REPUBLIC OF TURKEY

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

ROLE OF VERY-LARGE-SCALE-INTEGRATED (VLSI) CIRCUITS AND LOGIC GATES IN INTERNET OF THINGS (IOT)

Ali Abdulkareem Rajab ALMAZEDI

Master`s Thesis

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Istanbul, 2022

ROLE OF VERY-LARGE-SCALE-INTEGRATED (VLSI) CIRCUITS AND LOGIC GATES IN INTERNET OF THINGS (IOT)

Ali Abdulkareem Rajab ALMAZEDI

Electrical and Computer Engineering

Master`s Thesis

ALTINBAŞ UNIVERSITY

2022

The thesis titled ROLE OF VERY-LARGE-SCALE-INTEGRATED (VLSI) CIRCUITS AND LOGIC GATES IN INTERNET OF THINGS (IOT) prepared by ALI ABDULKAREEM RAJAB ALMAZEDI and submitted on 20/05/2022 has been **accepted unanimously** for the degree of Master of Science in Electrical and Computer Engineering.

____

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Faculty Of Engineering and Architecture, Altinbas University	____________________
Asst. Prof. Dr. Sefer KURNAZ	Faculty Of Engineering and Architecture, Altinbas University	____________________
Asst. Prof. Dr. Mohammed ABUBAKR	Faculty Of Business , Antalya Bilim University	____________________

I hereby declare that this thesis meets all format and submission requirements of a Master`s Thesis.

Submission date of the thesis to the Graduate Education Institute: ____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ali Abdulkareem Rajab ALMAZEDI

Signature

DEDICATION

I devote and pledge this research work to my supervisor who is salient for guiding me through whole research work as well as my family for always assisting me in my hard time.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor Asst. Prof. Dr. Abdullahi Abdu IBRAHIM for guiding and helping me along the way in writing this dissertation. Discussing my progress, problems, and ideas with my supervisor Asst. Prof. Dr. Abdullahi Abdu IBRAHIM a couple of times every week helped me tremendously in understanding the logic behind the research. It made me better realize the technical need for this research work.

ABSTRACT

ROLE OF VERY-LARGE-SCALE-INTEGRATED (VLSI) CIRCUITS AND LOGIC GATES IN INTERNET OF THINGS (IOT)

ALMAZEDI, Ali

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: 01/2022

Pages: 66

The aim of this thesis is to work out the Role of Very-Large-Scale-Integrated (VLSI) Circuits and Logic Gates in IOT, we do that by Developing Real Life, Task Oriented Applications for the Internet of Things. The Internet of Things is becoming ubiquitous. A growing number of objects are being equipped with devices that interact with its environment and exchange data via the Internet. Such devices are not powerful enough to run Logic gates (an implementation of the Task Oriented Programming paradigm written in Clean) applications. The VLSI circuits Embedded Domain Specific Language was created with one goal in mind: to bring Internet of Things devices and the Task oriented programing paradigm together. But so far, only trivial applications were developed using VLSI circuits and Logic gates. This paper assesses whether VLSI circuits can be used to develop real-life applications. Auto house, the home automation application developed during the research, guided improvements in the m Task development environment and unearthed problems yet to be resolved. This dissertation focuses on automating the design process of analog designs in advanced CMOS technology nodes, as well as reciprocal quantum logic (RQL) superconducting circuits. For CMOS analog circuits, our design automation technique employs digital automatic placement and routing tools to synthesize and lay out analog blocks along with digital blocks in a cell--- based design approach. This technique was demonstrated in the design of a digital--- to--- analog converter. In the domain of RQL circuits, the automated design of several functional units of a commercial Processor is presented. These automation techniques enable the design of VLSI--- scale circuits in this technology. In addition to the investigation of new technologies, several new baseband signal processor architectures are presented in this dissertation. These architectures are suitable for low--- power mm3--- scale WSNs and enable high frequency transceivers to operate within the power constraints of standalone IoT nodes using Matlab.

Keywords: IOT, VLSI, Logic gates, CMOS, Nodes.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

AI	:	Artificial Intelligence
BIS	:	Bank of International Settlement
CPP	:	Cryptocurrency Price Prediction
DLT	:	Distributed Ledger Technology
FTR	:	Funds Transfer Regulation
FIR	:	Fixed Investment Rate
LCL	:	Legal Cryptocurrency Law
ML	:	Machine Learning
OTF	:	Organized Trading Facility
PoS	:	Proof of Stake
PoW	:	Proof of Work
TCI	:	Technical Trade Indicator

1. INTRODUCTION

1.1 INTRODUCTION:

A network of physical devices combined with electronic sensors and connected to the Internet is called Internet of Things (IoT) as shown in Figure 1 These devices can collect data from surrounding environment, communicate with each other, and exchange the information [1]. Connected objects can provide valuable information from their nearby environment [2].

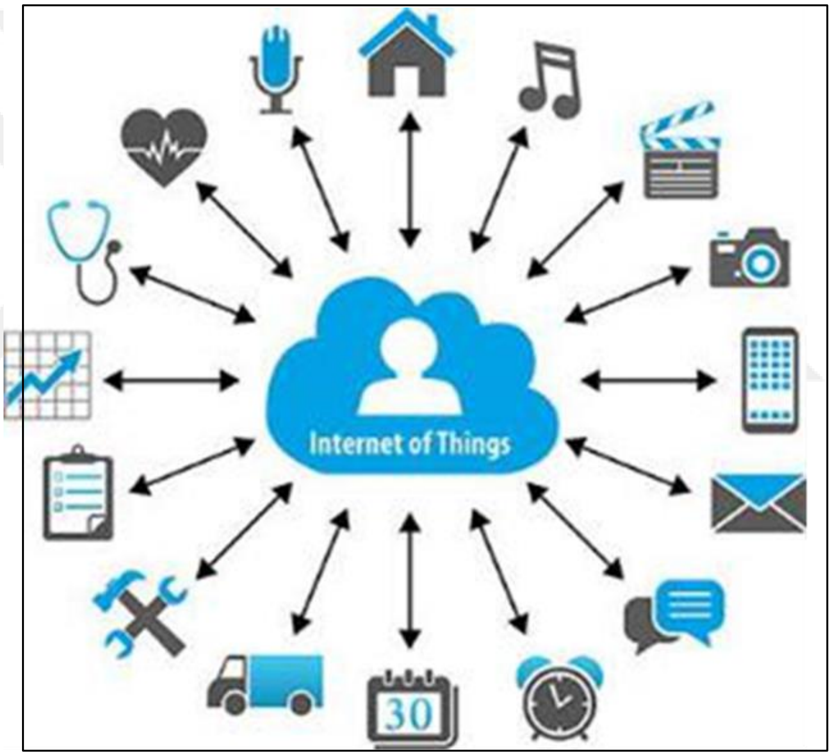

Figure 1.1: Internet of Things [3]

Emerging Techniques for IoT to tackle the challenges, lot of new techniques are proposed in IoT system architecture. The Internet of Things (IoT) consists of a network of "things" (devices, computers, systems, etc.) that interact with each other via the Internet. These components can exchange data, monitor and manage each other. IoT is a global, growing phenomenon. According to Looking to the future, Cisco IBSG predicts there will be 25 billion devices connected to the Internet by 2015 and 50 billion by 2020[4]. IoT has been used in a myriad of applications including home automation, fitness tracking, health care, warehouse monitoring, agriculture and industry manufacturing. IoT devices can be dedicated servers, personal computers, tablets, smartphones, smartwatches or compact devices operated by microcontrollers. Microcontrollers are small, cheap computers with limited resources and low power consumption commonly used to interface with the real world.

They often gather data from sensors (movement, light, temperature, etc.), act on actuators (motors, LEDs, switches, etc.) and communicate with other devices. Task Oriented Programming (TOP) is a new programming paradigm used to develop online, collaborative applications. Its central concept, a task, can be used to model diff erent types of work performed both by users and systems. TOP provides a high level of abstraction, liberating the programmer from the burden of technical details, such as user interfaces. The Logic gates system implements TOP in the functional programming language Clean [5] as an Embedded Domain Specific Language (EDSL). It automatically generates as many assets as possible, turning it into a great tool for rapid prototyping. Given a Logic gates program, the system automatically generates a web application that can be accessed via a web browser. Users can access this application to inspect and work on tasks. The system has been proved useful in many fields, including incident response operations and navy vessels automation Although Logic gates applications often require user interaction, some tasks could be automated. Examples are tasks that interact with the external world: reading room temperature, blinking an LED, detecting movement, unlocking a door, etc. The Logic gates environment could benefit from such automation. Microcontrollers pose as great candidates to interface with the external world. They are aff ordable, energy effiicient and are seamlessly combined with sensors and actuators. Unfortunately, — due to hardware limitations — microcontrollers are not suitable to run Logic gates tasks. To bridge this gap, the VLSI Domain Specific Language (DSL) was created to enable the execution of simple tasks on microcontrollers, bringing such devices to the Logic gates world. Task Oriented Programming (TOP) is a new programming paradigm used to develop online collaborative applications. In a TOP application, users — both people and other systems work together to accomplish a common goal. Its central concept is called a task, an abstraction that can be used for many diff erent types of work. This abstraction, along with other TOP concepts, allows the programmer to focus on design decisions instead of technical details.

TOP programs are declarative: they focus on what work should be performed, rather than on how to perform it.

1.2 PROBLEM STATEMENT

The work scenarios stated in research surveys, clearly shows the following are observed as the prime focus, need for storage and data handling capabilities for the IoT devices. An adequate data computation, low latency system with a cost-effective solution for cloud service utilization is needed. Need for further improvement in the training simulation where the user should be exposed to real-time data from the machines to utilize their training simulation better. By comprehending the above statement and the initial introduction section, it would undoubtedly give an insight of what is to be done. The solution for all the above problems combined is the integration of the IoT and XR. Fog solves the delivery of real-time data to the immersive headgear. And now the user has access to real-time data from the working industrial IoT during 3-D simulations. The integration also provides virtual interaction with the actual physical device thus delivering an immersive mixed reality experience to the user.

Even though VLSI was created to allow Logic gates tasks to run on IoT devices, it has not been proved capable of running real-life applications yet. The examples built during its development were simple demonstrations and were far from real-life IoT applications. Given that, I propose the following I plan to tackle the research by example. Namely, trying to develop a real life IoT application using VLSI. The attempt to develop such an application should display VLSI 's capability to create real life applications while displaying new opportunities to improve the development process. The application I chose to develop tackles a popular problem in IoT: home automation. This application would be responsible for automating simple home management tasks such as turning the central heating system off when the room is warm or opening up the curtains at a set time. The proposed application requires several IoT devices equipped with sensors (temperature, light, humidity, etc.) and actuators (LEDs, motors, relays, etc.) spread across rooms.

The work scenarios stated in research surveys, clearly shows the following are observed as the prime focus, need for storage and data handling capabilities for the IoT devices. An adequate

data computation, low latency system with a cost-effective solution for cloud service utilization is needed. Need for further improvement in the training simulation where the user should be exposed to real-time data from the machines to utilize their training simulation better. By comprehending the above statement and the initial introduction section, it would undoubtedly give an insight of what is to be done.

1.3 RESEARCH QUESTIONS AND/OR HYPOTHESES

Attacker may perform a selective packet dropping or change its behavior over time in order to deceive the underlying detection mechanism. How can we exploit the evolution of a node during a time period to identify the nature of its behaviors? Although Business infrastructures utilizing machines from small scale to large scale have seen the advantages of field ready IoT devices. These IoT devices are also called Industrial IoT, which accumulate a vast amount of data from simple temperature sensors to multiple vibration sensors from machines optimized to work without stop. These data are stored for monitoring the performance and reliability of the machines. Later, the introduction of cloud computing made use of the extensive data in separate data centers where the data is used to analyze the performances of the machines and schedule calibration or maintenance breaks and other tasks which required computation. The established system performs well for functions which do not need low latency services [7]. However, it lags behind for operations requiring immediate attention. Moreover, the cloud platform is required to stay always active for live monitoring and data computation, this causes a significant constraint on the operational cost. The scenarios will benefit from Fog concepts, which could provide a solution for storage, computing, and control by distributing the access to the current task by workload-based priority [8].

1.4 THESIS CONTRIBUTION

This thesis focuses on circuits, methodologies, and tools for designing energy-efficient digital very-large-scale integration (VLSI) modules that operate in the precise, near-PoFF, or inexact regions. The clock-distribution network, sequential, signal-processing modules for the data channel, and embedded memories, among other circuits, are all tuned for low energy usage.

The framework should be simple to implement at all industrial levels. Antenna Toolbox, Machine Learning Toolbox, Wireless Toolbox, and more advanced MATLAB toolboxes

2. RELATED WORK

2.1 INTERNET OF THINGS DEVICE

The function of IoT device mainly consists of collecting data and processing data and storing data. The development board selected for this thesis is Arduino Uno.

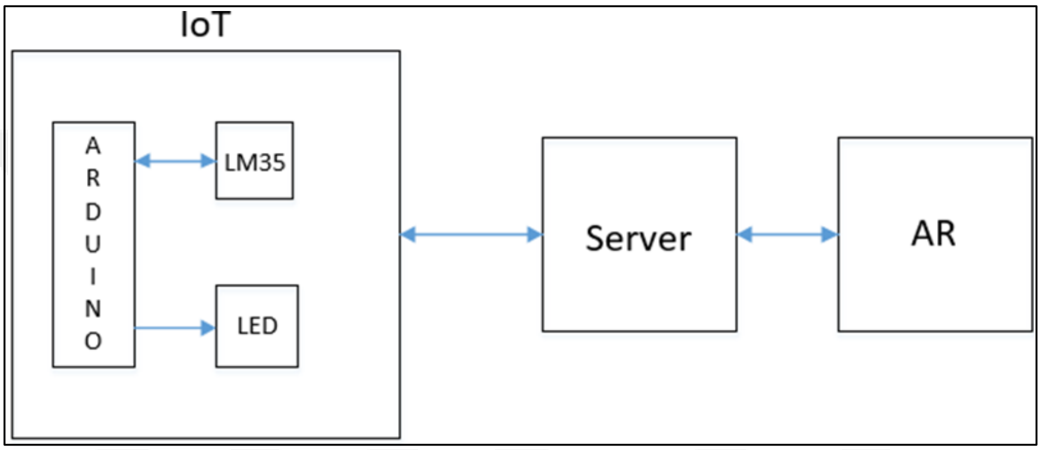

Figure 2.1: IoT Block Diagram [8]

Arduino is an open-source microcontroller development board based on Atmel micro controller chips and development by Arduino.cc. It is programed using an Arduino IDE (Integrated Development Environment). It works on 5 volts powered through battery or USB when connected to computer. It is perfect development board to run experiments and projects for beginners to experts. It is widely used in Internet of Things do-it-yourself project for the reason that it is mainly open source and vast number of communities working with the device. It is clear to say that Arduino is the most popular embedded development board among hobbyists. The reason behind choosing this board for this thesis is well understood. The support available for the board in means of firmware, software is within reach, and we can move ahead with the thesis objectives with ease. The board we have chosen for the research is Arduino Uno which host an 8-bit AVR Atmel ATmega328 [9] with a clock speed of 16MHz. The Uno development board also host numerous input and output ports using which an external device such as sensors, actuators, etc., can be programed and added to the board with ease [9]. The technical specification of the Arduino Uno can be found in Table 2.1.

