

트랜지스터 회로의 直接入力を 爲한 ECAP 프로그램

(ECAP for the direct input of bipolar transistor)

安 秀 桔*

(Ann, Sou Guil)

要 約

한 회로에서 각 트랜지스터의 파라미터를 한장의 카드에 한꺼번에 수용할 수 있게 하기 위하여 ECAP 콤파일러·프로그램을 修正하였다. 이로서 트랜지스터의 베이스, 에미터 및 콜렉터에 관한 모든 파라미터가 한꺼번에 入力되고 따라서 相互關係 再確認을 爲해서 여러 썬브·루우틴을 거칠 必要도 없어진다. 베이스 및 콜렉터等 2個의 브랜치·카드와 相互關係를 나타내는 카드 代身, 하나의 카드로서 入力시킬 수 있으므로 相當히 머신타임(Machine time)이 節約되고, 한편 트랜지스터 회로에서 等價回路로 옮겨 각 브랜치와 노드를 番號 붙일 必要없이 直接 데이터·카드(회로를 나타내는)를 作成할 수 있게 되었다.

Abstract

A modification of ECAP compiling program for electronic circuit analysis is attempted in order to permit the global input of the parameters of each bipolar transistor in one card.

A considerable machine time reduction is achieved as the new system permits the treatment of three cards concerning base branch, collector branch and transfer characteristics in one, thus avoiding the necessity of calling several subroutines to confirm the identity of branches and their relations.

It also permits one to make the data cards (describing the circuit) without transforming the circuit into the otherwise necessary equivalent circuit.

1. 序 論

電氣 및 電子회로의 計算은 回路網 理論의 急激한 發達로 因해 루프(loop)方程式, 노드(node)方程式, 스테이트 方程式등 여러가지 便利한 方法에 依해서 完壁하게 行하여지게 되었다.

이 聯立方程式은 Kramer의 方法에 依하여 한꺼번에 풀어낼 수 있고, 경우에 다른 便法이 아니고 항상 같은 方法, 다시 말해서 一般的인 方法으로 行할 수 있게 되었다. 이것은 回路 解釋

의 自動化 또는 機械化를 可能하게 해 주었고 한편 Kirchhoff의 인시덴스·마트릭스(incidence matrix)는 機械와 人間사이의 交換이 不可能하였던 回路의 토폴로지의 見地에서의 構造를 代數學의 方法에 依하여 傳達하는 것을 可能하게 해주었다.

예를 들어, (1)式으로 주어지는 A 매트릭스를 생각해 보면,

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 & 0 \end{pmatrix} \quad (1)$$

* 서울大學校 工科大學 電子工學科

Dept. of Electronics Eng., College of Eng. Seoul National University.

各各의 버-텍스(vertex)에 인시덴트(incident) 하는 에지(edge)들이 列로서 나타나고 있는데, 한편 이 다섯개의 에지(edge)에 들어 있는 素子の 立場에서 보면 이 素子が 서로 結線되어 있지 않은, 또는 하나의 버-텍스만이 共通點으로 되어 있는 즉, 自由度를 상실하지 않았을 경우의 全體 Y 매트릭스

$$Y = \begin{pmatrix} y_1 & 0 & 0 & 0 & 0 \\ 0 & y_2 & 0 & 0 & 0 \\ 0 & 0 & y_3 & 0 & 0 \\ 0 & 0 & 0 & y_4 & 0 \\ 0 & 0 & 0 & 0 & y_5 \end{pmatrix} \quad (2)$$

에서 出發하여

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_1 & \circ \\ y_2 \\ y_3 \\ \circ & y_4 \\ y_5 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} y_1+y_4 & -y_4 & 0 & -y_1 \\ -y_4 & y_2+y_4+y_5 & -y_5 & -y_2 \\ 0 & -y_5 & y_3+y_5 & -y_3 \\ -y_1 & -y_2 & -y_3 & y_1+y_2+y_3 \end{pmatrix} \quad (3)$$

와 같이 되며, 다름 · 버-텍스(datum vertex)에 該當하는 列과 行을 버리면

$$Y \text{ transformed} = \begin{pmatrix} y_1+y_4 & -y_4 & 0 \\ -y_4 & y_2+y_4+y_5 & -y_5 \\ 0 & -y_5 & y_3+y_5 \end{pmatrix} \quad (4)$$

가 된다.

따라서 Y와 Y transformed 사이에는 구속조건이 없는(連結되지 않은) 空間에서 구속조건에 의해 自由度를 喪失하여 次元數가 줄어든 空間으로

의 變換된 差異點을 갖고 있는 것이다. (4)로서 表現된 어드미턴스群은 그림 1과 같은 結線을 이루고 있다.

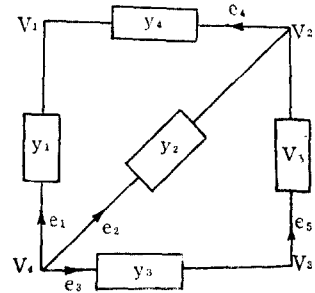


그림 1. (4)式 매트릭스의 實際回路
Fig 1. Realization of Ytransformed

Ytransformed 매트릭스는 노-드方程式에서의 聯立方程式의 個數와 같은 次元의 매트릭스로서 回路의 토폴로직한 面에서의 樣相이 完全히 包含되어 있어서 空間變換에 의해 그래프의 零리터(nullity)만큼 次元數가 줄어들음을 안다.

回路의 랭크(rank)가 커감에 따라 計算이 몹시 번거롭게 될 것은 明白하므로 電子計算機에 의하지 않고서는 에라(error)없이 풀어내기는 힘들고 또한 回路設計에서 最終인 設計가 이루어질때 까지 많은 修正을 行하여야 하는데 그때마다 計算을 다시 하기란 몹시 번거롭다.

그밖에 各 素子が 回路出力을 支配하는 정도를 나타내는 素子 感度비티(sensitivity)計算과 各 素子の 오차한계를 주어 行하는 最惡條件(worst case)의 計算등등 때문에 컴퓨터·에이디드·디자인·테크닉(computer aided design technique)은 必要不可缺의 存在가 되어 버리고 만 것은 主지의 事實이다.

