

SONA COLLEGE OF TECHNOLOGY

Learning is a Celebration!

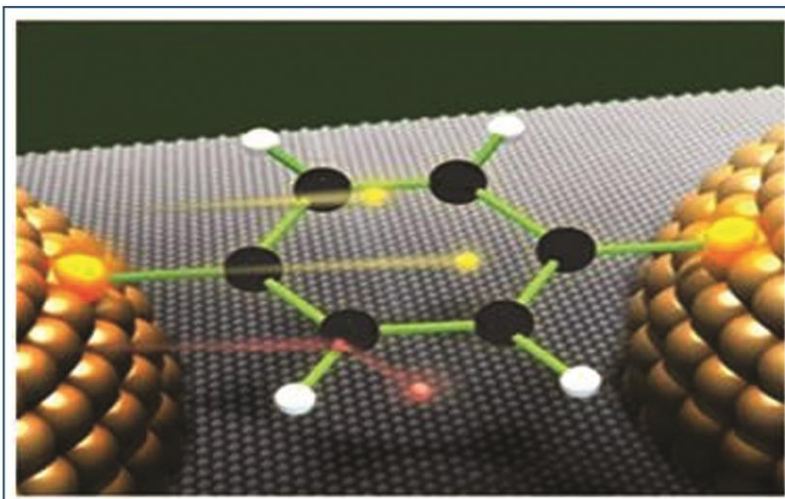
| An Autonomous Institution |

**Department of Electronics and
Communication Engineering**



MICROELECTRONICS TO NANOELECTRONICS

2022-23



Junction Main Road
Suramangalam (PO)
Salem-636 005. TN. India
www.sonatech.ac.in

Editorial Head

Dr.R.S.Sabeenian,
Professor &Head, Dept of ECE,
Head R&D Sona SIPRO



Editorial Members

Dr.K.R.Kavitha
Professor



Mr.J.Harirajkumar
Associate
Professor



Ms.M.Susaritha
Assistant
Professor



Magazine
co-ordinator
Dr.K.Manju
Assistant
Professor



PREFACE

Microelectronics relates to the study and manufacture (or microfabrication) of very small electronic designs and components. These devices are typically made from semiconductor materials. Many components of normal electronic design are available in a microelectronic equivalent.

These include transistors, capacitors, inductors, resistors, diodes and insulators and conductors can all be found in microelectronic devices. Unique wiring techniques such as wire bonding are also often used in microelectronics because of the unusually small size of the components, leads and pads. This technique requires specialized equipment and is expensive.

Nano electronics refer to the use of nanotechnology in electronic components. The term covers a diverse set of devices and materials, with the common characteristic that they are so small that inter-atomic interactions and quantum mechanical properties need to be studied extensively. Some of these candidates include hybrid molecular/semiconductor electronics, one-dimensional nano tubes/nano wires (e.g., Silicon nano wires or Carbon nano tubes) or advanced molecular electronics. Recent silicon CMOS technology generations, such as the 22-nanometer node, are already within this regime.

ASSISTIVE VEHICLE PARKING SLOT

O.MAHESH BABU, P M.VASANTH

ABSTRACT

The objective of this project is to design a assistive vehicle parking slot. In this project slot availability is going to display. This project helps the drivers of the vehicles to park with minimum wastage of time and also, the accurate information of the availability of space to park. This project monitors the slot availability by using the IR sensor. IR sensor is used to detect the object's. This project used Arduino UNO as the Microcontroller where it process the input from the sensor's. If IR sensor detects an object , then the system gives the output as slot is not available by using LCD display and if slots are not available then servo motor won't work then it would be considered as slots are not available and display on LCD.

1.INTRODUCTION

In the present scenario around us excess vehicles and the ineffectiveness to manage them in the correct order is very difficult. As the population increases day by day the rate of utilization also increases and coping up with the numbers becomes a task.

An omnipresent problem around the world is finding a parking space to park the vehicles. This task looks simple on side roads and interior lanes but the actual problem arises when parking in malls, multistorey parking structures, IT hubs and parking facilities where several hundreds of cars are parking and it becomes arduous to find a spot. The general approach to finding a parking space is to go around and drive aimlessly until a free space is found. Finding a parking space could be the easiest task or could be the most tedious one when it involves wide acres of distributed space across one level or multiple levels. The time and fuel are consumed unnecessarily because the destination is unknown. The easiest way of approach is to provide a destination specific driving with in the parking structure. An Assistive parking system

gives a visual output indicates an available parking space rather than driving aimlessly. Smart Parking is used to sense the existing parking spaces. Based on the parking controller space occupancy, the status (engaged/empty) is displayed on the LCD Display. In real time, the atmosphere has sensors and devices embedded into parking spaces, communicating data on the occupancy status and the automobile drivers can look for parking availability using their mobile or any system that is attached to the automobile. Hence the driver would know where there is an existing spot to park his automobile in less time, reducing the energy consumption and air contamination.

Studies says that traffic generated by cars searching for vacancies in Parking Spaces is up to 50% of the total traffic. Now this is a stern issue to look after, and Smart Parking System is one of the finest available solutions to at least reduce the traffic congestion caused due to the above problem. This application gives information about the occupancy status of the spaces in the parking lot equipped with sensors that sense the occurrence of automobiles.

Traffic congestion caused by vehicle is an alarming problem at a global scale and it has been growing exponentially. Car parking problem is a major contributor, still a major problem with increasing vehicle size in the luxurious segment and confined parking spaces in urban cities. Searching for a parking space is a routine activity for many people in cities around the world. This search burns about one million barrels of the world's oil every day. As the global population continues to urbanize, without a well-planned, convenience-driven retreat from the car these problems will worsen. Smart Parking systems typically obtains information about available parking spaces in a particular geographic area and process is real-time to place vehicles at available positions. It involves using low-cost sensors, real-time data collection, and mobile-phone-enabled automated payment systems that allow people to reserve parking in advance or very accurately predict where they will likely find a spot.

The system, smart parking reduces car emissions in urban centers by reducing the need for people to needlessly circle city blocks searching for parking. It also

permits cities to carefully manage their parking supply Smart parking helps one of the biggest problems on driving in urban areas finding empty parking spaces and controlling illegal parking. It support to the government traffic police and implement the system in the government lands at some cities corners where the traffic is causing high in rate

1.1 Objectives

In the past, someone will operate the gates like opening and closing but with this project the traditional way and going to automatically operating the gates with the help of servo motors and IR sensor will detect the object's in front of those sensor's and it will sense and displaying on LCD. To display the slot availability status. To update the slot availability status time to time and also going to control the gates automatically based on the slot availability. The gates are controlling by using the servo motor that will operate with the help of IR sensor. The slot availability is going to display with some short names. Here in this project doing a parallel parking, it will comfortable to take out off the vehicle easily and parking is easy. The main objective of this project is to overcome the parking system, which currently facing with parking spaces.

2.EXISTING METHOD

The methods which were already existed and ongoing such that they are different kind of parkings are available:

- 1. ON STREET PARKING**
- 2. OFF STREET PARKING**
- 3. RIGHT ANGLE PARKING**

2.1 ON STREET PARKING:

On street parking means the vehicles are parked on the sides of the street itself. This will be usually controlled by government agencies itself. Common types of on-street parking are as listed below. As per IRC the standard dimensions of a car is taken as 5× 2.5 m and that for a truck is 3.75× 7.5 m. There is no automation.



On street parking

The figure shows the on street parking which is one type of parking already existed.

2.2 OFF STREET PARKING:

Off street parking means vehicles are parked off the street itself. This will be usually controlled by commercial agencies itself, There is no automation.

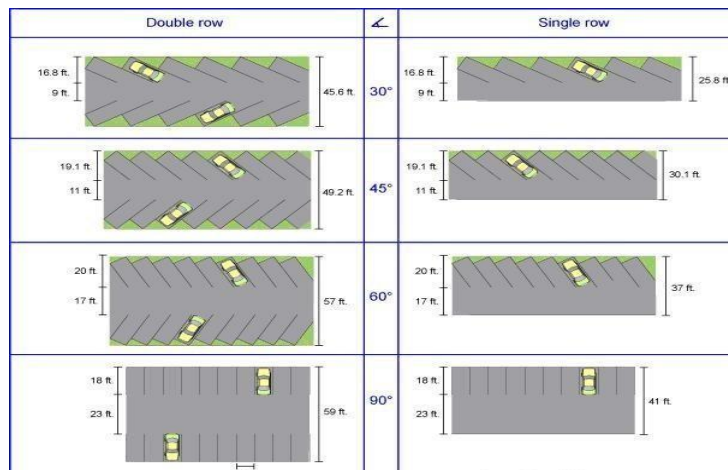


Off street parking

The figure shows the off street parking which is one type of parking system already existed, doing manually.

2.3 RIGHT ANGLE PARKING:

In right angle parking or 90degree parking, the vehicles are parked perpendicular to the direction of the road. Although it consumes maximum width kerb-ed length required is very little. In this type of parking, the vehicles need complex maneuvering and this may cause severe accidents. This arrangement causes obstruction to the road track particularly if the road width is less. However, it can accommodate maximum number of vehicles for a given kerb-ed length.



Right angle parking

The figure shows the Right angle parking which already existed, but its a lengthy space consuming parking system.

In order to accommodate the large volume of vehicles, small cities and towns must develop their infrastructure. One solution may be a multi-level car parking system to maximize car parking capacity by utilizing vertical space, rather than expand horizontally. With land in metros and ‘a’ grade cities becoming scarce and dearer, and plots getting smaller, conventional parking is proving infeasible.

2.4 Silent features:

- Dynamic peak hour management to configure the system in line with the demand pattern.
- Intelligent buffering system offering zero wait time for the public.
- Flexible, scalable and modular design to accommodate wide range of layout and capacities and offers seamless capacity addition.

- Simple design by the way of electrical push pulls mechanism with direct drives.
 - Use of electrical drives to optimize power consumption.
- Intelligent emergency management system via. manual override option

3. PROPOSED METHOD

3.1 ARDUINO UNO:

Arduino Uno is a microcontroller board based on the ATmega328P. Figure has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller, simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst-case scenario replace the chip for a few dollars and start over again.



Arduino Uno

The figure 4.1 shows a block diagram for Assistive vehicle parking slot. This system helps to detect the object and display on LCD. This project used Arduino UNO as the Microcontroller, it processes the input from the sensor's.

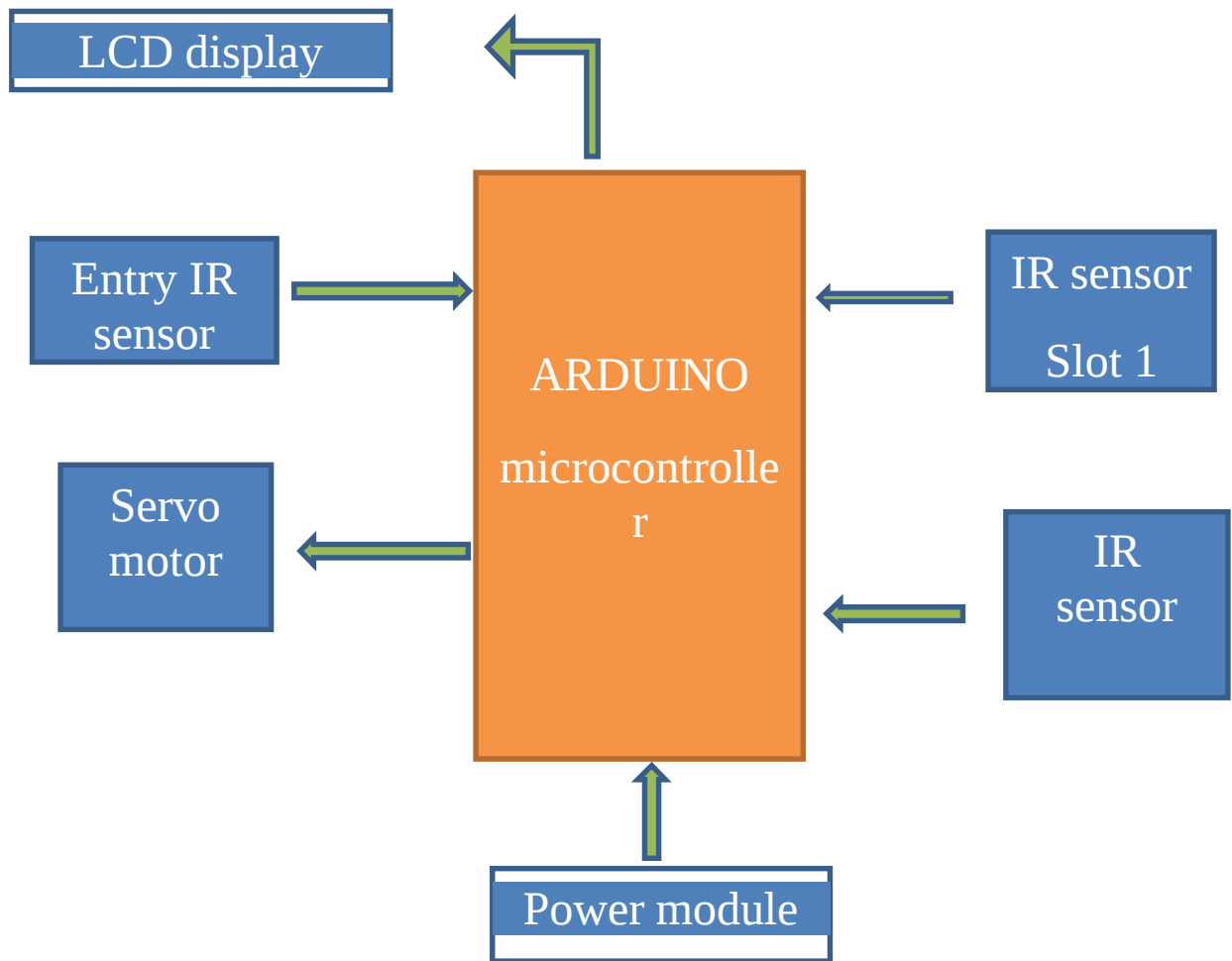


Figure 4.1 Proposed block diagram

If IR sensor detects an object, then the system gives the input as slot is not available by using a LCD display and if all the slots are not available then servo motor won't work so would be considered it as slots are not available.

“UNO” means one in Italian and chosen to mark the release of Arduino Software (IDE)1.0. The Uno board and version1.0 of Arduino Software (IDE) were the reference versions of Arduino, evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform for an extensive list of current, past or outdated boards see the Arduino index of boards.

Power

The Arduino Uno board powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board become unstable. If using more than 12V, the voltage regulator overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

Vin - The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). The supply voltage through this pin or if supplying voltage via the power jack, access it through this pin. 5V this pin outputs a regulated 5V from the regulator on the board. The board is supplied with power either from the DC power jack (7-12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. don't advise it. 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND - Ground pins.

IOREF - This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

Memory - The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output - See the mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical. Each of the 14 digital pins on the Uno used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions: **Serial** - 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) UART, serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB to-TTL Serial chip.

External Interrupts - 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function.

There are a couple of other pins on the board:

AREF Reference voltage for the analog inputs. Used with `analogReference()`.

Reset Bring this line LOW to reset the microcontroller. Typically used to add a

reset button to shields which block the one on the board.

Communication

Arduino Uno has a number of facilities for communicating with a computer, another Arduino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus, see the documentation for details. For SPI communication, use the SPI library.

Automatic (Software) Reset

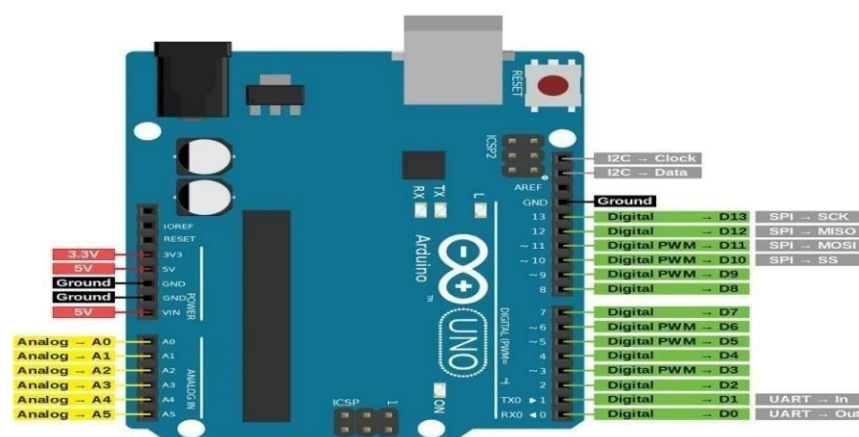
Rather than requiring a physical press of the reset button before an upload, the Arduino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines: (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e., anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labelled "RESET-EN". To disable the auto-reset by connecting a 110-ohm resistor from 5V to the reset line.

Revisions

Revision 3 of the board has the following new features 1.0 pinout added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes. Stronger RESET circuit Atmega 16U2 replace the 8U2.



Arduino UNO Pin Configuration

3.2 IR SENSOR:

The figure shows an infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measure only infrared radiation, rather than emitting it that is called a passive IR sensor.

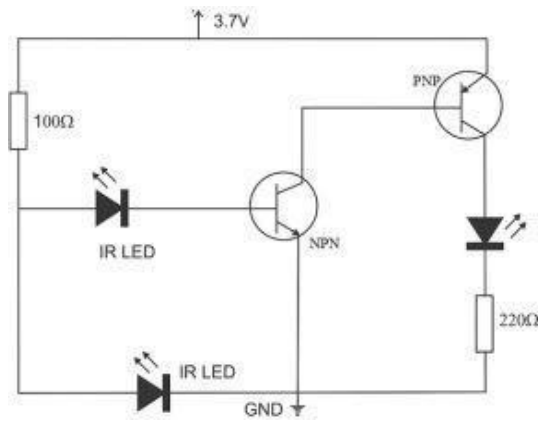


IR Sensor.

Characteristics

- 3.2.1 Low power consumption
- 3.2.2 Noise immunity is strong
- 3.2.3 Detects motion when the light is present or absent
- 3.2.4 These sensors are not affected by rust
- 3.2.5 They do not need to get in touch with objects for detection.
- 3.2.6 No data leakage because of the directionality infrared radiation of ray.

The figure is a circuit of sensor. Once the infrared LED is detected, then the reflected light from the thing will activate a small current that will supply throughout the IR LED detector. This will activate the NPN transistor & the PNP, therefore the LED will switch ON. This circuit is applicable for making different projects like automatic lamps to activate once a person approaches close to the light.



Sensor Circuit Diagram

Types of infrared rays:

Infrared sensors are classified into two types like active IR sensor and passive IR sensor.

Active IR sensor:

This active infrared sensor includes both the transmitter as well as the receiver. In most of the applications, the light-emitting diode is used as a source. LED is used as a non-imaging infrared sensor whereas the laser diode is used as an imaging infrared sensor.

