

Departamento de Electrónica

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

U. N. R.

Proyecto de Ingeniería

Programa Estimador De Parámetros de Transistores Bipolares y de Efecto de Campo de Juntura

Carrera: Ingeniería Electrónica

Alumno: Carlos E. Christoffersen Legajo: C-2357/4

Directora de Proyecto: Ing. María Isabel Schiavon

Asignatura: A-365

- Marzo 1993 -

Resumen: El trabajo consistió en el desarrollo del Programa Estimador de Parámetros (ProEsp), que es una herramienta para calcular parámetros de transistores bipolares y de efecto de campo de juntura. Luego de la introducción hacemos una descripción general del programa desde el punto de vista del usuario, seguida por los fundamentos teóricos que hicieron posible la realización del programa y cuyo conocimiento permite un mejor aprovechamiento del mismo. A continuación repasamos los algoritmos para la determinación de raíces de funciones usados en el programa para luego describir su estructura interna. En en final damos detalles, sugerencias y observaciones acerca de la utilización del programa.

Contenido

1. Introducción	1
2. Descripción general del sistema	2
2.1. Esquema de funcionamiento	2
2.2. Requerimientos del sistema	5
3. Fundamentos Teóricos	6
3.1. Transistor Bipolar	6
3.1.1. Descripción del Modelo de Gummel-Poon	6
3.1.2. El modelo de Gummel-Poon modificado en SPICE2G.1	11
3.1.3. Cálculo de IS y nF	16
3.1.4. Cálculo de Vaf:	18
3.1.5. Cálculo de Bf, IKf, ISE y nE:	18
3.1.6. Cálculo de rB y rBm:	21
3.1.7. Cálculo de ISC, Br y rC:	22
3.1.8. Cálculo de CjE, VjE, mjE, CjC, VjC y mjC:	24
3.1.9. Cálculo de Tf:	25
3.1.10. Cálculo de Tr:	26
3.2. Transistor de efecto de campo de juntura	26
3.2.1. Cálculo de λ :	28
3.2.2. Cálculo de β :	29
3.2.3. Cálculo de las capacidades CGD0 y CGS0:	30
4. Algoritmos	33
4.1. Método Dicotómico	33
4.1.1. Características generales:	33
4.1.2. Diagrama del algoritmo:	34
4.2. Método de Newton-Raphson	34
4.2.1. Características Generales:	34
4.2.2. Diagrama del algoritmo:	35
5. Implementación	36
5.1. Lenguaje de programación utilizado	36
5.2. Estructura interna	36
5.2.1. PRINC.C	37
5.2.2. GRAF.C	40
5.2.3. Subrutina de grupo general	41
5.2.4. IFORWARD.C	42
5.2.5. VAF.C	42
5.2.6. HFE.C	42
5.2.7. RBASE.C	42
5.2.8. BREV.C	42
5.2.9. CAPBJT.C	42
5.2.10. TFOR.C	43
5.2.11. TREV.C	43
5.2.12. LAMDFET.C	43
5.2.13. BETAFET.C	43
5.2.14. CAPFET.C	43
5.3. Códigos fuente y ejecutable	43
6. Manual del Usuario	44
6.1. Ejemplo	44
6.2. Resumen de comandos	51
6.3. Observaciones y Recomendaciones	51
7. Bibliografía	53

1. Introducción

En los programas de simulación de circuitos como por ejemplo el SPICE, se presentan frecuentemente problemas cuando los circuitos a simular contienen elementos activos y no se dispone de los valores de los parámetros de sus modelos. Aún contando con una biblioteca de modelos para el simulador, no siempre podemos estar seguros que los parámetros contenidos en esta son adecuados para nuestro problema en particular.

Para realizar una simulación de un circuito electrónico es necesario conocer, al menos en forma superficial, los modelos que representan a los elementos que lo componen y los métodos de resolución empleados por el simulador para poder analizar si los resultados son válidos o no, ya que por mejor que sea el simulador, existen siempre limitaciones que pueden eventualmente invalidar los resultados.

El **Programa Estimador de Parámetros** (ProESP) permite determinar los parámetros para transistores bipolares y de efecto de campo de juntura, partiendo de datos que se encuentran normalmente en manuales de componentes. El ProESP también permite, si ya se dispone del modelo (por ejemplo de una biblioteca incluida con el simulador), trazar las curvas del elemento y eventualmente modificar los parámetros si fuera necesario, observando los efectos que eso ocasiona en las características del mismo.

Para poder utilizar el ProESP, no es necesario conocer los modelos de los elementos ni los métodos para determinar sus parámetros, ya que el mismo ofrece las indicaciones necesarias de cómo proceder en cada caso. No obstante para sacar la máxima utilidad (y obtener mayor seguridad) es recomendable tener cierta noción de lo que se está calculando.

2. Descripción general del sistema

2.1. Esquema de funcionamiento

La función principal del programa es calcular parámetros de modelos de transistores bipolares o de efecto de campo de juntura para permitir la simulación eléctrica de circuitos que contengan a estos elementos. Para lograr esto, el programa utiliza los datos del manual del fabricante correspondiente. Veamos el funcionamiento en forma general desde el punto de vista del usuario. Comencemos con el diagrama de flujo del mismo en la fig. 1.

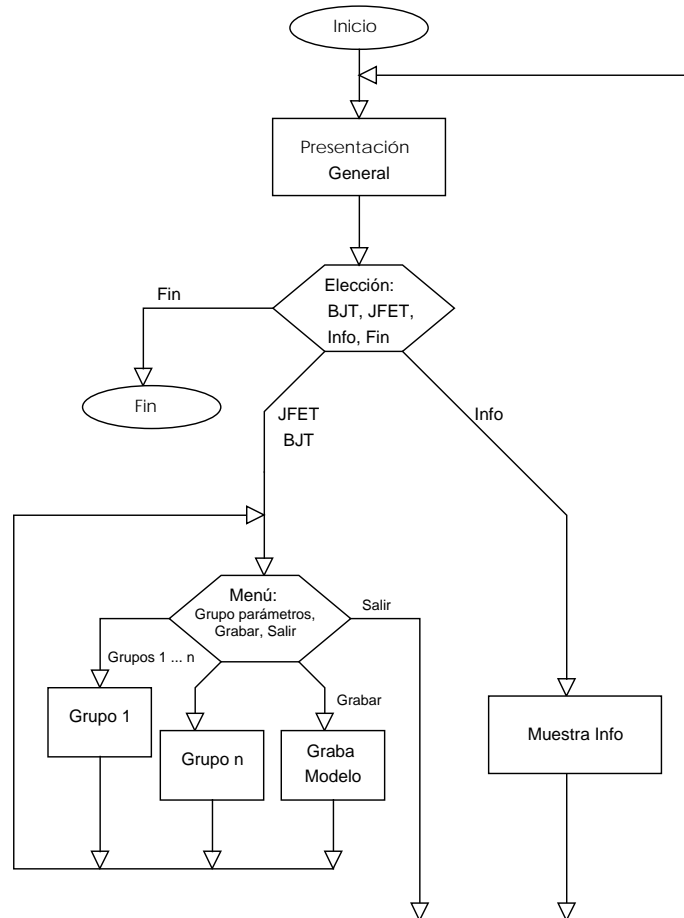


Figura 1

Al cargar el programa nos encontramos con una pantalla de presentación en la que se ofrecen 4 opciones: la primera y la segunda son *calcular parámetros del transistor bipolar y del transistor efecto de campo de juntura* respectivamente, la tercera es *mostrar información acerca del programa* y la última es *salir al sistema operativo*. En el diagrama de la fig. 1 las dos primeras opciones están consideradas en una sola rama porque su funcionamiento es similar desde el punto de vista que consideramos ahora.

Cuando elegimos calcular parámetros, ya sean del BJT o del JFET, tenemos otra pantalla en la que debemos elegir el grupo de parámetros a calcular, o grabar el modelo con los parámetros calculados hasta el momento, o volver al menú (*menú*, en el diagrama) principal. Aquí entenderemos por *grupo de parámetros* un subconjunto del total de

parámetros de un modelo (BJT o JFET), cuyos elementos son calculados en una sola pantalla del programa.

Cuando entramos en el cálculo de parámetros del grupo i ($i=1..n$), se nos presenta la pantalla de cálculo de este grupo y ya es posible comenzar la estimación. Para cambiar de grupo (dentro de los pertenecientes a un tipo de transistor, i.e., BJT o JFET) no es necesario volver al *menú* sino que es posible ir 'deslizándose' entre las pantallas de los distintos grupos como se muestra en la fig. 2. Como vemos es posible ir al grupo anterior, al grupo siguiente o volver al *menú* del tipo de transistor que corresponda.

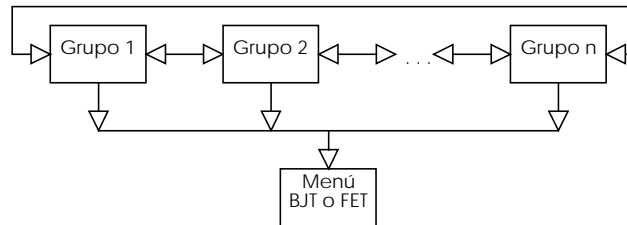


Figura 2

Pasemos ahora a ver las distintas secciones de la pantalla de cálculo para cada grupo mediante el ejemplo de la fig. 3. Aunque tienen pequeñas diferencias entre sí, todas siguen el mismo esquema general.

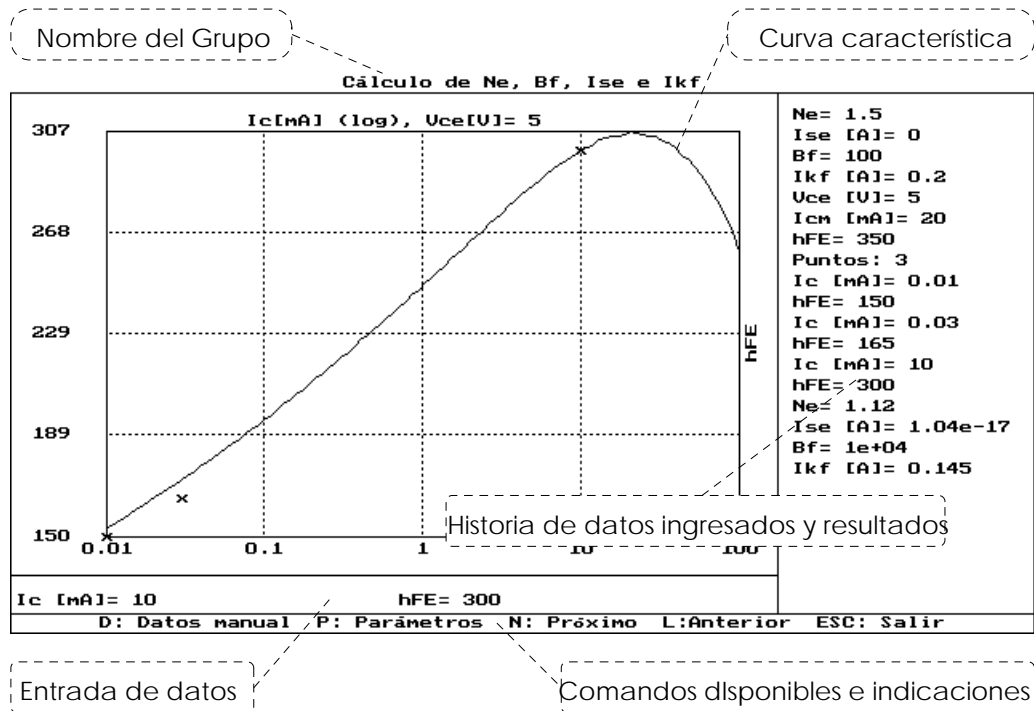


Figura 3

En el *nombre del grupo* figuran los parámetros que permite estimar esta pantalla (en este caso N_c , I_{se} , etc.). En la parte inferior tenemos una línea de *comandos e indicaciones*, en este caso hay comandos (letra N: ir al próximo grupo, ESC: salir, etc), pero podría haber indicaciones respecto al rango o el orden magnitud de los parámetros a ingresar, o preguntas acerca de los datos que disponemos. Cuando es necesaria la entrada de datos numéricos el cursor se sitúa en la zona *entrada de datos*. Estos datos numéricos pueden ser datos del manual o también los parámetros a estimar, ya que es posible asignarle valores arbitrarios mediante el comando **P**. A medida que estos números se van entrando quedan

registrados en el sector *historia de datos ingresados y resultados*, seguidos de los resultados correspondientes. Además, cuando se ingresan puntos pertenecientes a la *curva característica*, estos se marcan con una 'x', para poder compararlos con la curva calculada en base a los parámetros estimados (esta gráfica se actualiza automáticamente con cualquier cambio en los parámetros o en los puntos dato).

Con respecto al **orden de los grupos** hay que destacar que por regla general los parámetros calculados en el grupo *i* intervienen en el cálculo del grupo *j*, con $i < j$. Esto significa que si por ejemplo estimamos los parámetros del grupo 3 y luego modificamos los del grupo 1, la curva del grupo 3 puede alejarse de manera significativa de los datos correspondientes. En ese caso puede ser necesario forzar un nuevo cálculo en el grupo 3 (esto se hace con la opción **D**). Considerando lo anterior, en principio es conveniente comenzar por el grupo 1 y luego ir avanzando siempre hacia adelante. Los usuarios más experimentados podrán alterar esa secuencia, ya que hay veces que eso agiliza el trabajo.

Observación: todas las referencias a corrientes y tensiones en las pantallas de cálculo están tomadas considerando que el transistor es NPN si es bipolar, o canal N si es JFET. Si el transistor es PNP o canal P no es necesario considerar el signo en la entrada de datos (el programa tomará solo el valor absoluto) y algunas gráficas quedarán con valores positivos donde (estrictamente) deberían ir negativos; esto es coherente con la entrada del valor absoluto de los datos: se corresponden las posiciones de las 'x' y la curva. Los parámetros serán calculados correctamente en todos los casos.

Volviendo al diagrama de la fig. 1, la opción *grabar* permite crear archivos ASCII con extensión .MOD que pueden ser utilizados como elemento de biblioteca por cualquier simulador eléctrico del tipo SPICE. Estos archivos creados ya contienen las instrucciones necesarias para el simulador, por lo que solo hay que incluirlos en la descripción del circuito a simular. Al seleccionar esta opción se nos pregunta el nombre del transistor (igual al nombre del archivo sin extensión) y el tipo, luego de lo cual se graba el modelo en el directorio actual.

Las opciones *mostrar información acerca del programa*, y *salir* (en el diagrama de la fig. 1) son muy simples por lo que no las comentamos.

Para finalizar una observación sobre la permanencia de los datos y resultados (i.e., los parámetros estimados): estos se mantienen hasta que el usuario los cambie explícitamente o salga del programa. Esto permite por ejemplo calcular el modelo de un transistor, grabarlo en un archivo, luego modificar solo algunos parámetros y grabar el nuevo modelo con **otro nombre** (¡si no se sobrescribe al archivo anterior!), lo que es práctico para transistores que difieren solo en la serie, por ej: BC547-A, BC547-B, etc.

2.2. Requerimientos del sistema

- Corre en cualquier PC IBM compatible bajo DOS 3.0 o mejor.
- Memoria libre mínima requerida: aproximadamente 100 K.
- Soporte display: CGA, EGA, VGA, Hércules.
- Coprocesador matemático: recomendado pero no imprescindible.

3. Fundamentos Teóricos

Describiremos brevemente los modelos que utiliza SPICE tanto para el transistor bipolar como para el de efecto de campo de juntura, para luego ver los métodos para determinar los parámetros más relevantes y las condiciones en las que estos métodos son válidos.

