Investigation on Designing Procedure of Power Supply

N. K. Poon[1] C. P. Liu[1]

[1] PowerELab Ltd. Hong Kong. nkpoon@powerELab.com, www.powerEsim.com

Abstract-Circuit theories describe the behaviors of a circuit, however it is not the full part of knowledge used in the completion of a circuit design. Nevertheless, the rest of part is not commonly discussed or trained. Domain knowledge is mathematically needed to set up a system of homogenous equations and make the system becomes solvable. However, different application and product has different domain knowledge and resulting engineering has no choice but postpone learning that knowledge to the start of their career life, and the worse is that knowledge change from company to company even they product the same kind of product. This Knowledge Oriented design approach required intensive knowledge and skill on engineer, and it also becomes an index of differentiate good and bad engineer. Result Oriented design approach require no domain knowledge, and one can shows the common "try and error" process, is a special kind of Result Oriented approach. Nevertheless, a fast simulator with no extra modeling knowledge can help both Knowledge and Result Oriented design procedure speed up in a much faster way.

I. INTRODUCTION

Circuit theories can describe the behavior of a circuit, it itself is not the whole part of circuit design. To find out all the value or select parts in the circuit, assuming each component need only one value to be determined, we need N equations to solve for that N unknown. It is no surprised that engineer usually cannot find enough equations or in other cases, finding exceed equations for the system.

Assuming client need engineer to design a potential divider, which need to divide the input voltage by half, which is all the client needed, and he send a specification to an engineer and ask for a design solution. Figure 1(a) shows the circuit and its corresponding equation from KVL, obvious we do not have enough equation to solve for R1 and R2. Of course, we should insert the user specification as another equations as shown in Figure 1(b), but interestingly, values still cannot be solved. To complete the job, the design engineer need to insert another arbitrary equation with domain knowledge, Figure 1(c) shows the engineer have some domain knowledge and it try to force the input resistance of the potential divider equal to 2 Ohm. With the new domain equation, the system can be solved. However, the above case do not fully describe all situation, a situation as shown in Figure 1(d) may happen in everyday life, which is the client may give engineer a specification that will turn more equations that needed. Now client is looking for a potential divider that divide input voltage to half, with input resistance of 2 Ohm and the overall power loss should be equal to 1 W at 1 V RMS input. Obviously, no solution can be made to satisfy these requirements.

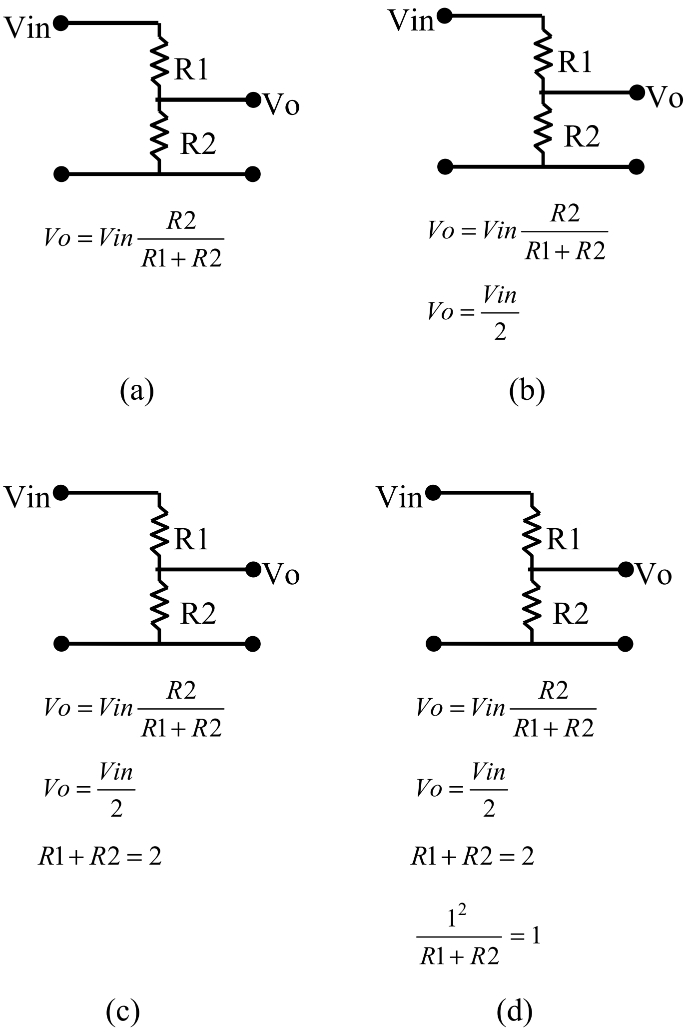


Figure 1 Specification and equations of designing a potential divider

II. KNOWLEDGE ORIENTED DESIGN APPROACH

The above design example illustrates a very typical knowledge oriented design approach. Extra domain knowledge is needed to put into the system in order to make the whole system becomes a homogenous equation system and become solvable. That is very interesting that the two kind of equation, one is generated from circuit theories, and other is the extra domain equation needed are usually learned by the design engineer in two separate environment. It is no doubt that we learn circuit theories in college life, and the other half is learned during our career life. The very typical example of the knowledge generate the domain equation are $B_s < 0.3$ T, $\Delta I = I_p/5$, etc.