These sensors work through energy radiation, received & detected through radiation. Further, it can be processed by using the signal processor to fetch the necessary information. The best examples of this active infrared sensor are reflectance and break beam sensor.

Passive IR sensor:

The passive infrared sensor includes detectors only but they don't include a transmitter. These sensors use an object like a transmitter or IR source. This object emits energy and detects through infrared receivers. After that, a signal processor is used to understand the signal to obtain the required information.

The best examples of this sensor are pyroelectric detector, bolometer, thermocouple-thermopile, etc. These sensors are classified into two types like thermal IR sensor and

quantum IR sensor. The thermal IR sensor doesn't depend on wavelength. The energy source used by these sensors is heated. Thermal detectors are slow with their response and detection time. The quantum IR sensor depends on the wavelength and these sensors include high response and detection time. These sensors need regular cooling for specific measurements.

3.3 SERVO MOTOR:

A **servo motor** is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. The figure 4.7 shown the servo motor, If we want to rotate an object at some specific angles or distance, then it is used a servo motor. It is just made up of a simple motor which runs through a servo mechanism.



Servo motor

The figure shown is Servo motor consists of three parts Controlled device, Output sensor and Feedback system. It is a closed-loop system where it uses a positive feedback system to control motion and the final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

Here reference input signal is compared to the reference output signal and the third signal is produced by the feedback system. And this third signal acts as an input signal to the control the device. This signal is present as long as the feedback signal is generated or there is a difference between the reference input signal and reference

output signal. So the main task of servomechanism is to maintain the output of a system at the desired value at presence of noises.

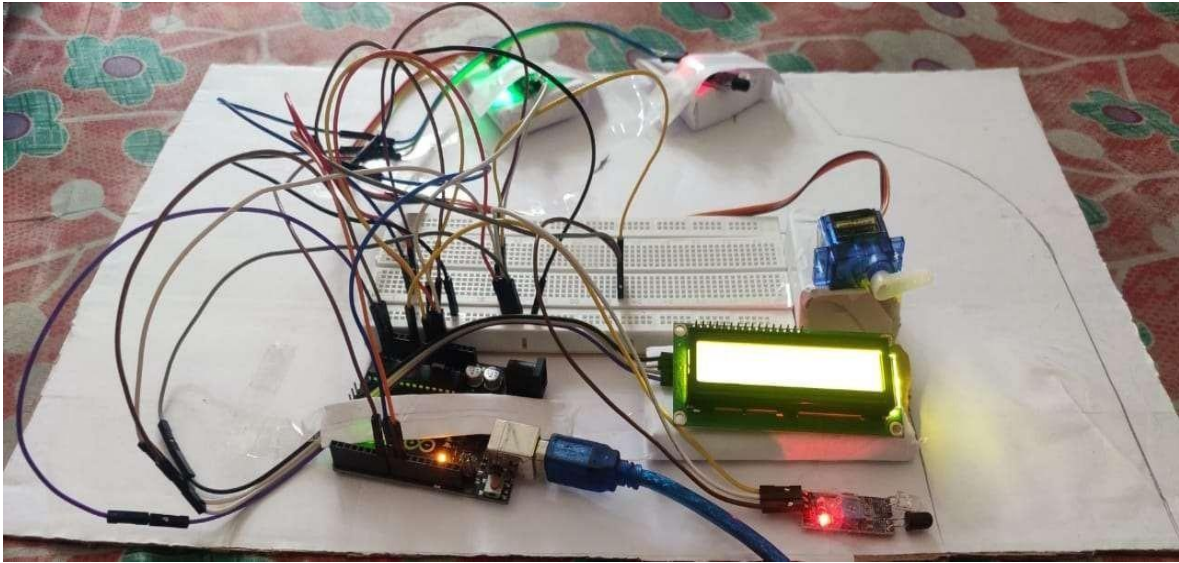
Working:

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly, and a controlling circuit. First of all, use gear assembly to reduce RPM and to increase torque of the motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now the difference between these two signals, one comes from the potentiometer and another comes from other sources, will be processed in a feedback mechanism and output will be provided in terms of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with the potentiometer and as the motor rotates so the potentiometer and it will generate a signal. So as the potentiometer's angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

4. RESULTS AND DISCUSSION

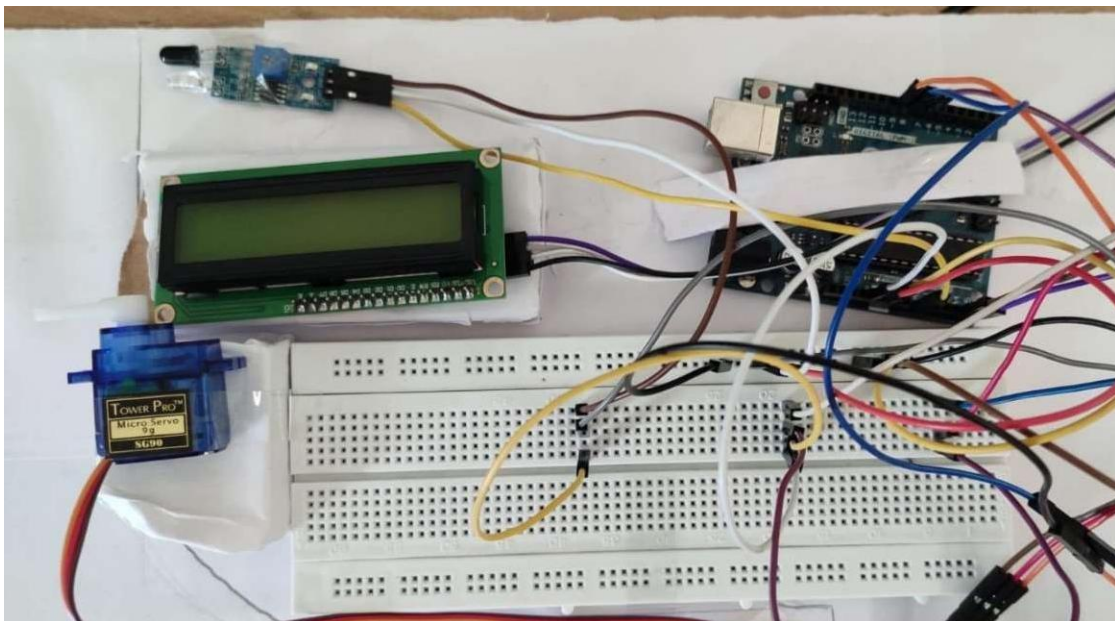
The output of the proposed system is as follows:

The figure shows the Assistive vehicle parking slot is designed by using following components such as IR sensor, Arduino UNO, LCD display, Servo motor which are connected in the breadboard. The below figure shows the prototype of all the components connected with in the breadboard In which the LCD display module is connected and it will display the status of the system, whether the slot is available or not with the help of IR sensor.



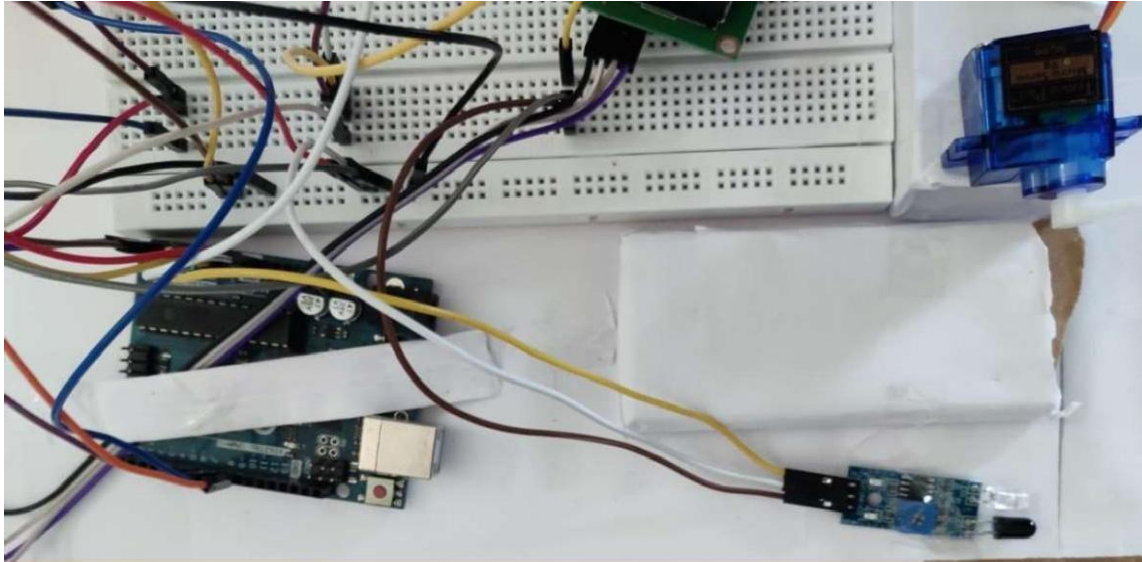
Assistive vehicle parking slot- Prototype

The figure shows the LCD display which will display on the LCD screen about the availability of slots. As we given a slot names as A and B there in the lcd display it will shown like ,if A is slot is available it will be like A is AV otherwise A is NAV(not available) similarly B also slot also display inLCD.



LCD display prototype

The figure shows the Servo motor prototype and if the slots are not available then servo motor won't run.



Servo motor prototype

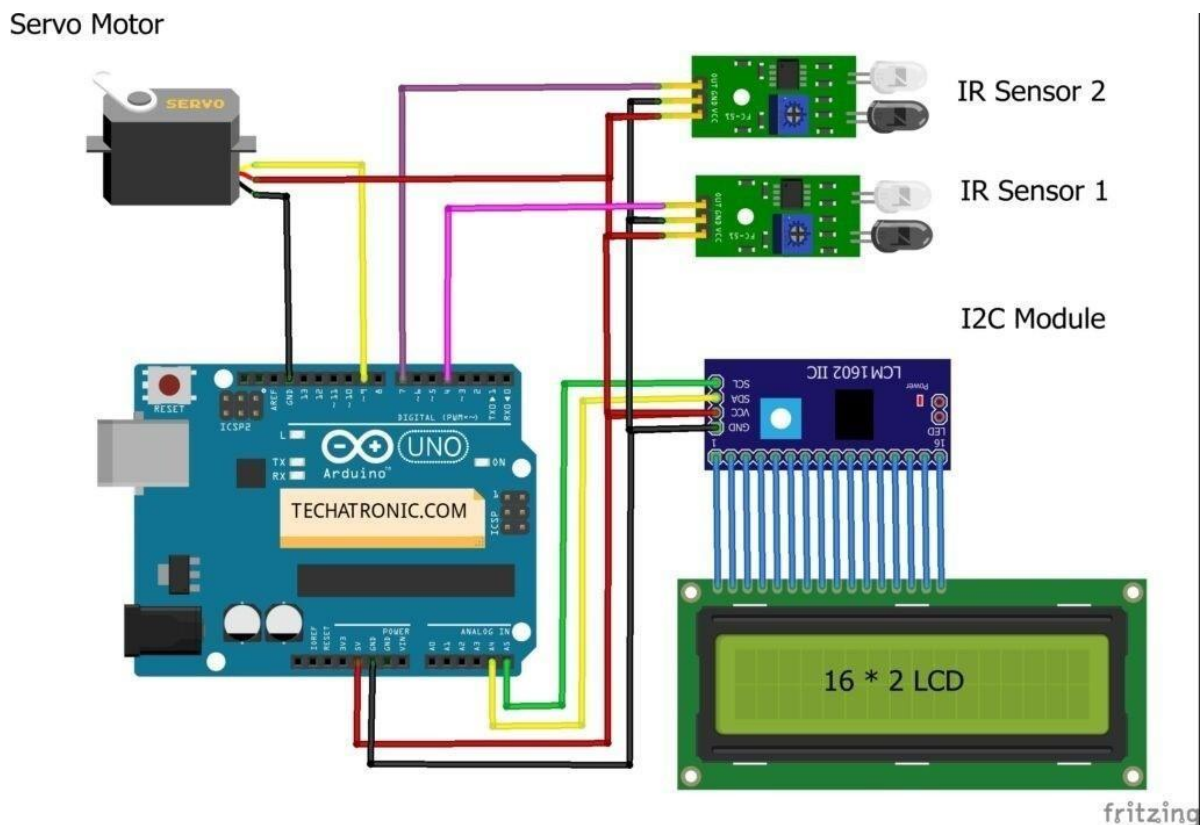
Final prototype:

The figure shows the final prototype of Assistive vehicle parking slot. This project used Arduino UNO as the Microcontroller where it process the input from the sensor's. If IR sensor detects an object , then the system gives the output as slot is not available by using LCD display and if slots are not available then servo motor won't work then it would be considered as slots are not available and display on LCD.

5.CONCLUSION

The proposed system demonstrates the Assistive vehicle parking slot. This design provides an effective solution to the common problem discussed. The Assistive car parking system was designed and tested which provided accurate results when the object was detected.The vehicle in the parking space was instantaneous based on no

vehicle and vehicle detection. If the space is empty then the servo motor will open the gate and then allot that particular space to that Vehicle. If there is no vehicle is detected then Servo motor won't open the gate. It can be concluded that with correct connection of some simple electrical components, it is possible to create an automatic smart car parking system, thus decreasing aimless driving, fuel and time, as well as making the process of parking considerably simpler. Explained how the assistive parking system may help improve the life style of many car owners and also save their time while needing a parking around a certain area such as mall, school, or any way around.



Final prototype

SONA SMART CAFETERIA

SWETHA. N, SHREENIVAS. K. M

ABSTRACT

This application is used to improve the canteen or restaurants. Using this application, restaurants or canteens can take the order themselves. They have food in the canteen at the time of more crowded times. Most of the people can go to the canteen at the same time, because, in educational institutions, corporate offices giving the same lunchtime or all employees, so within few hours the canteen employees serve a crowd. So they need an application for taking orders online. In this application, people can order food online using sites, once they have paid for food, they can get the order id or booking id. They have to note that id and take it with them while they go to the canteen. They should show that id to the canteen service people. The service people can verify that id and serve the food to the concerned person.

1.1 Introduction

An ordering system is a collection of specific procedures used to manage the ordering process. 60% of Americans place a delivery order at least once a week. Revenue from meal deliveries in the US rose by 17% between 2019 and 2020, from \$22 billion to \$26.5 billion. The CAGR (compound annual growth rate) for the Indian online food ordering market is predicted to be 16.2%, reaching USD 17.02 billion by 2023. Ordering food can be done manually or through a computer. Many opportunities are emerging on the web or in mobile applications in the modern generation due to the rapid development in the use of the internet and the technology linked with it. The customer self-ordering system, which is made possible by this, aids the customer in placing their own food orders. It can be described as a computerized system that college personnel and students use to place their own orders at the "**SONA SMART CAFETERIA**".

This is made feasible by the deployment of a payment system that accepts electronic payments. For staff and students, Google Pay or Phone Pay can be used to make the payment. Anyone (staff or student) can order food from the Sona smart

cafeteria website, pay the price inside the website, and receive a special code for that order. After that, head to the canteen and buy the food by scanning the code. By using a computer system to connect each and every food ordering transaction rather than the data recorded on it, the system will become a crucial tool used by the canteen staff to update the food availability and the canteen to improve the management aspect. It can also increase productivity for the cafeteria, staff, and students by reducing human error or delivery, consuming less time, and offering clients high-quality service.

1.2 Objective

Nowadays, ordering food is one of the most fundamental aspects of daily living. In the college canteen, students can buy their food by standing in line for a long time to pay the bill, and occasionally there are some billing problems due to human error. Students choose to use Swiggy and Zomato to order food while in college, however, this had an impact on smaller businesses like campus canteens. To get over these challenges, we developed the "SONA SMART CAFETERIA" website, which allows anyone to order food from the campus cafeteria from anywhere. It generates a pin or number for their orders after payment. We use HTML, CSS, PHP, and MYSQL to build websites and store data in databases. The fundamental components of websites around the world are HTML, CSS, and PHP, with CSS managing a page's look and PHP scripting its functionality. Students must have a mobile device and access to the internet in order to submit orders. Staff in the canteen need a single system to update the food's availability and a scanner to scan the code/pin.

1.3 About domain

A web application (or web app) is application software that runs on a web browser, unlike software programs that run locally and natively on the operating system (OS) of the device. Web applications are delivered on the World Wide Web to users with an active network connection.

Writing web applications is often simplified with the use of web application

frameworks. These frameworks facilitate rapid application development by allowing a development team to focus on the parts of the application which are unique to their goals without having to resolve common development issues such as user management. Many of the frameworks in use are open-source software.

The use of web application frameworks can often reduce the number of errors in a program, both by making the code simpler and by allowing one team to concentrate on the framework while another focuses on a specified use application that is exposed to constant hacking attempts on the Internet, security-related problems can be caused by errors in the program. Frameworks can also promote the use of best practices such as getting after posts.

In addition, there is potential for the development of applications on the Internet operating system, although currently there are not many viable platforms that fit this model.

1.3.1 Online Shopping

Online shopping is a form of electronic commerce that allows consumers to directly buy goods or services from a seller over the Internet using a web browser or a mobile app. Consumers find a product of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping search engine, which displays the same product's availability and pricing at different e-retailers. As of 2020, customers can shop online using a range of different computers and devices, including desktop computers, laptops, tablet computers, and smartphones.

An online shop evokes the physical analogy of buying products or services at a regular "bricks-and-mortar" retailer or shopping center; the process is called business-to-consumer (B2C) online shopping. When an online store is set up to enable businesses to buy from another business, the process is called business-to-business (B2B) online shopping. A typical online store enables the customer to browse the firm's range of products and services, and view photos or images of the products, along with information about the product specifications, features, and prices.

Online stores usually enable shoppers to use "search" features to find specific

models, brands, or items. Online customers must have access to the Internet and a valid method of payment in order to complete a transaction, such as a credit card, an Interac-enabled debit card, or a service such as PayPal. For physical products (e.g., paperback books or clothes), the e-tailer ships the products to the customer; for digital products, such as digital audio files of songs or software, the e-tailer usually sends the file to the customer over the Internet. The largest of these online retailing corporations are Alibaba, Amazon.com, and eBay.

1.3.2 Online Ordering

Online ordering is the process of ordering products, for delivery or pickup, from a website or other application. The product can be either ready-to-eat food (e.g., direct from a home kitchen, restaurant, or a ghost kitchen) or food that has not been specially prepared for direct consumption (e.g., vegetables direct from a farm/garden, fruits, frozen meats. etc.).

2.SYSTEM ANALYSIS

2.1 Existing System

When people are working or studying somewhere they should have an interval in between the work or classes. So in that interval time, they have to buy some snacks or food at the canteen. So if people want to buy snacks or food at the canteen at the interval time, there is much crowd. So the much time takes to buy food at the canteen and it's a big task standing in the queue at the canteen.

