Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

Implementing Yosys & OpenROAD for Physical Design (PD) of an IoT Device for Vehicle Detection via ASAP7 PDK

S. H. Rakib [1,*], ***S. N. Biswas*** [2]

[1]*Physical Design Engineer, Neural Semiconductor Limited, Dhaka, Bangladesh.*

[2]*Department of Electrical and Electronic Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.*

Article Info

Article History:
Received 12 December 2024
Reviewed 10 February 2025
Revised 07 March 2025
Accepted 16 March 2025

Keywords:
VLSI
IoT
Synthesis
Physical Implementation
Setup and Hold Timing

*Corresponding Author's Email Address:
s.h.rakib.153@gmail.com

Abstract

Background and Objectives: The automobile industry is becoming more technologically advanced. Modern vehicles are expensive, but they have cutting-edge security features. As a result, the average individual who can afford low-end vehicles must forego the latest improvements, such as greater safety. Therefore, the main goal was to create a small Internet of Things device that could be used on a mobile device to notify the user when a car comes from the opposite direction. It will promote human safety by alerting users to that vehicle. The preparation, integration, and deployment of a modern IoT-based vehicle detection device have been described in this work. From there, it goes through the OpenROAD toolchain, and OpenSTA is used for static timing analysis (STA) and the ASAP7 PDK is used for the design. In this paper, provide a performance evaluation study across all three metrics (power, performance and area), as well as the entire design flow from hardware description to final implementation.
Methods: The entire behavioral level code goes through several stages before reaching a physical perspective, where various tools are used for multiple tasks. To obtain the desired physical-level architecture, first, use a tool to obtain the netlist file, including every G-cell map with a PDK-specific gate-level representation. Then, several stages will be followed to get the device's physical view.
Results: Throughout the entire experiment, the transition from RTL to GDSII was successfully achieved. Once the complete design is finished, the area, power, and timings all appear fine. Another unique characteristic is that the chip employed 7nm technology. The 5 GHz frequency was attained when the chip functioned flawlessly without DRC or any connection problems, timing, or DRV violations. Less than 1 percent is the maximum allowable IR loss maintained. Over 80% of the total space was utilized effectively.
Conclusion: To build an IoT gadget manufacturable with the best PPA, the general experiment was to write RTL code and proceed to the tap-out stage. The experiment achieved the best result by utilizing open-source chip design tools. Additionally, there are no DRC violations, timing problems, or power loss.

This work is distributed under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

Introduction

It was Kevin Ashton who first introduced the Internet of Things or IoT. All of the physical objects in the environment that communicates with one another via the internet is referred to as IoT systems [1].

According to several industry projections, there will be approximately 50 billion smart devices linked to the Internet of Things (IoT) by 2030. These devices will aid in the development of novel solutions for issues affecting society as a whole, including telemetry, healthcare, home automation, energy conservation, security, wearable computing, asset tracking, public infrastructure maintenance, etc. [2].

The Association for Safe International Road Travel (ASIRT) estimates that 20–50 million people are injured or incapacitated and that approximately 1.3 million people die in traffic accidents annually. Worldwide, traffic accidents cost $518 billion, or 1% to 2% of each nation's yearly GDP [3]. As one of the biggest causes of death, road accidents claim 1.3 million lives annually. Therefore, vehicle detection technology may be a useful way to lower traffic accidents and improve public safety [4].

This IoT device was introduced to assist in decreasing accidents by warning drivers in advance. However, this gadget will be upgraded in the future to include the capability of providing a prompt alert to the nearby rescue team and ensuring prompt treatment to save the life of the passenger in the event of an accident.

The Internet of Things (IoT) is a computer process in which every physical thing has sensors, microcontrollers, and transceivers for enabling communication. It also has proper protocol stacks built in to enable the objects to communicate with one another and with users [5].

Various techniques can be employed for vehicle detection. When it comes to reducing accidents or promoting automated vehicles, this detecting procedure is essential. The statistical approach was one of the strategies. Statistical techniques effectively integrate activity data with sophisticated classify arithmetic knowledge about vehicle flight patterns. This method effectively locates and tracks the cars [6].