3.1. Transistor Bipolar

3.1.1. Descripción del Modelo de Gummel-Poon

El modelo de Gummel-Poon se puede considerar como una 'extensión' del modelo de Ebers-Moll. Una de las ventajas del modelo de Gummel-Poon es su amplio rango de validez: están contemplados tanto los efectos de baja como de alta corriente. La siguiente descripción es para un transistor NPN. El punto de partida es la corriente de base, que está compuesta por dos componentes: una de continua ($i_{BE} + i_{BC}$), y otra que representa las variaciones de la carga de base con el tiempo (dq_B/dt).

$$i_B = i_{BE} + i_{BC} + \frac{dq_B}{dt} \quad (q1)$$

Analizamos primero las componentes de continua, i_{BE} e i_{BC} . En la componente base-emisor (i_{BE}) pueden considerarse dos contribuciones: i_{BE1} , que representa la corriente a través de un diodo ideal y describe la recombinación de los huecos de la corriente de base en la zona del emisor, e i_{BE2} , que contempla la corriente a través de un diodo no ideal:

$$\begin{aligned} i_{BE} &= i_{BE1} + i_{BE2} = & (q2) \\ &= I_{s1} \left[e^{\left(\frac{v_{BE}}{V_t}\right)} - 1 \right] + I_{sE} \left[e^{\left(\frac{v_{BE}}{n_E \cdot V_t}\right)} - 1 \right], \end{aligned}$$

análogamente, para i_{BC} :

$$\begin{aligned} i_{BC} &= i_{BC1} + i_{BC2} = & (q3) \\ &= I_{s3} \left[e^{\left(\frac{v_{BC}}{V_t}\right)} - 1 \right] + I_{sC} \left[e^{\left(\frac{v_{BC}}{n_C \cdot V_t}\right)} - 1 \right] \end{aligned}$$

con $V_t = kT/q \approx 25 \text{ mV}$ a 300K.

Para tensiones base-emisor medias y grandes predomina la parte ideal. El diodo no ideal modela los efectos predominantes con pequeñas tensiones. A esa división se debe el decrecimiento de la ganancia de corriente $\beta = i_C/i_B$ del transistor en el modelado de pequeñas corrientes de colector (fig. q1).

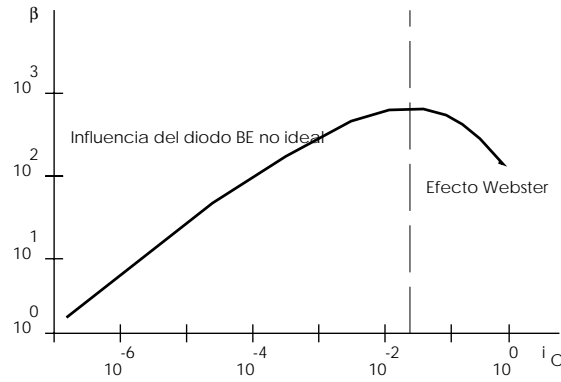


Figura q1

Se puede hacer el mismo análisis para la corriente i_{BC} en funcionamiento inverso, con lo cual la distribución de las componentes de las corrientes en funcionamiento directo e inverso es parecida en la forma de operar a la del conocido modelo de Ebers-Moll.

Para la carga de la base vale:

$$q_B = Q_{B0} + q_C + q_E + B \cdot T_f \cdot I_f + T_r \cdot I_r \quad (q4)$$

Aquí Q_{B0} es la carga de la base cuando no se aplica tensión directa. Las q_C y q_E describen las cargas de transición de las junturas BC y BE.

$$q_C = \int C_{SBC}(v_{BC}) \cdot dv_{BC} \quad (q5)$$

Con esto se tiene en cuenta en el modelo de Gummel-Poon la influencia de v_{BC} en la carga de la base, i.e., el efecto Early (fig. q2).

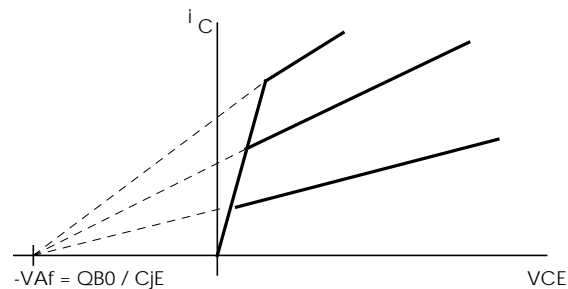


Figura q2

La carga de transición q_E es análoga a q_C en funcionamiento inverso y modela el efecto Early para ese caso:

$$q_E = \int C_{SBE}(v_{BE}) \cdot dv_{BE} \quad (q6)$$

La capacidad C_{SBE} está definida bajo la suposición de una tensión en la juntura menor que la tensión de difusión V_{JE} (V_{JC}):

$$C_{SBC} = \frac{C_{JC}}{(1 - u_{BC}/V_{JC})^{m_{JC}}} \quad (q7)$$

$$C_{SBE} = \frac{C_{JE}}{(1 - u_{BE}/V_{JE})^{m_{JE}}} \quad (\text{q8})$$

El comportamiento de la capacidad de transición C_{SBC} (C_{SBE}) frente a la tensión u_{BC} (u_{BE}) se muestra en la figura q3.

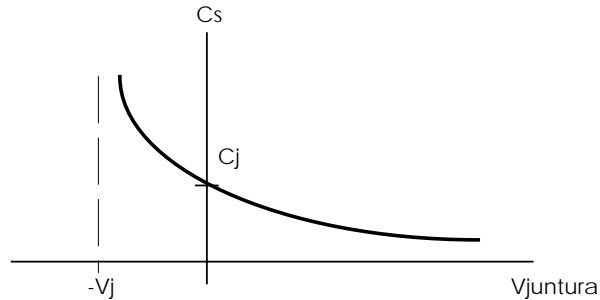


Figura q3

Los dos últimos términos en la ec. q4 describen los efectos de difusión. Allí las corrientes i_f e i_r son componentes de la corriente principal:

$$i_{CE} = i_f - i_r \quad (\text{q9})$$

Resultan:

$$i_f = I_S \frac{Q_{B0}}{q_B} \left[e^{\left(\frac{v_{BE}}{V_T}\right)} - 1 \right] \quad (\text{q10})$$

$$i_r = I_S \frac{Q_{B0}}{q_B} \left[e^{\left(\frac{v_{BC}}{V_T}\right)} - 1 \right] \quad (\text{q11})$$

en las cuales I_S (del orden de 10^{-15}A) corresponde a la intersección de la asintota de $i_C(u_{BE})$ con el eje de las ordenadas en una gráfica semilogarítmica (mirar más adelante, fig. q6). El término T_{fif} describe el incremento de la carga de la base con el aumento de la corriente de colector y con esto el decrecimiento de β (efecto de Webster). El factor de proporcionalidad B representa el cociente al cuadrado del ancho eléctrico (eficaz) de la base (W_{eff}) y el ancho geométrico (W_B):

$$B = \left(\frac{W_{eff}}{W_B} \right)^2 \quad (\text{q12})$$

El término $T_r I_r$ tiene influencia en el caso de saturación del transistor y representa el consiguiente aumento de la carga de la base.

Si se normaliza la carga de la base q_B de la eq. 4 sobre Q_{B0} , se obtiene:

$$N_{qB} = \frac{q_B}{Q_{B0}} = \quad (\text{q13})$$

$$\begin{aligned}
&= 1 + \frac{q_C}{Q_{B0}} + \frac{q_E}{Q_{B0}} + B \cdot T_f \cdot I_s \frac{1}{N_{qB} \cdot Q_{B0}} \left[e^{\left(\frac{v_{BE}}{V_t}\right)} - 1 \right] + T_r \cdot I_s \frac{1}{N_{qB} \cdot Q_{B0}} \left[e^{\left(\frac{v_{BC}}{V_t}\right)} - 1 \right] = \\
&= q_1 + \frac{q_2}{N_{qB}} \tag{q14}
\end{aligned}$$

$$q_1 = 1 + \frac{q_C}{Q_{B0}} + \frac{q_E}{Q_{B0}}$$

$$q_2 = B \cdot T_f \cdot I_s \frac{1}{N_{qB} \cdot Q_{B0}} \left[e^{\left(\frac{v_{BE}}{V_t}\right)} - 1 \right] + T_r \cdot I_s \frac{1}{N_{qB} \cdot Q_{B0}} \left[e^{\left(\frac{v_{BC}}{V_t}\right)} - 1 \right]$$

en donde q_1 representa la carga almacenada y las de transición normalizadas y q_2 las capacidades de difusión normalizadas. De la ec. q4 obtenemos:

$$N_{qB} = \frac{q_1}{2} + \sqrt{\left(\frac{q_1}{2}\right)^2 + q_2} \tag{q15}$$

Operando con las ecuaciones 1 a 11:

$$i_B = -i_C - i_E = \tag{q16}$$

$$= C_{SBE} \cdot \dot{v}_{BE} + C_{SBC} \cdot \dot{v}_{BC} + B \cdot T_f \cdot i_f + T_r \cdot i_r + i_{BE} + i_{BC} + (i_f - i_f) + (i_r - i_r)$$

Esta se divide de la misma manera que en el modelo de Ebers-Moll en las dos corrientes restantes del transistor:

$$i_E = -i_f + i_r - i_{BE} - C_{SBE} \cdot \dot{v}_{BE} + \frac{B \cdot T_f}{V_t} \frac{I_s}{N_{qB}} e^{\left(\frac{v_{BE}}{V_t}\right)} \dot{v}_{BE} \tag{q17}$$

$$i_B = i_f - i_r - i_{BC} - C_{SBC} \cdot \dot{v}_{BC} + \frac{T_r}{V_t} \frac{I_s}{N_{qB}} e^{\left(\frac{v_{BC}}{V_t}\right)} \dot{v}_{BC} \tag{q18}$$

Si además se consideran las resistencias extrínsecas r_B , r_C y r_E , entonces los nodos anteriores B, E y C se transforman en B', E' y C', así se obtiene el diagrama de la fig. q4.

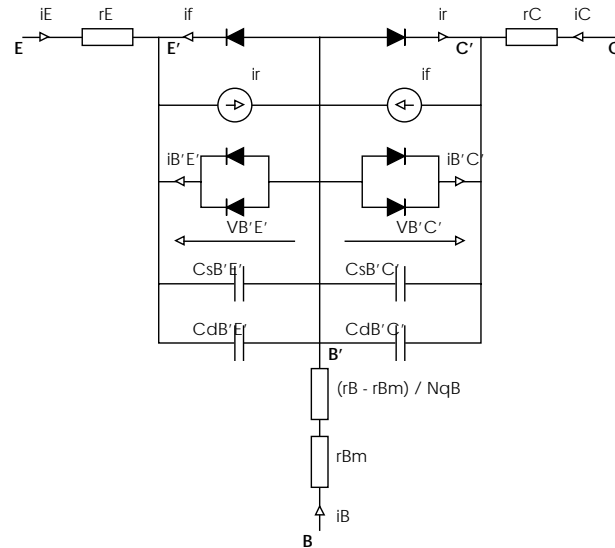


Figura q4

El circuito se comporta (ec. q17) de la siguiente manera: i_f es la corriente a través de un diodo entre E' y C' (ec. q10), mientras que i_r circula conforme al modelo de Ebers-Moll a través del diodo $B'C'$, cuya corriente viene de la fuente entre E' y B' . $i_{B'E'}$ es la corriente de la base resultante en comportamiento directo y circula a través de los diodos que tienen distintas corrientes de saturación y exponentes de la ec. q2.

La expresión $C_{SB'E'} du_{B'E'}$ es la corriente a través de la capacidad de transición de la ec. q8. El término restante en la ec. q17 representa la dependencia de la corriente de la capacidad de difusión de la juntura $B'E'$:

$$C_{DB'E'} = \frac{B \cdot T_f}{V_t} \frac{I_s}{N_{qB}} e^{\left(\frac{v_{B'E'}}{V_t}\right)} \quad (q19)$$

$$C_{DB'C'} = \frac{T_r}{V_t} \frac{I_s}{N_{qB}} e^{\left(\frac{v_{B'C'}}{V_t}\right)} \quad (q20)$$

La resistencia de la base está dividida para tener en cuenta la modulación de la conductancia con altas corrientes:

$$r_{BB'} = r_{Bm} + \frac{(r_B - r_{Bm})}{N_{qB}} \quad (q21)$$

donde r_{Bm} representa la parte óhmica lineal. El segundo término disminuye para tensiones crecientes, i.e., aumenta N_{qB} , como veremos más adelante.

3.1.2. El modelo de Gummel-Poon modificado en SPICE2G.1

A la vista de las ecuaciones características de la corriente de base i_B y la carga de base normalizada N_{qB} , debe deducirse la relación entre el modelo en SPICE y el modelo de Gummel-Poon.

Las dos ecuaciones esenciales en SPICE resultan:

$$\begin{aligned}
i_B &= i_{BE} + i_{BC} + \frac{dq_B}{dt} = \tag{q22} \\
&= \frac{I_s}{B_f} \left[e^{\left(\frac{v_{BE}}{n_f \cdot V_t}\right)} - 1 \right] + I_{SE} \left[e^{\left(\frac{v_{BE}}{n_E \cdot V_t}\right)} - 1 \right] + \frac{I_s}{B_r} \left[e^{\left(\frac{v_{BC}}{n_r \cdot V_t}\right)} - 1 \right] + I_{SC} \left[e^{\left(\frac{v_{BC}}{n_C \cdot V_t}\right)} - 1 \right] + \frac{dq_E}{dt} \\
N_{qB} &= 1 + \frac{1}{V_{Af}} \int_0^{v_{BC}} \frac{du}{\left(1 - \frac{u}{V_{JC}}\right)^{m_{JC}}} + \frac{1}{V_{Ar}} \int_0^{v_{BE}} \frac{du}{\left(1 - \frac{u}{V_{JE}}\right)^{m_{JE}}} + \\
&\quad + \frac{1}{I_{Kf}} \frac{I_s}{N_{qB}} \left[e^{\left(\frac{v_{BE}}{n_f \cdot V_t}\right)} - 1 \right] + \frac{1}{I_{Kr}} \frac{I_s}{N_{qB}} \left[e^{\left(\frac{v_{BC}}{n_r \cdot V_t}\right)} - 1 \right] \tag{q23}
\end{aligned}$$

Pero estas ecuaciones serán utilizadas solo en DC, como veremos luego.

La comparación de la ec. q22 con las 1, 2 y 3 muestra la concordancia directa del modelado de la corriente i_B en ambos casos. Solamente que en SPICE los factores exponenciales de los diodos fueron modelados de forma más flexible. La relación entre los coeficientes I_s , B_f y B_r con I_{S1} e I_{S2} se muestra en la *tabla 1*. Ahí se pueden interpretar B_f y B_r como ganancias de corriente ideales para comportamiento directo e inverso respectivamente. Esto se aplica en especial cuando queremos modelar β independiente de la corriente como en el modelo de Ebers-Moll. Para eso hay que tomar $I_{SE}=0$ (se desprecia el efecto de baja corriente) y $N_{qB}=1$ (se desprecia el efecto de alta corriente). Lo último significa tomar $I_{Kf} \rightarrow \infty$. Los parámetros I_{SE} e I_{Kf} tienen ese valor por defecto.

Gummel-Poon	Tabla 1	SPICE
I_{S1}	Corriente de saturación del diodo ideal BE	I_s/B_f
I_{S3}	Corriente de saturación del diodo ideal BC	I_s/B_r
Q_{B0}/C_{JC}	Tensión Early en funcionamiento inverso	V_{Ar}
Q_{B0}/C_{JE}	Tensión Early en funcionamiento directo	V_{Af}
$Q_{B0}/B \cdot T_f$	Corriente de codo directa	I_{Kf}
Q_{B0}/T_r	Corriente de codo inversa	I_{Kr}
-	Factor de corrección para ajustar la curva del diodo BE ideal	$n_f \approx 1$
-	Idem para el diodo BC	$n_r \approx 1$

Para comparar con N_{qB} de Gummel-Poon:

$$\begin{aligned}
N_{qB} = \frac{q_B}{Q_{B0}} = & 1 + \frac{C_{JC}}{Q_{B0}} \int_0^{v_{BC}} \frac{du}{\left(1 - \frac{u}{V_{JC}}\right)^{m_{JC}}} + \frac{C_{JE}}{Q_{B0}} \int_0^{v_{BE}} \frac{du}{\left(1 - \frac{u}{V_{JE}}\right)^{m_{JE}}} + \\
& + \frac{B \cdot T_f}{Q_{B0}} \frac{I_s}{N_{qB}} \left[e^{\left(\frac{v_{BE}}{n_f \cdot V_t}\right)} - 1 \right] + \frac{T_r}{Q_{B0}} \frac{I_s}{N_{qB}} \left[e^{\left(\frac{v_{BC}}{n_r \cdot V_t}\right)} - 1 \right] \quad (q24)
\end{aligned}$$

También aquí se ve la diferencia entre los coeficientes que se muestra en la *tabla 1*.

La descripción del transistor bipolar en SPICE se divide también en un modelo para el ensayo de continua y otro para el de alterna. Mientras que, como dijimos, para DC se emplea la N_{qB} de la ec. q23, en el caso de AC se emplean las capacidades de la ec. q24. Con eso se invierte el problema, hay que determinar la carga de la base en el caso DC. Para esto son suficientes los 4 parámetros explicados en la *tabla 1*, V_{Af} , V_{Ar} , I_{Kf} e I_{Kr} , que son relativamente simples de determinar.