2. E C A P

筆者가 設計에 使用할 機會가 많았던 IBM社의 ECAP(Electromic Circuit Analysis Program)는 DC, AC 및 과도현상의 경우를 다 다루는 巨大한 프로그램인데 트랜지스터를 爲해서는 等價回路를 使用하여 各 에지(edge)의 番號를 割當한 다음 入力카드를 만들어 넣어야 하기 때문에

특수한 設計를 爲해서는 柔軟性이 많아 좋지만, 一般 設計를 爲해서는 트랜지스터를 入力·出力 저항과 β 정도만 알고 等價回路를 거치지 않고 直接 入力카드를 만들 수 있기를 願하는 사람이 많고 또한 이것을 위해 하나의 카드로서 하나의 트랜지스터에 關한 모든 情報를 다 실게 된다면 카드의 수도 적고 머신타임도 줄일 수 있다. 이러한 長點을 考慮하여 ECAP를 修正하게 된 것이다.

ECAP에는 처음 메인·프로그램(main program)이 있고 그 다음 ECA, ECA 01~ECA 09 등 열개의 썬브·루틴이 있어 入力 언어(language)의 整理分類와 情報抽出를 맡고 ECA 19로부터 約 20개의 썬브·루틴은 DC 解析(analysis), ECA 39로부터 約 15개의 썬브·루틴은 AC 解析, 다시 ECA 69로부터 約 30개의 썬브·루틴이 過渡現象 解析을 맡게 되는데 入力 언어(input language)部分을 修正하여 上記 Y transformed 매트릭스만 제대로 形成해 주면 우리가 새로 만든 트랜지스터·데이터·카드(以下 TR 카드라 부름)를 使用할 수 있게 된다.

勿論 ECAP가 가지고 있는 元來의 機能은 그대로 維持해 가면서 需要者(user)가 T 카드를 그대로 使用하기를 願하거나 極端인 경우에는 T 카드 및 TR 카드를 混用해도 使用 可能케 해야 하지만, 카드의 순서에 關해서는 어떠한 制限을 가해도 좋다는 限度로의 修正 可能性을 살펴 보았고, 이에 따라서 修正은 카드의 分離를 擔當하는 ECA 02에서 行하여졌고, 다시 TR 카드를 取擇하기 위해서 썬브·루틴을 하나 만드는 方法으로 解決한다.

3. ECAP 修正과 바이폴라(Bipolar)·트랜지스터의 等價回路 選定

ECAP에서 入力 카드로서 하나의 바이폴라(bipolar)·트랜지스터의 모든 데이터를 한꺼번에 記載한다는 것은 다음과 같은 難點을 먼저 解決해야 한다는 것을 뜻한다.

첫째, 元來 T 카드는 모든 B 카드가 끝난 다음에 들어가게 되는데 새로 使用하게 될 TR 카드 시스템에서는 TR 카드가 들어올 때마다

既存 全 枝路 外에 새로 두개의 枝路씩 늘어나게 되고 거기에 또한 하나의 T 카드의 모든 素子가 附加된다는 점이다. 이것은 TR 카드에 의하여 B 카드와 T 카드를 大幅으로 줄이는데 本 研究目的의 一端이 있는 만큼 在來式에서 3개의 카드에 들어 있는 모든 情報를 1개의 TR 카드로부터 기억장치(memory)에 收容해 가야 한다는 것으로 이 過程에서 新枝路의 命名이 行하여지는 것이어서 表面上 프로그램에서 B 카드가 n 개이고 TR 카드가 m 개라하면 實際에서는 枝路의 數는 다음 式으로 주어진다.

$$\text{枝路의 數} = n + 2m$$

따라서 n+1번째 카드인 TR 카드에 包含되어 있는 베이스 枝路和 콜렉터 枝路는 各各 n+1, n+2라는 枝路番號를 가지게 되는 것이다.

元來 ECAP는 線型回路 解析에 利用할 수 있는 것이기 때문에 모든 線型 等價回路가 使用될 수 있으나, 從屬電壓電源은 許用되지 않는다.

高周波 領域에서는 베이스·스프레드(base spread) 抵抗 R_{bb}' 를 考慮해야 하지만 低周波 領域에서는 4개의 파라미터가 꼭 必要되는 것은 잘 알려져 있다.

低周波에 對한 等價回路는 다음과 같다.

1) 임피던스·파라미터(impedance parameters)

이 等價回路는 그림 2와 같이 入力, 出力의 電流 i_1, i_2 를 獨立 變數로 하고 入力, 出力의 電壓 V_1, V_2 를 從屬變數로 하는 것으로 다음 式으로 나타낸다.

$$\begin{aligned} V_1 &= Z_{11}I_1 + Z_{12}I_2 \\ V_2 &= Z_{21}I_1 + Z_{22}I_2 \end{aligned} \quad (1)$$

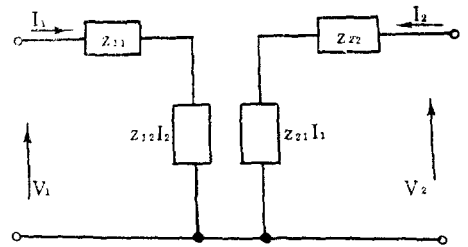


그림 2. 임피던스·파라미터 等價回路
Fig. 2. Impedance Parameter Equivalent circuit

各式의 各項은 디멘손의 見地에서 電壓을 나타내며 크로스·텀(cross term) $Z_{12} I_2$ 및 $Z_{21} I_1$ 은 出力에서 入力 또는 入力에서 出力을 支配하는 것으로서 從屬電壓電源이 必要하기 때문에 ECAP에 使用 不可能하다. 같은 理由때문에 從屬電壓發電機 T等價回路 및 채인·파라미터(A & B) 등도 使用 不可能하다.

ㄴ) y 파라미-터

元來 高周波에서 널리 使用되는 파라미-터로서, 低周波에서 使用 可能할지 모르나 아무에게도 익숙하게 使用되고 있지 않다.