On numerous fronts, open-source EDA is quickly facilitating new waves of innovation. It expedites the scientific process and makes study findings applicable to contemporary business practices, according to academic scholars. Open-source EDA is a supplement and enhancer of commercial EDA for EDA experts and the industry ecosystem [7].

This device is based on real-time data capturing and is MCU-powered. OpenROAD flow and the Yosys, OpenROAD, and OpenSTA tools are utilized in the design of this [8]. A fully autonomous RTL-GDSII flow for quick architectural and design space exploration, early QoR prediction, and thorough physical design implementation is called OpenROAD-flow-scripts (ORFS) [9].

Select open-source tools because they are user-friendly, widely available, and compliant with industry standards. This project's complete ASIC flow is shown next.

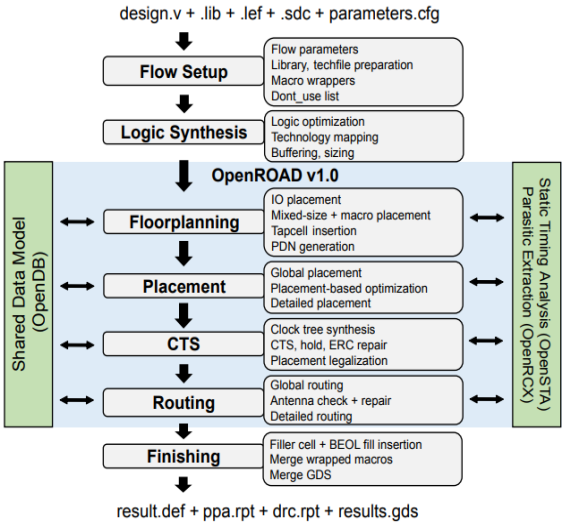


Fig. 1: An overview of the OpenROAD tool's implementation of the OpenROAD design flow.

The design flow from RTL to the final physical design using OpenROAD, help understand the whole design process covered in the study.

In the summer of 2020, OpenROAD fulfilled many "proof points" enabling the automated construction of a manufacturing layout in TSMC 65LP and GLOBALFOUNDRIES 12LP technologies, including a 12nm SOC tape-in. These "proof points" included passing all physical verification checks as well as electrical and timing correctness checks [10].

The selection of Yosys was based on its wide range of features, broad support for Verilog-2005, and ability to map to any standard cell library used in ASICs. The goal of OpenROAD usage in this design is to lower the obstacles that now prevent designers from implementing innovative technologies on hardware, including those related to cost, skill, and unpredictability [11].

A detailed analysis of power, time, and area wraps up the whole flow. Make use of 7nm PDK. To ensure that the design operates flawlessly, choose the best achievable frequency. And appropriately set all the restraints. The Objectives Provide a comprehensive system design and support the deployment of high-performing IoT endpoints for traffic safety.

Technology is getting smaller day by day. Additionally, since smaller technology uses less space, it can accommodate a larger, more complicated design in a smaller space. And because of its tiny size, the design uses less power overall. That's why 7nm PDK was selected for the design, as it was the smallest technology available on OpenRoad Flow [12].

ASAP7 is an Advanced-Node Research PDK that is open-source. OpenROAD has also made the ASAP7 advanced-node research PDK from Arizona State University publicly available to supplement the SKY130 open-source manufacturable PDK [13]. Advanced

patterning technologies and scaling boosters (single diffusion break, contact-over-active-gate, dense crossovers, etc.) are reflected in the design principles of this incredibly realistic PDK [10].

First, strive to reach the optimal frequency that can accept the design without any violations. Power consumption is also given significant consideration, as it represents a major challenge for Internet of Things (IoT) devices. Since most IoT devices are portable charging systems that do not connect directly to a charging system, the primary goal when designing IoT devices is to minimize power consumption and maximize speed while maintaining functionality.

The flow's default units are Time: 1ps; capacitance: 1fF; voltage: 1v; power: 1pW; distance: 1um.

Device Structural Overview

Based on information from the neighboring mobile network about whether or not any cars are approaching me, the MCU in this device manages network connectivity, sensor data, and the generation of the clock, reset, and data input signals for the IoT device. An LED display cannot display unwanted data until a reset signal is received. A clock signal is required for the synchronous operation of an Internet of Things device. Data from the MCU is saved by an IoT device, which also manages system operation by displaying it on an LED display. Manage the display_enable switch as well.