Las ecuaciones necesarias para la corriente i_C en gran señal resulta de las ec. q18, 10, 11 y 13, pero con los cambios de la *tabla 1*:

$$\begin{aligned}
i_C = i_f - i_r - i_{BC} = & \quad (q25) \\
= \frac{I_s}{N_{qB}} \left[e^{\left(\frac{v_{BE}}{n_f \cdot V_t}\right)} - e^{\left(\frac{v_{BC}}{n_r \cdot V_t}\right)} \right] - \frac{I_s}{B_r} \left[e^{\left(\frac{v_{BC}}{n_r \cdot V_t}\right)} - 1 \right] - I_{SC} \left[e^{\left(\frac{v_{BC}}{n_c \cdot V_t}\right)} - 1 \right]
\end{aligned}$$

La carga N_{qB} se obtiene de resolver la ec. q23 para N_{qB} en forma análoga a la ecuación q15:

$$N_{qB} = \frac{q_{1s}}{2} + \sqrt{\frac{q_{1s}}{4} + q_{2s}} \approx \frac{q_{1s}}{2} \left(1 + \sqrt{1 + 4 \cdot q_{2s}} \right) = N_{qB_{SPICE}} \quad (q26)$$

De la ec. q14:

$$q_{2s} = \frac{I_s}{I_{Kf}} \left[e^{\left(\frac{v_{BE}}{n_f \cdot V_t}\right)} - 1 \right] + \frac{I_s}{I_{Kr}} \left[e^{\left(\frac{v_{BC}}{n_r \cdot V_t}\right)} - 1 \right] \quad (q27)$$

Operando con las ec. q14, 5 y 6 se obtiene:

$$q_{1s} = \frac{1}{1 - \frac{v_{BE}}{V_{Ar}} - \frac{v_{BC}}{V_{Af}}} \quad (q28)$$

El punto de partida para describir el transistor en AC es la ec. q24. Si se toman las cargas de difusión y transición, sin normalizar, juntas se obtiene:

$$q_{Eges} = B_s \cdot T_f \frac{I_s}{N_{qB}} \left[e^{\left(\frac{v_{BE}}{n_f \cdot V_t} \right)} - 1 \right] + C_{JE} \int_0^{v_{BE}} \left(1 - \frac{u}{V_{JE}} \right)^{-m_{JE}} du \quad (q29)$$

$$q_{Cges} = T_r \frac{I_s}{N_{qB}} \left[e^{\left(\frac{v_{BC}}{n_r \cdot V_t} \right)} - 1 \right] + C_{JC} \int_0^{v_{BC}} \left(1 - \frac{u}{V_{JC}} \right)^{-m_{JC}} du \quad (q30)$$

Derivando las ecuaciones 29 y 30 se obtienen las capacidades de difusión y transición del modelo en SPICE:

$$C_{BEges} = \frac{dq_{Eges}}{du_{BE}} = \frac{B_s \cdot T_f}{n_f \cdot V_t} \frac{I_s}{N_{qB}} e^{\left(\frac{v_{BE}}{n_f \cdot V_t} \right)} + \frac{C_{JE}}{\left(1 - \frac{v_{BE}}{V_{JE}} \right)^{m_{JE}}} = C_{DBE} + C_{SBE} \quad (q31)$$

$$C_{BCges} = \frac{dq_{Cges}}{du_{BC}} = \frac{T_r}{n_r \cdot V_t} \frac{I_s}{N_{qB}} e^{\left(\frac{v_{BC}}{n_r \cdot V_t} \right)} + \frac{C_{JC}}{\left(1 - \frac{v_{BC}}{V_{JC}} \right)^{m_{JC}}} = C_{DBC} + C_{SBC} \quad (q32)$$

Las ecuaciones 31 y 32 muestran la conformidad del modelo en SPICE con el de Gummel-Poon.

El factor de proporcionalidad B_s , análogo a B de la descripción anterior, que representa el crecimiento de T_f con la polarización (i.e., u_{BE} y u_{BC}), es en SPICE:

$$B_s = 1 + X_{Tf} \left(\frac{i_f}{i_f + I_{Tf}} \right)^2 e^{\left(\frac{v_{BC}}{1.44 \cdot V_{Tf}} \right)} \quad (q33a)$$

con

$$i_f = I_s \left[e^{\left(\frac{v_{BE}}{n_f \cdot V_t} \right)} - 1 \right] \quad (q33b)$$

Esto explica la particularidad del modelo SPICE, en el que están considerados el efecto Webster (fig. q1), la caída de $\beta = i_C/i_B$ para grandes corrientes de colector a través de I_{Kf} , y en AC el decrecimiento de la f_T y también el aumento del tiempo de tránsito T_f , en forma separada.

Una posibilidad complementaria del modelo de Gummel-Poon en el SPICE es dividir la capacidad BC de la ec. q32 en dos partes. Una parte queda entre B' y C' y la otra entre B y C' (fig. q4). Además se puede modelar la resistencia de base de una manera más complicada que en la ecuación q21, así como también la capacidad con el sustrato necesaria para el modelado de ICs.

En la fig. q5 se muestra el circuito linealizado para el análisis de AC con $X_{JC} = 1$.

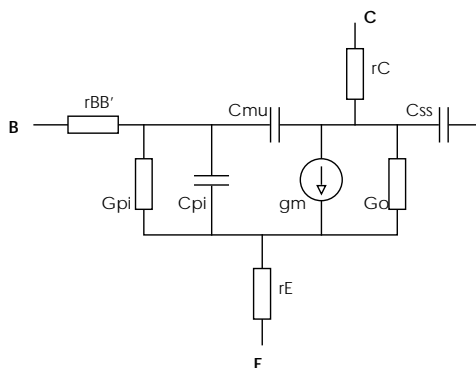


Figura q5

Ese circuito se determina a partir del punto de trabajo del transistor que fue calculado previamente mediante un análisis de continua.

$$G_{pi} = \left| \frac{\partial i_B}{\partial v_{BE}} \right|, \quad G_0 = \left| \frac{\partial i_C}{\partial v_{BC}} \right| + \left| \frac{\partial i_B}{\partial v_{BC}} \right|,$$

$$g_m = \left| \frac{\partial i_C}{\partial v_{BE}} \right| + \left| \frac{\partial i_C}{\partial v_{BC}} \right| + \left| \frac{\partial i_B}{\partial v_{BC}} \right|,$$

$$C_{pi} = C_{BEges}, \quad C_{mu} = C_{BCges}$$

Para finalizar digamos que en el caso de un transistor PNP, todos los parámetros ingresados en la instrucción .MODEL de SPICE tienen el mismo signo y la corrección se efectúa internamente en el simulador.

Pasemos a ver ahora los fundamentos del cálculo de cada uno de los siguientes grupos de parámetros.

I_S : corriente de saturación.

n_f : coeficiente de emisión de I directa.

V_{Af} : Tensión de Early directa.

β_f : β máximo ideal directo.

I_{Kf} : corriente de decaimiento de β con alta I (efecto Webster).

I_{SE} : corriente de saturación de la pérdida B-E.

n_E : coeficiente de emisión de la pérdida B-E.

r_B : resistencia de la base sin polarización.

r_{Bm} : mínima resistencia de base.

I_{SC} : corriente de saturación de la pérdida B-C.

β_r : β máximo ideal inverso.

r_C : resistencia óhmica de colector.

C_{jE} : Capacidad de depleción B-E sin polarización.

V_{jE} : potencial interno B-E.

m_{jE} : factor exponencial P-N de la juntura B-E.

C_{jC} : Capacidad de depleción B-C sin polarización.

V_{jC} : potencial interno B-C.

m_{jC} : factor exponencial P-N de la juntura B-C.

T_f : tiempo de tránsito directo ideal.

T_r : tiempo de tránsito inverso ideal.

3.1.3. Cálculo de I_S y n_F

El siguiente razonamiento es de [1] y [4]. Por ahora, vamos a despreciar la influencia de las resistencias, lo que hará posible el cálculo. Para que esto sea válido deberemos considerar corrientes pequeñas. En zona activa, para lo cual vale lo siguiente, es $v_{BE} \cong 0,6V \ll V_{AR}$ Y $v_{BC} \ll V_{AF}$, $v_{BC} < 0V$. Con eso se simplifcan las ec. q22 , 25 y 26 a:

$$i_B = \frac{I_S}{B_f} e^{\left(\frac{v_{BE}}{n_f \cdot V_T}\right)} + I_{SE} \cdot e^{\left(\frac{v_{BE}}{n_E \cdot V_T}\right)} \quad (q34)$$

$$i_C = \frac{i_S}{N_{qB}} e^{\left(\frac{v_{BE}}{n_f \cdot V_T}\right)} \quad (q35)$$

$$N_{qB} = \frac{1}{2} \left[1 + \sqrt{1 + 4 \cdot \frac{I_S}{I_{Kf}} e^{\left(\frac{v_{BE}}{n_f \cdot V_T}\right)}} \right] \quad (q36)$$

La fig. q6 muestra el comportamiento esquemático de las 2 corrientes en escala semilogarímic. De las ec. q35 y 36:

$$N_{qB} = 1 + \frac{i_C}{I_{Kf}} \quad (q37)$$

para $i_C \ll I_{Kf}$ ($< 100 \mu A$ típica) se puede suponer $N_{qB} \cong 1$ (de la ec. q26). De ahí sigue de la ec. q35:

$$i_C = I_S \cdot e^{\left(\frac{v_{BE}}{n_f \cdot V_T}\right)} \quad (q38)$$

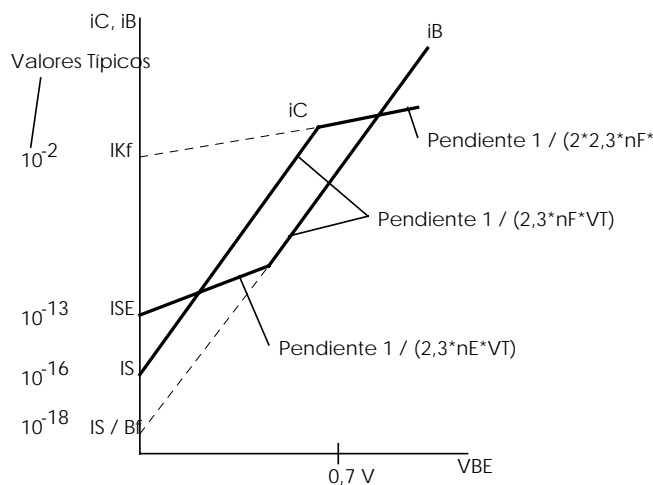


Figura q6

Tomando dos pares de valores para la zona en la que i_C es pequeña, se obtiene (fig. q6):

$$n_F = \frac{v_{BE1} - v_{BE2}}{V_T \cdot \ln\left(\frac{i_{C1}}{i_{C2}}\right)} \quad (q39)$$

con eso se tiene :

$$I_S = i_{C1} \cdot e^{\left(\frac{-v_{BE1}}{n_F \cdot V_T}\right)} \quad (q40)$$

Recordemos entonces que para que lo anterior lleve a buenos resultados debemos utilizar dos puntos en los que i_C sea pequeña (< 1mA a los fines prácticos).

3.1.4. Cálculo de V_{Af} :

Se puede determinar a partir de la fig. q2 y se debe elegir i_C cercana al punto de trabajo propuesto. En la ecuación siguiente hemos supuesto $V_{Af} \gg v_{CE}$:

$$V_{Af} \approx \frac{i_C}{\frac{\partial i_C}{\partial v_{CE}}} = \frac{i_C}{h_{OE}} \quad (q41)$$

3.1.5. Cálculo de B_f , I_{Kf} , I_{SE} y n_E :

El método que sigue fue extraído de [1]. Para la ganancia de corriente = i_C/i_B se obtiene de la ecuación q34

$$\beta = \frac{i_C}{\frac{I_S}{B_f} e^{\left(\frac{v_{BE}}{n_F \cdot V_T}\right)} + I_{SE} \cdot e^{\left(\frac{v_{BE}}{n_E \cdot V_T}\right)}} \quad (q42)$$

de las ec. q35 y 37 se obtiene $v_{BE} = f(i_C)$ y reemplazando en la ec. q42 resulta:

$$\frac{1}{\beta(i_C)} = \frac{1}{B_f} \left(1 + \frac{i_C}{I_{Kf}}\right) + \frac{I_{SE}}{i_C} \left[\frac{i_C}{i_S} \left(1 + \frac{i_C}{I_{Kf}}\right)\right]^{n_E} \quad (q43)$$

La fig. q7 muestra el comportamiento de β con el $\log(i_C)$:

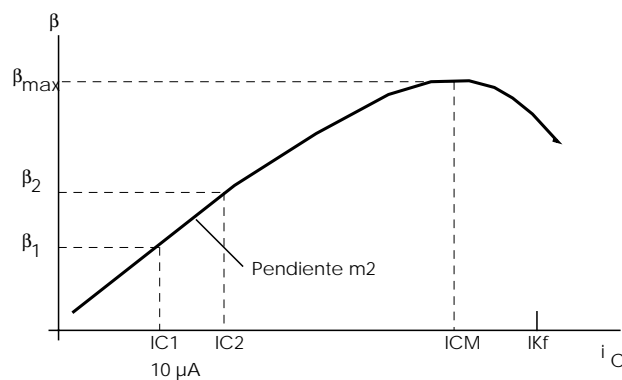


Figura q7

En la figura 7 podemos observar el comportamiento cuasilineal de β para corrientes pequeñas ($i_C \ll I_{Kf}$). De ahí resulta:

$$\frac{1}{\beta(i_C)} = \frac{1}{B_f} + \frac{I_{SE}}{i_C} \left(\frac{i_C}{I_S} \right)^{\frac{n_F}{n_E}} \quad (\text{q44})$$

Con los valores de un transistor típico, se puede estimar:

$$\frac{1}{B_f} < \frac{I_{SE}}{i_C} \left(\frac{i_C}{I_S} \right)^{\frac{n_F}{n_E}} \quad \text{para } i_C > 0.5 \mu A$$

Con eso resulta de la ec. q44:

$$\log(\beta) = \log\left(\frac{I_S^{\frac{n_F}{n_E}}}{I_{SE}}\right) + \left(1 - \frac{n_F}{n_E}\right) \cdot \log(i_C) \quad , \quad 0.5 \mu A < i_C < I_{Kf} \quad (\text{q45})$$

el valor:

$$m_2 = 1 - \frac{n_F}{n_E} = \frac{\log(\beta_2) - \log(\beta_1)}{\log(I_{C2}) - \log(I_{C1})} \quad \left(\frac{\text{Década}}{\text{Década}} \right) \quad (\text{q46})$$

es la pendiente resultante en la fig. q7. Leyendo m_2 de la curva de la hoja de datos se determina n_E :

$$n_E = \frac{n_F}{1 - m_2} \quad (\text{q47})$$

i_C debe estar por lo tanto en el entorno antes mencionado: 10 ... 100 μA .

Ahora B_f : para esto hallamos el máximo de la ec. q43:

$$\left. \frac{\partial \beta(i_C)}{\partial i_C} \right|_{i_C = i_C(\beta_{\max}) = I_{CM}} = 0$$

se obtiene:

$$\frac{I_{SE}}{I_{CM}} \left[\frac{I_{CM}}{I_S} \left(1 + \frac{I_{CM}}{I_{Kf}} \right) \right]^{n_E} = \frac{1}{B_f} \frac{1 + \frac{I_{CM}}{I_{Kf}}}{m_2 \left(2 + \frac{I_{Kf}}{I_{CM}} \right) - 1} \quad (q48)$$

El miembro de la izquierda en la ec. q48 es el de la derecha de la ec. q42 para $i_C = I_{CM}$, $\beta = \beta_{\max}$. Reemplazando, de la ec. q43 se obtiene el coeficiente B_f buscado:

$$B_f = \beta_{\max} \left(1 + \frac{I_{CM}}{I_{Kf}} \right) \left[1 + \frac{1}{m_2 \left(2 + \frac{I_{Kf}}{I_{CM}} \right) - 1} \right] \quad (q49)$$

La corriente I_{Kf} necesaria aquí se puede obtener de la fig. q7 solamente en forma muy inexacta. Tampoco determinándola a partir de la interpretación de la fig. q6 se obtiene un valor suficientemente exacto. De ahí la siguiente aproximación: los parámetros SPICE B_f e I_{Kf} deben ser mayores que cero. Eso impone un límite para I_{Kf} en la ec. q49:

$$I_{Kf} > \left(\frac{1}{m_2} - 2 \right) \cdot I_{CM} = I_{Kf \min} \quad (q50)$$

En la ecuación anterior se supuso $m_2 < 1/3$ y es: $I_{Kf} > I_{CM} > 0$

En primer lugar se hace $I_{Kf} = i_{Kf \min} + \varepsilon$; $\varepsilon \approx 0$, y se determina B_f de la ec. q49 Entonces se puede separar de la ec. q44 con la ec. q46:

$$I_{Kf} > \left(\frac{1}{\beta_1} - \frac{1}{B_f} \right) \cdot I_{C1}^{m_2} \cdot I_S^{n_E} \quad (q51)$$

Así, se determinan los parámetros n_E , B_f e I_{SE} para $\beta(i_C)$ (ec. q43). Solo debe estimarse $I_{Kf \min}$ de la ec. q50. Luego hay que aumentar i_{Kf} hasta obtener el juego de parámetros que más se aproxime a la curva del h_{FE} , utilizando por ejemplo el criterio del error cuadrático medio entre puntos de la curva del modelo y del manual.

