RESEARCH ARTICLE OPEN ACCESS

Power Inverter Circuit Generator Software

Dr.Anthony Nosike Isizoh[1], Ofunneka Jennifer Okonkwoabutu [2]
Tochukwu Cyprain Onyia[3]
[1]anthonyisizoh@yahoo.com, [2]janieabutu@gmail.com, [3]onyiacypraintochi@gmail.com

Department of Electronics and Computer Engineering, Nnamdi Azikiwe University,
Awka, Anambra State, Nigeria.

--************************----------------------------------

Abstract:

This paper discusses the design and implementation of an inverter system circuit generator software which helps the user perform the inverter design calculations and generates the circuit diagram. It narrows down the inverter system design processes. All calculations involved in the design process will be accurately and automatically done by the software. The software was designed with the aid of a Java programming language and its graphical user interface was implemented using an application programming interface called JavaFx. The software runs on any operating system that has the Java virtual machine installed on it. All that is required of the user is to key in the parameters of the desired inverter, ranging from the desired input voltage to the desired amount of KVA needed while the software does the job of generating values of the required parameters. It also generates the circuit diagram and the bill of engineering measurement and evaluation for the design.

Keywords — **Power inverter, JavaFx, MOSFETS, Oscillator.**

--************************----------------------------------

I. INTRODUCTION

Due to the inconsistent supply of electricity, there is need for a backup which necessitated the design of an inverter system. An inverter is an electrical device which converts direct current (DC) voltage almost always from batteries, into standard household alternating current (AC) voltage so that it is able to be used by common appliances [1]. An inverter is useful for generating power to equipment like household items, computers, power tools and much more by simply plugging the equipment into the inverter. They are used in power generation systems such as electric utility companies or solar generating systems to convert DC power to AC power and they are also used within any larger electronic system where engineering need exists for deriving an AC source from a DC source [2].

Since this device will end up powering most of the home and industrial appliances based on its design rating, the need for an efficient design must be taken into cognizance to ensure that no damage occurs as a result of wrong designs or errors during construction. Thus, care needs to be taken to ensure that accurate calculations are carried out while embarking on the design most especially that of the transformer which could cause heating, explosions and other damages if not accurately designed. Hence, the need for this software

A software is a set of instructions or programs instructing a computer to do specific tasks [3]. This software replicates all manual calculations and will help overcome human mistakes, assumptions, fatigue and unavailability. It takes advantage of JAVA programming language and two Java platforms namely JavaFX and Java Standard Edition to accelerate the inverter system design process.

II. METHODOLOGY AND DESIGN ANALYSIS

There are numerous software methodologies employed for software development which includes waterfall model, spiral model, prototype model and agile model. The waterfall model was used in this project. It requires that each major phase of the development process be completed independently. The requirement phase deals with information gathering and knowledge elicitation regarding the project to be carried out. The design phase will be dealing with the software architecture to be employed as well as detailed design of each of the section of the work. The development phase deals with the algorithm development. The testing phase focuses on testing the subsystems making up the entire system as well as testing the entire system as a whole [4].

A. *Project Initiation*

This work was initiated by the desire to make the inverter system design faster, easier and accurate with the aid of graphical user interface application. Java programming language was used to develop this system with the aid of the combination of two Java platforms which are JavaFX and Java standard edition.

The work took into cognizance the three basic sections of an inverter system listed below:

- Transformer section
- Oscillator section
- Switching section

Transformer section: The adopted design type in this work is the shell type design. This is because of the several advantages it has especially when used in high voltage, large capacity and higher medium voltage applications such as better short-circuit strength, higher mechanical and dielectric strength and better control of leakage magnetic flux [5]. Also, this type of transformer has an efficient cooling system, a flexible design and a compact size. The flexibility in this design lies in the selection of the wire size, which further helps prevent local heating.

Oscillator section: The work was designed using only two major integrated circuits being SG3525 and SG3524. The major difference between these two les in the fact that the latter is open collector while the former is totem pole sink and source [6].

Switching section: Seven common integrated circuits (MOSFETs) namely IRF3205, IRF3808, IRF150, IRF250, IRF608, IRF1407, IRF260. The most important part of this section is the determination of the number of MOSFETs required per switch.