Figure 2.2: Arduino Uno Development Board [9]

Table 2.1: Arduino Uno Technical Specifications [9]

Parameter	Specification
Input Voltage	**7-12v (DC Jack) 5v (USB)**
Max Output Current	**40mA Per Pin**
Processor	**ATmega328P**
Oscillator	**16 MHz**
Analog Pins	**6**
Digital Pins	**14**
PWM Pins	**6**
Other Specifications	**4 LED's (TX, RX, Pin 13, Power Indicator), Reset Switch**
Flash	**32 kB**

2.2 EMBEDDED PROGRAMMING

The Arduino.cc community also provides user-friendly graphical user interface enabled software Figure 2.3 which make it all the more ideal for the research objective. The software highlights the keywords and functions which are in sync with the Arduino libraries in the backend automatically. Thus, allowing the user to focus on the program scripts and its output results. The programming language used is embedded C++

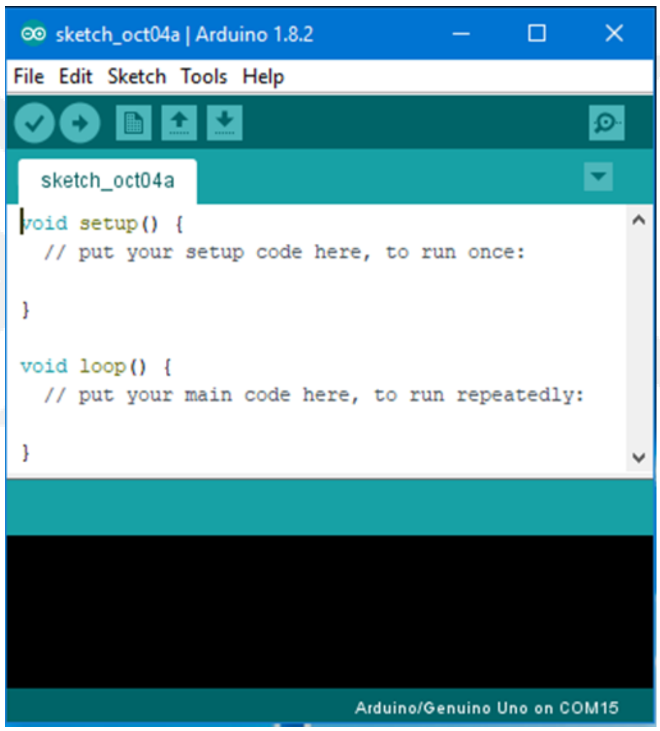

Figure 2.3: Arduino Integrated Development Environment [10]

The IDE presets the programing environment with two major functions.

i. Void setup ( )

ii. Void loop ( )

Void setup ( ): It marks the start of the programing script. It is a built-in function in which the initialization of the user variables and the input/output ports which will be in use are described.

Void loop ( ): It is a built-in function with-in which the execution of tasks take place. The task described inside the loop function run in an endless loop at the clock rate of 16MHz for Arduino UNO [11]. For example, take a look at a basic program script used to blink a light emitting diode.

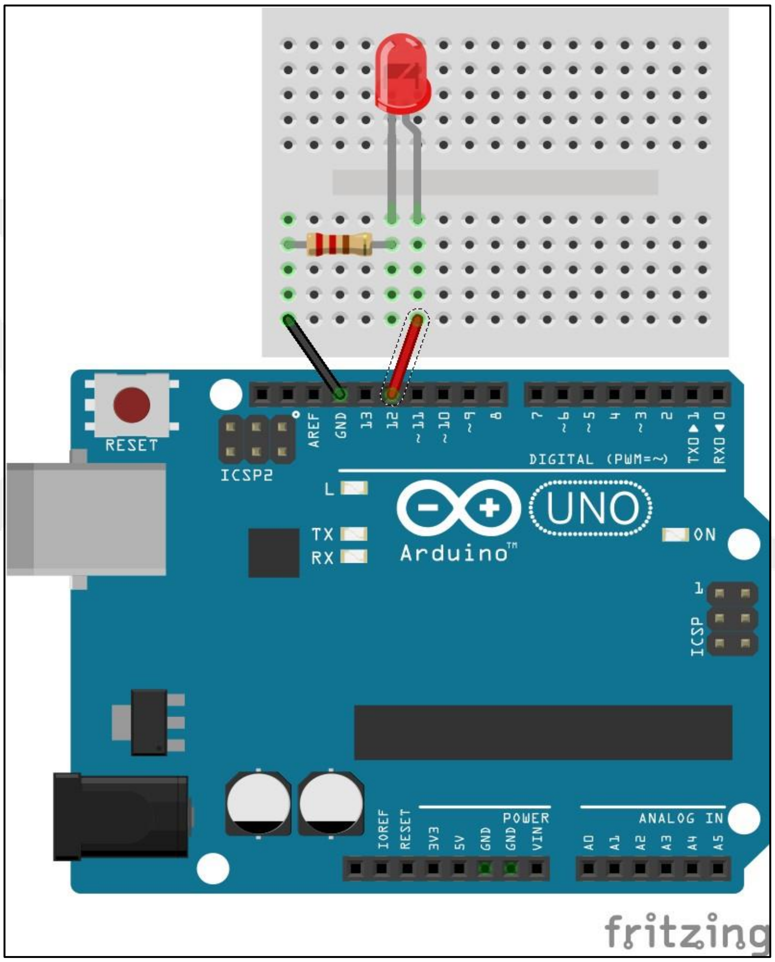

Figure 2.4: Circuit Design for LED Blink [12]

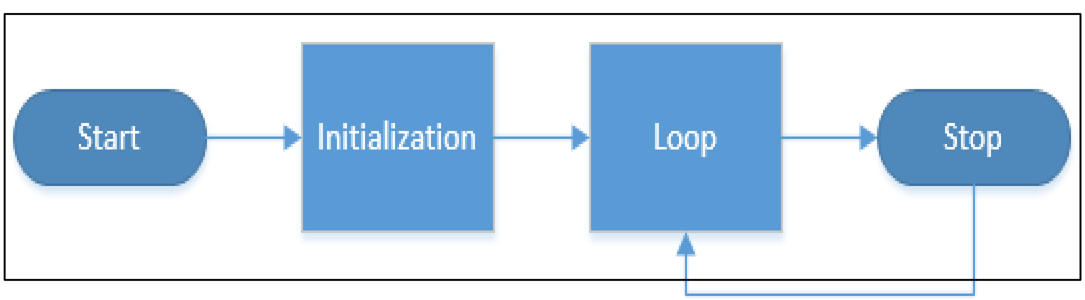

Figure 2.5: LED Blink flowchart [13]

```
Blink §
void setup()
{
  pinMode (12,OUTPUT);
}

void loop()
{
  digitalWrite(12,HIGH);
  delay(1000);
  digitalWrite(12,LOW);
  delay(1000);
}
```

Figure 2.6: LED Blink script [14]

In the above example script “pin Mode (12, OUTPUT);” as explained earlier, this line shows us that pin number 12 is initialized as output port. Further down the script “digital Write (12, HIGH);” is the task the Arduino is to perform, the script says the Arduino to make the digital port 12 to HIGH meaning set a digital voltage value to the port for “delay(1000);” delay of 1 second and the same to LOW state for a delay of 1 second to set it off.

2.3 SENSOR AND ACTUATOR

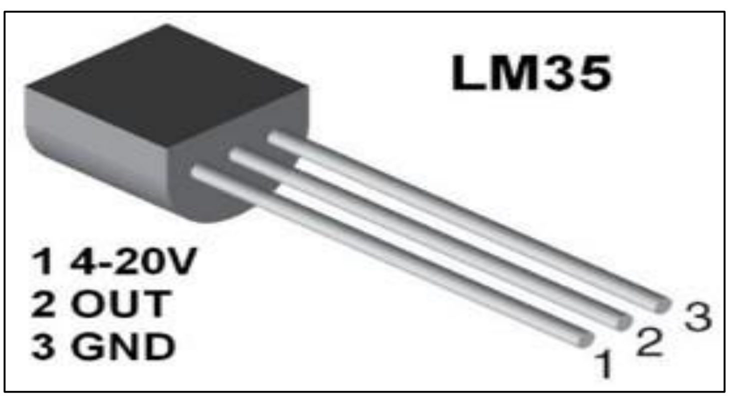

Figure 2.7: LM 35 with Pin Configuration [15]

The Arduino board hosts a temperature sensor and an LED to demonstrate the working of features available in an IoT device. This is one of the basic operations an IoT device could be programed to do which can clearly highlight the functions or tasks carried out by an IoT device over a graphical user interface enabled dashboard. In our case it is through a basic HTML page over the internet. The temperature sensor used in this research is the LM 35 it is an analog temperature sensor with three pins, as shown in the figure 2.7. It is a product of a company

called Texas Instruments which specializes in making various types of semiconductors and integrated circuits [15]. LM 35 is calibrated to give temperature value in Celsius by default. It works on 5volt dc supply and has a range of Negative 55 to Positive 150 Celsius. It is an idle sensor for research purpose and simple to use in an Arduino programming environment.

The actuator used is a simple red color light emitting diode. It performs as a substitute to any other actuators by simply being an output to indicate a response to an input task such as On/Off.

2.4 DATA COLLECTION

This section consists of the C++ program script uploaded to the Arduino Uno. The elaborate explanation of the code will follow below every significant part of the script which harbors the Fog concepts.

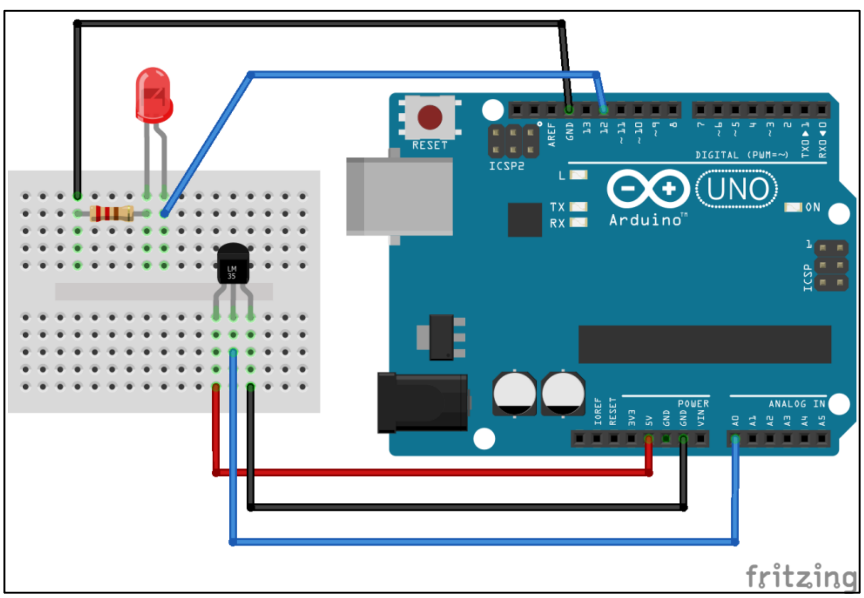

Figure 2.8: IoT device Circuit Design [16]

XrCode4.0 §

```
int reading;
float tempC,tempF;
char data[5];
bool receivedCmd = false;
long int previous;
int state,pinread;
```

Figure 2.9: IoT script Data Initialization

The above initialized global variables come in handy with the functions performed by the Arduino. Some of the variables will have stronger meaning when used in the functions which follows below

The pin 12 is used as an output and pin A0 is used as input to receive data from the temperature sensor The functions that are to be performed by the Arduino are distributed into local functions within the scripts. This shows the concept of Fog can be implemented into the root level of programing an IoT board. The distribution of functions decentralizes the workload and guides the Arduino to perform the tasks when necessary, rather than running in an endless loop of execution. The void loop functions are shown in Figure 2.10.

```
void loop()
{
 serialread();
 processCommand();
 tempRead();
}
```

Figure 2.10: Void Loop Functions

The serial read function Figure 2.12 and figure 2.13 looks for any input available at the port during every start of new execution loop. If the data is available at the port, then another loop starts making the port to stay open while it stores the incoming data in an array element called “data’ ’After the data is stored the function stores an internal acknowledgement by saving a bool element into the variable “received Cmd = true”. This will indicate whether the next function should execute or not based on the bool value of true or false.

Figure 2.11: Serial Read Flowchart [17]

```
void serialread()
{
  if(Serial.available()>0)
  {
    int i = 0;
    while(Serial.available()>0)
    {
       data[i] =  Serial.read();
       i++;
    }
    receivedCmd = true;
  }
}
```

Figure 2.12: Serial Read Function script

2.5 INTERNET OF THINGS SERVER

Server is a computer program, or a computer designated to provide functionality for other computers or programs [18]. The general idea behind the server architecture is server-client relation, the use of global platform to run applications which shares the interfaces between devices. Centralization of applications to run on a single communication bus, lend to the current systems which run in local and global network. Which brought the rise of cloud servers and portability by enabling people to access content over Internet. Later in the technological development, the workload on servers began increasing high causing severe traffic in the network causing application interfaces to fail. Then the term of decentralization brought about new ways of transporting data from one point to another. The pendulum of centralization and decentralization has been an ongoing problem faced even in today's technological era. The term Fog computing was recently coined by CISCO who are the lending pioneers in networking. The fundamentals of the server-client come into play but, this time the process is made flexible so that the pendulum can swing back and forth. Meaning, the devices can change their functionalities based on all the necessities involved in handling a task. For example, a person using a smart phone to view some navigational content, can be moved over to a car's stereo which also utilizes similar features synced from phone. However, the content has moved from phone to stereo, and this shows the portability in platform. Both devices running in a synced network but now the stereo has taking over the task of showing the person the data and the phone has simply become the communication channel through which the stereo is synced.

Fog concepts dispels the idea of distributing the computation, communication, control and storage thus, creating a decentralized process [19]. However, it also works in sync with the cloud servers creating the smooth transition to centralization for computation heavy tasks and moving back to distributed system for priority and less computational tasks enabling a fast-responsive system

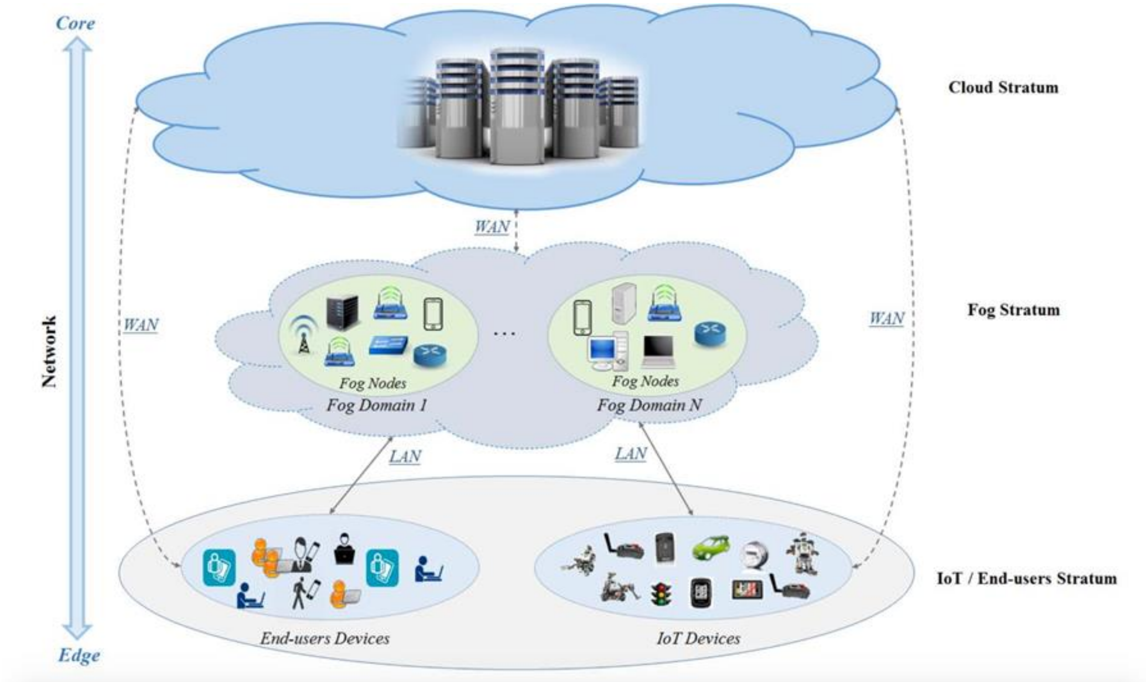

Figure 2.13: Topology showing the Fog is closer to the End-Users [20]

2.6 SERVER DESIGN

The elaborated server section showing the internal structures involved in the server program can be seen in Figure 2.15. The main part will be the server code which interfaces the other application such as communication, user interface and storage to provide client with complete service expected from the server at hand. The current task at hand is to view the real-time data from an IoT device which is in sync with this server and provide that data to any number of clients accessing this content in the local network as well as on the Internet.