EXISTING SYSTEM-DIS ADVANTAGES:

- People should stand in the queue to buy food in the canteen,
- Time waste,
- Not easy to select the food,
- No menu card,
- No centralized web application for food orders in canteens.

2.2 Proposed System

The objective of the system will provide people with all the information about the foods available at the different canteens. And they can know about the special items at the canteen. At the same time, they can order the foods online and receive them at the canteen itself and payment can be done online. The application has modules that

include updating the daily menu, ordering, the status of an order, payment system, and restriction of order counts.

PROPOSED SYSTEM- ADVANTAGES:

- A user-friendly web app,
- No need to stand in the queue in the canteen,
- A web-based menu card,
- A category-wise search is very useful to search the food items,
- A secure online payment.

3. MODULES

3.1 Admin

Admin can add the food list with the description, price, and available quantity. He can add the food category-wise. Users can view the orders given by the end-user. The canteen owner acts as an admin of this application. He can manage the canteen products details, ordered details, etc

3.2 Customer

The user can register them. After that, he can search the food theatre-wise, etc. Then he can order the needed food online. He can pay for the food online. The people are acts as end-user.

3.3 Food Ordering

In this module, the customers can search, view, and select the food and order it by giving their personal details. Once they order the food, they can get the order id or booking id for further reference.

3.4 Payment

In this module, the payment can be done using a payment gateway. After the amount is transferred intimation can send to this user.

4.SYSTEM SPECIFICATION

4.1 Front End

The original release of PHP was designed and created by Rasmus Lerdorf back in the middle of the 90s as a way of making various common web tasks easier and less repetitive. Back then, the main goal was to have the minimum amount of

logic as was possible in order to achieve results, and this led to PHP being HTML-centric – that is, PHP code was embedded inside HTML.

The first popular version of PHP was called PHP/FI 2.0, for Personal Home Page / Form Interpreter, and, despite its parsing inconsistencies, managed to attract a fair few converts, including myself. The main issue with this version was that the PHP/FI parser was largely hand-written, and so users often encountered scripting errors that were not technically errors – they were just the PHP/FI parser screwing up. Furthermore, the parser was absolutely tied to the Apache web server and was hardly renowned for its speed.

PHP 3 also finally made the language extensible – something that was seriously lacking from prior versions. Particularly keen developers were able to write their own modules for the language, adding functionality at the core level. The parser itself, though, was still tied to Apache, and, although speed was improved a great deal from PHP/FI, it still was not anything to shout about.

The only downside to upgrading to PHP 3 was that the language was a lot stricter – some code that worked on PHP/FI would no longer work after upgrading. The language was still young, though, so not many were affected.

With PHP 3, the language had gained limited object-oriented support, and this only added extra fuel to the fire of PHP's growth. By the time PHP 3 was replaced in the middle of 2000, it was installed on over 2,500,000 website domains, as compared to 250,000 just 18 months before.

In the middle of 2000, PHP 4 was released to the world, containing major differences from PHP 3 in all aspects. Extensive work had been done to ensure that backward compatibility with older PHP scripts would remain – upgrading from PHP 3 to PHP 4 was much smoother than the PHP/FI to PHP 3 upgrade.

Perhaps the most important change made for PHP 4 was the switch to what is called the Zend Engine. The Zend Engine was created by Zend, a company founded by Zeev Suraski and Andi Gutmans (the name Zend is a contraction of Zeev and Andi) to promote PHP in the corporate environment, allowing much more flexibility than had ever been seen in PHP before. The engine took over the core of PHP and introduced reference counting, whereby all resources used in scripts (database connections, files, etc) are tracked automatically by the engine and freed when no longer used to minimize memory usage and ensure there were no memory leaks.

PHP 5 was a big step forward for the language, although admittedly not as big as the jump from PHP 3 to PHP 4. The release is focused on language maturity and offers a lot of new functionality that has simply been missing from previous versions simply because the language was a little too simple to properly support larger projects.

PHP 5 brought with it huge steps forward for object-oriented scripts – developers are now able to declare how their objects may be used, which makes it easier for one developer to work with another's code. Furthermore, there is a wide variety of functions available for objects that make them much more flexible and easy to work with – unified constructors are only available from v5 upwards.

PHP 5 also brought with it new error checking in the form of try/catch – something that programmers from other languages had been enjoying for a long time. Furthermore, objects are now always handled as references in order to help programmers who just do not understand how objects work.

There are three main areas where PHP scripts are used.

- Server-side scripting. This is the most traditional and main target field for PHP. You need three things to make this work. The PHP parser (CGI or server module), a web server, and a web browser. You need to run the web server,

with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server. All these can run on your home machine if you are just experimenting with PHP programming. See the installation instructions section for more information.

- Command-line scripting. You can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on *nix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks. See the section about Command line usage of PHP for more information.
- Writing desktop applications. PHP is probably not the very best language to create a desktop application with a graphical user interface, but if you know PHP very well, and would like to use some advanced PHP features in your client-side applications you can also use PHP-GTK to write such programs. You also have the ability to write cross-platform applications this way. PHP-GTK is an extension to PHP, not available in the main distribution. If you are interested in PHP-GTK, visit its own website.

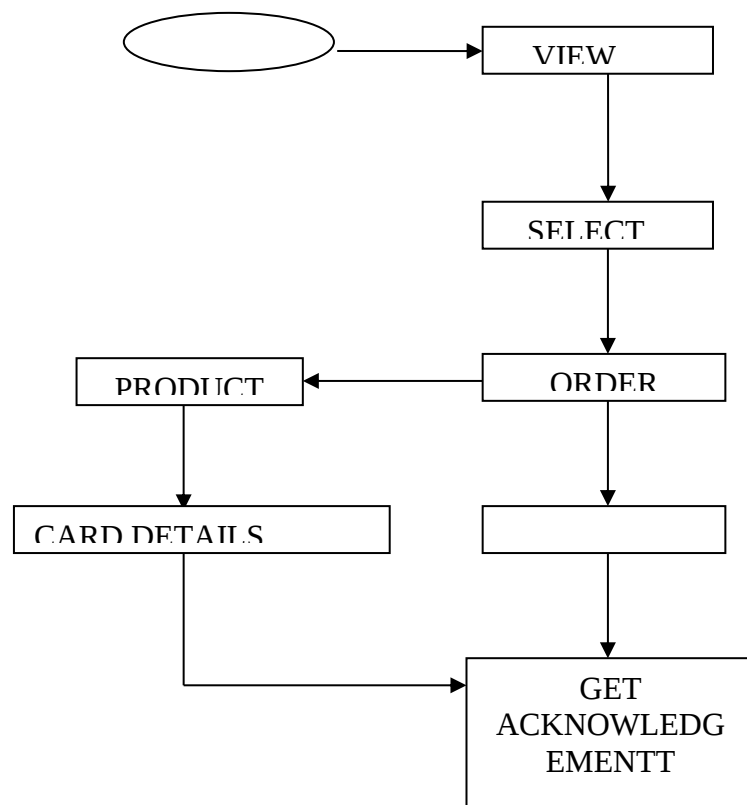
5.IMPLEMENTATION

5.1 SYSTEM DESIGN

The user must first log in using their ID on the login page before choosing from the various food options in the list and paying online or by COD. Obtain their payment acknowledgment as shown in fig.

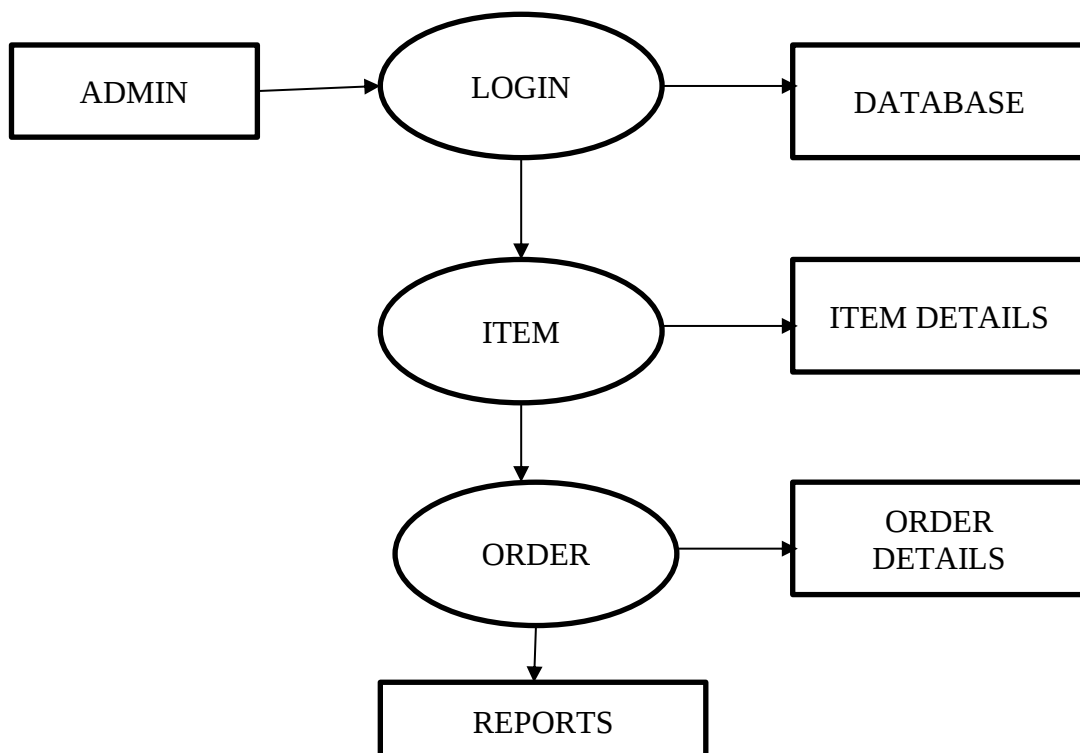
On the admin side, the database that stores the food products that are available in the canteen must be updated as shown in fig.





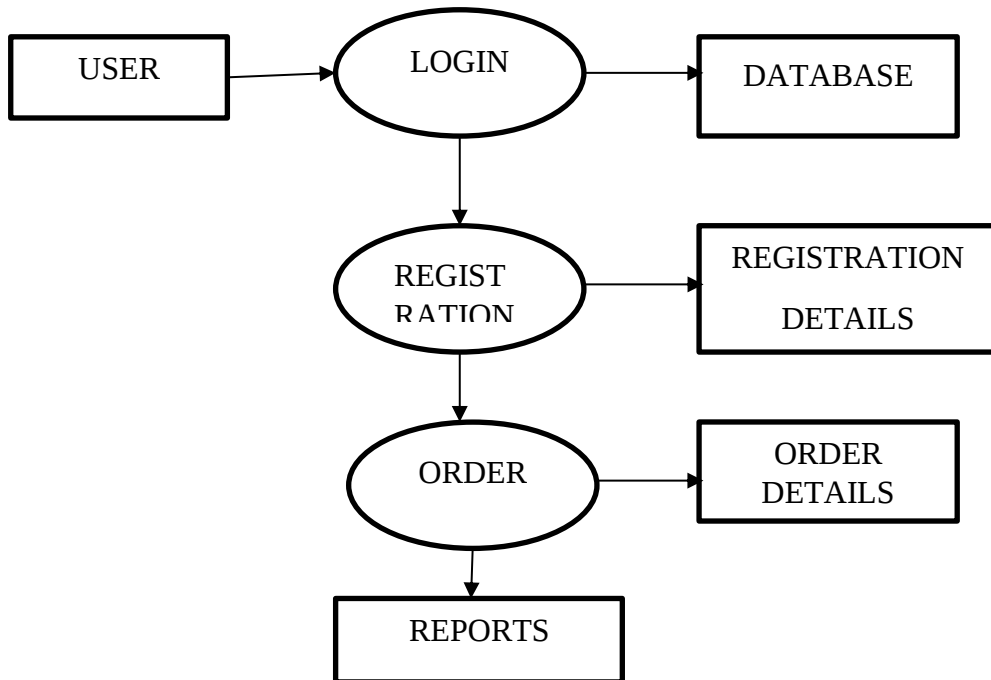
LEVEL: 1 Admin

The admin can view order information, including payment and the number of food items, from the admin side which is stored in the database fig.



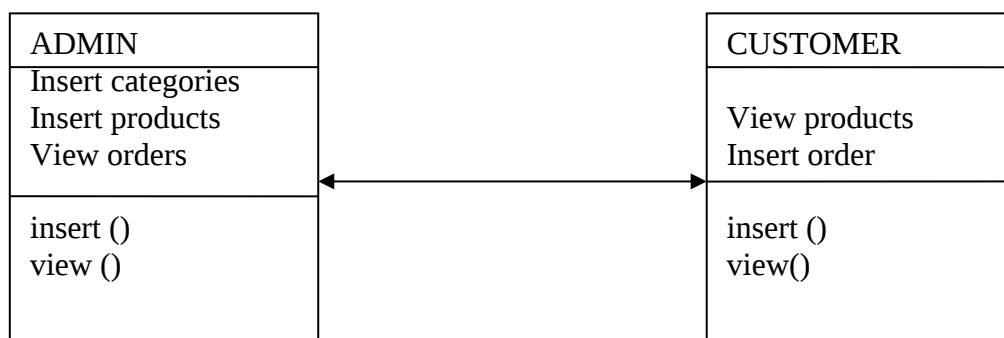
LEVEL: 2 User Login

On the user's end, they must first register in order to log in, pay for their order, and obtain the report as show in fig.



CLASS DIAGRAM:

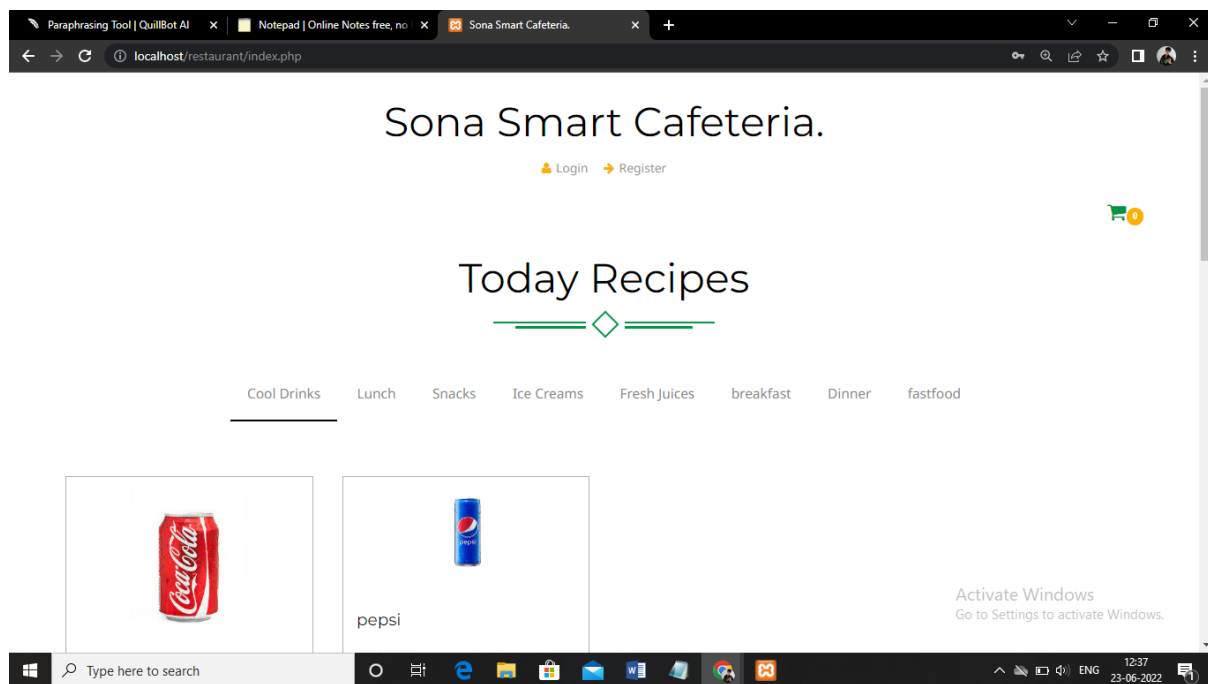
The admin has updated the categories and products by using insert() and view().on the client side is also able to view and select products insert(), and view() as shown in fig.



