOT - żądanie wykresu wyników na drukarce wierszowej,
- *PLOT - żądanie wydruku tabelarycznego i wykresu na drukarce wierszowej,
- *PUNCH - żądanie wydruku tabelarycznego i dziurkowania wyników (na tasience papierowej).

Postać listy 2 jest następująca:

*PRINT	[X nazwa]	lista wyjściowa ...
* { PLOT PLOT PUNCH }	[[X _p [X nazwa] [min max]]]	
		nazwy wyjściowa [min max] ...

gdzie: X_p - liczba punktów w kierunku osi X (pionowym)
 X nazwa - zmienna niezależna,
 min - minimalna wartość osi,
 max - maksymalna wartość osi.

Postać listy 2 zależy od poprzedzającego ją podrozkazu rodzaju wydruku. W przypadku użycia podrozkazu *PRINT, wystarczy użyć po nim jednej z opisanych w tym punkcie opcji listy wyjściowej.

W przypadku żądania drukowania wyników na drukarce wyjściowej należy określić format tego wykresu. Ostatnia instrukcja zmiany parametru przed instrukcją analizy (p. 4.7.6) określa standardową wartość programową zmiennej niezależnej w wykresach. Istnieje jednak możliwość dowolnego określenia tej zmiennej przez użytkownika, poprzez podanie nazwy zmiennej niezależnej (X nazwa). Liczba punktów w kierunku osi pionowej jest definiowana poprzez podanie parametru X_p. Jeżeli jest on opuszczony, to oś pionowa musi być zdefiniowana w instrukcji poprzedniej (dotyczy wykresów w skali logarytmicznej w przypadku analizy AC).

W przypadku nie podania początkowych i końcowych punktów osi zostają one przez program określone automatycznie.

Przykład 1

*DCTR *PLOT (100 ICX) V5 -2V 2V V6



W przykładzie 1 napięcia V5 i V6 zostaną wykreślone w funkcji prądu ICX. Wykres będzie zawierał 100 punktów w kierunku osi X. Minimalna wartość skali na osi V5, wynosi - 2 V, a maksymalna 2 V. Skala osi V6 zostanie dobrana automatycznie.

Przykład 2

*FREQ 1Hz 100kHz -50

*AC *PLOT *DB *PH V1

W przykładzie 2 zmienną niezależną wykresu będzie częstotliwość, a amplituda w decybelach i faza w stopniach napięcia V1 zostanie wydrukowana w 50 punktach.

Przykład 3

*TIME RC.E 0 10V

*DC *PLOT (50) IRLOAD

W przykładzie 3 zmienną niezależną będzie napięcie źródła RC.E, a prąd IRLOAD zostanie wydrukowany tabelarycznie i wykreślony na drukarce wierszowej przy 50 punktach na osi X.

Lista 3 służy do zdefiniowania wskaźnika jakości do optymalizacji. Jest ona postaci

$$\text{nazwa wyjściowa} = \left\{ \begin{array}{l} \text{wymagana wartość [big]} \\ \text{nazwa 1 [nazwa 2]} \end{array} \right\} \dots$$

gdzie: big - wstępne oszacowanie "bigabs", które jest równe maksymalnej wartości odpowiedzi wyjściowej (p. 4.7.8),
nazwa 1 - nazwa parametru uzależnionego funkcyjnie, który określa wymaganą wartość,
nazwa 2 - nazwa wartości parametru uzależnionego funkcyjnie, który określa wstępne oszacowanie "bigabs".

Do optymalizacji jest przyjmowany następujący wskaźnik jakości:

$$WJ = \min \sum \left(\frac{WNW - WW}{BN2} \right)^2, \quad (4.26)$$



gdzie: WJ - wskaźnik jakości,
 WNW - wartość nazwy wyjściowej,
 WW - wymagana wartość lub wartość nazwy.1,
 BN2 - wartość nazwy 2 lub big.

4.7.8. Optymalizacja

W aktualnej wersji programu NAP2 jest możliwa optymalizacja w dziedzinie prądu stałego *DC. Podczas optymalizacji pewne parametry obwodu są modyfikowane w celu uzyskania żadanego rozwiązania, określonego w liście 3 (p. 4.7.7).

Do określenia elementów, które mają być zmieniane w czasie optymalizacji służy instrukcja *OPTIM, o następującej postaci:

```
*OPTIM [tole [tolefac]] nazwa [big [min [max [krok]]]] nazwa ...
```

gdzie: tole - proces optymalizacji zostaje zatrzymany, gdy zmiana parametrów "tole *bigabs" nie poprawia zbieżności rozwiązania. "bigabs" jest maksymalną bezwzględną wartością parametru, jaką przyjmuje on w procesie optymalizacji.

tolefac - określa minimalny krok zmian parametru, jako równy: tole + tolefac * bigabs,

nazwa - nazwa elementu, który ulega zmianie w trakcie optymalizacji,

min - dolne ograniczenie wartości elementu,

max - maksymalne ograniczenie wartości elementu,

big - wstępna ocena maksymalnej wartości elementu (bigabs),

krok - wstępne oszacowanie wielkości kroku optymalizacji.

Wbudowane wartości tych parametrów są podane w tablicy 4.17.

Tablica 4.17

tole	tolefac	min	max	big	krok
0.01	0.01	0	1E74	wartość	tole * big



W przypadku, gdy początkowa wartość parametru obwodu jest równa zero, trzeba bezwzględnie określić wartość big lub kroku w celu rozpoczęcia obliczeń. W przeciwnym przypadku krok zostanie przyjęty jako równy zero i proces optymalizacji nigdy się nie skończy.

Proces optymalizacji jest wykonywany najszybciej, gdy żądanymi rozwiązaniami są wartości odpowiedzi pierwotnych, a zmieniane parametry obwodu są wartościami elementów. W takim przypadku pochodne stosowane w optymalizacji mogą być wyznaczane analitycznie, a nie szacowane metodą perturbacji.

Przykład 1

*OPTIM R1 R2

*DC * PRINT R1 R2 VROUT = 5V

W przykładzie 1 rezystancje R1 i R2 są dobierane aż do momentu, gdy VROUT będzie równe 5 V. Pośrednie rozwiązania będą drukowane tabelarycznie.

Przykład 2

*ORTIM R7 R9

*DC ALL V5 = 6/6 IRLOAD = 5M/5M

W drugim przykładzie zostaną wydrukowane wszystkie informacje o obwodzie po optymalizacji. Wstępne oszacowanie wartości big jest równe wymaganej wartości rozwiązania.

4.7.9. Analiza Fouriera

Podrozkaz *F analizy Fouriera dowolnego przebiegu analizowanego w dziedzinie czasu może wystąpić w liście 2, po rozkazie *PRINT, *PLOT lub też *PPLOT, i jest postaci:

... * F [LWF] odpowiedź obwodu ...

gdzie: LWF określa liczbę współczynników Fouriera badanej odpowiedzi, wartość wbudowana LWF = 50.



Przedział analizy FPER powinien być określony w liście parametrów rozkazu *RUN, gdyż jego pominięcie spowoduje przyjęcie do analizy całego analizowanego czasu T. W tabelicy 4.18 podano parametry odnoszące się do analizy Fouriera i podawane w rozkazie *RUN.

T a b l i c a 4.18

Parametr	Komentarz	Wartość wbudowana
FPER	Długość analizowanego przedziału czasu T (okres)	Cały przedział czasowy analizy
NPER	Ilość analizowanych okresów T	1
FDB	Dynamika widma w DB	60

Należy pamiętać, że przy podaniu wartości okresu FPER zawsze analizowany jest końcowy fragment przedziału analizy, a więc już po ustaleniu się odpowiedzi.

Przykład

*TIME 0 SMS

*DCTR *PLOT(50) *F V10

*RUN FPER 1MS

Za ostatnią 1MS analizy zostaną obliczone współczynniki Fouriera napięcia V10.

4.7.10. Instrukcja *RESET

Podczas wykonywania analizy układu instrukcje sterujące (*DC, *DCTR, *TR, *AC, *PZ) są pamiętane przez cały czas przez program. W celu wyzerowania (wykasowania) już nieaktualnych instrukcji sterujących jest używany rozkaz

***RESET**

Użycie tej instrukcji usuwa również wszystkie zmiany parametru, a wszystkie parametry sterujące przebiegiem analizy (podawane po zdaniu *RUN) przyjmują swoje wartości standardowe.



4.7.11. Instrukcja * RUN. Parametry sterujące przebiegiem analizy

Instrukcja *RUN przekazuje sterowanie do pakietu analizy, tak więc podanie jej rozpoczyna właściwy cykl obliczeń. W instrukcji tej można dokonać zmiany wartości parametrów sterujących przebiegiem analizy w stosunku do ich wartości wbudowanych.

Zdanie *RUN jest postaci

*RUN [parametr sterujący wartość] ...

W tablicy 4.19 podano wszystkie parametry sterujące i ich wartości wbudowane.

T a b l i c a 4.19

Lp.	Parametr sterujący	Uwagi	Komentarz	Wartość wbudowana
1	VMIN	1	Początek przedziału (*VARY)	0
2	VMAX	1	Koniec przedziału (*VARY)	0
3	VSTEP	1	Krok (*VARY)	0
4	FMIN	1	Początek przedziału częstotliwości (*FREQ)	0
5	FMAX	1	Koniec przedziału częstotliwości (*FREQ)	0
6	FSTEP	1	Krok częstotliwości (*FREQ)	0
7	IMIN	1	Początek przedziału czasu (*TIME)	0
8	TMAX	1	Koniec przedziału czasu (*TIME)	1.0E20
9	STEP	2	Krok czasowy w metodzie całkowania ze zmiennym rzędem i krokiem; TMAX i TMIN są końcami przedziału czasu podawanymi bez nazwy w instrukcji * TIME	1.0E-20

1 Parametr nie przeznaczony do zmieniania przez indywidualnego użytkownika.

2 Parametr, który może być zmieniany tylko po konsultacji z prowadzącym zajęcia.

3 Parametr, który może być używany przez indywidualnych użytkowników.



T a b l i c a 4.19 (cd.)

Lp.	Parametr sterujący	Uwagi	Komentarz	Wartość wbudowana
10	MINSTEP	3	Minimalna wartość kroku całkowania w obliczeniach ze zmiennym krokiem	1.0E-20
11	MAXSTEP	2	Maksymalna wartość kroku całkowania w obliczeniach ze zmiennym krokiem. Jeżeli użytkownik poda granice przedziału czasu, wówczas wbudowana wartość kroku całkowania STEP oraz MINSTEP i MAXSTEP przyjmuje następujące wartości: STEP = ABS (TMAX - TMIN)/1.0E6 MINSTEP = ABS (TMAX - TMIN)1.0E6 MAXSTEP = ABS (TMAX - TMIN)	1.0E-20
12	TRUNC	2	Względny błąd całkowania	1E-3
13	ORDER	2	Maksymalny rząd całkowania przy analizie ze zmiennym rzędem aproksymacji wielomianowej; jeżeli ORDER < 0, to całkowanie odbywa się ze stałym rzędem = - ORDER	6
14	CONV	2	Względne kryterium zbieżności w metodzie NEWTONA-RAPHSONA; optymalna wartość CONV jest równa około 5 *TRUNC, gdyż powoduje minimalną liczbę iteracji	1E-6
15	ITER	2	Przeciętna liczba iteracji w każdym kroku całkowania równań nieliniowych	4
16	MAXII	2	Maksymalna liczba iteracji w każdym kroku całkowania równań nieliniowych (*TR)	25
17	CYCLE	2	Kryterium zatrzymania, w cyklach iteracji	1E6
18	CPU	3	Kryterium zatrzymania, w sekundach czasu procesora (cpu)	1E6
19	INCR	2	Współczynnik ograniczający szybkość wzrostu kroku czasowego	0,25



T a b l i c a 4.19 (cd.)

Lp.	Parametr sterujący	Uwagi	Komentarz	Wartość wbudowana
20	REJC	2	Współczynnik kontrolujący odrzucania kroku całkowania; jeżeli błąd zbieżności jest większy niż TRUNC/REJC, to krok jest odrzucany i wartość kroku jest zmniejszana	0,25
21	CONV1	2	Względne kryterium zbieżności w metodzie Newtona-Raphsona (*DC)	1E-6
22	MAXIT1	2	Maksymalna liczba tolerancji w nieliniowej analizie (*DC)	25
23	RSTEP	1	Stała interpolacji A podzielona przez wartość kroku, czyli $RSTEP=A/STEP$ (wartość stałej $A = 1$)	0
24	DAMP	2	Współczynnik tłumienia w metodzie Newtona-Raphsona	0.75
25	ZTOLE	1	Odpowiednik parametru "tole" w zdaniu *OPTIM	0.01
26	ZFAC	1	Odpowiednik parametru "tole-fac" w zdaniu *OPTIM	1.E-4
27	ZITER	2	Maksymalna ilość iteracji przy optymalizacji	100
28	XDAMP	1	Współczynnik tłumienia dla macierzy współczynników w optymalizacji	1
29	YDAMP	1	Współczynnik tłumienia dla wektora błędu w optymalizacji	1
30	LAMD1	1	Początkowa wartość ogólnego współczynnika tłumienia w optymalizacji	0.1
31	LAMD2	1	Mnożnik dla ogólnego współczynnika tłumienia w optymalizacji	3
32	LAMD3	1	Mnożnik dla wektora błędu w optymalizacji	1



T a b l i c a 4.19 (cd.)

Lp.	Parametr sterujący	Uwagi	Komentarz	Wartość wbudowana
33	ZERO	1	Minimalna wartość bezwzględna mogąca być dzielnikiem	1E-20
34	RZERO	1	Minimalna wartość bezwzględna rezystora, zastępowanego konduktancją	1E-9
35	OFFL	2	Numer urządzenia, na które mają być wyprowadzane w trybie "off line"	6
36	VAXIS	1	Nie używany	0
37	HOLD	3	Utrzymuje wyjście w stanie zatrzymania, tak że przebiegi z różnych obliczeń mogą być drukowane na tym samym wykresie; wystarczy podać sam parametr, bez podawania jego wartości	0
38	FUNIT	3	Zadeklarowanie wartości FUNIT = 1 spowoduje, że w analizie częstotliwość (*AC) będzie wyrażona w RAD/SEC	2
39	PERT	1	Obliczanie wrażliwości metodą zaburzeń numerycznych; wyjście jest równe różnicy pomiędzy rzeczywistymi przebiegami wyjściowymi i wartościami z poprzedniego przebiegu SAVEPERT	1
40	SCAN	1	Nie używany	10
41	XPLOT	1	Nie używany	1
42	FPER	3	Przedział czasowy, w którym następuje transformata Fouriera; FPER = TMAX - TMIN	0
43	NPER	2	Liczba analizowanych okresów przy analizie Fouriera	1
44	FDB	2	Dynamika widma w dB	60
45	RUN45	1	Nie używany	0



Należy podkreślić, że nie należy dokonywać żadnych zmian wartości parametrów sterujących bez dokładnej analizy ich wpływu na przebieg obliczeń.

4.7.13. Ustawianie tytułu wydruku

Przy każdym wydruku wyników (p. 4.7.7) można ustawić ich tytuł za pomocą następującej instrukcji:

* :	1 wiersz tytułu [>]
	[2 wiersz tytułu]

gdzie: * : - musi być w 1 i 2 kolumnie rekordu (pierwszy wyjątek od języka swobodnoformatowego);

tytuł - może stanowić dowolny tekst i o ile mieści się on w pierwszym wierszu, to nie trzeba podawać znaku kontynuacji (i wówczas drugi wiersz nie występuje).

Tytuł obliczeń można zmienić w dowolnym miejscu programu na inny. Jeżeli nowy tytuł będzie zawierać tylko jeden wiersz, to drugi wiersz nie ulegnie zmianie.

4.7.13. Rozkaz *END

Rozkaz *END przekazuje sterowanie do systemu operacyjnego i powoduje usunięcie programu NAP2 z pamięci operacyjnej. Oznacza on zakończenie obliczeń.



5. OPIS JĘZYKA WEJŚCIOWEGO PROGRAMU PAUT2

5.1. Charakterystyka programu PAUT2

Pierwsza wersja programu PAUT została opracowana w Instytucie Elektroniki Politechniki Łódzkiej w latach 1979-1980 [28, 29]. Po doświadczeniach związanych z wykorzystaniem tego programu do prowadzenia zajęć z przedmiotu "Wybrane Zagadnienia Projektowania Komputerowego" program zmodyfikowano zmieniając w sposób zasadniczy jego język wejściowy [27].

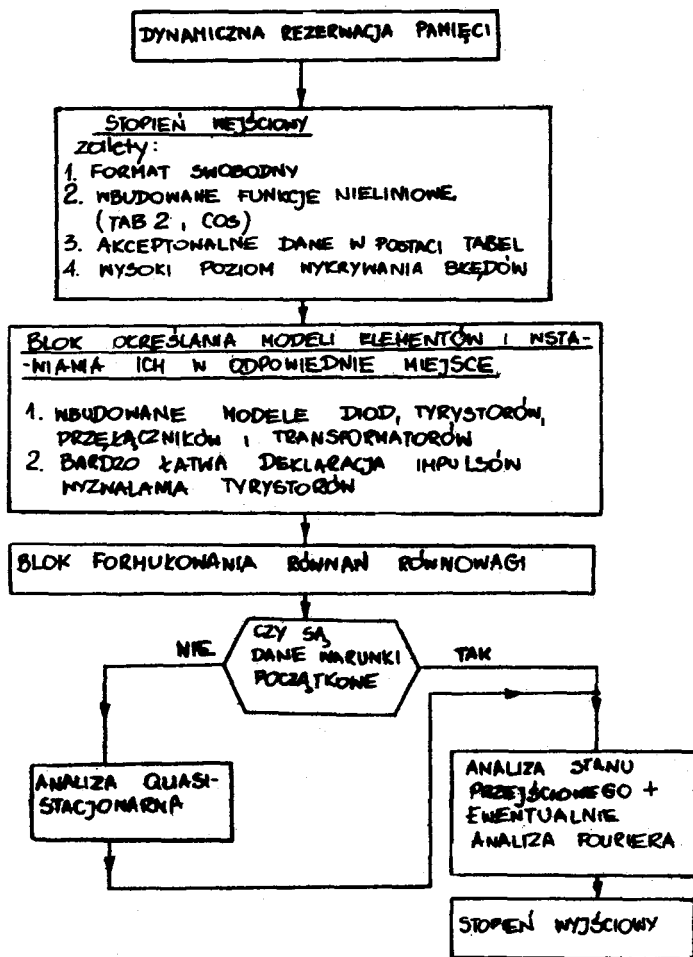
Program PAUT2 jest programem przeznaczonym do analizy układów tyrystorowych. Użycie tego typu programów jest konieczne w przypadku, gdy analizowany obwód jest bardzo duży, a wartość wymaganej pamięci operacyjnej i koszt symulacji uniemożliwiają zastosowanie programu uniwersalnego. W związku z tym, że tyrystor można traktować jako element sterowany prądem [43, 45], narzuca to odpowiednio wybór sposobu formułowania równań układu elektronicznego [33, 41].

Metoda prądów obwodowych daje bezpośrednio wartości prądów do sterowania stanu tyrystorów i diod a ze względu na swoją prostotę została wybrana do formułowania równań układu. Rozważono również zastosowanie metody hybrydowej, która w pewnych przypadkach wykazuje szereg zalet, ale odrzucono ją ze względu na znacznie dłuższy czas formułowania równań równowagi, gdyż zasadniczym celem było opracowanie programu o możliwie największej szybkości obliczeń.

Poprzez zastosowanie prostego modelu RLE [43, 45, 46, 47] tyrystora, wraz ze specjalnym sposobem opisu jego sterowania, jest on - z punktu widzenia obwodu - elementem dwuzaciskowym. Ma to zasadniczy wpływ na szybkość obliczeń, ponieważ w ich trakcie topologia układu nie zmienia się i nie trzeba modyfikować raz ułożonych równań [28].

W celu umożliwienia użytkownikowi deklarowania a priori minimalnej wystarczającej wielkości pamięci i kompilowania programu o rozmiarach odpowiednich do wielkości analizowanego układu, wprowadzono specjalną kartę sterującą VARI. Pozwala ona na dynamiczną rezerwację pamięci komputera. Ponieważ pamięć przeznaczona na kod oraz bufor





Rys. 5.1. Schemat blokowy programu PAUT2



wej/wyj są stałe, to można dynamicznie rezerwować obszar zajmowany przez tablice danych i tablice posiłkowe. Użytkownik może zadeklarować następujące parametry: maksymalną liczbę elementów, węzłów, cykli, funkcji, transformatorów, źródeł oraz wartości y i współczynników do analizy Fouriera. Standardowe wartości tych parametrów nie są wbudowane i brak ich deklaracji spowoduje zatrzymanie się programu. Ma to na celu zmuszenie użytkownika do jak najoszczędniejszego gospodarowania pamięcią komputera, poprzez deklarowanie właściwych wartości tych parametrów dla danego układu.

5.1.1. Organizacja programu PAUT2

Program PAUT2 jest skomplikowany na dysku 000005 wraz z programem NAP2 i jest również programem nakładkowym [28, 29]. W zależności od wielkości wektora VE z jaką został on skomplikowany zajmuje 37696 słów (VE = 5000) lub 67712 słów (VE = 20 000) i umożliwia obliczanie układów o odpowiedniej w stosunku do zadeklarowanego wektora wielkości. Przyjęcie tego typu koncepcji sprawia, że maksymalna wielkość analizowanego układu jest ograniczona jedynie dostępną wielkością pamięci operacyjnej maszyny cyfrowej.

```
* CIRCUIT
VARI ...
: komentarz
RL1
TF45 /MF/ ...
:
CL44 ...
: komentarz
* TR * PLOT
* TIME ...
* PARA ...
* RUN
* * *
```

} karty opisu obwodu

} karta ządania analizy i wydruku

} karty parametryczne

Rys. 5.2. Struktura pliku danych programu PAUT2

Ogólny schemat blokowy programu przedstawiono na rysunku 5.1, a strukturę pliku danych wejściowych na rysunku 5.2.



Tak samo, jak w przypadku programu NAP2, dane są wprowadzane z kart perforowanych lub z symulującego je zbioru dyskowego. Każda karta zawiera jedno zdanie w języku wejściowym programu, z tym że jeżeli nie mieści się ono na jednej karcie (do 79 kolumny) należy użyć - jako ostatniego - znaku przeniesienia > i kontynuować je na następnej karcie. Znak przeniesienia musi być poprzedzony przynajmniej jedną spacją. Zdania komentarza rozpoczynają się od ":" i nie mogą być przenoszone.

5.1.2. Zakres zastosowań programu PAUT2

Program PAUT2 jest programem przeznaczonym do analizy układów tyrystorowych i pozwala na analizę układów składających się (oprócz tyrystorów) z liniowych i nieliniowych rezystancji, indukcyjności i pojemności, indukcyjności sprzężonych, przełączników, diod, różnych typów transformatorów oraz źródeł prądu i napięcia. Umożliwia on:

1) w zakresie analizy stałoprądowej *DC obliczenia quasi-stacjonarne dla ustalenia warunków początkowych symulacji (przy częstotliwości sieciowej),

2) w zakresie analizy stanu nieustalonego *TR
 - wyznaczenie odpowiedzi czasowych dla dużych sygnałów,
 - analizę Fouriera wybranych przebiegów czasowych.

Obliczenia prądów i napięć w funkcji czasu są wykonywane krok po kroku, przy użyciu jednej z trzech wbudowanych metod całkowania [28, 29, 33] - metody trapezów, Eulera wstecz oraz zmodyfikowanej Eulera wstecz.

Otrzymane przebiegi mogą być rysowane na drukarce wierszowej bądź też wyprowadzane w postaci tabelarycznej.

5.2. Opis wprowadzania danych do programu PAUT2

W celu ułatwienia opisu języka wprowadzimy jeszcze dodatkowo (do podanych w p. 4.2) następujące reguły [27, 28, 57]:

IWARTN - liczba typu całkowitego, zapis liczby taki, jak liczby typu INTERGER w FORTRANIE (maksymalnie 8 znaków),

RWARTN - liczba typu rzeczywistego, zapis liczby taki, jak liczby typu REAL w FORTRANIE (maksymalnie 8 znaków). Zakończenie liczby przez jedną z liter P, N, V, M, K, MEG powoduje pomnożenie jej odpowiednio przez 10^{-12} , 10^{-9} , 10^{-3} , 10^3 i 10^6 .



Węzły obwodu muszą być numerowane od 1 do n, bez pominięcia żadnego numeru. Typ elementu jest określony przez pierwsze dwie litery.