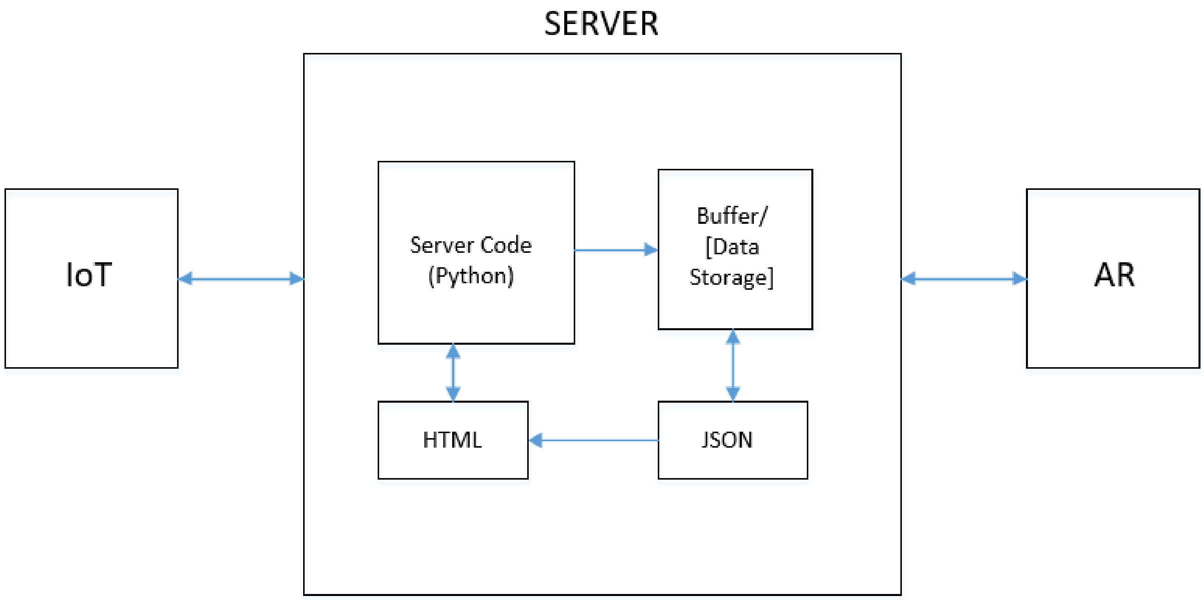

Figure 2.14: Server Block Diagram [21]

Server is the integrator in this solution where it is the API between the IoT and AR device. The major research done is reflected upon the system when the data logging to a database is eliminated in the process. This shows a valid proof of concept in play showing the unnecessity of database for the above task. The idea is derived from the fog computing concepts and simply by experimentation and use of database in the proposed system. It is found after multiple test and results that the use of database to log the temperature is not required for the integration. Making it slightly fast in response

2.6.1 Holographic Application Design

The main objective of this proposed thesis is to integrate the IoT with the XR experience to visualize the data from the IoT in real-time. The Figure 2.16 elaborates the architectural block of the AR application where it clearly shows that the UI is the major part which helps facilitate the integration.

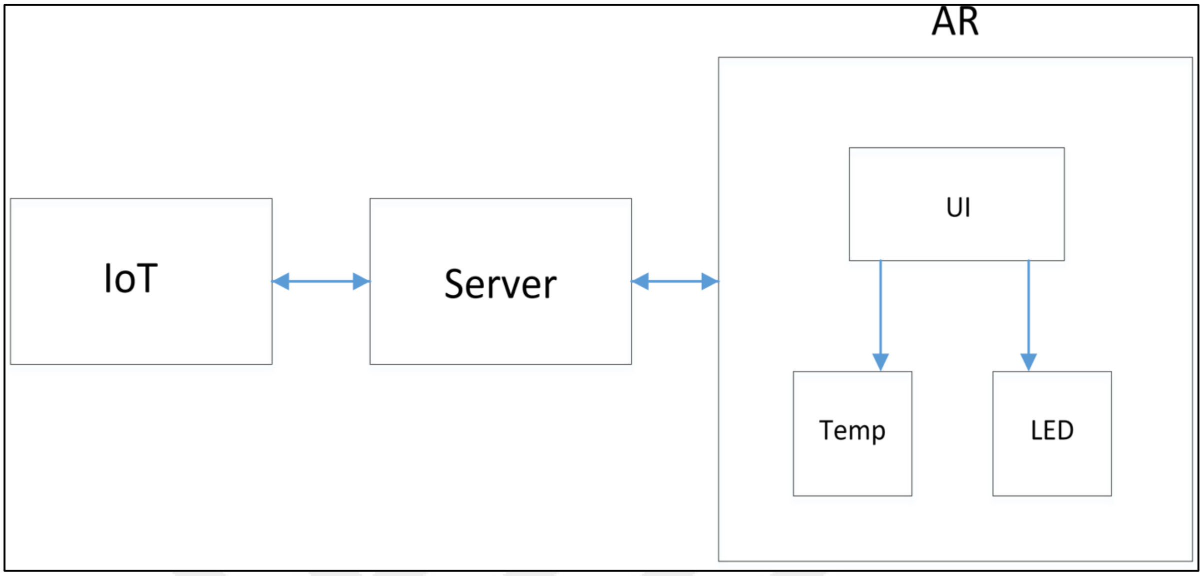

Figure 2.15: AR Block Diagram [22]

The IoT data is visualized through Microsoft HoloLens which enables Augmented Reality to project data in 3D space. The preview of the design from the Unity 3D software can be seen in Figure 2.17.

Figure 2.16: Unity 3D Version 2017.1.3 Preview

Unity 3D is a software which can be executed on different computing platforms, and it is designed for the development of video games [23]. Basically, it is a game engine which provides essential tools for developers, enabling them for rapid deployment.

2.7 DEVELOPMENT TOOLS

It is to help AR users to access and interact with the IoT device. The created AR experience should synchronize with the real-world object. Therefore, we make use of the development tools made for HoloLens by Microsoft. The development tools work as a plug and play when added to the Unity project to create the desired game view. Mixed Reality toolkit from Microsoft is required to setup the initially settings for the Unity project. Which contains pre-made tools such as Holographic buttons, application templates and basic scripts involved in general UI design. The features available from the Mixed Reality tools can be seen in Figure 2.18.

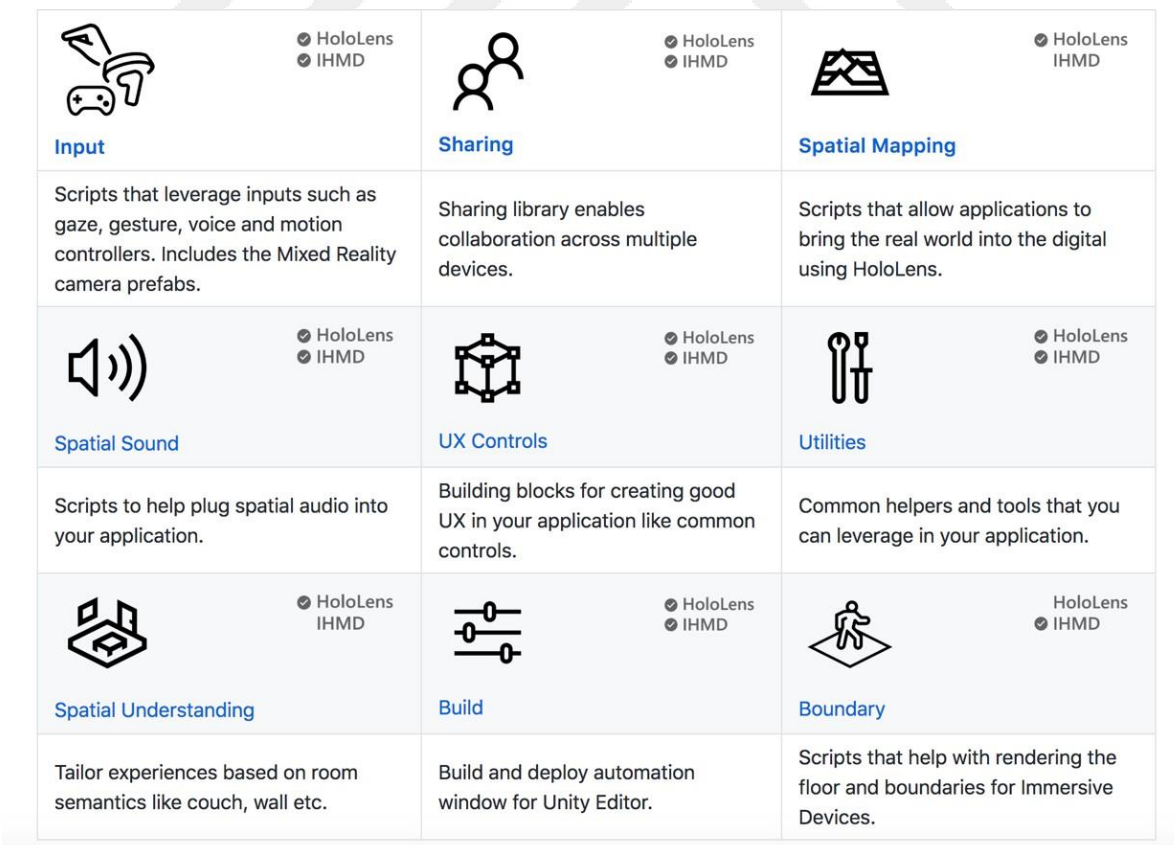

Figure 2.17: Mixed Reality Toolkit [25]

To help overlay the digital object over the IoT device, we make use of Vuforia augmented reality tools. Vuforia tools work in sync with the AR device's in-built visionary sensors to align the digital object with the real-world object [24]. There are mainly two types of object recognition, they are

i. Marker design

ii. Marker less design

In the Marker design, a pre-processed 2D image is loaded into the build files of the holographic application. The Vuforia tools help process a selected image as a marker and when detected through the camera it projects the digital object over the physical image in the real-world. In the Marker less design, the 3D object which is to be recognized is scanned step by step using a Vuforia mobile application. The scanning process of the 3D object is set in bright environment and have to follow precise instructions from the Vuforia scanning guidelines. The scanned file is then uploaded on to the Vuforia dev portal, where the scanned file is processed with Vuforia technology and returned as a Unity Asset which can then be added to the Unity project. The proposed Unity project has been built with no Markers. The use of Vuforia to recognize the device is still widely researched and are not stable [25]. Though the marker less design has slight advantage over the marker less design, the detection of the object is sometimes lost due to lighting conditions and appearance of the object itself. Whereas, the marker design can be selected with high focus points, such as whites, greys, and black, which are major detection points of Vuforia's recognition algorithm.

2.7.1 User Interface Scripts

Every object designed in the Unity using mixed reality tools has pre-built scripts which contains the codes for its operations. For example, if a holographic button is added to the project for on/off, to assign the on/off function to it, one must edit its pre-built scripts to match the required execution. These are the backend scripts which enables or performs the operations assigned to its counterpart UI. In the Figure 2.20, we can see the UI designed to show the information received from the IoT device. The "Temp:" displays the temperature value beside it. The

Holographic button is programed to turn on/off the LED hosted on the IoT device. The scripts which perform the data retrieval from the IoT is explained in detail below.

2.8 SYSTEM ARCHITECTURE

The following presents the complete conceptual model of the IoT, Server, AR integrated together to show the final structure and pattern of the proposed system shown in Figure 2.18

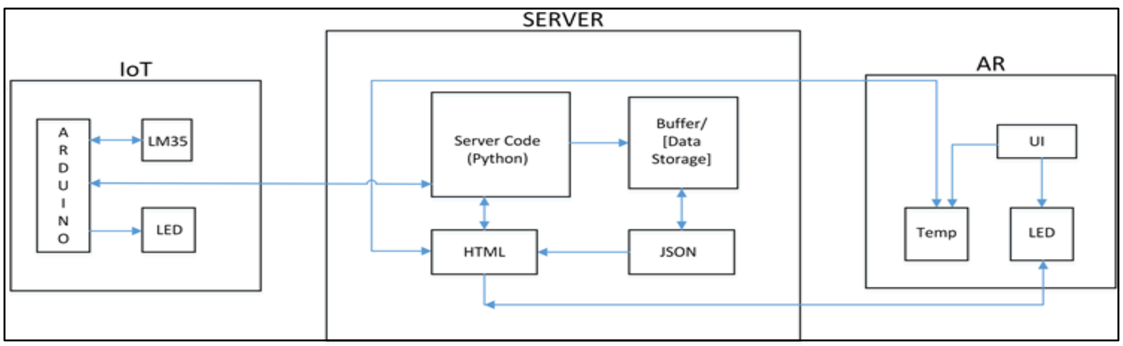

Figure 2.18: Block diagram of the Proposed System Architecture [26]

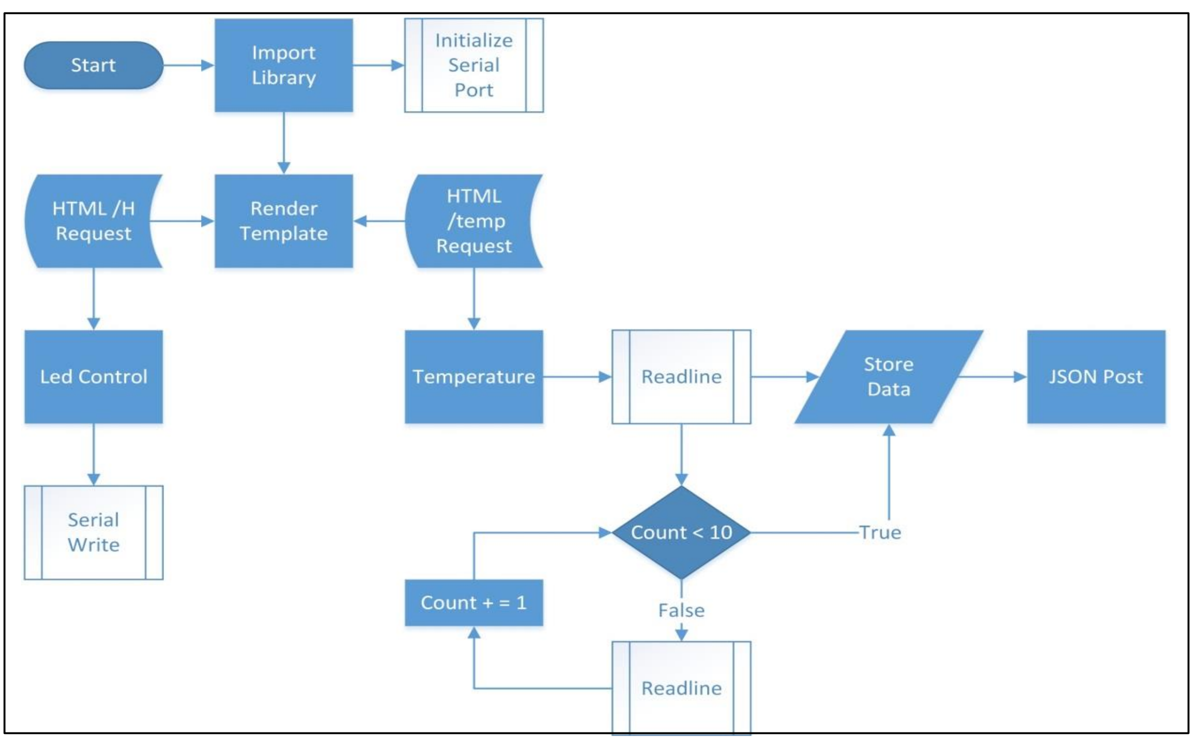

Figure 2.19: Complete flowchart of the Server Script [27]