Of course, for the condition as described in Figure 1(d), which specification give more than enough equations, engineer needs to take away equations according to some knowledge, or design a system that non of the equation can be fully satisfied. That is why we always hear design is a "compromise." Usually that kind of knowledge is so called "experience."

There always has some noise that industrial people asking academic to teach knowledge that is more "practical" to students, while academic believe they should focus on basic theories. I believe both side are not wrong as no one are having the whole picture. The extra domain equations added or subtracted to form a solvable system depend very much on the type of application, and even under the same application, different level of product will resulting in different knowledge. It is impossible to teach all that knowledge during college period. On the other hand, some student are too much bias on circuit theories to describe a system but very weak in generate more equations to make the system solvable and finally complete the design.

One more implication of the need of extra domain equations is that the domain equations are not unique; otherwise, all 60W adaptors will be the same. In fact, if one collects all that domain equations available, it is more than one needed. Different engineer will pick different equations for their system, and hence each model is different. Most of the said equations can be found in many electronic cookbook or seminar notes and packed in the form of design rules. There have two major reasons for the popular equations spread so wide and used by most of the engineer. First, those can result in reasonable losses and reliability. Second, those can be very easy to solve or make the system very easy to solve. For example, rule of keeping a Flyback turn ratio of 6 : 1 for 12V output, is more popular than the other rule of keeping a Flyback converter duty cycle D=0.6 at low line, although they may be the same after all.

Finding the necessary set of equation is not the end of the story. Engineer need to solve the set of equations to figure out the solution, and it may not be solvable. If it is the case, engineer may need to select any set of equation that can be solvable, in other word, engineer is inherently biased to like those extra domain rules or equations that lead the system to a easy and clear solution.

Figure 2 try to illustrate the typical knowledge oriented design flow, and it is helpful to analysis our common design process. It contain two loops, first is the loop that help to solve for the value needed. The equations set may or may not be including the performance constraint of the final circuit; hence, prediction of the result is usually needed to investigate how the circuit performs. If it is not satisfied, another set of rules are equations which favor the wish are selected to generate another set of solution and repeat the procedure all over again until a satisfaction is achieved.

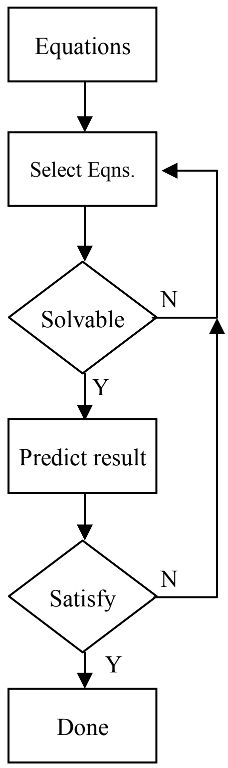


Figure 2 Knowledge oriented design flow

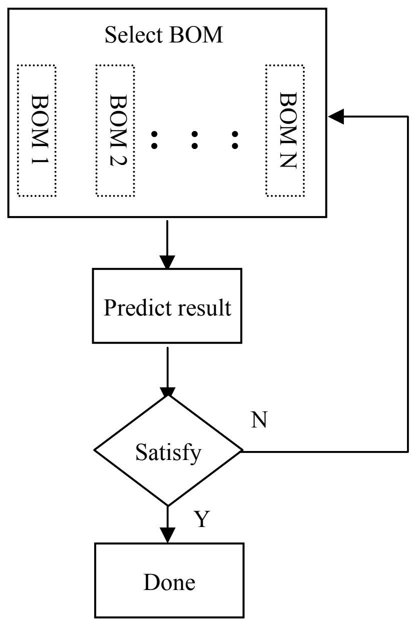


Figure 3 Result oriented design flow

The whole process is driven by knowledge and as the system is complex, it is not unusual for a engineer to take more than several years to manage all those knowledge, especially the domain knowledge part. Nevertheless, the domain knowledge or equations are different from engineer to engineer, and most interestingly is it is different from time to time too. After all, mathematically the equations can be arbitrary. For example, the domain knowledge of choosing suitable converter change from time to time, Flyback converter had considered to handle power within 40W, now 200W Flyback converter can be easily found in market.