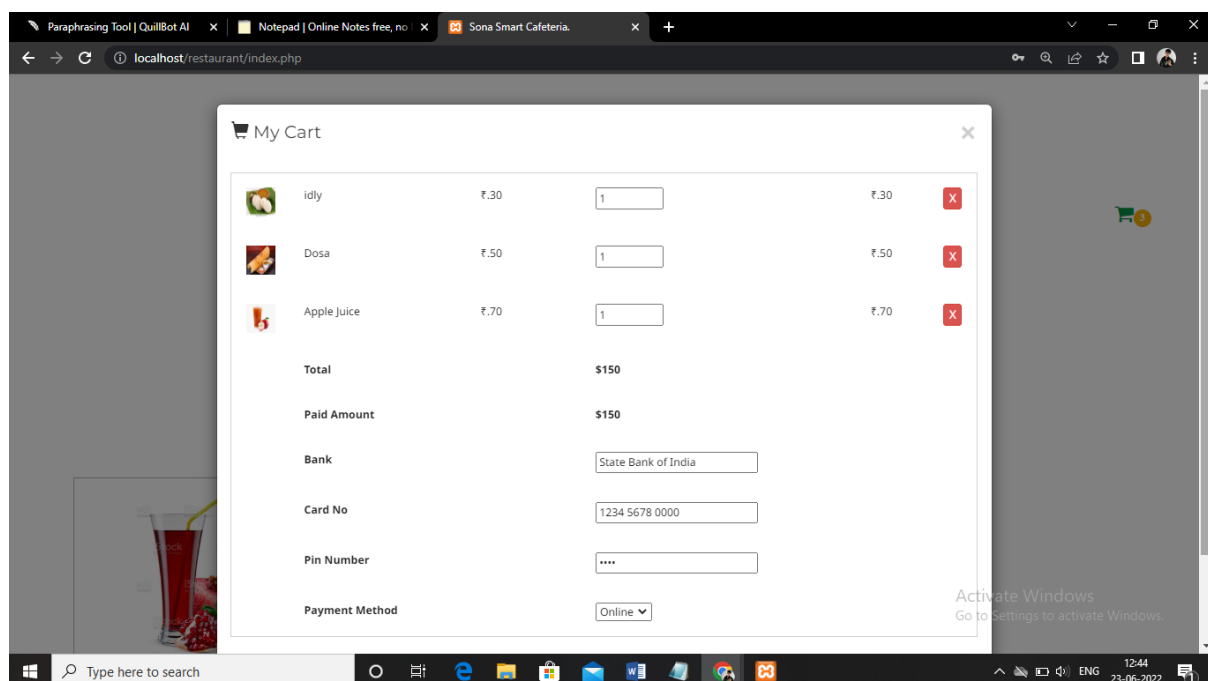
6. RESULTS AND DISCUSSION

6.1 Screenshots of Output

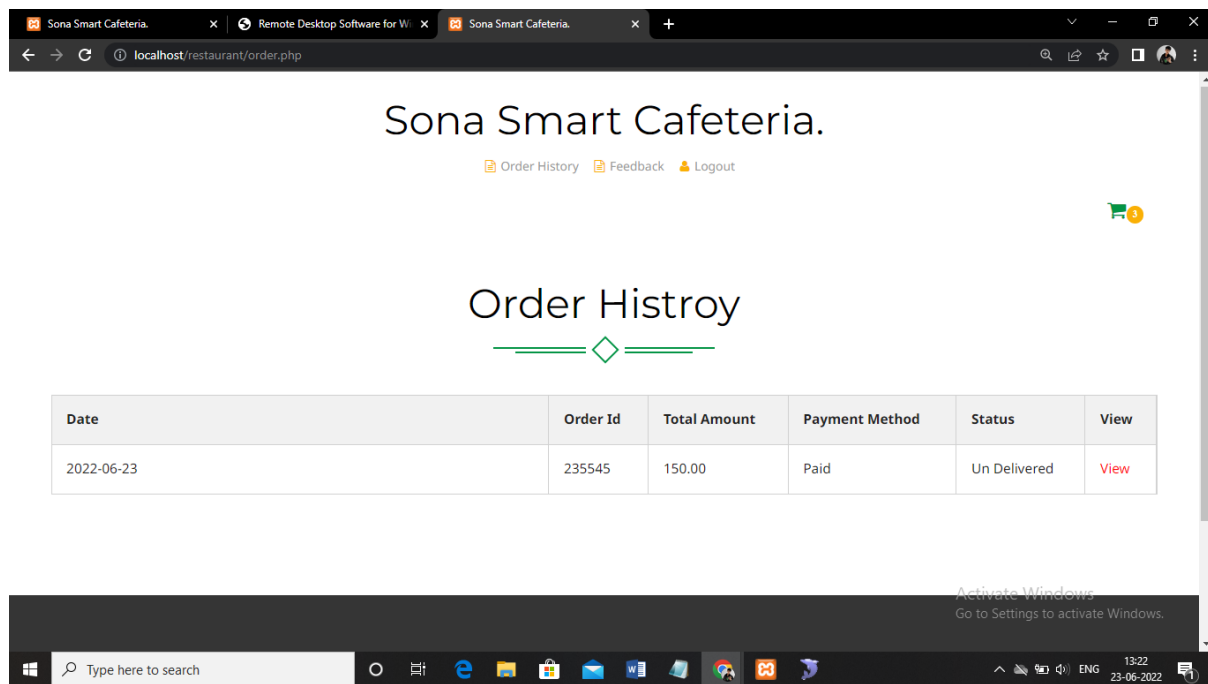
Clients must register if they are using the site for the first time. After that, they are able to log in as shown in fig.



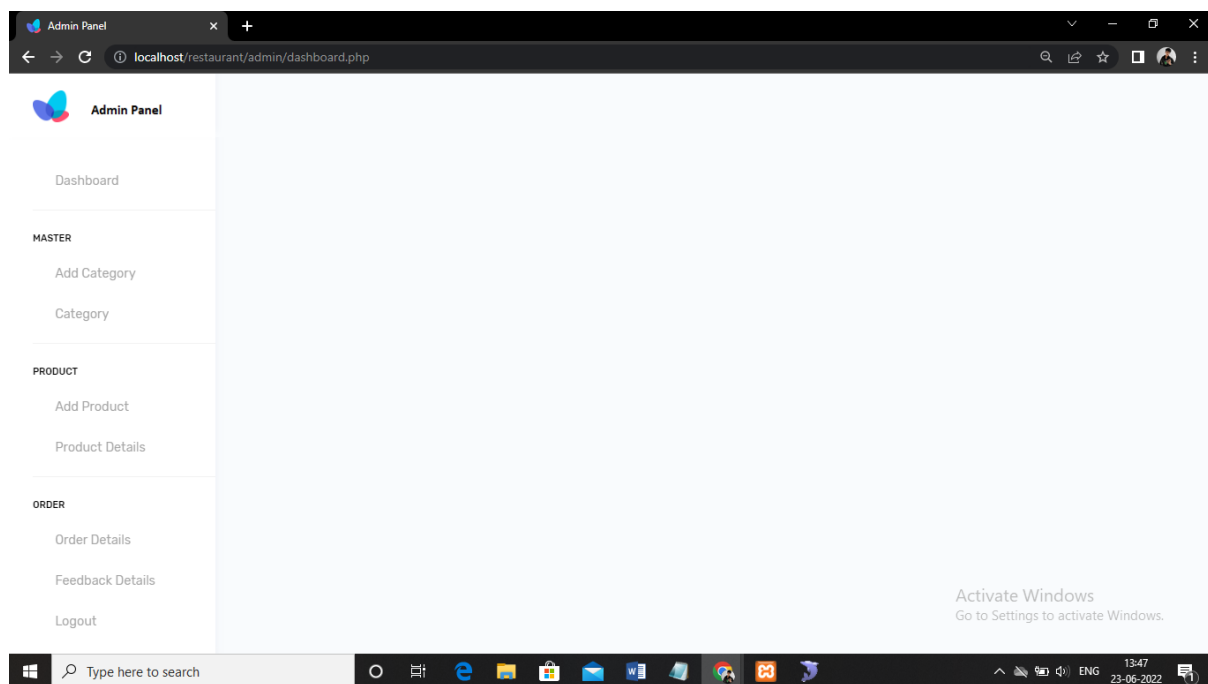
Following login, the customer has chosen the food items in the available categories based on their needs. The chosen food products are in the cart, the client confirms their order once again, and they must pay online or by COD as shown in fig.



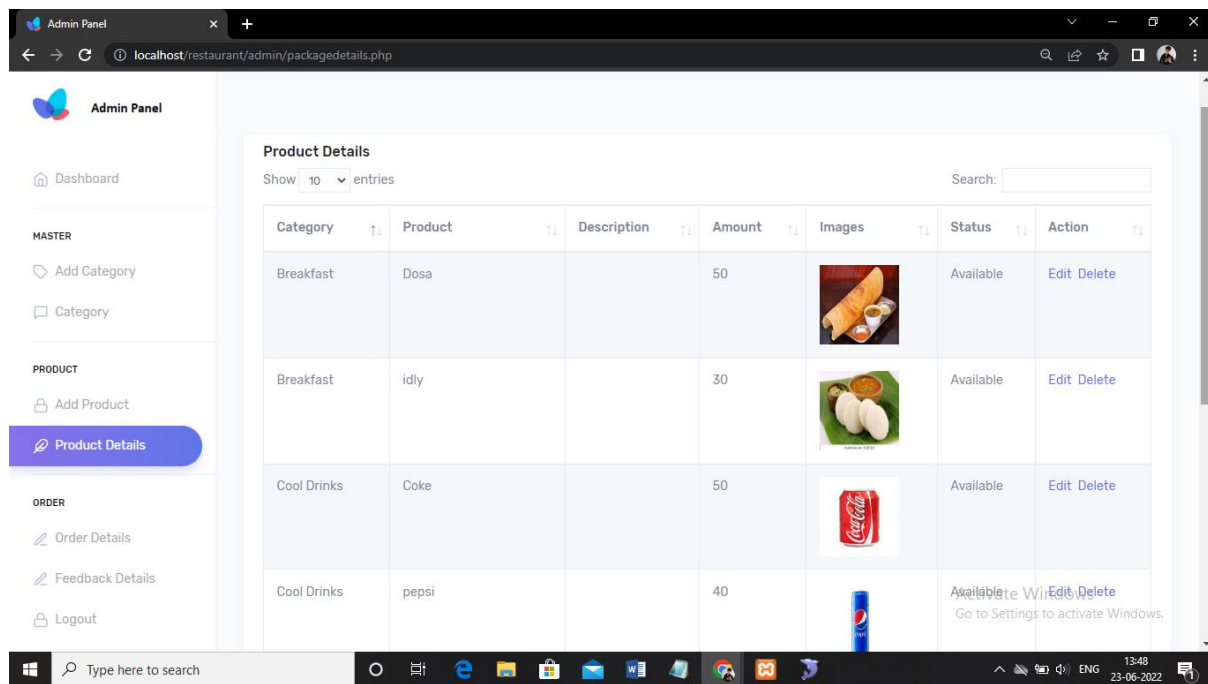
Following payment completion, we can obtain the ID, check it in the order history, and you can provide feedback as shown in fig.



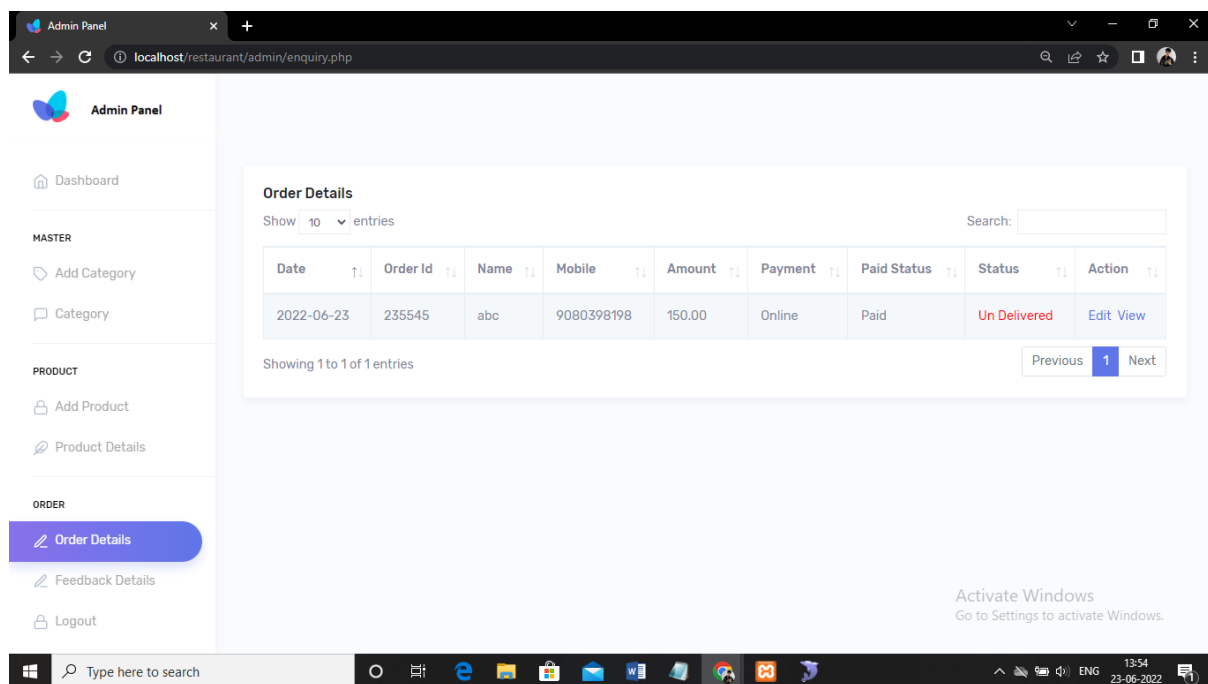
At the admin panel, after login, the admin can able to edit the details in the following options as shown in fig.



The admin can modify the food item's price, availability, and description in the product information and delete the food items as needed as shown in fig.



The admin can look up the customer ID, payment information, and edit status in the order data as shown in fig.



7. CONCLUSION AND FUTURE ENHANCEMENT

This project “Food Order Online” has been developed by using PHP as a front end and MY-SQL as a backend. This system has been developed for the management of the event. The system has been designed to be flexible and can therefore support any level of expansion and customization also maintenance costs will be drastically

reduced. The programming techniques used in the design of the system provide scope for further expansion and implementation on any charges. This project gives the complete details about the customer's registration, order, product maintenance, and the corresponding reports generated from the system.

This proposed system will be useful in reducing manpower. This system will avoid maintaining a huge amount of records. Searching the record is made easy with this system and hence the system saves a lot of time. The system provides flexibility for incorporating new features which may be necessary for the future.

This is a web-based application, in the future we implement this as a mobile application, so we can order the food using the mobile phone itself using an android studio or some other.

DESIGN OF HYBRID ADDER USING KOGGE-STONE

V.TAMILSELVAM , L.SATHEESHKUMAR

ABSTRACT

The Arithmetic Logic Unit is an essential component of the CPU. The ALU contains the logical circuit to perform mathematical operations like addition, subtraction, multiplication, division, logical operations and logical shifts on the values held in the processor registers or its accumulator. The adders are elementary component of ALU. In critical path of the ALU Design, adders are the main component. The critical path adder design based on the speed, power consumption, area level and delay improvement qualities. In this Project, a Hybrid adder design approached for Low power application circuits and critical path ALU Design. A Hybrid adder concept based on the Hybrid Variable Latency method and Parallel Prefix Architecture (PPA). The Hybrid adder consists of Ripple carry adder, Carry Lookahead adder for LSB and Kogge-stone adder for MSB. The adder was designed in Verilog HDL on Altera (Quartus II 13.0sp1). Each adder showed some advantages, but the proposed adder design have more advantage compared to the Ripple carry adder, Carry Lookahead adder, Carry save and carry skip adder. Because, the proposed design have a minimum improvement delay in ALU design.

1.INTRODUCTION

In most of the digital circuits arithmetic addition is important and elementary operation that is encountered in the critical path. Adders have influence in system performance. If the hardware integrated, the adder plays a crucial role. Basic binary adders are arithmetic circuits in the form of half adders and full adders used to add together two binary digits. Which can be constructed using just a few basic logic gates. It can provide the sum and carry of the operand. The analysis time of addition is directly proportional to numbers of operands. The propagation of the carry affects the operational speed of the device. There are different types of adders in

the literature. Different types of adders are single bit, multi-bit, tree-based structures. Half adder and full adder are single bit type. Ripple carry adder (RCA), Carry save adder (CSA), Carry skip adder (CSKA) and Carry lookahead adder (CLA) comes under Multi-bit adder category. Tree based structure include a Kogge stone adder, Brent Kung adder and Ling adder. The Carry skip adder represents how propagation of carry takes place from the initial to final stage along with Parallel Prefix Architecture (PPA).

In this project, the Hybrid adder design based on Hybrid variable latency method and PPA. The Parallel Prefix Architecture mainly deals with the carry propagation part. The Hybrid Variable latency method is Varied the stage size. The stage size varies from least significant bit (LSB) to the most significant bit (MSB). The carry propagation also has an influence on the stage size. Hybrid variable latency technique is introduced to optimize the carry propagation path. Each adder have the different level of power consumption and area. In this design we use the tree-based Kogge stone adder for MSB part and the LSB part again split into the form of the MSB and LSB for the MSB Carry Lookahead adder(CLA) design used and LSB part the Ripple Carry adder (RCA) used. Compared to the other adders like CLA, CSKA, CSA it is improve the Minimum delay. In future we increase the speed according to reduce the speed.

2.1 Review on Adder Architecture

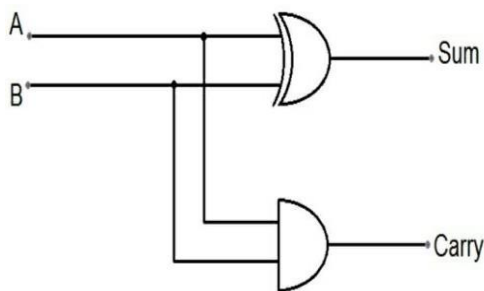
An adder is a digital logic circuit in electronics that is extensively used for the addition of numbers. In many computers and other types of processors, adders are even used to calculate addresses and related activities and calculate table indices in the ALU and even utilized in other parts of the processors. These can be built for many numerical representations like binary coded decimal. Addition operation performed by many common ALU designs, DSP functional units such as a Finite impulse response (FIR) filters, FFT modules and Microprocessor and controllers. Adders are basically classified into two types: Half Adder and Full Adder. Multiple bits digital circuits are critical path design circuits. In critical path design, critical

path delay, power consumption and area levels are decide the quality of the adder design. For the speed of the adder design, the analysis time of addition is directly propotional to the number of operands.

2.2 Half adder and Full adder

Half Adder is a combinational logic circuit which is de- signed by connecting one EX-OR gate and one AND gate. The half ad- der circuit has two inputs: A and B, which add two input digits and generates a carry and a sum.

The output obtained from the EX-OR gate is the sum of the two numbers while that obtained by AND gate is the carry. There will be no forwarding of carry addition because there is no logic gate to pro- cess that. Thus, this is called Half Adder circuit.



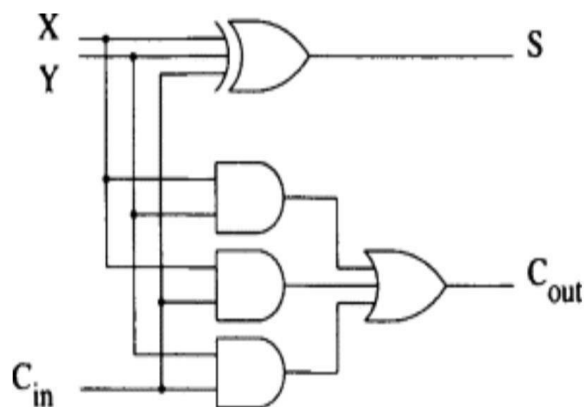
Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

2.1 Half adder

2.1 Truthtable for Half adder

Full adder

Full Adder is the adder that adds three inputs and produces two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM. A full adder logic is designed in such a manner that can take eight inputs together to create a byte- wide adder and cascade the carry bit from one adder to another. we use a full adder because when a carry-in bit is available, another 1-bit adder must be used since a 1-bit half-adder does not take a carry-in bit. A 1- bit full adder adds three operands and generates 2-bit results.