2.8.1 IoT Device Test

The Arduino script is uploaded to the Arduino Uno and manually tested if they respond with the desired result. The below flowchart shows the functions which need to be processed. The functions which are to be tested is as follows,

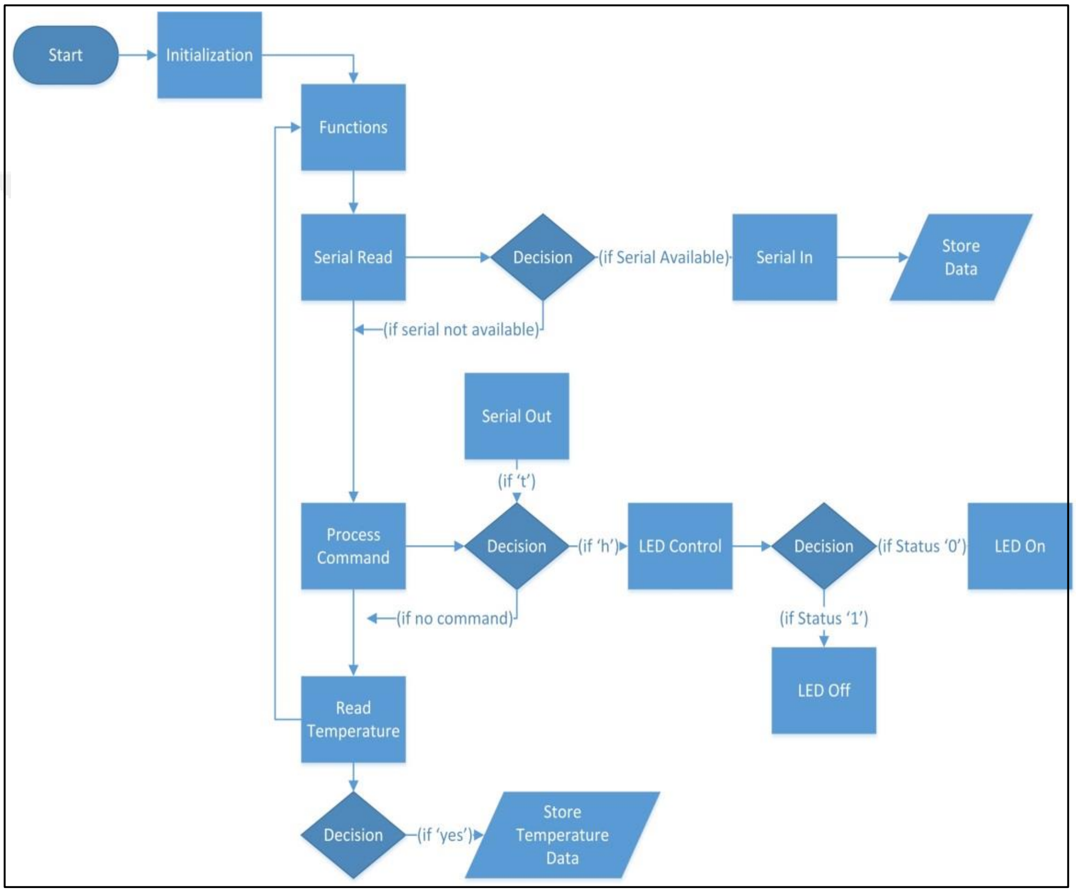

Figure 2.20: Complete flowchart of the Arduino script [28]

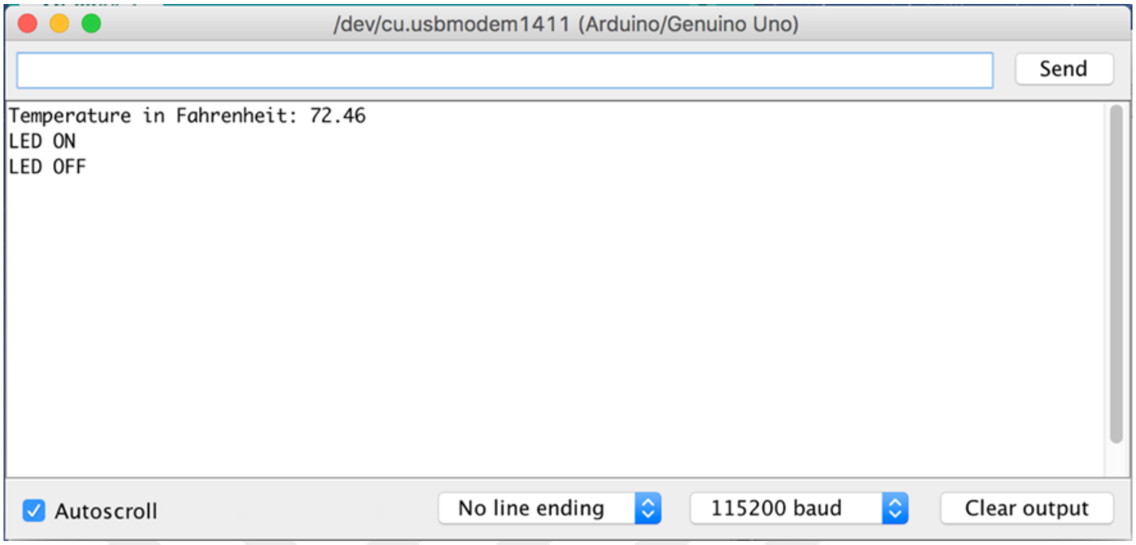

Figure 2.21: Serial Output from the IoT Device

The tests are executed successfully. The embedded program loaded on to the Arduino has performed as expected. The serial output of the Arduino can be seen in Figure 2.22

2.9 SERVER OPERATION TEST

The Flask program in executed in the system through terminal window commands. Initially the server is set to run and host data in the localhost of the computer. Next the server is set to host data onto the edge of the network by running on the computer's local IP. At first the server script is set to run in localhost and observed for any errors and then set to run in local IP of the computer (edge of the network) [29]. The output of both the test can be seen in Figure 2.23 and 2.24 respectively.

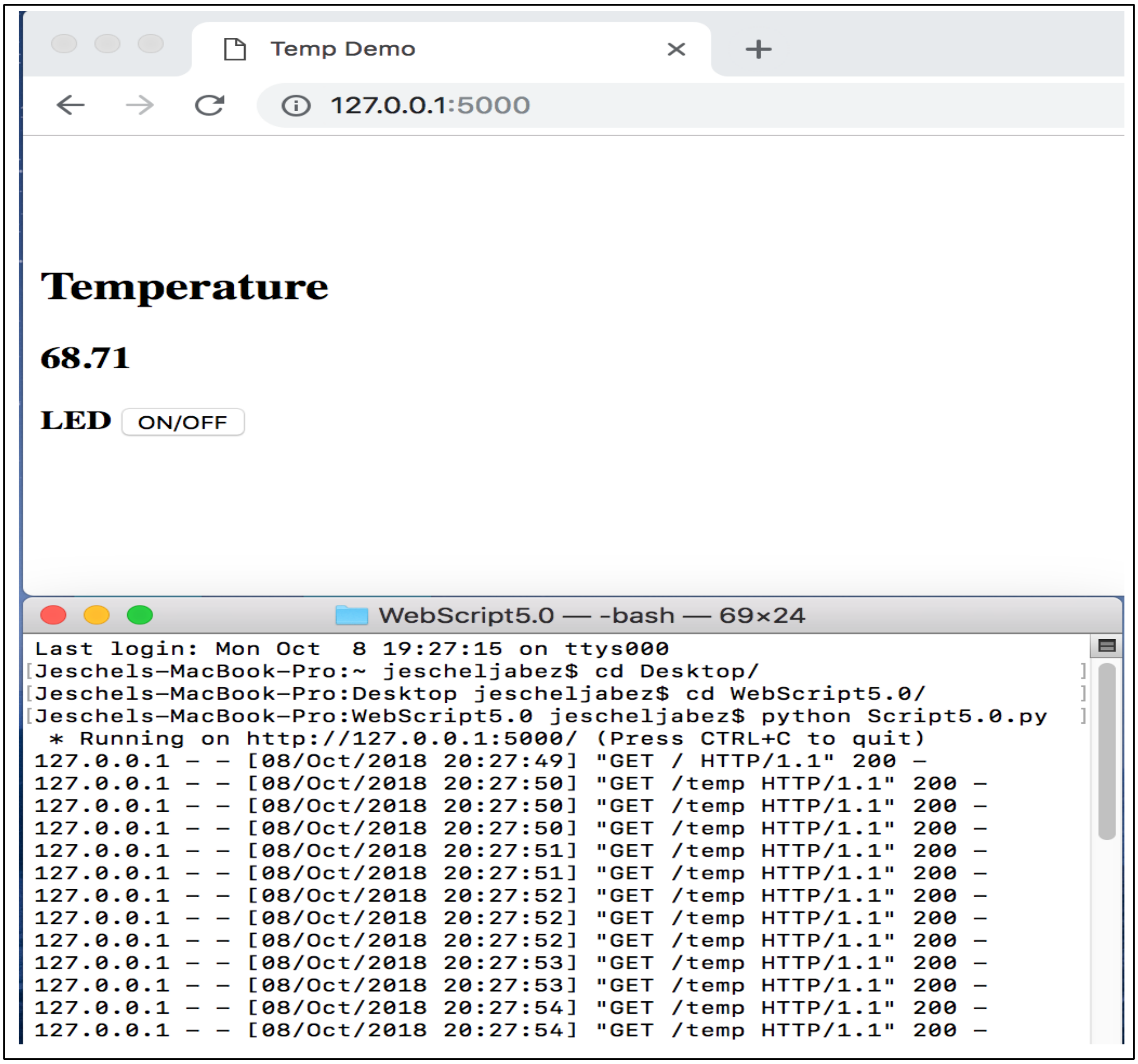

Figure 2.22: Server running in Localhost.

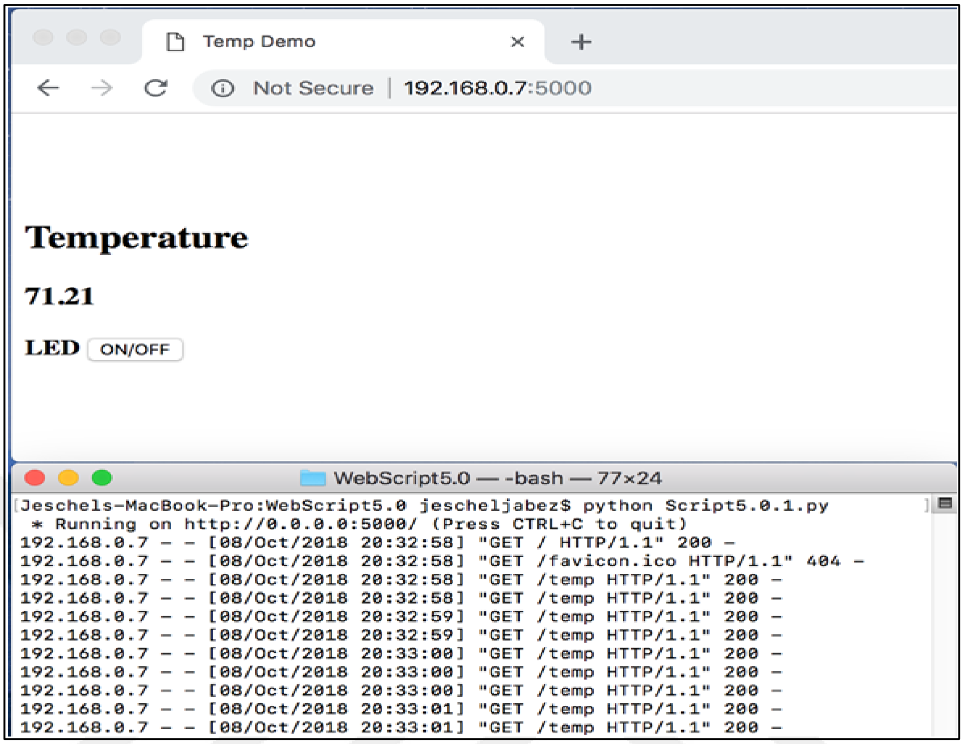

Figure 2.23: Server running in Network Edge.

2.9.1 Application of Machine Learning

The application of a Two-Stage Bayesian Optimization (TSBO) algorithm to the coils and power inductor, using the simulation procedures developed as part of this dissertation, has already been demonstrated [30]. The results of this showed that the coil area can be greatly decreased with only small decrease in efficiency. However, this optimization was done without the calculated efficiency analysis, specifically not considering the source efficiency. In addition, relying on simulators such as ADS to determine what designs have optimal efficiency is less accurate than using analytical methods such as those developed in this dissertation, as was already demonstrated. Also, the optimization was done for

RF near field coupling system with bridge rectifier - while this dissertation has demonstrated a half- wave rectifier is superior for efficiency. The same algorithm can be reapplied to account for the transmit coil inductance (for improved source efficiency) while maximizing k and Q to further push the system eff Cincy higher while decreasing the area of the receiver coil. The coils in this dissertation were designed with mainly two geometric parameters varied - the side length and the line width.

The TSBO algorithm can include several more geometric factors, such as rectangu- lar shapes, the ground plane cutout shape, etc., to find a more optimal design than possible using the ’hand-tuned’ process of this dissertation.

2.9.2 Flexible Hybrid Electronics

Flexible hybrid electronics (HBE) is a rapidly developing research area. Based on the application space for low power, compact devices suitable for near field coupling, FHE is a natural area to apply RF near field coupling [31], [32]. For low power warbles, RF near field coupling can help develop FHE charging systems with better efficiency and coil size than any other WPT method may offer.

The microstrip line loss in this thesis is not negligible. Depending on the design, over 5% loss could be observed just from the contribution of the microstrip line loss. An obvious improvement to the demonstrated system would be to design for high quality RF substrate rather than use FR4. While this dissertation used off the shelf components for the rectifier and regulator chip, Chapter 5 discussed the power inductor design for an integrated, 10 MHz buck converter. Specifically, smaller inductance was needed for this chip compared to the LTC3564, and a fully integrated system on package RF near field coupling system could integrate very high frequency buck converter, synchronous rectifier, and the designed RF coils for a very compact receiver with minimal area [33]. Such a demonstration could take advantage of advanced substrates and multiple layers, the latter of particular interest to further decreasing power inductor area [34].

2.9.3 Figure of Merit for Inductive WPT Systems

One challenge with comparing WPT systems is that different efficiencies are possible at varying system parameters, making comparisons very difficult. For instance, refer back [35] to observe how the systems demonstrated in this dissertation are substant- tially different in operating conditions from any RF near field coupling system previously demonstrated. As such, comparisons are difficult when substantial difference in one sys- tem parameter, such as load power [36], has huge implications on efficiency. It is meant mainly to show that compared to the best inductive cou- pling systems, this work offers a unique combination of high efficiency and small coil area not necessarily an improved system. These RFNFC systems operate at short transmission distance, where higher efficiency is more easily achieved compared to the transmission distances demonstrated in the other compared systems. However, this dissertation work also operates at lower power and with much smaller coil area, making high efficiency more challenging. A figure of merit (FOM) [37] to compare these systems would be of great utility. Varying system metrics, such as transmission distance, receiver coil area, load power, output voltage, etc would be included to robustly compare inductive WPT systems of very differ- ent operating conditions. The resultant FOM should show similar value/quality of work between this dissertation and the best systems compared representing high quality WPT implementations for a variety of application spaces [38]. The development of this FOM could draw from data in published literature and the efficiency relationships drawn from the analysis in this dissertation.