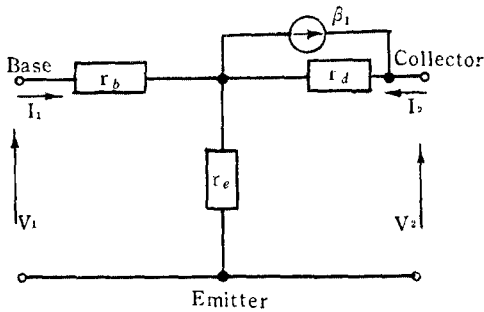


그림 3. T 등가회로
Fig. 3. Current Source Tee Equivalent Circuit

ㄷ) 從屬電流發電機 T等價回路

이 等價回路는 그림 3과 같다. 이 경우에는 4개의 파라미터가 쓰이고 그 크기의 分布가 서로 그다지 멀지 않아서 그 어느 것도 無限大나 0으로 略해 버릴 수가 없다. 따라서 枝路의 數가 많아지고 또한, 노-드의 수도 많아지는데 이는 小型計算機(IBM 1620 or 1130)에서 取扱할 수 있는 回路의 크기가 制限되고 특히 能動素子와 被動素子의 수의 比가 큰 자이래이타(gyrator)나 NIC에 대한 活用時 더욱 만 모델에 比해 不利하다.

1) 하이브리드·파라미터(hybrid parameters) 이 等價回路는 그림 4와 같다.

바이폴라(bipolar)·트랜지스터에 關한 限 이 等價回路에 使用되는 파라미터가 가장 正確한 값을 주기 때문에 널리 使用되고 있다.

變數들은 다음 式으로 關聯된다.

$$\left. \begin{aligned} V_1 &= h_{ie} I_1 + h_{re} V_2 \\ I_2 &= h_{fe} I_1 + h_{oe} V_2 \end{aligned} \right\} \quad (2)$$

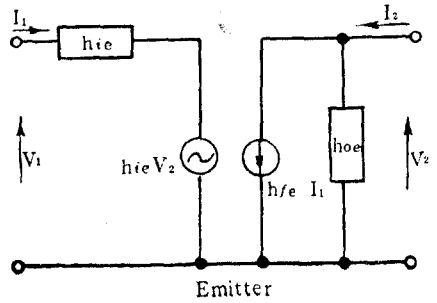


그림 4. 하이브리드 파라미터 등가회로
Fig. 4. Hybrid Parameter Equivalent Circuit

트랜지스터 製作技術의 發達에 따라 이 4개의 파라미-터는 더욱 더 理想에 가까워져서 h_{re} , h_{oe} 등은 (1) 信號 바이폴라·트랜지스터에 있어서 $h_{re} = 1.25 \times 10^{-4}$, $h_{oe} = 75 \text{ micro mho}$ 정도로 작은 값이니 이들을 0으로 간주해도 큰 지장이 없다.

단 負荷가 極히 적은 경우를 위해서 h_{oe} 는 無視하지 않고 두며 디멘손의 統一을 위해서도 그 逆數인 出力抵抗($1/h_{oe}$)을 從屬 電流電源($h_{fe} I_1$)와 並列로 하여 콜렉터 枝路를 形成케 한다.

결국 修正(modify)된 等價回路는 그림 5와 같이 된다.

이 모델에서는 枝路의 數가 2, 노-드의 수가 3으로서 T等價回路의 경우 枝路數 3, 노-드數 4에 比해 有利함을 알 수 있고, 또한 파라미-터의 경우에서 h_{re} 만 0으로 取扱한 것으로서 正確한 數值가 製作會社에 의해서 供給되고 있기 때문에 가장 有利하여 이 모델을 採擇하였다.

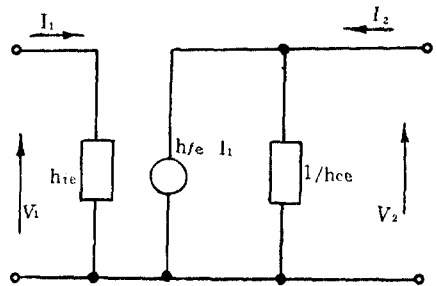


그림 5. 修正된 하이브리드 등가회로
Fig. 5. Modified Hybrid Parameter Equivalent Circuit

4. TR 카아의 構成

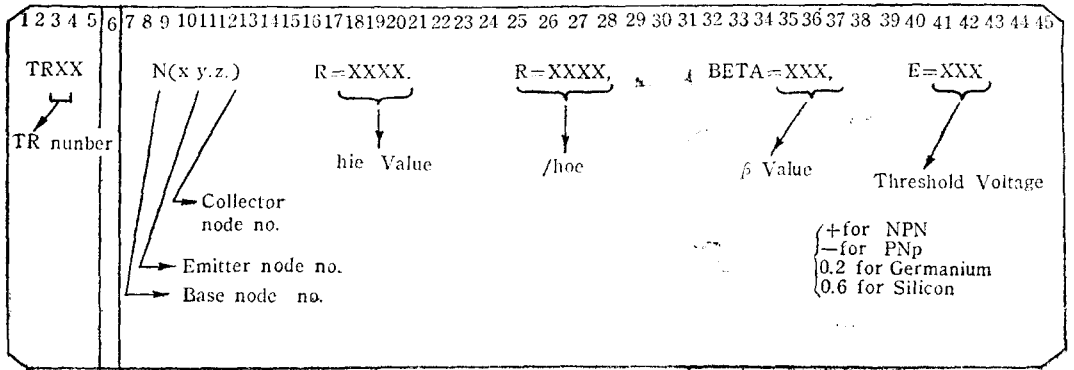


그림 6. TR 카아의 1例
Fig. 6. TR Card

5. TR 카아의 檢出

B나 T카아의 檢出은 썬브·루우틴 ECA(LA 000630/ECAP 360)에서 行해지고 다음에 썬브·루우틴 ECA 02에서 T카아의 處理가 行해지는 것이기 때문에 本 研究에서는 썬브·루우틴 ECA를 修正하여 T文字의 檢出 直後에 ICOL를 1만큼 增加시켜 R文字의 檢出을 行하게 하고 그것이 發見되지 않았을 경우에는 원 줄거리에 따라 갈 수 있게 하고 R文字가 發見되었을 경우에는 TR카아 處理를 위해 새로 作成된 썬브·루우틴(TRSS로 命名)을 거쳐 處理가 끝난 다음 다시 다음 카아를 읽기 위해서 命令文(statement) No. 100(LA 000420/ECA 360)인 리드·命令文(read statement)에 돌아가서 다음 카아를 繼續할 수 있게 하였다. 따라서 ECA에서 B, T, M, S의 分離 以後의 알고리즘(algorithm)은 다음과 같다.(그림 7)