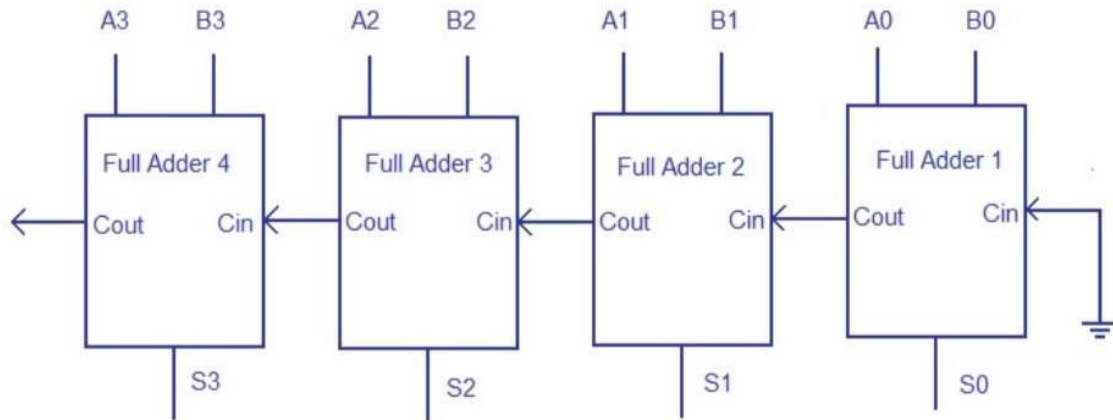
2.2 Fulladder

Inputs			Outputs	
A	B	C _{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2.2 Truth table for Full adder

2.3 Ripple Carry Adder

Multiple full adder circuits can be cascaded in parallel to add an N-bit number. For an N-bit parallel adder, there must be N number of full adder circuits. A ripple carry adder is a logic circuit in which the carry-out of each full adder is the carry in of the succeeding next most significant full adder. It is called a ripple carry adder because each carry bit gets rippled into the next stage. In a ripple carry adder the sum and carry out bits of any half adder stage is not valid until the carry in of that stage occurs. Propagation delays inside the logic circuitry is the reason behind this. Propagation delay is time elapsed between the application of an input and occurrence of the corresponding output. Consider a NOT gate, When the input is “0” the output will be “1” and vice versa. The time taken for the NOT gate’s output to become “0” after the application of logic “1” to the NOT gate’s input is the propagation delay here. Similarly the carry propagation delay is the time elapsed between the application of the carry in signal and the occurrence of the carry out (C_{out}) signal. Circuit diagram of a 4-bit ripple carry adder is shown below.



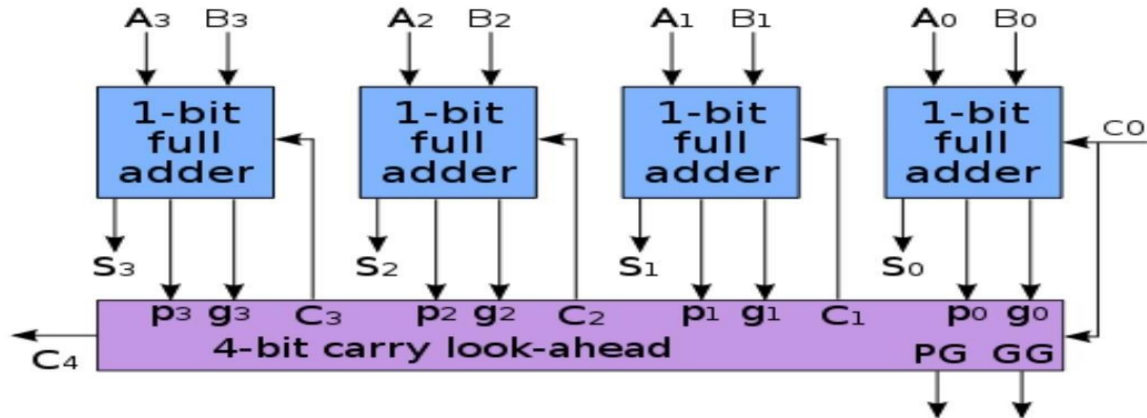
2.3 Ripple Carry adder

Sum out S_0 and carry out C_{out} of the Full Adder 1 is valid only after the propagation delay of Full Adder 1. In the same way, Sum out S_3 of the Full Adder 4 is valid only after the joint propagation delays of Full Adder 1 to Full Adder 4. In simple words, the final result of the ripple carry adder is valid only after the joint propagation delays of all full adder circuits inside it.

2.4 Carry Look-ahead adder

The adder produce carry propagation delay while performing other arithmetic operations like multiplication and divisions as it uses several additions or subtraction steps. This is a major problem for the adder and hence improving the speed of addition will improve the speed of all other arithmetic operations. Hence reducing the carry propagation delay of adders is of great importance. There are different logic design approaches that have been employed to overcome the carry propagation problem. One widely used approach is to employ a carry look-ahead which solves this problem by calculating the carry signals in advance, based on the input signals. This type of adder circuit is called a carry look-ahead adder. A carry look-ahead adder reduces the propagation delay by introducing more complex hardware. In this design, the ripple carry design is suitably.

Carry Lookahead Adder



Hence a carry signal will be generated in two cases:

1. Input bits A and B are 1
2. When one of the two bits is 1 and the carry-in is 1.

In ripple carry adders, for each adder block, the two bits that are to be added are available instantly. However, each adder block waits for the carry to arrive from its previous block. So, it is not possible to generate the sum and carry of any block until the input carry is known. The i th block waits for the $(i-1)$ th block to produce its carry. So there will be a considerable time delay which is carry transformed such that the carry logic over fixed groups of bits of the adder is reduced to two-level logic. Let us discuss the design in detail.

Consider the full adder circuit shown above with corresponding truth table. We define two variables as ‘carry generate’ G_i and ‘carry propagate’ P_i then,

$$P_i = A_i \text{ xor } B_i \quad G_i = A_i \cdot B_i$$

The sum output and carry output can be expressed in terms of carry generate G_i and carry propagate P_i as,

$$S_i = P_i \text{ xor } C_i \quad C_{i+1} = G_i + P_i \cdot C_i$$

where G_i produces the carry when both A_i , B_i are 1 regardless of the input carry. P_i is associated with the propagation of carry from C_i to C_{i+1} .

The carry output Boolean function of each stage in a 4 stage carry look-ahead

adder can be expressed as

$$C1 = G0 + P0.Cin$$

$$C2 = G1 + P1.C1 = G1 + P1.G0 + P1.P0.Cin$$

$$C3 = G2 + P2.C2 = G2 + P2.G1 + P2.P1.G0 + P2.P1.P0.Cin$$

$$C4 = G3 + P3.C3 = G3 + P3.G2 + P3.P2.G1 + P3.P2.P1.G0 + P3.P2.P1.P0.Cin$$

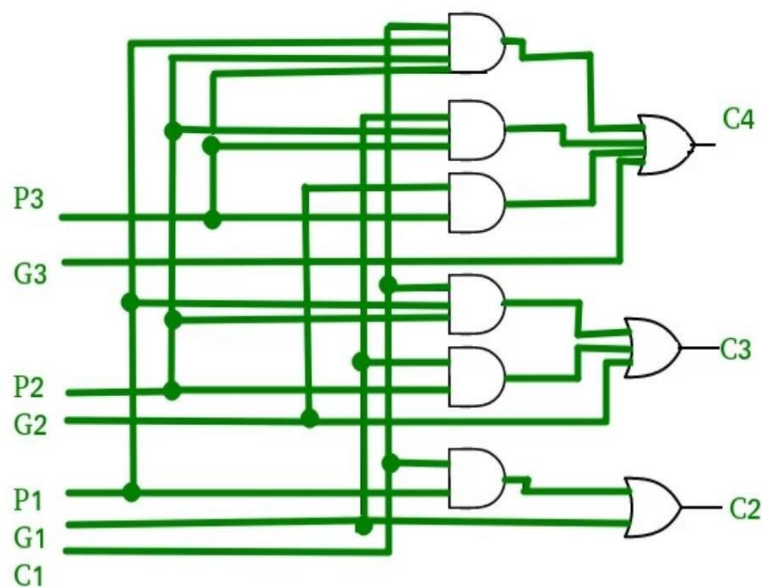
A	B	C	C +1	Condition
0	0	0	0	No Carry Generate
0	0	1	0	
0	1	0	0	
0	1	1	1	No Carry Propagate
1	0	0	0	
1	0	1	1	
1	1	0	1	Carry Generate
1	1	1	1	

2.4

Table for Carry Generation and Propagation

2.3.1

Carry and Propagation Unit



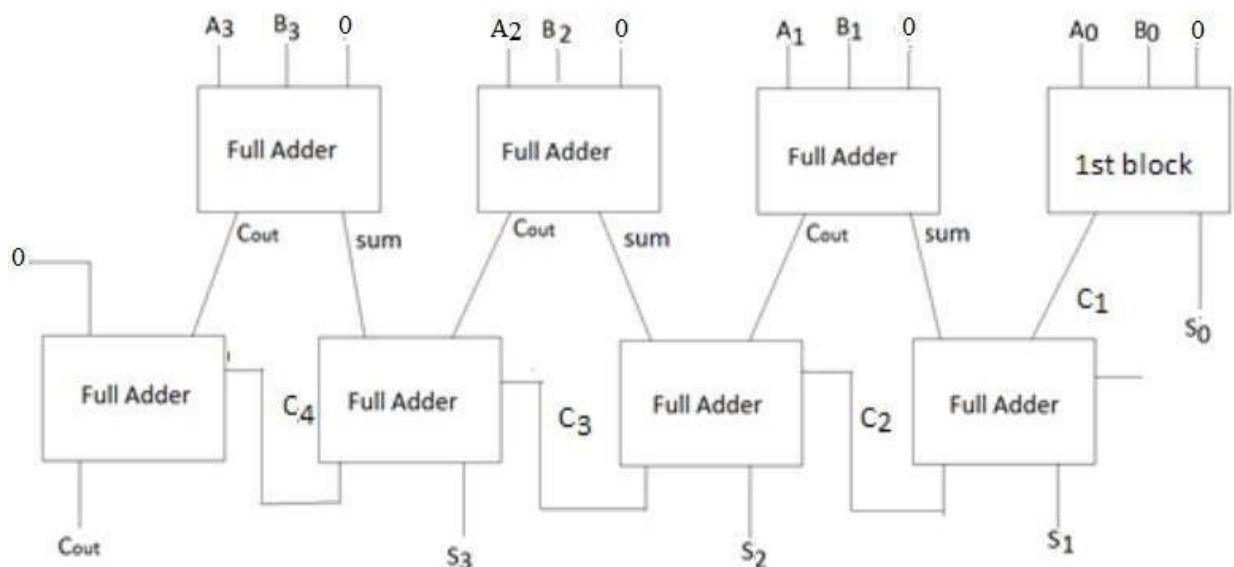
2.5 Carry Save adder

The carry save adder implements the method of calculating the partial carry and partial sum values at each stage to which is then, after left shifting the partial carry outputs by one place and padded with a zero, are added together using a ripple carry adder as shown in Figure The formulae to calculate the partial carries and partial sums are,

$$Psi = a_i \text{ xor } b_i \text{ xor } c_i$$

$$Sci = (a_i \wedge b_i) \vee (a_i \wedge c_i) \vee (b_i \wedge c_i)$$

Where, ps is the partial sum and pc is the partial carry at bit $i = 0, 1, 2, \dots$, Although this model is efficient in the case of small bit additions, it consumes high area and power at a small reduction in delay. Also the addition results are still not known at once, and the result of the addition is not known to be larger or smaller than a given number in the case of subtraction without the implementation of a comparator.

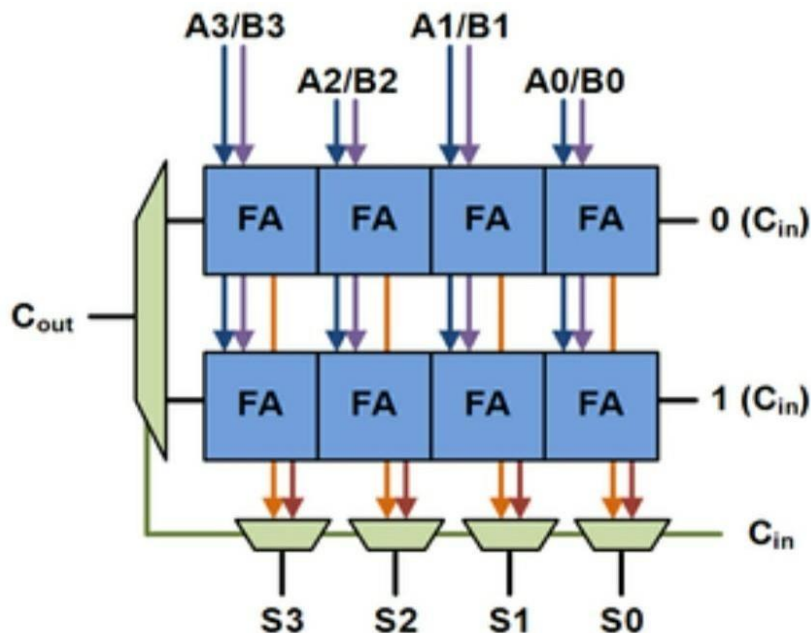


2.5

Carry Save Adder

2.6 Carry Select Adder

The carry-select adder generally consists of ripple-carry adders and a multiplexer. Adding two n -bit numbers with a carry-select adder is done with two adders (therefore two ripple-carry adders), in order to perform the calculation twice, one time with the assumption of the carry-in being zero and the other assuming it will be one. After the two results are calculated, the correct sum, as well as the correct carry-out, is then selected with the multiplexer once the correct carry-in is known. The number of bits in each carry select block can be uniform, or variable. In the uniform case, the optimal delay occurs for a block size of $\lceil \sqrt{n} \rceil$. When variable, the block size should have a delay, from addition inputs A and B to the carry out, equal to that of the multiplexer chain leading into it, so that the carry out is calculated just in time. The $O(\sqrt{n})$ delay is derived from uniform sizing, where the ideal number of full-adder elements per block is equal to the square root of the number of bits being added, since that will yield an equal number of MUX delays.



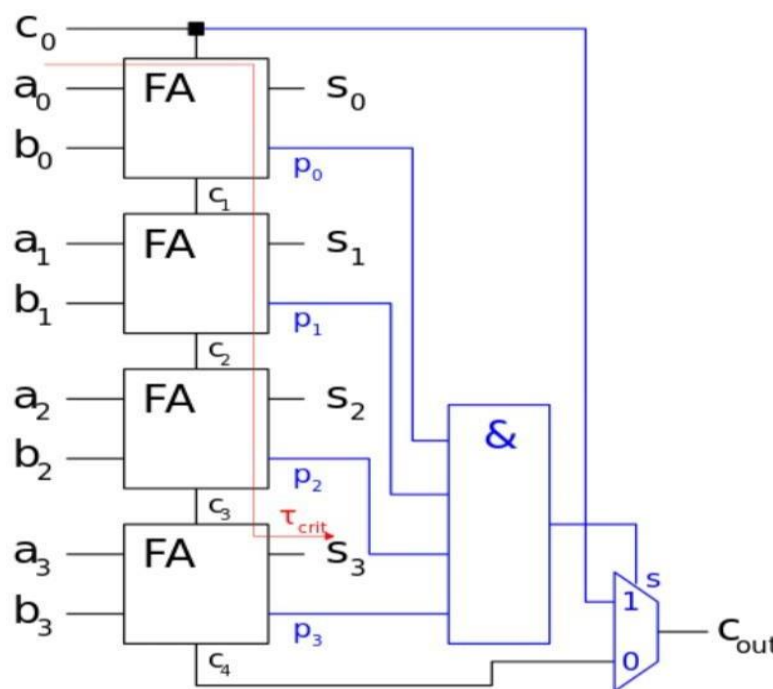
2.6 Carry Select Adder

Above is the basic building block of a carry-select adder, where the block size is 4. Two 4-bit ripple-carry adders are multiplexed together, where the resulting carry and

sum bits are selected by the carry-in. Since one ripple-carry adder assumes a carry-in of 0, and the other assumes a carry-in of 1, selecting which adder had the correct assumption via the actual carry-in yields the desired result.

2.7 Carry Skip Adder (Carry-bypass Adder)

Carry-Skip Adder (also known as a carry-bypass adder) is an adder implementation that improves on the delay of a ripple-carry adder with little effort compared to other adders. The improvement of the worst-case delay is achieved by using several carry-skip adders to form a block-carry-skip adder. The carry skip adder performs best when higher bits are produced through sliced inputs supplied to smaller carry skip blocks that are concatenated. The adder works in the fashion that the propagates are obtained from full adders are supplied to an and gate who's output determines whether the output carry is equal to the input carry or the output carry of the ripple carry block through a mux.



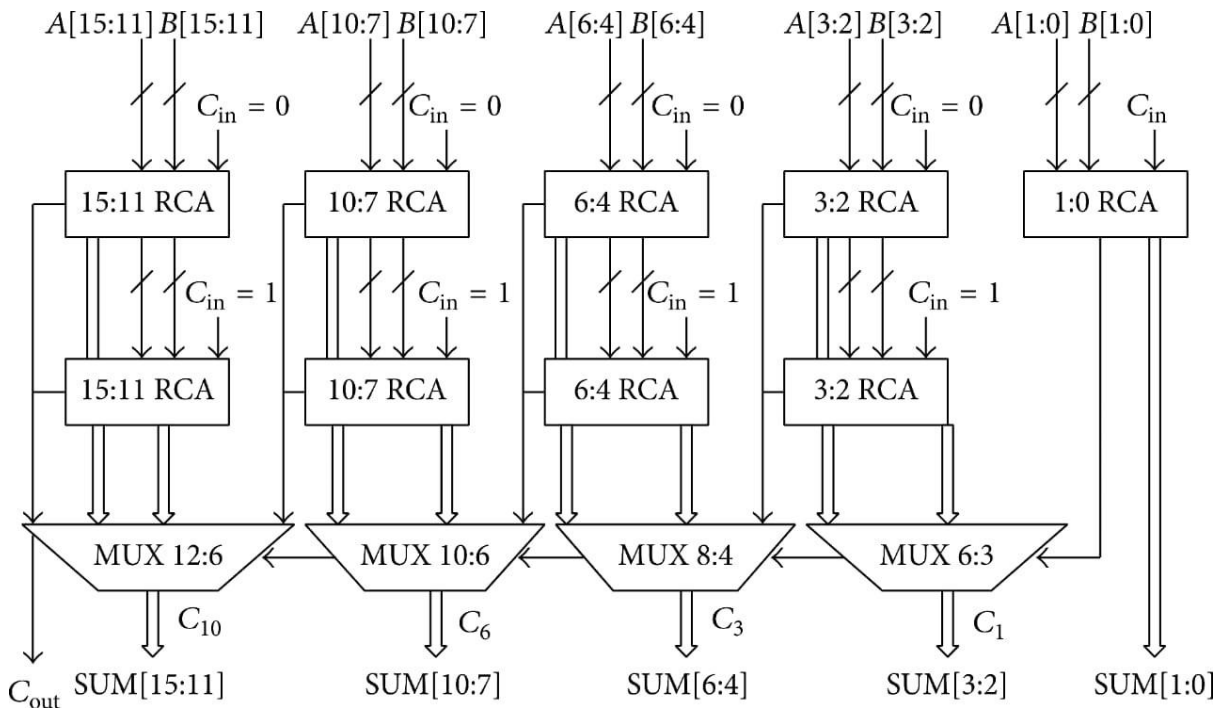
2.7 Carry Select Adder

previous blocks must be held in mind before optimizing each carry skip block of the adder. Unlike other fast adders, carry-skip adder performance is increased with only some of the combinations of input bits. This means, speed improvement is only