2.9.4 Improved Efficiency Analysis

The demonstrated efficiency analysis has proven useful for guiding the design of RFNFC systems and for capturing many of the peculiarities of RFNFC systems [39]. However, two major improvements would be useful for more consistent, precise analysis. First, understanding the complex interaction of the half wave rectifier with WPT systems, and any resulting changes in the efficiency analysis, needs to be completed. While the introduction of Half as a modeling term is useful for predicting the shift in resonant frequency, how these effects the calculation of the coil and source efficiency specifically is unknown. Second, several of the measurements, specifically the measurement of the PDU input voltage, point towards peculiarities in the chosen PDU (LTC3564) that are not sufficiently captured by the efficiency analysis. Understanding the exact relationship of Vin,P DU and Rin,P DU under various loading conditions is needed for more accurate system efficiency prediction, though the used modeling does provide a general approach that has good accuracy in the majority of cases, with sufficient accuracy for basing coil design on [40].

2.9.5 Multi-power Charging Platform

One area of interest, especially for the IoT, is how to charge devices with varying power requirements. One of the largest problems is providing enough power to charge high power devices, like a laptop, without overloading a lower power device being charged at the same time. RF near field coupling can provide a simple solution for this. Higher power, larger area, higher efficiency coils could be used to charge higher power devices at a frequency such as 13.56 MHz (such as previously demonstrated) [41]. Meanwhile, the charging platform can be configured to concurrently charge low power devices using RF near field coupling. Setup this way, a low power, compact device could be concurrently charged with high efficiency, without fear of being overloaded by the 13.56 MHz higher power source as the RF receiver coil is configured for much, much higher frequency. And because of the very different coil areas, integrating the different frequency charging coils in the same platform would also pose little issue [42].

2.9.6 Key Accomplishments

The main objective for this dissertation was completed: demonstrate that RF near field coupling offers a superior combination of coil area and efficiency for milliwatt-scale load power over other methods of WPT [43], while also demonstrating an integrated solution with embedded inductor and regulated output. In order to accomplish this, several research

contributions were accomplished, listed as follows with some key findings and conclusions in this dissertation for each contribution in what serves as a summary of the preceding:

Novel efficiency modeling approach specific to RF near field coupling that deter- mines not only system efficiency, but distribution of losses in the system. The mod- eling approach was validated with several measurements

2.9.7 Power losses specific to low power, RF system were categorized

i. The subsystem design tradeoffs were explored as a result of the analysis

ii. The analysis determines the input voltage of the PDU, for a controlled power source system, by enforcing power matching at the input of the PDU and the output of the rectifier

iii. Calculated efficiency analysis demonstrates improved accuracy over simulation-based techniques, though error occurs when assuming no matching loss over a wide range of load resistance

iv. System architecture and design process for efficient RF near field coupling system, specifically identifying how the design approach differs from conventional inductive magnetic resonance coupling design

v. System architecture developed for continuous/discontinuous power demand

vi. The design approach considers factors such as the transmit inductance, source efficiency, and complex relationship of load resistance to system efficiency not as significant in other WPT systems

vii. Half-wave rectifiers yield optimal efficiency in a RFNFC, low power system

viii. Demonstration of a reverse power distribution network with measurement validation, as a key component for linking RF-DC conversion with a PDU.

ix. Established target impedance from both the PDU and rectifier perspective Identified phase as an important consideration in reverse PDN design Developed design rules for integrating large storage capacitance in RFNFC sys- tem with the reverse PDN Identification and modeling of dynamic electromagnetic effects in RF near field cou- pling systems to pick the right component values in the circuit, especially the sec- ondary resonant capacitor. Half wave rectifiers interact with the RF coil resonant network differently than bridge rectifiers, leading to a downshift in frequency Developed a process for selecting component values via the aid of simulators such as ADS, rather than relying on conventional equations Demonstration of cheap, simple packaging technology for integrating an embedded inductor with magnetic core into RFNFC receiver.

Completed design and characterization of embedded inductor for an integrated buck converter that developed a revised inductor loss analysis for DCM buck regulator Demonstrated integrated inductor with COTS buck regulator showing improved PDU efficiency compared to similar SMT inductor. Meeting main dissertation objective of achieving very high efficiency with very small coil area in a RFNFC system in a reliable, integrated solution All full systems demonstrated superior efficiency to all other RFNFC systems Demonstrated systems demonstrate improved combination of efficiency and coil area compared to resonant coupling system at any frequency, at the ex- pense of transmission distance Discussion and identification of the application space the developed RFNFC systems are advantageous compared to other low power WPT architectures

2.10 SUMMARY

This dissertation discussed the advantages and limitations of RF near field coupling. This dissertation identified low power applications as a particularly suitable area to apply RF near field coupling. The body of this dissertation focused on the efficiency analysis, de- sign process, integration and modeling challenges, packaging technology integration, and measurement demonstration of RF near field coupling systems with the best combination of system efficiency and coil area ever demonstrated in literature. Along the way, novel design approaches, new analysis techniques and equations, and innovative solutions to integration challenges were developed to meet the dissertation objectives

3. METHODOLOGY

Many potential IoT applications were considered (e.g., greenhouse automation) to be developed during research. After a systematic selection process based on the selection criteria a home automation solution was chosen. A detailed analysis of why this domain was chosen. Home automation might refer to diff erent levels of automation of home tasks. By definition, any tool or machine that automates a home task constitutes a home automation solution. Historically, home automation became popular with the advent of distributed electricity. Daily tasks as dishwashing or drying clothes were automated by appliances that today are common in many households around the world. In the last decades, home automation gained another meaning with the invention of electronic solutions that control virtually any electronic in a house. Lighting, air conditioners, heaters, entertainment systems and doors are common components controlled by home automation solitons. Frequently, these systems are composed by a central control unit with a user interface (e.g. computer, tablet, smartphone, wall control panel), a communication channel (e.g. Bluetooth, LAN, Internet, infrared) and devices to be controlled (e.g. lamp, air conditioning unit, doors, TV, appliances).Automated tasks might be as simple as turning a light on when someone enters the room, controlling the heater based on a target room temperature, locking the main door at a set time and closing the curtains based on the amount of natural light outside.

3.1 LOGIC GATES

The Logic gates system is an EDSL that implements the TOP paradigm. Its host language is the pure and lazy functional programming language Clean. Logic gates uses generic programming to automatically generate code for user-specified first-order data types whilst also allowing for specialization. Given a Logic gates program, the system generates code for both the server and the client. A Logic gates task is a function that transforms a state, reacts to events and returns an observable value. A task is represented by Task and its observable value by board Value a, where a is the type of the task. A board Value contains the current state of the value that a Task is processing. The possible states of a Board Value are shown on Figure 3.1. As seen below, once a value stabilizes, it cannot become unstable.

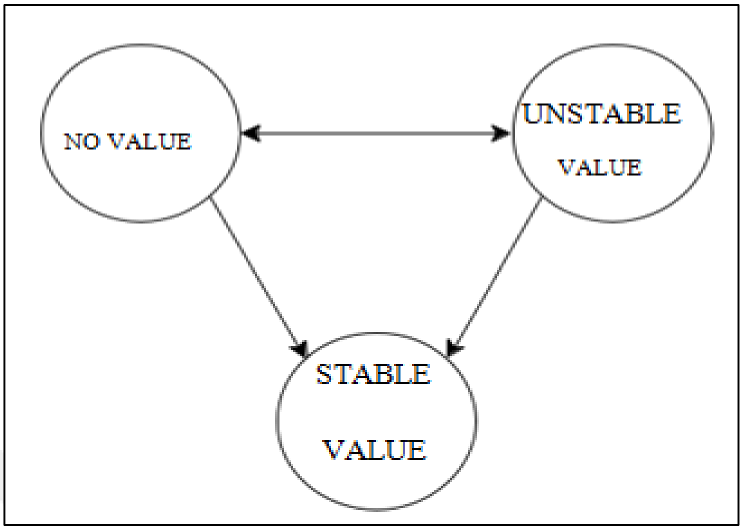

Figure 3.1: Possible states of a Task Value

A Task can be of any type, as long as it provides instances for the type classes in the Logic gates type class collection. These instances can be automatically derived or explicitly declared. Automatic derivation is available for anyfirst-order type as long as it is not abstract. Basic types have instances already defined in the Logic gates library. Explicit declaration allows the user to define custom instances of the Logic gates Task class collection if the default derived instance is not suitable. Moreover, it allows instances for extendable, abstract and the function types.

3.2 COIL DESIGN

3.2.1 Selection of Coil Geometry

To demonstrate an integrated solution, the RF coils for this design are integrated on the same substrate as the rest of the transmitter or receiver circuit (instead of off board coils often seen in WPT demonstrations). Various coil implementations have been previously demonstrated, including wire wound, planar spirals, and planar rings. For RF operation, single turn rings give the best maximum PTE, as shown in Fig. 3.2. The coil geometry for all designs in the dissertation has the parameters shown in Fig. 3.3. In the Fig. 3.4 HFSS simulation comparison, 1, 2, and 3 turn planar inductors on FR4 are compared, with l = 7.5mm, w = 1mm for the 1 turn coil, l = 5mm, w = 0.5mm for the 2-turn coil, and l = 4.5mm, w = 0.25mm for the 3 turn coil. The Q and k are significantly degraded when more than one turn is simulated. Because the parasitic capacitance is much greater in the multiple turn coils (from the interwinding coupling), the self-resonant frequency is much lower. This causes an artificial increase in k as frequency increases as shown in Fig. 3.2.

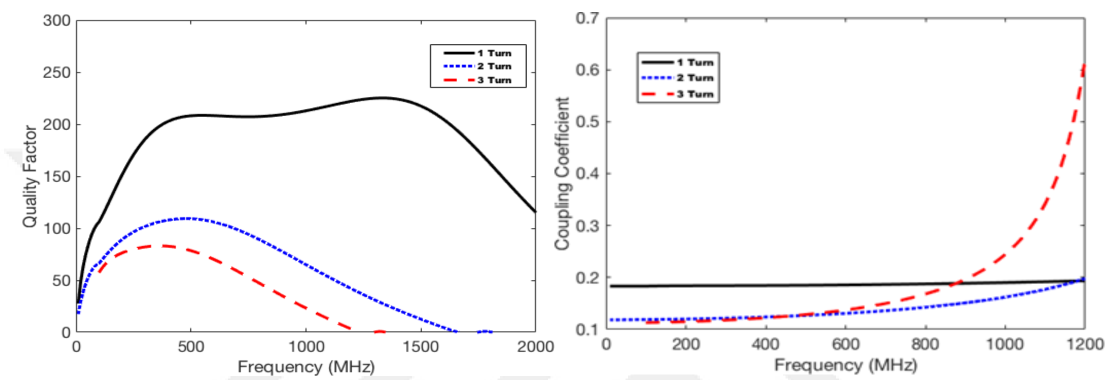

Figure 3.2: Comparison of Q and k for 1, 2, and 3 turn RF coils

To keep coil area small and prevent the inductance from increasing and further lowering the self-resonant frequency, smaller line width must be used when more turns are added. This greatly increases the resistance and decreases the quality factor at RF. Because of this, single turn rings are used for all designed coils. Both the transmit and receive coils are designed with the same geometry. However, area is only a consideration for the receiver coil. An HFSS model for the coils is shown in Fig 3.4 with 1 oz copper on FR4 substrate. One important geometry and modeling note for the coils is a void is added in the ground plane directly beneath the coil (with slightly larger area). This leads to increased inductance, and therefore greater efficiency, for the coils. The simulated magnetic fields, viewed from the side, for the coil2,2 link is shown in Fig. 3.5. HFSS is used for all full wave 3D electromagnetic simulations in this dissertation.

3.2.2 Coil Parameter Modeling

Before introducing the coil design, a couple of notes should be made about the parameters used in the calculated efficiency analysis. First, the HFSS simulated quality factors change greatly with the transmission distance This simulation sweep shows coil4 simulated Q at 1 GHz as a function of transmission distance. One rea- son for this variation is the capacitive coupling between the coils increases as transmission distance decreases, decreasing the self-resonant frequency and artificially increasing the

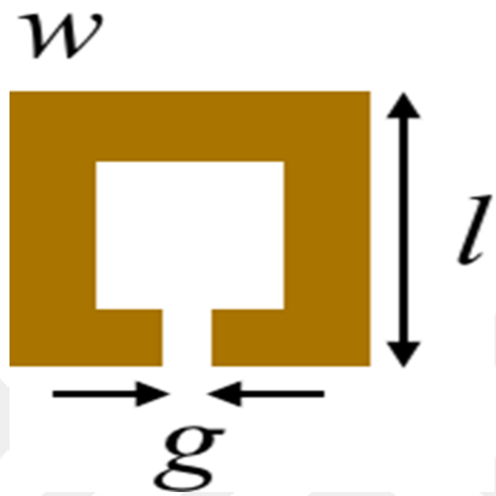

Figure 3.3: Coil geometry for this dissertation, with side length l, line width w, and gap distance g

Resistance of the coil. The reason it is artificial is that the coils are designed to operate at resonance, thus operating closer to the self-resonant frequency does not degrade efficiency as much the HFSS simulation would suggest but rather lessens the extra capacitance needed from the resonant capacitor. While the Q does degrade some in practice as distance shortens, using this simulated value results in large efficiency underestimation when compared to measurement. The most accurate calculated efficiency, compared to measurements, occurs when using the quality factor of the coils when not closely coupled to the other coil. Therefore, the quality factor of each coil is simulated alone, with this value used in all efficiency calculations, with some slight tuning allowed. Second, the microstrip feedline on the transmitter side needs to be modeled to calculate the source efficiency.

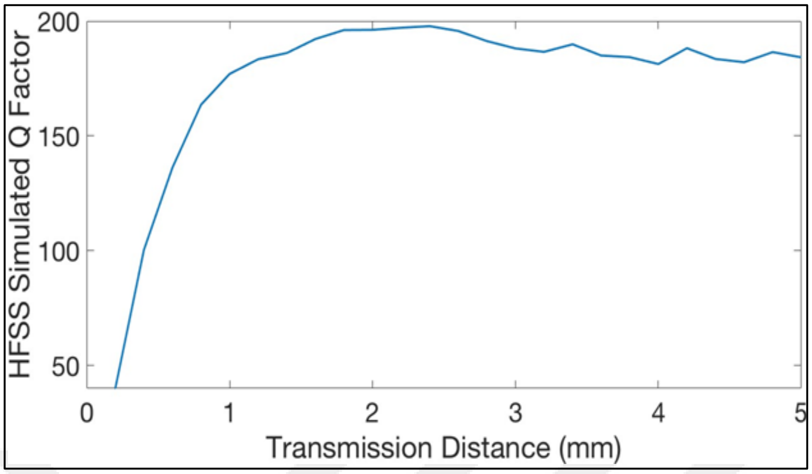

Figure 3.4: Effect of the transmission distance on the simulated Q factor for coil4 geometry

agreement between the calculated and measured efficiency is achieved.

The extra inductance from the microstrip feed line also adds to the reflected impedance to the transmit coil and correspondingly increases the source efficiency. Microstrip line length is approximated and simulated in HFSS for each transmit coil used, and this value added to L1 for use in efficiency calculation.

