Software Engineering for Cloud Computing Role of Secure SDLC

Dhruvil Shah
Institue of Technology
Nirma University
19BCE248

Jurin Vachanni
Institue of Technology
Nirma University
19BCE286

***Abstract*—This paper presents a theoretical analysis of harmonically-terminated high-efficiency power rectifiers and experimental validation on a class-C single Schottky-diode rectifier and a class-F^{-1} GaN transistor rectifier. The theory is based on a Fourier analysis of current and voltage waveforms which arise across the rectifying element when different harmonic terminations are presented at its terminals. An analogy to harmonically-terminated power amplifier theory is discussed. From the analysis, one can obtain an optimal value for the DC load given the RF circuit design. An upper limit on rectifier efficiency is derived for each case as a function of the device on-resistance. Measured results from fundamental frequency source-pull measurement of a Schottky diode rectifier with short-circuit terminations at the second and third harmonic are presented. A maximal device rectification efficiency of 72.8% at 2.45 GHz matches the theoretical prediction. A 2.14 GHz GaN pHEMT rectifier is designed based on a class-F^{-1} power amplifier. The gate of the transistor is terminated in an optimal impedance for self-synchronous rectification. Measurements of conversion efficiency and output DC voltage for varying gate RF impedance, DC load and gate bias are shown with varying input RF power at the drain. The rectifier demonstrates an efficiency of 85% for a 10 W input RF power at the transistor drain, with a DC voltage of across a 98 Ω resistor.**

***Index Terms*—harmonic terminations, high efficiency power amplifiers, load pull, microwave rectifiers, nonlinear analysis, time-domain measurements**

I. Introduction

When the enterprise needs to increase their computation efficiency they had two choices either buy additional hardware or make the IT operation more efficient. Cloud computing is a completely different approach in providing resources to organizations in which computing resource maintenance will not be a concern for the organizations. Cloud Computing is providing services such as storage, software, analytic, intelligence over the internet. It allows the developer to focus more on the business product rather than worrying more about side services. Moreover, It also benefits financially by just paying for the services used within the duration.

If we talk in terms of Software Engineering then developing a particular software from scratch is a big challenge, as we need to choose a suitable model, following a set of activities throughout the project mainly termed as umbrella activities. The above-mentioned things are common to each project but for large scale projects, there are always big challenges on the way. Particularly talking about Cloud services plays a major role as the majority of the services are provided by platforms such as AWS, IBM, Google is not directly implemented by software developers and they need to integrate inside the software.

There are many challenges associated with using conventional software development methods in the cloud. So, before you start developing any application or software in the cloud, you should consider the challenges and issues in the software development lifecycle. This article explores the challenges of the cloud software development lifecycle. Now the question arises why such things are needed, so a few instances can clarify more like, In 2010, Microsoft experienced a breach within its Business Productivity Online Suite that was traced back to a configuration problem. The issue allowed non-authorized users in their offline address books to access employee contact info. It affected only a small number of users, but it is worth noting. In 2012, the DropBox accounts were hacked to more than 68 million. Robbed credentials have gone to a dark web marketplace allegedly. The credentials price at the time was around $1,141 in Bitcoins. Dropbox responded by asking the user base to reset the sitewide password. Hackers robbed 167 million LinkedIn email and passwords and placed them on the dark web in May 2016. LinkedIn, in response, performs a 2 step authentication, which is the option of entering a pin code on your mobile device.An agile software development model, considering the software development challenges from the developer and provider view. Although the study is comprehensive, but it's focused on role categorization and there is a small amount emphasize on development challenges.

para for work done

II. Related Work

So Far much insightful work has been done on the defined topic. Many researchers have proposed different works related to it. Some of them have been depicted here. In[1] the author has proposed a solution moreover related to services of cloud which is most probably concern to software developers rather than audience using it. Also, there is as such no role in SDLC mentioned throughout the paper. In [2] the author mentioned the importance of SDLC but in a very generalized form not a part of deep research. In[3], author considering the provider and software development challenges from the developer's point of view. The study is more often a comprehensive one, but more light are given on role categorization and there is a little emphasize on development challenges.