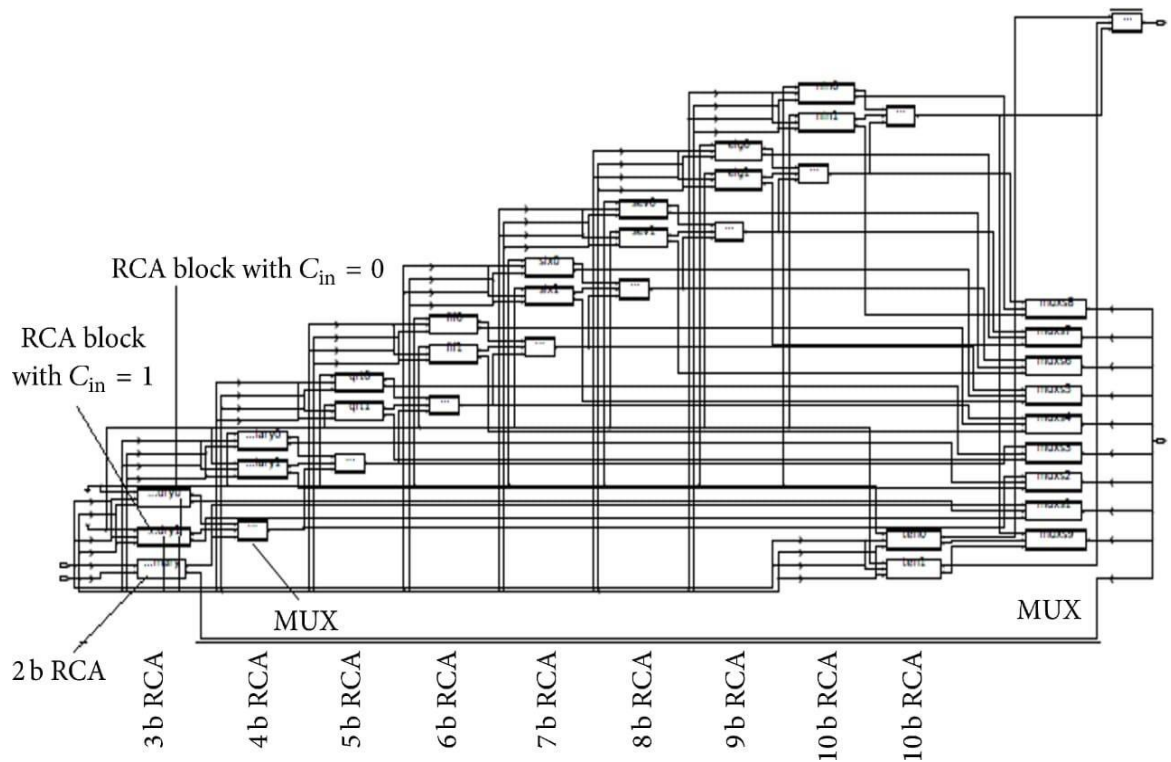
probabilistic. The drawbacks of this model is that the adder block must be sliced into smaller blocks of carry skip adders, as shown in Figure to obtain the output quickly in the case of higher bit additions as waiting for the propagate at each step would then resemble that of the Ripple Carry Adder's wait for carries. Also delays of previous blocks must be held in mind before optimizing each carry skip block of the adder. Unlike other fast adders, carry- skip adder performance is increased with only some of the combinations of input bits. This means, speed improvement is only probabilistic.

2.8 Square-Root Carry-Select Adder (SQRT-CSA)

A 16-bit CSA with variable size is created by cascading four CSA with variable input size. The block diagram for the same is given in Figure1 . We have five stages of CSA with different inputs bits to each stage. We have an adder with block sizes of 2, 2, 3, 4, and 5, respectively. In the modified SQRT-CSA, the block of CSA with $C_{in}=1$ is replaced with BEC (Binary to Excess Converter)



2.8 Square-Root Carry-Select Adder (SQRT-CSA)



2.8 RTL diagram of Square-Root Carry Select Adder

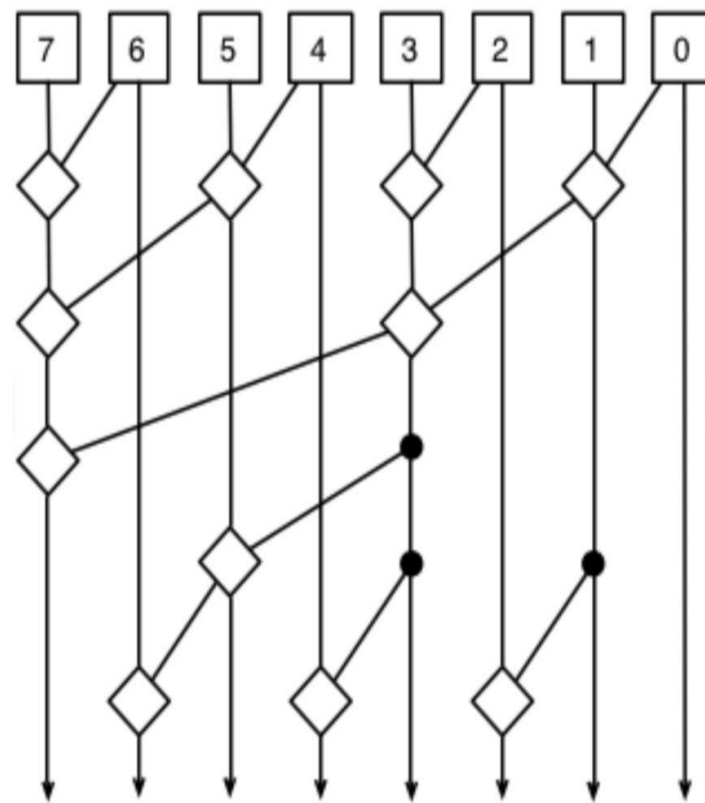
2.9 Brent kung Adder:

The Brent–Kung adder is a parallel prefix adder (PPA) form of carry- lookahead adder (CLA). Proposed by Richard Peirce Brent and Hsiang Te Kung in 1982 it introduced higher regularity to the adder structure and has less wiring congestion leading to better performance and less necessary chip area to implement compared to the Kogge–Stone adder (KSA). It is also much quicker than ripple-carry adders (RCA).

Ripple-carry adders were the initial multi-bit adders which were developed in the early days and got their name from the ripple effect which the carry made while being propagated from right to left. The time taken for addition was directly proportional to the length of the bit being added. This is reverse in Brent–Kung adders where the carry is calculated in parallel thus reducing the addition time drastically. Further work has been done on Brent–Kung adders and other parallel adders to reduce the power consumption and chip area as well as to increase the speed thus making them

suitable for low-power designs.

A Brent–Kung adder is a parallel adder made in a regular layout with an aim of minimizing the chip area and ease of manufacturing. The addition of n -bit number can be performed in time $O(\log_2 n)$ with a chip size of area $O(n \cdot \log_2 n)$ thus making it a good-choice adder with constraints on area and maximizing the performance. Its symmetry and regular build structure reduces costs of production effectively and enable it to be used in pipeline architectures. In parallel adders the critical path is decided by computation of the carry from least significant bit (LSB) adder to the Most significant bit (MSB) adder, therefore efforts are in reducing the critical path for the carry to reach the MSB.



2.9

Brent–Kung adder

Carry generation:

In general, most of the adders use carry-in and the corresponding bits of two numbers (A and B) to get the corresponding sum bit and carry-out -with ripple carry adders taking $O(n)$ time for carry to reach MSB.

(i) Considering that $A = a_n a_{n-1} \dots a_1$ and $B = b_n b_{n-1} \dots b_1$ both be n -bit binary numbers.

(ii) With sum being $S = s_{n+1} s_n \dots s_1$ and carry generated in each stage $C = c_n \dots c_0$ will be carry-in to next stages.

(iii) For RCA, $c_0 = 0$, and i the sum bit and carry bit generated are $c_i = g_i \vee (a_i \wedge c_{i-1}) \vee (b_i \wedge c_{i-1})$,

$s_i = a_i \oplus b_i \oplus c_{i-1}$ for $i = 1, 2, \dots, n$ $s_{n+1} = c_n$ respectively.

(iv) It is possible to transform the above ripple carry into carry-lookahead (CLA) by defining the carry bit i as

$$C_0 = 0,$$

$$c_i = g_i \vee (p_i \wedge c_{i-1}) \text{ where } g_i = a_i \wedge b_i \text{ and}$$

$$p_i = a_i \oplus b_i \text{ for } i = 1, 2, \dots, n.$$

p and g are known as carry propagate and carry generate. This corresponds to the fact that the carry c_i is either generated by a_i and b_i or propagated from the previous carry c_{i-1} .

They also defined a function $(G_i, P_i) = (g_1, p_1)$ for $i = 1$;

otherwise $(g_i, p_i) = (G_{i-1}, P_{i-1})$ for $i = 2, 3, \dots, n$.

It can be derived that G_i in the function is equivalent to c_i . Also (G_n, P_n) can be non-recursively written as $= (g_n, p_n) \circ (g_{n-1}, p_{n-1}) \circ \dots \circ (g_1, p_1)$.

Final Processing stage:

Using the carry propagation and generation transformation for working out addition and carry used by Brent and Kung, the performance of the adder increases considerably and also leads to an increase in regularity. The final sum can be calculated as follows: $s_i = p_i \oplus c_{i-1}$

2.10 Ling Adder:

Ling (1981) has proposed different carry generation equations, in which one propagate term is factored out to

simplify the group generate function, which in turn makes the first level of prefix tree simple. This further reduces the

critical path delay. Ling equations exploit the property that the group generate term is much more complex than the

group propagate term. Hence, by absorbing some complexity of the generate term to propagate term both the

functions will become balanced. Ling formulated a new generate function referred to as the pseudo carry or reduced

generate $H_{n::0}$ which is less complex than the conventional group generate $G_{n::0}$. The propagate term in Ling's

formulation is referred to as the pseudo propagate. The reduced generate makes the first level of carry tree simpler.

The factored propagate term will be combined with the simplified group generate function or the pseudo carry to

compute the final sum. According to Ling, the Ling pseudo carry can be given by,

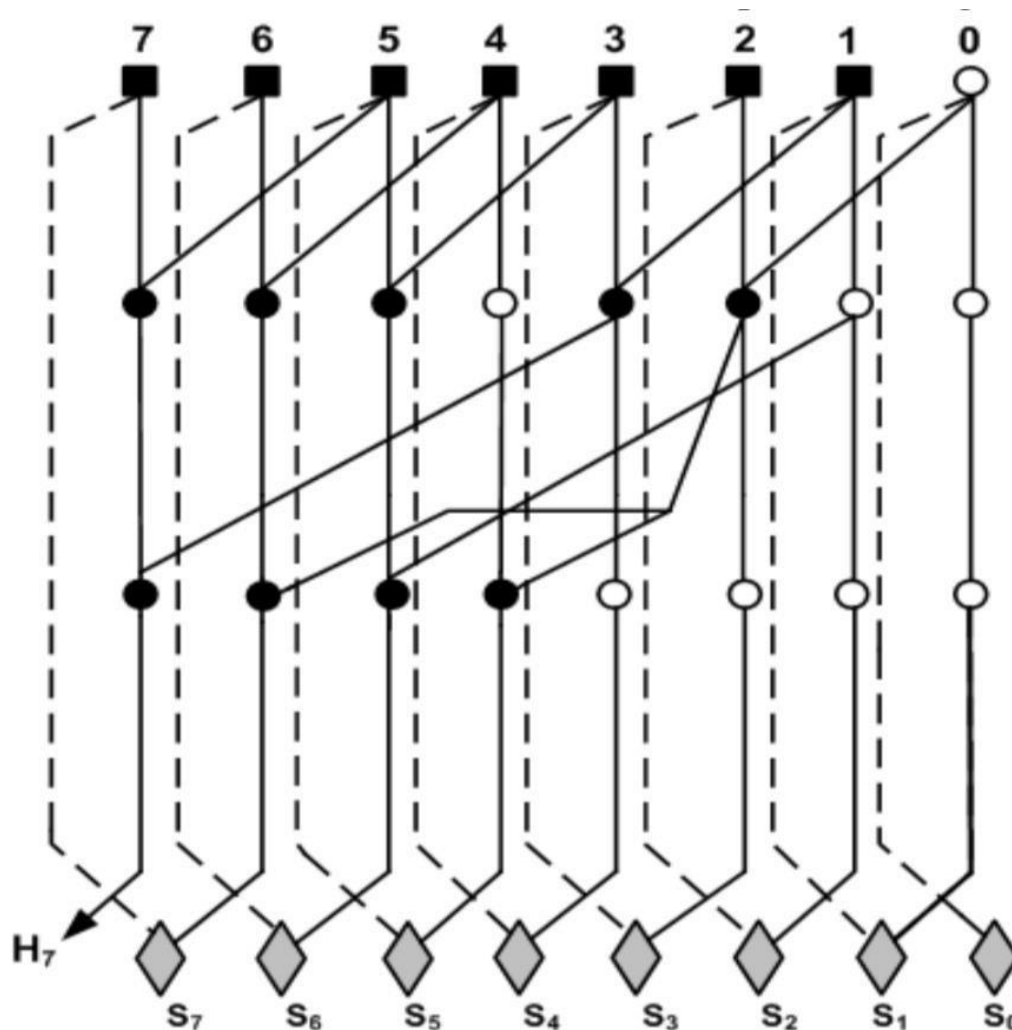
$$H_{i:j} = g_i + G_{i-1:j}$$

By taking the logical AND of H_i with p_i the conventional carry c_i can be recreated using equation

$$C_i = P_i \cdot H_i$$

Final sum bits for each bit position i , can be computed by $S_i = d_i \oplus C_{i-1}$

On the other hand, due to the reduced complexity of the Ling carries compared to the conventional carries, the computation time for the sum bits will be reduced.



2.10 Ling Adder

3.EXISTING METHOD

3.1 16 Bit Carry Lookahead Adder

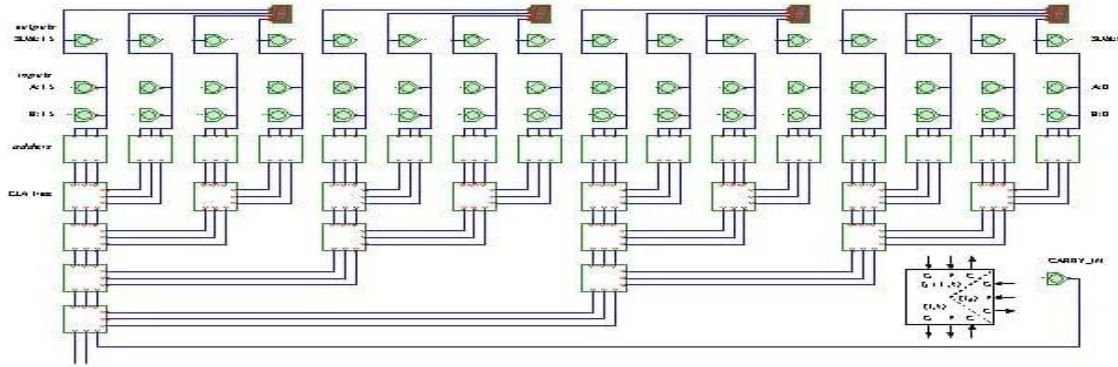
This circuit demonstrates a slowed-down version of the modular 16-bit carry-

lookahead adder, so that you can watch how the several carry signals propagate through the carry-lookahead tree.

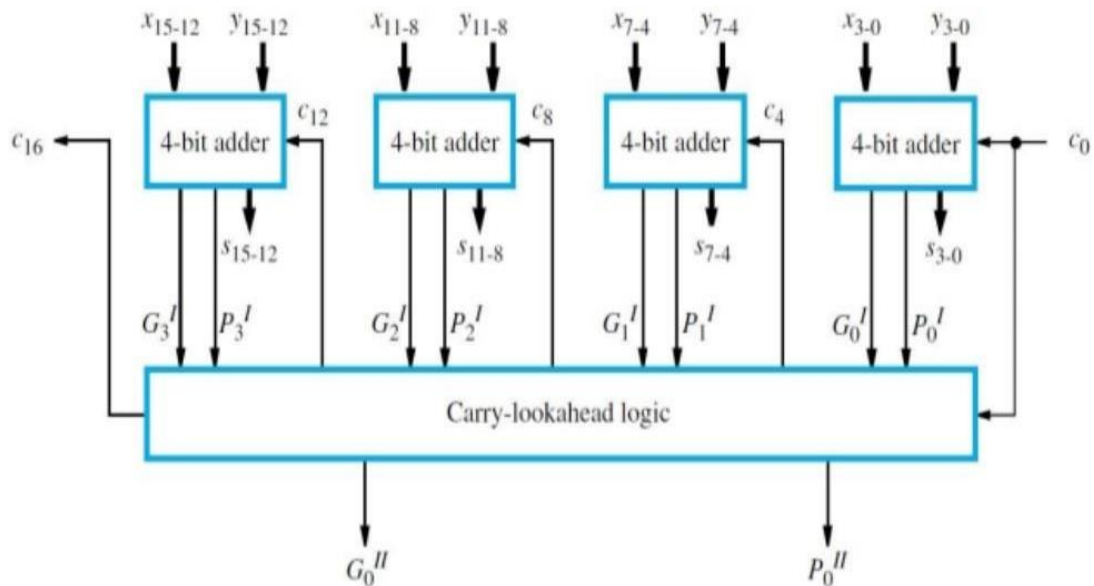
Click the input switches or type the '0'... '9' and 'a'... 'f' bind-keys to control the values of the B-input switches B0,..., B15 and type the 'x' bindkey to toggle the carry-input. The default values are zero for all A-inputs and ones for all B-inputs. Therefore, changing the carry-input will result in a full carry through all stages of the adder.

As in the previous slowed-down versions of the carry-ripple and carry-select adders, very long gate delays of 0.3 seconds have been pre-set for each gate. This means that you can easily watch how input changes propagate through the adder-blocks and the carry-lookahead tree. Again, the slowest path is from the least-significant adder block (A0,B0 on the right) to the most-significant adder block (A15,B15). Here, a change of A0 or B0 ripples down through the carry-lookahead tree back to the top level carry-lookahead block (bottom left), and then back up to the upper carry-lookahead blocks. This involves two adder-stage delays and 7 stages of the carry-lookahead blocks, while the 16-bit ripple-carry adder obviously involves 16 adder-stage delays. The speed advantage is even greater, because the CLA blocks are simpler and faster than the adder blocks.

The following just repeats the description already presented in the normal-speed version. The adder consists of 16 1-bit adder blocks and a tree of $(8+4+2+1)$ carry lookahead blocks, which are realized as sub designs called sum.



