AutoAmp : An Open-Source Analog Amplifier Design Tool - For Classroom and Lab Purposes

Om Prasad Patri K Sanmukh Rao

{o.patri, kuppannagari}@iitg.ernet.in

Supervised by : Dr. Amit K. Mishra, Assistant Professor, Department of ECE, IIT Guwahati

Department of Computer Science and Engineering,
Indian Institute of Technology Guwahati

Abstract

This correspondence presents an open-source tool *AutoAmp* developed at the Indian Institute of Technology, Guwahati. It is available at

Windows and Linux-based systems. The necessary adaptations for the OS are mentioned in the AutoAmp website [1]. Autoamp is available for free (along with the source code) at [1].

There are not many free tools which automatically design basic amplifier circuits given the design specifications. It is expected that industries may be maintaining customized circuit design tools to solve their purpose. However use of such commercial tools for academic purposes is likely to be prohibitively expensive. Basic amplifier-circuit design along with its analysis is required in any complex circuit in electronics. Thus availability of such a tool will be a boon for teachers and students alike.

Among the existing tools for amplifier circuit design, tools for operational amplifiers are available, including online tools, like [3], however, there is no open-source design tool available for designing amplifier circuits with BJT amplifiers and class-A power amplifiers. Most importantly, none of the available tools have a provision for the analysis of the circuit generated. We aim to provide an interface where the user can get the design of the circuit in LTSpice™ after providing certain design specifications. Such a tool will be very useful in classrooms and for other non-industrial purposes where such circuit design is warranted.

The existing design tools are pretty complicated (especially for classroom purposes), difficult to use, expensive, not open-source (user cannot change the source to suit his own purpose) and lack a Spice or similar interface. Moreover, design tools for BJTs, power amplifiers are very hard to find. The commercial tools come with a whole package of electronic design automation tools with lot of circuit-options, which makes them complicated. For learning or teaching a course in analog electronics, only a few numbers of these circuits are required. Further, addition or deletion of components and changing the source according to individual requirements can not be done.

Our design tool tries to overcome most of these problems. It is a simple and user-freindly tool. AutoAmp is easy to operate, takes minimum input and generates an LTSpice™ netlist which can be used to design the circuit in LTSpice™ directly. Being open-source, customized changes can be easily made to the source code to give the desired results; components can be easily added or removed by writing some extra functions in the source code.

Section II describes our design approach in detail including the software design methodology and the circuit design approach. Section III shows our demo/experimentation results and includes screenshots from the working of the tool. The elaborate theoretical and mathematical analysis for each type of amplifier can be found in the Appendix A . Section IV sums up the proposal in the conclusion and talks about possible future work related to the tool.

II. Design approach

This section describes our design approach of the tool in detail. A blackbox representation of the tool is given by Fig. 1.

A. Software Design Methodology

The tool is a command line software designed in C++ programming language. The program has a class named autoAmp which consists of various functions for computation of the amplifiers' components and one function for printing in the file. A struct data type is defined to store all the computed values and is finally used to create the output file. The user is asked for the name of the input file, to select an amplifier of her choice in a menu based environment and finally to enter voltage gain and other parameters based on the type of amplifier chosen. Based on user's choice the respective functions are called which compute the values of components and store then into the struct defined. Now another function uses this struct to create the netlist of the respective type of amplifier into the file specified by the user in the beginning.

B. Circuit Design Approach

1) Single Stage BJT CE Amplifier: We have a designed a small-signal voltage amplifier operating in the audio frequency range. We have used an n-p-n transistor, namely, 2N2222. Two port h-parameters are

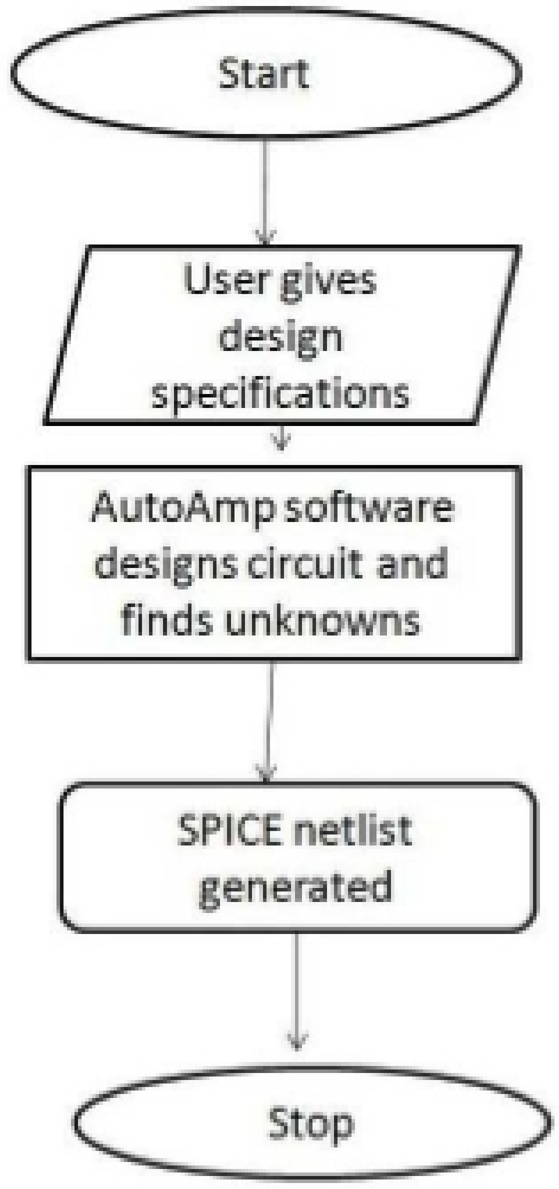

Fig. 1. Blackbox Diagram for AutoAmp

used for circuit analysis. Maximum, minimum, and typical values as required, of the h-parameters are obtained from the transistor datasheet [4]. One method for obtaining the hybrid parameters of the BJT amplifier is given by Al-Zobi et al [5]. These values along with the other known values are used by the software to get optimized values of the circuit components, i.e., resistances and capacitors. The detailed theoretical and mathematical analysis for this part can be found in the appendix A.

2) Two Stage BJT CE Amplifier: This design consists of two CE amplifier stages in cascade. Two amplifying stages thus give us a higher overall gain. The design methodology remains the same as in single stage CE amplifier but is applicable over two stages in this case. The detailed theoretical and mathematical analysis for this part can be found in the appendix B.

3) Operating Amplifiers: This is the simplest ampifier designing strategy in which we are using the inverting and non-inverting configurations of the ideal Universal OpAmp2 (as given in LTSpice™). The detailed analysis is given by Sedra and Smith [6]. Assuming an ideal op amp with infinite open-loop gain, $R_{in} = R_1$. Now to avoid the loss of signal strength, voltage amplifiers are required to have high input resistance. In the case of the inverting op-amp confguration we are studying, to make R_{in} high we should select a high value for R. However, if the required gain $\frac{R_2}{R_1}$ is also high then R could become impratically large (e.g. greater than a few megaohms). Hence in our design we use a different feedback mechanism by which the circuit is able to realize a large voltage gain without using large resistances in the feedback path. The details of the design can be found in the Appendix. Standard circuit design can be used for the non-inverting input configuration as the input resistance is infinity as desired. The detailed theoretical and mathematical analysis for this part can be found in the appendix C.