These sections above are the three core sections of the inverter system design process but the work also incorporates a section that aids the user to develop the bill of engineering measurement and evaluation for the design. It also goes further to integrate a browser containing pre-coded hyperlinks featuring some major sections that will assist the user in the design process such as how to cascade transformers, batteries, MOSFETs etc.

B. *Design Requirements*

The software is a desktop application designed with Java programming language and it uses two Java platforms namely Java standard edition and JavaFX. It has the ability to run on any operating system which has the minimum Java run-time environment installed in them. The design spans the three basic sections of the inverter system being the transformer, oscillator and switching section. The user will be able to choose whether he wants to carry out a full design which works sequentially or whether he wants to focus on only a particular section of the design. The user can also choose to design the lamination he wants to use for the design. For each of the designs, a bill of engineering measurement and evaluation can be generated. Additionally, a browser which has quick links to help the user in online research will be embedded in the software. The user can also carry out simultaneous designs with different windows of the software. Also, the software will the user to key in parameters into it to get accurate results on button clicks and above all, it will be user friendly.

C. *System Requirements*

The software is designed to be a desktop application so therefore the target devices for the software are laptops and desktop computers.

TABLE 1
SYSTEM HARDWARE SPECIFICATIONS

Processor	Pentium iv and above
Processor Speed	1.4GHz onward
System Memory	128Mb minimum 256Mb recommended
Cache Size	512KB
RAM	512MB (|Minimum)
Network card	Any card can provide a 100mbps speed
Network connection	UTP or Coaxial cable connection
Hard disk	80Gb

System Software Specification

I. Java Runtime Environment (JRE)
II. Windows 7 and above

D. *Workflow of the System*

The workflow of the system can be depicted in figure 1. It shows a schematic form of the general arrangement of components of the system.

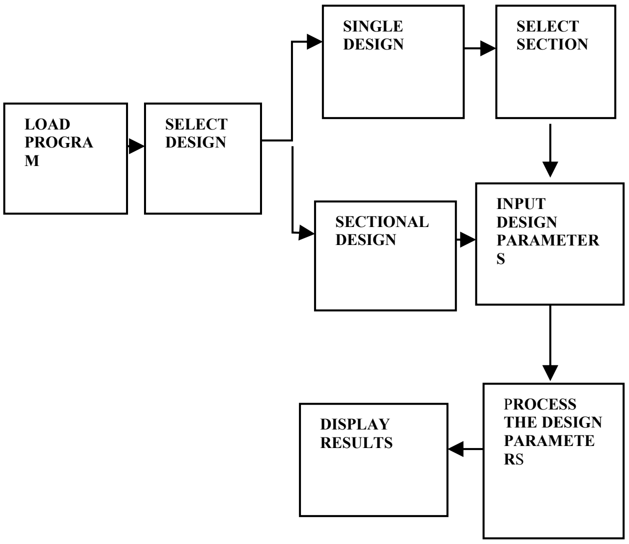


Fig.1 Diagram showing workflow of the system

E. *System Flowchart*

A flowchart depicting the functionality of the system as well as the pseudo code of some sections of the work is shown in figure 2.