Como generalmente la medición de los dos puntos para determinar la pendiente (m_2) suele tener bastante error, debemos considerar además un coeficiente (k) que multiplique a m_2 con el objeto de compensar el efecto de ese error. La idea es calcular los parámetros utilizando el valor de m_2 'ponderado' por distintos valores de k hasta encontrar el juego de parámetros que mejor se ajuste a la curva de datos. Por ejemplo, comenzamos con $k=0,80$; calculamos los parámetros y el error cuadrático medio correspondiente a ellos. Luego hacemos lo mismo para $k=0,81$ y así sucesivamente hasta encontrar el juego de parámetros (para un cierto k) que tenga el mínimo error.

Finalmente para compensar en parte el efecto de haber despreciado V_{Af} , debemos disminuir B_f mediante la siguiente fórmula:

$$B_{f \text{ corregido}} = \frac{B_f}{1 + v_{cb}/V_{Af}}$$

3.1.6. Cálculo de r_B y r_{Bm} :

Si ahora consideramos la gráfica i_C en función de v_{BE} pero para corrientes de colector grandes (figura q8), la influencia de la resistencia de base será importante.

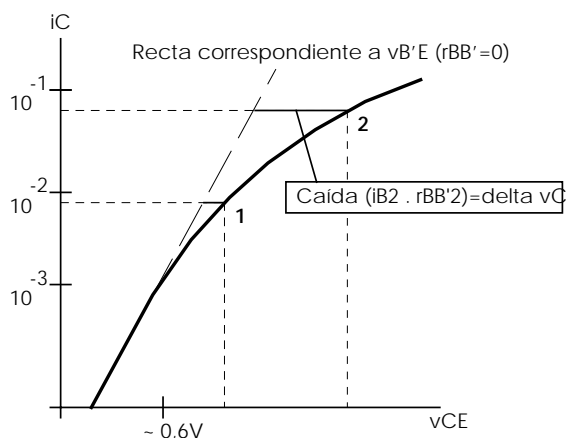


Figura q8

Como el transistor sigue estando en la zona activa, las ecuaciones que gobiernan la recta para $r_{BB'}=0$ (en la gráfica con escala semilogarítmica de la fig. q8) son la 35 para la corriente de colector, en la que reemplazamos v_{BE} por $v_{B'E}$ porque consideramos la r_C , y para N_{qB} tenemos:

$$N_{qB} = \frac{1 + \sqrt{1 + 4 \cdot \frac{I_S}{I_{Kf}} e^{\left(\frac{v_{B'E}}{n_F \cdot V_T}\right)}}}{2 \left(1 - \frac{v_{B'E} - v_{CE}}{V_{Af}}\right)}, \quad (q52)$$

ya que en este caso no despreciamos V_{Af} .

Si tomamos dos puntos como en la fig. q8, podemos calcular numéricamente $v_{B'E}$ para cada par, lo que nos permite a su vez determinar i_{B1} e i_{B2} reemplazando a v_{BE} en la ecuación q34. Luego observando el circuito de la fig. q4, la gráfica de la fig. q8, y la ecuación q21, obtenemos el siguiente sistema de ecuaciones cuyas soluciones son los parámetros buscados, i.e. r_B y r_{Bm} :

$$\begin{cases} \frac{\Delta v_{CE1}}{i_{B1}} = r_{Bm} + \frac{(r_B - r_{Bm})}{N_{qB1}} \\ \frac{\Delta v_{CE2}}{i_{B2}} = r_{Bm} + \frac{(r_B - r_{Bm})}{N_{qB2}} \end{cases}, \quad (q53)$$

en donde los subíndices 1 y 2 corresponden a los puntos considerados.

3.1.7. Cálculo de I_{SC} , B_r y r_C :

En este caso usaremos curvas con el transistor en saturación (estos parámetros no son relevantes en la zona activa), por lo que tendremos que considerar algunas hipótesis adicionales. Partimos de la expresión para N_{qB} en la ec. q26. Para n_r y n_c tomaremos los valores por defecto en

SPICE (esto es, $n_r=1$ y $n_c=2$), y si v_{BC} es cercano a 0,5 V, podemos desprejciar el segundo término de la ec. q27, lo que es equivalente a tomar $I_{Kr} = \infty$ que también es el valor predefinido en SPICE. Por lo demás, seguimos desprejciando a V_{Ar} como lo veníamos haciendo hasta ahora, lo cual es perfectamente válido mientras el transistor no esté en la zona inversa. Además, para simplificar las expresiones siguientes vamos a usar las componentes de las ec. q2 y q3:

$$\begin{aligned} I_{BE1} &= I_S \left(e^{\left(\frac{v_{BE}}{n_F V_T} \right)} - 1 \right), & I_{BE2} &= I_{SE} \left(e^{\left(\frac{v_{BE}}{n_E V_T} \right)} - 1 \right), \\ I_{BC1} &= I_S \left(e^{\left(\frac{v_{BC}}{n_R V_T} \right)} - 1 \right), & I_{BC2} &= I_{SC} \left(e^{\left(\frac{v_{BC}}{n_C V_T} \right)} - 1 \right) \end{aligned} \quad (q54)$$

Así las cosas, tenemos las expresiones para N_{qB} , i_C e i_B sustituyendo en las ecuaciones q22, q25 y q26, y además aplicando las hipótesis mencionadas:

$$N_{qB} = \frac{1 + \sqrt{1 + 4 \cdot \frac{I_{BE1}}{I_{Kf}}}}{2 \left(1 - \frac{v_{BE} - v_{CE}}{V_{Af}} \right)} \quad (q55)$$

$$i_C = \frac{I_{BE1}}{N_{qB}} - \frac{I_{BC1}}{N_{qB}} - \frac{I_{BC1}}{B_r} - I_{BC2} \quad (q56)$$

$$I_B = \frac{I_{BE1}}{N_{qB}} + I_{BE2} + \frac{I_{BC1}}{B_r} + I_{BC2} \quad (q57)$$

Operando con estas ecuaciones, podemos encontrar una expresión para i_C independiente de I_{BC2} y B_r :

$$i_C = \frac{I_{BE2} + \frac{I_{BE1}}{B_f} + \frac{I_{BE1} - I_{BC1}}{N_{qB}}}{1 + \frac{i_B}{i_C}} \quad (q58)$$

en donde suponemos conocida la relación i_B/i_C .

Consideremos ahora dos puntos de la gráfica v_{CEsat} en función de i_C (figura q9) en los que i_C no sea muy grande, digamos del orden de 1 mA, como los puntos 1 y 2. En ese caso podremos desprejciar momentáneamente la resistencia de colector, lo que implica $C \equiv C'$.

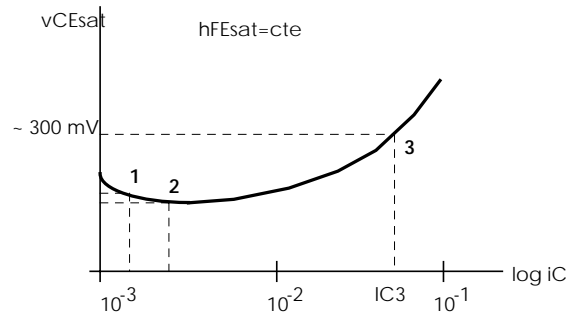


Figura q9

De esos 2 puntos conocemos i_C y v_{CEsat} . Además, conocemos h_{FEsat} , por lo que es muy simple calcular i_B para cada par ya que $h_{FEsat} = i_C/i_B$. Si observamos la ec. q58 solo tenemos todos los datos necesarios para calcular numéricamente $v_{B'E}$ (recordemos que como despreciamos r_C , $v_{B'C} = v_{B'E} - v_{CE}$ en las ec. q54). Luego solo nos queda resolver el siguiente sistema de ecuaciones derivado de las ec. q54 y q57 para obtener B_r e I_{SC} :

$$\begin{cases} i_{B_1} = \frac{I_{BE1_1}}{B_f} + I_{BE2_1} + \frac{I_{BC_1}}{B_r} + I_{SC} \cdot sum_1 \\ i_{B_2} = \frac{I_{BE1_2}}{B_f} + I_{BE2_2} + \frac{I_{BC_2}}{B_r} + I_{SC} \cdot sum_2 \\ sum_j = e^{\left(\frac{v_{B'C_j}}{n_C \cdot V_T}\right)} - 1 \end{cases} \quad (q59)$$

Como tenemos dos ecuaciones independientes con dos incógnitas (sum_j es conocido) podemos obtener así los valores de B_r e I_{SC} .

Para calcular r_C , consideremos el punto 3 de la fig. q9. Como aquí la corriente i_C provoca una caída de tensión no despreciable en r_C , tendremos que una diferencia entre la tensión v_{CE} prevista por el modelo para I_{C3} y la real correspondiente al punto 3. El valor de r_C será:

$$r_C = \frac{v_{CE3} - v_{CE_{\text{modelo sin } r_C}}}{I_{C3}} \quad (q60)$$

La $v_{CE_{\text{modelo sin } r_C}}$ se calcula en forma numérica usando las ecuaciones de q54 a q57, considerando $r_C = 0$, i.e. $C \equiv C'$.

3.1.8. Cálculo de C_{jE} , V_{jE} , m_{jE} , C_{jC} , V_{jC} y m_{jC} :

Este método está en [1]. Tomaremos una capacidad de transición que puede ser la de la juntura de colector o la de emisor. El punto de partida son las ec. q31 y 32. Ellas describen las capacidades de las junturas P-N. Para una juntura en general, tenemos:

$$C_{S_{Bi}} = C_{ji} \left(1 - \frac{v_{Bi}}{V_{ji}}\right)^{-m_{ji}}, \quad i = E, C \quad (q61)$$

$$\ln(C_{S_{Bi}}) = \ln(C_{ji}) - m_{ji} \cdot \ln\left(1 - \frac{v_{Bi}}{V_{ji}}\right)$$

La sustitución por :

$$Y = \ln(C_{S_{Bi}}) \quad , \quad X = \ln\left(1 - \frac{V_{Bi}}{V_{ji}}\right), \quad (q62)$$

$$A = \ln(C_{ji}) \quad , \quad B = -m_{ji}$$

resulta en la relación lineal:

$$Y = A + B \cdot X \quad (q63)$$

Con la ayuda de regresión lineal se ajustan N pares de valores de la curva de la hoja de datos $C_{iB} = f(v_{iB})$; $i = E, C$. Así, resulta:

$$B = \frac{\sum_{k=1}^N x_k y_k - \frac{1}{N} \sum_{k=1}^N x_k \sum_{k=1}^N y_k}{\sum_{k=1}^N x_k^2 - \frac{1}{N} \left(\sum_{k=1}^N x_k\right)^2} \quad (q64)$$

$$A = \frac{1}{N} \left(\sum_{k=1}^N y_k - B \sum_{k=1}^N x_k\right) \quad (q65)$$

El hasta ahora todavía libre parámetro $V_{ji} > 0$ debe adoptarse de tal manera que sea óptima la calidad de la adaptación, esto es r^2 de la ec. q66 debe ser máximo ($0 < r \leq 1$):

$$r^2 = B^2 \frac{\sum_{k=1}^N x_k^2 - \frac{1}{N} \left(\sum_{k=1}^N x_k\right)^2}{\sum_{k=1}^N y_k^2 - \frac{1}{N} \left(\sum_{k=1}^N y_k\right)^2} \quad (q66)$$

Los valores típicos de V_{ji} son cercanos a 0,5 V (tensión de difusión).

Sustituyendo en forma inversa se obtienen los coeficientes buscados a partir de A y B :

$$m_{ji} = -B, \quad C_{ji} = e^A; \quad i = E, C \quad (q67)$$

3.1.9. Cálculo de T_f :

El tiempo de tránsito directo (consultar [1] y [2]) T_f se puede determinar a partir de la fig. q10:

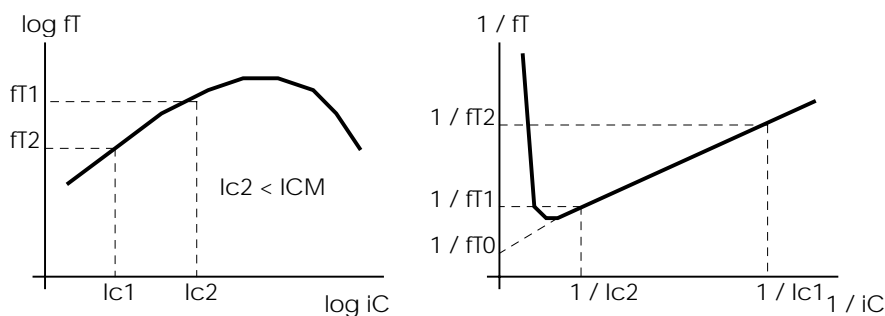


Figura q10

Haciendo eso:

$$f_{T0} = \frac{I_{C1} - I_{C2}}{f_{T1} \cdot I_{C2} - f_{T2} \cdot I_{C1}} f_{T2} \cdot f_{T1} \quad (\text{q68})$$

$$T_f = \frac{1}{2\pi \cdot f_{T0}} - C_{jC} \cdot r_C \quad (\text{q69})$$

3.1.10. Cálculo de T_r :

Podemos utilizar la fórmula q70. Este parámetro es casi directamente proporcional al tiempo de almacenamiento del transistor, en la conmutación. En la zona activa su efecto es despreciable.

Los datos necesarios podemos obtenerlos a partir de t_s (el tiempo de almacenamiento), la corriente de colector i_C y el h_{FEsat} para los cuales se hizo el ensayo, datos comunes para los transistores de conmutación.

$$T_r = \frac{t_s}{B_r \cdot \ln \left(\frac{2 \cdot i_C}{h_{FEsat} \cdot (i_{Ba} + i_C / h_{FEsat})} \right)} \quad (\text{q70})$$

En la fórmula, i_{Ba} es la corriente de base que tendría el transistor si estuviera en la zona activa con la misma corriente de colector, i.e., es igual a i_C dividida por el h_{FE} de la zona activa.

Con los parámetros que hemos calculado hasta ahora es suficiente para modelar el transistor bipolar en las zonas de corte, activa y saturación. El resto puede dejarse en general con los valores por defecto en SPICE.

3.2. Transistor de efecto de campo de juntura

El modelo para el JFET es mucho más simple que el del transistor bipolar, por lo que lo describiremos en forma más resumida. Analizaremos un transistor canal N, igual que en el caso del transistor bipolar, los parámetros incluidos en la instrucción .MODEL se ingresan siempre con el mismo signo, i.e., el correspondiente al tipo canal N, y la corrección se realiza internamente.

Como punto de partida consideremos el circuito del modelo en la fig 1. Las dos resistencias r_d y r_s tienen en cuenta la resistencia óhmica del drenaje y la fuente respectivamente, de forma análoga a las resistencias del modelo del BJT que analizamos anteriormente.