Przykład

* - rozkaz	*CIRCUIT, *RUN
RL - rezystor liniowy	RL1, RL5
LL - indukcyjność liniowa	LL15
CL - pojemność liniowa	CL30, CL1
RN - rezystor nieliniowy	RNLOAD
LN - indukcyjność nieliniowa	LN7
CN - pojemność nieliniowa	CNOBC
ML - sprzężenie pomiędzy liniowymi indukcyjnościami	ML1
TY - wbudowany model	TY1/PNPN/
PK - przełącznik	PK7
TF - transformator	TF1/MF/, TF5/00/

Do opisu nieliniowości, parametrów modeli wbudowanych oraz sterowania tyrystorami i zaworami służą:

- karty sterowania tyrystora	IGTY1
- karty sterowania przełącznikiem	STERPK7
- listy parametrów	TD, TR2
- funkcje	FZM, FL7

5.3. Zdania opisu obwodu i modele wbudowane

5.3.1. Elementy bierne

W programie PAUT2 mogą występować wszystkie typy elementów biernych. Zdanie opisujące rezystor liniowy jest postaci

RL _{xxx} nr węzła 1 nr węzła 2 wartość	E wartość [*YYY(TIME)] J wartość [*FAAA(TIME)] IZZZ wartości *I$\begin{Bmatrix} \text{RL} \\ \text{TY} \end{Bmatrix}$ YYY
---	---



gdzie: wartość - dopuszczalna jest tylko wartość stała,

FYYY, FAAA - nazwy funkcji źródeł,

IZZZ - nazwa źródła sterowanego,

$I \begin{Bmatrix} RL \\ TY \end{Bmatrix}$ - prąd sterujący.

Należy zaznaczyć, że:

1) nazwa elementu może zawierać dowolną liczbę znaków, lecz jest identyfikowana na podstawie pierwszych ośmiu;

2) dodatni kierunek prądu jest określony przez kolejność podania numerów węzłów; prąd płynie od węzła 1 do węzła 2 (rys. 5.3);

3) jeżeli wartość jest wartością elementu w jednostkach podstawowych, to trzeba opuścić nazwę jednostki; w przypadku użycia jednego z przyrostków, jednostka może być podana:

RL5 3 4 1000

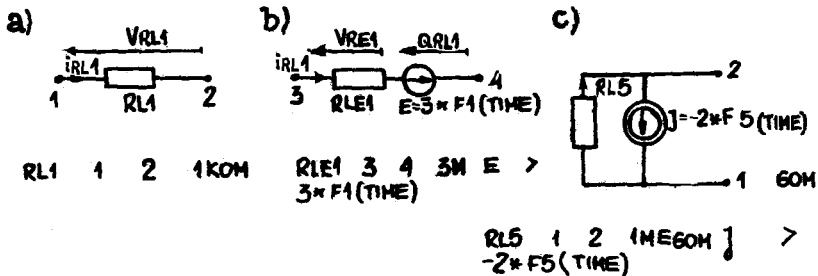
RL5 3 4 1KOM

4) do każdego elementu może być przyłą-

Rys. 5.3. Kierunki odniesienia dla elementów typu RL

czono tylko jedno źródło, którego dodatni kierunek jest dodatnim kierunkiem przepływu prądu w elemencie; dla źródła można określić kierunek przeciwny, poprzez podanie ujemnej wartości jego napięcia lub prądu,

5) karta elementu, którego prąd jest prądem sterującym, musi poprzedzać kartę elementu z przyłączonym źródłem sterowanym.



Rys. 5.4. Przykłady: a)-c) ilustrujące sposób wprowadzania elementów biernych typu RL do programu PAUT2

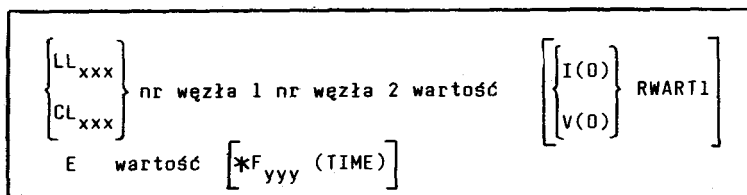


Kierunki odniesienia podano na rysunku 5.3, przy czym dopuszczalne jest oczywiście tylko jedno z tych źródeł jednocześnie.

Na rysunku 5.4 przedstawiono następujące przykłady ilustrujące sposób wprowadzania elementów biernych typu RL:

- rysunek 5.4a - rezystancja liniowa RL1 o wartości $1k\Omega$,
- rysunek 5.4b - rezystancja liniowa o wartości $3m\Omega$ z przyłączonym źródłem napięciowym o sile elektromotorycznej opisanej funkcją F1 i kierunku od węzła (3) do węzła (4) ($E = 3 * F1(TIME)$),
- rysunek 5.4c - rezystancja liniowa o wartości $1M\Omega$; kierunek przyłączonego do niej źródła prądowego, zależnego od czasu, jest przeciwny (znak "-") do kierunku prądu przyjętego dla elementu.

Zdanie opisujące liniową indukcyjność lub pojemność jest postaci



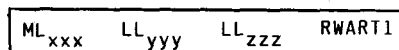
gdzie RWART1 - wartość prądu w indukcyjności lub napięcia na pojemności, w chwili $t=0$ (warunek początkowy).

Kierunki odniesienia dla elementów typu LL CL są analogiczne do przedstawionych na rysunku 5.3 kierunków odniesienia dla elementów RL.

Na rysunku 5.5 przedstawiono następujące przykłady ilustrujące sposób wprowadzania elementów biernych typu LL i CL:

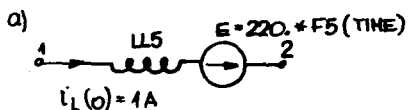
- rysunek 5.5a - indukcyjność liniowa LL5 o wartości $0.1H$ i prądem w chwili $t=0$ $1A$; jest do niej przyłączone niezależne źródło napięciowe opisane funkcją F5;
- rysunek 5.5b - liniowa pojemność CL7 o wartości $1\mu F$ i napięciem w chwili $t=0$ $100V$;
- rysunek 5.5c - szeregowe połączenie pojemności CL1 o wartości $1mF$ i indukcyjności LL1 o wartości $1\mu H$.

Dwie indukcyjności liniowe mogą być wzajemnie sprzężone i są opisane przy pomocy następującego zdania:

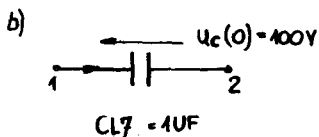


gdzie: ML_{xxx} - nazwa sprzężenia,
 LL_{yyy} , LL_{zzz} - indukcyjności sprzężone,
 $RWART1$ - wartość sprzężenia.

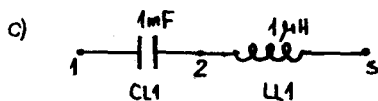
W przypadku, gdy prądy wpływają lub wypływają do początków cewek magnetycznie sprzężonych, to $RWART1$ jest większe od 0, w przeciwnym przypadku $RWART1 < 0$.



LL5 1 2 0.1 I(0) 1 E 220.*F5 (TIME)



CL7 1 2 1UF V(0) 100V



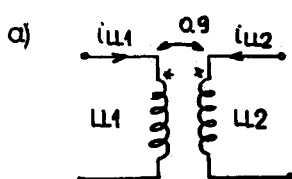
CL1 1 2 1MF
 LL1 2 3 1UH

Rys. 5.5. Przykłady a)-c) ilustrujące sposób wprowadzania elementów biernych typu LL i CL do programu PAUT2

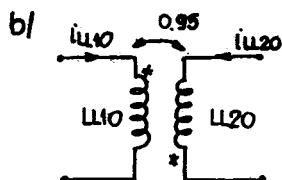
Na rysunku 5.6 przedstawiono dwa przykłady wprowadzania cewek magnetycznie sprzężonych.

Bierne elementy nieliniowe wprowadzane do programu PAUT2, dla odróżnienia ich od elementów liniowych, jako drugą literę identyfikującą nazwę mają literę N.





ML12 LL1 LL2 0.9



ML1020 LL10 LL20 -0.95

Rys. 5.6. Przykłady a), b) ilustrujące sposób wprowadzania cewek magnetycznie sprzężonych do programu PAUT2

Zdanie opisujące rezystor nieliniowy jest postaci

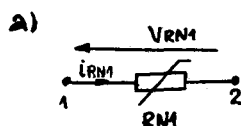
$$RN_{xxx} \text{ nr wężła 1 nr wężła 2 wartość } * F_{yyy} (IRN_{xxx}) >$$

$$\left[\begin{array}{l} E \text{ wartość } [* F_{zzz} (TIME)] \\ IAAA \text{ wartość } * I \left\{ \begin{array}{l} RL \\ TY \end{array} \right\} BBB \end{array} \right]$$

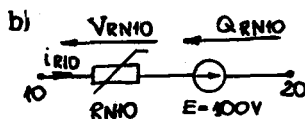
gdzie: F_{yyy} - nazwa funkcji nieokresowej typu (TAB2),

F_{zzz} - nazwa funkcji typu (COS),

$IAAA$ - nazwa źródła sterowanego.



RN1 1 2 1*F32(IRN1)



RN10 10 20 5*F10(IRN10) E 100

Rys. 5.7. Przykłady a), b) ilustrujące sposób wprowadzania elementów biernych typu RN do programu PAUT2

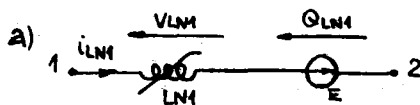


Na rysunku 5.7 przedstawiono następujące przykłady ilustrujące sposób wprowadzania elementów biernych typu RN:

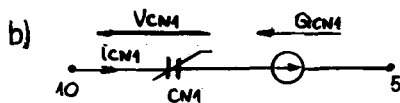
- rysunek 5.7a - rezystancja nieliniowa RN1 pomiędzy węzłami (1) i (2) ma wartość zależną od płynącego przez nią prądu zgodnie z funkcją F32;
- rysunek 5.7b - do rezystancji nieliniowej RN10 opisanej funkcją F10 jest przyłączone źródło napięcia o sile elektromotorycznej 100 V.

Zdanie opisujące nieliniową indukcyjność lub pojemność jest postaci

$$\left\{ \begin{array}{l} \text{LN}_{xxx} \\ \text{CN}_{xxx} \end{array} \right\} \text{ nr wężia 1 nr wężia 2 wartość } *F_{yyy} \left(I \left\{ \begin{array}{l} \text{LN}_{xxx} \\ \text{CN}_{xxx} \end{array} \right\} \right) > \\ [E \text{ wartość } [*F_{zzz}(\text{TIME})]]$$



LN1 1 2 3.1 * FN1 (ILN1) E 380V



CN1 10 5 1 * FC1 (ICN1) E 3 * FSTER (TIME)

Rys. 5.8. Przykłady a), b) ilustrujące sposób wprowadzania elementów biernych typu LN i CN do programu PAUT2

Na rysunku 5.8 przedstawiono następujące przykłady ilustrujące sposób wprowadzania elementów biernych typu LN i CN:

- rysunek 5.8a - nieliniowa indukcyjność LN1 jest włączona pomiędzy węzeł (1) i (2) i ma wartość zależną od płynącego przez nią prądu (zgodnie z funkcją FN1 * 3.1); do indukcyjności jest dołączone źródło napięcia o sile elektromotorycznej 380 V;



- rysunek 5.8b - nieliniowa pojemność CN1 jest włączona pomiędzy węzły (10) i (5) i ma wartość zależną od płynącego przez nią prądu (zgodnie z funkcją FC1); do elementu jest przyłączone źródło napięciowe zależne od czasu, opisane funkcją FSTER (pomnożoną przez 3).

5.3.2. Źródła

W programie PAUT2 można wyróżnić następujące trzy typy źródeł:

- 1) niezależne źródła napięciowe,
- 2) źródła napięciowe zależne od czasu,
- 3) niezależne źródła prądowe,
- 4) źródła prądowe zależne od czasu,
- 5) źródła prądowe sterowane prądem.

Źródła mogą być deklarowane wraz z elementami biernymi, przełącznikami PK lub elementami półprzewodnikowymi TY (dioda lub tyrystor).

W tablicy 5.1 przedstawiono dopuszczalne sposoby deklarowania źródeł.

Tablica 5.1

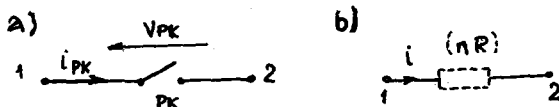
Typ źródła	Możliwość deklarowania z elementem typu:								Uwagi
	RL	RN	LL	LN	CL	CN	PK	TY	
1	+	+	+	+	+	+	+	+	-
2 *	+	+	+	+	+	+	+	+	opisane funkcją typu /COS/
2 *	+	-	+	-	+	-	-	-	opisane funkcją typu /TAB2/
3	+	+	-	-	-	-	-	-	-
4	+	-	-	-	-	-	-	-	-
5 *	+	+	-	-	-	-	-	-	prąd sterujący musi być prądem jednego z elementów RL, LL, CL, RN, CN, LN, TY, PK

Przykłady wprowadzania źródeł zostały podane w p. 5.3.1, przy okazji omawiania elementów biernych.



5.3.3. Przełączniki

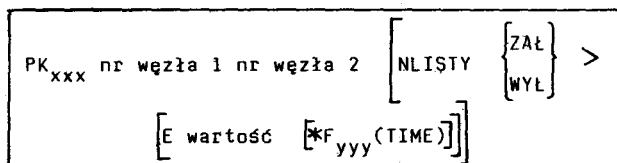
Przełącznik wraz z jego schematem zastępczym przedstawiono na rysunku 5.9.



Rys.5.9. Model przełącznika a) idealny, b) schemat zastępczy

Rozpoznawany jest on na podstawie pierwszych dwóch liter nazwy PK. Zostaje on następnie zastąpiony przez rezystor, o małej wartości rezystancji r dla stanu przewodzenia i dużej R dla stanu rozwarcia. Dla każdego przełącznika na osobnej karcie określa się sposób jego sterowania.

Zdanie opisujące przełącznik ma postać następującą:



gdzie: NLISTY - nazwa listy parametrów opisującej przełącznik,

ZAŁ - przełącznik zamknięty w chwili $t=0$,

WYL - przełącznik otwarty w chwili $t=0$.

W przypadku pominięcia nazwy listy parametrów przełącznika, należy pominąć również opis jego stanu w chwili $t=0$; wówczas program określi stan przełącznika jako zamknięty i przyjmie wbudowane wartości jego parametrów:

- rezystancja zamkniętego przełącznika

$$r = RZ = 0.01 \Omega ,$$

- rezystancja otwartego przełącznika

$$R = RW = 10^6 \Omega .$$



Jedną listę można opisać kilka przełączników i wówczas musi ona wystąpić za kartą ostatniego z nich.

Karta opisująca listę parametrów przełącznika jest postaci

NLISTY	RZ	wartość	RW	wartość
--------	----	---------	----	---------

gdzie nazwa listy - nie może zaczynać się od liter określających typ elementu (RL, RN ...), a także od litery F i znaku *; kolejność parametrów w liście jest dowolna.

Na rysunku 5.10 przedstawiono następujące przykłady ilustrujące sposób wprowadzania przełącznika:

- rysunek 5.10a - przełącznik PK1 jest włączony pomiędzy węzłami 1 i 2 w chwili $t=0$ jest zamknięty, rezystancja zamkniętego klucza wynosi 0.01Ω , a otwartego $10^6 \Omega$; do przełącznika jest dołączone źródło napięciowe o sile elektromotorycznej 100 V;

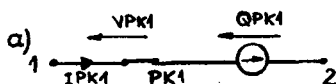
- rysunek 5.10b - przełącznik PK8 jest włączony pomiędzy węzłami 3 i 4; w chwili $t=0$ jest otwarty; lista parametrów LPK określa rezystancję zamkniętego klucza na 0.1Ω , a otwartego na $1 \text{ k}\Omega$.

W celu sterowania pracą przełącznika należy posłużyć się kartą sterowania o postaci

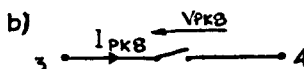
STERPK _{xxx}	[UP U IWART1]	U RWART1	U RWART2
-----------------------	---------------	----------	----------

gdzie: IWART1 = K, jeżeli okres rozpoczyna się w chwili t_k ,
RWART1, RWART2 ... - chwile czasowe określające zmianę stanu przełącznika.

Należy zaznaczyć, że karta sterowania przełącznika musi wystąpić za kartą przełącznika i można określić w niej maksymalnie 36 chwil czasowych. W przypadku podania parametru IWART1 i sterowania okresowego, ostatnia określona w karcie sterowania chwila czasowa jest końcem okresu.



PK1	1	2	E	100
-----	---	---	---	-----



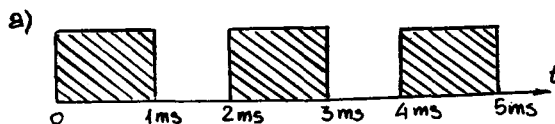
PK8	3	4	LPK	WYK
LPK	RZ	Q1	RW	1kom

Rys. 5.10. Przykłady a), b) ilustrujące sposób wprowadzania przełącznika w programie PAUT2



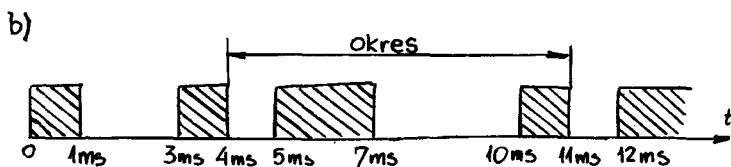
Na rysunku 5.11 przedstawiono następujące przykłady ilustrujące sposób sterowania przełącznika:

- rysunek 5.11a - jeżeli w chwili $t=0$ przełącznik był zamknięty, to zakreskowane obszary wskazują na przedziały czasowe zamknięcia przełącznika, a jeżeli w chwili $t=0$ był otwarty, to zakreskowane obszary wskazują na przedziały czasowe otwarcia przełącznika; obie postacie instrukcji sterowania są równoważne;
- rysunek 5.11b - okres rozpoczyna się w chwili $t = 4 \text{ MS}$ (bo $P = 3$).



STERPK1 1MS 2MS 3MS 4MS 5MS
lub

STERPK1 P 1 1MS 2MS



STERPK2 P 3 1MS 3MS 4MS 5MS 7MS 10MS 11MS

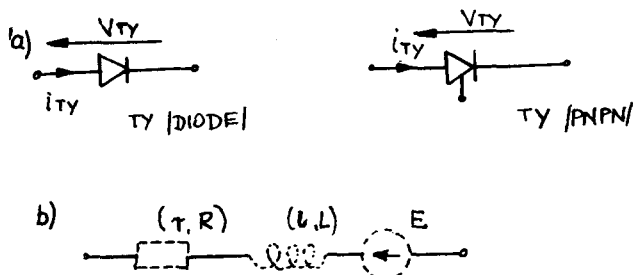
Rys. 5.11 Przykłady: a), b) ilustrujące sposób sterowania przełącznika

5.3.4. Diody i tyrystory

Zawór półprzewodnikowy (dioda lub tyrystor), wraz z jego schematem zastępczym, przedstawiono na rysunku 5.12. Jest on rozpoznawany na podstawie pierwszych dwóch liter TY i określenie typu zaworu /PNPN/ lub /DIODE/. Zostaje on następnie zastąpiony obwodem równoważnym (rys. 5.12b). W stanie przewodzenia rezystancja r i indukcyjność l są



małe, a źródło napięciowe E jest skierowane przeciwnie do przyjętego kierunku prądu [45, 53]. W stanie blokowania lub zaporowym rezystancja R jest duża ($R = 10^8 \text{ } \Omega$), a wartość siły elektromotorycznej $E = 0$. Indukcyjność dla stanu zaworowego ($L = 1 \text{ WK}$) można zmienić poprzez podanie współczynnika WK .



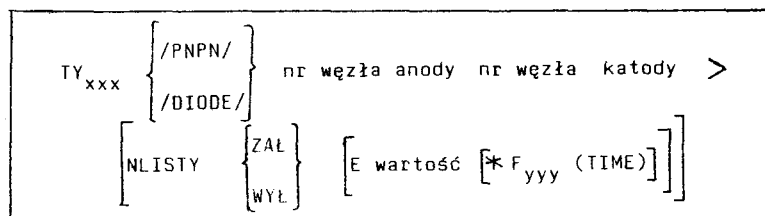
Rys. 5.12. Zawory półprzewodnikowe: a) dioda, b) tyrystor, c) schemat zastępczy

Tyrystor zaczyna przewodzić wtedy, gdy prąd płynący przez niego jest dodatni oraz podano sygnał sterujący na bramkę. Przez przyrząd płynie zawsze jakiś prąd, gdyż wartość rezystancji R nie może być nieskończenie duża

Dioda zaczyna przewodzić, gdy kierunek prądu zmienia się z ujemnego na dodatni.

Zawór przestaje przewodzić, gdy prąd zmienia kierunek z dodatniego na ujemny, za wyjątkiem przypadku, gdy jest określony prąd wsteczny - wtedy zawór przestaje przewodzić, gdy wartość prądu spadnie poniżej wartości prądu wstecznego. W chwili $t=0$ stan zaworu można określić jako przewodzenia lub zaporowy.

Zdanie opisujące zawór jest postaci



gdzie: /PNPN/, /DIODE/ - typ elementu - tyrystor lub dioda,
 NLISTY - nazwa listy parametrów,
 ZAŁ, WYŁ - stan zaworu w chwili $t=0$ przewodzenia lub zaporowy.

W przypadku pominięcia nazwy listy parametrów należy pominąć również opis stanu zaworu w chwili $t=0$; wówczas program określi stan zaworu jako zaporowy i przyjmie wbudowane wartości parametrów podane w tablicy 5.2.

Tablica 5.2

Parametr	Komentarz	Standard programowy
WK	Stosunek indukcyjności w kierunku zaporowym do indukcyjności w kierunku przewodzenia	1
TR	Rezystancja w kierunku przewodzenia	0.01Ω
TL	Indukcyjność w kierunku przewodzenia	10^{-9} H
UP	Napięcie na strukturze przewodzącej (wartość napięcia źródła E schematu zastępczego)	0
IW	Prąd wsteczny	0
$I\phi$	Prąd początkowy w indukcyjności schematu zastępczego zaworu	0

Jedną listą można opisać kilka zaworów i wówczas musi ona wystąpić za kartą ostatniego z nich.

Karta opisująca listę parametrów zaworu jest postaci

NLISTY	WK	wartość	TR	wartość	TL	wartość	>
	UP	wartość	IW	wartość	$I\phi$	wartość	

gdzie nazwa listy nie może zaczynać się od liter określających typ elementu, a także od litery F i znaku $\times$, kolejność parametrów w liście - jest dowolna.



Należy zaznaczyć, że w przypadku określenia typu zaworu jako /PNPN/ i nie podania jego karty sterowania zostanie on potraktowany jak dioda.

Przykłady

a)

TY1/PNPN/ 2 3 T100 WYL

T100 UT 0.9

b)

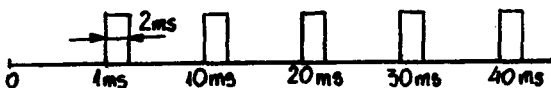
TY1/DIODE/ 7 8' BAY ZAŁ-

TY2/DIODE/ 9 10 BAY WYL

BAY TR 0.1 UP 0.8

W przykładzie a) tyrystor TY1 jest dołączony do węzła(2) - anoda i (3) - katoda i jest opisany listą T100. W chwili $t=0$ jest on w stanie zaporowym. Lista parametrów określa napięcie E schematu zastępczego w kierunku przewodzenia na 0.9 V. Pozostałe parametry mają wartości wbudowane; podane w tabelicy 5.2.

a)

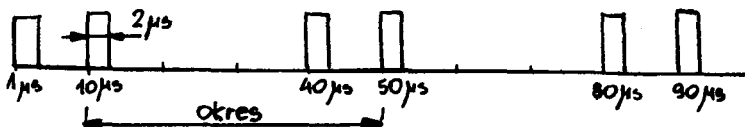


IGTY1 1M 2M 40M 2M 20M 2M 30M 2M 40 2M

lub

IGTY1 P 1 1M 2M 10M 2M

b)



IGTY2 P 2 1US 2US 40US 2US 40US 2US >

50US 2US

Rys. 5.13. Przykłady: a), b) ilustrujące sposób sterowania tyrystora



Do sterowania pracą tyrystora służy karta w postaci

$$\text{IGTY}_{xxx} [\cup \cup \text{I} \text{WART1}] \cup \text{R} \text{WART1} \cup \text{R} \text{WART11} \cup \text{R} \text{WART2} \cup \text{R} \text{WART22} \dots$$

gdzie:

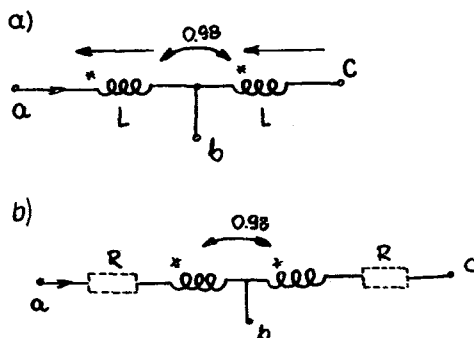
I	WART1	= k,	jeżeli okres zaczyna się w chwili t_k ,
R	WART1, R	WART11	- pary liczb określające parametry impulsu
R	WART2, R	WART22	bramki; pierwsza liczba określa chwilę,
:	:	:	w której impuls zostaje podany, a druga
:	:	:	- czas trwania impulsu.