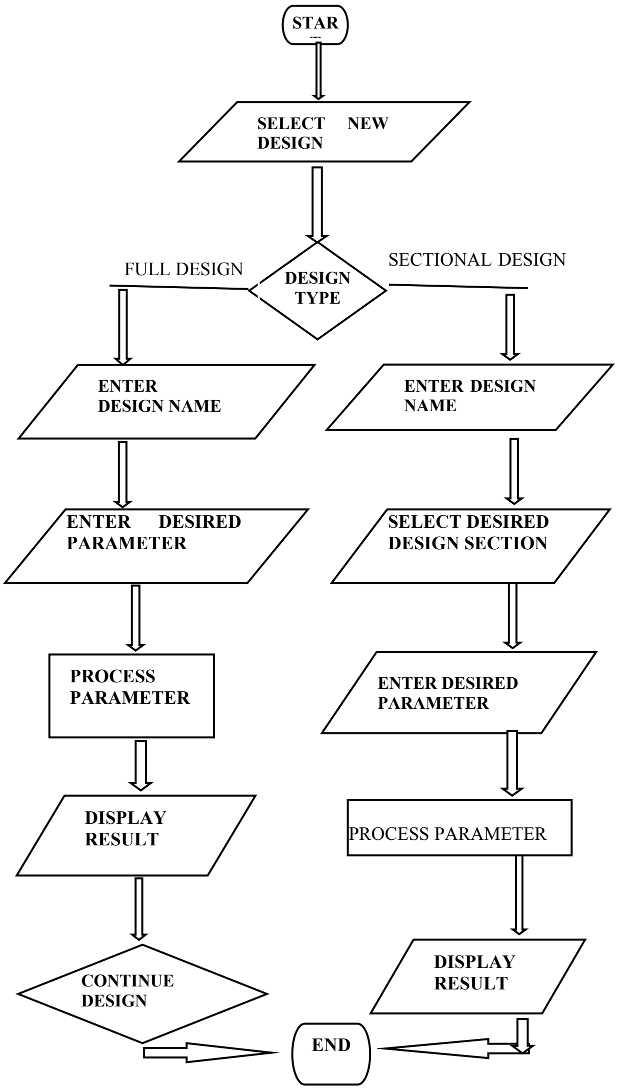


Fig.2 Diagram showing the flowchart of the three basic sections

III. IMPLEMENTATION

A bottom-up implementation approach was used. As the heart of the inverter system design lies in the transformer section, its class was implemented first followed by the oscillator section which was then connected to the transformer section and tested. The subsystem worked as required

The sensing section was implemented separately before being connected to the existing subsystem and also implemented together with it. The entire system was implemented and it was fit for purpose as it solved all the problems stated. Below are the results obtained after the implementing it.

A. *Load Program*

On implementation of the program, the first window that appears is shown in figure 4 depicting the loader window.

loading classes.

SWIFTI-inverter is a software from YCBTR corporation.
For more information, visit www.ycbtr.swiftibee.com

Fig.4 The loader window

B. *Home Window*

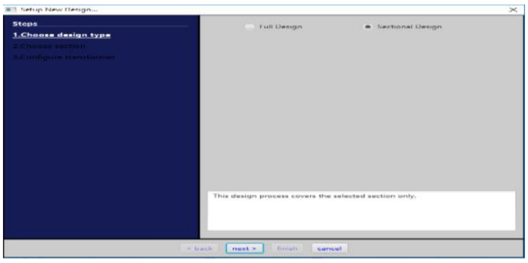
This window appears at the termination of loader window and is depicted in fig. 5

© 2017 YCBTR Corporation. All rights Reserved.

Fig.5 Home window

C. *Design Type Window*

This window pops up when the user clicks the new design menu item found in the file menu. Here the user can either choose the full design or the sectional design as shown in figure 6.



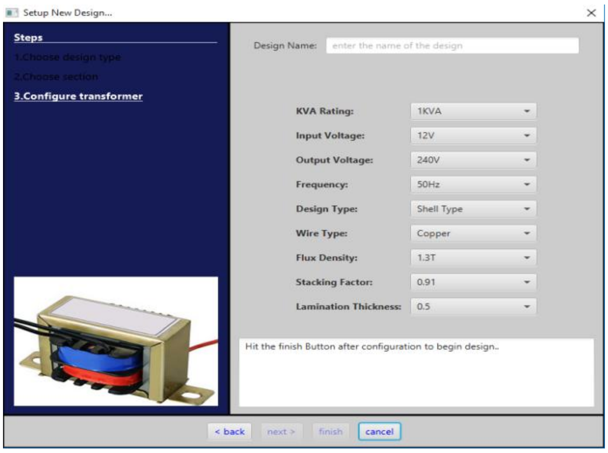
○ Full Design ● Sectional Design

This design process covers the selected section only.

Fig.6 Design type window

D. *Full Design Window*

This window shown in figure 7 pops up whenever the user selects the full design type and clicks the next button while the window shown in figure 8 pops up when the sectional design type is selected by the user.