In real life, one can find some engineer do not involve the looping processing of estimate whether the design satisfy the wish. In short, they read the specification, choose the equations from his domain knowledge, work out all the component value once and believe that is the ultimate solution as he has the best domain knowledge. Topologically, it is a very special form of Knowledge Oriented design process of making the looping to zero time, and also without estimating the performance of the design. It is very commonly found in designing power supply, however it is not encourage as no prediction involved.

III. Result Oriented Design Approach

Knowledge Oriented Design Approach is a mainstream approach found on educational institute and design handbook. One reason is it give a seamless fitting with the knowledge system engineer are used to learn in their school life. However, the mainstream in industrial is engineer like to use so called try and error method in their design process. . This issue needs to be addressed.

One inherent drawback of Knowledge Oriented approach is it need a lot of knowledge to solve the system or and as the circuit equation are already there, it seems engineer also like to predict the result mathematically on their own way. In most of the time, it is not an easy task, and that is why most engineers has their own like spreadsheet to help for part of the calculation. Some engineer may use simulator to predict the result in order to reduce the computation workload, however, as there exist quite a large gap between the real component and the simulator model, e.g. copper wire winding structure and it's electrical transformer model, engineers usually still need a lot of time to follow on this Knowledge Oriented design approach.

One major drawback of Knowledge Oriented approach is that if the domain knowledge do not cover some better solutions, engineer may never have the chance to design a better power supply.

Figure 3 illustrate the typical Result Oriented design approach, it contain only loops and inherently do not involve solving system equations. Engineers like to obtain the result in some way, e.g. simulation or on working bench. When engineer choose to use real evaluation for predicting the result, it becomes the very famous so-called "try and error" method.

$$\Re(BOM_1) \rightarrow \Re(BOM_2) \quad . \; . \; . \quad \Re(BOM_i) \rightarrow$$

Figure 4 State change of Result Oriented design characteristic

Figure 4 shows how the state change on Result Oriented design flow, BOM_i is a combination of a selected value circuit component at an particular instant i , where $\Re(BOM_i)$ is the result of that set of circuit component. Result can be losses, phase margin or cost, etc.

Obviously, Result Oriented design approach is mathematically a typical discrete iterative problem, where no domain knowledge about the system is assumed to be needed. Most traditional iterative method, e.g. multi-dimensional Newton method, GA Genetic Algorithms, is designed to tackle this typical issue. One can show an abstract design issue has now transformed to a discrete iterative issue of no domain knowledge is needed. One may need to know about circuit theories to predict the system response, or it can simple take real measurement from real unit on the bench. Nevertheless one do not need that extra domain knowledge to start the design. Hence a power supply can be designed by an engineer who do not know any domain knowledge about a power supply.

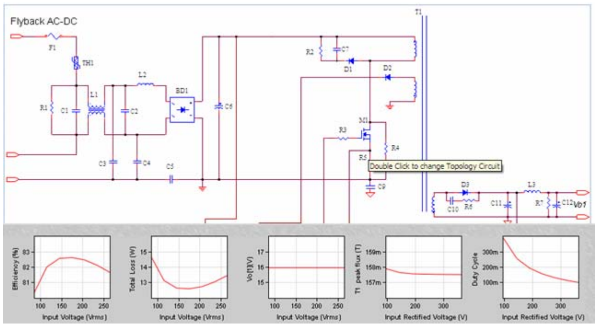
Nevertheless, to make the iterative process fast, one should not rely on real measurement to get the result of $\Re(BOM_i)$, fast and convenient simulator is very helpful. Also the simulator should not ask for domain knowledge or abstract domain concept to carry on the simulation. The best idea of this simulator is that it only ask for real life part No. only. It is interesting that once this simulator builds, it is helpful in both Knowledge Oriented and Result Oriented approach.

The skill of an engineer includes his or her own iterative mechanism as shown in Figure 4. Good engineer may need a few step and consume a few days to fix a design, while a non-experienced engineer may take months and at worst fall into diverge condition.

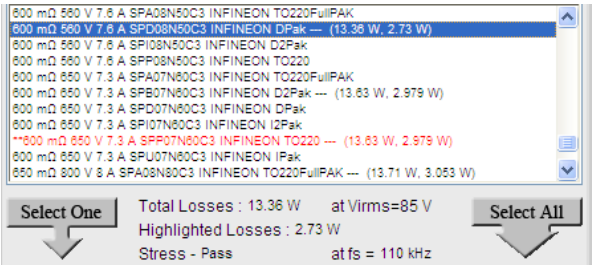
IV. Tool for Result Oriented Approach

SPICE simulator is a very good tool for circuit simulation and can help a lot in Knowledge Oriented design approach. However, user need to learn quite a lot of knowledge about it and some domain knowledge about the it's module, so as to start the simulation process. Transformer design and simulation is a large burden for simulation process, a lot of physical concept has to be learned before using tools available on the market. Nevertheless, a transformer construction is very simple, it is simple putting some wires around a ferrite core. No special abstract knowledge is needed to define a transformer.