Należy podkreślić, że karta sterowania zaworu musi wystąpić za kartą zaworu i można określić w niej maksymalnie 18 impulsów. W przypadku podania parametru I WART1 i sterowania okresowego ostatni z impulsów jest końcem okresu.

Na rysunku 5.13 przedstawiono dwa przykłady sterowania tyrystorem.

5.3.5. Transformatory

W celu maksymalnego uproszczenia obliczeń, w program są wbudowane modele różnych typów transformatorów, rozpoznawane na podstawie dwóch pierwszych liter nazwy TF i określenia typu.



Rys. 5.14. Transformator międzyfazowy:
a) idealny, b) z uwzględnieniem rezystancji



Na rysunku 5.14 przedstawiono transformator międzyfazowy (typ/ME/), który składa się z dwóch identycznych, wzajemnie sprzężonych cewek o współczynniku sprzężenia 0,98. Jego schemat zastępczy przedstawiono na rysunku 5.14b, przy czym rezystancję rozproszenia R można pominąć. Zdanie opisujące transformator międzyfazowy jest postaci

TF _{xxx} /MF/ węzeł a węzeł b węzeł c NLISTY

gdzie: węzeł a, c - numery węzłów skrajnych transformatora międzyfazowego (rys. 5.14),
 węzeł b - numer węzła środkowego (rys. 5.14),
 NLISTY - nazwa listy parametrów transformatora międzyfazowego.

Nazwa listy parametrów transformatora międzyfazowego nie może być pominięta. Jest ona postaci

NLISTY	TR	wartość	TL	wartość	I1	wartość	I3	wartość
--------	----	---------	----	---------	----	---------	----	---------

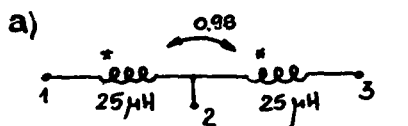
gdzie: TR - rezystancja rozproszenia schematu zastępczego (rys. 5.14b),
 TL - indukcyjność schematu zastępczego,
 I1 - prąd początkowy w indukcyjności dołączonej do węzła a,
 I3 - prąd początkowy w indukcyjności dołączonej do węzła c.

Reguły dotyczące nazwy listy są takie same jak dla zaworu i przełącznika. Należy podkreślić, że zawsze w liście parametrów musi być zadeklarowany parametr TL, a pozostałe, gdy nie zostaną podane, przyjmą swoje wbudowane wartości równe 0. Jedną listą można opisać kilka transformatorów międzyfazowych i wówczas karta listy parametrów musi wystąpić za kartą ostatniego transformatora opisanego tą listą.

Na rysunku 5.15 podano przykłady ilustrujące sposób deklarowania transformatora międzyfazowego w programie PAUT2. Rysunek 5.15a przedstawia transformator międzyfazowy włączony pomiędzy węzła 1, 2, 3 złożony z dwóch indukcyjności sprzężonych o wartościach 25 μH i rezystancjach rozproszenia 0,05 Ω ($k = 0,98$). Rysunek 5.15b przedstawia transformator międzyfazowy włączony pomiędzy węzły 7, 8 i 9 złożony z dwóch indukcyjności sprzężonych o wartościach 10 μH i rezystancji rozprosze-



nia 0Ω . Prąd początkowy w indukcyjności przyłączonej do węzła (7) wynosi 15 mA, a w indukcyjności przyłączonej do węzła (9) wynosi 20 mA.



TF1/MF/ 1 2 3 TM1
TM1 TL 25UH TR 0.05



TF3/MF/ 7 8 9 TM3
TM3 TL 10UH I1 15MA I3 20MA

Rys. 5.15. Przykłady: a), b) ilustrujące sposób wprowadzania transformatora międzyfazowego w programie PAUT2

Oprócz transformatora międzyfazowego są wbudowane w program cztery typy transformatorów jednofazowych. Transformatory trójfazowe można modelować poprzez układ złożony z trzech transformatorów jednofazowych.

Dostępne w programie typy transformatorów przedstawiono na rysunku 5.16. Transformator typu /00/ zostanie po wczytaniu zastąpiony schematem zastępczym przedstawionym na rysunku 5.17 [28, 29],

gdzie: P_S - moc strat,

$R_S = P_S / i_p^2$ straty omowe,

i_p - prąd uzwojenia pierwotnego w stanie jałowym,

$L_z = X_z / \omega_n$ - indukcyjność zwarcia,

X_z - reaktancja zwarcia,

$f_n = \omega_n / 2\pi$ - częstotliwość znamionowa,

$\sigma = n_1 / n_2$ - przekładnia transformatora,

$\beta_1 = 1/\sigma$,

$\beta_2 = \sigma$,



n_1 - liczba zwojów uzwojenia pierwotnego,

n_2 - liczba zwojów uzwojenia wtórnego,

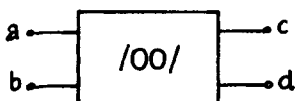
R_1 - wbudowana wartość $0,1 \Omega$,

$$R_{q1} = 10^6 * R_1,$$

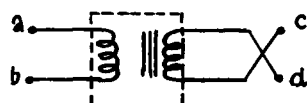
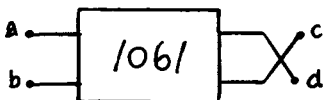
$$R_2 = R_1 \beta_1^2,$$

$$R_{q2} = 10^6 * R_2.$$

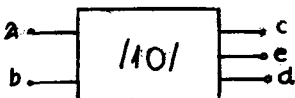
a)



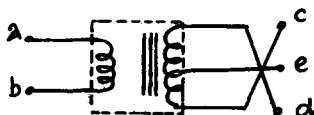
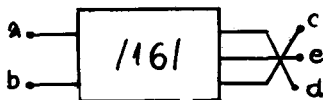
b)



c)



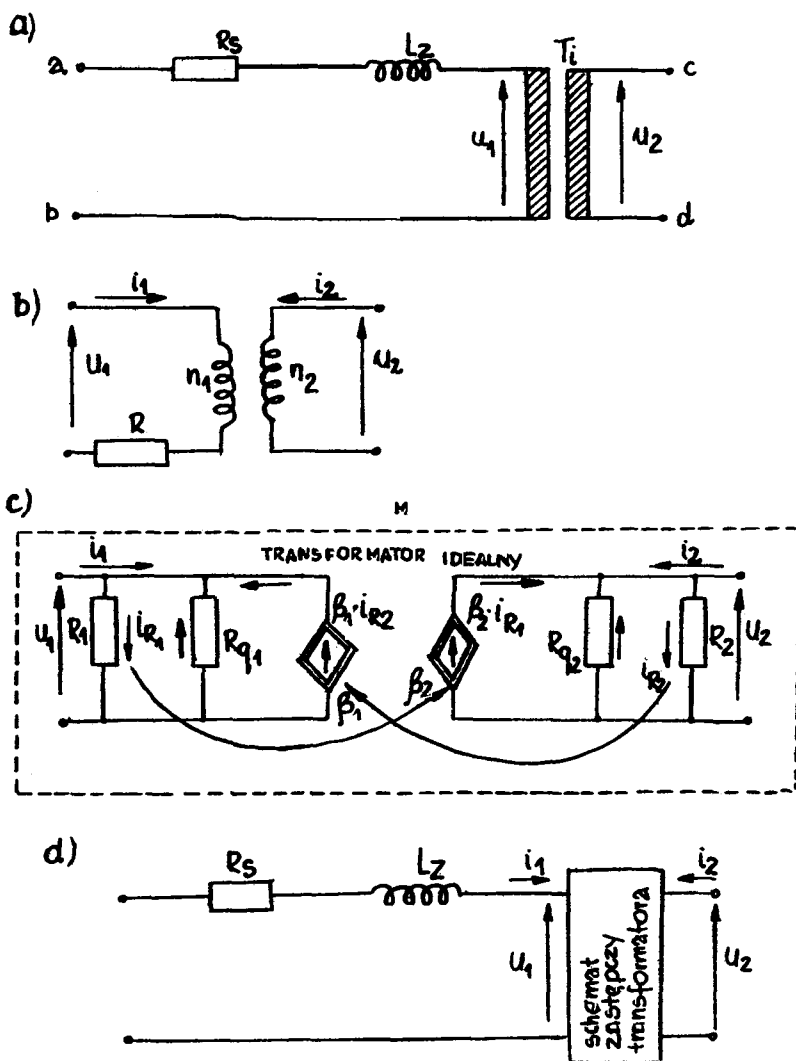
d)



Rys. 5.16. Typy transformatorów jednofazowych wbudowane w program PAUT2: a) transformator typu /00/, b) transformator typu /06/, c) transformator typu /10/, d) transformator typu /16/

Transformator typu /06/ zostanie zastąpiony analogicznym obwodem równoważnym z zamienionymi końcówkami wyjściowymi. Transformator typu /10/ po wczytaniu zostanie zastąpiony schematem zastępczym przedstawionym na rysunku 5.18 [28, 29],



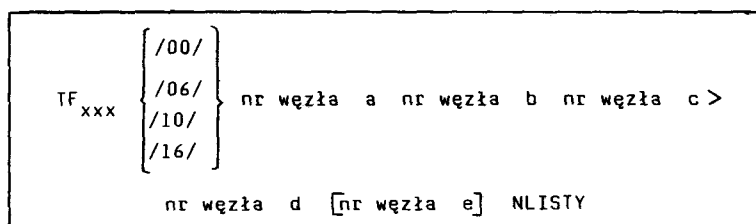


Rys. 5.17. Transformator typu /00/: a) schemat, b) idealny, c) schemat zastępczy transformatora idealnego, d) schemat zastępczy - postać końcowa



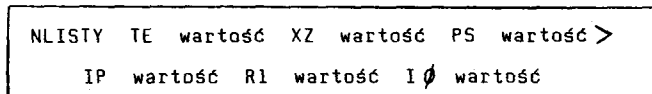
gdzie: R_S , X_Z - analogicznie jak dla transformatora typu /00/,
 $\nu = n_1/n_2$ - przekładnia transformatora,
 n_1 - liczba zwojów uzwojenia pierwotnego,
 n_2 - liczba zwojów górnej połówki uzwojenia wtórnego,
 n_3 - liczba zwojów dolnej połówki uzwojenia wtórnego ($n_2 = n_3$),
 $\beta_1 = 1/\nu$
 $\rho_2 = \rho$
 $R_1 = 0.1 \Omega$ (wartość wbudowana),
 $R_2 = R_3 = R_1 \cdot \beta_1^2$,
 $R_{Q2} = R_{Q3} = R_1/4 \rho^2$,
 $\tau_2 = \tau_3 = R_{Q2}/100 = R_{Q3}/100$.

Transformator typu /16/ zostanie zastąpiony analogicznym, obwodem równoważnym z zamienionymi końcówkami wyjściowymi. Zdanie opisujące transformator jest postaci



gdzie NLISTY - nazwa listy parametrów transformatora.

Karta opisująca listę parametrów transformatora jest postaci



gdzie: TE - przekładnia transformatora,

XZ - reaktancja zwarcia,

PS - moc strat,

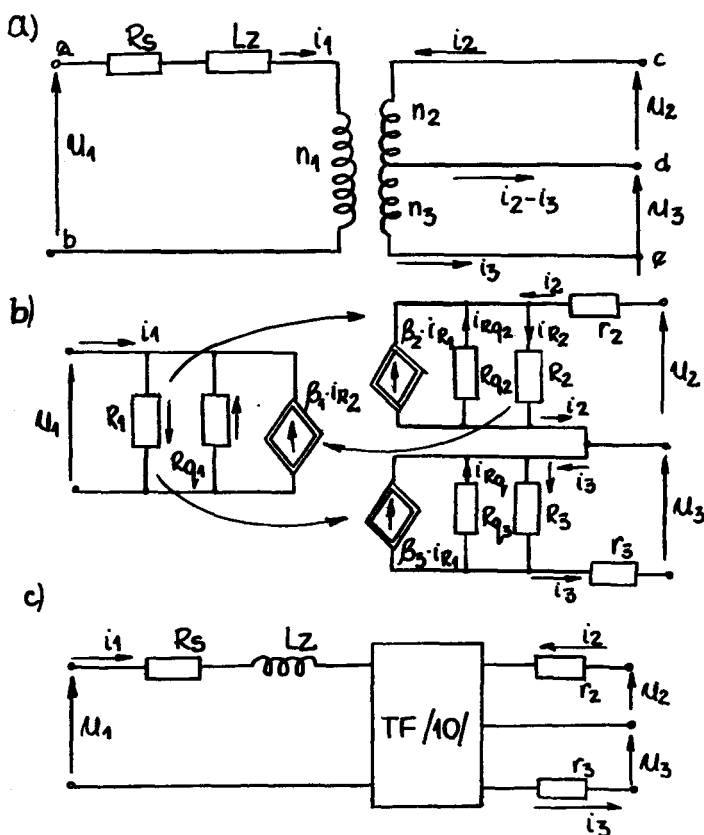
IP - prąd uzwojenia pierwotnego w stanie jałowym,

R1 - rezystancja (wbudowana wartość 0.1Ω),

I $\emptyset$ - prąd początkowy płynący przez reaktancję zwarciovą.



W karcie transformatora zawsze musi być podana nazwa opisującej go listy. W liście parametrów zawsze muszą być określone wartości parametrów TE oraz XZ, natomiast pozostałe (oprócz R_1), w przypadku braku deklaracji, zostaną przyjęte jako równe zero. Jedną listą można opisać kilka transformatorów i wówczas karta listy parametrów musi wystąpić za kartą ostatniego transformatora opisanego tą listą.

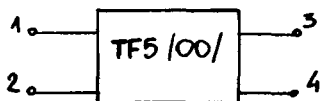


Rys. 5.18. Transformator typu 11Q : a) schemat, b) schemat zastępczy transformatora idealnego, c) schemat zastępczy - postać końcowa



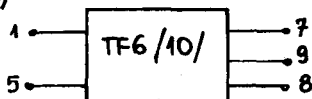
Bliższe szczegóły dotyczące sposobu otrzymania schematów zastępczych transformatorów można znaleźć w pracach [27, 28, 29, 53].

a)



TF5 /00/ 1 2 3 4 LTF5
LTF5 TE 20 KZ 30UH PS 0.01 IP 0.01 R1 1 >
10 0

b)



TF6 /10/ 1 5 7 8 9 LTF10
LTF10 TE 10 KZ 30UH

Rys. 5.19. Przykłady a), b) ilustrujące sposób deklarowania transformatorów w programie PAUT2

Na rysunku 5.19 podano przykłady ilustrujące sposób deklarowania transformatorów w programie PAUT2. Transformator TF5 (rys. 5.19a) jest typu /00/ i jest włączony pomiędzy węzły (1), (2), (3), (4) - odpowiednio a, b, c, d. Przekładnia transformatora wynosi 20, reaktancja zwarcia 30 μH , moc strat 0,01 W, prąd uzwojenia pierwotnego w stanie jałowym 0,01 A, rezystancja schematu zastępczego $R1 = 1\Omega$ i prąd początkowy przez reaktancję zwarcia 0 A. Transformator TF6 (rys. 5.19b) jest typu /10/ i jest włączony pomiędzy węzły (1), (5), (7), (8), (9) - odpowiednio a, b, c; d, e. Jest to transformator bezstratny, o przekładni równej 10 i reaktancji zwarcia równej 30 μH .



5.4. Funkcje

W program PAUT2 są wbudowane dwa typy funkcji. Są to funkcja typu /COS/ oraz funkcja typu /TAB2/. Zdanie opisujące funkcję typu /COS/ jest postaci

$$F_{xxx} / \text{COS/} \quad \text{MA wartość} \quad \text{PH wartość} \quad \text{FR - wartość}$$

gdzie: F_{xxx} - nazwa funkcji,
 MA, PH, FR - parametry funkcji zdefiniowanej wyrażeniem (5.1)
 (wartość skuteczna, faza i częstotliwość).

$$F_{xxx}(X) = \sqrt{2} \text{ MA } \cos(2\pi \text{ FR } X + \text{PH}), \quad (5.1)$$

W deklaracji funkcji typu /COS/ wartość parametru PH (fazy) podaje się w stopniach. Funkcja typu /COS/ może opisywać tylko źródła napięciowe zależne od czasu. W szczególnym przypadku źródło to może być traktowane jako źródło w dziedzinie częstotliwości (w analizie *DC i *DCTR). Jedna funkcja typu /COS/ może opisywać kilka źródeł i wówczas karta funkcji musi wystąpić po karcie ostatniego elementu, w którym jest wywoływana.

Przykład 1

```
...E 1*FSIN(TIME)
```

```
FSIN/COS/ MA 220 FR 50 PH - 90
```

Te dwie karty opisują źródło napięciowe o sile elektromotorycznej $e(t)$, określonej zależnością

$$e(t) = 1 * \sqrt{2} * 220 * \sin(2\pi \cdot 50t).$$

Przykład 2

```
...E 10 * F51(TIME)
```

```
F51/COS/ PH 45 MA 10 FR 1000
```

Te dwie karty opisują źródło napięciowe o sile elektromotorycznej $e(t)$, określonej zależnością

$$e(t) = 10 * \sqrt{2} \cdot 10 \cos(2\pi \cdot 1000 + 45^\circ),$$

Funkcja typu /TAB2/ jest funkcją typu tablicowego, analogiczną do stosowanej w programie NAP2. Zdanie deklaracji funkcji typu /TAB2/ jest postaci



$$F_{xxx}/TAB2/ \quad [P \quad I\text{WART1}] \quad R\text{WART1} \quad R\text{WART11} \quad R\text{WART2}>$$

$$R\text{WART22} \dots$$

gdzie: F_{xxx} - nazwa funkcji,

$\left. \begin{array}{l} R\text{WART1}, R\text{WART11} \\ R\text{WART2}, R\text{WART22} \\ \vdots \quad \quad \quad \vdots \end{array} \right\}$ - kolejne pary współrzędnych punktów (x_k, y_k)
 (argument, wartość)

$I\text{WART1} = k$ jeżeli funkcja jest okresowa i okres zaczyna się w punkcie x_k .

Funkcja /TAB2/, wbudowana w program PAUT2, posiada pewne ograniczenia w stosunku do występującej w programie NAP2:

- 1) $R\text{WART1} = 0$ - definicja funkcji musi się zaczynać od chwili $t=0$,
- 2) żadna z wartości funkcji ($R\text{WART1}, R\text{WART2} \dots$) nie może być ujemna,
- 3) można podać minimum 3 i maksimum 12 par wartości ($R\text{WARTi}, R\text{WARTii}$).

Uwagi

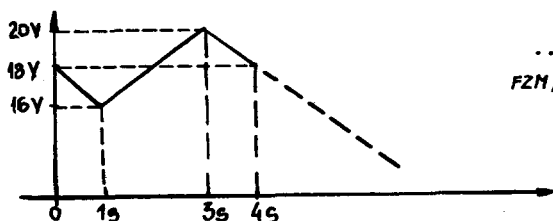
1. Funkcja jest ekstrapolowana przy użyciu liniowej ekstrapolacji dla ostatniego przedziału.
2. Dla funkcji okresowej ostatnia para współrzędnych określa koniec okresu, tak więc wartości funkcji y_k dla końca i początku okresu muszą być sobie równe.

Jedna funkcja typu /TAB2/ może opisywać kilka źródeł zależnych od czasu lub też kilka elementów nieliniowych i wówczas karta funkcji musi wystąpić po karcie ostatniego elementu, który ją wywołuje.

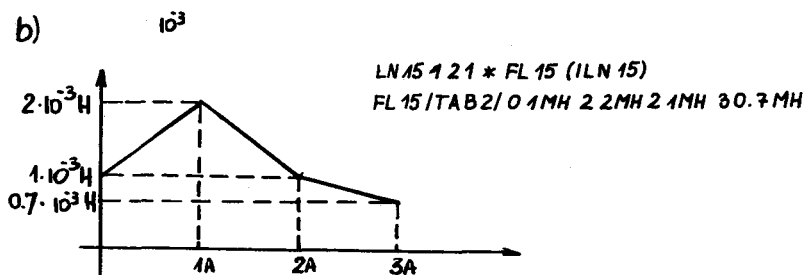
Na rysunku 5.20 podano szereg przykładów zastosowania funkcji typu /TAB2/. Rysunek 5.20a przedstawia deklarację źródła napięciowego zależnego od czasu zgodnie z funkcją FZM. Dla czasu większego od 4 s wykres funkcji będzie ekstrapolowany liniowo. Rysunek 5.20b przedstawia indukcyjność nieliniową LN15, której wartość zależy od płynącego przez nią prądu zgodnie z funkcją FL15. Na rysunku 5.20c przedstawiono okresowe źródło napięciowe zależne od czasu, opisane funkcją FZM.



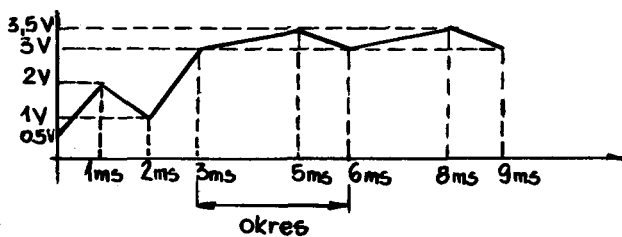
a)



b)



c)



... E 1 * FZM (TIME)

FZM/TAB2/ P 4 0 0.5 1MS 2 2MS 1 3MS 3 5MS 3.5 6MS 3

Rys. 5.20. Przykłady a)-c) zastosowania funkcji typu [TAB2] w programie PAUT2



5.5. Zdania sterujące

Zdania sterujące służą do wyboru rodzaju analizy oraz do sterowania rodzajem wyjścia i parametrami obliczeń. Podobnie jak w programie NAP2, zdania sterujące rozpoczynają się od gwiazdki (*) i mogą składać się tylko z pojedynczego rozkazu lub też zawierać listę parametrów.

5.5.1. Instrukcje wyboru rodzaju analizy i wyjścia

W celu wykonania analizy stanu ustalonego (quasi-stacjonarnego) należy użyć zdanie sterujące o postaci

*DC

Wyniki analizy DC wyprowadza rozkaz

* RUN 20

Analizę stanu przejściowego (*TR) lub też stanu przejściowego z automatycznym określeniem warunków początkowych (*DCTR) inicjuje się zdaniem o postaci

$\left\{ \begin{array}{l} *TR \\ *DCTR \end{array} \right\}$	$\left\{ \begin{array}{l} [\text{lista wyjściowa 1}] \\ [\text{lista wyjściowa 2}] \end{array} \right\}$
--	--

gdzie: lista wyjściowa 1 - wyjście zwykłe,

lista wyjściowa 2 - wyjście selektywne.

Lista wyjściowa 1 jest postaci

* PRINT	$\left\{ \begin{array}{l} IALL \\ VALL \\ AALL \end{array} \right\}$
---------	--

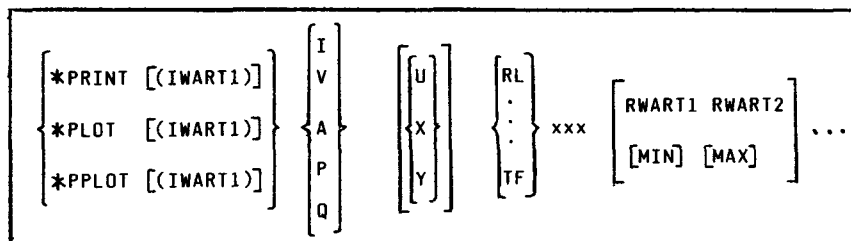
gdzie: IALL - drukowanie wszystkich prądów elementów,

VALL - drukowanie wszystkich napięć na elementach,

AALL - drukowanie wszystkich napięć i prądów elementów.



Lista wyjściowa 2 jest postaci



gdzie:

- I - żądanie wydruku prądu elementu xxx,
- V - żądanie wydruku napięcia na elemencie xxx,
- A - żądanie wydruku napięcia i prądu, elementu xxx,
- P - żądanie wydruku prądu i całki kwadratu prądu elementu; patrz [28, 29],
- Q - żądanie wydruku funkcji źródła dołączonego do elementu,
- U - strona pierwotna transformatora,
- X - górne uzwojenie strony wtórnej transformatora,
- Y - dolne uzwojenie strony wtórnej transformatora,
- IWART1 - liczba punktów do analizy przejściowej,
- *PRINT - drukowanie w formie tabeli żądanej opcji wyjściowej,
- *PLOT - wykres na drukarce wierszowej,
- *PPLOT - drukowanie w formie tabeli i wykres na drukarce wierszowej,
- RWART1, RWART2 - do instrukcji *PLOT i *PPLOT podaje minimum i maksimum skali wykresu,
- MIN, MAX - minimum skali zostanie przyjęte jako równe najmniejszej z wszystkich wartości opcji wyjściowej, a maksimum skali jako największej z wszystkich wartości opcji wyjściowej.

Uwagi

1. Przy braku deklaracji RWART1, RWART2, MIN i MAX skale wykresów zostaną dobrane automatycznie.
2. Można określić maksymalnie 18 funkcji zależnych od czasu dla wyjścia selektywnego w postaci tablicy i 5 w postaci wykresu.
3. Nie wolno dla tego samego elementu dwa razy żądać tego samego wydruku (na przykład ... PR5 IR5).