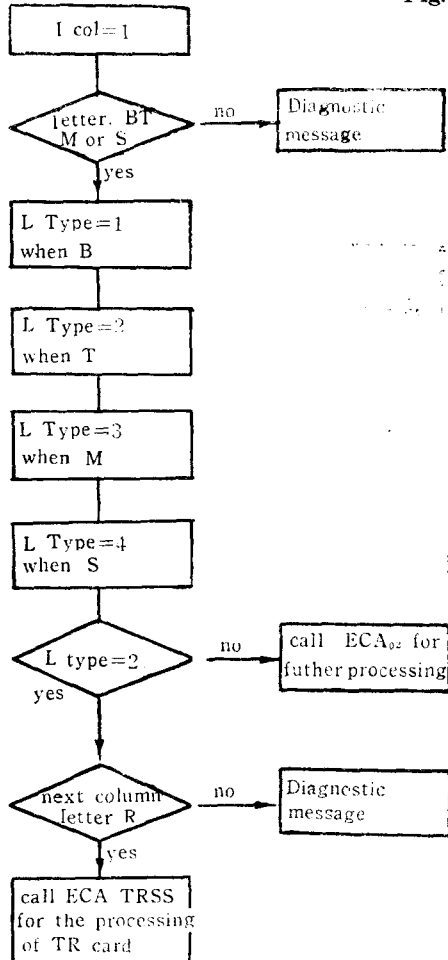


그림 7. TR 카아 處理의 論理과정
Fig. 7. Algorithm for TR Card Input Language

6. 썬브·루우틴 TRSS

썬브·루우틴 ECA(LA 000420/ECA 360)에서 읽힌 카아의 72個의 數文字 데이터(Alphanumeric Data)는 NWORDS(72)라는 알레이(array) 變數로 읽히고 ECA(LA 000490~LA000520/ECA 360)에서 第7欄(column) 以後는 空白(blank)이 除去된 다음 實際 占有(occupy)된 NWORDS(72)最終欄의 番號가 KCOL이라는 變數에 割當