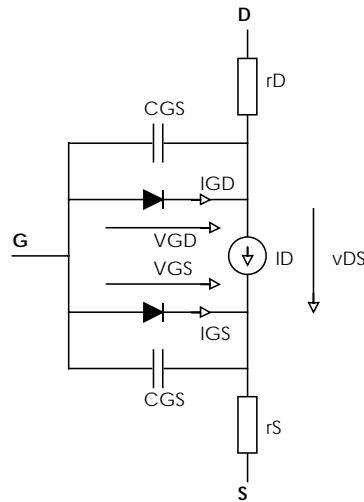


Figura j1

La corriente en la compuerta es la suma de la corriente por los dos diodos (GD y GS):

$$I_G = I_{GS} + I_{GD} = I_s \left[e^{\left(\frac{V_{GS}}{V_t} \right)} - 1 \right] + \left[e^{\left(\frac{V_{GS}}{V_t} \right)} - 1 \right] \quad (j1)$$

La corriente en los otros terminales es:

$$I_D = -I_{drain} - I_{GD} \quad (j2)$$

$$I_S = I_{drain} - I_{GS} \quad (j3)$$

con I_{drain} conforme al modelo cuadrático (para $V_{D'S'} > 0$):

$$I_{drain} = \begin{cases} \beta \cdot (1 + \lambda \cdot V_{D'S'}) \cdot (V_{GS} - V_{TO})^2, & \text{si } 0 \leq V_{GS} - V_{TO} \leq V_{D'S'} \\ \beta \cdot (1 + \lambda \cdot V_{D'S'}) \cdot V_{D'S'} \cdot (2(V_{GS} - V_{TO}) - V_{D'S'}), & \text{si } V_{D'S'} < V_{GS} - V_{TO} \\ 0, & \text{si } V_{GS} - V_{TO} < 0 \end{cases} \quad (j4)$$

Restan las capacidades que son las correspondientes a los dos diodos del modelo. Notemos que en el transistor efecto de campo de juntura no hay capacidades de difusión ya que no hay junturas polarizadas directamente (en funcionamiento normal).

$$C_{GS} = \frac{C_{GS0}}{\left(1 - \frac{V_{GS}}{P_B} \right)^M} \quad (j5)$$

$$C_{GD} = \frac{C_{GD0}}{\left(1 - \frac{V_{GD}}{P_B} \right)^M} \quad (j6)$$

Este modelo de las capacidades está bastante más limitado que en el caso de las capacidades de transición del transistor bipolar, ya que M es fijo igual a 0.5 y P_B es común a las dos capacidades, pero hay que tener en cuenta que en realidad en el transistor de efecto de campo de juntura hay una sola juntura.

El modelo en señal usado para el ensayo de AC en SPICE es el de la fig. j2. Las derivadas se determinan en el punto de operación del elemento.

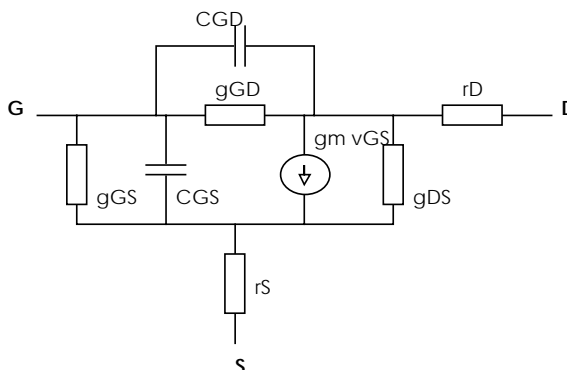


Figura j2

$$\begin{aligned} g_m &= \frac{\partial I_D}{\partial V_{GS}}, & g_{DS} &= \frac{\partial I_D}{\partial V_{DS}}, \\ g_{GS} &= \frac{\partial I_{GS}}{\partial V_{GS}}, & g_{GD} &= \frac{\partial I_{GD}}{\partial V_{DS}} \end{aligned} \quad (j7)$$

Determinaremos los siguientes parámetros:

λ : modulación del largo del canal.

β : coeficiente de transconductancia.

C_{GD0} : capacidad P-N entre G y D sin polarización.

C_{GS0} : capacidad P-N entre G y S sin polarización.

3.2.1. Cálculo de λ :

λ se determina a partir de la pendiente de la zona de saturación de la familia de curvas de salida en la figura j3. Como siempre, es conveniente elegir los puntos cercanos a la zona de trabajo del elemento.

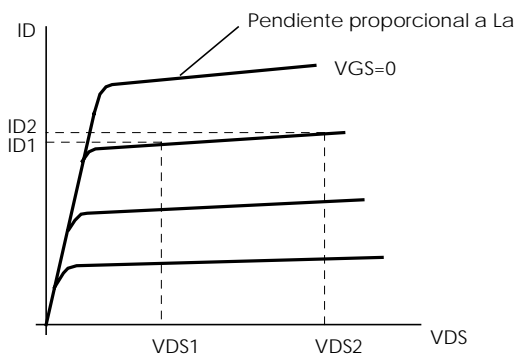


Figura j3

$$\lambda = \frac{I_{DS2} - I_{DS1}}{I_{DS1} \cdot V_{DS2} - I_{DS2} \cdot V_{DS1}}, \quad (j8)$$

ecuación obtenida de la 4 para la zona de saturación.

3.2.2. Cálculo de β :

El único parámetro de continua que nos resta calcular es β , ya que despreciaremos las resistencias y V_{TO} lo tenemos como dato en los manuales (es igual a V_p). Para eso tenemos dos posibilidades, según los datos que disponemos. Veamos la primera:

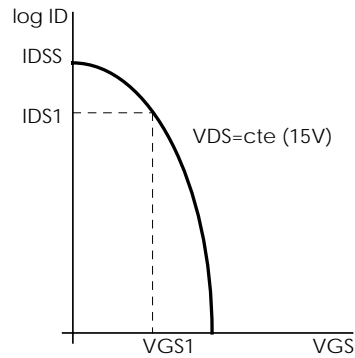


Figura j4

Si disponemos de una gráfica como la de la fig. j4 (o también si solo disponemos de i_{DSS}), podremos utilizar la siguiente fórmula, que se puede deducir a partir de la ec. j4:

$$\beta = \frac{I_{DS1}}{(V_{TO} - V_{GS1})^2 (1 + \lambda \cdot V_{DS})} \quad (j9)$$

Si en cambio tenemos datos referentes de $r_{DSon} = 1/g_{DS}$ de la ec. j7 con $V_{DS} = 0$ (figura j5), hacemos el cálculo de β de la siguiente forma:

$$\beta = \frac{1}{2(V_{GS1} - V_{TO}) \cdot r_{on1}} \quad (j10)$$

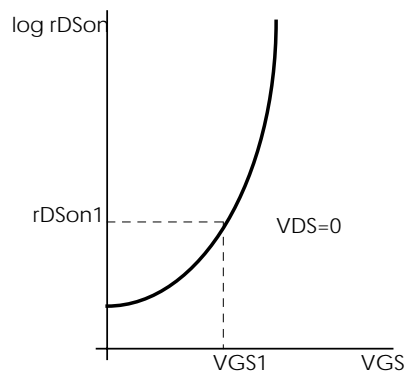


Figura j5

3.2.3. Cálculo de las capacidades C_{GD0} y C_{GS0} :

Esta estimación se realiza a partir de las curvas de C_{is} y C_{rs} , ya que estas son las que se encuentran en los manuales porque son los valores medibles. Veamos las definiciones:

Capacidad de entrada C_{is} : es la capacidad medida entre compuerta y fuente con el drenaje y la fuente cortocircuitados en señal, con una polarización y a una frecuencia determinadas.

Capacidad de transferencia inversa C_{rs} : es la capacidad medida entre drenaje y compuerta con la fuente conectada a tierra e iguales especificaciones.

Como consecuencia de estas definiciones, valen las siguientes relaciones, en donde todas las capacidades intervinientes son variables con la polarización:

$$\begin{cases} C_{GD} \cong C_{rs} \\ C_{GS} = C_{is} - C_{rs} \end{cases} ; \quad (j11)$$

lo que implica que primero deberemos calcular C_{GD0} para poder luego obtener valores de C_{GS} a partir de los datos (C_{is}), como veremos luego.

Trataremos de encontrar los valores de C_{GD0} y C_{GS0} que ajusten lo mejor posible las curvas de C_{is} y C_{rs} . Comencemos con C_{GD0} , para poder calcularlo en forma independiente de C_{GS0} debemos tomar fijo el valor de P_B . El valor por defecto en SPICE es $P_B=1$. Aplicando logaritmo a la ec. j6 nos queda:

$$\log(C_{GD}) = \log(C_{GD0}) - M \cdot \log\left(1 - \frac{V_{GD}}{P_B}\right), \quad (j12)$$

que es la ecuación de una recta y procediendo de forma similar al caso del transistor bipolar:

$$Y = A + B \cdot X;$$

$$Y = \log(C_{GD}), \quad X = \log\left(1 - \frac{V_{GD}}{P_B}\right), \quad A = \log(C_{GD0}), \quad B = -M \quad (j13)$$

Ahora bien, en la figura j6 tenemos la gráfica de C_{rs} como se encuentra en los manuales:

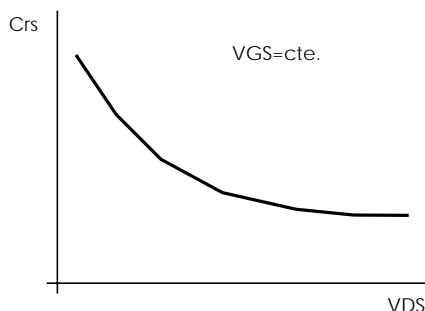


Figura j6

Tomaremos N puntos de esta gráfica. Surge entonces el problema que $C_{rs} = C_{GD}$ está en función de V_{DS} y no de V_{GD} como necesitamos, por lo que deberemos calcular esta última para cada punto, lo cual es posible porque

conocemos V_{GS} que es constante para toda la gráfica. Entonces $V_{GD_i} = V_{GS} - V_{DS_i}$; $i=1 \dots N$.

Volvamos a la ec. j13, tenemos los N pares (X,Y) y además la pendiente de la recta (B) es fija igual a $-0,5$ (recordemos que $P_B=1$). Resta ajustar la intersección de la recta con el eje Y , i.e. el valor de A , tomamos el que más se aproxima a los datos:

$$A = \frac{\sum_{k=1}^N Y - B \sum_{k=1}^N X}{N} \quad (j14)$$

El valor de C_{GD0} surge inmediatamente de la ec. j13.

Para la determinación de C_{GS0} disponemos en los manuales de la curva de C_{is} (fig. j7), que está en función de V_{GS} como necesitamos.

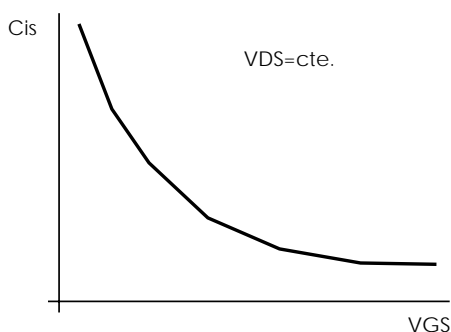


Figura j7

Para cada punto calculamos C_{GS} mediante la siguiente fórmula derivada del sistema de ec. j11 y la ec. j6:

$$C_{GS} = C_{is} - \frac{C_{GD0}}{\left(1 - \frac{V_{GS} - V_{DS}}{P_B}\right)^M} \quad (j15)$$

Desde acá en adelante podemos proceder como en el caso anterior utilizando la ec. j5 en lugar de la j6, determinando así a C_{GS0} .

Podemos dejar el resto de los parámetros de JFET con sus valores predefinidos en SPICE.

Como observación final general (tanto para el BJT como para el JFET), siempre habrá diferencias entre las curvas en los manuales y las resultantes de los modelos, pero estas diferencias serán en general menores que las existentes entre componentes del mismo tipo debidas a la dispersión.

4. Algoritmos

En esta sección haremos un breve repaso de los algoritmos utilizados en el ProEsp para encontrar las raíces de funciones. Estos algoritmos son dos: el método dicotómico y el de Newton-Raphson.

Para describir los métodos consideremos una función f definida sobre el conjunto de los números reales o sobre un intervalo del mismo. Estos métodos permiten calcular las raíces o ceros (z) de la ecuación $f(x)=0$. En principio esto parecería una limitación, pero no lo es ya que podemos encontrar la solución de la ecuación $g(x)=c$ definiendo $f(x)=g(x)-c$.

4.1. Método Dicotómico

4.1.1. Características generales:

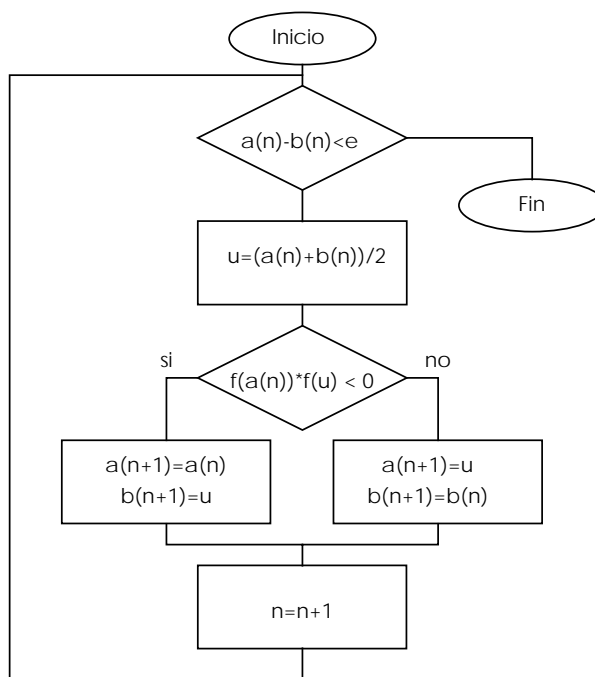
El método consiste en dividir sucesivamente por la mitad el intervalo en el cual se sabe existe una (única) raíz. Se supone la función continua y que asume valores de signos distintos en los extremos del intervalo.

Entonces, si f tiene una sola raíz en el intervalo (a, b) , y en ambos extremos asume valores de signo diferente, existirá un punto intermedio en el cual f se anulará.

Este es el principio que permitirá iterar el proceso hasta hallar un punto intermedio en el cual $f=0$, o encontrarse con un subintervalo de trabajo suficientemente pequeño. En cada interacción la longitud del intervalo será la mitad de la del obtenido en la iteración anterior.

La convergencia de este método es lenta (lineal), pero tiene la ventaja de ser muy simple y por lo tanto de muy fácil implementación.

4.1.2. Diagrama del algoritmo:



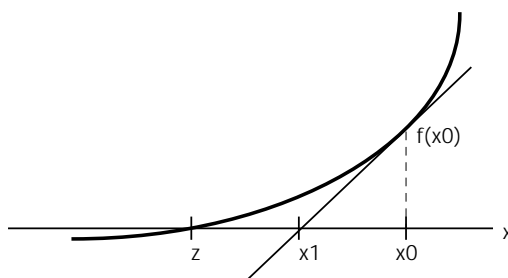
a, b : extremos del intervalo
 n : número de iteración

u: aproximación a la solución (z)
 e: acotación del error

4.2. Método de Newton-Raphson

4.2.1. Características Generales:

Consiste en reemplazar la curva por su tangente. Partiendo de un punto inicial x_0 (valor de arranque) y trazando la tangente a la curva en $(x_0, f(x_0))$, y luego prolongando esta se obtiene el punto x_1 que bajo ciertas condiciones será una mejor aproximación al valor z que el x_0 (ver figura).



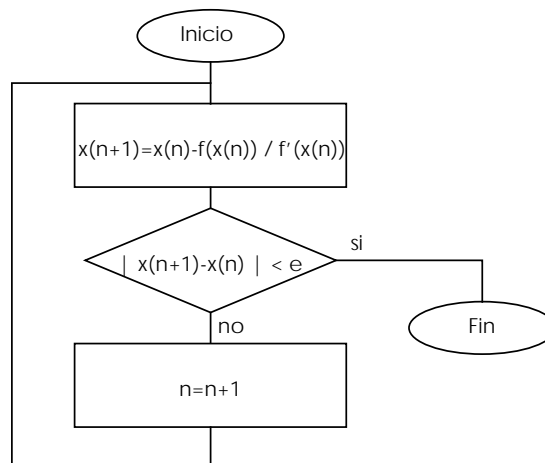
En general se tiene que:

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

Se puede demostrar que siendo f' y f'' continuas, f'' de igual signo en todo el intervalo, y f' no nula en él, entonces si se toma como valor de arranque un punto en el cual el valor de la f tenga igual signo que f'' , este método resulta convergente.

La convergencia de este método es más rápida que la del anterior (es cuadrática), pero hay más dificultades en su implementación.

4.2.2. Diagrama del algoritmo:



n: número de iteración
 e: acotación del error

5. Implementación

En este punto consideramos la estructura interna del ProESP, así como también la elección del lenguaje para su implementación.