4. Przy nieautomatycznym doborze skali wykresów (użycie parametrów RWART1, RWART2) należy zachować następującą kolejność opcji wyjściowych:

a) prądy i napięcia na elementach w kolejności wczytywania elementów,
b) funkcje źródeł przyłączonych do elementów w kolejności wczytywania elementów,

c) prądy i napięcia na transformatorach w kolejności wczytywania transformatorów.

5. Należy pamiętać, że używanie parametru MIN i MAX ma sens tylko wtedy, gdy żąda się wykresów opcji tego samego typu (np. napięć lub prądów).

Przykład 1

*TR *PLOT (50) IRL0AD

Wykonaj analizę procesu przejściowego w 50 punktach (niezależnie od długości przedziału czasowego podanego w karcie *TIME). Wydrukuj i wykreśl, w dobranej automatycznie skali, wartość prądu IRL0AD.

Przykład 2

*OCTR *PLOT VRL10 VTY21 MIN MAX

Wykonaj analizę procesu przejściowego po uprzednim wyznaczeniu warunków początkowych w przedziale czasu podanym w karcie *TIME. Wydrukuj i wykreśl napięcie na rezystancji RL10 i tyrystorze TY21, przy czym przyjmij następującą skalę wykresów:

$\min = \min(VRL10, VTY21),$

$\max = \max(VRL10, VTY21).$

Przykład 3

*TR *PRINT VRLOAD *PLOT IRL1 IRL2 VRL2

Wykonaj analizę procesu przejściowego. Wydrukuj w formie tabeli napięcie na elemencie RLOAD, a w formie wykresu prąd elementów RL1 i RL2 oraz napięcie na elemencie RL2. Skalę wykresu dobierz automatycznie.

Przykład 4

*TR *PLOT IRL1 -10 10 VRL1 -100 1000 >

QRL1 -200 200 UXTF1 -300 300

Wykonaj analizę stanu przejściowego. Wydrukuj w formie wykresu prąd elementu RL1 w skali od -10 A do 10 A, napięcie na elemencie RL1



w skali od -100 V do 1000 V i funkcję źródła przyłączonego do elementu RL1 w skali od -200 do 200, oraz napięcie uzwojenia pierwotnego transformatora TF1 w skali od -300 V do 300 V.

5.5.2. Analiza Fouriera

Analizę Fouriera można wykonać w ramach analizy stanu przejściowego. Inicjuje się ją następującym rozkazem:

$$*F(RWART1 \quad RWART2) \quad [-] \quad IWART1) \quad \left\{ \begin{matrix} V \\ X \\ I \end{matrix} \right\} \quad \left[\begin{matrix} U \\ X \\ Y \end{matrix} \right] \quad \left\{ \begin{matrix} RL \\ \vdots \\ TF \end{matrix} \right\} \quad xxx \quad \left\{ \begin{matrix} V \\ I \end{matrix} \right\} \quad \dots$$

gdzie: RWART1 - początek okresu przyjętego w analizie Fouriera,

RWART2 - koniec okresu,

IWART1 - liczba liczonych par współczynników; w przypadku ządanania wykresu w skali logarytmicznej jest ona poprzedzona znakiem minus "-".

Należy pamiętać o pewnych ograniczeniach związanych z pojemnością pamięci maszyny cyfrowej. Są to:

1) w danym okresie nie może być więcej niż 400 wartości y,

2) liczba funkcji poddanych analizie Fouriera musi być mniejsza lub równa 5 (ze względu na maksymalną ilość wykreślanych zmiennych na drukarce wierszowej) i muszą one być podane jako wyjście selektywne *PPLOT a analizie *TR lub *DCTR,

3) w wyniku wykonania analizy Fouriera funkcji $f(t)$ otrzymamy współczynniki a_k , b_k , c_k i φ_k [rad] w zależności

$$f(t) = a_0 + a_1 \cos \omega t + a_2 \cos 2\omega t + \dots +$$

$$+ b_1 \sin \omega t + b_2 \sin 2\omega t \dots, \quad (5.2)$$

gdzie: $c_k = \sqrt{a_k^2 + b_k^2}$,

$$\varphi_k = \arctan \left(\frac{b_k}{a_k} \right)$$



oraz graficzny wykres w procentach wartości podstawowej każdej z harmonicznych. Poza tym zostanie również obliczona i wydrukowana wartość skuteczna analizowanej funkcji czasowej (p. 5.5.5).

Przykład 1

*F(256U 322U -10) VCLC1

Analiza Fouriera napięcia na pojemności CLC1 zostanie przeprowadzona za okres rozpoczynający się w 256 μ s i kończący się w 322 μ s. Zostanie obliczone 10 par współczynników i wykres na drukarce wierszowej zostanie wykonany w skali logarytmicznej.

Przykład 2

*F(0 1MS 25) VRL5 IRL15

Analiza Fouriera napięcia na rezystorze RL5 i prądu rezystora RL15 zostanie przeprowadzona za okres 0-1 ms. Zostanie obliczone 25 par współczynników, a wykres będzie wykonany w skali liniowej.

5.5.3. Instrukcja *TIME

Instrukcja *TIME służy do dokonania wyboru kroku oraz przedziału czasowego obliczeń w analizie *TR i *DCTR. Jest ona postaci

*TIME	t_1	T_1	t_2	T_2	t_3	T_3	[-]	T_4
-------	-------	-------	-------	-------	-------	-------	-------	-------

gdzie: t_1, t_2, t_3 - krok obliczeń w danym przedziale czasowym,

T_1, T_2, T_3 - przedział czasowy,

T_4 - koniec obliczeń (może być poprzedzony znakiem "-", jeżeli okres nie zaczyna się od $t=0$), ostatnia liczba oznacza zawsze koniec obliczeń.

Postać instrukcji *TIME wbudowanej w program PAUT2 pozwala na następujące możliwości doboru kroku i przedziału czasowego obliczeń:

1) przedział od $t = 0$ do $t = T_1$ ze stałym krokiem całkowania t_1 dla całego przedziału obliczeń (rys. 5.21a),

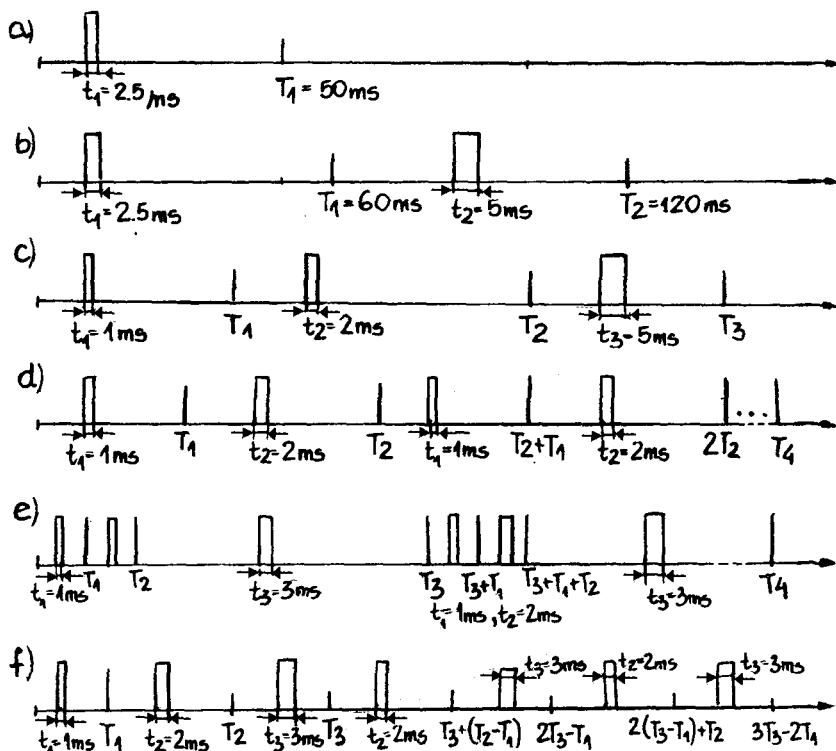
2) dwa przedziały z dwoma różnymi krokami całkowania (t_1 w przedziale $[0, T_1]$ i t_2 w przedziale $[T_1, T_2]$ - rysunek 5.21b,

3) trzy przedziały z trzema różnymi krokami całkowania: t_1 w przedziale $[0, T_1]$, t_2 w przedziale $[T_1, T_2]$ i t_3 w przedziale $[T_2, T_3]$



(rys. 5.21c), istnieje również możliwość, w przypadku użycia więcej niż jednego przedziału czasowego, zastosowania dużych kroków czasowych zmieniających się okresowo; należy wówczas rozróżnić dwa przypadki:

- gdy okres zaczyna się od $t = 0$,
- gdy okres zaczyna się od $t = T_1$;



Rys. 5.21. Przykłady a)-f) ilustrujące zastosowanie instrukcji *TIME

4) okres zaczyna się od $t = 0$ i mamy dwa przedziały czasowe z dwoma różnymi krokami całkowania. Czas zakończenia obliczeń jest określony przez T_4 (rys. 5.21d);

5) okres zaczyna się od $t = 0$ i mamy trzy przedziały czasowe z trzema różnymi krokami całkowania (rys. 5.21e);



6) okres zaczyna się od $t = T_1$; koniec przedziału czasowego należy podać ze znakiem ujemnym dla odróżnienia od przypadków, gdy okres zaczyna się od $t = 0$ (rys. 5.21f).

Na rysunku 5.21 przedstawiono szereg przykładów ilustrujących graficznie przedstawione powyżej 6 przypadków użycia instrukcji *TIME.

Przykład 1 (rys. 5.21a)

*TIME 25 MS 50 MS

Obliczenia zostaną wykonane z krokiem $t_1 = 25$ ms w przedziale czasu $[0, 50 \text{ ms}]$.

Przykład 2 (rys. 5.21b)

*TIME 2.5 MS 60 MS 5 MS 120 MS

Analiza zostanie wykonana dla czasu od 0 do 120 ms. W przedziale $[0, 60 \text{ ms}]$ krok t_1 wynosi 2.5 ms, a w przedziale (60 ms, 120 ms) krok t_2 wynosi 5 ms.

Przykład 3 (rys. 5.21c)

*TIME 1 MS 40 MS 2 MS 100 MS 5 MS 140 MS

Analiza zostanie wykonana dla czasu od 0 do 140 ms. W przedziale $[0, 40 \text{ ms}]$ krok t_1 wynosi 1 ms, w przedziale (40 ms, 100 ms) krok t_2 wynosi 2 ms i w przedziale (100 ms, 140 ms) krok t_3 wynosi 5 ms.

Przykład 4 (rys. 5.21d)

*TIME 1 MS 30 MS 2 MS 70 MS 150 MS

Analiza zostanie wykonana dla czasu od 0 do 150 ms. Przedziały czasowe i kroki całkowania będą następujące:

- $[0, 30 \text{ ms}]$ - krok 1 ms,
- $[30 \text{ ms}, 70 \text{ ms}]$ - krok 2 ms,
- $[70 \text{ ms}, 100 \text{ ms}]$ - krok 1 ms,
- $[100 \text{ ms}, 140 \text{ ms}]$ - krok 2 ms,
- $[140 \text{ ms}, 150 \text{ ms}]$ - krok 1 ms.

Przykład 5 (rys. 5.21e)

*TIME 1 MS 10 MS 2 MS 20 MS 3 MS 80 MS 150 MS

Analiza zostanie wykonana dla czasu od 0 do 150 ms. Przedziały czasowe i kroki całkowania będą w niej następujące:



[0, 10 ms] - krok 1 ms,
 [10 ms, 20 ms] - krok 2 ms,
 [20 ms, 80 ms] - krok 3 ms,
 [80 ms, 90 ms] - krok 1 ms,
 [90 ms, 110 ms] - krok 2 ms,
 [110 ms, 150 ms] - krok 3 ms.

Przykład 6

*TIME 1 MS 15 MS 2 MS 40 MS 3 MS 60 MS -150 MS

Analiza przeprowadzona zostanie dla czasu od 0 do 150 ms. Przedziały czasowe i kroki całkowania będą w niej następujące:

[0, 15 ms] - krok 1 ms,
 [15 ms, 40 ms] - krok 2 ms,
 [40 ms, 60 ms] - krok 3 ms,
 [60 ms, 85 ms] - krok 2 ms,
 [85 ms, 105 ms] - krok 3 ms,
 [105 ms, 130 ms] - krok 2 ms,
 [130 ms, 150 ms] - krok 3 ms.

5.5.4. Instrukcja *PARA

Instrukcja *PARA służy do określenia wartości parametrów używanych do obliczeń numerycznych. Jest ona postaci

*PARA	$\left\{ \begin{array}{l} \text{ZAW} \\ \text{AWE} \\ \text{MTR} \\ \text{MTF} \end{array} \right\}$	EM wartość FN wartość NS wartość WT >
		wartość LN wartość WP wartość

gdzie: $\left\{ \begin{array}{l} \text{ZAW} \\ \text{AWE} \\ \text{MTR} \\ \text{MTF} \end{array} \right\}$ - wybór jednej z wbudowanych w program metod całkowania numerycznego:

ZAW -- zmodyfikowany algorytm wsteczny Eulera,
 AWE - algorytm wsteczny Eluera,
 MTR - metoda trapezów,
 MTF - metoda trapezów z "filtrowaniem" [22, 28, 29];



- EM - maksymalny błąd względny (wartość wbudowana 0.01),
 LN - liczba podziału kroku całkowania (wartość wbudowana 10),
 NS - sterowanie przerwą w obliczeniach przy pracy interaktywnej; jeżeli:
 NS = 1 - przerwa następuje po wczytaniu danych,
 NS = 2 - po skompletowaniu i określeniu schematów zastępczych dla przełączników, zaworów i transformatorów międzyfazowych,
 NS = 3 - określeniu schematów zastępczych pozostałych transformatorów,
 NS = 4 - po przygotowaniu do obliczeń i wydruku,
 NS = 5 - przed wydrukiem graficznym;
 WT - żądanie wykresu dla systemu trójfazowego następuje dla
 WT = 1 (wartość wbudowana 0),
 WP - żądanie wydruku wyników pośrednich poczynwszy od wydruku WP (integer),
 FN - częstotliwość znamionowa.

Użycie karty * PARA jest w zasadzie opcjonalne. Jej brak spowoduje wybór do obliczeń zmodyfikowanego algorytmu wstecznego Eulera. Musi ona być użyta tylko wówczas, gdy w analizowanym obwodzie znajdują się transformatory, gdyż wtedy staje się niezbędne określenie wartości parametru FN.

Pewnego komentarza wymaga też sposób obliczania błędu względnego obliczeń. Przy obliczaniu układów z elementami nieliniowymi, prąd w elementach R, L, C będzie przyjęty jako równy i_k dla k-tego kroku. Jeżeli i_k jest założoną wartością prądu w chwili $t = t_k$, a i_k jest wartością rzeczywistą (otrzymaną w wyniku rozwiązywania układu równań), to błąd względny wynosi

$$\mathcal{E}' = \frac{|\bar{i}_k - i_k|}{\frac{|\bar{i}_k| + |i_{k-1}|}{2}}, \quad (5.3)$$

gdzie i_{k-1} jest wartością prądu z poprzedniego kroku czasowego.

Jeżeli $\mathcal{E}'$ jest większe od przyjętej w programie wartości $\mathcal{E}$ (EM), to ostatni obliczony prąd zostanie zaniedbany i obliczenia zostaną przeprowadzone ponownie z krokiem LN razy mniejszym od poprzedniego.



5.5.5. Instrukcja *RUN. Parametry sterujące
wydrukiem wyników pośrednich

Instrukcja *RUN powoduje przejście programu z fazy wczytywania danych do fazy obliczeń. Można w niej określić również żądania dotyczące wydruku wyników pośrednich. Zdanie *RUN jest postaci

*RUN [nr parametru sterującego wydrukiem pośrednim]...

W tablicy 5.3 podano numery parametrów sterujących wydrukami pośrednimi wraz z nazwą podprogramu, w którym są one realizowane.

Tablica 5.3

Numer sterujący	Komentarz	Podprogram, w którym jest realizowany wydruk	Uwagi
1	2	3	4
38	Wszystkie dane sterujące i parametry	CZYT 2	
10	Wszystkie wczytane dane elementów, źródeł i transformatorów przed uzupełnieniem sieci	ORG	
11	Jak wyżej - po uzupełnieniu sieci	ORG	
12	Wszystkie elementy i źródła w obwodzie po określeniu schematów zastępczych dla przełączników, zaworów i transformatorów jednofazowych	SZAST	
13	Wszystkie elementy i źródła w obwodzie, po określeniu schematów zastępczych dla transformatorów	STRAFO	
24	Wektor do sterowania obliczeń	LISTER	
39	Macierz IRIS dla wyjścia selektywnego i wektor do obliczeń kontrolnych	PWYD	
14	Macierz wzbudzeń ITREE	TOPOLO	
15	Wszystkie macierze przekształceń	TOPOLO	
16	Macierz jednostkowa i obwodowa	TOPOLO	



Tablica 5.3 (cd.)

1	2	3	4
17	Wartość impedancji (R , ωL , $-1/\omega C$, ωM) i macierz elementów	WART1	
18	Część rzeczywista i urojona macierzy impedancji dla obliczeń stanu ustalonego	MAOCZ	
19	Prądy obwodowe dla stanu stacjonarnego	WYNIK1	
20	Prądy elementów i napięcia na elementach	WYNIK2	
21	Warunki początkowe	WYNIK1	
22	Tablice przed i po przygotowaniu	PDTAB	
42	Wektory STER, KNOWN, TEXT	PSTER	
34	Wektory STER, macierz TABEL i wektor KNOWN	CZAS1	*
25	Numer, wartość, pochodna i sprzężenie wzajemne elementów	WART2	*
23	Współrzędne i pochodne	INTERP	*
26	Macierz obwodowa elementów R , L , C	MAIMP	*
27	Macierz obwodowa ze sprzężeniami wzajemnymi	MAIMP	*
28	Macierz obwodowa po odwróceniu	MAIMP	*
29	Prawa strona układu równań bez źródeł	ROZMAT	*
31	Prawa strona układu równań ze źródłami	ROZMAT	*
32	Nowe obliczone prądy obwodowe	WYNIK2	*
33	Napięcia i prądy w każdym elemencie	WYNIK2	*
35	Wektory STER, KNOWN i TEXT1	CZAS2	*
36	Macierz ze wszystkimi prądami i napięciami (10 kroków całkowania)	MAGAZ	
40	Funkcje do wykreślenia	WYJ1	
46	Funkcje do analizy Fouriera	WYJ1	
41	Funkcje do wykreślania	WYJ2	



Tablica 5.3 (cd.)

Numer sterujący	Komentarz	Podprogram, w którym jest realizowany wydruk	Uwagi
1	2	3	4
45	Funkcje do analizy Fouriera i tablice pomocnicze	WYJ2	
43	Funkcje do analizy Fouriera i 4 tablice pomocnicze	FOURIE	
44	Napięcia i prądy wszystkich elementów w ostatnim kroku całkowania	MAGAZ	
99	Wszystkie wartości pośrednie		
20	Wyjście analizy *DC		

Uwaga. * oznacza, że rozkaz wydruku pośredniego może być wykonany po określonym kroku całkowania (parametr WP w karcie *PARA).

W celu łatwiejszego posługiwania się wydrukami wyników wprowadzono następujące reguły:

1) elementy schematów zastępczych mają następujące oznaczenia

R - REZ, L - IND., E - SEM,

2) w wydrukach pośrednich obowiązują oznaczenia:

SD - źródła stałoprądowe,
SA - źródła typu /COS/,
GS - źródła sterowane,
SC - źródła zależne od czasu;

3) parametry funkcji /TAB2/ są gromadzone w tablicach RN, LN, CN - dla elementów nieliniowych i TS dla źródeł zależnych od czasu.

4) chwile czasowe z kart typu IG oraz STER są gromadzone w tablicach TT i TP.

W przypadku wykonywania analizy *DC często lepiej jest użyć rozkazu *RUN 20 13, zamiast *RUN 20. Pozwala to na określenie numerów elementów (numer sterujący 13), do których odwołuje się rozkaz 20.



5.5.6. Rozpoczęcie i zakończenie obliczeń

Podobnie jak w programie NAP2 rozpoczęcie obliczeń rozpoczyna się poprzez podanie rozkazu

*CIRCUIT

Rozkaz ten zeruje wewnętrzną bazę danych i przygotowuje program do wczytania danych. Po wczytaniu tej karty zostaje wydrukowany nagłówek programu oraz data i czas rozpoczęcia obliczeń.

W celu zakończenia obliczeń programu i usunięcia go z pamięci operacyjnej maszyny cyfrowej należy użyć zdania o postaci

* * * *

5.6. Dynamiczna rezerwacja pamięci

Do dynamicznej rezerwacji pamięci służy zdanie o postaci

VARI	ME	wartość	MW	wartość	MO	wartość	MZ	wartość	>
	MT	wartość	MF	wartość	MR	wartość	MC	wartość	

gdzie: ME - maksymalna liczba elementów,

MW - maksymalna liczba węzłów,

MO - maksymalna liczba cykli,

MZ - maksymalna liczba źródeł,

MT - maksymalna liczba transformatorów,

MF - maksymalna liczba funkcji,

MR - maksymalna liczba wartości y do analizy Fouriera,

MC - maksymalna liczba współczynników obliczanych w analizie Fouriera.

Karta VARI musi wystąpić jako druga w pliku danych wejściowych tuż za kartą *CIRCUIT. Pozwala ona na dynamiczną rezerwację (w ramach dostępnego rozmiaru wektora roboczego VE) obszaru pamięci przeznaczonego



na tablice gromadzące dane, wektory i macierze. Maksymalny rozmiar analizowanego obwodu jest oczywiście limitowany ustalonym na etapie kompilacji rozmiarem wektora roboczego VE. Zastosowanie karty VARI pozwala na zwiększenie ilości jednych parametrów obwodu kosztem innych (np. zwiększenie liczby węzłów kosztem zmniejszenia liczby elementów i transformatorów) oraz na wykonanie w ciągu paru minut kompilacji nowego programu z wektorem VE, zwiększonym do rozmiarów pozwalających analizować nawet największy obwód (pod warunkiem, że program zmieści się w pamięci maszyny cyfrowej).

W karcie VARI muszą wystąpić wszystkie parametry i żaden z nich nie może mieć wartości 0. Gdy dany parametr nie występuje w układzie (np. brak transformatorów), to jego wartość deklaruje się jako równą 1 ($MT = 1$).

Należy zwrócić uwagę, że parametry karty VARI określają maksymalne wartości obwodu, otrzymanego po zastąpieniu wbudowanych modeli elementów i transformatorów ich schematami zastępczymi. Należy więc pamiętać, że zadeklarowanie w opisie obwodu zaworu, przełącznika czy też transformatora zwiększa liczbę elementów obwodu zgodnie z tablicą 5.4.

Tablica 5.4

Zadeklarowany element	Zwiększenie liczby:			
	węzłów	elementów	obwodów	źródół
Zawór półprzewodnikowy	2	1	-	1
Przełącznik	-	1	-	-
Transformator międzyfazowy	2	3	-	-
Transformator typu /00/ lub /06/	2	5	2	3
Transformator typu /10/ lub /16/	4	7	3	3



5.7. Uwagi praktyczne

Wszystkie karty danych są wczytywane w programie PAUT2 w postaci znakowej, a następnie analizowana kolumna po kolumnie. Wszelkie błędy syntaktyczne są wykrywane na etapie wczytywania danych, a informacja o błędzie jest drukowana bezpośrednio po błędnym zdaniu. Jeżeli błąd wystąpi w karcie *CIRCUIT lub VARI, program zostanie usunięty z pamięci operacyjnej maszyny cyfrowej. Przy błędach w pozostałych kartach będą one czytane aż do instrukcji *RUN z tym, że wystąpienie błędu nie zezwala na dalszą analizę układu.

Szczególną uwagę należy zwrócić na zasadnicze różnice pomiędzy językiem programu PAUT2 i NAP2. Są to:

1) w programie PAUT2 należy numerować węzły kolejno od 1 do n, nie pomijając numeru żadnego węzła; w programie NAP2 można numerować węzły dowolnie, lecz musi być określony węzeł masy (0);

2) w programie PAUT2 nazwa listy elementu musi być umieszczona za kartą elementu, natomiast w programie NAP2 nazwa listy musi być zdefiniowana przed odwołaniem się do niej;

3) w programie PAUT2 rozpoznawanie typu elementu następuje na podstawie dwóch pierwszych liter (pozwala to na łatwe wyodrębnienie elementów nieliniowych), natomiast w programie NAP2 na podstawie pierwszej litery;

4) w programie PAUT2 karty opisu funkcji muszą być umieszczone po ostatniej instrukcji, w której była ona użyta, natomiast w programie NAP2 funkcja musi być zdefiniowana przed odwołaniem się do niej;

5) w programie PAUT2 do każdego elementu można przyłączyć tylko jedno źródło; w programie NAP2 można przyłączyć dwa źródła, lecz tylko do elementów typu R i G; źródła sterowane mają osobne karty opisu;

6) należy zwrócić uwagę na różne zdania opisu funkcji;

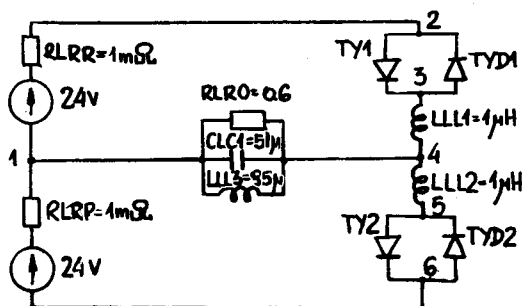
7) należy zwrócić uwagę na różnice w instrukcji *TIME.