III. Cloud Computing

Cloud computing can be a distributed computing technology, the basic thought is to automatically split outsized computation programs into numerous little packages through AN oversized computing network, and such package would be two-handed to an oversized system composed of multiple servers and thus the results would be back to the user once search, calculation, and analysis. Through this technology, network service suppliers can trot out tens of assorted or even billions of items of data at intervals some seconds, reaching the constant powerful performance of network services as a result of the "supercomputer".

The sender Cloud Computing refers to delivery and usage patterns of IT infrastructure, that is, to urge the required resources (hardware, platform, software) through the network inside the approach of demands and straightforward extension. the ultimate Cloud computing refers to the delivery and usage pattern of service, It indicates accessing required services through the network inside the way of simple to expand access to services. Such service is IT, code, and Internet-related, it may also be the opposite service. Cloud computing is repeatedly confused with parallel computing, Distributed Computing, and Grid Computing. The concepts of Cloud computing square measure the results of the mix and evolution of virtualization, utility computing (Utility Computing), infrastructure as a service (IaaS), platform as a service (PaaS), code as a Service (SaaS).

Cloud computing services square measure offered to the purchasers in 3 models: Software as a service: during this model, the on-demand application is obtainable to the user through the net. Platform as a service: Providing the platform layer resources as well as the software system and its package development framework. Infrastructure as a service: Provisioning the on-demand infrastructure resources that square measure usually a virtual machine.

IV. Life Cycle of Software Development in the Cloud

The Skyworks SMS7630 Schottky diode in the SC-79 package was selected for the half-wave rectifier. Source-pull was performed at 2.45 GHz with 0-10 dBm available input power for various DC loads in order to identify the combination of input power, fundamental load and DC load resulting in highest efficiency. The best case occurred at 6 dBm input power, with the source-pull contours being shown in Fig. **??**. The on-resistance of the SMS7630 is 20 Ω with the optimal DC load of 1080 Ω. Therefore R_{on} is approximately 2% of R_{DC}, which in theory is 4% of $R_s(f_0)$. From Fig. **??**, a peak efficiency of 87% occurs with infinite harmonic terminations, therefore the achieved 77.6% is very reasonable considering only the 2^{nd} and 3^{rd} harmonics were explicitly terminated.

Measurements of a rectifier designed using the source-pull data show a maximum RF-DC conversion efficiency of 72.8% when matched to 50Ω, obtained after the 0.6 dB matching network loss is de-embedded. The fabricated rectifier and DC load sweep measurements are shown in Fig. **??**. Open circuit shunt stubs are used to present short-circuit terminations at the second and third harmonic. A shunt capacitor is used for presenting the fundamental frequency impedance to reduce size and allow tunability. The reduction in efficiency relative to the source-pull measurements is due to the matching circuit not presenting the ideal impedance found during source-pull.

The class-C rectifier can be applied to improving the efficiency of a wireless powering reception device as demonstrated in [**?**] with a dual-linearly polarized patch rectenna, with a rectifier circuit for each polarization. In this circuit, the first 5 harmonics are shorted and the impedances are validated by calibrated measurements and are presented in [**?**].

V. Software Development Challenges in Cloud

So Till now in broader aspects, we have analyzed some of the methodologies to overcome the problem. Now talking in terms of the Software Development lifecycle we can examine each stage and act accordingly. Mainly the stages considered here are Software requirement challenges, Design challenges, Implementation phase, Testing, Support, and Maintainance.

A. *Determining software requirement challenges*

Before developing any software we need to analyze at the first phase why we need to develop such a thing? What are the requirements? , Is it feasible or not? So Mainly requirements are categorized into three parts Function, Non-Functional and other requirements. Let's look into each phase one by one.

1) Functional requirement : Functional necessities square measure related to applications and business model. however considering the utilization of cloud computing, there square measure challenges with useful necessities:

- Prioritizing specific requirements: specific SaaS
- Requirements should be thought-about and prioritized

2) Non-Functional Requirements: In addition to the useful necessities and package-related non-functional necessities, cloud computing desires to consider other non-functional necessities like security and privacy, reliableness, delay, expandability, and availability. so as to grasp these necessities, first the parameters below the management of client and marketer in A different cloud services model should be known. Cloud service models square measure shown in figure three thoroughly. This model shows the responsibility level of security and privacy and reliability for clients and marketers.