5.1. Lenguaje de programación utilizado

En principio podríamos haber usado cualquier lenguaje, pero hay ciertos factores que llevaron a elegir al C. El mismo es un lenguaje de nivel medio que soporta programación estructurada y el código objeto generado por el compilador es más rápido y compacto que el correspondiente a otros lenguajes de más alto nivel. Es uno de los más utilizados en los últimos tiempos para la realización de todo tipo de aplicaciones, convirtiéndose casi en un estándar, por lo tanto su uso permite una mayor probabilidad de reutilización de rutinas e inclusión de partes del programa en sistemas más complejos que puedan realizarse. Además hay que mencionar su portabilidad, esto es, se puede trasladar el código fuente entre distintas plataformas (cada una con su propio compilador C) casi sin realizar modificaciones.

El programa se desarrolló para IBM-PC compatibles por ser la plataforma más difundida en la actualidad y además puede realizar sin demoras excesivas el volumen de cálculo necesario. El compilador elegido es el Turbo C 2.0 (de Borland), que viene con un entorno de desarrollo que incluye entre otras cosas un editor del código fuente, un debugger y ayuda on-line. Además cuenta con una biblioteca de funciones muy completa que facilita la tarea de programación y algunas facilidades como por ejemplo la autodetección y uso del coprocesador matemático si este se halla presente o su emulación en caso contrario.

Nota: Con respecto a la portabilidad, el programa utiliza funciones para el manejo de pantalla en modo gráfico que probablemente no estén disponibles en los compiladores para otras plataformas distintas de la IBM PC compatible. No obstante preferimos esto a realizar un programa con presentaciones solo texto pero totalmente portable ya que este último sería mucho menos interactivo y más limitado al no poder dibujar gráficas.

5.2. Estructura interna

Para implementar el programa se recurrió a dividirlo en módulos, cada uno con funciones específicas, conforme a la programación top-down. El criterio para la división fue que las interacciones entre los módulos fueran las menos posibles y que cada uno realizara una tarea completa, facilitando de esta manera el diseño de cada uno y el debugging. En la tabla siguiente se enumeran esos módulos y se comentan los contenidos de cada uno:

PRINC.C	Contiene el programa principal y las funciones necesarias para la presentación.
GRAF.C	Contiene la rutina de detección de tarjeta gráfica, las funciones de entrada y salida de datos en pantalla gráfica, y la rutina que realiza la gráfica de funciones.

IFORWARD.C VAF.C HFE.C RBASE.C BREV.C CAPBJT.C TFOR.C TREV.C LAMDFET BETAFET.C CAPFET.C	Cada uno contiene las funciones necesarias para el control de la pantalla y el cálculo de un grupo de parámetros. Por ejemplo, IFORWARD.C contiene lo necesario para el grupo Is y Nf del transistor bipolar.
GLOBAL.H	Contiene definiciones comunes a todos los demás módulos.

Describiremos a continuación cada uno de los módulos y las funciones que los componen en el orden en que se encuentran en la tabla, esto es, primero los generales, luego los correspondientes a grupos de parámetros del transistor bipolar y por último los correspondientes a los grupos del JFET.

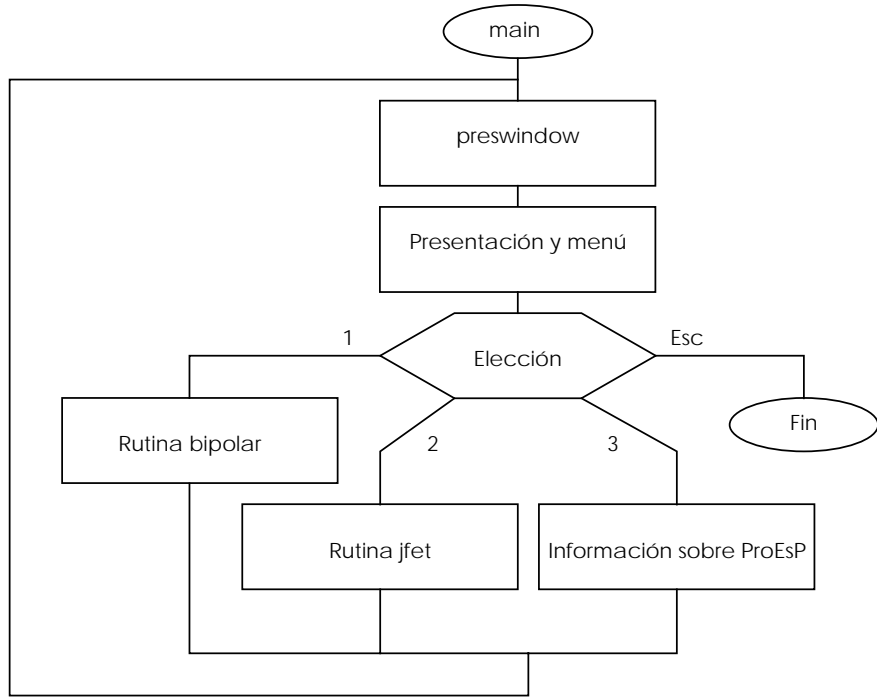
5.2.1. PRINC.C

Este módulo contiene lo siguiente:

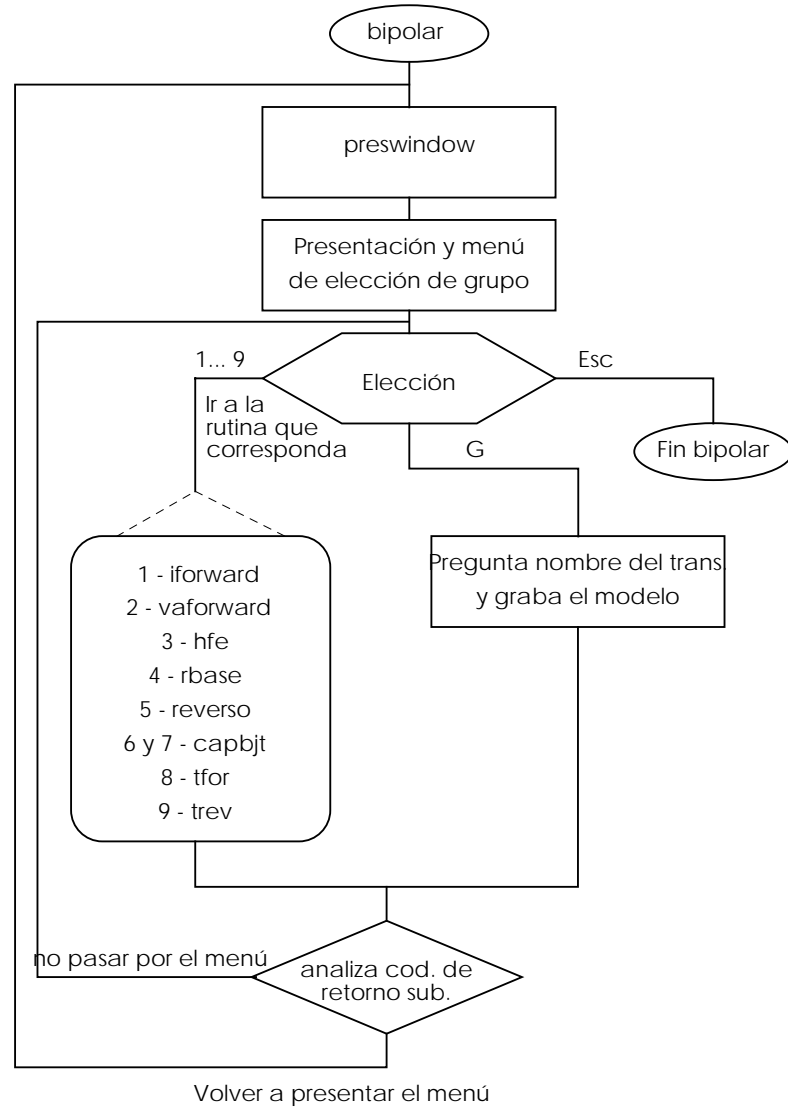
- Las definiciones y los valores iniciales de todos los parámetros de los modelos para el BJT y el JFET usados en el ProESP.
- El programa principal (*main*), que se ocupa entre otras cosas de la presentación, la elección del tipo de transistor (BJT o JFET), y la salida al sistema operativo.
- La función *preswindow* es la encargada de 'preparar' la pantalla gráfica para la presentación y los menús.
- La función *floatwindow* dibuja ventanas 'flotantes' para carteles y cuadros de diálogo durante la presentación y los menús específicos de cada transistor.
- La rutina *bipolar* es la encargada de generar el menú para este tipo de transistores, llamar a la rutina de cálculo que corresponda y grabar el modelo cuando el usuario lo solicite.
- La rutina *jfet* realiza las funciones de la anterior pero para transistores de efecto de campo de juntura.

Veamos ahora un diagrama de flujo simplificado de las rutinas más importantes dentro de PRINC.C.

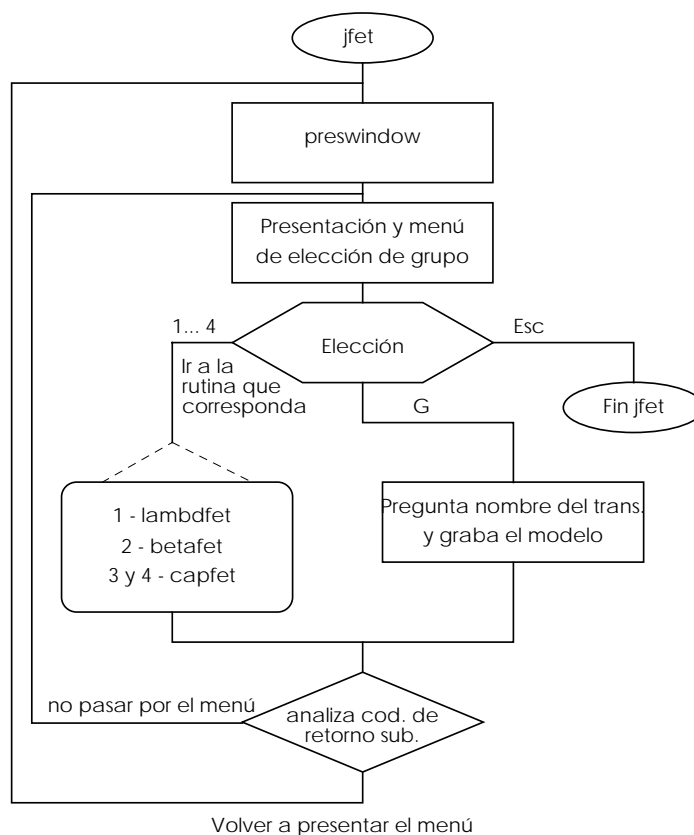
En las rutinas *bipolar* y *jfet* hay un cuadro redondeado que indica la subrutina llamada según el número elegido, por ejemplo si en bipolar elegimos 3 se ejecutará *hfe*. El cuadro de decisión 'analiza el código de retorno de la subrutina' decide si hay que volver a presentar el menú e ingresar una opción o pasar a ejecutar directamente otra subrutina en función de un código que retorna la última subrutina llamada, por ejemplo este código puede indicar *ejecute la próxima subrutina de la lista sin pasar por el menú*, o puede indicar *presentar el menú e interrogar al usuario*.



Programa principal



Rutina bipolar



Rutina jfet

5.2.2. GRAF.C

Aquí encontramos las siguientes funciones:

- *Initialize*: inicializa el sistema y reporta cualquier error que pueda ocurrir.
- *MainWindow*: prepara la pantalla para que la usen las rutinas de cálculo de grupos de parámetros (por ej. *brev*).
- *StatusLine*: muestra mensajes en una zona especial en la parte inferior de la pantalla.
- *DrawBorder*: dibuja el borde de la ventana activa.
- *changetextstyle*: cambia el tamaño, la orientación y el tipo de letra con que se presenta el texto en la pantalla gráfica.
- *prompt*: realiza la entrada de datos desde el teclado con pantalla en modo gráfico presentando un cursor en la posición indicada y con la posibilidad de usar teclas de edición como Bs (Back space).
- *getdata*: igual que *prompt* pero es específica para entrar números en representación punto flotante y en zonas definidas de la pantalla.
- *print*: imprime valores de variables con su correspondiente cartel en una zona de la pantalla destinada a tal fin.
- *plot*: realiza la gráfica de una función en el rango que se le indique con escala lineal o semilogarítmica y autorango en el eje y, además ubica pares ordenados separados dentro de la gráfica.

Todas estas funciones son utilizadas en el resto del programa para facilitar la entrada y salida de datos. Hay que destacar que se han utilizado coordenadas relativas con el fin de hacer las funciones independientes de la resolución de la pantalla en particular que se utilice. También se tuvo en cuenta si esta es color o monocromática, de manera que en cualquier caso la presentación es buena.

5.2.3. Subrutina de grupo general

Presentaremos ahora el diagrama de flujo general de una subrutina de grupo de parámetros. Con este diagrama queremos indicar la filosofía general que siguen estas subrutinas.

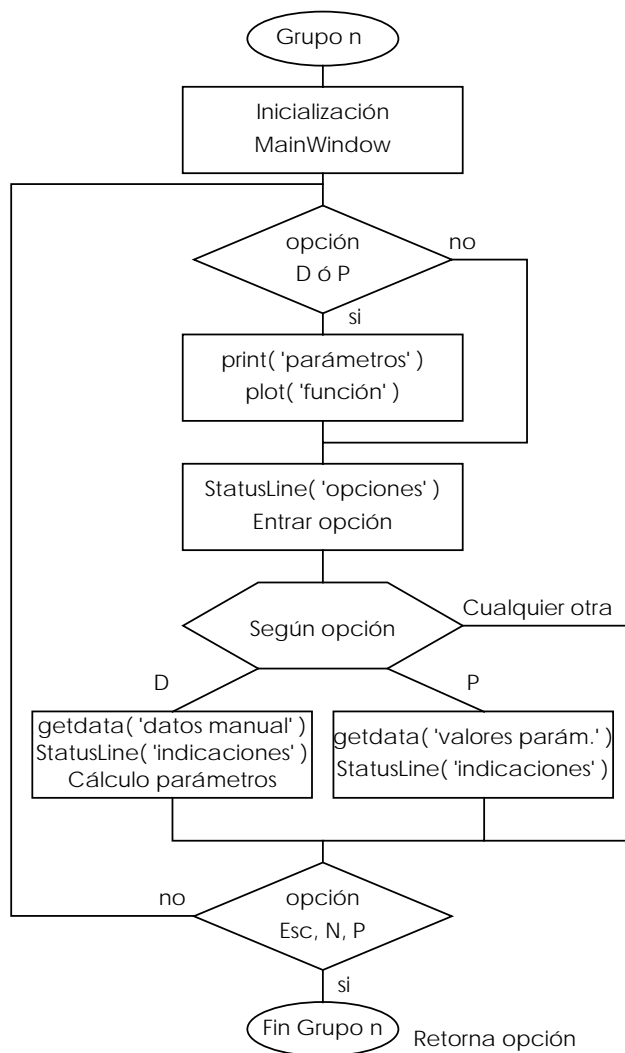


Diagrama general de una rutina de cálculo de grupo

Estas rutinas son en realidad, desde el punto de vista del lenguaje, funciones enteras que pueden retornar tres valores que son los correspondientes al código ASCII de los caracteres N, P o Esc. Ellos serán interpretados por la rutina *bipolar* o *jfet* según corresponda. Los significados atribuidos a estos caracteres son: N, pasar al próximo grupo; P, pasar al grupo previo y Esc, volver al menú del transistor.

Enumeramos a continuación las funciones contenidas en los módulos correspondientes a cada grupo de parámetros. Para el cálculo de los parámetros las distintas funciones utilizan los métodos que describimos.

Comenzamos con los módulos relacionados con el transistor bipolar.

5.2.4. **IFORWARD.C**

- *iforward*: esta es una subrutina de grupo encargada de los parámetros del transistor bipolar Is y Nf.
- *Vbe*: esta es la función que se grafica en la pantalla correspondiente a *iforward*.

5.2.5. **VAF.C**

- *vaforward*: subrutina de grupo encargada de VAF.
- *Icvaf*: es la función a graficar en este grupo.

5.2.6. **HFE.C**

- *hfe*: subrutina de grupo para Bf, Ise, Ne e Ikf.
- *isebf*: función de uso interno de *hfe*.
- *hFE*: función a graficar del grupo.

5.2.7. **RBASE.C**

- *rbase*: subrutina del grupo de rb y rbm.
- *VbeRb*: es la función que se grafica en este grupo.