Poza tymi zasadniczymi różnicami sama koncepcja opisu obwodu jest podobna i użytkownik znający i umiejący posługiwać się programem NAP2 nie powinien mieć większych problemów ze stosowaniem programu PAUT2.



5.8. Przykład zastosowania programu PAUT2

W celu zilustrowania sposobu przygotowania danych oraz możliwości analizy jakie daje program PAUT2, jest rozpatrywany start falownika szeregowo-równoległego przedstawionego na rysunku [50, 53]. Plik danych wejściowych służących do opisu i wykonania analizy tego układu przedstawiono na rysunku 5.23. Poniżej opiszemy w skrócie znaczenie każdej karty.



Rys. 5.22. Układ falownika szeregowo-równoległego

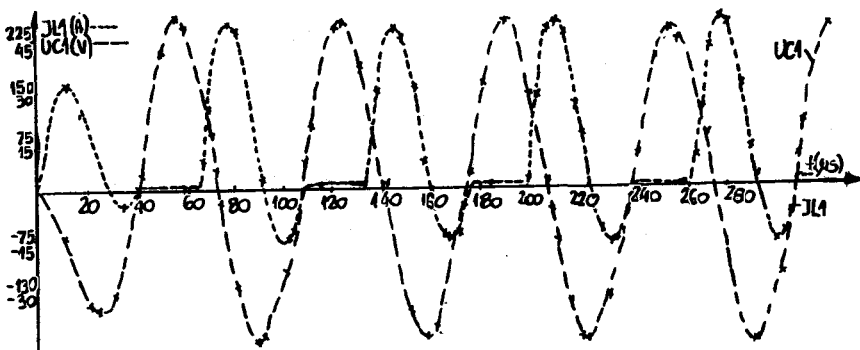
```
*CIRCUIT
VARI ME 25 MW 14 MO 10 MF 10 MZ 8 MT 1 MR 200 MC 50
; THE SERIES-PARALLEL INVERTER
RLRR 1 2 0.001 E 24
RLRP 6 1 0.001 E 24
RIR0 1 4 0.6
LLL1 3 4 1UH
LLL2 4 5 1UH
LLL3 1 4 95UH
CLC1 1 4 51UF
TY1/PNPN/ 2 3 T00300 WYE
TY2/PNPN/ 5 6 T00300 WYE
T00300 TR .001 TL 1E-8 UP 0.8
TYD1/DIODE/ 3 2 D00300 WYE
TYD2/DIODE/ 6 5 D00300 WYE
D00300 TR .001 UP 0.5
IGTY1 0 10U 66.6U 10U 133.3U 10U 200U 10U 266.6U 10U
IGTY2 33.3U 10U 100U 10U 166.6U 10U 233.3U 10U 300U 10U
*TIME 1US 330US
*TR *PLOT VCCL1 VTY1 VTY2 ILLL1 ILLL3
*F(256U 322U -10) VCCL1
*RUN
****
```

Rys. 5.23. Plik danych wejściowych do analizy układu z rysunku 5.22



Karta *CIRCUIT rozpoczyna symulację i przygotowuje program do czytania danych wejściowych. Karta VARI służy do dynamicznej rezerwacji pamięci komputera. Trzecia karta, z dwukropkiem ":" na początku, jest kartą komentarza i jej zawartość będzie drukowana w wydruku wyjściowym. Czwarta karta oznacza, że do rezystora liniowego RLRR o wartości $0,001\Omega$ dołączonego do węzłów(1) i(2) jest przyłączone źródło napięciowe o sile elektromotorycznej 24 V. Szósta karta oznacza, że liniowy rezystor RLRO jest przyłączony do węzłów(1) i(4) i ma wartość 0.6Ω . Następnie cztery karty służą do wczytania liniowych indukcyjności i liniowej pojemności CLC1. Jedenasta i dwunasta karta dotyczy wczytania tyrystorów TY1 i TY2 połączonych anodą do węzła(2) i(5) i katodą do węzła(3) i(6). Lista parametrów modelu tyrystora nosi nazwę T00300 i jest on wyłączony (WYL) w chwili $t = 0$.

Trzynasta karta określa parametry modelu tyrystora T00300. Czternasta i piętnasta karta służą do wczytania diod: TYD1 i TYD2, których listę parametrów opisuje szesnasta karta D00300. Siedemnasta i osiemnasta karta służą do sterowania tyrystorów TY1 i TY2. Dziewiętnasta karta podaje czas obliczeń stanu nieustalonego od 0 do 330 μs , z krokiem 1 μs . Dwudziesta karta podaje odpowiedzi obwodu, które mają być drukowane lub też wykreslane na drukarce wierszowej. Dwudziesta pierwsza karta podaje, które z przebiegów mają zostać poddane analizie Fouriera oraz określa początek i koniec okresu. Żąda ona obliczenia dziesięciu par współczynników i wykreślenie ich w skali logarytmicznej.



Rys. 5.24. Rezultaty symulacji startu falownika dla $f_{pulse} = 15 \text{ kHz}$



ZADEKLAROWANE PARAMETRY:

VARI ME 25 MW 14 MO 10 MF 10 MZ 8 MT 1 MR 200 MC 50
KOMENTARZ: THE SERIES-PA ALLEL INVERTER

RIIR 1 2 0.001 E 24
RIIR 6 1 0.001 E 24
RIIR 1 4 0.6
LLL1 3 4 1UH
LLL2 4 5 1UH
LLL3 1 4 95UH
CLC1 1 4 51UF
TY1/PNPN/ 2 3 T00300 WYŁ
TY2/PNPN/ 5 6 T00300 WYŁ
T00300 TR .001 TL 1E-8 UP 0.8
TYD1/DIODE/ 3 2 D00300 WYŁ
TYD2/DIODE/ 6 5 D00300 WYŁ
D00300 TR .001 UP 0.5
IGTY1 0 10U 66.6U 10U 133.3U 10U 200U 10U 266.6U 10U
IGTY2 33.3U 10U 100U 10U 166.6U 10U 233.3U 10U 300U 10U
*TIME 1US 330US
*TR *PILOT VCCL1 VTY1 VTY2 ILL1 ILL3
*F(256U 322U -10) VCCL1
*RUN

2

WSPÓŁCZYNNIKI FOURIERA FUNKCJI CZASOWEJ VCCL1

SKŁADOWA STAŁA WYNOŚI 8.0081E-06

WARTOŚĆ SKUTECZNA WYNOŚI 3.4188E 01

	COS	SIN	WARTOŚĆ	FAZA
1	3.7170E 01	-3.1418E 01	4.8670E 01	2.2725E 00
2	-4.5416E-01	-5.6173E-01	7.2236E-01	-2.4617E 00
3	-1.4781E 00	1.4968E 00	2.1037E 00	-7.7912E-01
4	4.0675E-02	4.4308E-02	6.0147E-02	7.4268E-01
5	-1.1248E-01	4.0943E-01	4.2459E-01	-2.6811E-01
6	4.8109E-02	4.0178E-03	4.8276E-02	1.4875E 00
7	-1.5904E-01	3.0320E-02	1.6191E-01	-1.3824E 00
8	-7.9714E-03	-4.9400E-02	5.0039E-02	-2.9816E 00
9	-1.1322E-01	-1.4555E-01	1.8440E-01	-2.4805E 00
10	-3.7179E-02	-3.3606E-03	3.7331E-02	-1.6609E 00

GRAFICZNE PRZEDSTAWIENIE WARTOŚCI
(W PROCENTACH WARTOŚCI PODSTAWOWEJ)

I 2 3 5 7 10 20 30 50 70 100
I-----
I
I
I
I
I*****
I****
I*****
I
I
I
I
I
I

** KONIEC ANALIZY FOURIERA **

DATA: 09/03/84 CZAS: 10/2

Rys. 5.25. Wyniki analizy Fouriera



Rozkaz * RUN przekazuje sterowanie do pakietu analizy i powoduje rozpoczęcie wykonywania obliczeń. Ostatnia karta **** kończy obliczenia i usuwa program z pamięci operacyjnej maszyny cyfrowej.

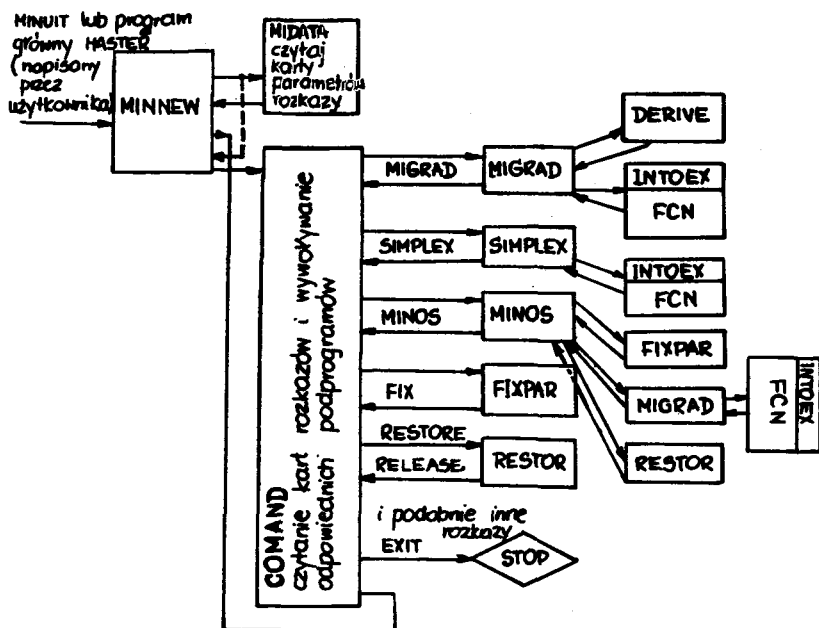
Na rysunku 5.24 przedstawiono rezultaty symulacji startu falownika przy pomocy wyżej omówionych danych. Wyniki analizy Fouriera zamieszczono na rysunku 5.25.



6. OPIS PROGRAMU MINUIT

6.1. Wprowadzenie

Program MINUIT został opracowany w CERN COMPUTER CENTER przez F. James'a i M. Roos'a w 1975 r. na maszynę cyfrową IBM (360-370) i CDC 6000-7000 [34, 38, 57]. Po pewnych przeróbkach został on przystosowany do pracy w systemie GEORGE 3 na maszynie cyfrowej Odra 1305 i jest od kilku lat wykorzystywany podczas zajęć z Wybranych Zagadnień Projektowania Komputerowego w Instytucie Elektroniki Pł.



Rys. 6.1 Schemat blokowy programu MINUIT



Zasadniczym celem zastosowania programu, a właściwie systemu programów MINUIT jest poszukiwanie minimum funkcji celu zadeklarowanej przez użytkownika w specjalnej procedurze FCN. Obecna, standardowa wersja programu pozwala na minimalizację funkcji 15 zmiennych, lecz istnieje również specjalna wersja tego programu - MINUITL, która pozwala na poszukiwanie minimum funkcji celu aż 55 zmiennych. Jednak bardzo zwiększa to pojemność pamięci maszyny cyfrowej oraz wydłuża znacznie czas obliczeń. Na rysunku 6.1 przedstawiono ogólny schemat blokowy programu.

Sterowanie przebiegiem obliczeń odbywa się w procedurze MINNEW oraz COMMAND. Procedura MINNEW wywołuje w odpowiednim porządku potrzebne procedury, jak również procedurę COMMAND, w której zgodnie z kartami rozkazów użytkownika są wywołane właściwe procedury optymalizacyjne.

Dokładny opis wszystkich procedur i funkcji programu wykracza poza ramy niniejszego skryptu i będzie przedmiotem następnego opracowania. W celu umożliwienia posługiwania się programem, użytkownikowi nie znającemu szczegółów budowy pakietu MINUIT, przedstawiono poniżej sposób wczytania kart danych i rozkazów oraz omówiono funkcje każdej z nich.

6.2. Karty danych

Plik kart danych powinien być czytany w następującej kolejności:

- 1) karta tytułowa,
- 2) karta definiująca parametry (zmienne do optymalizacji); jedna karta dla każdego parametru,
- 3) karta pusta,
- 4) karty czytane przez procedurę FCN,
- 5) karty rozkazów sterujące pracą MINUIT'a.

Tego typu zestaw kart tworzy jeden blok danych. Przy jednym wykonaniu programu można liczyć całą serię takich bloków - jeden za drugim, oczywiście dla tej samej funkcji celu FCN.

Karta tytułowa składa się z dowolnej sekwencji znaków alfanumerycznych i służy do nadania tytułu wydrukowi wyjściowemu.

Karta parametrów musi być napisana w następującym porządku:

- kol. 1 - 10 numer parametru, taki jak w procedurze FCN (≤ 30),



Zastosowanie danego rozkazu	Kolumna				
	1-10	11-20	21-30	31-40	
	ROZKAZ	ARG 1	ARG 2	ARG 3	...
Wywoływanie procedury FNC Czytanie macierzy kowariancji	CALL FNC COVARIANCE	IFLAG NPAR			
		tu następuje macierz elementów o formacie 7F10			
Rysowanie konturu FNC	CONTOUR	Nr parametru (oś X)	Nr parametru (oś Y)	Ilość konturów (w.w=2)	
Sygnał końca bloku danych	END	p. 6.3.			
Koniec bloku danych. Powrót do wywołującej procedury	END RETURN	p. 6.3.			
Ustala wartość UP do zdefiniowania błędu	ERROR DEF	wartość UP (w.W=1)			
Sygnał końca zadania	EXIT	p. 6.3.			
Ustalenie wartości parametru	FIX	numer parametru			
Wskazanie, że pochodne są obliczane przez FNC	GRADIENT	0-sprawdzenie 1-bez sprawdzenia			
Listowanie rozkazów MINUIT'a	HELP				
Obliczanie i drukowanie macierzy kowariancji	HESSE	Maksymalna ilość (w.w=1800)			



"Odskok" po znalezieniu minimum lokalnego w celu szukania innego	IMPROVE	Maks.ilość wywołań (w.w=1000)	Liczba prób (w.w = NPAR+2)	
Zmiana limitu parametrów	LIMITS	Nr parametru (0=wszystkie)	Dolny limit	Górny limit
Drukowanie macierzy kowariancji (obliczanie jeżeli to konieczne)	MATOUT	patrz HESSE		
Wykonanie optymalizacji przez procedurę MIGRAD, bazującej na metodzie Fletchera ze zmienną metryką	MIGRAD	Maks. ilość wywołań (w.w=1000)	Tolerancja (w.w=0.1)	V-stabilność (w.w=0,1)
Minimalizacja funkcji celu, danej w FCN przy użyciu procedury SIMPLEX oraz MIGRAD	MINIMIZE	Maks. ilość wywołań (w.w=1000)	Tolerancja (w.w=0,1)	V-stabilność (w.w=0,1)
Analiza błędu przez procedurę MINOS	MINOS	Maks. liczba wywołań każdego parametru (w.w=1000)	Numery parametrów w formacie I2 (w.w = wszystkie par.)	
Czytanie nowych kart parametrów	PARAMETER	Tu następują karty parametrów, które kończy karta pusta		
Ustawienie poziomu wydruku	PRINTOUT	Poziom (w.w=1)		
Dziurkowanie wartości parametru oraz macierzy błędu	PUNCH			



T a b l i c a 6.1 (cd.)

Zastosowanie danego rozkazu	Kolumna				
	1-10	11-20	21-30	31-40	...
	ROZKAZ	ARG 1	ARG 2	ARG 3	...
Powrót ustalonego uprzednio parametru do roli zmiennej	RELEASE lub RESTORE	Nr parametru 0=wszystkie parametry 1=ostatni ustalony parametr		(aż do 6)	
Przedstawienie wartości funkcji celu FCN w funkcji wybranego parametru	SCAN	Nr parametru (brak=wszystkie)	Dolny limit (w.w=2xodchylenie stand. od minimum)	Górny limit	
Optymalizacja metodą Monte Carlo	SEEK	Liczba wywołań (w.w=100x xNPAR)			
Nadanie parametrowi dolnej wartości	SET PARAM	Nr parametru	Wartość		
Wykonanie optymalizacji metodą Simplex'ów	SIMPLEX	Maks. liczba wywołań (w.w=1000)	Tolerancja (w.w=0,1)		
Wywołanie procedury STAND	STANDARD	Parametry dostępne dla procedury STAND poprzez COMMON (CARD)			

w.w - wartość wbudowana w program.



- kol. 11-20 nazwa alfanumeryczna parametru,
 - kol. 21-30 wartość początkowa (punkt startowy),
 - kol. 31-40 przybliżona wartość błędu lub wartość kroku; przyjęcie wartości 0 (zero) powoduje, że parametr będzie stały,
 - kol. 41-50 Dolna granica parametru
 - kol. 51-60 Górna granica parametru
- } - jeżeli obie granice nie są podane - optymalizacja bez ograniczeń.

Należy zaznaczyć, że w standardowej wersji programu zmiennymi do optymalizacji może być co najwyżej 15 parametrów.

Karta pusta - sygnalizuje koniec kart parametrów.

Karty czytane - przez procedurę FCN (o ile są potrzebne) mogą być poprzedzone przez jedną lub więcej kart pustych.

Karty rozkazów są wszystkie zapisane w formacie A10, 7F10.0. Każda z nich musi się więc zaczynać od nazwy rozkazu, za którą to nazwą występują pewne argumenty.

W tablicy 6.1 przedstawiono wszystkie typy kart rozkazów oraz ich funkcje.

6.3. Karty rozkazów

Każdy z rozkazów rozpoznawalnych przez program zostanie opisany na tyle dokładnie, ażeby umożliwić użytkownikowi ich racjonalne stosowanie.

6.3.1. Rozkaz CALL FCN

Powoduje wywołanie procedury FCN z daną wartością IFLAG, zadeklarowaną jako pierwszy argument rozkazu. Na przykład

CALL FCN 3

Procedura FCN zostanie wywołana z wartością IFLAG = 3. Na ogół jest to używane do drukowania przez FCN krzywych wynikowych, tablic itd. Jeżeli tego typu wywołanie nie nastąpi przed końcem bloku danych, to rozkaz END lub EXIT wykona je automatycznie przed zakończeniem wykonywania się programu.

Normalnie są używane tylko numery IFLAG = 1, 2, 3 oraz 4, a 5 jest zarezerwowane dla ewentualnego wykorzystania w przyszłości. Oczywiście można użyć dowolnej innej wartości IFLAG w celu przeprowadzenia specjalnych obliczeń zaplanowanych przez użytkownika lub też zmiany wartości parametrów czy stałych.



6.3.2. Rozkaz CONTOUR

Rozkaz CONTOUR ma następujący format:

CONTOUR i j k

gdzie: i, j - numery parametrów,
k - żądana liczba konturów.

Użycie tego rozkazu powoduje wykreślenie poziomicy funkcji celu FCN w funkcji dwóch parametrów i oraz j, przy założeniu, że wszystkie pozostałe parametry są stałe i równe swojej wartości w minimum. W momencie użycia rozkazu CONTOUR parametry i oraz j muszą być zmiennymi. K określa liczbę wykreślonych poziomicy przy czym:

- jeżeli $k \leq 0$, to $k = 2$;
- jeżeli $k > 5$, to $k = 5$.

Pierwsza poziomica odpowiada wartości FCN równej

$$F = F_{\min} + UP$$

gdzie: $F_{\min}$ - jest lokalnym minimum,
UP - stała w definicji błędu (w.w. = 0,1).

Jeżeli żąda się wykreślenia więcej niż jednej poziomicy, wówczas N-ta poziomica odpowiada wartości FCN równej

$$F = F_{\min} + UP * N^2.$$

6.3.3. Rozkaz COVARIANCE

Powoduje on czytanie z kart macierzy kowariancji, która może być następnie użyta w optymalizacji procedurą MIGRAD lub analizie błędu procedurą MINIOS. Pierwszy argument podaje liczbę parametrów, której dana macierz odpowiada. Jest ona równa pierwiastkowi kwadratowemu z liczby elementów macierzy, które będą czytane.

Przykład

COVARIANCE	2		
27.42	210	19.41	57.3



Elementy macierzy muszą być czytane w formacie (7F10.0) - (tab. 6.1) i ponieważ musi ona być symetryczna, to nie ma znaczenia czy czyta się ją rzędami czy kolumnami. Należy pamiętać, ażeby odpowiadała ona parametrom, które są zmiennymi.

6.3.4. Rozkaz END

Jest to ostatni rozkaz w bloku danych i wskazuje, że występuje jeszcze co najmniej jeden blok danych. Koniec ostatniego bloku danych, a więc i koniec całego pliku, jest sygnalizowany przez rozkaz EXIT.

Rozkaz END, przed przejściem do następnego bloku danych, sprawdza czy zostało wykonane CALL FCN 3 i jeżeli nie, to powoduje automatyczne wykonanie tego rozkazu.

6.3.5. Rozkaz END RETURN

Tego rozkazu można używać tylko wówczas, gdy program główny MASTER został napisany przez użytkownika. Po każdym wywołaniu procedury MIN-NEW, która powoduje wykonanie jednego pełnego bloku danych, blok ten musi być zakończony rozkazem END RETURN.

6.3.6. Rozkaz EXIT

Spełnia te same funkcje co rozkaz END, lecz sygnalizuje koniec ostatniego bloku danych.

6.3.7. Rozkaz ERROR DEF

Służy on do określenia stałej błędu UP. Kryterium zbieżności zastosowane w programie bazuje na "estymowanej odległości od minimum" EDM

$$EDM = \underline{GS}^T * W * \underline{GS}$$

gdzie: $\underline{GS}$ - wektor pierwszych pochodnych,
 W - macierz kowariancji.



Wartość EDM powinna być połową odległości ($F - F_{\min}$), jeżeli funkcja celu byłaby dokładnie kwadratowa z macierzą kowariancji W . Ze względu na trudności w dokładnym obliczeniu elementów macierzy W , jest stosowane następujące kryterium:

$$\begin{cases} EDM < EPSI * UP = 0.1 * UP \\ VTEST < 0.01 \end{cases}$$

lub też

$$EDM < 10^{-5} * EPSI * UP = 10^{-6} * UP$$

gdzie: $VTEST$ - średnia zmiana elementów diagonalnych macierzy W
 UP - stała definicji błędu (w.w. = 1.0).

6.3.8. Rozkaz FIX

Rozkaz FIX powoduje ustalenie wartości danego parametru. Na przykład

FIX	5
-----	---

Parametr nr 5 przestał być zmienną, lecz jego wartość jest stała w dalszym ciągu obliczeń. Maksymalnie można ustalić 15 parametrów. Ustalony parametr można ponownie wprowadzić do zbioru zmiennych poprzez podanie rozkazu RELEASE lub RESTORE.

6.3.9. Rozkaz GRADIENT

Rozkaz ten wskazuje, że w procedurze FCN są obliczane pochodne funkcji celu. Wywołanie FCN z etykietą 2 przez program główny powoduje, że wartości pierwszych pochodnych po obliczeniu zostaną zapamiętane w wektorze G , który jest drugim parametrem formalnym procedury FCN. Oczywiście, tak jak przy każdym wywołaniu FCN, również wartość funkcji celu zostanie obliczona.

Ponieważ tylko w procedurze MIGRAD są potrzebne wartości pochodnych, to użycie rozkazu GRADIENT ma sens tylko wtedy, gdy używamy rozkazów MIGRAD lub MINOS.



Argumentem rozkazu GRADIENT jest zero, spacja lub 1.0. W przypadku użycia spacji lub zera, pochodne obliczane przez FCN są sprawdzane poprzez obliczenia metodą różnic skończonych. W przypadku braku zgodności program nie akceptuje pochodnych obliczanych przez FCN.

W przypadku, gdy jest pewność, że pochodne w procedurze FCN są obliczane poprawnie, to można użyć argumentu 1.0, w celu pominięcia sprawdzania pochodnych i przyspieszenia obliczeń.

6.3.10. Rozkaz HELP

Rozkaz ten powoduje drukowanie wszystkich dopuszczalnych przez program rozkazów. Po użyciu rozkazu HELP lista dopuszczalnych rozkazów przestaje być drukowana na początku każdego zadania.

6.3.11. Rozkaz HESSE

Rozkaz ten powoduje obliczanie i drukowanie macierzy kowariancji. Jeżeli macierz ta jest już obliczona, to zostaje wyzerowana i obliczona na nowo.