Application Description

The home automation application developed is called Auto house. It enables and manages the automation of a home (hereafter referred to as smart home) using VLSI. A smart home is composed by rooms that can be added and removed by its user. Each room contain devices (called units in the application) that execute tasks chosen by the user. A smart home is managed via the web control panel. Multiple instances of the control panel can run simultaneously. There, the user has access to the main features of the application:

a. Manage home: add and remove rooms to the smart home.

b. Manage room: add and remove units to a room.

c. Send task: send a new task to one of the available units.

d. Stop task: delete a task that is currently running on a unit.

e. Inspect unit: see which tasks are running on a unit.

3.3 ARCHITECTURE

Auto house is a centralized solution: a server (ideally located in the home) is the central communication hub for both users and devices. User communication is accomplished via the web control panel, which is hosted in the server. Device communication is accomplished via the VLSI circuits and logic gates library and its communication protocol an example architecture of Auto house deployed on a home with three units.

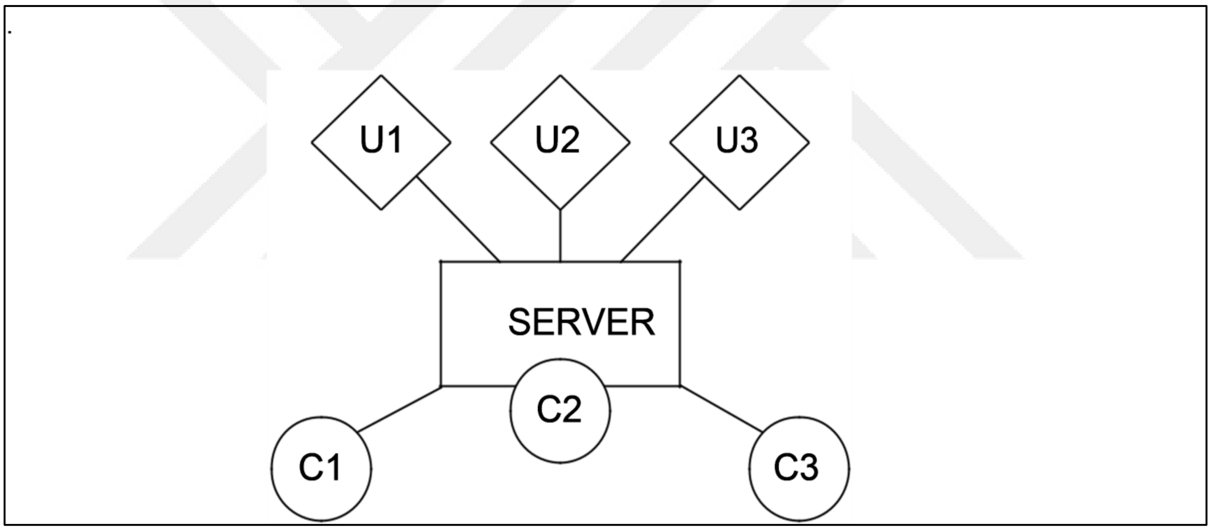

Figure 3.5: Auto house Architecture

layer to ease the use of Auto house and therefore are not represented in its architecture. Units are microcontrollers equipped with peripherals (sensors, actuators) and with a communication interface. The server can be any computer with networking capabilities that is supported by Clean and that has enough resources to run the application. A client is a device accessing the control panel using a web browser. Note that the server itself may be a client since it can access the control panel via a web browser. In addition, if the server is exposed to the Internet, the control panel can be accessed remotely. It is important to point that the development did not regard security. When exposed to the Internet, the application can be accessed by anyone.

3.3.1 Devices

Three platforms are supported by: Arduino, POSIX and Mbed. Arduino was chosen as the main platform for Auto house. Arduino is open-source electronics prototyping platform which has single-board microcontrollers specifications and software to support them. It was chosen because of its open-source nature, popularity, well-established community and open-source code availability. The Arduino platform currently has many boards with diff erent specifications and purposes. The Arduino Uno Rev31 was chosen as the target device because of its popularity, extensibility (using shields2), cost and limited resources. The Arduino Uno is based on the ATmega328P microprocessor, which has only 2 KB of RAM. This memory limitation is a desired characteristic, given that VLSI's suitability for microcontrollers is being tested and that such devices might have extremely limited resources. The Arduino Uno operates at a frequency of 16 MHz, has 6 analog pins, 14 digital pins, 32 KB of flash memory and a built-in LED. Its digital and analog pins are often used to interface with peripherals, including sensors and actuators. Sensors and actuators are extremely relevant to Auto house because they allow the application to interface with the real world without human interference. Examples of sensors are temperature, humidity and movement sensors. Examples of actuators are buttons, switches, LEDs and motors.

3.3.2 Sensors

Auto house relies on the interpreted VLSI and its Arduino client for peripheral support. Therefore, it supports the same sensors the Arduino client does. Digital and analog pins can be controlled explicitly, but VLSI also provides native support for some sensors. Namely, for the DHT22 temperature and humidity sensor, the HCSR04 ultrasonic sensor, digital light sensor, Grove analog light sensor (P) V1.1 and Passive Infrared Sensor (PIR). Native sensor supports al-lows programming in a high level of abstraction, in which sensor values represent their semantics. For example, reading from a temperature sensor returns a Temperature, an ADT that represents temperature in Celsius. Some of these sensors were not previously supported by VLSI.

3.3.3 Actuators

Similarly, to sensors, Auto house relies on the interpreted VLSI and its Arduino client for actuator support. Some actuators (e.g., switches, buttons) can be directly controlled through pins and therefore do not require custom support. Other actuators, though, require custom support to be practically used. The interpreted VLSI provides native support for LCD displays (to print integers only), LEDs and servos. Servos were not previously supported by VLSI.

3.4 APPLICATION ANALYSIS

The selection process that chose Auto house as the research application was guided by the selection criteria. In this section, the Auto house choice is motivated by analyzing the application under those same criteria.

Suitable Auto house is suitable for VLSI circuits. Its components are small devices connected to a server that (if connected to the internet) can be accessed anywhere. In addition, Auto house targets Arduino Uno, a board supported by VLSI circuits.

Non-trivial It is not a trivial application. The complexity of Auto house goes beyond a simple reactive system. It is an application with a user interface where its user can dynamically add devices and send automation tasks to such devices. Its tasks can be simple reactive tasks but can also contain complex logic based on input from diff erent sensors and devices.

Simple Auto house was simple enough to be developed during the research. Thanks to the prototyping nature of logic board, its development could abstract from many technical details and focus on design decisions.

Interesting Home automation is becoming increasingly popular, and its industry is growing consistently3. There is no industry standard, and the market is highly fragmented. Therefore, Autohouse solves an existing, interesting problem.

Significant Autohouse might improve many aspects of its user's life. For example, Autohouse tasks can save time (e.g., automatically opening the garage door when a car moves backwards). They can also save energy (e.g., turn the hallway light off if no one is walking by). They might also improve the user’s wellbeing (e.g., automatically regulate room temperature). In addition, tasks can be simply convenient (e.g., open the curtains when it is bright outside).

Comprehensible The home automation domain is comprehensible to most people. We all live in homes and can relate to most of the tasks Auto house off ers. Therefore, no explanation of the application domain is required to comprehend it.

Robust Auto house is a robust application when it comes to device disconnection. If a unit is disconnected from the server, the system migrates the tasks that were running on that unit to another, suitable unit. The application does not support server fault tolerance. Although the distributed version of Logic gates could have been used, Auto house runs on the single server version instead.

Highly connected Auto house supports simultaneous devices. Although wireless connections are supported (e.g., Bluetooth), Auto house does not enforce them.

Dynamic the Auto house system allows for dynamic addition and removal of devices. In addition, a user might send tasks to devices dynamically. Home automation is a domain that naturally requires dynamicity. It is expected that Auto house users send and remove tasks from devices on a daily basis.

Diverse Auto house supports five diff erent sensors and three actuators. Therefore, in total, eight peripherals are integrated into the application. Although this number is small when com-pared to some commercial applications out there, it is enough to display Auto house’s diversity.

Extensive The application tests VLSIf eatures thoroughly. Regarding the VLSI language, Auto house’s default tasks use all of the language constructs implemented by the VLSI

interpreted view. In addition, it makes use of VLSI functionality to connect to devices and to send tasks to it. Auto house benefits from logic gates abstraction layer over devices and does not handle them directly.

4. RESULTS

This chapter provides a detailed discussion of the results. As mentioned above, various VLSI circuits are proposed in the literature with different power usage strategies, Application Development After the application domain was chosen, development began. Although Autohouse was developed during the research, it was not the research object. It was used merely as a tool to assess VLSI circuits capabilities and thus answer the research question. Therefore, this chapter will not focus on Autohouse itself, but on the limitations of VLSI circuits unearthed during its development. Some limitations of VLSI circuits were overcome and are described other limitations remain and are described in Section how automatic task migration was accomplished without modifying IOT.

4.1 DEVELOPMENT OVERVIEW

Autohouse is an application developed using Clean, logic gates and VLSI. Due to time constraints, it was thoroughly tested only on macOS 10.13 (High Sierra) using Matlab Tool. It was tested on Linux (Ubuntu 16.10) on early stages of development. Autohouse's source code is available at its GitHub repository.

4.1.1 Application Architecture

Autohouse development started just like many logic boards applications: defining ADTs and SDSs. The application has ADTs that model key concepts in the home automation domain: House (representing the smart home), Room and Unit (representing a device). A House is a list of Rooms, and a Room is essentially a list of Units. All the application data lives in one SDS that represents the entire house. Other SDSs (e.g., for rooms and units) are derived from this main SDS using parametric lenses. Rooms and units have unique ids that are used to locate them in the house SDS. The default automation tasks are stored in a list of Program in Autohouse's source code, automation tasks are named programs to avoid name clashing with VLSI and Logic board' Task ADT. The Program record contains all the information inherent to an automation task. The application contains three main tasks running in parallel: manage house, manage units and send new task. The first one lets the user create and edit rooms. The second allows the user to inspect, send tasks to and disconnect units. The last task lets the user pick a VLSI task to send to a device that is compatible with it. Once the logic board foundation was created, the VLSI circuits development could begin.

4.2 USING THE SIMULATOR

As expected, device features were first implemented using simulators. These devices proved to be great for early stages of development. First, multiple simulators can be easily instantiated, which was particularly useful when physical devices were not available yet. Simulators allowed the development of peripheral tasks even before some peripherals were avail-able for testing. Additionally, given that simulators are highly customizable, some features that depended on diﬀ erent device configurations were tested. Such freedom to quickly choose between diﬀ erent device configurations does not exist when dealing with physical devices.

Although the simulator was particularly helpful during early stages of development, it proved to be a great debug tool during later stages as well. Whenever a device behaved unexpectedly, the same environment was reproduced using a simulator. Then, using the debugging UI, the device's state (i.e., memory, tasks, program counter, peripheral values) could be inspected. Using the simulator as a debug tool became standard practice during Autohouse's development.

The coil inductance and resonant capacitance values demonstrated in the previous chapter are in the range of 5-20 nH and 0.2-10 pF. Because these values are smaller than observed for lower frequency systems and the source frequency is at RF, the microstrip lines in the design have a considerable effect on the resonant frequency of the system. HFSS exported S-Parameters of half4,1 was simulated in ADS, shown Fig. 4.1(a), with different lengths of microstrip line between the receive coil and receive resonant capacitor. With just 2 mm of microstrip line added, the resonant frequency is shifted by over 20%. The receive coil has only 6 nH inductance, thus it is more susceptible to this frequency variation. When half is simulated, the effect is less pronounced as the inductance of coil4 in the half4,4 receivers is larger than coil1, though almost 100 MHz shift is still observed.

The influence of microstrip lines is verified through measurement of the half4 sys- tem at 0.5mm transmission distance and 14.5 dBm source and 1000Ω load,The rectifier output voltage was measured with C2 and a 100 pF voltage smoothing capacitor in different locations (shown in Fig. 4.1(b)) for three cases as follows: i) C2 at position 1 and the smoothing capacitor at position 3, ii) C2 at position 2 and the smoothing capacitor at position 3, and iii) C2 at position 1 and the smoothing capacitor at position 2. The circuit was measured with no matching circuit (short Lm and C1, open Cm) and C2= 3.5 pF. Even though the circuit/schematic is the exact same in each case, the amount of microstrip line between various points is not. Accordingly, the rectifier output voltage

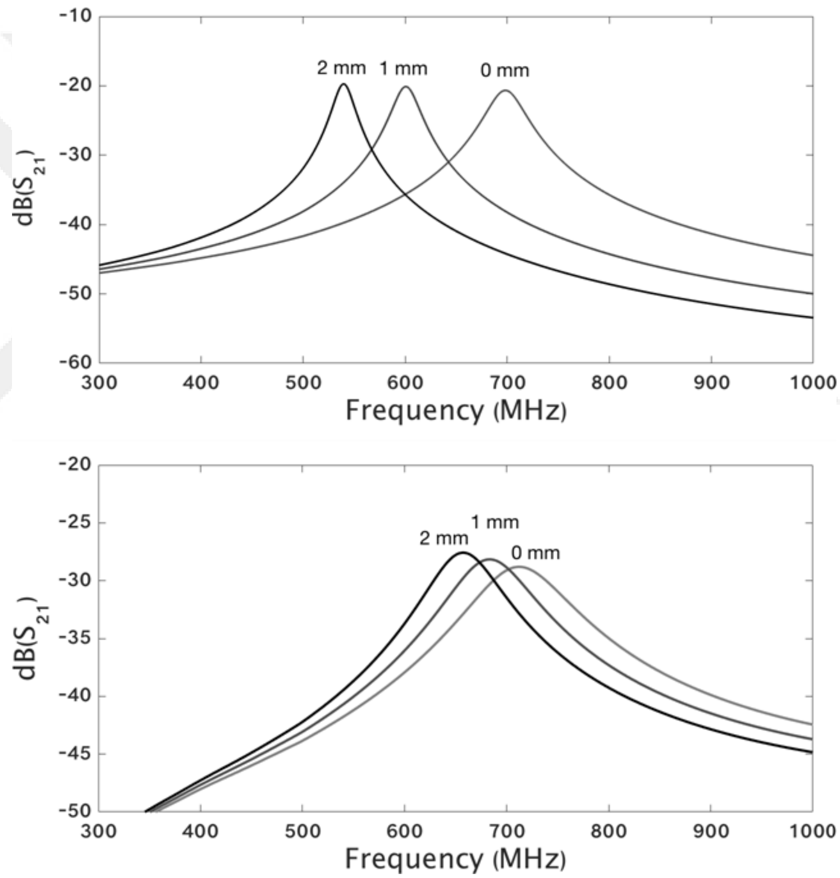

Figure 4.1: Effect of the transmission line length between L2 and C2 on the resonant frequency

frequency response greatly changes in each of the cases.

It should be noted that the efficiency with 1000Ω load peaks around 60% (4.1V output) without matching network, emphasizing the significance of determining C2 first and then selecting the transmit circuit values to further increase efficiency. The circuit was operated in very close proximity (0.5mm), but this highlights the significance of determining C2 primarily, and then selecting the other values to squeeze extra efficiency out of the system. In particular, the value of C2 must be adjusted to account for transmission lines in the receiver for operation at the desired frequency. The influence of microstrip lines and vias are modeled directly in ADS through MLine and via components, to help select the optimal values of C2 and transmit impedance matching values. The microstrip lines in the receiver are modeled in ADS using the equivalent WPT cir- cuit shown in Fig. 4.2, using equivalent circuit values for the coils and equivalent modeled microstrip line geometry. Note that in this simulation, no transmit circuit values are used as they are not needed to determine the receiver resonant frequency. A generic diode component is shown as the rectifier, to be replaced with the appropriate rectifier topology being simulated. The simulated S21 parameter can determine the needed down adjustment of C2 to resonate at a specific frequency. If a bridge rectifier is selected for the design, this is the final adjustment of C2 to account for the detuning of the resonant frequency.