3.1.1 16 bit Carry Lookahead adder



3.1.2 16 bit Carry lookahead adder using 4bit Adder

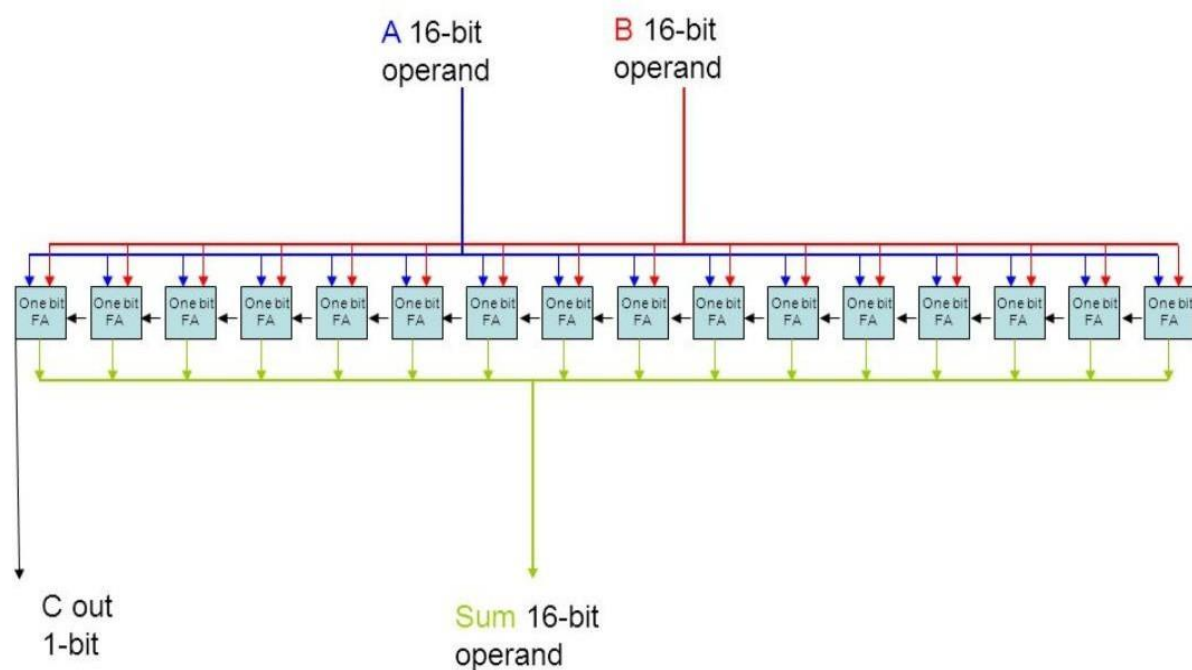
3.2 16 Bit Ripple Carry Adder

We already discussed the Ripple carry adder in chapter2.3

- Ripple Carry Adder is a combinational logic circuit.
- It is used for the purpose of adding two n-bit binary numbers.
- It is also called as n-bit parallel adder.

It consists of 16 full adders which are connected in cascaded form. Each full adder carry output is connected as an input carry to the next stage full adder. The input sequences are denoted by (A1 A16) and (B1 B16) and its relevant output sequence is denoted by (S.. S16).

The addition process in a 16-bit ripple-carry-adder is the same principle which is used in a 4-bit ripple-carry adder i.e., each bit from two input sequences are going to add along with input carry. This will use when the addition of two 16 bit binary digits sequence.



3.1.3

16 bit Carry Ripple Adder

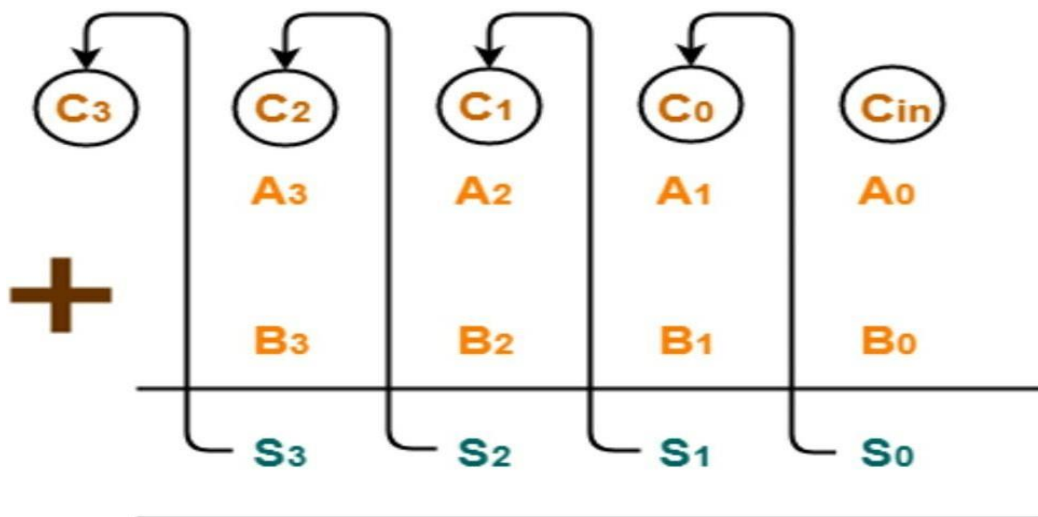
For 16 bit Ripple adder design we use the 4, 4bit Ripple carry adder this method help to reduce the area. Here we can see the LSB bits are A0, B0 and C0 where C0 is the input carry bit. Output of 1-bit full adder is S0 and C1. where S0= LSB sum bit and C1=Carry Out of first 1-bit full adder.

The boolean equation for 1 bit full adder output, S0 and C1, is given by,

$$S_0 = A_0 \oplus B_0 \oplus C_0$$

$$C_1 = (A_0 \oplus B_0) \cdot C_0 + A_0 B_0$$

pple Carry adder is built by cascading four 1-bit full adders in series. The out carry bit of every full adder is fed to the input of the next full adder. Now, as we have 4-bit ripple carry adder so it can further be cascaded in series to get higher bit adder. For example, we can get 16-bit Ripple Carry Adder by cascading in series four 4-bit ripple carry adders. While writing the Verilog code for 16-bit Ripple carry adder the same procedure is used. First the Verilog code for 1-bit full adder is written. From this, we can get the 4-bit ripple carry adder. Now, by using this 4-bit ripple carry adder 16-bit ripple carry adder Verilog code has been written. Binary addition for 4bit,



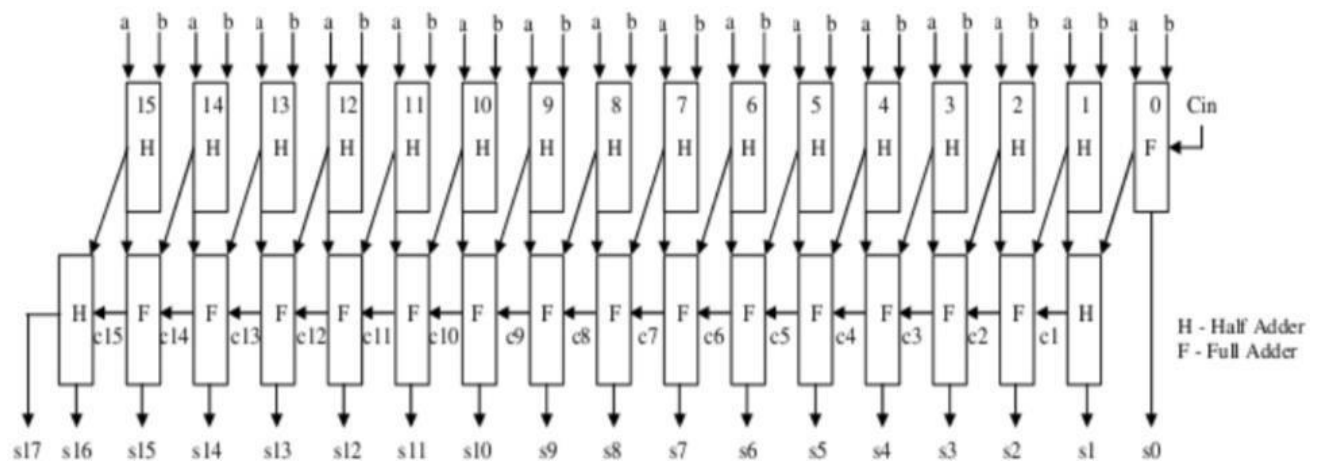
3.1.4 4bit addition

3.3 16 Bit Carry Save Adder

We already discuss about the Carry Save Adder in Chapter 2.5 For 16 bit Carry Save Adder we use the 4, 4bit Carry save Adder. The 4bit CSA adder partial sums and partial carry equations are,

$$Psi = a_i \text{ xor } b_i \text{ xor } c_i$$

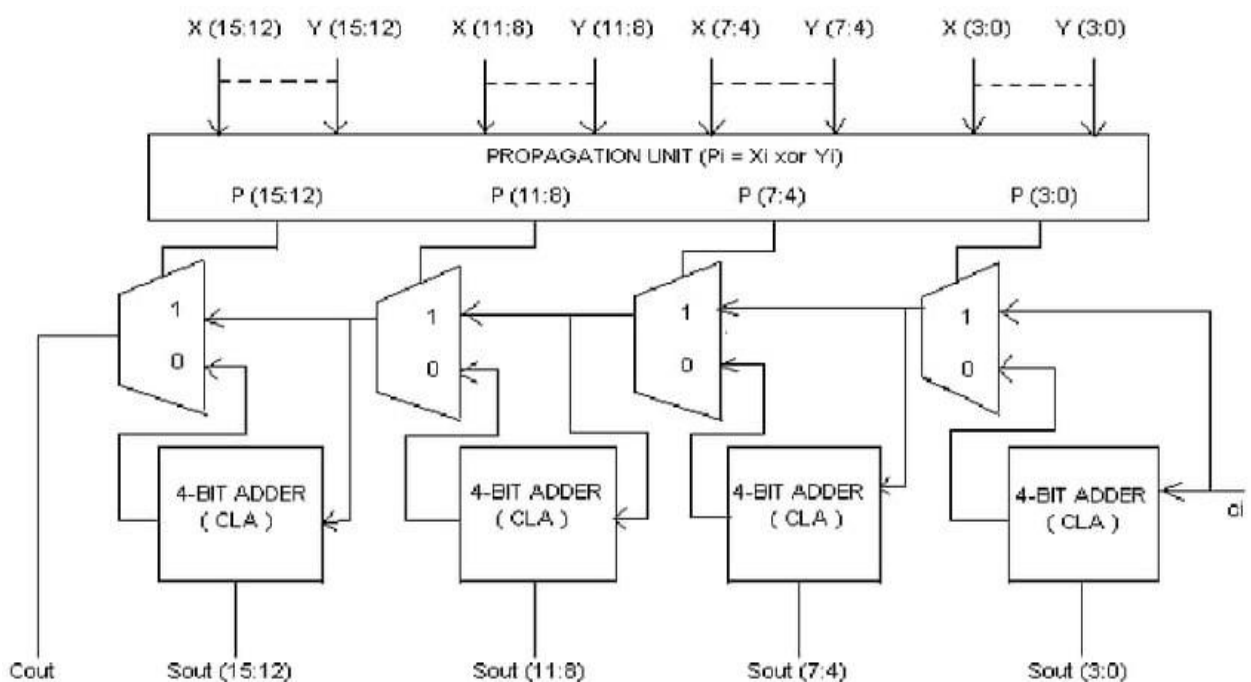
$$S_{ci} = (a_i \wedge b_i) \vee (a_i \wedge c_i) \vee (b_i \wedge c_i)$$



3.3 16 bit Carry Save Adder

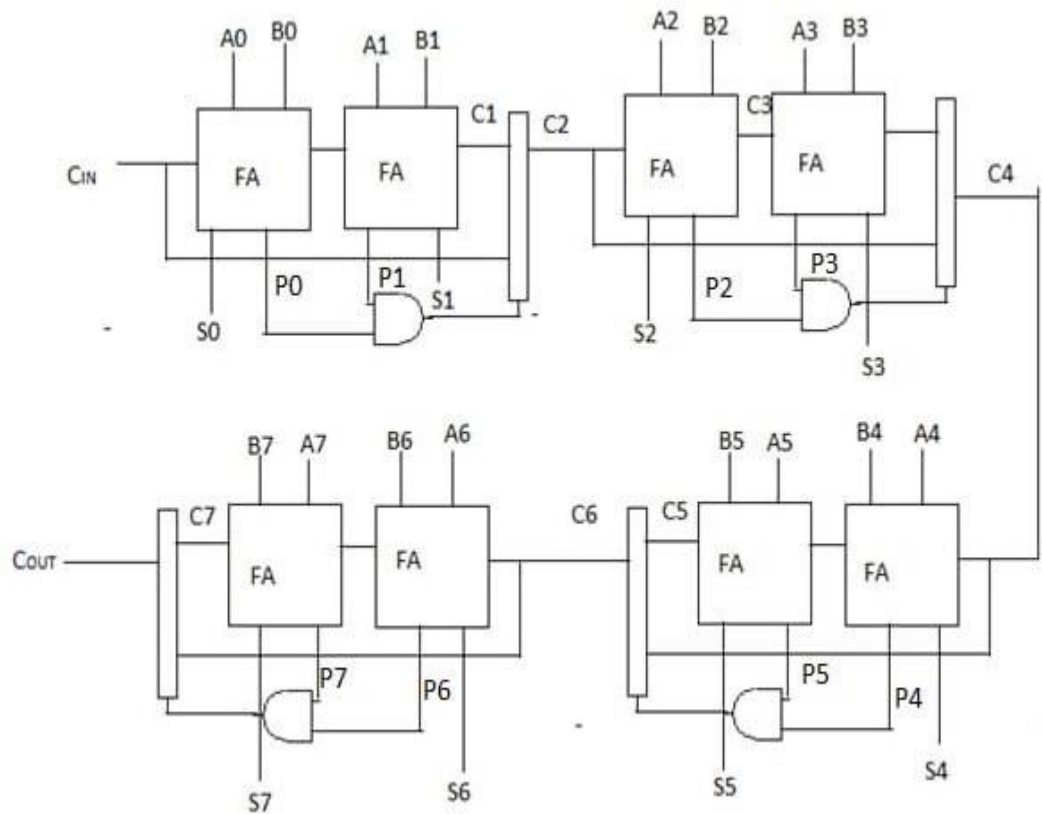
3.4 16 Bit Carry Skip Adder

We already discuss about the CSA adder in chapter 2.7 A carry skip adder (also known as a carry-bypass adder) is an adder implementation that improves on the delay of a ripple-carry adder with little effort compared to other adders. The improvement of the worst-case delay is achieved by using several carry-skip adders to form a block-carry-skip adder.

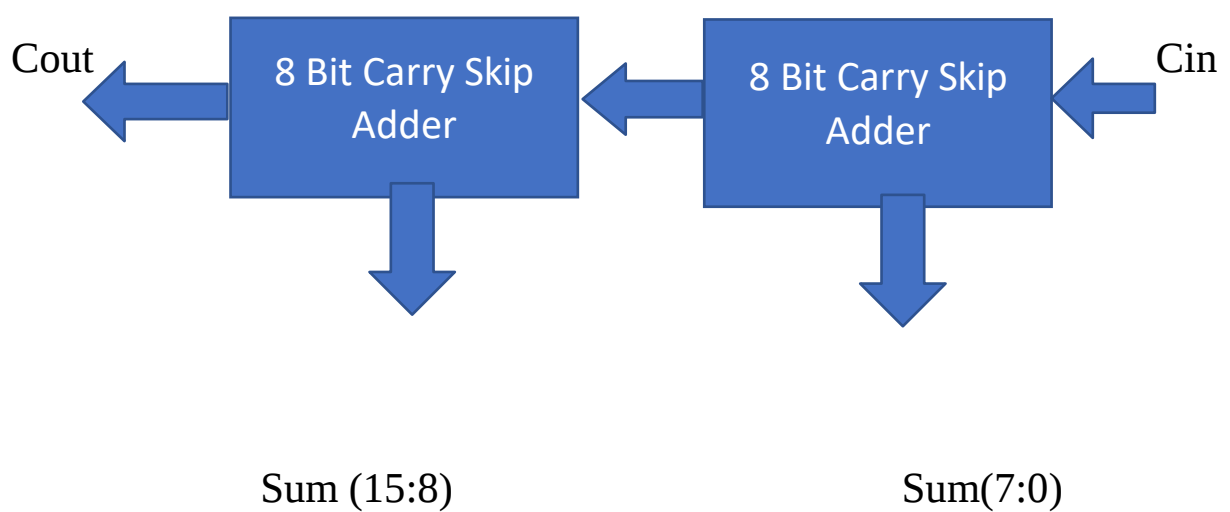


3.3.1 16 bit Carry Skip Adder

In this 3.4.1 design, we use the 2:1 mux and 4 bit Carry Lookahead adder for 16 bit Carry Skip Adder



3.3.2 8 bit Carry Skip Adder

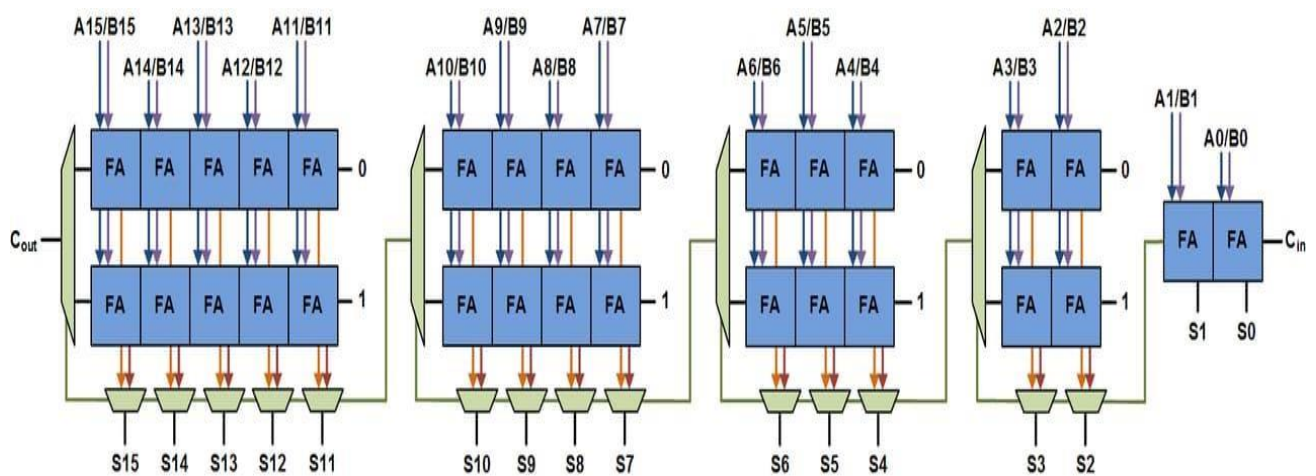


3.3.3 16 bit Carry Skip adder using 8 bit CSKA

In 3.4.3 design, 2-8bit Carry Skip adder used for 16 Bit Carry skip adder design, the 8 bit Carry skip adder design shown in figure.

3.4 16 Bit Carry Select adder

The Carry Select adder divide into 2 bit full adder, 2bit, 3bit, 4bit, 5bit Carry Skip Adders. In chapter 2.6 we have Information about Carry select Adder.