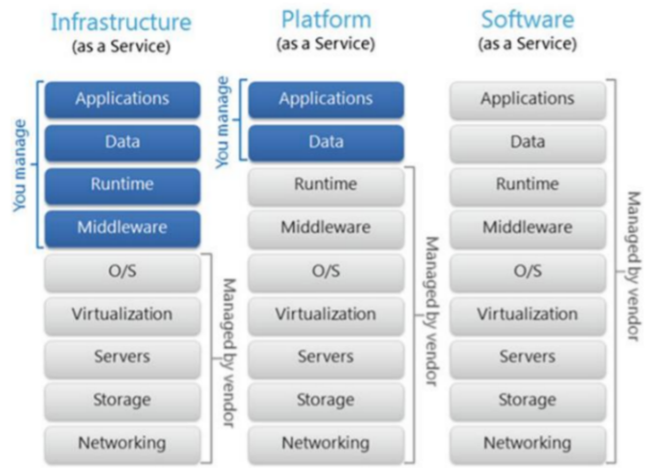

Fig. 1. Services Provided at each interface

different necessities The other necessities that square measure nor useful are neither nonfunctional, and it's necessary to contemplate them in cloud software development, demand analysis part square measure as follows:

3) Lack of Standards for development: Cloud technology is in the early stages of its development. therefore there's no unique commonplace for its API and applications development.

4) Vendor Lock-in: relating to completely different cloud service vendors, customers ought to be ready to opt for acceptable services from completely different vendors. There should be no vendor lock-in for patrons.

5) Cloud Evaluation: Includes enterprise readiness assessment to maneuver toward the cloud. This assessment helps to see the business cases and come of interest

6) Service Level Agreement : The shoppers ought to be assured regarding computing resources within the cloud and the quality of service and performance. These square measures are typically considered in a sort of Service Level Agreement

B. Designing Challenges

Designing doesn't make a UI of software or some templates it has a broader concept which we are going to see in the below description. This phase plays an important role because everything is nearby future is going to be on virtual platforms. Heterogenous resources are used everywhere so compatibility within software can't be compromised. A few of the challenges are mentioned below:

- The first and foremost thing to be considered is compatibility because the cloud is everywhere and it should provide services to each type of virtual platform. Let's take an example of google drive, where billions of users store their information, and also it can be integrated with multiple platforms.
- Reusability should be taken into action.
- Parallelism should be achieved which means the cloud should be at least as powerful as the hardware configuration of the user and smooth services should be provided.
- Apart from all the functionality it primary focus should be on user privacy and error detection techniques. If the service provided to the user is not satisfying then the above-performed actions are in vain.

C. Implementation Challenge

After verifying the design, the implementation/coding phase begins. The main Challenges associated are :

- If you want a large number of users to be using your services then they should be reliable as well as cost-efficient. The transfer cost for data or storage should be minimal. The concern associated is that world s changing at a rapid rate so if it is not developed keeping future technologies in consideration then redesigning costs will be much higher.
- The next thing is topological problems as the things are moving from status to dynamic network architecture where dependency changes. Accessing things all over the globe should be the primary concern of any cloud technology.
- Billing must be done in terms of usage, not the static one which attracts more amount of users and is also beneficial to both parties. Some amount of usage should be costless which would bound the user to use that service more frequently and become used to it like Google photos, drive, email, and many more.

D. Testing challenges

Challenges concerned with this phase are:

- Expandable and Performance test which tests on the efficiency of services when the product is in the market. Severs should be made available to give service to billions of users at the same time. The second thing is Expandable which means when new features are added it should not get slow down or shouldn't lag.
- Security Test is the primary thing needed in any cloud environment. When your product is dealing with major audience which has not to have that much technical background then these things shouldn't be avoided in any situation. The privacy should always be given more priority than some penny of money.
- Integrity testing should be concerned about connections, communication with legacy systems and their connections to the applications, appropriate test scenarios and criteria must be used.