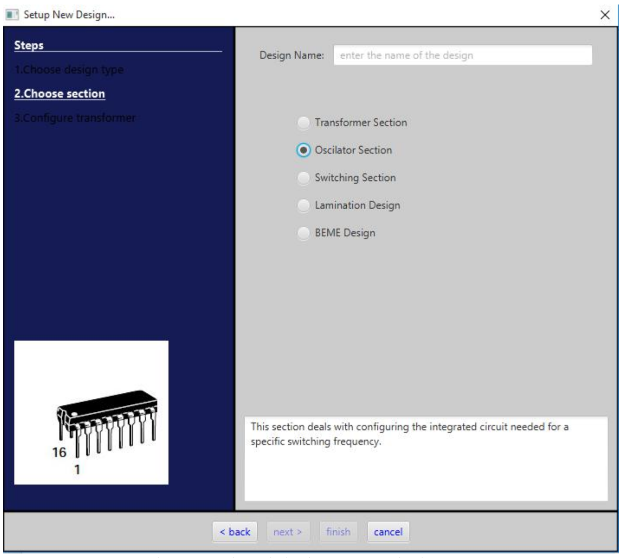


Design Name: enter the name of the design

KVA Rating: 1KVA
Input Voltage: 12V
Output Voltage: 240V
Frequency: 50Hz
Design Type: Shell Type
Wire Type: Copper
Flux Density: 1.3T
Stacking Factor: 0.91
Lamination Thickness: 0.5

Hit the finish Button after configuration to begin design..

Fig.7 Full design type window



Design Name: enter the name of the design

○ Transformer Section
● Oscilator Section
○ Switching Section
○ Lamination Design
○ BEME Design

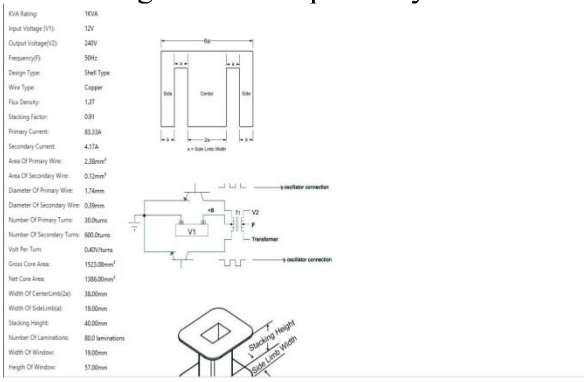
This section deals with configuring the integrated circuit needed for a specific switching frequency.

Fig.8 Sectional design type window

As visualised above, the execution of the full design type begins from the transformer section while the execution of the sectional design type demands the user to select the section of the design they want to carry out.

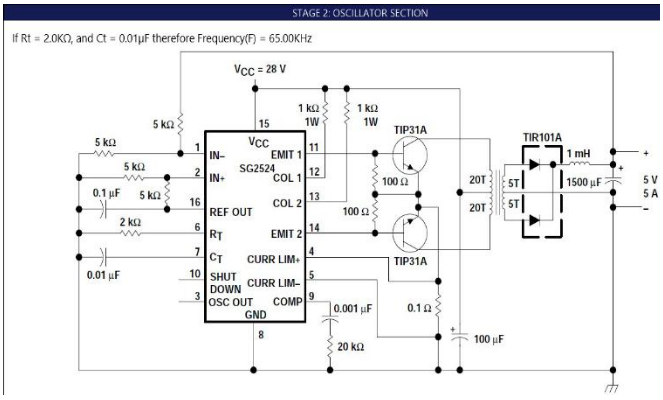
E. *Core Sectional Outputs*

When the user inputs the parameters of any of the sections he desires and clicks on the start design button, the system takes the parameters into the model and then returns their prospective values back to the controller which it uses to update the view. Figures 9, 10 and 11 shows the outputs of parameters modelled in the transformer, oscillator and switching sections respectively.