Należy podkreślić, że obliczanie macierzy kowariancji wymaga około $1/2 * NPAR * * 2$ wywołań procedury FCN (nie są używane pochodne obliczane przez FCN) w celu określenia wszystkich drugich pochodnych cząstkowych i następnie odwrócenia tak otrzymanej macierzy.

6.3.12. Rozkaz IMPROVE

Po uzyskaniu minimum lokalnego można, używając rozkazu IMPROVE, "odskoczyć" w inny punkt w celu znalezienia innego lepszego minimum. Użycie tego rozkazu jest dopuszczalne tylko po uzyskaniu zbieżności do lokalnego minimum i znajomości macierzy kowariancji.

Na ogół używamy rozkazu IMPROVE po rozkazie MIGRAD.

6.3.13. Rozkaz LIMITS

Użycie tego rozkazu powoduje nadanie ograniczeń jednemu lub wszystkim parametrom.



Numer parametru podaje się w kolumnie 11-20. Jeżeli zamiast numeru zostanie podana spacja zostaną ustalone ograniczenia dla wszystkich parametrów. Górne ograniczenie podaje się w kolumnie 21-30, a dolne w 31-40.

6.3.14. Rozkaz MATOUT

Rozkaz ten powoduje wydrukowanie wartości parametrów i wszystkich współczynników korelacji, znormalizowanych względem elementów diagonalnych macierzy kowariancji

$$V_{ij}/SQRT(V_{ii} * V_{jj})$$

oraz współczynników korelacji globalnej

$$P_k^2 = 1 - (V_{kk} * (V^{-1})_{kk})^{-1}.$$

W przypadku, gdy ustawiony poziom wydruku jest większy od jedności, jest drukowana pełna (wewnętrzna) macierz kowariancji.

6.3.15. Rozkaz MIGRAD

Użycie rozkazu MIGRAD inicjuje wykonanie minimalizacji funkcji celu przy użyciu zmodyfikowanego przez Fletchera'a [3] algorytmu Davidona-Fletcher a-Powell'a [8].

Rozkaz ma trzy argumenty służące do kontroli zbieżności. Są to:

a) NFCNMX - maksymalna liczba wywołań procedury FCN; minimalizacja zostanie zakończona po wykonaniu tej liczby odwołań, nawet wtedy, gdy minimum nie zostanie jeszcze osiągnięte; w przypadku nie podania tego argumentu lub zadeklarowania wartości ≤ 0 , wartość NFCNMX zostanie przyjęta jako równa 1000;

b) EPSI - dokładność wyznaczania minimum; program może przewidywać jak daleko znajduje się od minimum, na podstawie macierzy kowariancji i wartości pierwszych pochodnych obliczonych przez FCN; kiedy ta przewidywana odległość ϑ staje się mniejsza niż EPSI w dwóch kolejnych iteracjach, to program uznaje, że kryterium zostało spełnione; jeżeli wartość EPSI nie zostanie zadeklarowana lub będzie podana wartość zero, to EPSI zostanie przyjęte jako równe $0.1 * UP$ (patrz rozkaz ERROR DEF);



c) VTEST - dokładność obliczania macierzy błęd; program oblicza średnią zmianę elementów diagonalnych macierzy kowariancji w stosunku do poprzedniej iteracji; jeżeli ta zmiana jest mniejsza od VTEST w dwóch kolejnych iteracjach, to program uznaje kryterium za spełnione.

Zadeklarowanie wartości VTEST = 0 lub niepodanie jej powoduje przyjęcie VTEST = 0.01. Zbieżność do minimum zostaje osiągnięta, gdy oba kryteria (b i c) zostaną spełnione jednocześnie, lub kiedy f stanie się mniejsze od $10^{-5} * EPSI$.

6.3.16. Rozkaz MINIMIZE

Użycie tego rozkazu powoduje rozpoczęcie procesu optymalizacji poprzez wywołanie najpierw procedury SIMPLX, a następnie MIGRAD. Argumenty rozkazu są takie same jak dla rozkazu SIMPLEX, a mianowicie:

- maksymalna liczba wywołań procedury FCN (wartość wbudowana 1000),
- tolerancja obliczania wartości FCN (wartość wbudowana 0.1).

Maksymalna liczba wywołań FCN dotyczy w tym przypadku łącznie procedury SIMPLX oraz MIGRAD.

6.3.17. Rozkaz MINOS

Rozkaz ten powoduje wykonanie analizy błęd dla jednego lub kilku parametrów.

Pierwszym argumentem rozkazu MINOS jest wartość NFCNMX, która określa maksymalną liczbę wywołań procedury FCN dla analizy błęd każdego parametru. W przypadku przekroczenia zadeklarowanej wartości NFCNMX (wartość wbudowana 1000), program rozpoczyna analizę błęd następnego parametru, nawet gdy nie została wykonana analiza błęd poprzedniego.

Począwszy od kolumny 21 rozkazu MINOS, należy podać kolejno numery parametrów, które mają być analizowane, przy czym muszą one być od siebie oddzielone jedną lub dwoma spacjami (Format 30I2). Przy braku deklaracji zostanie dokonana analiza błęd dla wszystkich parametrów.

Numery parametrów jednocyfrowe muszą się zaczynać w kolumnach parzystych, a dwucyfrowe w kolumnach nieparzystych

MINOS	500.	1 1 7 1 1 1 8 0 5
-------	------	-------------------



Użycie tej karty spowoduje analizę błędu dla parametrów o numerach 1, 7, 18 i 5, z maksymalną liczbą 500 wywołań procedury FCN dla każdego parametru. W przypadku, gdy chcemy użyć wartości $UP \neq 1$, rozkaz MINOS powinien być poprzedzony przez rozkaz ERROR DEF. Jeżeli podczas analizy błędu przez procedurę MINOS zostanie znalezione nowe minimum, zostanie przeprowadzona optymalizacja za pomocą procedury MIGRAD (w pełnej przestrzeni parametrów) z:

$$NFCNMX = 4 * NFCNMX$$

$$VTEST = 0.5 * VTEST$$

Po wykonaniu procesu optymalizacji analiza błędu rozpocznie się od nowa (ze starą wartością NFCNMX). Gdyby nowe minimum zostało znalezione po raz drugi, zostanie wykonana ostateczna optymalizacja przy użyciu procedury MIGRAD bez zmiany wartości NFCNMX i VTEST.

6.3.18. Rozkaz PARAMETER

Rozkaz ten powoduje czytanie przez program dodatkowych kart parametrów. Może on być użyty do wprowadzenia nowych parametrów lub zmiany wartości kroku czy ograniczeń istniejących parametrów. Pusta karta oznacza koniec kart parametrów.

6.3.19. Rozkaz PRINTOUT

Rozkaz ten służy do ustawienia poziomu wydruku, który może się zawierać w granicach 0-5. Wartością wbudowaną jest poziom 1, a w tablicy 6.2 podano znaczenie poszczególnych poziomów.

Tablica 6.2

Poziom wydruku	Komentarz
0	Absolutne minimum wydruku, lecz na ogół wystarczające
1	Wydruk standardowy
2	Wydruk używany w trudnych przypadkach
3÷5	Wydruk bardzo wielu informacji, poziom nie zalecane dla indywidualnych użytkowników



6.3.20. Rozkaz PUNCH

Rozkaz ten służy do wydziurkowania aktualnych parametrów, wraz z ich wartościami, błędami i ograniczeniami, w formacie takim samym, jaki jest wymagany przy pisaniu danych wejściowych. Jeżeli przy wykonywaniu rozkazu PUNCH znana jest macierz kowariancji, to również i ona zostanie wydziurkowana. Użycie rozkazu PUNCH bardzo ułatwia ponowne uruchomienie programu.

6.3.21. Rozkaz RELEASE

Rozkaz ten powoduje, że ustalony parametr znów staje się zmienną. Może on dotyczyć tylko tych parametrów, które zostały zadeklarowane jako zmienne, a następnie ich wartości zostały ustalone przy pomocy rozkazu FIX. Argumentem rozkazu jest numer ustalonego parametru, który ma znowu stać się zmienną do optymalizacji.

6.3.22. Rozkaz RESTORE

Rozkaz ten ma takie samo znaczenie, jak RELEASE, lecz użytkownik nie musi podawać numeru parametru. Jeżeli argumentem rozkazu jest zero, to wszystkie ustalone parametry zostaną uzmiennione, a jeżeli jeden - to ostatni z nich.

Przykład

```
FIX      3.
FIX      7.
FIX      11.
MIGRAD
RESTORE  1.  (ustalony parametr 11 znów staje się zmienną)
MIGRAD
RESTORE  0.  (ustalone parametry 3 i 7 znów stają się zmiennymi).
```

6.3.23. Rozkaz SCAN

Rozkaz ten powoduje, że program wykreśla wartość funkcji celu w funkcji jednego (lub wszystkich) parametrów. Argumenty rozkazu SCAN są następujące:



Kolumna	Argument
11-20	numer parametru, przy braku numeru wszystkie parametry,
21-30	liczba kroków (wartość wbudowana 40)
31-40	dolne ograniczenie parametru (dla SCAN),
41-50	górne ograniczenie parametru (dla SCAN)

Przy braku deklaracji wartości dolnego i górnego ograniczenia zmiana parametru nastąpi wokół najlepszej jego wartości o dwa odchylenia standardowe.

6.3.24. Rozkaz SEEK

Rozkaz ten powoduje użycie metody Monte Carlo do poszukiwania minimum funkcji celu, obliczanej przez procedurę FCN.

Argumentem rozkazu jest ilość wywołań procedury FCN podczas poszukiwań. Przy każdym wywołaniu FCN wartości wszystkich parametrów są wybierane losowo, przy założeniu równomiernego rozkładu przy najlepszej wartości parametru i szerokości równej poprzedniej wartości kroku.

Po wykonaniu procedury SEEK do dalszej optymalizacji są wybierane wartości parametrów, które odpowiadają najmniejszej wartości funkcji celu.

6.3.25. Rozkaz SET PARAM

Rozkaz ten powoduje nadanie parametrowi konkretnej stałej wartości. Numer parametru należy zadeklarować w kolumnie 11-20, a jego wartość w kolumnie 21-30.

6.3.26. Rozkaz SIMPLEX

Rozkaz ten powoduje wykonanie optymalizacji przy użyciu metody Simplex'ów. Rozkaz ma dwa argumenty do kontroli zbieżności. Są to:

NFCNMx - maksymalna liczba wywołań procedury FCN; wartość wbudowana 1000;

EPSI - dokładność wyznaczania minimum; wartość wbudowana 0.1 * UP.



Uważa się, że zbieżność z minimum została osiągnięta, gdy wartości funkcji celu w $NP\bar{A}R+1$ punktach simpleksu różnią się o mniej niż ϵPSI .

6.3.27. Rozkaz STANDARD

Rozkaz ten powoduje wywołanie procedury STAND, dołączonej przez użytkownika.

6.4. Procedury dołączane przez użytkownika.

Procedura FCN (NP\bar{A}R, G, F, X, IFLAG)

Na etapie kompilacji użytkownik musi dołączyć napisaną przez siebie procedurę FCN, w której będzie obliczana wartość funkcji celu $F(X)$.

Dodatkową procedurą, która może (lecz nie musi być) dołączona przez użytkownika, jest procedura STAND, której wywołanie jest sterowane kartą rozkazów STANDARD.

Procedura FCN oblicza wartość funkcji celu, która ma być minimalizowana przez program. Argumentami procedury są:

NP\bar{A}R - liczba zmiennych parametrów (maksymalnie 15)

G - wektor pierwszych pochodnych

F - wartość funkcji celu obliczanej w FCN

X * - wektor zawierający zewnętrzne wartości parametrów

IFLAG * - zostanie opisany w tablicy 6.3.

Argumenty oznaczone * są danymi wejściowymi dla procedury FCN i nie mogą być przez nią zmienione (tzn. nie mogą znaleźć się po lewej stronie znaku równości).

Parametr IFLAG podaje, co powinno być obliczane przez procedurę w danym wywołaniu, w zależności od jego wartości.

Tablica 6.3 wyjaśnia znaczenie poszczególnych wywołań.

Szczególną uwagę należy zwrócić na jak najlepsze opracowanie strategii obliczania wartości funkcji celu F oraz wektora pochodnych G, gdyż największa część czasu obliczeniowego jest zużywana na te obliczenia.

Poniżej podano przykładową strukturę procedury FCN przy założeniu, że funkcja F jest funkcją:

a) parametrów (zmiennych) zawartych w wektorze $\underline{X}$,

b) stałych C,

c) wektorów stałych $E(I)$; ich użycie zmusza do wykonywania obliczeń w pętli I.



Wartość IFLAG	Obliczenia lub czynności, które mają być wykonane przez procedurę FCN
1	wejście początkowe, czytanie wszystkich danych do procedury, obliczanie stałych, drukowanie tablic wejściowych itd.
2	normalne wejście z obliczeniami pochodnych; wartości obliczone w punkcie X są gromadzone w wektorze G , a wartość funkcji celu - w zmiennej F
3	wejście kończące obliczenia; wydruk wszystkich specjalnych tablic wyjściowych, wykresów itd. dla punktu będącego minimum
4	normalne wejście bez obliczeń pochodnych; jest obliczana tylko wartość funkcji celu F w punkcie X
5	etykieta rezerwowa; nie używana
Inne	zgodnie ze specjalnymi podprogramami napisanymi przez użytkownika

W celu sprawienia, aby obliczenia wykonywane przez procedurę FCN były jak najszybsze należy stałe C i wektory $E(I)$ używać tylko raz do obliczeń oraz unikać zbędnych pętli względem I . Biorąc powyższe pod uwagę przykładowa struktura procedury FCN jest następująca:

```

SUBROUTINE FCN (NPAR, G, F, X, FLAG)
  DIMENSION G(15), X(15)
  COMMON /DUMMY/
  GO TO (10, 20, 30, 40, 50, 60) IFLAG
10  CONTINUE
    [czytaj karty danych
    ustal stałe C
    oblicz funkcje od C
    oblicz funkcje od C i E (I)]
20  CONTINUE
    [oblicz funkcje od X i C potrzebne do obliczenia G
    oblicz unkcje od X, C i E(I) potrzebne do obliczenia G
    oblicz G]

```



40 CONTINUE

[oblicz funkcje od $\underline{X}$ oraz C potrzebne do obliczenia F
oblicz funkcje od $\underline{X}$, C oraz $E(I)$ potrzebne do obliczenia F
oblicz F]

RETURN

30 CONTINUE

[wykonaj wszystkie operacje, które wymagają znajomości
wartości parametrów $\underline{X}$ w minimum F. Na przykład, drukuj
tabele końcowe, rysuj wykresy itd.]

GO TO 40

50 [nie używana]

60 [zgodnie ze specjalnymi życzeniami użytkownika]

RETURN

END

Należy zauważyć, że stałe C, $E(I)$ oraz zmienne $\underline{X}$ nie mogą się zmieniać pod etykietami 20 i 40. Parametry $\underline{X}$ zmieniają się w innych częściach programu, zgodnie ze strategią poszukiwania minimum funkcji celu F.

6.5. Wydruk wyników

Wyniki drukowane przez program są bardzo łatwe do analizowania i następują zgodnie z sekwencją wczytywania danych wejściowych. Poniżej omówimy kolejno poszczególne wydruki.

6.5.1. Wydruk kart danych

Wartości początkowe parametrów oraz kroki obliczeń są drukowane w formacie F15.6. W przypadku wystąpienia błędu fatalnego programu drukuje komentarz, po czym zatrzymuje się po zakończeniu czytania wszystkich kart.



6.5.2. Wydruk kart rozkazów

Wszystkie karty rozkazów są drukowane natychmiast po wczytaniu i przed rozpoczęciem obliczeń. Są one otoczone trzema krótkimi rzędami gwiazdek i znajdują się po lewej stronie wydruku wyników. Argumenty rozkazów są drukowane w formacie F15.4. Jeżeli argument nie jest wydrukowany, to znaczy że był on zadeklarowany jako zero lub spacja.

W przypadku nierozpoznania rozkazu (jeżeli jest błędny) program opuszcza go i drukuje informacje COMMAND IGNORED.

6.5.3. Normalny poziom wydruku (PRINTOUT = 1)

Przy każdym wydruku jest podawana aktualna wartość funkcji celu, całkowita liczba wywołań procedury FCN od początku bloku danych, czas i aktualne oszacowanie odległości do minimum (EDM), o ile zostało zrobione.

Dla każdego parametru są drukowane następujące informacje:

- 1) wewnętrzny numer parametru - tylko zmienne parametry posiadają wewnętrzne numery,
- 2) zewnętrzny numer parametru - taki jak w procedurze FCN,
- 3) nazwa parametru, taka jak w karcie danych,
- 4) zewnętrzna wartość parametru - jest to wartość czytana i używana w obliczeniach przez procedurę FCN,
- 5) zewnętrzny błąd parametru - jest to aktualne i najlepsze przybliżenie rzeczywistego błędu parametru, przy wzięciu pod uwagę wartości UP,
- 6) wewnętrzna wartość parametru - jest to wartość parametru przekształcona do użytku wewnętrznego; w przypadku, gdy parametr nie ma ograniczeń, to wartość ta jest taka sama, jak wartość zewnętrzna,
- 7) wewnętrzna wartość kroku - na początku programu jest to błąd początkowy czytany z kart danych, przetransformowany na układ współrzędnych wewnętrznych; podczas i po optymalizacji jest on równy wartości ostatniego kroku.

Na końcu procedury MINOS jest drukowany cały zbiór informacji dotyczących zarówno rezultatów optymalizacji, jak i analizy błędów.



6.5.4. Wydruk macierzy kowariancji i macierzy korelacji błędu

Macierz kowariancji jest drukowana zawsze we współrzędnych wewnętrznych, które są takie same, jak prawdziwe parametry zewnętrzne tylko wtedy, gdy nie ma ograniczenia parametrów. Macierz jest zawsze symetryczna, tak więc elementy powyżej przekątnej głównej nie są nigdy drukowane.

6.5.5. Rysowanie poziomic

Poziomice błędu są rysowane za pomocą drukarki wierszowej i kolejne kontury stałego błędu są oznaczone cyframi od 1 do 9. Kontur oznaczony cyfrą 0 odpowiada minimum funkcji celu, natomiast I-ty kontur odpowiada wartości

$$F = F_{\min} + UP * I^{**2}.$$

Po rysowaniu każdego konturu pojawiają się informacje dotyczące funkcji celu i wartości parametrów.

6.6. Przykład użycia programu MINUIT

Rozważmy funkcję celu o postaci

$$F = E(S) * * 2 + D(S) * * 2, \quad (6.1)$$

gdzie: $E(S)$ - wartość oczekiwana,

$$E(S) = S_1 E(Q_1) + S_2 E(Q_2) + S_3 E(Q_3), \quad (6.2)$$

$D(S)$ - odchylenie standardowe

$$\begin{aligned} D(S)^2 = & S_1^2 D^2(Q_1) + S_2^2 D^2(Q_2) + S_3^2 D^2(Q_3) + \\ & + 2\rho_{12} D(Q_1) D(Q_2) + 2\rho_{13} D(Q_1) D(Q_3) + 2\rho_{23} D(Q_2) D(Q_3); \end{aligned} \quad (6.3)$$



```

LIST
DUMP ON(BETABIN)
SEMICOMPILED(MINUITC,FORTSEMICOMP)
LIBRARY(SUBGROUFPSC)
PROGRAM(BETA)
INPUT 1=CRO
OUTPUT 2=LPO
USE3=EDO
TTRACE 2
END
MASTER BETAIEN
CALL MINUITS
STOP
END
SUBROUTINE FCN(NPAR,G,F,X,IFLAG)
DIMENSION G(15),X(15)
GO TO (10,20,30,40,50) IFLAG
10 CONTINUE
READ(1,1000) E1,E2,E3
READ(1,1000) D1,D2,D3
READ(1,1000) R01,R02,R03
READ(1,1001) KS
1000 FORMAT(3F0.0)
1001 FORMAT(1I0)
20 CONTINUE
G(1),G(2),G(3) =0.0
ES=X(1)*E1+X(2)*E2+X(3)*E3
D2S=X(1)*X(1)*D1*D1+X(2)*X(2)*D2*D2+X(3)*X(3)*D3*D3+2*R01*D1*D2+
& 2*R02*D1*D3+2*R03*D2*D3
F=ES*ES+D2S
G(1)=2*ES*E1+2*X(1)*D1*D1
G(2)=2*ES*E2+2*X(2)*D2*D2
G(3)=2*ES*E3+2*X(2)*D3
40 CONTINUE
ES=X(1)*E1+X(2)*E2+X(3)*E3
D2S=X(1)*X(1)*D1*D1+X(2)*X(2)*D2*D2+X(3)*X(3)*D3*D3+2*R01*D1*D2+
& 2*R02*D1*D3+2*R03*D2*D3
F=ES*ES+D2S
WF =F
RETURN
30 CONTINUE
WRITE(2,2000) X(1),X(2),X(3),WF
2000 FORMAT(1X,4G14.7)
GO TO 40
50 CONTINUE
RETURN
END
FINISH

```

Rys. 6.2. Listing programu BETA do minimalizacji funkcji celu danej wzorem (6.1)



zmiennymi są wrażliwości układowe S_1 , S_2 , i które mogą się zmieniać w następujących granicach:

Wrażliwość	Dolne ograniczenia	Górne ograniczenia
S_1	0.1	1
S_2	-0.5	0.5
S_3	-0.2	0.7

Pozostałe wielkości są stałymi parametrami, mogącymi przyjmować następujące wartości:

Stałe	Dolne ograniczenie	Górne ograniczenie
$E(Q_1)$, $E(Q_2)$, $E(Q_3)$	-0.5	0.5
$D(Q_1)$, $D(Q_2)$, $D(Q_3)$	0	1
ρ_{12} , ρ_{13} , ρ_{23}	-1	1

```

DANELESZCZYN
1      S1      0.5      0.001      0.1      1
2      S2      0.4      0.001      -0.5     0.5
3      S3      0.1      0.0001     -0.2     0.7
0.6    0.7    -0.1
0.3    0.2     0.5
0.3   -0.4     0.6
2
PRINTOUT      1
ERROR DEF     0001
SEEK          9000
GRADIENT      0
MINIMIZE      9000
IMPROVE       1000
CONTUR        1      2      3
HESSE         100
MATOUT
EXIT
****

```

Rys. 6.3. Plik danych wejściowych



W celu rozwiązania tak postawionego zagadnienia napisano program przedstawiony na rysunku 6.2. Jako dane do tego programu są czytane wartości stałych parametrów przez procedurę FCN pod etykietą 10 (IFLAG = 1). Plik danych wejściowych przedstawiono na rysunku 6.3. Wartości stałych przyjęto następujące:

$$E(Q_1) = E1 = 0.6$$

$$E(Q_2) = E2 = 0.7$$

$$E(Q_3) = E3 = -0.1$$

$$D(Q_1) = D1 = 0.3$$

$$D(Q_2) = D2 = 0.2$$

$$D(Q_3) = D3 = 0.5$$

$$\rho_{12} = R01 = 0.3$$

$$\rho_{13} = R02 = -0.4$$

$$\rho_{23} = R03 = 0.6$$

Uzyskane wyniki przedstawiono na rysunku 6.4. Ze względu na zamieszczenie wyników wydrukowanych na OPTIMIE, musiano obciąć ostatnie 10 kolumn, gdyż OPTIMA pozwala na wydruk tylko 110 kolumn.

Po wydruku wszystkich rozkazów rozpoznawanych przez program, następuje wydruk tytułu DANE LESZCZYN oraz wstępnych wartości parametrów, założonej dokładności i przedziałów ich zmienności. Pierwsze wywołanie procedury FCN dało w wyniku wartość funkcji celu $F = 0.3923$.

Po wykonaniu procedury SEEK, przy 9000 wywołaniach procedury FCN, metoda Monte Carlo znalazła punkt dający wartość funkcji celu 0.03841587.

Rozkaz GRADIENT wskazuje, że w procedurze FCN są obliczane pochodne funkcji celu. Argumentem jest 0, więc obliczenia są sprawdzane: pochodna względem zmiennej S_3 jest obliczona błędnie w FCN, co jest sygnalizowane przez program.

Rozkaz MINIMIZE powoduje wykonanie optymalizacji najpierw metodą Simplex'ów, a następnie metodą Davidona-Fletcher a-Powell'a (D-F-P).

Po wykonaniu optymalizacji metodą Simplex wartość funkcjonału zmalała do 0.037233826, a po wykonaniu obliczeń metodą D-F-P do 0.03717088.