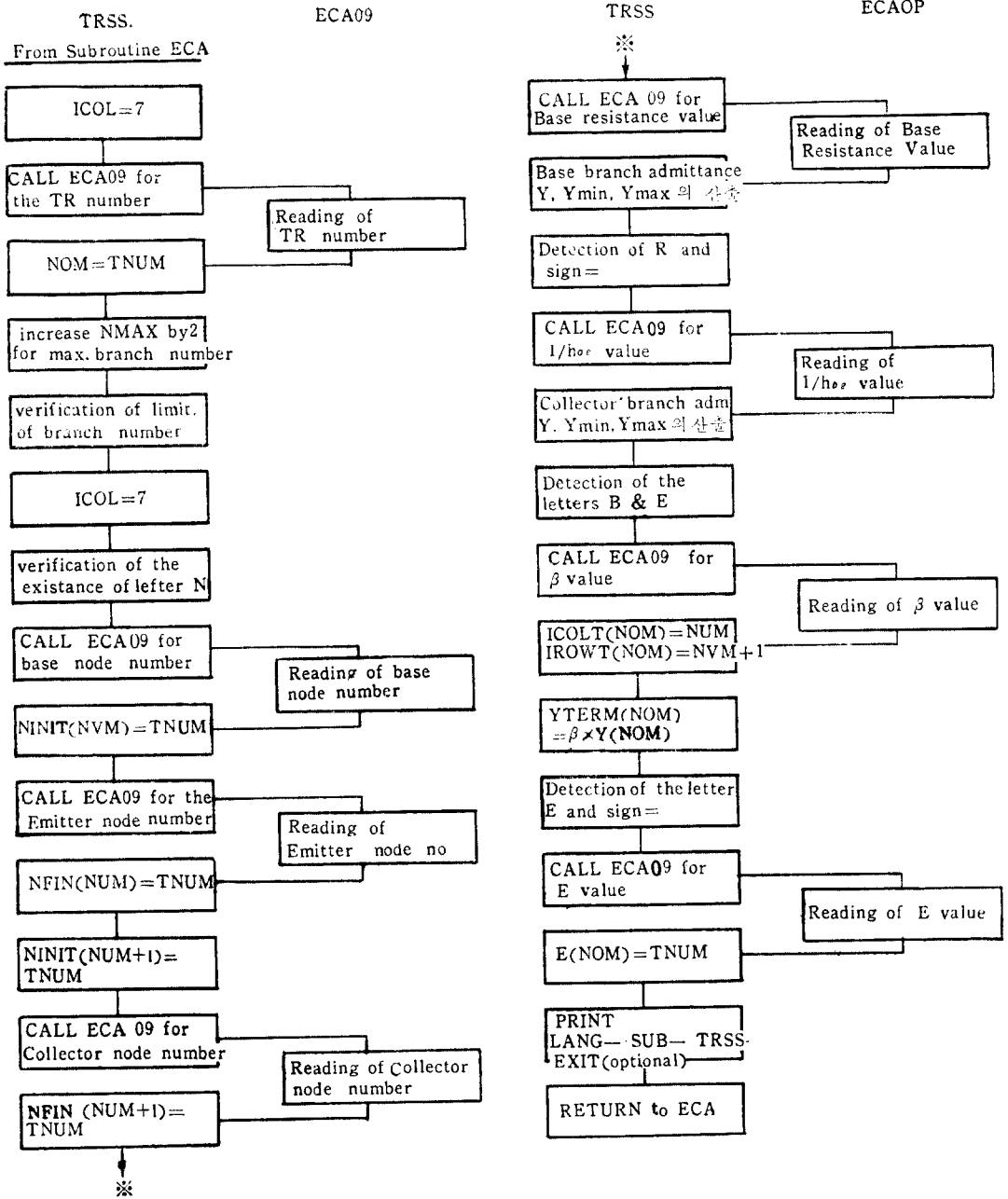


그림 8. 알고리즘
Fig 8. Algorithm

```

// FOR
*ONE WORD INTEGERS
*LIST SOURCE PROGRAM
SUBROUTINE FCA
CARD IMAGE LOADER FOR NETWORK ANALYSIS.
C
C
DIMENSION LIST4(70), LABEL(70),LIST1(5),LIST1(5),NUM1(5),NUM1(5)
DIMENSION LK1(70),LK1(70)
COMMON NMAX,MODE,ATRS5,NBL,NRHR,NRRC,INTR,ITRAC,NSWC,KTO,
ITRST(70)
COMMON F(70),FMIN(70),FMAX(70),AMP(70),AMPBN(70),AMPX(70)
COMMON Y(70),YMIN(70),YMAX(70),YINIT(70),YFIF(70),MODE1(70)
COMMON YTRM(70),YTRMH(70),YTRML(70),INON(70),ICOLT(70)
COMMON FROR1,ISTG,STG,NO,NOF,VERST(20),FSCND(20),VLAST(20)
COMMON INORR(20),NORR(20),NORST(20),ICOT(4),NLTRM,DELTA
COMMON IROST(20),ICOLM(20),FCOL(20),FLMH(20),FLM(20),EPHAT(70),
LAMPHA(70),AREC,MAJOR,FROR2,FROR3,ETIME(5,2),ATIME(5,2)

```

```

C
C THE FOLLOWING VARIABLES ARE USED ONLY IN THE ECAP LANGUAGE PROG.
COMMON INORD(77),NCC(27,20),KLABEL(4),KPNCT(5),INDCT(2,20)
COMMON INPTR(9),NPCD(20),KTYPE(5),NBLNK,NOEXC,ITOL,REQM,IPC
COMMON ITRAL,ECOL,LTYP,ICOL,NOUIT,ITRAN,KO,KS,KFLST,NUM,MC
COMMON M2,M3,KCARD,KC,RP,RT,MAC,INODE,INUM,NOEL,NOF,NOI,KOIC
COMMON FOIVH(20),KOUT(2,10)
DO 10 I=1,70
LIST1(I)=0
LABEL(I)=0
LK1(I)=0
LK1E(I)=0
10 CONTINUE
DO 20 J=1,5
LIST1(J)=0
LISTE(J)=0
NUM1(J)=0
NUM1E(J)=0
20 CONTINUE
IMEGA=0.0
IF (INTRAC) 375,3
2 FORMAT(' LANG MAINLINE=FCA ENTERED KTO= ',I3)
3 WRITE (3,2) KTO
4 GO TO (1,44),KTO
1 CALL ECAP1
WRITE (3,702)
702 FORMAT(I11)
100 READ (2,700) (NWORD (J),J=1,72)
IF (NWORD(1)-NACD(1)) 99,1000,99
99 WRITE (3,701) (NWORD (J),J=1,72)
92 NOEXC=NOEXC+NOUT
KCARD = KCARD + 1
IF (NWORD(1)-NACD(2)) 777,100,777
777 KCOL=6
DO 1002 ICOL=7,77

```

```

PARF 7 A=04
IF (NWORD(1)-NBLNK) 1001,1002,1001
1001 KCOL=KCOL+1
NWORD(KCOL)=NWORD(1)
1002 CONTINUE
IF (NWORD(6)-NACD(1)) 777,776,777
776 IF (NOUT) 500,14,100
14 KCOL=6
IF (N1) 500,109,19
777 NOUT=0
M1=1
DO 24 ICOL=1,5
IF (NWORD(1)-NBLNK) 21,24,21
21 DO 23 LTYPE=1,4
IF (NWORD(1)-KLABEL(LTYPE)) 23,17,23
23 CONTINUE
GO TO 104
24 CONTINUE
IF (KCOL=6) 500,1126,125
1126 374
GO TO 805
104 ITRAN=1
GO TO 126
109 ITRAN=2
126 CALL PCAG
GO TO (500,500,500,100,11,110),ITRAN
125 CALL PCAG
GO TO (5,500,45,150,110,110),ITRAN
46 IF (NOEXC) 500,5004,6667
5004 IF (LTYPE) 5006,5006,5006
5005 GO TO (130,137,138),NTR
5006 NAC=0
47 NAC=NAC+1
NACFL=KTYPE(NAC)
GO TO (210,210,214,216,219,220,220,238,226,228,230),NACFL
210 NOMP=SHDEF
DO 3000 F=1,2000F
DO 2000 I=1,NMAX
IF (INIT(I)-K) 2004,3000,2004
2004 IF (NFI(I)-K) 2006,3000,2006
2007 CONTINUE

```

```

NOEXC=NOEXC+1
WRITE(3,3001) K
3600 CONTINUE
DO 3205 K=1,NMAX
IF(MOD(I,K)) 3205,3202,3205
3202 WRITE(3,3003) K
NOEXC=NOEXC+1
3205 CONTINUE
ITRN=1
IF(NOEXC) 500,5005,4667
C
4667 WRITE(3,2390)
WRITE(3,778) NOEXC
GO TO 219
136 RETURN
137 IF(OMEGA) 1390,1390,1370
1470 RETURN

```

PAGE 8 A=04

```

138 IF(DLTA) 1381,1381,1380
1381 IF(INFORM) 1390,1390,1382
1482 DELTA=1.E-6
1380 RETURN
1390 WRITE(3,1391)
1391 FORMAT(//66H FREQUENCY OR TIME STEP IS IMPROPERLY DEFINED FOR
THIS PROBLEM//)
NOEXC=NOEXC+1
GO TO 4667
214 IF(INFORM) 500,210,218
218 TRAD=1 SHORT
SHORT=OPEN
OPEN=TRADE
GO TO 210
216 GO TO (94,95,500),NTR
94 ITRN=3
GO TO 136
95 ITRN=3
GO TO 137
220 WRITE(3,231)
219 MAC=0
NO=0
WRITE(3,702)
GO TO 100
44 ISFO=0
ITRAC=0
NFOIM=0
WRITE(3,702)
GO TO 47
226 CALL USPO1
GO TO 44
228 CALL USPO2
GO TO 44
230 CALL USPO3
GO TO 44
238 CALL EXIT
C
231 FORMAT (// 25H ILLEGAL INPUT STATEMENT.//)
278 FORMAT (1X,13,24H,ERROR(S) WERE DETECTED.)
2390 FORMAT (40H INPUT ERRORS MADE EXECUTION IMPOSSIBLE.)
3001 FORMAT (//9H NONE NO.14,13H IS MISSING.//)
3003 FORMAT (//11H BRANCH NO.14,13H IS MISSING.//)
500 ITRAN=5
GO TO 110
805 ITRAN=6
GO TO 110
1000 ITRAN=7
NGUIT=0
110 CALL FCA07
GO TO 100
17 ITRAN=1
IF(NWORD(ICGL+1)-NMCD(2,3)) 1713,2846,1713
2846 CALL TRSS
GO TO 100
19 ITRAN=2
1713 CALL FCA02
GO TO (500,500,1313,100,110,110),ITRAN