KVA Rating:	1KVA
Input Voltage (V1):	12V
Output Voltage(V2):	240V
Frequency(F):	50Hz
Design Type:	Shell Type
Wire Type:	Copper
Flux Density:	1.3T
Stacking Factor:	0.91
Primary Current:	83.33A
Secondary Current:	4.17A
Area Of Primary Wire:	2.38mm²
Area Of Secondary Wire:	0.12mm²
Diameter Of Primary Wire:	1.74mm
Diameter Of Secondary Wire:	0.39mm
Number Of Primary Turns:	30.0turns
Number Of Secondary Turns:	600.0turns
Volt Per Turn:	0.40V/turns
Gross Core Area:	1523.08mm²
Net Core Area:	1386.00mm²
Width Of CenterLimb(2a):	38.00mm
Width Of SideLimb(a):	19.00mm
Stacking Height:	40.00mm
Number Of Laminations:	80.0 laminations
Width Of Window:	19.00mm
Heigth Of Window:	57.00mm

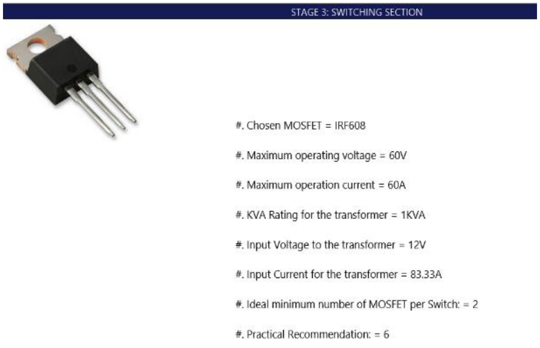
Fig.9 Output of a user's desired parameters for the transformer section



STAGE 2: OSCILLATOR SECTION

If Rt = 2.0KΩ, and Ct = 0.01μF therefore Frequency(F) = 65.00KHz

Fig.10 Output of a user's desired parameters for the oscillator section



STAGE 3: SWITCHING SECTION

#. Chosen MOSFET = IRF608

#. Maximum operating voltage = 60V

#. Maximum operation current = 60A

#. KVA Rating for the transformer = 1KVA

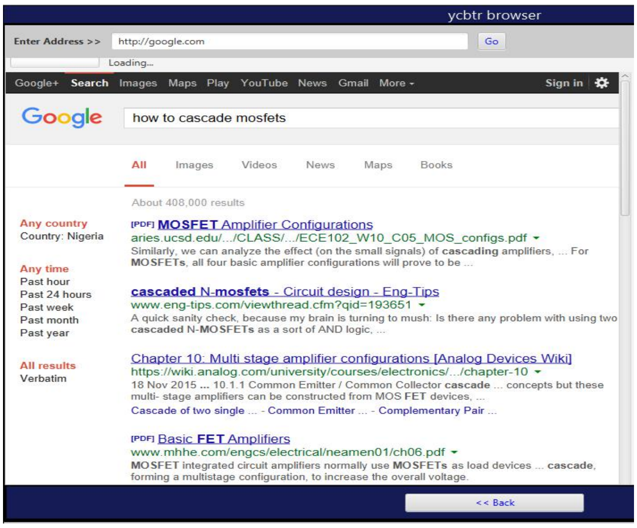
#. Input Voltage to the transformer = 12V

#. Input Current for the transformer = 83.33A

#. Ideal minimum number of MOSFET per Switch: = 2

#. Practical Recommendation: = 6

Fig.11 Output of a user's desired parameters for the switching section



ycbtr browser

Enter Address >> http://google.com Go

Loading...

how to cascade mosfets

About 408,000 results

Any country
Country: Nigeria

Any time
Past hour
Past 24 hours
Past week
Past month
Past year

All results
Verbatim

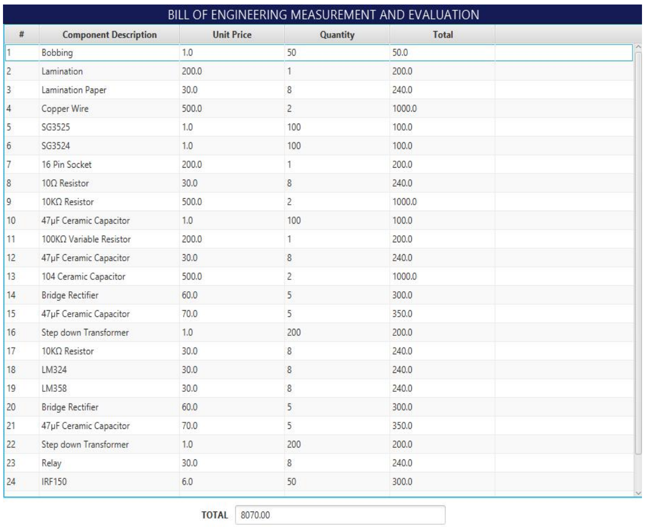
[PDF] MOSFET Amplifier Configurations
aries.ucsd.edu/.../CLASS/.../ECE102_W10_C05_MOS_configs.pdf
Similarly, we can analyze the effect (on the small signals) of **cascading** amplifiers, ... For **MOSFETs**, all four basic amplifier configurations will prove to be ...