4.2.1 The Dynamic Behavior of Half Wave Rectifiers in RF Near Field Coupling

As discussed in the previous chapter, to achieve high source efficiency, large k is necessary. However, operating the coils with large k has an unusual effect on the half-wave rectifier: it shifts the resonant frequency lower. This is shown by a measurement of half4,4 for two cases with no PDU. First, at 1mm transmission distance with C1 = 5 pF, C2 = 3.5 pF, and no matching network, both a bridge and half-wave rectifier were measured with 14.5 dBm source and 1000Ω load, shown in Fig. 4.2. There is more than 100 MHz difference in source frequency giving the best output voltage despite the same circuit. This is not

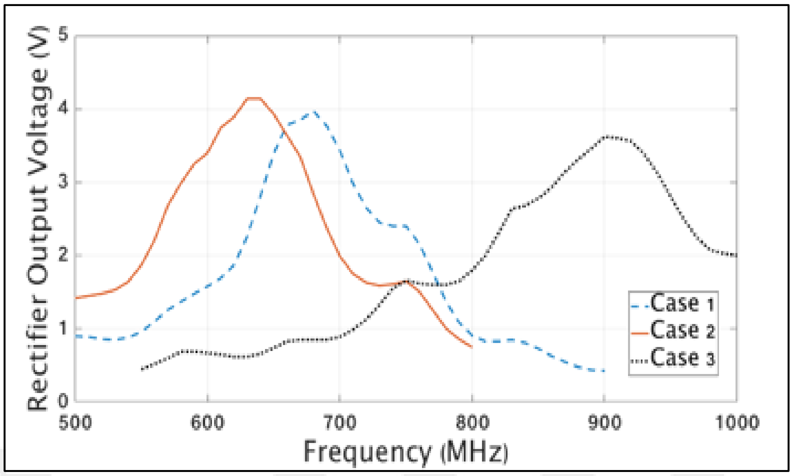

Figure 4.2: Measured rectifier output voltage of half coil rectifier system with different positions of C2 and the smoothing capacitor as classified

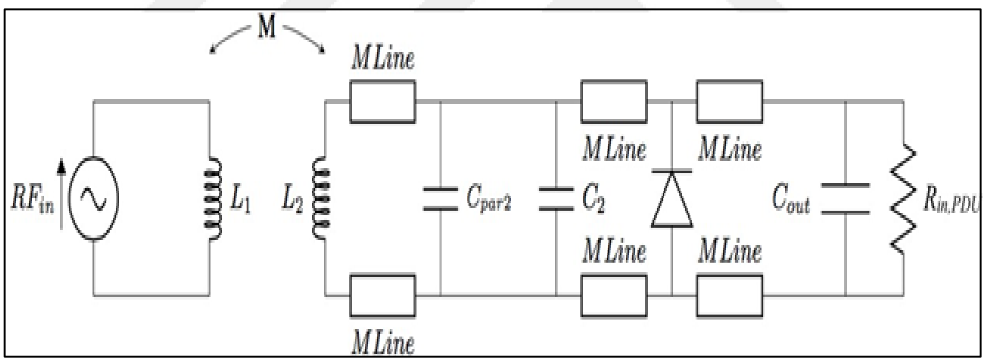

Figure 4.3: VLSI circuit model for determining resonant frequency detuning from microstrip lines

just the influence of the microstrip line, because the current path in the bridge rectifier is more inductive than the half-wave rectifier. Next, the circuit was operated in very close proximity - 0.5mm - with a completely identical comparison. There was no matching network or C1, C2 = 2.4 pF, and the exact same board was used in the comparison. This was accomplished by using a diode where appropriate for a half-wave rectifier, using a shorting resistor at the other diode location for the positive half-cycle for the bridge rectifier, and keeping the locations for the diode in the bridge negative half-cycle open. This way, the half-wave circuit has a completely identical current path to the bridge rectifier in the positive half-cycle, with only an extra shorting resistor added in the current path for the half-wave rectifier. The rectifier output voltage comparison for these two cases is shown in Fig. 4.2. There is over 300 MHz shift - over 25% resonant frequency shift - between the two rectifiers. Even accounting for possible minor variations in the measurement, diode impedance, shorting resistor, etc, there has to be a fundamental difference in how the RF coil link interacts with the two rectifier types to account for such a drastic shift.

When utilizing a half-wave rectifier, the powerful inductance of the loop increments and downshifts the resounding recurrence. A half-wave rectifier is the prevalent decision for high proficiency in a light burden framework. The forward voltage of the diode has a huge factor on the productivity at the voltages inspected in this paper and having just one forward voltage drop in the half-wave rectifier prompts remarkable effectiveness improvement. This varies from standard way of thinking however as the half-wave rectifier in principle just amends a large portion of the sign and squanders the other half, with the end goal that more noteworthy than half effectiveness should be unthinkable. In a resounding WPT framework, this isn't the situation as energy is put away in the receptive parts in the negative half cycle.

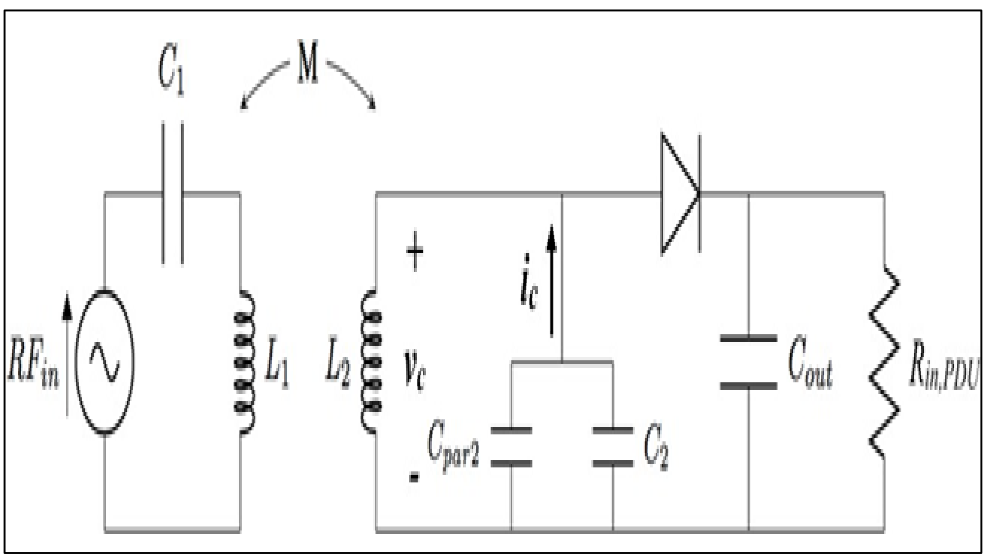

Figure 4.4: Circuit model for determining Lhalf

This is proven in a correlation of the half and extension rectifiers in an ADS transient reproduction of the WPT circuit, with equal inductance and k to the half connection. Here, microstrip lines are excluded from reenactment as they upset the transient waveforms, and the estimation of C1 is changed for comparative yield power across Sdifferent reproductions.

4.3 RECTIFIER TOPOLOGY

Three different rectifiers were considered for RF-DC conversion: half-wave, bridge (full-wave), and Class E. A Class E rectifier is a modified version of the half-wave rectifier. A class E rectifier is a resonant rectifier that seems ideal for low power, RF operation because it uses only a single diode (only one forward voltage loss) and because it uses zero-voltage switching to mitigate switching loss in the diode. Using the equations from for a 1000Ω rectifier load at 868 MHz, the calculated class E filter inductance value is 472 nH and filter capacitance value is 0.07 pF. For RF design, this is very, very difficult to achieve in practice. Moreover, the class E rectifier has its own resonant frequency which needs to be perfectly matched to the coils to have optimal efficiency - adding extra design complexity. A RF near field coupling system designing a custom chip rectifier might be able to use the class E rectifier, but it is unsuitable for the work in this dissertation using components off the shelf. Full systems using both the half-wave rectifier and the bridge are designed to compare the two rectifier topologies. Both the calculated efficiency and the ADS optimized, sim- elated efficiency for the coil4,4 link with both a half-wave and bridge rectifier are shown in Fig. 4.6 for 14.5 dBm, 700 MHz source. The ADS simulation used the same HFSS simulated coil data as the calculated efficiency, with microstrip line components added. The values of Cm, Lm, C1, and C2 were optimized for best efficiency. The correlation of efficiency to load resistance in the two cases is very different. In fact, the ADS simulation suggests the bridge rectifier should have the best efficiency. This cannot possibly be true, as the only major difference with respect to efficiency is that the bridge rectifier has an extra forward voltage loss such that, if both rectifiers are optimally matched, the half wave should have higher efficiency. This represents a key finding of this dissertation - simulation based techniques are less accurate for efficiency analysis than the demonstrated analysis. While this example was chosen specifically to make this point, other ADS comparisons between the bridge and half wave rectifiers have shown both the half wave and bridge as

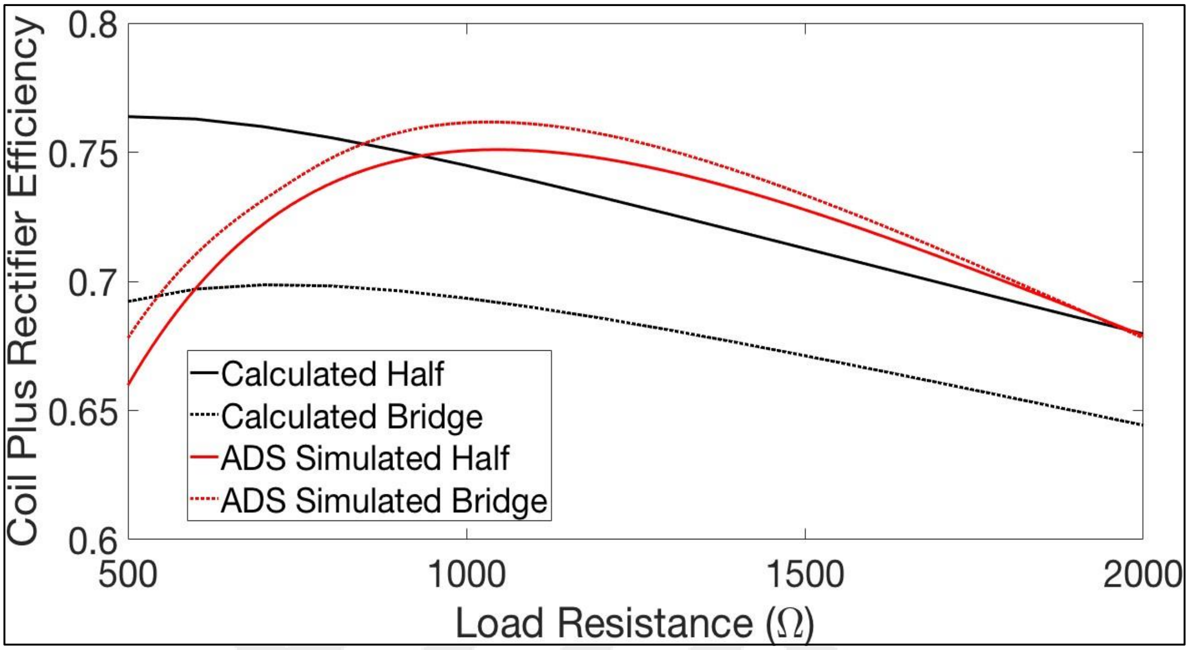

Figure 4.5: Comparison of analysis techniques for half and bridge rectifier with coil4,4

with 14.5 dBm, 700 MHz source, 1mm transmission distance

superior for different designs. Again, this defies reason as one of the two should be supe- rior to the other more generally, as will be shown in measurement results. ADS simulation does have value for RF near field coupling systems, as will be discussed in the next chapter, but not for efficiency analysis. Bridge rectifiers will still be used in full system measure- ments for analysis purposes, as the bridge rectifier is less susceptible to dynamic effects with a RFNFC system for better correlation with modeling, as will also be discussed in the following chapter.

4.4 COIL + RECTIFIER EXPERIMENTAL WORK

Various aspects of the coil design were validated through measurement. Without the PDU, isolating the influence of various parameters is more easily demonstrated. In addition, most of these measurements will not be repeated in Chapter 6 measurements of full systems, as the purpose (validating design considerations) is already completed. The measurements were performed with a HP 8648D RF signal generator and an oscilloscope, with variable resistor rectifier load.

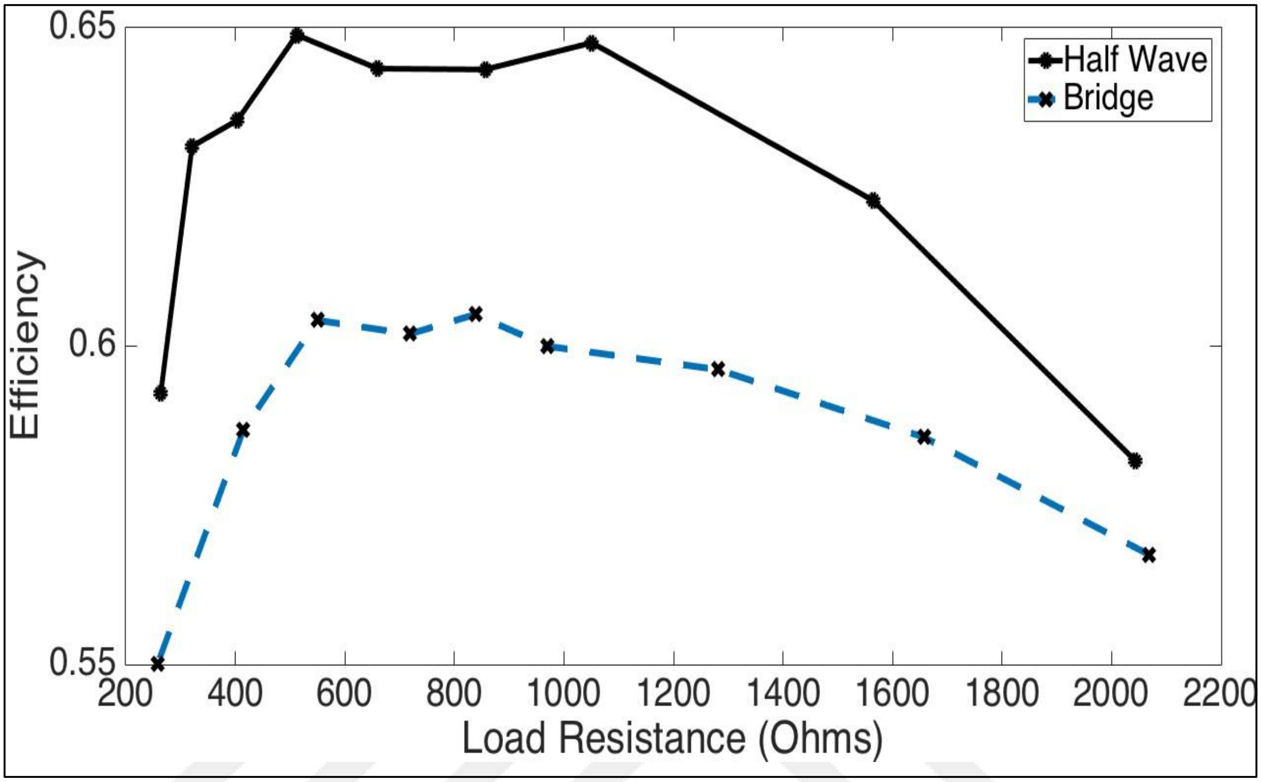

Figure 4.6: Comparison of the bridge and half-wave rectifier with coil2,2

Different Frequencies

A comparison for the rectifier efficiencies of half2,1 and half1,1 at 600 MHz and 1000 MHz is shown in Fig. 4.8. At load resistance less than 1000Ω, the efficiency of all systems is