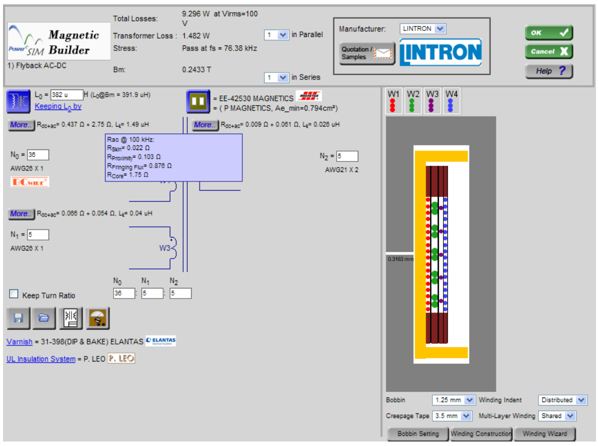
To explore the possibility of using Result Oriented approach for design process, a simulator is needed to build without the need of knowing domain knowledge, which is not found available in the market. As this tool should only ask for information of a BOM, e.g. part no, or drawing of a transformer, no need to build model the circuit, fill in or even understand the coupling coefficient of a transformer. A new tool has build and freely available at Web site www.powerEsim.com to demonstrate the merit and design process of Result Oriented design approach. The tool is open for public and require no login. Some component manufacturers have sponsor the tool and user can click their sponsor's link to have advanced features. The tool also named as PowerEsim.



(a)



(b)



(c)

Figure 5 Tool for Result Oriented approach

Instead of using numeric method to solve for differential equations, the tool is using close form equations to run the result. Approximate has also been used to speed up the calculation process and make the system becomes solvable. Hence the normal time for a steady state simulation is usually within 0.1 second, at least 1000 times faster than SPICE simulation, and those make it very good for Result Oriented and Knowledge Oriented approach.

Figure 5(a) shows a Flyback topologies with a PWM manufacturer's circuit. To start the iteration, one need a initial condition or design first, the tool will recommend a first design to the user, based on only simple operating input voltage range and output voltage and current. Within a few seconds, the tool will recommend the whole BOM and give a summary of results under different operating condition at the lower end of the page. This summary of results will be updated once the BOM is changed.

Figure 5(b) shows every component can be changed and immediately providing the total losses and the component's losses on present and previous component selected. Unlike SPICE tool, user DO NOT need to click an run button again for every change, user simple need to highlight the component, losses and other results will be immediately seen. This feature is very important in Result Oriented design approach, if the speed is not fast enough, the whole tool will just becomes an BOM generator rather than a result predictor that help the user to optimize the design.

Figure 5(c) shows an interface for engineer to build the real transformer with simple core and wire placement. User only need to place wires in the winding space as they wish, as leakage inductance, skin and proximity effect are only depend on the geometrical layout of wires and core. Again, all the corresponding total losses, transformer losses, stress and other result will be immediately seen. It greatly help user to carry on their design iteration. A lot of different wire, e.g triple insulated wire and cores from different manufacturers are ready for user to select as in their real life working environment.

By using the above interfaces and repeat changing the component, one can ask the system itself to chose the best component according to an index, or engineer can do it on their own. A 96% 150 W PFC has been done manually within 3 hours by iterating more than 100 different BOM or design.

As a result, user can build a virtual power supply, which has the same look as in real life. User can try thousands of MOSFET or diodes and re-wind hundreds of transformers within a day in this virtual environment. Which is 100 times faster than in real bench work.

A Smart Optimizer also build by means of GA iteration process can help user to optimize a design too.

Of course, the selection of topologies can also be turned to a discrete parameters and involved in this iteration process. More than 30 topologies are found in PowerEsim, which should cover more than 95% of real life application.

V. Conclusion

One's discussion concludes that the so-called "try and error" approach is a respectful behaviors, good engineer has a method that make the iteration very short and can be many times faster than an bad engineer. Classical iteration method can be apply on this approach. Nevertheless, it is believed that good engineer has a special iteration process which we have not been investigated, and it is very wealth to\o explore a new method that combining a small portion of domain knowledge in iteration process to help fast converge. It turn the domain knowledge of knowing the principle of a power supply to knowing the principle of iteration.

A fast simulation tool PowerEsim, www.powerEsim.com, with no abstract knowledge can make both Knowledge and Result Oriented approach more efficiency.