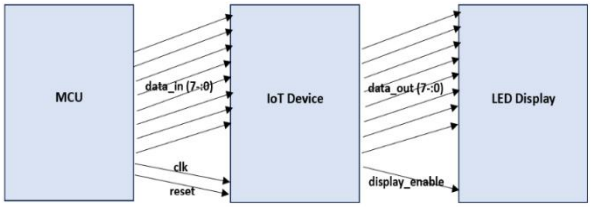


Fig. 2: Structural overview of the device, highlighting its key components and their interconnections.

The architecture is described before the detailed design begins which shows the structure of the IoT device also how the components are connected.

The entire design flow is divided into three sections:

- RTL Design
- RTL to Gate Level Netlist Conversion
- Physical Implementation

The SDC file contains design constraints that are specified following the design's operating frequency of 5GHz. As an example, the clock uncertainty is set to 20% of the clock period, the IO delay is set to 30%, and the maximum transition for both the data path and the clock path is set to 10%. As opposed to the typical setting of 5 to 15% of the clock duration. It's probably set at 5 to 10 for the clock path and 10 to 15% for the data paths [14]. The transition time of a CMOS gate is known to have a significant impact on its performance, including its propagation delay time and short circuit power dissipation [15].

Verilog Module

For the IoT gadget, here is the code for functional work.

```
module iot_device (
    input wire      clk,          // Clock signal
    input wire      reset,        // Reset signal
    input wire [7:0] data_in,     // Input data from network/microcontroller
    output reg [7:0] data_out,    // Output data to display
    output reg      display_enable // Enable signal for display
);

    // State register
    reg [7:0] internal_data;

    // Control logic
    always @(posedge clk or posedge reset) begin
        if (reset) begin
            internal_data   <= 8'b0;
            data_out        <= 8'b0;
            display_enable  <= 1'b0;
        end else begin
            internal_data   <= data_in;
            data_out        <= internal_data;
            display_enable  <= 1'b1;  // Enable display when not in reset
        end
    end

endmodule
```

Fig. 3: Verilog code for implemented IoT device.

Physical design starts with the Verilog code for the IoT device, which represents the logic of the device.

Synthesis with Yosys

Yosys, a platform for managing and storing data, was first used to synthesize the Verilog of the Hardware device (IoT) into a gate-level netlist [16]. Additionally, Yosys provides us with a glimpse of the device schematic.

Following synthesis using the ASAP7 PDK and this is the Yosys report.

```
Number of wires:                 32
Number of wire bits:             46
Number of public wires:          13
Number of public wire bits:      27
Number of ports:                  5
Number of port bits:             19
Number of memories:               0
Number of memory bits:            0
Number of processes:              0
Number of cells:                 36
  DFFASRHQNx1_ASAP7_75t_R        17
  INVx3_ASAP7_75t_R              18
  TIEHIx1_ASAP7_75t_R             1

Chip area for module '\iot_device': 7.800300
  of which used for sequential elements: 6.444360 (82.62%)
```

Fig. 4: Report of gate level netlist.

Physical Design with OpenROAD

For getting a concrete view of the design; several physical implementation phases need to be run each having different target and tasks corresponding the gate-level netlist and generated SDC from Yosys output.

A. Floor Planning

The overall goal of floor planning is to reduce the design's length and overall area [17]. The chip's area and shape must be determined at this point. Therefore, there

needs to be a few basic activities carried out, such as setting the block's core area, die area, utilization, and aspect ratio as well as positioning the physical cell, macros, and IO port on the appropriate edge.

According to this design, the floorplan should begin at 50% utilization and terminate at 55%. However, a 75% utilization rate is initially adopted in real projects [18].

For this design, the die area and core areas are set at (0.0 0.0 4.5 4.5) and (0.108 0.27 4.374 4.32). Port placement is done using Metals 4 and 5, respectively, for the vertical and horizontal layers.