cascaded N-**mosfets** - Circuit design - Eng-Tips
www.eng-tips.com/viewthread.cfm?qid=193651
A quick sanity check, because my brain is turning to mush: Is there any problem with using two **cascaded** N-**MOSFETs** as a sort of AND logic, ...

Chapter 10: Multi stage amplifier configurations [Analog Devices Wiki]
https://wiki.analog.com/university/courses/electronics/.../chapter-10
18 Nov 2015 ... 10.1.1 Common Emitter / Common Collector **cascade** ... concepts but these multi- stage amplifiers can be constructed from MOS FET devices, ...
Cascade of two single ... - Common Emitter ... - Complementary Pair ...

[PDF] Basic **FET** Amplifiers
www.mhhe.com/engcs/electrical/neamen01/ch06.pdf
MOSFET integrated circuit amplifiers normally use **MOSFETs** as load devices ... **cascade**, forming a multistage configuration, to increase the overall voltage.

F. *Outputs of other sections*



BILL OF ENGINEERING MEASUREMENT AND EVALUATION

#	Component Description	Unit Price	Quantity	Total
1	Bobbing	1.0	50	50.0
2	Lamination	200.0	1	200.0
3	Lamination Paper	30.0	8	240.0
4	Copper Wire	500.0	2	1000.0
5	SG3525	1.0	100	100.0
6	SG3524	1.0	100	100.0
7	16 Pin Socket	200.0	1	200.0
8	10Ω Resistor	30.0	8	240.0
9	10KΩ Resistor	500.0	2	1000.0
10	47μF Ceramic Capacitor	1.0	100	100.0
11	100KΩ Variable Resistor	200.0	1	200.0
12	47μF Ceramic Capacitor	30.0	8	240.0
13	104 Ceramic Capacitor	500.0	2	1000.0
14	Bridge Rectifier	60.0	5	300.0
15	47μF Ceramic Capacitor	70.0	5	350.0
16	Step down Transformer	1.0	200	200.0
17	10KΩ Resistor	30.0	8	240.0
18	LM324	30.0	8	240.0
19	LM358	30.0	8	240.0
20	Bridge Rectifier	60.0	5	300.0
21	47μF Ceramic Capacitor	70.0	5	350.0
22	Step down Transformer	1.0	200	200.0
23	Relay	30.0	8	240.0
24	IRF150	6.0	50	300.0

TOTAL 8070.00

Fig. 12 Browser section of the software

Fig. 12 BEME section of the software

IV. RESULT

From the bottom-up test approach used, each class was tested one after the other and the combined to form subsystems. These subsystems were then tested one after the other, after which the entire system was tested. The results gotten from the software matches that of the manual calculation.

V. CONCLUSION

The software aims to replicate all manual calculations done in the design of an inverter in order to help overcome human mistakes, assumptions, fatigue and unavailability which occurs when using the manual approach. Java programming language with the aid of two java platforms were used to develop the system. The aim of the software was achieved as an inverter system can be easily designed faster and more efficiently. However, the software can be upgraded and more features added to the system.

REFERENCES

[1] http://engineering.electrical-equipment.org/electrical-distribution/what-is-an-inverter.html

[2] https://HenrichShow.com/posts/674893442638990

[3] https://www.techopedia.com/definition/4356/software

[4] *Software-Engineering* 7th edition by Roger S. Pressman

[5] Hemant Pandey, Upendra Joshi, Protim Paul, Aditya Naithani, "Transformer Design GUI Application for high frequency electrical circuits" International Journal of Advanced Technology in Engineering and Science, Volume No 02, Issue No 09, September 2014. "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.

[6] Theraja, B.L and Theraja A.K, " A Text book of Electrical Technology", volume 2; S. Chand & Co.,2007.

.