Rozkaz IMPROVE powoduje poszukiwanie nowego minimum, lecz w tym przypadku skończyło się to niepowodzeniem i znalezione minimum okazało się najlepszym. Rozkaz CONTOUR powoduje narysowanie trzech poziomów stałego błędu w funkcji zmiennej $S1$ i $S2$. Należy zauważyć, że parametr



```
MINOS
RESTORE
COVARIANCE
CONTOUR
END RETURN
```

```
*****
*   D506   MINUITS   *
*   VERSION 8.75     *
*   DATA BLOCK NO. 1 *
*****
```

```

*****
DANELESZCZYN*****TIME18/06/50
*****
1      S1      0.500000      0.001000      0.1000E 00      0.1000E 01
2      S2      0.400000      0.001000      -0.5000E 00      0.5000E 00
3      S3      0.100000      0.000100      -0.2000E 00      0.7000E 00
*****

```

FCN VALUE	CALLS	TIME	EDM	INT	EXT.	PARAMETER	VALUE	ERROR	INTERN. VALUE	INT
0.3923000E 00	1	18/06/51	0.00E 00	1	1	S1	0.50000E 00	0.10000E-02	-0.11134E 00	
				2	2	S2	0.40000E 00	0.10000E-02	0.92730E 00	0
				3	3	S3	0.10000E 00	0.10000E-03	-0.33984E 00	

```
*****
*** 1****PRINTOUT 1.00000
*****
*****
*** 2****ERROR DEF 0.00010
*****
*****
*** 3****SEEK 9000.00000
```

SEEK -- MONTE CARLO MINIMUM SEARCH
BEST VALUE FOUND IN SEEK IS

FCN VALUE	CALLS	TIME	EDM	INT	EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3841587E-01	9003	18/09/58	0.00E 00	1	1	S1	0.10001E 00	0.10000E-02	-0.15646E 01	
				2	2	S2	-0.69883E-01	0.10000E-02	-0.14023E 00	
				3	3	S3	0.72262E-01	0.10000E-03	-0.40604E 00	

*** 4***GRADIENT

CHECK OF GRADIENT CALCULATION IN FCN

PARAMTER		G (IN FCN)	G (MINUIT)	ERROR	AGREEMENT
1	S1	0.6306E-04	0.6306E-04	0.1138E-05	GOOD
2	S2	-0.9181E-04	-0.9181E-04	0.4330E-04	GOOD
3	S3	-0.2921E-01	0.1462E-01	0.1109E-05	BAD

MINUIT DOES NOT ACCEPT DERIVATIVE CALCULATIONS BY PCN

```
*** 5***MINIMIZE          9000.00000
*****
```

START SIMPLEX MINIMIZATION

CONVERGENCE CRITERION -- ESTIMATED DISTANCE TO MINIMUM (EDM) .LT

FCN VALUE	CALLS	TIME	EDM	INT	EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3796989E-01	9025	18/09/59	0.45E-03	1	1	S1	0.10000E 00	0.23332E-05	-0.15682E 01	
				2	2	S2	-0.69619E-01	0.11704E-02	-0.13969E 00	
				3	3	S3	0.58182E-01	0.46780E-02	-0.44036E 00	

SIMPLEX MINIMIZATION HAS CONVERGED

FCN VALUE	CALLS	TIME	EDM	INT	EXT.	PARAMETER	VALUE	ER OR	INTERN.VALUE	INT
0.3723836E-01	9044	18/09/59	0.30E-05	1	1	S1	0.10001E 00	0.22832E-01	-0.15782E 01	
				2	2	S2	-0.6 551E-01	0.20701E-02	-0.13552E 00	
				3	3	S3	0.49498E-02	0.36792E-01	-0.57586E 00	

START MIGRAD MINIMIZATION.

```

ERRORS CORRESPOND TO FUNCTION CHANGE OF      0.0001
CONVERGENCE CRITERIA -- ESTIMATED DISTANCE TO MINIMUM (EDM) .LT
OR EDM .LT. 0.10E-04 AND FRACTIONAL CHANGE IN VARIANCE MATRIX .LT

```

FCN VALUE	CALLS	TIME	EDM	INT	EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3723836E-01	9050	18/09/59	0.13E-03	1	1	S1	0.10001E 00	0.30536E-02	-0.15782E 01	
				2	2	S2	-0.67551E-01	0.13710E-01	-0.13552E 00	
				3	3	S3	0.49498E-02	0.19602E-01	-0.57586E 00	

MIGRAD MINIMIZATION HAS CONVERGED

FCN VALUE	CALLS	TIME	EDM	INT	EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3717088E-01	9065	18/09/59	0.22E-0	1	1	S1	0.10000E 00	0.30533E-02	-0.15707E 01	-
						WARNING -	- ABOVE PARAMETER IS AT LIMIT.			
				2	2	S2	-0.79007E-01	0.13943E-01	-0.15868E 00	-
				3	3	S3	0.18076E-02	0.19874E-01	-0.58421E 00	-

ERRORS CORRESPOND TO FUNCTION CHANGE OF 0.0001

INTERNAL COVARIANCE MATRIX

CORRELATION COEFFICIENTS

INT.	1	2
2-	0.001	
3-	0.004	0.189

LAST FRACTIONAL CHANGE WAS 0.020837



GLOBAL CORRELATION
PARAMETER COEFFICIENT

1	S1	0.00002
2	S2	0.03569
3	S3	0.03570

*** 6***IMPROVE 1000.00000

START ATTEMPT NO. 1 TO FIND NEW MINIMUM

START SIMPLEX MINIMIZATION

CONVERGENCE CRITERION -- ESTIMATED DISTANCE TO MINIMUM (EDM) .LT

PCN VALUE	CALLS	TIME	EDM	INT EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3728158E-01	9085	18/10/00	0.35E-03	1 1	S1	0.100008E 00	0.30527E-02	-0.15519E 01	
				2 2	S2	-0.77567E-01	0.13950E 01	0.15576E 00	
				3 3	S3	-0.18137E-01	0.19134E-01	-0.63834E 00	

SIMPLEX MINIMIZATION HAS CONVERGED

PCN VALUE	CALLS	TIME	EDM	INT EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3717209E-01	9107	18/10/12	0.33E-05	1 1	S1	0.10001E 00	0.30532E-02	-0.15637E 01	
				2 2	S2	-0.78230E-01	0.13947E-01	-0.15711E 00	
				3 3	S3	0.35701E-02	0.19936E-01	-0.57952E 00	

IMPROVE HAS RETURNED TO REGION OF ORIGINAL MINIMUM

START ATTEMPT NO. 2 TO FIND NEW MINIMUM

IMPROVE HAS RETURNED TO REGION OF ORIGINAL MINIMUM

START ATTEMPT NO. 3 TO FIND NEW MINIMUM

IMPROVE HAS RETURNED TO REGION OF ORIGINAL MINIMUM

START ATTEMPT NO. 4 TO FIND NEW MINIMUM

START SIMPLEX MINIMIZATION

CONVERGENCE CRITERION -- ESTIMATED DISTANCE TO MINIMUM (EDM) .LT

PCN VALUE	CALLS	TIME	EDM	INT EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3726970E-01	9141	18/10/13	0.24E-03	1 1	S1	0.10362E 00	0.66279E-02	-0.16978E 01	
				2 2	S2	-0.77179E-01	0.13951E-01	-0.15498E 00	
				3 3	S3	0.52585E-02	0.19994E-01	-0.57504E 00	

SIMPLEX MINIMIZATION HAS CONVERGED

PCN VALUE	CALLS	TIME	EDM	INT EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3717298E-01	9158	18/10/13	0.84E-05	1 1	S1	0.10001E 00	0.30532E-02	-0.15774E 01	
				2 2	S2	-0.77109E-01	0.13952E-01	-0.15484E 00	
				3 3	S3	0.21374E-02	0.19886E-01	-0.58333E 00	

IMPROVE HAS RETURNED TO REGION OF ORIGINAL MINIMUM

START ATTEMPT NO. 5 TO FIND NEW MINIMUM

IMPROVE HAS RETURNED TO REGION OF ORIGINAL MINIMUM

START ATTEMPT NO. 6 TO FIND NEW MINIMUM

IMPROVE HAS RETURNED TO REGION OF ORIGINAL MINIMUM

START ATTEMPT NO. 7 TO FIND NEW MINIMUM

START SIMPLEX MINIMIZATION

CONVERGENCE CRITERION -- ESTIMATED DISTANCE TO MINIMUM (EDM) .LT

PCN VALUE	CALLS	TIME	EDM	INT EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3968547E-01	9205	18/10/14	0.12E-02	1 1	S1	0.17176E 00	0.28351E-01	-0.21433E 01	
				2 2	S2	-0.15584E 00	0.13417E-01	-0.31695E 00	
				3 3	S3	0.72829E-02	0.20063E-01	-0.56969E 00	

SIMPLEX MINIMIZATION HAS CONVERGED

PCN VALUE	CALLS	TIME	EDM	INT EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3717097E-01	9246	18/10/15	0.29E-05	1 1	S1	0.10000E 00	0.30533E-02	-0.15697E 01	
				2 2	S2	-0.79352E-01	0.13942E-01	-0.15938E 00	
				3 3	S3	0.14459E-02	0.19862E-01	-0.58517E 00	

IMPROVE HAS RETURNED TO REGION OF ORIGINAL MINIMUM

PCN VALUE	CALLS	TIME	EDM	INT EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3717088E-01	9246	18/10/15	0.00E 00	1 1	S1	0.10000E 00	0.30533E-02	-0.15707E 01	
				2 2	S2	-0.79007E-01	0.13943E-01	-0.15868E 00	
				3 3	S3	0.18076E-02	0.19874E-01	-0.58421E 00	

ERRORS CORRESPOND TO FUNCTION CHANGE OF 0.0001



*** 7***CONTUR

1.00000

2.00000

3.00000

PARAMETERS NO. 1 AND 2 CONTOUR 1 F = FMIN + 0.000100

CONTOUR LEAVING ALLOWED REGION AT 0.9908095E-01 -0.6290227E-01

CONTOUR NOT COMPLETED BECAUSE OF ABOVE ERROR CONDITION

10 POINTS ON THE CONTOUR HAVE BEEN DETERMINED

PARAMETERS NO. 1 AND 2 CONTOUR 2 F = FMIN + 0.000400

CONTOUR LEAVING ALLOWED REGION AT 0.9700656E-01 -0.4658660E-01

CONTOUR NOT COMPLETED BECAUSE OF ABOVE ERROR CONDITION

12 POINTS ON THE CONTOUR HAVE BEEN DETERMINED

PARAMETERS NO. 1 AND 2 CONTOUR 3 F = FMIN + 0.000900

CONTOUR LEAVING ALLOWED REGION AT 0.9718716E-01 -0.3376893E-01

CONTOUR NOT COMPLETED BECAUSE OF ABOVE ERROR CONDITION

18 POINTS ON THE CONTOUR HAVE BEEN DETERMINED

FUNCTION CONTOURS FOR PARAMETERS

1 2

CONTOUR 1 = FMIN + 0.00010

PARAMETER 2

```

-0.0417706 .....
-0.0444619 ....
-0.0471531 ....
-0.0498444 .....
-0.0525356 .....
-0.0552268 .....
-0.0579181 .....
-0.0606093 .....
-0.0633005 .....
-0.0659918 .....
-0.0686830 .....
-0.0713742 .....
-0.0740655 .....
-0.0767567 .....
-0.0794479 .....
-0.0821392 .....
-0.0848304 .....
-0.0875216 .....
-0.0902129 .....
-0.0929041 .....
-0.0955953 .....
-0.0982866 .....
-0.1009778 .....
-0.1036690 .....
-0.1063603 .....
-0.1090515 .....
-0.1117428 .....
-0.1144340 .....
-0.1171252 .....

```

0.100000 0.100000

ONE COLUMN = 0.0003365

0.132980

PARAMETER 1,

FCN VALUE	CALLS	TIME	EDM	INT EXT.	PARAMETER	VALUE	ERROR	INTERN. VALUE	INT
0.3717088E-01	9325	18/10/33	0.00E 00	1 1	S1	0.10000E 00	0.30533E-02	-0.15707E 01	
					WARNING	- ABOVE PARAMETER IS AT LIMIT.			
					2 2	S2	-0.79007E-01	0.13943E-01	-0.15868E 00
					3 3	S3	0.18076E-02	0.19874E-01	-0.58421E 00

ERRORS CORRESPOND TO FUNCTION CHANGE OF 0.0001

ERRORS CORRESPOND TO FUNCTION CHANGE OF 0.0001

*** 8****HESSE 100.00000

START SECOND DERIVATIVE CALCULATION
SECOND DERIVATIVE MATRIX INVERTED

FCN VALUE	CALLS	TIME	EDM	INT	EXT.	PARAMETER	VALUE	ERROR	INTERN.VALUE	INT
0.3717088E-01	9340	18/10/33	0.77E-10	1	1	S1	0.10000E 00	0.42603E-02	-0.15707E 01	
						WARNING -	- ABOVE PARAMETER IS AT LIMIT.			
				2	2	S2	-0.79007E-01	0.13734E-01	-0.15868E 00	
				3	3	S3	0.18076E-02	0.19603E-01	-0.58421E 00	

ERRORS CORRESPOND TO FUNCTION CHANGE OF 0.0001

INTERNAL COVARIANCE MATRIX
CORRELATION COEFFICIENTS

INT. 1 2
2-0.000
3 0.000 0.000

	PARAMETER	GLOBAL CORRELATION COEFFICIENT
1	S1	0.00000
2	S2	0.00000
3	S3	0.00000

*** 9****MATOUT

INTERNAL COVARIANCE MATRIX
CORRELATION COEFFICIENTS

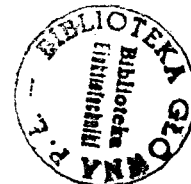
INT. 1 2
2-0.000
3 0.000 0.000

	PARAMETER	GLOBAL CORRELATION COEFFICIENT
1	S1	0.00000
2	S2	0.00000
3	S3	0.00000

*** 10****EXIT

CALL TO FCN WITH IFLAG = 3

0.1000000 -0.7900663E-01 0.1807608E-02 0.3717250E-01



Rys. 6.4. Wyniki obliczeń przy użyciu programu BETA



S1 osiągnął swoje dolne ograniczenie 0.1. Rozkaz HESSE powoduje obliczenie i wydrukowanie wewnętrznej macierzy kowariancji, natomiast rozkaz MATOUT wydrukowanie wartości parametrów i wszystkich współczynników korelacji.

Ostatnie wywołanie procedury FCN z IFLAG = 3 powoduje wydruk otrzymanych rezultatów.



7. WYKORZYSTANIE PROGRAMÓW NAP2, PAUT2 I MINUIT POD KONTROLĄ SYSTEMU OPERACYJNEGO GEORGE 3

7.1. Obliczenia z wykorzystaniem programu NAP2

W celu użycia programu NAP2 pod kontrolą systemu operacyjnego GEORGE 3, napisano szereg makroinstrukcji ułatwiających użytkownikowi posługiwanie się programem.

```
#LISTING OF :NAP2.NAPRUNO5S(1/) PRODUCED ON 16AUG84 AT 16.21.18
#OUTPUT BY LISTFILE IN ':NAP2.NAPIERN' ON 16AUG84 AT 16.27.50 USING  US
DOCUMENT    NAPRUNO5S
#INSTYTUT ELEKTRONIKI POLITECHNIKI ŁÓDZKIEJ
#MACRO NAPRUNO5 SŁUŻY DO URUCHAMIANIA PROGRAMU NAP2
#W WERSJI ZAPISANEJ NA PAKIECIE DYSKOWYM NROOOO05 I WYMAGA
#USTAWIENIA PARAMETRÓW INSTALACYJNYCH INSTRUKCJAMI
#*****
IF NEG(%;INS(OBJECTQUOTA);-59000),QN 1,CZY PARAMETRY-
-INSTALACYJNE USTAWIONE?
RP FB,CM,OJ
MZ 60000
WHENEVER COMERR,GO TO 1NAP
FIND #NAPS,DA,(000005,PROGRAM NAPS)
#RE ON
IF PRE(*CR) AND NOT STR(%*CR")=(),RV %(*CR)
IF PRE(*CR) AND STR(%*CR")=(),OL *CRO
IF ABS(*CR),OL *CRO
IF PRE(*LP),AS *IPO,%( *LP)
IF ABS(*LP),OL *LPO
IF PRE(*CR) AND NOT STR(%*CR")=(),AS *CRO,%( *CR)
IF ABS(TI),TIME 5MINS
IF PRE(TI),TIME %(TI)
OL *DA1(OVERLAY),(000005,PERT)
OL *DA2(OVERLAY),(000005,WWY)
OL *DA3(OVERLAY),(000005,LIBRARY)
OL *DA4(OVERLAY),(000005,CATALOG)
OL *DA5(OVERLAY),(000005,MAA)
EN
IF FAILED(TIME UP),GO TO 2NAP
3NAP IF PRE(*LP),LF %(*LP),*LP
4NAP IF MOP,(EXIT)ELSE(ENDDJOB)
2NAP DP 0,CZAS ZADANIA PRZEKROCZONY
TI 10MINS
RM
GO TO 3NAP
1NAP DP 0,BŁĄD W MACRO NAPRUN:MACRO ZANIECHANE .
GO 4NAP
```

Rys. 7.1. Makroinstrukcja NAPRUNO5S do uruchamiania programu NAP2 pod kontrolą systemu GEORGE-3 przez indywidualnego użytkownika



Na rysunku 7.1 przedstawiono listing makroinstrukcji NAPRUN05S, która służy do uruchamiania programu przez indywidualnego użytkownika. Jeżeli użytkownik pracuje z końcówki MOP, wówczas należy uruchomić program w sposób następujący:

```
NAPRUN L *CRDANE, *LPWYNIK, TI L 25MINS.
```

Jeżeli czas zadania nie zostanie podany, wówczas program przyjmie $TI = 5$ MINS i po wyczerpaniu tego czasu udzieli dodatkowych 10 MINS na zakończenie obliczeń. Jeżeli tak jak w przykładzie zadeklarujemy czas, np. 25 MINS, to również po wyczerpaniu się tego czasu zostanie przydzielone 10 MINS na zakończenie obliczeń programu.

Brak deklaracji zbioru danych spowoduje, że dane będą czytane ON LINE z końcówki MOP. Również brak deklaracji zbioru wynikowego, w którym mają być gromadzone wyniki spowoduje przydzielenie ON LINE końcówki MOP i wyprowadzenie bezpośrednio na nią wyników obliczeń. Praca w systemie interakcyjnym powinna odbywać się na DZM-ce, z założonym szerokim papierem, gdyż wydruk wyników obliczeń programu na 80-kolumnowym monitorze będzie nieczytelny.

W przypadku pracy w zadaniu drugoplanowym (BACK) deklaracja RUNJOB z użyciem makroinstrukcji NAPRUN05S jest postaci:

```
RJ L NAZWA ZADANIA, :NAP2, NAPRUN05S, PARAM (*CRDANE, *LPWYNIK, -  
TI L 25 MINS)
```

Znaczenie parametrów jest takie samo, jak w przypadku pracy z końcówki MOP, lecz zbiór danych oraz zbiór wynikowy muszą być zadeklarowane.

W komentarzu makroinstrukcji napisano o wymaganiach dotyczących parametrów instalacyjnych maszyny cyfrowej Odra 1305. Dotyczyły one sytuacji, gdy Instytut Informatyki PŁ dysponował pamięcią operacyjną 64k i były postaci:

```
IP OBJECTQUOTA, 60K  
IP COREOBJECT, 60K  
IP CHAPTERQUOTA, 10K  
IP BACKGROUND, 1  
IP MOPLIMIT, 1
```

W aktualnej konfiguracji m.c. Odra 1305 dysponującej pamięcią operacyjną 192K nie jest już potrzebne takie ustawienie parametrów instalacyjnych i zostało to pominięte w opisie makroinstrukcji NAPRUN05S.



Do uruchamiania programu NAP2, zapisanego na pakiecie dyskowym 000016 i 000017, służą makroinstrukcje NAPRUN16 i NAPRUN17 przedstawione odpowiednio na rysunku 7.2 i rysunku 7.3. Wersje programów zapisane na tych pakietach nie są przeznaczone do używania przez indywidualnych użytkowników i mogą być używane po konsultacji z prowadzącym zajęcia.

```
#ANDRZEJ NAPIERALSKI
#INSTYTUT ELEKTRONIKI POLITECHNIKI ŁÓDZKIEJ
#MACRO NAPRUN16 SŁUŻY DO URUCHAMIANIA PROGRAMU NAP2
#W WERSJI ZAPISANEJ NA PAKIECIE DYSKOWYM NRO00016 I WYMAGA
#USTAWIENIA PARAMETROW INSTALACYJNYCH INSTRUKCJAMI
#IP OBJECTQUOTA, 60K
#IP CHAPTERQUOTA, 10K
#IP COREOBJECT, 60K
#IP BACKGROUND, 1
#IP MOPLIMIT, 1
#*****
IF NEG(%;INS(OBJECTQUOTA);-59000),QN 1,CZY PARAMETRY-
-INSTALACYJNE USTAWIONE?
RP FB,CM,OJ
MZ 60000
WHENEVER COMERR,GO TO 1NAP
FIND #NAP2,DA,(000016,PROGRAM NAP2)
RE ON
IF PRE(*CR) AND NOT STR("%*CR")=(),RV %(*CR)
IF PRE(*CR) AND STR("%*CR")=(),OL *CRO
IF ABS(*CR),OL *CRO
IF PRE(*LP),AS *LPO,%( *LP)
IF ABS(*LP),OL *LPO
IF PRE(*CR) AND NOT STR("%*CR")=(),AS *CRO,%( *CR)
IF ABS(TI),TIME 100MINS
IF PRE(TI),TIME %(TI)
AS *ED1(OVERLAY),(000016,PERT)
AS *ED2(OVERLAY),(000016,WWY)
AS *ED3(OVERLAY),(000016,LIBRARY)
AS *ED4(OVERLAY),(000016,CATALOG)
AS *ED5(OVERLAY),(000016,MAA)
EN
IF FAILED(TIME UP),GO TO 2NAP
3NAP IF PRE(*LP),LP,%( *LP),*LP
IF ABS(*LP),GO TO 4NAP
2NAP DP 0,CZAS ZADANIA PRZEKROCZONY
TI 10MINS
RM
GO TO 3NAP
1NAP DP 0,BŁĄD W MACRO NAPRUN:MACRO ZANIECHANE
4NAP
****
```

Rys. 7.2. Makroinstrukcja NAPRUN16



Podczas zajęć dydaktycznych z Analizy i Projektowania Komputerowego Układów Elektronicznych są używane makroinstrukcje NAPY i NAPZ przedstawione odpowiednio na rysunku 7.4 i rysunku 7.5. Służą one do wykonywania pliku programów studenckich umieszczonych jeden za drugim (p. 4.1.3).

```
#ANDRZEJ NAPIERZALSKI
#INSTYTUT ELEKTRONIKI POLITECHNIKI ŁÓDZKIEJ
#MACRO NAPRUN17 SŁUŻY DO URUCHAMIANIA PROGRAMU NAP2
#W WERSJI ZAPISANEJ NA PAKIECIE DYSKOWYM NROOOO17 I WYMAGA
#USTAWIENIA PARAMETROW INSTALACYJNYCH INSTRUKCJAMI
#IP OBJECTQUOTA, 60K
#IP CHAPTERQUOTA, 10K
#IP COREOBJECT, 60K
#IP BACKGROUND, 1
#IP MOPLIMIT, 1
#*****
IF NEG(%;INS(OBJECTQUOTA);-59000),QN 1,CZY PARAMETRY-
-INSTALACYJNE USTAWIONE?
RP FB,CM,OJ
MZ 60000
WHENEVER COMERR,GO TO 1NAP
FIND #NAPS,DA,(000017,PROGRAM NAP2)
RE ON
IF PRE(*CR) AND NOT STR(%"*CR")=(),RV %(*CR)
IF PRE(*CR) AND STR(%"*CR")=(),OL *CRO
IF ABS(*CR),OL *CRO
IF PRE(*LP),AS *LPO,%( *LP)
IF ABS(*LP),OL *LPO
IF PRE(*CR) AND NOT STR(%"*CR")=(),AS *CRO,%( *CR)
IF ABS(TI),TIME 100MINS
IF PRE(TI),TIME %(TI)
AS *ED1(OVERLAY),(000017.PERT)
AS *ED2(OVERLAY),(000017.WWY)
AS *ED3(OVERLAY),(000017.LIBRARY)
AS *ED4(OVEPLAY),(000017.CATALOG)
AS *ED5(OVERLAY),(000017.MAA)
EN
IF FAILED(TIME UP),GO TO 2NAP
3NAP IF PRE(*LP),LF,%( *LP),*LP
IF ABS(*LP),GO TO 4NAP
2NAP DP O,CZAS ZADANIA PRZEKROCZONY
TI 10MINS
RM
GO TO 3NAP
1NAP DP O,BŁĄD W MACRO NAPRUN:MACRO ZANIECHANÉ
4NAP
****
```