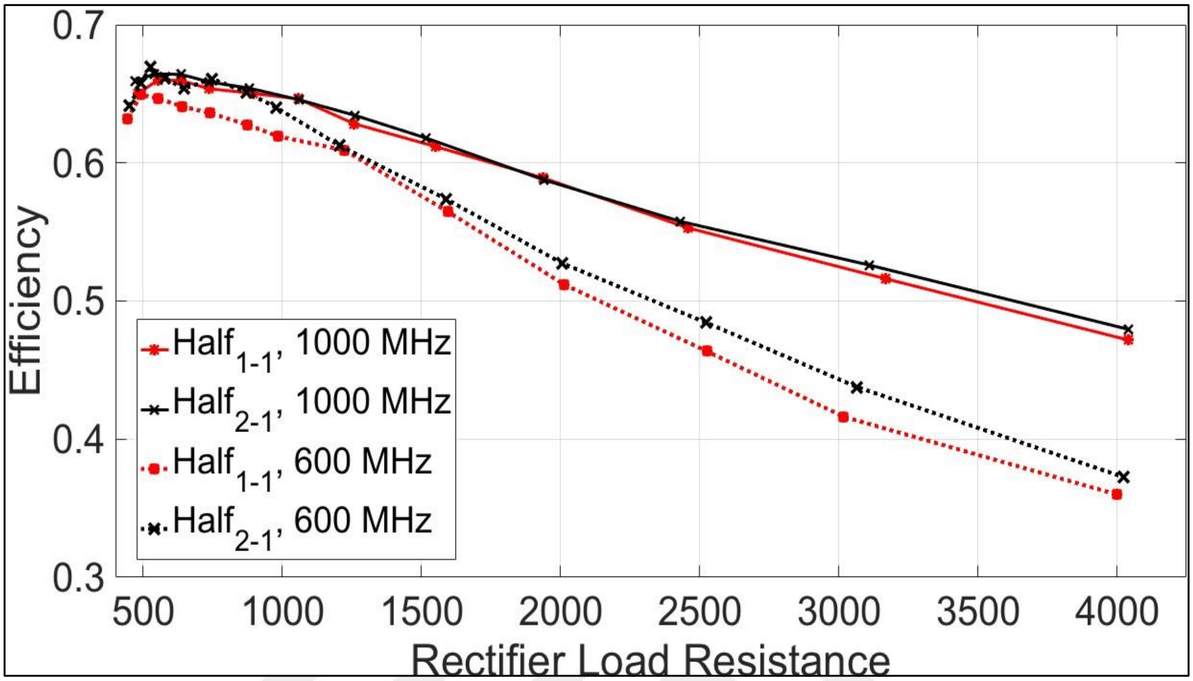

Figure 4.7: Efficiency at 600 MHz and 1000 MHz for half2,1 and half1,1

similar. In general, using coil2 as the transmit has slightly higher efficiency as the quality factor of this coil is much improved over coil1 and has slightly higher L1, while the k is very similar. However, as the load resistance increases, the 600 MHz design efficiency decreases much faster. This is not intuitive, as at higher frequency it would be expected there are greater diode switching and microstrip losses.

However, a larger value of C2 is needed to resonate at 600 MHz (C2 = 7pF was used) than 1000 MHz (C2 = 0.2pF). This leads to a corresponding larger value of α, which leads to a greater decrease in the coil efficiency. If larger load resistance is needed for a system, it is beneficial to increase the frequency of the system to limit the value of C2 necessary for resonance. As discussed, the main design considerations for the coils for RFNFC systems. The modeling approaches and design tradeoffs, using the previous chapter analysis as a guide, were explained in this chapter. In particular, the transmit inductance, the effect of load resistance on the coils, and implications of source efficiency were examined both through calculation and measurements.

5. DISCUSSION

The VLSI circuits was created to bring IoT devices to the logic boards environment. Its language is imperative, and the system is not task centered. Also, tasks cannot be combined as in logic board, limiting expressiveness. Naturally logic boards and VLSI circuits diff er on some aspects, but ideally, the gap between them would be minimal, both in syntax and semantics. Current research has been performed to bridge this gap. A functional, task-centered version of VLSI circuits has been proposed [44]. Parallel and sequential task combinators were introduced, improving VLSI's expressiveness. Due to time constraints, this version of logic boardcould not be used in this research. Assessing the abilities of this new version of boards is proposed as future research. Although Autohouse is not the research focus, it would be interesting to extend the application to push the boundaries of VLSI. Currently, devices have no information about what its peripherals represent. For example, a servo can be used to control curtains or to lock a door [45]. The user might know what which peripheral represents when sending a new task to a device, but the migration algorithm does not. As a consequence, a task might migrate to a compatible device with a peripheral that controls a diff erent object than the intended one. For example, a task that uses a servo to automatically close curtains might migrate to a device that uses its servo to lock a door. Ideally, the application user would attribute tags to its peripherals. The migration algorithm would take that information into consideration when migrating tasks.

6. CONCLUSION

In this work an approach was developed Envisioning the future of Internet of Things (IoT) and Artificial Intelligence (AI) in the world around us: The research reported in this documented tested VLSI circuit's ability to develop real-life IoT applications. The research question was tackled by example: the Autohouse application intended to assess VLSI circuits capabilities. The application is a home automation system that allows users to dynamically manage automation tasks running on devices spread across different rooms. Limitations of VLSI circuits surfaced during the development of Autohouse. Some limitations were overcome by changing the VLSI circuits and Clean Serial libraries. Task variables were added to the language. Device disconnection recognition was implemented, allowing the application to automatically migrate tasks when a device is lost. A new view was added to the EDSL which generates minimum device requirements for a task. This view can be used to filter available devices based on whether they support a given task. Six new peripherals were added to the language and to the Arduino client. Peripheral code was restructured, easing the addition of new peripherals, increasing code maintainability and bringing a better separation of concerns between the language core constructs and peripheral constructs. Finally, the simulator for the interpreted VLSI was modified to support the setting of peripheral values and breakpoints, which improved testing and debugging considerably. Other limitations could not be overcome during this research. SDSs are never removed from devices and live there indefinitely. There is an unwanted communication loop between devices and server whenever a device publishes an SDS. The library does not communicate task acknowledgment. Although these limitations were not overcome, they did not stop the development of Autohouse.

6.1 FUTURE WORK

The Autohouse application was successfully developed using VLSI, but some aspects still have to be analyzed. The tests performed during development were limited to five Arduino boards running a maximum of three simultaneous tasks for a maximum duration of one hour. It is unknown how VLSI performs running more than five simultaneous devices. Given that the number of devices in IoT applications can escalate quickly, it is important to assess how VLSI escalates in regard to the number of connected devices. Similarly, given that IoT applications often run continuously, it would be important to perform tests with tasks running for longer periods of time. Another characteristic of IoT solutions is lower power consumption Although, no power consumption analysis of VLSI applications was performed. Assessing VLSI's power consumption, scalability and behavior over long periods is suggested as future research. Ideally, the limitations of VLSI Circuits should be eliminated. First, SDSs and tasks should be sent in a bundle. Thus, they would be removed altogether, eliminating dangling SDSs. Second, VLSI would avoid sending update messages to the exact same device which triggered the SDS update. Lastly, the VLSI circuits library would provide callbacks to signal task acknowledgment. Solving these problems is also proposed as future research. The Raspberry Pi1 is a compact ARM computer often used in IoT projects both as a device and as a server. Since it is capable of running a logic board core, it should be able to run an VLSI server. Also, since the Pi hardware is Linux compatible, it might host an VLSI POSIX client. Testing with the Raspberry Pi is suggested as future research. The VLSI EDSL was created to bring IoT devices to the logic board environment. Its language is imperative, and the system is not task centered. Also, tasks cannot be combined as in logic boards limiting VLSI's expressiveness. Naturally logic board and VLSI diff er on some aspects, but ideally, the gap between them would be minimal, both in syntax and semantics. Current research has been performed to bridge this gap. A functional, task-centred version of VLSI circuits has been proposed. Parallel and sequential task combinators were introduced, improving VLSI circuits expressiveness. Due to time constraints, this version of VLSI circuits could not be used in this research. Assessing the abilities of this new version of VLSI circuits is proposed as future research.

REFERENCES

[1] P. Ray. “A survey on Internet of Things architectures”. In: Journal of King Saud University- Computer and Information Sciences (2016).

[2] L. Sanchez et al. “SmartSantander: IoT experimentation

[3] D. Miorandi, S. Sicari, F. D. Pellegrini and I. Chlamtac. “Internet of things: Vision, applications and research challenges”. In: Ad Hoc Networks 10.7 (2012), pp. 1497–1516.

[4] P. Ray. “A survey on Internet of Things architectures”. In: Journal of King Saud University - Computer and Information Sciences (2016).

[5] Baid A, Mathur S, Seskar I, Raychaudhuri D. Spectrum MRI: Towards Diagnosis of Multi-Radio Interference in the Unlicensed Band. 2011.

[6] J. Fu, Y. Liu, H. Chao, B. K. Bhargava, and Z. Zhang, “Secure Data Storage and Searching for Industrial IoT by Integrating Fog Computing and Cloud Computing,” IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4519–4528, Oct. 2018.

[7] H. S. Raju and S. Shenoy, “Real-time remote monitoring and operation of industrial devices using IoT and cloud,” in 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016, pp. 324–329.

[8] https://www.researchgate.net/figure/Block-Diagram-of-IoT-Based-Smart-Energy-Meter-Reading-and-Monitoring-System_fig1_333262502

[9] A. Nayyar and V. Puri, “A review of Arduino board’s, Lilypad’s amp; Arduino shields,” in 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 1485–1492.

[10] https://www.researchgate.net/figure/Arduino-integrated-development-environment-IDE_fig7_329042229

[11] A. Nayyar and V. Puri, “A review of Arduino board’s, Lilypad’s amp; Arduino shields,” in 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 1485–1492.

[12] https://www.instructables.com/Arduino-Blinking-LED/

[13] https://www.electronica-pt.com/circuitos/en/pics/led-flasher-pic-circuit

[14] https://medium.com/@n.vitebsky1327/using-an-arduino-to-make-an-led-light-blink-can-be-done-in-less-than-5-minutes-dc7635e18833

[15] “LM35 ±0.5°C Temperature Sensor with Analog Output and 30V Capability | TI.com.” [Online]. Available: http://www.ti.com/product/LM35.

[16] https://circuitsify.com/internet-of-things/

[17] https://www.researchgate.net/figure/Flowchart-of-Arduino-Module_fig6_324044452

[18] “Client/Server Computing.” [Online]. Available: http://www.scis.nova.edu/~lloydjam/week08.html.

[19] C. Matt, “Fog Computing,” Bus Inf Syst Eng, vol. 60, no. 4, pp. 351–355, Aug. 2018.

[20] https://www.researchgate.net/figure/Fog-computing-is-an-extension-of-the-cloud-but-closer-to-end-devices_fig1_324280213

[21] https://www.researchgate.net/figure/Personal-server-block-diagram-Physically-the-designs-are-intended-to-be-as-small-and_fig1_4162856

[22] https://www.chegg.com/homework-help/questions-and-answers/1-following-block-diagram-circles-adders-multipliers--create-signal-flow-diagram-b-find-ov-q34788344

[23] E. Kucera, O. Haffner, and R. Leskovský, “Interactive and virtual/mixed reality applications for mechatronics education developed in unity engine,” in 2018 Cybernetics Informatics (K I), 2018, pp. 1–5.

[24] “Vuforia Developer Portal |.” [Online]. Available: https://developer.vuforia.com/.

[25] MixedRealityToolkit-Unity uses code from the base MixedRealityToolkit repository and makes it easier to consume in Unity.: Microsoft/MixedRealityToolkit-Unity. Microsoft, 2018.

[26] https://www.researchgate.net/figure/Block-diagram-of-the-proposed-system_fig1_334193786

[27] https://docs.devexpress.com/XtraReports/401687/web-reporting/general-information/application-diagnostics

[28] H. G. C. Ferreira, E. D. Canedo, and R. T. de Sousa, "IoT architecture to enable intercommunication through REST API and UPnP using IP, ZigBee and arduino," in 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2013, pp. 53–60.

[29] https://www.electronicsforu.com/technology-trends/learn-electronics/vlsi-developments-ic-fabrication

[30] J. van Diggelen, W. Post, M. Rakhorst, R. Plasmeijer and W. van Staal. "Using Process-Oriented Interfaces for Solving the Automation Paradox in Highly Automated Navy Vessels". In: Active Media Technology. Ed. by D. Sl¸ezak, G. Schaefer, S. T. Vuong and Y.-S.

[31] https://dl.acm.org/doi/10.5555/832285.835520

[32] https://seekingalpha.com/article/4361298-intels-7nm-delays-are-amds-gains

[33] "Internet of things - the complete online guide to the iot." [Online]. Available: https://www.i-scoop.eu/internet-of-things-guide/

[34] F. Richter, "The predicted wearables boom is all about the wrist," Statista, April, vol. 8, 2015.

[35] G. Dolmans, "Ban and pn in health," in BODYNETS2012, 7th Int. Conf. on Body Area Netw, 2012.

[36] P. F. Butzen and R. P. Ribas, "Leakage current in sub-micrometer cmos gates," Universidade Federal do Rio Grande do Sul, pp. 1–28, 2006.

[37] A. Morgenshtein, "Short-circuit power reduction by using high-threshold transistors," Journal of Low Power Electronics and Applications, vol. 2, no. 1, pp. 69–78, 2012. doi: 10.3390/jlpea2010069. [Online]. Available: http://www.mdpi.com/2079-9268/2/1/69

[38] S. Saha and B. Ravindran, "An experimental evaluation of real-time dvfs scheduling algorithms," in Proceedings of the 5th Annual International Systems and Storage Conference, ser. SYSTOR '12. New York, NY, USA: ACM, 2012. doi:10.1145/2367589.2367604. ISBN 978-1-4503-1448-0 pp. 7:1–7:12. [Online]. Available: http://doi.acm.org/10.1145/2367589.2367604

[39] M. Konijnenburg, S. Stanzione, L. Yan, D. W. Jee, J. Pettine, R. van Wegberg, H. Kim, C. van Liempd, R. Fish, J. Schluessler, H. de Groot, C. van Hoof, R. F. Yazicioglu, and N. van Helleputte, "28.4 a battery-powered efficient multi-sensor acquisition system with simultaneous ecg, bio-z, gsr, and ppg," in 2016 IEEE International Solid-State Circuits Conference (ISSCC), Jan 2016. doi: 10.1109/ISSCC.2016.7418116 pp. 480–481.

[40] H. Vestburg, "Ceo to shareholders: 50 billion connections 2020," Ericsson, April, vol. 13, 2010.

[41] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," CISCO white paper, vol. 1, no. 2011, pp. 1–11, 2011.

[42] S. Grady, "Powering wearable technology and internet of everything devices," Cymbet Corporation, www. cymbed. Com, 2014.

[43] P. A. Gargini, "How to successfully overcome inflection points, or long live moore's law," Computing in Science Engineering, vol. 19, no. 2, pp. 51–62, Mar 2017. doi: 10.1109/MCSE.2017.32

[44] Y. L. Zheng, X. R. Ding, C. C. Y. Poon, B. P. L. Lo, H. Zhang, X. L. Zhou, G. Z. Yang, N. Zhao, and Y. T. Zhang, “Unobtrusive sensing and wearable devices for health informatics,” IEEE Transactions on Biomedical Engineering, vol. 61, no. 5, pp. 1538–1554, May 2014. doi: 10.1109/TBME.2014.2309951

[45] S. Seneviratne, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, andA. Seneviratne, “A survey of wearable devices and challenges,” IEEE Communications Surveys Tutorials, vol. PP, no. 99, pp. 1–1, 2017. doi: 10.1109/COMST.2017.2731979