```

PAGE 9 A=04

```

1313 GO TO (1314,1314,1315,1315,1316,1316,1315,1315,1314),INVAL
1314 CALL FCA03
GO TO 1316
1315 CALL FCA05
1316 GO TO (500,1713,1713,100,110,110),ITRAN
5 CALL FCA04
GO TO 100
722 CALL FCA01
723 CALL FCA01

```

표 2. Subroutine ECA
Table 2. Subroutine ECA

(assign)되어 있고 6번째欄以前은(LA 000600~ LA 000670)에서 따로 取扱되고 썬브·루우틴 TRSS 로 옮겨 올때 ICOL 의 값은 1이다.

처음 部分은 ECAP 의 단 썬브·루우틴의 경우와 共通인 커몬(common)과 이퀴발런스(equivalence)部分을 거친 다음에 需要者(user)가 트레이싱(tracing)을 要求하느냐(NTRAC=1), 안하느냐(NTRAC=0)에 따라 命令文 1에서 命令文 3 을 거치기도 하고, 안거치기도 한다.

다음에 ICOL 이 6번째欄의 左側에 있음을 確認한 다음 뒤 따라온 數値를 읽기 위해서 ECA 09 에 보내진다. 이 過程에서 TR 番號는 整數이므로 NOM 이라는 變數에 割當(assign)한다.

다음에 그 過程까지의 最大 枝路番號인 NMAX 를 2個(베이스 및 콜렉터 枝路) 增加시켜주고 이 數가 ECAP 프로그램의 限定量을 超過하지 않았음을 確認한 다음 다시 트랜지스터의 數 NTRMS 도 增加시켜 주어서 역시 制限에 들어 맞는 것을 確認한다.

다음은 7번째欄 以後를 處理하게 되는데 그 알 고리즘은 그림 8과 같다.

```

// 518
*LIST ALL
*DEF WORD INTEGERS
SUBROUTINE TRSS
DIMENSION LIST(70), LARFL(70), LISTE(5), LISTI(5), NUME(5), NUMI(5)
DIMENSION LIKID(70), LIKIF(70)

COMMON NMAX, NNODE, NTRMS, NARL, NARR, NARC, INTR, NTRAC, NSWCH, KTO,
  IPRINT(10)
COMMON E(70), ENIN(70), EMAX(70), AMP(70), ARFMN(70), ANPIK(70)
COMMON Y(70), YMIN(70), YMAX(70), XINIT(70), XFIN(70), MODE(170)
COMMON YTERM(70), YTRML(70), YTRNL(70), IROW(70), ICOL(70)
COMMON FPROP, ISEF, MSF3, MO, NUMMG, VERST(20), VSCND(20), VLAST(20)
COMMON NDIRN(20), NDIRM(20), NOSTP(20), IYCGT(4), NLTRN, DELTA
COMMON IPRWH(20), ICOLM(20), FLML(20), FLNH(20), FLML(20), EPHA(70),
  IANPHA(70), NRECF, MAJOR, ERROR2, ERROR3, ETIME(5,2), ATIME(5,2)
COMMON NORD(72), NMCD(2,20), KLARL(4), KPUNC(5), INDC(2,20)
COMMON INPTR(9), NACD(20), KTYPF(5), NBLNK, NOEXC, ITOL, NEGIN, IPC
COMMON INVAL, LL, ICOL, LTYPF, KCOL, NOUIT, ITRAN, KO, KS, KELST, NUM, M1
COMMON M2, M3, KCARD, KG, KP, NTP, MAC, HNODE, TRNUM, NOFL, NOE, NOI, NOIC
COMMON EQIVN(20), KOUT(2,10)
DO 10 I=1,70
  LIST(I)=0
  LARFL(I)=0
  LIKID(I)=0
  LIKIF(I)=0
10 CONTINUE
DO 20 J=1,5
  LISTE(J)=0
  LISTI(J)=0
  NUME(J)=0
  NUMI(J)=0
20 CONTINUE
OMEGA=0.0
III=3
IF (NTRAC)3,4,3
  FORMAT(1H ' LANG SUBR=TRSS ENTERED. ITRAN='12)
  WRITE(3,2)ITRAN
  CONTINUE
  ICOL=ICOL+2
  IF (ICOL=6) 331,1717,320
320 WRITE(3,365)
  CALL EXIT
330 FORMAT(1H ' IAN 651)
331 IF (NWORD(1COL)=NB,NV, 1714+325+1714)
332 WRITE(3,362)
333 WRITE(3,363)
334 FORMAT(1H ' IAN 651)
335 CALL EXIT
336 IF (NWORD(1COL)=NB,NV, 1714+325+1714)
337 WRITE(3,364)
338 WRITE(3,364)

```

7. 結果 및 檢討

얼어진 프로그램은 本 프로그램 以外에 ECA 01 로 부터 ECA 09, 그 밖에 ECA 19, TRSS 및 AR ANG 等 썬브·루우틴이고 이중 썬브·루우틴 ARANG 은 各 枝路上의 素子가 제대로 入力되었음을 보여주기 위한 프린트·아웃(print out) 썬브·루우틴으로서 本質的인 重要性은 없다.

實際 例로서 그림 9와 같은 回路에 적용해 보았다.

그 等價回路는 그림 10과 같으며 在來式 ECAP 로서는 데이터·카아드가 그림 11과 같아야 했던 것이다.

