



FUNCIÓN:

La finalidad del programa es facilitar la resolución de los cálculos, de los parámetros de pequeña señal, puntos de trabajo, impedancias y ganancias, de un amplificador. Diseñado a partir de transistores bipolares de tipo NPN, y con configuraciones de 4 tipos; – Emisor común – Emisor común con resistencia en el emisor – Colector común y – Base común.

MEMORIA DINÁMICA:

Para el desarrollo del problema se ha usado memoria dinámica. Por lo cual se ha diseñado una lista doblemente enlazada. En la cual los elementos son estructuras definidas como transistores. Se entiende que por cada elemento es una etapa en la que la señal de salida de la primera será la entrada de la siguiente etapa. Cada elemento tiene una serie de elementos; – Caracteres: Para definir la polarización y la configuración de cada etapa. – Floats: Para recoger los datos de los dispositivos propios de cada polarización. – Punteros: Para definir tanto el elemento anterior como el siguiente.

El hecho de utilizar memoria dinámica y en concreto las listas, deja al programa la libertad del número de elementos. Siendo el límite de este el límite de memoria. Lo cual es una ventaja frente a un array dinámico ya que una estructura ocupa relativa poca memoria. También tenemos que ver el punto de vista en el que el programa toma una clara función didáctica, y bajo este punto la lista supone un mayor manejo de memoria dinámica.

El programa al utilizar listas se le ha definido un tipo de estructura llamado “etapa” la cual es un puntero a una estructura de tipo transistor. Este trabaja con tres punteros de tipo etapa; – etp (primera etapa) – ant (anterior) y – act (actual). El puntero etp es fijo en el que guarda la dirección del primer elemento y no se modifica, así lo tenemos como referencia para cuando necesitemos recorrer todas las etapas. Sin embargo los otros dos punteros (ant y act), se van modificando según se requiera.



AYUDA:

En el menú principal se disponen varias opciones. La opción de ayuda muestra toda la ayuda del programa, es decir, intenta explicar de forma resumida el funcionamiento del programa en general y como este nos pedirá los datos. Inicialmente se da una pequeña introducción de la función del programa y de cómo introducir correctamente los datos. Seguidamente se ofrece información tanto de los tipos de polarización como de la configuración de las etapas.

También se incluyen dos esquemas explicativos sobre las polarizaciones. Aunque estas ya han sido explicadas anteriormente se han incluido debido a que la explicación redactada puede inducir a errores y así se muestra también la posición con el nombre con el que luego se van a requerir los datos. Estos esquemas se han desarrollado en librerías independientes. Debido al compilador se puede observar que los símbolos escritos no corresponden con los que luego se ven en el programa, esto es por que no utiliza los mismos códigos de la tabla ASCII. Es por ello que primero se ha tenido que buscar los códigos a los que se equiparaban y luego sustituirlos por estos.

También la información sobre la configuración o la polarización se le ofrece al usuario en el momento que se le pide que seleccione el tipo de cada una de estas. Y siempre que acaba la ayuda devuelve al usuario al último menú en el que se encontraba.

INICIACIÓN DE LA CONFIGURACIÓN:

En el programa se puede observar una diferencia entre la polarización de la primera etapa y el resto, esto se debe a lo anteriormente explicado sobre que el puntero etp no se modifica. Antes de iniciar el menú el puntero etp es inicializado a NULL. En el caso de que la memoria permita reservar memoria para el primer elemento este se guardará en etp. Después de esto el programa pedirá los valores de las resistencias generales del diseño como son la resistencia en serie con la señal de entrada y la resistencia de carga.

Seguidamente se pide que se especifique tanto el tipo de configuración como el de polarización y después empieza la recogida de datos de los valores propios de la polarización. Es importante reseñar que estas recogidas de datos tiene dispuesta la posibilidad de que no se le de el dato deseado en este caso volverá a pedirlo.

Cuando acabe el proceso de configuración de la etapa el programa pregunta sobre la posibilidad de que se desee aumentar el diseño en una etapa más. En este caso ya se empieza a trabajar con los punteros ant y act. Así se repetirá el proceso anteriormente explicado.

También es importante apuntar que este tipo de configuraciones se hace con el transistor trabajando en la región activa. En el caso de que después de pedir los datos se compruebe que este no trabaje en esta zona, se pedirá la reconfiguración de este.



VISION DE LOS DATOS:

La función principal del programa es en la que se recoge la visualización de los datos finales. Se ofrecen la posibilidad de ver tanto datos del diseño general como de cada una de sus etapas individualmente, hecho que ayuda por si los resultados no son los deseados ver que etapa debe ser modificada. Hay que tener en cuenta que si se elige esta función antes de haber inicializado la etapa el programa le responderá que no hay ninguna etapa.

Como datos individuales está el punto de trabajo, con el cual ya es posible conocer los dos trípticos de tensiones y corrientes, que se compone de la corriente de colector I_c y de la tensión colector emisor V_{ce} , también esta la opción de visualizar las ganancias e impedancias de cada una de las etapas.

Y como último dato de visualización está las ganancias tanto de corriente como de tensión y la impedancia de entrada y salida. Datos que serán los propios del amplificador.

FUNCIONES Y PRIMITIVAS:

Las funciones que utiliza el programa han sido incluidas en una librería esto no es que se deba a otra cosa que no sea el facilitar la visualización de esta y del programa en si, dejando así la función main aislada. El hecho de haber trabajado con listas ha implicado el desarrollar una serie de primitivas para trabajar con estas. Todas ellas también se incluyen en otra librería distinta a las de las funciones.

FINALIZACIÓN:

El programa se cierra con la opción del menú principal de <S>alir, para que así a través de la función free se libere toda la memoria reservada anterior.