E. Support and maintenance challenges

The issues of cloud software development and maintenance in the SaaS model are examined in this section. The following are the challenges:

- Support for development: If software is created and more resources, such as storage or bandwidth,

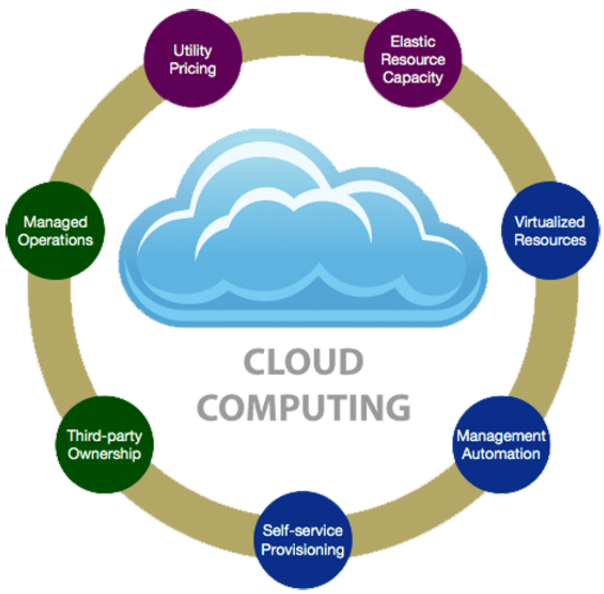

Fig. 2. Overall pictorial representation of cycle

are required, it is the provider's job to meet the demand. The provider is having issues with resource allocation.

- Service Level Agreement: To be able to meet SLA standards, the service provider must effectively manage resources.
- Cloud software support must be able to estimate cost and resource solutions, as well as cut maintenance expenditures.

VI. Conclusion

In summary, this paper addresses high-efficiency power rectifiers designed with harmonic terminations at the RF input, in analogy to high-efficiency power amplifier design with harmonic terminations at the output. The applications of such power rectifiers include wireless power beaming [**?**], recycling power in high-power circuits [**?**] and ultra-fast switching integrated DC-DC converters with no magnetics [**?**].

The theory for an ideal rectification element is based on Fourier analysis and establishes the basic design parameters such as the relationship between output DC resistance and impedance at the fundamental frequency at the rectifier input which optimizes efficiency. The analysis also predicts the time-domain waveforms at the terminals of the rectification element and the efficiency as a function of on-resistance and DC output resistance. Specific results are derived for class-C and class-F^{-1} classes of operation, as they are defined for power amplifiers. These two cases are chosen for experimental validation with a 2.45 GHz diode and 2.14 GHz transistor rectifier, respectively. It is straightforward to repeat the derivation for other classes of operation, such as class-F as shown in detail in [**?**].

The experimental results show that good agreement can be reached between theory and experiment with a Schottky-diode single-ended rectifier with finite class-C harmonic terminations, resulting in 72.8% efficiency for input power levels in the mW range, intended for wireless power harvesting detailed in [**?**], [**?**]. A GaN pHEMT class-F^{-1} power rectifier achieved 85% efficiency with 40 dBm input power across 98-Ω DC load with a DC output voltage $V_{DC} = 30V$. The efficiency and output voltage of the self-synchronous rectifier are shown to depend on the input power at the drain, the impedance at the gate port and the DC load at the output drain bias line, but not on the gate bias.

Time-domain large-signal measurements of a class-F^{-1} power amplifier configured as a rectifier show that one can accomplish the same rectifier efficiency as the amplifier drain efficiency in self-synchronous mode without external gate RF drive. This is somewhat surprising, and to the best of our knowledge, the first time this type of high-efficiency rectifier has been demonstrated.

Acknowledgment

The authors would like to thank Dr. David Root and Dr. Jean-Pierre Teyssier at Agilent Technologies for the loan of the time-domain nonlinear measurement equipment and TriQuint Semiconductor for the donation of the transistors.

References

[1] Pramod, N., Muppalla, A.k. and Srinivasa, K.G., “Limitations and Challenges in Cloud-Based Applications Development”, In Software Engineering Frameworks for the Cloud Computing Paradigm, Springer, pp. 55–75, 2013.

[2] Harman, M., Lakhotia, K., Singer, J., White, D.R. and Yoo, S., “Cloud Engineering is Search Based Software Engineering Too”, Journal of Systems and Software, Vol. 86, Issue. 9, pp. 2225–2241, Sept 2013.

[3] Patidar, S., Rane, D., and Jain, P., “Challenges of Software Development on Cloud Platform”, World Congress on Information and Communication Technologies (WICT), pp. 1009-1013, 2011.