B. Power Planning

An integrated circuit system's physical PDN extends its hierarchy across multiple tiers. The voltage regulator module (VRM), which is essentially a DC-DC converter that raises input DC voltage to the nominal supply voltage level as needed by the IC chip, provides power to the network [19].

In order to allocate power to every component in the design—including standard cells and macros—power planning is required.

```
add_pdn_stripe -grid {top} -layer {M1} -width {0.018} -pitch {0.54} -offset {0} -followpins
add_pdn_stripe -grid {top} -layer {M2} -width {0.018} -pitch {0.54} -offset {0}
add_pdn_stripe -grid {top} -layer {M5} -width {0.12} -spacing {0.072} -pitch {0.75} -offset {0.13}
```

The top layer in this block is made of metal 5, the intermediate layer is made of metal 2, and the special route is made of Metal1.

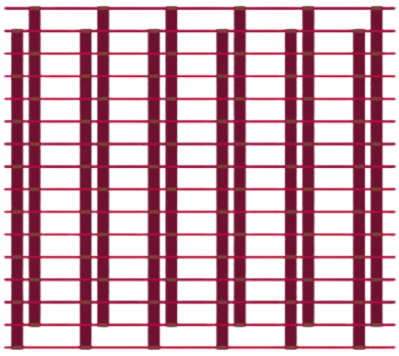


Fig. 5: Illustration of the correct power grid distribution to ensure proper power delivery to the cell.

The power grid distribution sends adequate power to all regions of the design, which is an important factor to take into account for the device's reliability.

C. Placement

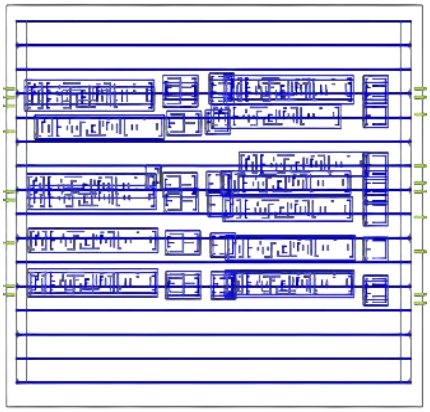
The goal of placement is to handle optimization goals like HPWL, routed wire length, time, power, routing, etc. while figuring out where to put instances (such as standard cells and macros) [20].

Some problems can be resolved during the placement phase; for example, hard blockage can be used to reduce notch congestion, partial blockage can be used to address issues with cell density, and padding can be used to address issues with pin density [21]. Additionally, it is necessary to verify utilization; if it deviates significantly from the floorplan utilization, debug it to determine the cause. After the placement stage, the DRV and setup time must also be checked.

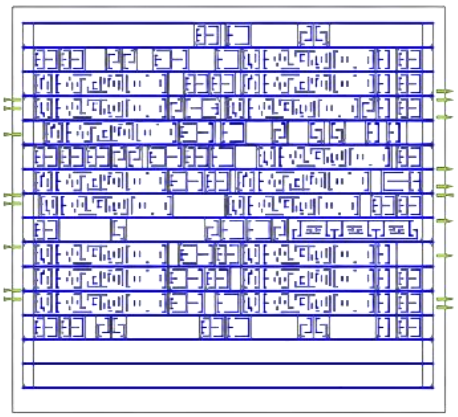
Proper placement of the standard cell will reduce the likelihood of timing violations, power outages, or physical violations. Proper placement in the core area will facilitate easier connections between ports and cells, and if the cell is placed in a scattered configuration, power outages will not arise during design. Also, in the absence of congestion, there won't be any problems with a lack of tracks, which was one of the causes of the short violation. Finished the placement stage since it has a big impact on the design and doesn't throw off the sign-off process.

The placement process consists of two basic steps: global placement and detailed placement [22].

The first tool uses global placement, which places cells that are genuinely existent in the netlist without adhering to any legality rules when doing so. Not positioned correctly in the row either. There are also overlaps between them. Subsequently, the tool places each cell precisely, adhering to the placement guidelines, so that they don't overlap, and placing them in between rows.