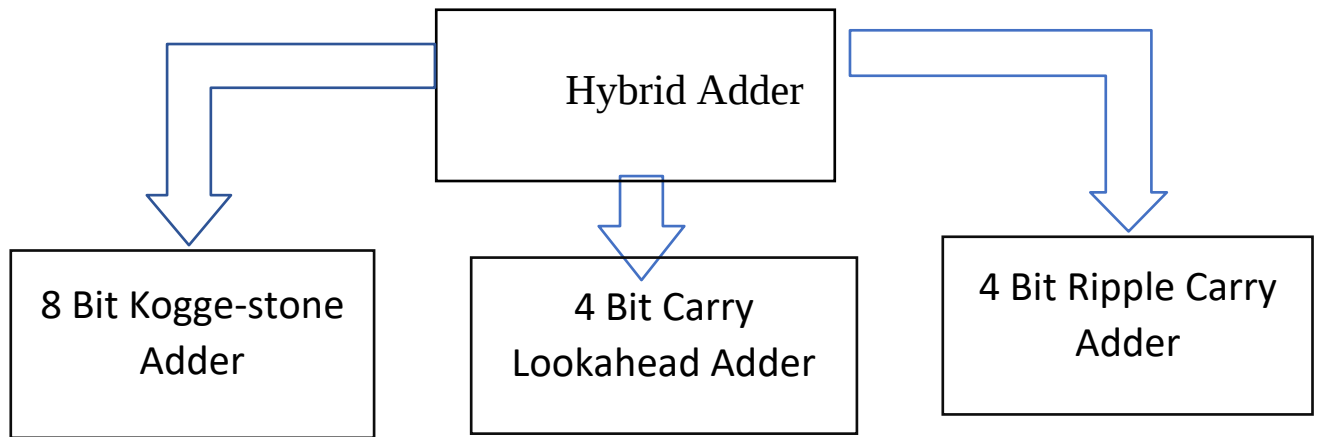


3.5 16 bit Carry Select adder

4 PROPOSED METHOD

4.1 Hybrid Adder

The Hybrid adder is a critical path design adder. The stage size of the adder varies from least significant bit (LSB) to the most significant bit (MSB). MSB have a different set of adder and LSB have a different set of adder. In this project, the Kogge stone adder used for MSB part and the Ripple Carry adder, Carry Lookahead adder used for LSB part.



4.1

Proposed Hybrid Adder

4.2 Kogge-stone Adder

The Kogge Stone has low logic depth, high node count, and minimal fan out. While a high node count implies a larger area, the low logic depth and minimal fanout allow faster performance. There are mainly three computational stages in Kogge Stone Adder. They are:

1. Pre-processing
2. Carry generation network
3. Post processing

Pre-processing Stage:

Pre-processing is the first stage where the generate and Propagate signals of all the input pairs of signals A and B are generated separately for each bit. The logical equations of the propagate and generate signals are given by the following equations:

$$P_i = A_i \text{ or } B_i \dots G_i = A_i \text{ and } B_i \dots$$

Carry Generation Stage:

Carry generation is the second stage of the KSA. At this stage the carries of all the bits are generated separately for each bit. They are divided into smaller pieces and this overall process is carried out in parallel for all the bits. Carry generate and Carry propagate bits are used as intermediate signals and their logical equations are given as follows:

$$CP(i:j) = P(i:k) + 1 \text{ and } P(k:j) \dots$$

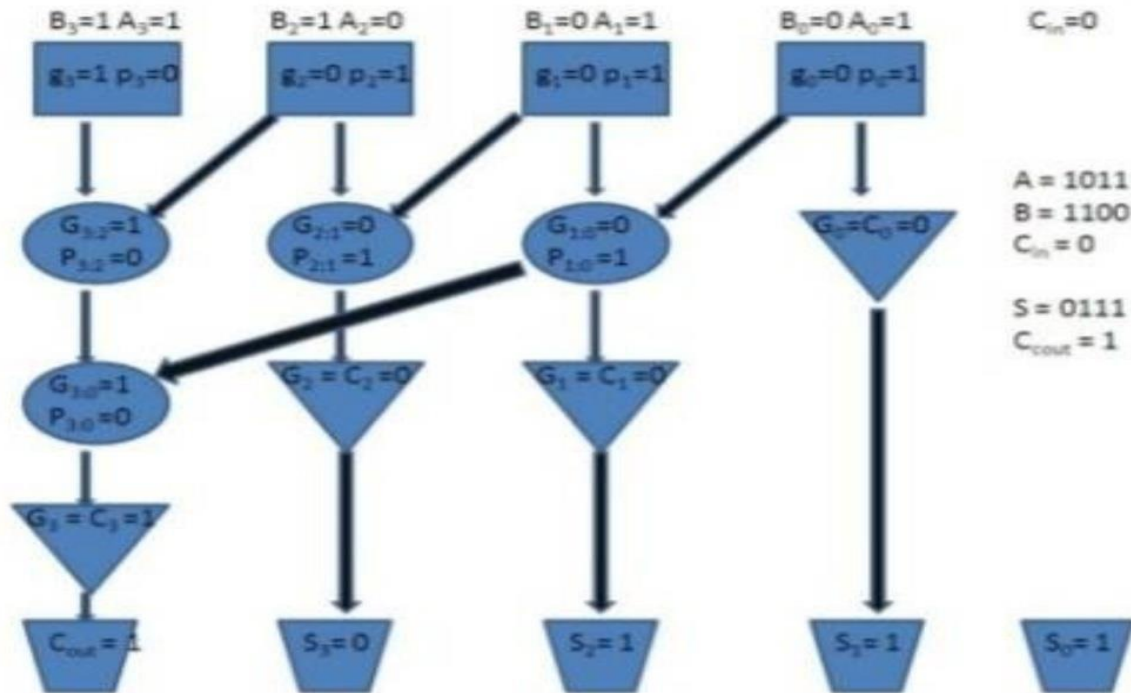
$$CG(i:j) = G(i:k) + 1 \text{ or } (P(i:k) + 1 \text{ and } G(k:j)) \dots$$

Post Processing Stage:

This is the final step or stage of the KSA which is common for all types of adders, i.e., calculation of summation of the bits given by the logical Equations are,

$$C_{i-1} = (P_i \text{ and } C_{in}) \text{ or } G_i \dots S_i = P_i \text{ or } C_{i-1} \dots$$

For 8 bit Koggestone Adder we use 2-4 bit Kogge stone Adder are Used.

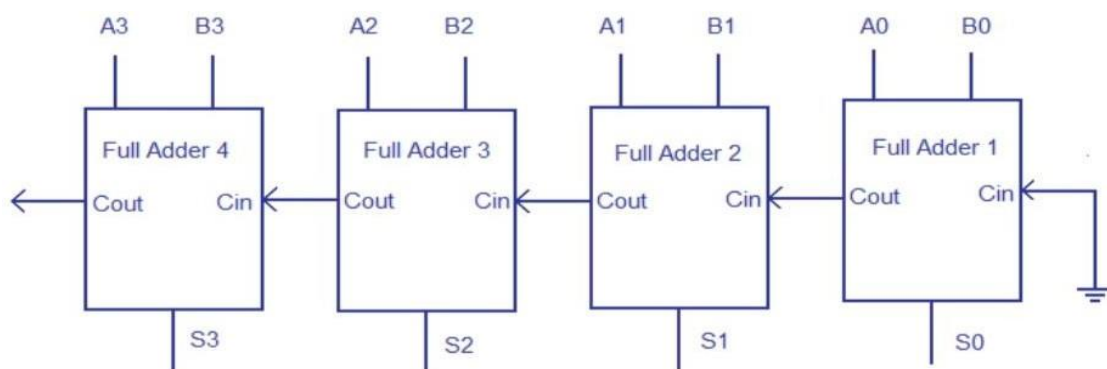


4.2

4 bit Kogge-Stone Adder

4.3 Bit Ripple Carry Adder

The Ripple Carry Adder is the Cascaded of single bit Full adder. The Carry in used for the 1st full adder and the carry out of the 1st full adder used as the 2nd full adder carry in same as the 3rd and 4th full adders.



4.3

4 bit Ripple Carry Adder

The 4 bit Ripple Carry adder design used for the first 4 bit of the LSB. For reducing the area purpose use the Single adder used for each bit operation.

4.4 Bit Carry lookahead Adder

For 4 bit Carry lookahead adder we use the 4 single bit Carry adder and Carry generation we use the equations,

$$C1 = G0 + P0.Cin$$

$$C2 = G1 + P1.C1 = G1 + P1.G0 + P1.P0.Cin$$

$$\begin{aligned} C3 &= G2 + P2.C2 = G2 + P2.G1 + P2.P1.G0 + P2.P1.P0.Cin \\ C4 &= G3 + P3.C3 \\ &= G3 + P3.G2 + P3.P2.G1 + P3.P2.P1.G0 + P3.P2.P1.P0.Cin \end{aligned}$$

4.5 Software Description

Intel Quartus Prime is programmable logic device design software produced by Intel; prior to Intel's acquisition of Altera the tool was called Altera Quartus Prime, earlier Altera Quartus II. Quartus Prime enables analysis and synthesis of HDL designs, which enables the developer to compile their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. Quartus Prime includes an implementation of VHDL and Verilog for hardware description, visual editing of logic circuits, and vector waveform simulation.

4.6 Design Entry

In Design Entry the first step is, you create the source file based on design objectives. After you can create your top level design file using a Hardware Description Language(HDL), such as Very High Speed Integrated Circuit Hardware Description Language(VHDL) or Verilog.

4.7 Synthesis

After design entry we have a optional simulation, we run the synthesis. During this step, VHDL, Verilog or mixed language design become netlist files that are accepted as input to the implementation step.

4.8 Verification

we can verify the functionality of your design at several points in the design flow. We can use simulator to verify the functionality and timing of the design portion of your design. The simulator interprets VHDL or Verilog code into circuit functionality and displays logical results of the described HDL to determine correct circuit operation. Simulation allows you to create and verify complex functions in a relatively small amount time. You can also run in circuit verification after programming your device.

5.RESULTS AND DISCUSSION

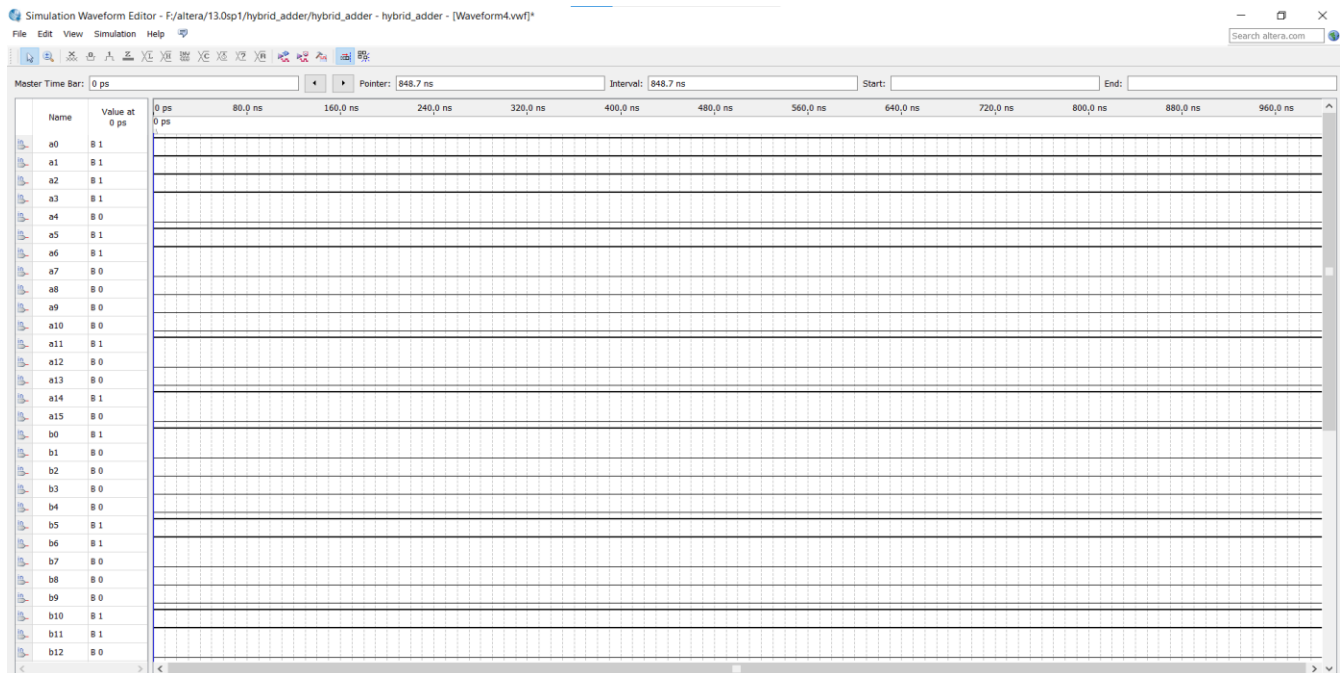
Explanation

The hybrid adder design Constructed for 16 bit Binary Addition.
The 16 bit Addition performed by,

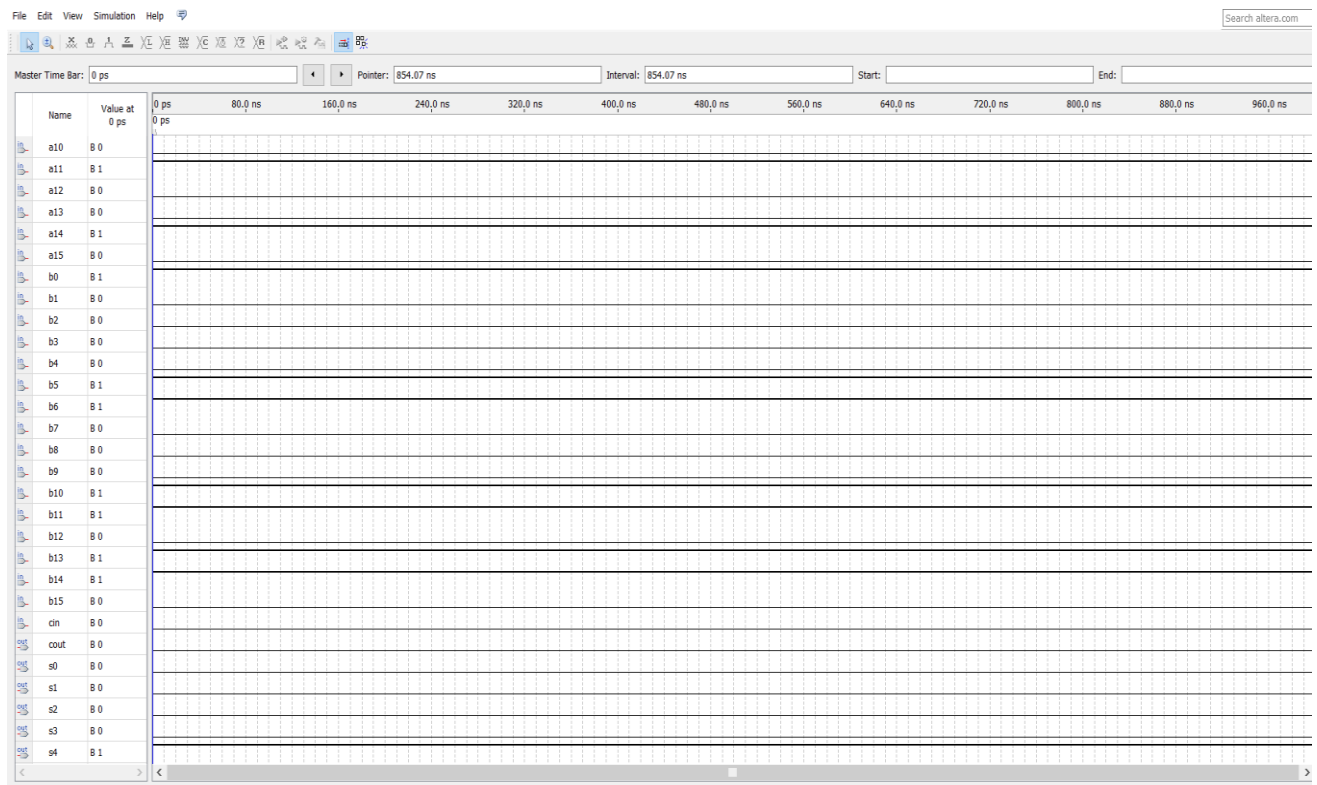
InputA (16 bit)	0100100001101111
InputB (16 bit)	0110110001100001
	1011010011010000

In this design the 16 bit part split into 8 bit, 2-4bit part so, the operation performed by three separate parallel parts.

5.1 Simulation Input



5.2 Simulation Output:



- 16bit Ripple carry adder minimum delay 3.696
- Proposed Hybrid adder minimum delay 3.431

Future Reference

This design provides, maximum delay 8.776 and minimum delay 3.431. If add the control unit we perform the 4bit, 8 bit, 12 bit, 16bit operations. Also, we use this design in Low power energy efficient Multiplier applications. It mainly designed for the ALU Design Purpose for future.

First Part - Ripple Carry Adder:

Carry (111) →

Input A (First 4bit)	1111
Input B (First 4bit)	0001
	10000
From the Output	10000 the 1 is Carry Out.

Second Part - Carry Lookahead Adder:

Carry (111) →

Input A (Next 4bit)	0
	110
Input B (Next 4bit)	0
	1101

In this, the Carry 111 the first carry 1 is Carry out of the First part Ripple Carry adder.

Third Part - 8 bit Kogge-Stone Adder:

Input A (last 8 bit) 01001000

Input B (last 8 bit) 01101100

10110100

Finally, all the 3 parts Output Combined into 16 bit output

1011010011010000.

6 CONCLUSION

The 16 bit Hybrid Adder is Proposed in this report. The Proposed Hybrid adder design have different Parallel Prefix sets. In this method we use the Kogge-Stone adder it is found to be effective in terms of delay than Carry select adder, Ripple Carry Adder, Carry Lookahead adder in the ALU Design. In this design we only improve the Minimum delay. In future we increase the Speed according to reduce the delay in the ALU design.

Department of Electronics and Communication Engineering

Vision

To be recognized by the society at large as a full- fledged department, offering quality higher education in the Electronics and Communication Engineering field with research focus catering to the needs of the stakeholders and staying in tune with the advancing technological revolution and cultural changes.

Mission

To achieve the vision, the department will

- Establish a unique learning environment to enable the students to face the challenges in Electronics and Communication Engineering field.
- Promote the establishment of centres of excellence in niche technology areas to nurture the spirit of innovation and creativity among faculty and students.
- Provide ethical and value-based education by promoting activities addressing the societal needs.
- Enable students to develop skills to solve complex technological problems and provide a framework for promoting collaborative and multidisciplinary activities.