5.2.8. **BREV.C**

- *reverso*: subrutina de grupo para Br, Isc y Rc.
- *tensbe*: función de uso interno de *reverso*.
- *Vce*: función para graficar del grupo.

5.2.9. **CAPBJT.C**

- *capbjt*: subrutina de grupo para Cje, Vje y mje o Cjc, Vjc y mjc; según los argumentos de entrada.
- *Cjbe*: función a graficar si se está tratando la juntura de emisor.
- *Cjbc*: función a graficar si se está tratando la juntura de colector.

5.2.10. **TFOR.C**

- *tfor*: subrutina de grupo para Tf
- *fT*: función a graficar en el grupo.

5.2.11. **TREV.C**

- *trev*: subrutina de grupo para Tr
- *iba*: función de uso interno de *trev*.

Observación: la gráfica para este grupo no está implementada en esta versión del programa.

A continuación los módulos relacionados con el JFET.

5.2.12. **LAMDFET.C**

- *lambdfet*: subrutina de grupo para Lambda.
- *Iclam*: función a graficar en el grupo.

5.2.13. **BETAFET.C**

- *betafet*: subrutina de grupo para Beta.
- *Vgsid*: función para graficar Vgs versus Id.
- *Vgsrd*: función para graficar Vgs versus rdson.

Nota: La subrutina *betafet* permite cambiar la gráfica visualizada entre las dos disponibles a petición del usuario.

5.2.14. **CAPFET.C**

- *capfet*: subrutina de grupo para Cgs0 o Cds0, según los argumentos de entrada.
- *Crs*: función a graficar si se está determinando Cgd0.
- *Cis*: función a graficar si se está determinando Cgs0.

5.3. Códigos fuente y ejecutable

Para aquellos que quieran saber mayores detalles acerca de la implementación se encuentran disponibles en el Laboratorio de Microelectrónica de la U.N.R. (Av. Pellegrini 250 - 2000 Rosario - Argentina) todos los módulos con el código fuente.

También quienes tengan interés en utilizar el ProEsP pueden conseguirlo listo para usar en el mencionado laboratorio. La versión actual es la 1.10.

6. Manual del Usuario

El manejo del programa es bastante intuitivo y en general solo hay que elegir entre varias opciones disponibles o ingresar los datos solicitados. Explicaremos el manejo mediante el siguiente ejemplo.

6.1. Ejemplo

Supongamos querer determinar los parámetros de los JFET BC264B y D. Lo primero que debemos hacer es conseguir los datos del manual para estos transistores. Una vez hecho esto, corremos el programa tipeando PROESP en el prompt del DOS. Aparecerá la siguiente presentación:

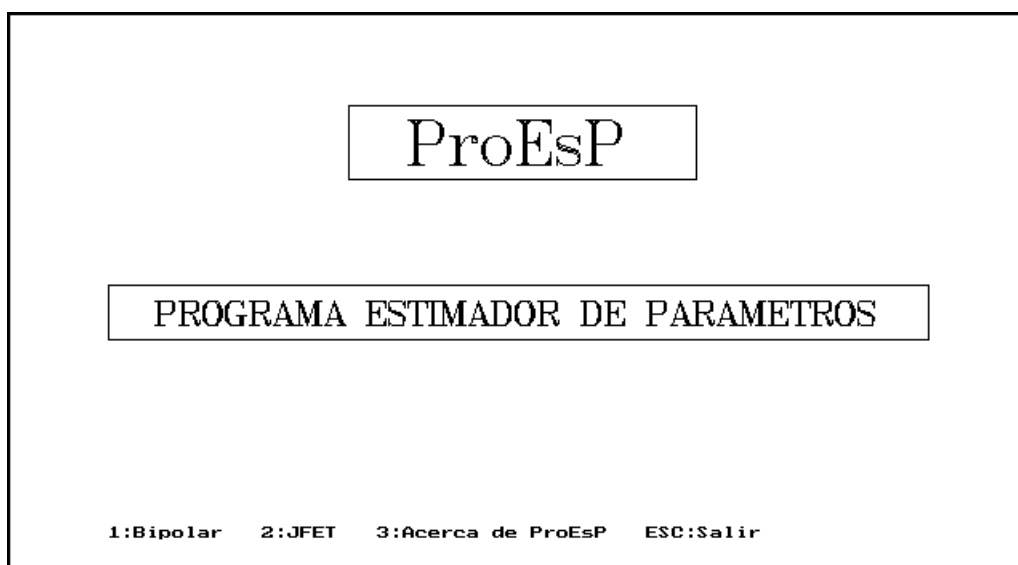


Figura m1

Como nuestros transistores son JFET, oprimimos el número '2' en el teclado para elegir esta opción. Al hacer esto, aparece el menú específico para el JFET:

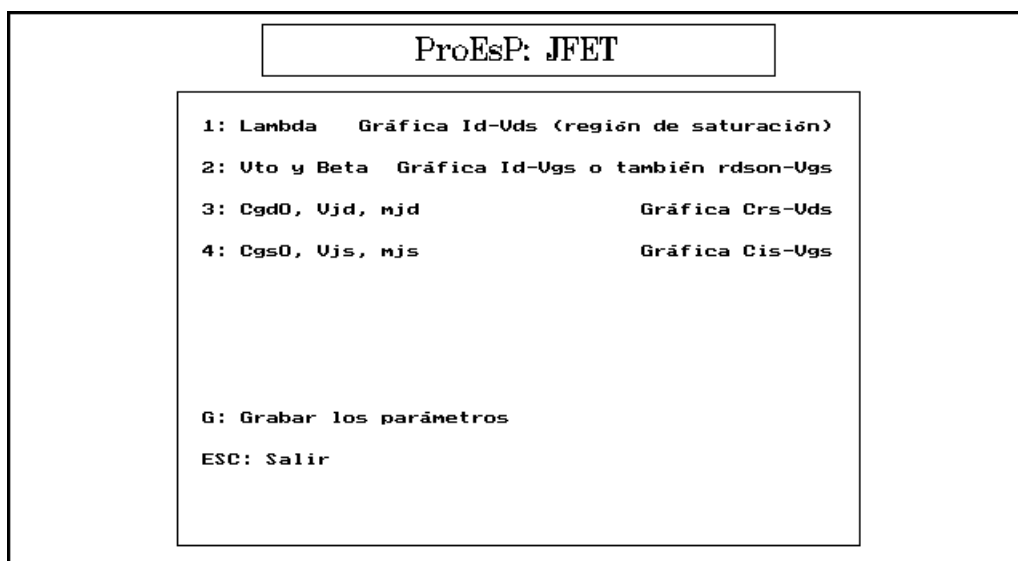


Figura m2

Ahora debemos elegir el grupo de parámetros a calcular. En el menú de la figura m2 tenemos primero una lista de los grupos de parámetros del JFET, a cada uno le corresponde una gráfica que es la que contiene los datos necesarios. La tecla que sirve para seleccionar cada item se encuentra indicada a la izquierda del mismo. Elegimos la opción '1' porque en general los parámetros calculados en un grupo son necesarios para el cálculo de los parámetros de los grupos siguientes. Al hacerlo pasamos a la pantalla de cálculo de lambda:

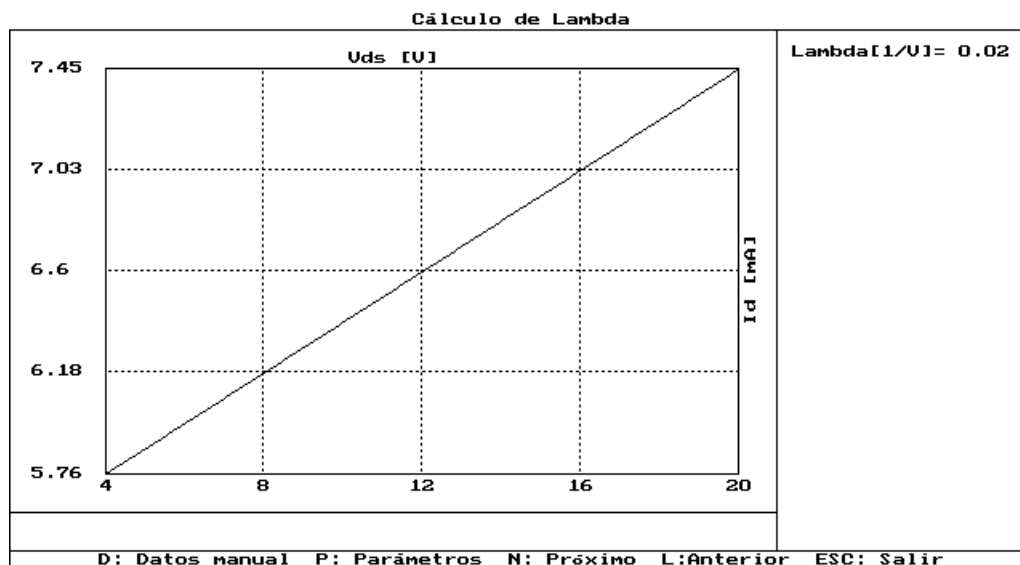


Figura m3

Aquí encontramos en el recuadro de la derecha el valor actual de lambda (valor por defecto). En ese recuadro figuran siempre los valores de los parámetros correspondientes a la pantalla y además los datos que vayamos ingresando. Los valores se van escribiendo formando una lista por orden cronológico, cuando se acabe el lugar en el cuadro la lista comienza a desplazarse hacia arriba. En la gráfica podemos ver la pendiente de la curva de I_D en función de V_{DS} para el valor actual de lambda. Abajo tenemos los comandos posibles: oprimimos la tecla 'D' para ingresar datos del manual. Cuando hacemos esto nos aparece la variable que debemos ingresar seguida de su unidad entre corchetes, un signo igual, su valor actual (en este caso el valor predefinido) y un cursor en la ventana destinada a la entrada de datos y una indicación en la ventana inferior. En este ejemplo sería: **vds1 [V]= 10_** . Como en este caso ingresamos pares ordenados, los dos valores que lo componen quedan uno al lado del otro en la ventana de entrada de datos. Mientras tecleamos los valores podemos utilizar la tecla Bs para borrar si nos equivocamos.

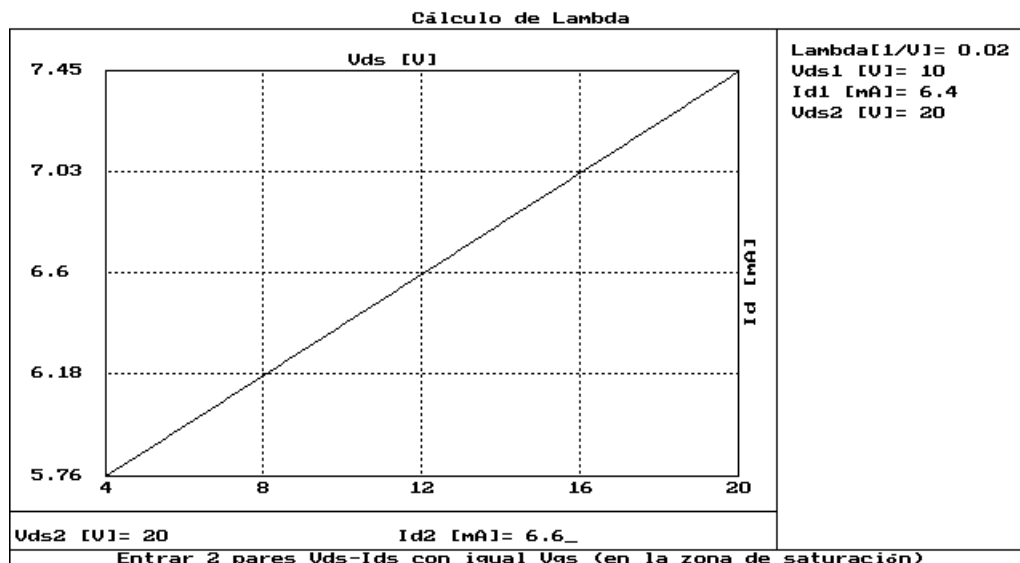


Figura m4

En la figura m4 vemos el estado luego de haber ingresado un par ordenado completo y estamos a punto de terminar de ingresar el segundo. Podemos observar que los valores ya confirmados van quedando en el cuadro de la derecha, y la gráfica no ha cambiado. Recién al entrar el último dato se presenta el nuevo valor de lambda y se actualiza la gráfica, que en este caso es siempre una recta con distinta pendiente. Luego de esto, tenemos nuevamente las opciones de la fig. m3 pasamos al grupo de parámetros siguiente con la tecla 'N'.

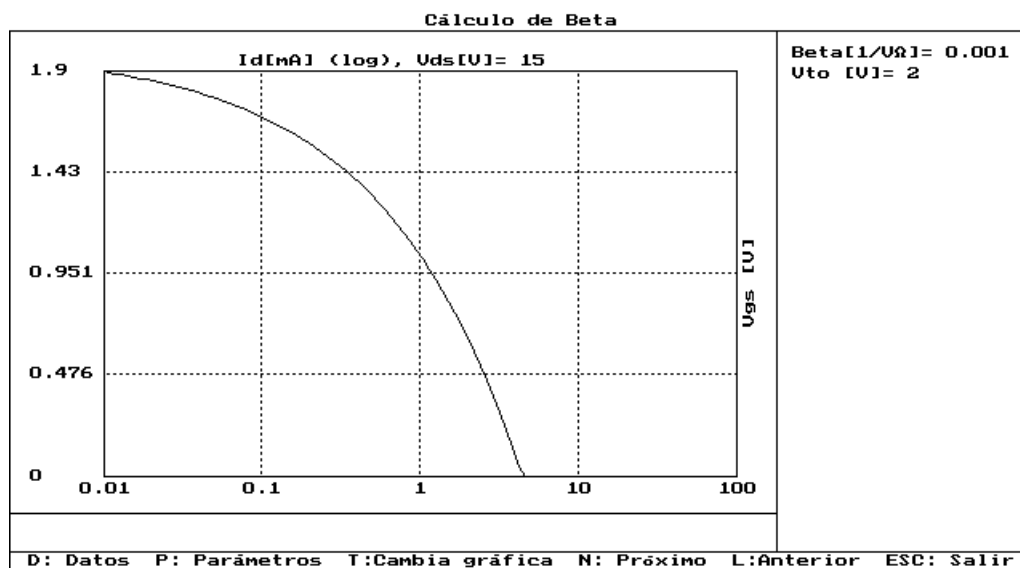


Figura m5

Estamos ahora en una situación similar que al principio de la pantalla anterior. Elegimos la opción 'D' e ingresamos lo que se nos pide.

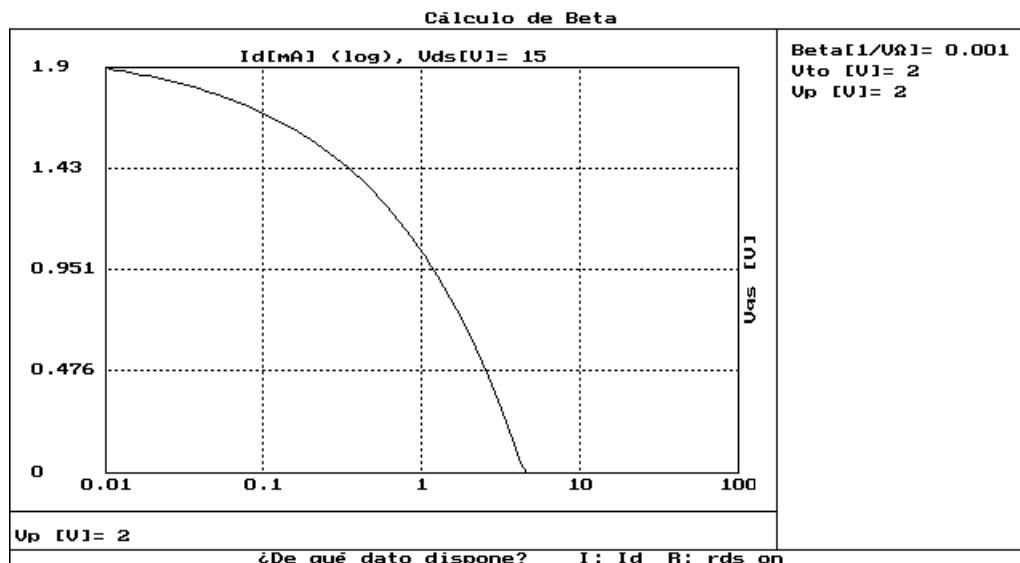


Figura m6

Aquí (en la fig m6) encontramos que luego de ingresar V_p debemos elegir entre ingresar datos relacionados con I_D o con r_{DSon} (recordar que en el menú de la fig. m2 había 2 gráficas posibles). Supongamos tener datos referentes a I_D . Apretamos 'I'.