筆者가 開發한 이 방식에 對한 데이터·카아드는 그림 12의 上半에 나타나 있다.

實際 계산기를 통한 結果는 그림 12와 같다. 上半에 再生되어 있는 需要者 카아드에서 볼 수 있는 바와 같이 트랜지스터에 關한 모든 情報(information)가 하나의 카아드에 실려 있고, 下半에서 볼 수 있는 바와 같이 이들은 各各 分離되어 베이스 枝路和 콜렉터 枝路가 命名되어 〇番

```

PAGE 2 A-C4
364 FORMAT(1H ' IAN 651)
CALL EXIT
1734 IF (ICOL=6) 1722,325,325
325 WRITE(3,365)
365 FORMAT(1H ' IAN 651)
CALL EXIT
1723 NOMATHIN
IF (MO) 327,1710,327
327 WRITE(3,367)
367 FORMAT(1H ' IAN 671)
CALL EXIT
1710 IF (NMAX=NMI) 42,41,41
41 NMAX=NMAX+2
GO TO 43
42 NMAX=NUM+2
43 CONTINUE
IF (NMAX=70)18,1R,328
328 WRITE(3,368)
368 FORMAT(1H ' IAN 681)
CALL EXIT
18 NTRMS=NTRMS+1
IF (NTRMS=70)40,40,329
329 WRITE(3,369)
369 FORMAT(1H ' IAN 691)
CALL EXIT
401 NOFL=0
NOE=0
NOF=0
NOI=0
NOIC=0
ICOL=6
ICOL=ICOL+1
IF (ICOL=KCOL) 2022,2022,230
330 WRITE(3,370)
370 FORMAT(1H ' IAN 701)
CALL EXIT
2022 IF (NWORD(1COL)=XPUNC(3)) 2222,331,2222
331 WRITE(3,371)
371 FORMAT(1H ' IAN 711)
CALL EXIT
2222 IF (NWORD(1COL)=KPUNC(4)) 2223,332,2223
332 WRITE(3,372)
372 FORMAT(1H ' IAN 721)

```

```

CALL EXIT
2273 IF(NWORD(ICOL)-NRCD(1)) 2R74,333,2R24
333 WRITE(3,373)
373 FORMAT(1H,'NN 73')
CALL EXIT
2R74 KO=0
IF(NWORD(ICOL)-INPTB(1)) 2R25,1721,2R25
2P25 M3=20
GO TO 334
374 WRITE(3,374)
374 FORMAT(1H,'NN 74')
CALL EXIT
1721 IF(MO) 335,76,335
335 WRITE(3,375)

34 ICOL=ICOL+2
IF(NWORD(ICOL)-INPTB(3)) 348,2R33,348
348 WRITE(3,409)
409 FORMAT(1H,'NN409')
CALL EXIT
2R33 ICOL=ICOL-1
IF(NWORD(ICOL)-KPUNC(1)) 349,2R34,349
349 WRITE(3,388)
3R8 FORMAT(1H,'NN 88')
CALL EXIT
2R34 ICOL=ICOL+1
CALL ECA09
Y(NUM)=1.0/TNUM
Y(MINUM)=Y(NUM)
Y(MAXNUM)=1.0/ TNUM
ICOL=ICOL+1
IF(NWORD(ICOL)-KPUNC(3)) 350,2R35,350
350 WRITE(3,383)
3R3 FORMAT(1H,'NN 83')
CALL EXIT
2R35 ICOL=ICOL+1
IF(NWORD(ICOL)-INPTB(3)) 351,2R36,351
351 WRITE(3,405)
405 FORMAT(1H,'NO 405')
CALL EXIT
2R36 IF(NWORD(ICOL+1)-KPUNC(1)) 352,2R37,352
352 WRITE(3,384)
3R4 FORMAT(1H,'NN 84')
CALL EXIT
2R37 ICOL=ICOL+2
CALL ECA09
Y(NUM+1)=1.0/TNUM
Y(MAXNUM+1)=1.0/TNUM
Y(MINUM+1)=Y(NUM+1)
ICOL=ICOL+1
IF(NWORD(ICOL)-KPUNC(3)) 353,2R38,353
353 WRITE(3,389)
3R9 FORMAT(1H,'NN 89')
CALL EXIT
2R38 ICOL=ICOL+1
IF(NWORD(ICOL)-INPTB(2)) 354,2R39,354
354 WRITE(3,394)
3R4 FORMAT(1H,'NN 94')
CALL EXIT
2R39 IF(NWORD(ICOL+1)-INPTB(5)) 355,2R40,355
355 WRITE(3,395)
3R5 FORMAT(1H,'NN 95')
CALL EXIT
2R40 ICOL=ICOL+4
IF(NWORD(ICOL)-KPUNC(1)) 356,2R41,356
356 WRITE(3,396)
3R6 FORMAT(1H,'NN 96')
CALL EXIT
2R41 ICOL=ICOL+1
CALL ECA09
ICOL(TNUM)=NUM
IPDWT(NUM)=NUM+1
NTRMS=PTRMS+1
NRRN=ICOL(TNUM)
YTRM(INUM)=TNUM*Y(NRRN)
YTRM(INUM)=YTRM(INUM)
YTRM(OM)=YTRM(INUM)
ICOL=ICOL+1
IF(NWORD(ICOL)-KPUNC(3)) 357,2R42,357
357 WRITE(3,397)
3R7 FORMAT(1H,'NN 97')
CALL EXIT
2R42 IF(NWORD(ICOL+1)-INPTB(5)) 407,2R43,407
407 WRITE(3,399)
3R3 FORMAT(1H,'NN 93')
2R43 ICOL=ICOL+2
IF(NWORD(ICOL)-KPUNC(1)) 358,2R44,358
358 WRITE(3,398)
3R8 FORMAT(1H,'NN 98')
CALL EXIT
2R44 ICOL=ICOL+1
CALL ECA09
F(NUM)=TRUM
F(MINUM)=TRUM
F(MAXNUM)=TRUM
NOF=NOF-1
NUM=NUM+1
ITRAN=4
ICOL=ICOL+1
IF(NWORD(ICOL)-NRCD(16)) 359,35,359
359 WRITE(3,399)
3R9 FORMAT(1H,'NN 99')
CALL EXIT
35 IF(NWORD(ICOL+1)-NRCD(16)) 408,36,408
408 WRITE(3,400)
400 FORMAT(1H,'NN 00')
CALL EXIT
36 IF(INTRAC) 999R,9999,999R
999R WRITE(3,999)
999R FORMAT(1H,'LANG SUBR=TRSS EXIT')
999R PFUBR
END