Rys. 7.3. Makroinstrukcja NAPRUN17



```

#POLITECHNIKA LODZKA CZERWIEC 1984R
#INSTYTUT ELEKTRONIKI
IF STR(%)=(),GO 8M
IF NEG(%;INS(OBJECTQUOTA);-59000),QN 1,CZY PARAMETRY INSTALACYJNE USTAWIONE?
SP X,VAL(1)
IF PRE(RES),GO 3M
RP FB,CM,OJ
WE COMERR,GO 11M
JT 1000MINS
11M WE COMERR,EXIT
MZ 60000
1M FIND #NAP2,DA,(000016,PROGRAM NAP2)
TI 1000MINS
IF ABS(#DEL),MN ON,DIS&REL&CLOSE
IF ABS(#DEL),MN ON,DELETE
RE ON
OL *LPO,NAP2-WYNIKI
3M IF TRUE,IF STR(%%X())=() OR STR(%%X())>(,),EXIT
IF TRUE,IF EXISTS(%%X()),GO 5M
IF TRUE,DP 0,BRAK %X-GO ZBIORU - %%X()
SP X,VAL(%X+1)
GO 3M
5M IF TRUE,AS *CRO,%%X()
IF TRUE.DP 0,DANE - %%X()
EN
4M IF FAI(ONLINE),GO 2M
IF DEL OR HAL(EX),EXIT
IF HAL(IP),GO 2M
IF HAL(CR),GO 3M
IF HAL(EE),GO 10M
IF HAL(OO),GO 10M
IF MON,GO 10M
IF FAI AND NOT FAI(FILE *CRO EXHAUSTED),GO 1M
SP X,VAL(%X+1)
GO 3M
2M OL *LPO,NAP2-WYNIKI
DP 0,LPO-FAILED
RM
GO 4M
10M RM 20
GO 4M
8M DP 0,MAKRO NAPY URUCHAMIA NAP2 DLA KOLEJNYCH
DP 0,PODANYCH ZBIOROW O DOWOLNEJ NAZWIE
DP 0,WYWOLANIE;
DP 0,(NAPY ZB1[,ZB2]*[,#DEL][,#RES])
DP 0,ZB1 ZB2 - SYMBOLIZUJA NAZWY ZBIOROW
DP 0,#DEL - UMOZLIWIA SKASOWANIE PROGRAMU
DP 0,#RES - PONOWNE WYWOLANIE MAKRA W ZADANIU

```

Rys. 7.4. Makroinstrukcja NAPI służąca do uruchamiania pliku danych studentów podczas zajęć dydaktycznych



```

#POLITECHNIKA LODZKA CZERWIEC 1984R
#INSTYTUT ELEKTRONIKI
IF STR(%A)=(),GO 8M
IF NEG(%;LNS(OBJECTQUOTA):-59000),QN 1,CZY PARAMETRY INSTALACYJNE USTAWIONE?
IF PRE(#RES),GO 3M
RP FB,CM,OJ
WE COMERR,GO 11M
JT 100OMINS
11M WE COMERR,EXIT
MZ 60000
1M FIND #NAP2,DA,(000016,PROGRAM NAP2)
TI 100OMINS
IF ABS(#DEL),MN ON,DIS&REI&CLOSE
IF ABS(#DEL),MN ON,DELETE
RE ON
OL *LPO,NAP2-WYNIKI
3M AS *CRO,D% A(GDR)
DP 0,DANE-D% A
EN
4M IF FAI(ONLINE),GO 2M
IF DEL OR HAL(EX),EXIT
IF HAL(LP),GO 2M
IF HAL(CR),GO 3M
IF HAL(E),GO 10M
IF HAL(OO),GO 10M
IF HAL,GO 9M
IF MON,GO 10M
IF FAI AND NOT FAI(FILE *CRO EXHAUSTED),GO 1M
IF ABS(#NER),ER D% A
SP A,VAL(%A+1)
IF EXISTS(D% A),GO 3M
EXIT
2M OL *LPO,NAP2-WYNIKI
DP 0,LPO-FAILED
9M RM
GO 4M
10M RM 20
GO 4M
8M DP 0,MAKRO NAPZ URUCHAMIA NAP2 DLA KOLEJNYCH
DP 0,ZBIOROW O NAZWIE D123 GDZIE 123 SYMBOLI-
-
DO 0,ZUJE KOLEJNE NUMERY ZBIOROW. BRAK ZBIORU
DP 0,O KOLEJNYM NUMERZE KONCZY OBLICZENIA
DP 0,WYWOLANIE:
DP 0,(NAPZ 123[,#DEL][,#RES][,#NER])
DP 0,#DEL - UMOZLIWIA SKASOWANIE PROGRAMU
DP 0,#RES - PONOWNE WYWOLANIE MAKRA W ZADANIU
DP 0,#NER - ZABRANIA KASOWANIA ZBIOROW

```

Rys. 7.5. Makroinstrukcja NAPZ służąca do uruchamiania pliku danych studentów podczas zajęć dydaktycznych

Makroinstrukcja NAPZ uruchamia program NAP2 dla zbiorów danych o nazwie "Dnr" (np. D1, D25, D985), przy czym numery zbiorów muszą występować kolejno jeden za drugim, tzn. D1, D2, D3 W każdym zbiorze może być umieszczony dowolnie duży plik danych wejściowych.



Wywołanie makroinstrukcji jest postaci

NAPZ nr danych [, # DEL] [, # RES] [, # NER]

gdzie: # DEL - umożliwia skasowanie programu,
 # RES - ponowne wywołanie makroinstrukcji w zadaniu,
 # NER - zaprzeczenie kasowania zbiorów.

Makroinstrukcja ta ustawia samoczynnie profil zadania oraz poziom raportowania, o ile nie użyto parametru # RES, który powoduje pominięcie ustawiania parametrów i wprowadzania programu do pamięci operacyjnej. Jeżeli nie użyto parametru # DEL, to program przez cały czas pozostaje w pamięci operacyjnej i ma przydzielone wszystkie wymagane urządzenia. Nie przerywa również wykonywania się programu wystąpienie instrukcji * END, czy też *** oraz błędów wykonania. W przypadku wystąpienia błędu, program pomija wszystkie kolejne wiersze aż do wiersza * CIRCUIT i rozpoczyna analizę programu następnej grupy studenckiej.

W celu kontroli błędów, które występują w programach (szczególnie na początku semestru), drukarka wierszowa cały czas jest przełączona ON LINE i prowadzący zajęcia może kontrolować wykonywanie kolejnych programów.

W przypadku, gdy aktualnie obliczana wersja danych zawiera błędy lub jest napisana niepoprawnie, to można wstrzymać obliczenia przez wykonywanie przerywania i podanie ciągu instrukcji

AL 8, 20

CU

Spowoduje to zaniechanie liczenia danej wersji danych i pominięcie wszystkich wierszy, aż do pierwszego napotkanego wiersza * CIRCUIT, sygnalizującego początek danych następnej grupy studenckiej.

W celu sprawdzenia stanu obliczeń można w przerwie zapytać o niego instrukcją

WS PROG

W celu zakończenia makroinstrukcji z zachowaniem programu w pamięci operacyjnej należy podać komendę

HT EXIT

CU

natomiast zakończenie makroinstrukcji ze skasowaniem programu z pamięci operacyjnej wymaga podania komendy

DL

CU



Podczas przerwy można sprawdzić, co zawierają dane aktualnie wykonywanego programu poprzez instrukcję

RL *CRO

LF Dnr., LI 20

AS *CRO, Dnr.

W celu wprowadzenia danych z kart do zbioru dyskowego o kolejnym numerze należy wykonać instrukcję operatorską

RO U16, IN :NAP2, Dnr. (TS(WR)), T ///),

gdzie Dnr jest kolejnym numerem pliku danych.

Po zatrzymaniu czytnika i prośbie systemu o ponowne jego przydzielenie należy wykonać instrukcję operatorską:

TE U16

Przykładowe wykonanie cyklu obliczeń przy pomocy makroinstrukcji NAPZ wygląda następująco:

1) wczytanie przez operatora kolejnych plików danych np. D1, D2 i D3. W każdym pliku może znajdować się dowolnie wiele programów studenckich,

2) sprowadzenie potrzebnych zbiorów do PZS'u przez wydanie instrukcji RETRIEVE

RV NAPZ

3) założenie przez operatora pakietu dyskowego (do zajęć dydaktycznych jest wykorzystywany program z dysku 000016),

4) wykonanie wywołania makroinstrukcji dla pliku danych o najniższym numerze

NAPZ 1

Po tym wywołaniu program rozpocznie automatycznie wykonywanie obliczeń i w razie braku ingerencji będzie przyłączał i wykonywał kolejne pliki danych, tzn. 2 i 3. Brak kolejnego pliku zakończy wykonanie obliczeń.

Makroinstrukcja NAPI jest podobna do makroinstrukcji NAPZ. Uruchamia ona program dla kolejnych zbiorów o dowolnej nazwie i jej wywołanie jest postaci

NAPI ZB1 [, ZB2] * [, # DEL] [, # RES]



gdzie: ZB1, ZB2 - kolejne pliki danych o dowolnych nazwach np. ZB1, ZB2,

DEL - umożliwia skasowanie programu,

RES - ponowne wywołanie makroinstrukcji.

W odróżnieniu od makroinstrukcji NAPZ, w tym przypadku zbiory danych nie będą kasowane.

7.2. Obliczenia z wykorzystaniem programu PAUT2

W zależności od stopnia złożoności układu należy korzystać bądź z dużej wersji programu PAUT2, zapisanej wraz z programem NAP2 na pakiecie dyskowym 000005, bądź też z mniejszej wersji, znajdującej się się w zbiorach PZS'u.

```
MZ 50K
LO PAUTBINMK
AS *CRO,DPAUTMK
AS *LPO,WPAUTMK
TI 10MINS
BN
LF WPAUTMK,*LP
BJ
****
```

Rys. 7.6. Makroinstrukcja MPAUTMK służąca do uruchamiania programu PAUT2 w zadaniu drugoplanowym

Makroinstrukcję służącą do uruchamiania zadania drugoplanowego (RUNJOB) z wykorzystaniem programu PAUT2 przedstawiono na rysunku 7.6. Jej wykonanie należy uruchomić komendą

RJL NAZWA ZADANIA, :I16B, MPAUTMK

przy czym zbiór danych powinien znajdować się w zbiorze DPAUTMK, a zbiór wyników będzie uzyskany w zbiorze WPAUTMK. Wszystkie zbiory znajdują się w karcie I16B i ich używanie wymaga oczywiście zezwolenia właściciela tej kartoteki. Wyniki obliczeń będą automatycznie listowane na drukarkę wierszową po zakończeniu wykonywania programu.

7.3. Obliczenia z wykorzystaniem programu MINUIT

W celu wykonania obliczeń przy wykorzystaniu programu MINUIT należy napisać własną procedurę FCN i skompilować ją z programem głównym. W celu zminimalizowania czasu tej kompilacji, program MINUIT znajduje się w postaci półskompilowanej w zbiorze MINUITC (z TRACE 0) i w zbiorze MINUITC 2 (z TRACE 2). Kompilacja programu źródłowego MINUIT od-
bywa się przez wydanie następującej instrukcji:

```
FORTRANCOMP *CRIMINUIT, BIN, *LPLISTMINUIT, COMP L MINUITC
```



Przed kompilacją własnego programu, np. programu z rysunku 6.2, należy utworzyć zbiór dyskowy, w którym zostanie zmagazynowany program binarny, przez wydanie rozkazu:

```
CE BETABIN (*DA, KWORDSØ, BUCK1, INTE3)
```

gdzie BETABIN jest nazwą zbioru binarnego, do którego kompilowany będzie program.

```
RV LESZCZYN, MINUITC, DANEßETA
FORTRANCOMP *CRLESZCZYN, BIN BETABIN, SEMI MINUITC, -
SEMI: LIB. SUBGROUPFSCE, *LPLIST1
LF LIST1, *LP
LO BETABIN
AS *CRO, DANELESZCZ
AS *LPO, WLESZCZ
EN
LF WLESZCZ, *LP
EJ
****
```

Rys. 7.7. Makroinstrukcja MACROMIN służąca do kompilacji i wykonania obliczeń przy pomocy programu MINUIT

Na rysunku 7.7 przedstawiono makroinstrukcję MACROMIN, która powoduje wykonanie kompilacji napisanego przez nas programu (o nazwie LESZCZYN) ze zbiorem binarnym BETABIN i przyłączenia do niego półskompiłowanego programu MINUITC oraz procedur fabrycznych z biblioteki SEM: LIB. SUBGROUPFSCE.

Utworzony program binarny może być następnie wykorzystany do obliczeń: Do czytelnika przydzielono zbiór DANELESZCZ (rys. 6.3), a jako zbiór wynikowy zbiór WLESZCZ, który następnie zostaje wyprowadzony na drukarkę wierszową. Wykonanie makroinstrukcji w zadaniu drugoplanowym (RUNJOB) odbywa się wydaniem rozkazu z końcówki MOP

```
RJ NAZWA ZADANIA, MACROMIN
```

lub z kart

```
RJ NAZWA ZADANIA, :I16B, MACROMIN
```

Jeżeli program jest już poprawny, nie ma potrzeby kompilować go po raz drugi, lecz można użyć makroinstrukcji MACROMIN2, przedstawionej na rysunku

```
RV LESZCZYN, MINUITC, DANEßETA
LO BETABIN
AS *CRO, DANELESZCZ
AS *LPO, WLESZCZ
EN
LF WLESZCZ, *LP
EJ
****
```

Rys. 7.8. Makroinstrukcja MACROMIN2 służąca do wykonania obliczeń przy pomocy programu MINUIT



7.8, która powoduje wykonanie obliczeń zestawu danych za pomocą już utworzonego programu BETABIN.

Wszystkie nazwy programów i zbiorów danych oraz wyników są oczywiście dowolnymi nazwami, nadanymi zbiorom przez indywidualnego użytkownika. Kolejność wykonywania obliczeń przy pomocy programu MINUIT jest następująca:

- 1) napisanie własnego programu zawierającego procedurę FCN,
- 2) utworzenie zbioru binarnego, z którym będzie kompilowany program,
- 3) kompilacja programu z przyłączaniem półskompilowanego programu MINUITC lub MINUITC 2 (TRACE 2) oraz segmentów półskompilowanych z biblioteki naukowej SEMI: LIB.SUBGROUPFSCE,
- 4) wykonywanie obliczeń dla różnych wariantów danych za pomocą utworzonego programu binarnego.



LITERATURA

- [1] Baranowski J., Ramotowski M.: Model wielkosygnałowy tranzystora bipolarnego. Elektronika, Nr 3, 1978.
- [2] Baranowski J.: Półprzewodnikowe układy impulsowe i cyfrowe. WNT, Warszawa 1976.
- [3] Bolek E., Napieralski A.: Symulacja i identyfikacja współczynnika wzmocnienia prądowego w przyrządach bipolarnych. I KKT0iUE, Podlesice 1977.
- [4] Bolek E., Napieralski A.: Analiza procesu przejściowego w tranzystorach bipolarnych z uwzględnieniem procesu samonagrzewania. III KKT0iUE, Stawiska k/Gdańska 1979.
- [5] Bolek E., Napieralski A.: Zmodyfikowany model transportowy tranzystora bipolarnego do celów komputerowej analizy układów elektronicznych. Konferencja Naukowo-Techniczna Instytutu Elektroniki Pł, Jedlicze-Grotniki 1980.
- [6] Bolek E., Napieralski A.: Modified transport model of bipolar semiconductor structures. Międzynarodowa Konferencja "Mikroelektronika", Siofok (Węgry) 1982.
- [7] Borak S., Kłaczak J., Korczak S.: System operacyjny George 3. WNT, Warszawa 1981.
- [8] Borys A., Żurada J.: Modele elementów półprzewodnikowych w jednoczesnej komputerowej analizie stało- i zmiennie-prądowej. Elektronika, Nr 10, 1977.
- [9] Bowers J. C.: SCEPTRE a computer program for circuit and systems analysis. Prentice Hall 1971.
- [10] Bowers J. C., Nienhaus H. E.: Model for high-power SCR, extends range of computer aided design. Electronics, April 1977.
- [11] Bowers J., Shaw G., Baran M.: Circuit analysis program can model many digital ICs automatically. Electronics, May 1975.
- [12] Calahan D.: Projektowanie układów elektronicznych za pomocą maszyny cyfrowej. WNT, Warszawa 1978.
- [13] Chua L. O., Lin P. M.: Komputerowa analiza układów elektronicznych. WNT, Warszawa 1981.
- [14] Dutton R. W.: Bipolar transistor modeling avalanche generation for computer circuit simulation. IEEE Trans. on Electron Devices, June 1975.



- [15] Fan S. P., Hsueh M. Y., Neuton A. R., Pederson D. O.: MOTIS-C: a new circuit simulator for MOS LSI circuits. IEEE Trans. 1971.
- [16] Grobelny M.: Projektowanie układów elektronicznych za pomocą komputerów. WKiŁ, Warszawa 1973.
- [17] Grobelny M., Musiałowski Z., Rodański B., Stanclik J., Szafarz M., Tarczyński W., Wal-szczak B.: Układy elektroniczne. Skrypt PW, Wrocław 1980.
- [18] Getreu I.: Modeling the bipolar transistor. Part 1, 2, 3, Electronics, September 1974, October 1974, November 1974.
- [19] Groch A., Gruszczyński W.: Analiza i projektowanie układów elektronicznych za pomocą maszyny cyfrowej. Elektronika, Nr 11, 1980.
- [20] Hart B. L.: Modeling the early effect in bipolar transistors. IEEE Journal of Solid-State Circuits, February 1983.
- [21] Ho C. W., RUEHLI A. E., Brennan P. A.: The modified nodal approach to network analysis. IEEE Trans. on Circuit and Systems, No. 6, 1975.
- [22] ILIN W. N.: Projektowanie układów elektronicznych przy użyciu maszyn cyfrowych. WNT, Warszawa 1975.
- [23] Kacprzak T., Napieralski A.: Analiza zniekształceń nieliniowych we wzmacniaczu CMOS. VII KKT0iUE, 1984.
- [24] Kacprzak T., Lindberg E.: MOS transistor modeling for NAP2 computer program implementation. Report IT-57 Technical University of Denmark, No. 5, 1981.
- [25] Kamburelis T.: System Odra 1305. Organizacja logiczna i podstawowe dane użytkowe. Wydawnictwo WZE ELWRO, Wrocław 1972.
- [26] Kandel M.: Makromodelowanie cyfrowych układów scalonych w programie komputerowej analizy sieci elektronicznej. NAP2. Praca magisterska, PŁ 1982.
- [27] Kopeć M.: Swobodnoformatowa wersja programu PAUT i jej wykorzystanie do analizy wybranych układów tyrystorowych. Praca magisterska, PŁ 1984.
- [28] Korzec Z., Napieralski A., Żywno M., Leśny K., Leszczyński A., Melon J.: Raport z wykonania pierwszego etapu zlecenia pŁ.: Propozycja programu do analizy układów tyrystorowych - PAUT. PŁ 1979.
- [29] Korzec Z., Napieralski A., Leśny K., Leszczyński A., Melon J.: Raport z wykonania drugiego etapu zalecenia pŁ.: Modelowanie i analiza układów tyrystorowych. Program do analizy układów tyrystorowych PAUT. PŁ 1980.



- [30] Korzec Z.: Układy półprzewodnikowe. WNT, Warszawa 1979.
- [31] Kulka Z., Nadachowski M.: Liniowe układy scalone i ich zastosowania. WKiŁ, Warszawa 1974.
- [32] Laha A. K., Smart D. W.: A Zener diode model with application to SPICE2. IEEE Journal of Solid-State Circuits, February 1981.
- [33] Leśny K.: Rozwiązywanie macierzowych równań układów elektronicznych. Praca magisterska, Pł 1982.
- [34] Lightner M.R., Director S.W.: Multiple criterion optimization for the design of electronic circuits. IEEE Trans. on Circuits and Systems, March 1981.
- [35] Luciński J.: Układy tyrystorowe. WNT, Warszawa 1987.
- [36] Luciński J.: Układy z tyrystorami dwukierunkowymi. WNT, Warszawa 1982.
- [37] Maciejczyk A.: Wpływ modelu tyrystora na dokładność i szybkość obliczeń układów falownikowych. Praca magisterska, Pł 1982.
- [38] Mc Afee L. C.: Optimized numerical models for semiconductor devices. IEEE Trans. on Electron Devices, No. 4, 1976.
- [39] Maciszewski W., Kalisz J., Kazimierski T.: Przegląd programów komputerowej analizy układów elektronicznych. Elektronika, Nr 4, 1976.
- [40] Marshall M., Waller J.: VLSI pushes super CAD techniques. Electronics, July 1980.
- [41] Melon J.: Metody komputerowego formułowania równań układów elektronicznych. Praca magisterska Pł 1981.
- [42] Miłosz Z.: Metody identyfikacyjne parametrów modeli przyrządów półprzewodnikowych dla programu NAP2. Praca magisterska Pł 1982.
- [43] Napieralski A.: Modelowanie półprzewodnikowych struktur czterowarstwowych dla celów analizy numerycznej. Rozprawa doktorska, Pł 1977.
- [44] Napieralski A.: Zastosowanie programu NAP2 do analizy układów tyrystorowych. KNTiE Pł, Jedlicze-Grotniki 1980.
- [45] Napieralski A.: Komputerowa analiza układów tyrystorowych. IV KKTOiUE, Drzonków k/Zielonej Góry 1981.
- [46] Napieralski A., Samujłło M.: Operational amplifier modelling for computer aided design. Międzynarodowa Konferencja "Mikroelektronika 82", Siofok (Węgry) 1982.
- [47] Napieralski A.: Program dla analizy sieci tyrystorowych



- (w języku rosyjskim). Konferencja Krajów Socjalistycznych "INTER-ELEKTRO", Poliana Brashow (Rumunia) 1982.
- [48] Napieralski A.: An operational amplifier macromodeling with application to NAP2. VI ECCTD, Stuttgart (RFN) 1983.
 - [49] Napieralski A.: Digital integrated circuits macromodeling with application to NAP2. International AMSE Summer Conference "Modelling and Simulation", Nice (Francja) 1983.
 - [50] Napieralski A.: Computer aided design of thyristor circuits. International IASTED Conference "Modelling Identification and Control", Innsbruck (Austria) 1984.
 - [51] Napieralski A.: Integrated circuits operational amplifiers macromodeling with application to NAP2. AMSE Review, No. 2, 1984.
 - [52] Napieralski A.: Power semiconductor devices macromodeling with application to NAP2. Yugoslav Conference on Microelectronics, Nis (Jugosławia) 1984.
 - [53] Napieralski A.: Analysis of power semiconductor circuits by using the PAUT2 program. Yugoslav Conference on Microelectronics, Nis (Jugosławia) 1984.
 - [54] Napieralski A.: Computer analysis of the thyristor MOC structure. International Conference on Electronic Circuits "Microelectronics", Prague (Czechosłowacja) 1984.
 - [55] Napieralski A., Kacprzak T.: Model stałoprądowy tranzystora VMOS. VII KKTOiUE, 1984.
 - [56] Pachelski W.: Fortran dla maszyn Odra serii 1300. WNT, Warszawa 1980.
 - [57] Pieruń M.: Zastosowanie uniwersalnych programów analizy komputerowej do symulacji układów elektronicznych. Praca magisterska, Pł 1984.
 - [58] Rabbat N.B., Ryan W.D., Hossain S.Q.: Transient analysis of multiple-input integrated digital structures. IEEE Trans. on Electron Devices, No. 8, 1975.
 - [59] Rabbat N.B., Ryan W.D., Hossain S.Q.: A computer modeling approach for LSI digital structures. IEEE Trans. on Electron Devices, No. 8, 1975.
 - [60] Ramotowski M.: Raport techniczny z wykonania I etapu pt.: Opracowanie metod identyfikacji parametrów modeli elementów półprzewodnikowych. Politechnika Warszawska 1978.
 - [61] Rubner-Petersen T.: NAP2 - a nonlinear analysis program for electronic circuits. User's Manual, 1973.



- [62] Rubner-Petersen I.: NAP2 - application examples. Technical University of Denmark, 1973.
- [63] Rudakowa B., Rudak B., Surma I.: System operacyjny George 3. Skrypt Politechniki Wrocławskiej, Wrocław 1979.
- [64] Samujłło M.: Makromodele wzmacniaczy operacyjnych dla programu komputerowej analizy sieci elektronicznej NAP2. Praca magisterska, Pt 1982.
- [65] Saulnier J.B.: Description of a thermal model library for use in electrical CAD programs. ESASP-146, Bologna (Włochy) 1979.
- [66] Vladimirecsu A., Newton A.R., Pederson D.O.: SPICE version 2G.1. User's Guide, University of California, Berkeley, USA, Oct. 1980.
- [67] Wilamowski B.: Tranzystory energetyczne. Elektronika, Nr 11, 1980.
- [68] Zilka Z.: Parameters identification in a transistor d.c. transport model. Tesla Electronics, No. 3 1975.
- [69] 1620 electronic circuit analysis program (ECAP). User's Manual, April 1970.
- [70] Electro-magnetic transients program (EMTP). User's Manual, Nov. 1977.
- [71] Opis programu NAP2-ODRA. WAT, Warszawa 1977.
- [72] Beurrier Z.: Bibliotheque electronique ASTEC3. Manuel d'Utilisation, Avril 1983.
- [73] Le Faou., Reynaud., Veran.: IMAG III- un programme de simulation de circuits electroniques. Manuel d'Utilisation, 1983.
- [74] Linberg E.: Analysis programs for analog circuits and systems. VI ECCTD, Stuttgart (RFN), 1983.
- [75] Poor H.C., Gummel H.K.: Modeling of emitter capacitance. Proc. IEEE, No. 12, 1969.