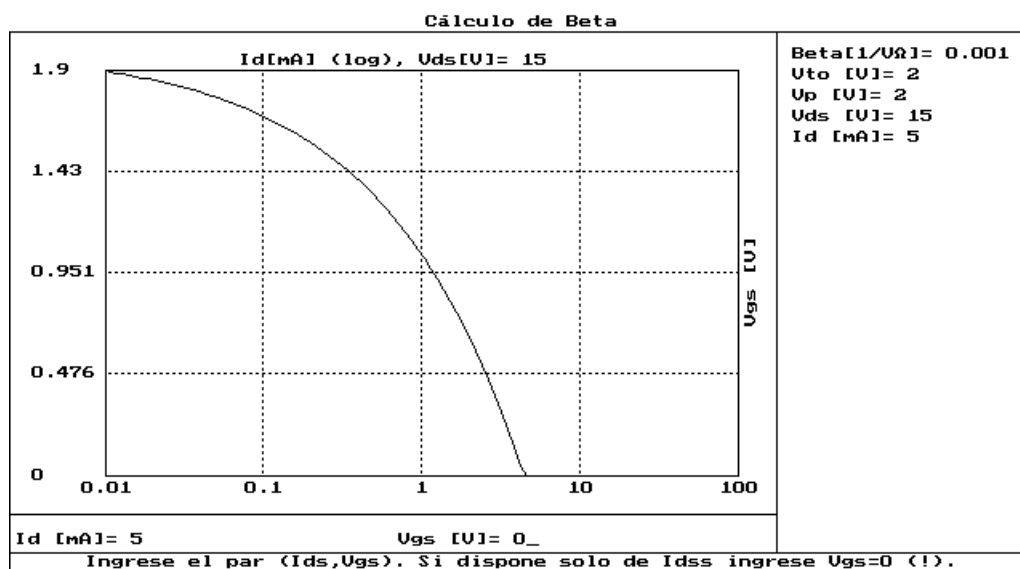


Figura m7

En la ventana inferior se nos indica que si solo disponemos de I_{DSS} podemos ingresar ese valor como I_D , entrando luego cuando se pida $V_{GS}=0$. Luego de entrar V_{GS} (que es el último dato), sucede lo siguiente:

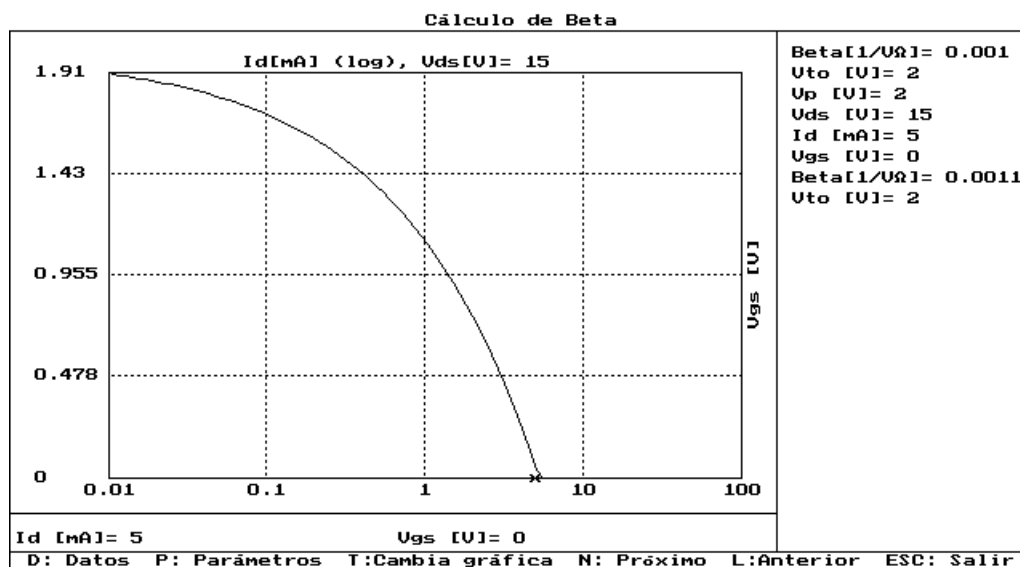


Figura m8

Tenemos los nuevos valores para los parámetros (los últimos en la lista) y la gráfica ha sido actualizada, además hay una 'x' representando el punto ingresado (5 mA, 0V). En la parte superior de la gráfica se nos indica que en el eje horizontal está I_D en escala logarítmica y que la gráfica está hecha para $V_{DS}=15V$. A la derecha de la gráfica vemos que el eje vertical corresponde a V_{GS} . Pasamos ahora al próximo grupo.

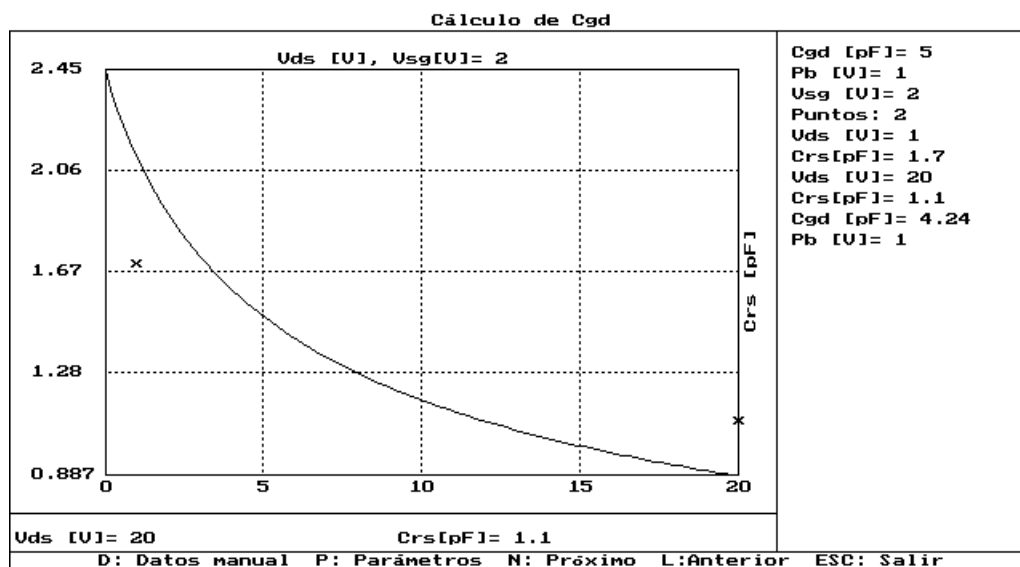


Figura m9

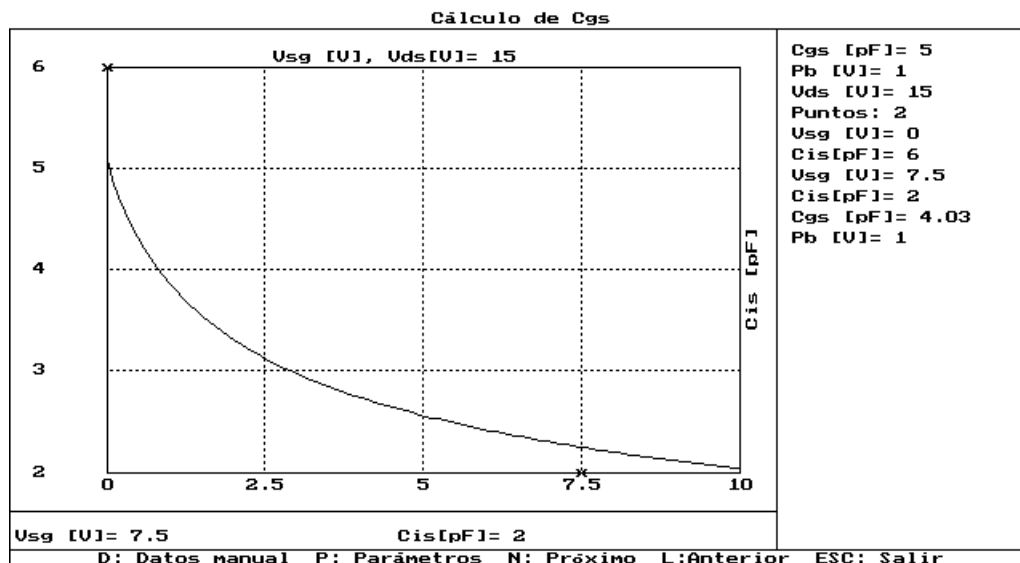


Figura m10

En las figuras m9 y m10 vemos las pantallas correspondientes a las capacidades del JFET. Las cruces en las gráficas se corresponden con los puntos ingresados, como siempre.

En estas dos últimas pantallas el parámetro P_B no es modificado por el programa, pero si por alguna razón nosotros quisiéramos hacerlo, utilizaríamos el comando 'P'. Al oprimir esta tecla (cuando está disponible en la ventana inferior de la pantalla) el programa nos va pidiendo los valores de los parámetros de la misma forma que con el comando 'D'.

Como ya calculamos todos los parámetros necesarios apretamos Esc desde cualquier pantalla de grupo y volvemos al menú de la fig. m2. debemos grabar el modelo que acabamos de generar si queremos utilizarlo, y lo hacemos oprimiendo la 'G', después de lo cual se abrirá una ventana en la que escribimos el nombre del modelo y el tipo de transistor (fig m11).

ProEsp: JFET

1: Lambda Gráfica Id-Uds (región de saturación)
2: Uto y Beta Gráfica Id-Ugs o también rdson-Ugs
3: Cgd0, Uj Gráfica Crs-Uds
4: Cgs0, Ujs Gráfica Cis-Ugs

Nombre del transistor:
 bc264b
Tipo: NJF
 OK

G: Grabar los parámetros
ESC: Salir

Figura m11

Si recibimos un Ok como respuesta, nuestro modelo ya está grabado en el disco y podemos continuar el trabajo. Si por alguna razón el programa no puede grabar el modelo nos lo dirá con un mensaje de error, esto puede ocurrir si por ejemplo estamos trabajando en un disco protegido contra escritura (¡la solución es sacar la protección para poder grabar!). Luego de esto presionando cualquier tecla volvemos a la situación de la figura m2. Hagamos un pequeño paréntesis en la explicación para ver que es lo que grabamos.

El archivo que acabamos de generar es el BC264B.MOD y su contenido es:

```
* Modelo creado por el ProESP *
.MODEL BC264B NJF( LAMBDA=0.003226
+   VTO=-2V BETA=0.001192
+   CGS=4.028PF CGD=4.235PF PB=1V
+ )
```

Este archivo ya está listo para ser usado incluyéndolo con el archivo de entrada al simulador o como elemento de biblioteca.

Nos queda ahora determinar los parámetros para el BC264D. Para ello entramos, en el menú de la fig m2, directamente en la opción '2' porque vemos en el manual que en lo único que se diferencian el BC264B y D es en las características correspondientes a este grupo.

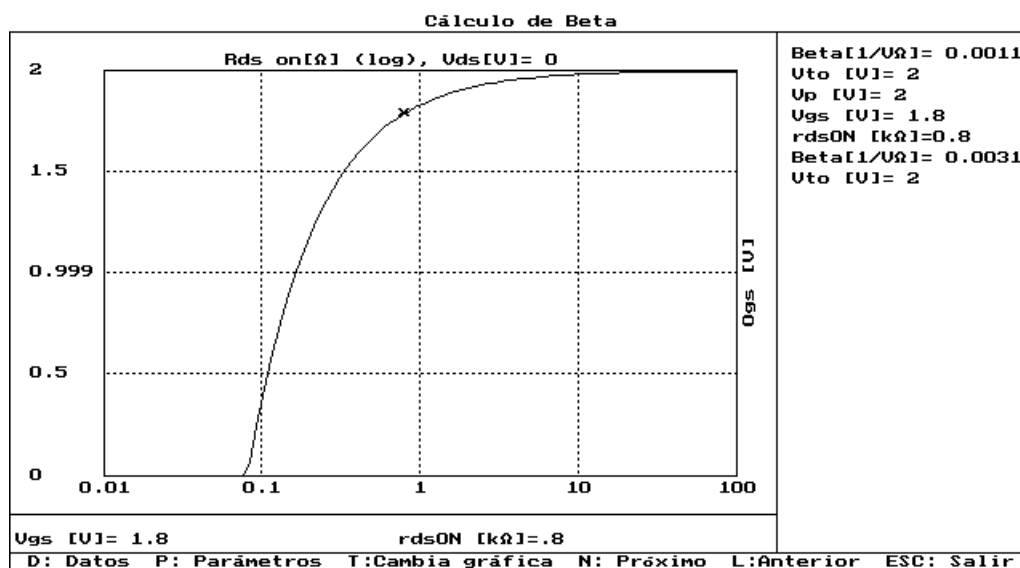


Figura m12

En este caso hicimos el cálculo a partir de r_{DSon} , y en la figura podemos ver la gráfica correspondiente, pero podemos ver la gráfica I_D - V_{GS} (como en la fig. m5) pero actualizada de acuerdo a los nuevos valores de los parámetros simplemente apretando la tecla 'T'. Con esa tecla conmutamos entre las dos gráficas.

Solo nos resta volver al menú del JFET (fig. m2) y grabar el nuevo modelo, cambiando el nombre BC264D, por ejemplo. Luego oprimiendo Esc volvemos al menú principal (fig m1). Aquí podríamos entrar en la parte para transistor bipolar si fuera necesario, ver información sobre el programa, volver al menú del JFET, o volver al sistema operativo volviendo a oprimir Esc y confirmando afirmativamente a la pregunta si queremos salir que nos hará el programa.

6.2. Resumen de comandos

La siguiente lista contiene los comandos disponibles en todas las pantallas de cálculo y su efecto (cuando están disponibles se muestran en la barra de estado como en la figura m3):

D (Datos): permite ingresar datos del manual.

P (Parámetros): es para asignar valores arbitrarios a los parámetros.

N (Next): es para avanzar a la próxima siguiente pantalla de grupo.

P (Previous): igual a N pero a la pantalla anterior

Esc (Escape): permite volver al nivel inmediato superior.

6.3. Observaciones y Recomendaciones

- Los datos ingresados y los valores de los parámetros se mantienen inalterados hasta que se salga del ProESP o se reemplacen por nuevos valores en la pantalla de cálculo correspondiente. Esto permite 'navegar' a través de los menús y pantallas sin miedo a perder el trabajo ya realizado.
- Recordar, no obstante lo anterior, que si cambiamos los parámetros de un grupo esto puede alterar las gráficas correspondientes a los grupos siguientes en el menú del transistor, obligándonos quizás a recalcular estos últimos.
- Para recalcular un grupo de parámetros sin cambiar los datos del manual lo más sencillo es elegir la opción 'D' y confirmar todos los valores de los datos que van apareciendo (oprimiendo Enter).
- En general, tratar de elegir los datos cercanos a la zona de trabajo esperada del elemento.
- No siempre es necesario calcular todos los parámetros que el programa permite, por ejemplo si vamos a realizar un ensayo DC no es necesario calcular las capacidades, o si vamos a usar un transistor bipolar en zona activa no tiene objeto calcular los parámetros que intervienen en la saturación.
- Si utiliza el programa, el autor agradecería el envío de comentarios y sugerencias.

7. Bibliografía

- [1] Franz Sischka, Eine Methode zur Bestimmung der SPICE-Parameter für Bipolare Transistoren. AEÜ, Band 39 [1985], Heft 4.
- [2] Paolo Antognetti, Giuseppe Massobrio, Semiconductor Modelling with SPICE, McGraw-Hill Book Company, 1988.
- [3] MicroSim Corporation, PSpice User's Guide, 1986.
- [4] Intusoft, PRE_Spice Manual, 1986.
- [5] Millman y Taub, Circuitos de Pulsos, Digitales y de Conmutación, McGraw-Hill Book Company, 1980.
- [6] Borland International, Turbo C User's Guide, 1987.
- [7] Brian Kernighan, Dennis Ritchie, C Programming, Prentice-Hall International, 1978.
- [8] Lessons in C, P.J.Ponzo, Dpt. of Applied Math., Univ. of Waterloo (software).
- [9] Philips, Field Effect Transistors, Eindhoven, 1982
- [10] Philips, Small-Signal Transistors, Eindhoven, 1982
- [11] Siemens, Discrete Semiconductors for Surface Mounting Data Book, 1987/88.