```

PAGE 3 A=04

PAGE 4 A=04

表 3. Subroutine TRSS

枝路, 7番 枝路로서 6番 枝路가 7番 枝路를 支配 하고 그 支配정도는 6番 枝路의 어드미턴스에 베타(beta)를 곱한 것으로 $0.0337500 (= 0.2500000E - 03 \times 135)$ 임을 보여 주며, 同時에 各 枝路의 始作 노드番號(NINIT)와 終端 노드番號

(NFIN)와 함께 各 枝路에 所屬된 電源의 電壓도 올바르게 기억(memory)되어 있음을 보여 주고 있다.

이 數值들은 다음 表 1과 같은 變數名으로 기억되어 있으며 썬브·루우틴 TRSS의 리스트

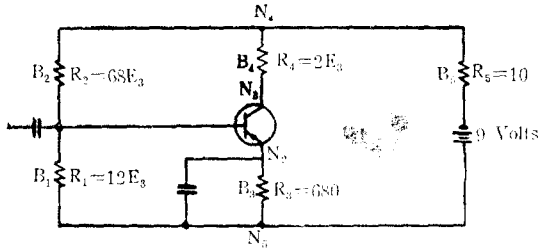


그림 9. 回路例
Fig. 9. Example

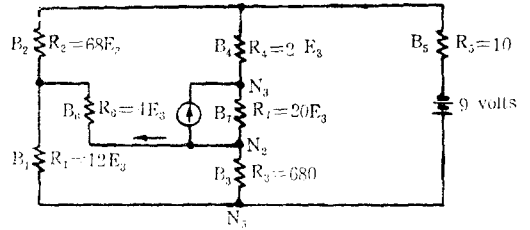


그림 10. 等價回路
Fig. 10. Equivalent Circuit of Example in Fig. 9

DC ANALYSIS

	-15808	-7360	-11200	-7616
B ₁	N(5,), R=12E3			
B ₂	N(1, 4), R=68E3			
B ₃	N(5, 2), R=680			
B ₄	N(3, 4), R=2E3			
B ₅	N(4, 5), R=10, E=9			
TR1	N(1, 2, 3), R=4E3, R=20E3, BETA=135, E=-0.2			

VERIFICATION OF ECAP VARIABLES

BRANCH	NINIT	NFIN	Y	E
	5	1	0.8333334E-04	0.000
	1	4	0.1470588E-04	0.000
	5	2	0.1470588E-02	0.000
	3	4	0.5000000E-03	0.000
	4	5	0.1000000E-00	9.000
	1	2	0.2500000E-03	-0.200
	2	3	0.5000000E-04	0.000
ICOLT IROWT			YTERM	
	6	7	0.03375000	

그림 12. 新方式에서의 ECAP 카아드의 例 및 結果
Fig. 12. Modified ECAP Data Card and the Result

1	2	3	4	5	6	7	8	9	10
						DC ANALYSIS			
B1						N(5, 1), R=12E3			
B2						N(1, 4), R=68E3			
B3						N(5, 2), R=680			
B4						N(3, 4), R=2E3			
B5						N(4, 5), R=10, E=9			
B6						N(1, 2), R=4E3, E=-0.2			
B7						N(2, 3), R=20E3			
T1						B(6, 7), BETA=135			

그림 11. 在來式 ECAP 데이터 카아드
Fig 11. Conventional Data Card Input

(listing)에서도 엿볼 수 있다.

8. 結 論

TR 카아드에 들어 있는 電壓項은 트랜지스터가 D.C 에서 나타내는 문턱(threshold) 電壓으로서 PNP에서 負數이고 NPN에서 正數이며 작은 信號의 경우 저-마니움(germanium)에 있어서 0.2V, 실리콘에 있어서 0.6V 이면 가능하다.

勿論 A.C에 대해서는 E項을 넣을 必要가 없다. 入力言語 처리部分인 ECA~ECA09 및 TRSS를 거쳤을 때 素子の 어드미턴스 β 및 문턱 電壓이 入力되면 나머지 部分에 대해서는 修正할 必要가 없다.

트랜지스터에 관한 모든 데이터의 패키지·入力(package input)이 可能함을 알 수 있고, 이는 特殊한 경우를 除外한 一般 取扱이 簡潔해 지며 等價回路로 바꾼 結線圖가 必要없으며 미실타임의 節約을 가져올 수 있다.

變 數 名	內 容
NINIT(NUM)	始作 Node 番號
NFIN(NUM)	終端 Node 番號
ICOLT(NOM)	支配하는 Branch 番號
IROWT(NOM)	支配받는 Branch 番號
Y(NUM)	各 Branch의 Admittance
E(NUM)	各 Branch의 電壓
YTERM(NOM)	支配하는 Branch의 Off Diagonal Matrix Element 값 (Admittance와 Beta와의 곱)

表 1. 數値의 變數名 및 內容
Table 1. Variables and what they represent.

參 考 文 獻

- (1) Souguil ANN, Honnkak KIM, "Modifications du Programme ECAP Pour la perforation des resultats", RAPPORT D.C.E. SACLAY No. 389.
- (2) Rancall W. Iensen, IBM ELECTRONIC CIRCUIT ANALYSIS.
- (3) Mark D. Lieberman, PROGRAM TECHNIQUES AND APPLICATIONS(Prentice Hall, Inc., 1968) pp. 1~11. 340~353.
- (4) IBM, IBM 1130 ELECTRONIC CIRCUIT ANALYSIS PROGRAM(ECAP)
- (5) IBM, 1620 ELECTRONIC CIRCUIT ANALYSIS PROGRAM (ECAP) (1620-EE-02X). User's Manual, pp. 1~49. (IBM Corporation, 1965).
- (6) Souguil ANN, "Computer를 이용한 回路設計" 電子工學會 Seminar Print(1971. 11).