(a)

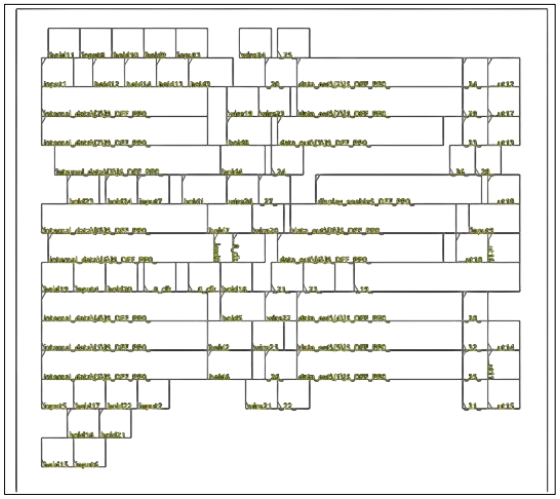


(b)

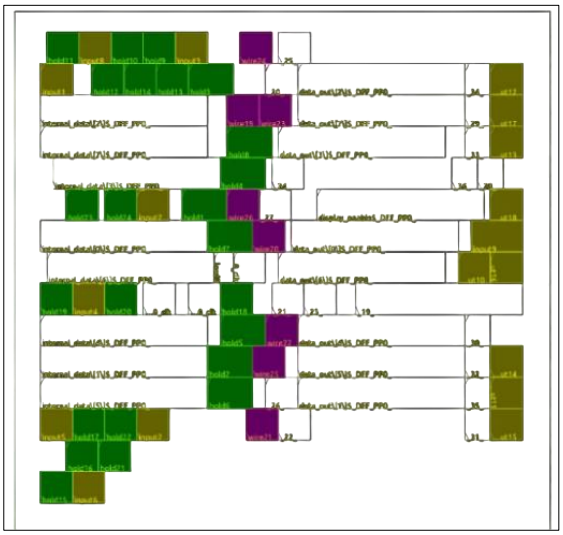
Fig. 6: (a) Incorrect cell placement after global placement, showing overlapping cells and misalignment. (b) Corrected detailed placement, ensuring proper cell alignment within designated rows.

A comparison of incorrect and corrected cell placements highlights the importance of proper cell placement for efficient routing and performance.

Place every cell closer together at this placement stage in order to fulfill the timing requirement as well. Then, by resizing the cell, optimization is carried out to decrease the slack.



(a)



(b)

Fig. 7: a) place all cells in the design listed in the netlist file; b) highlighting the cells that were optimized to meet the timing constraints. (Shown in violet, yellow, and green).

The placement of all cells and optimized cells for timing shows how cell placement affects timing performance.

Table 1: The report following placement details are included in the table

Parameter	Value
Instances	117
Nets	97
Core Area	17.277 um^2
Place Instances Area	11.474 um^2
Utilization	69.96%

D. Clock Tree Synthesis

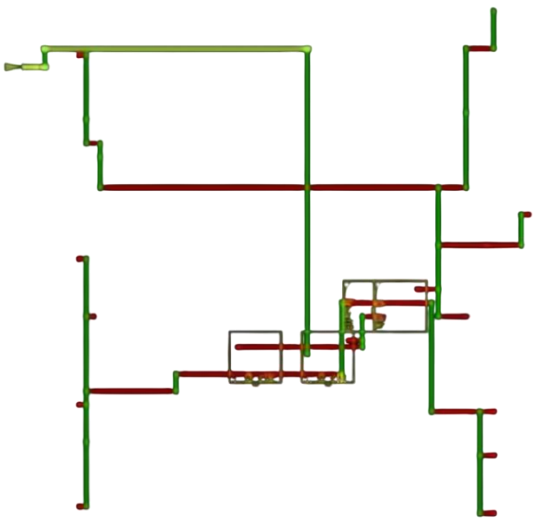
The movement and processing of data inside a chip are coordinated and controlled by a clock signal [23]. The clock port to every clock pin on the flop is connected by a clock tree.

A clock signal on the flop is required for design operation. Therefore, the tree must be balanced because all of the flops may run simultaneously. Minimize the clock skew to balance the tree. The time disparity between the arrival times of the clock signal at each different flip is known as clock skew. Use an inverter or buffer at this point to balance the clock tree.

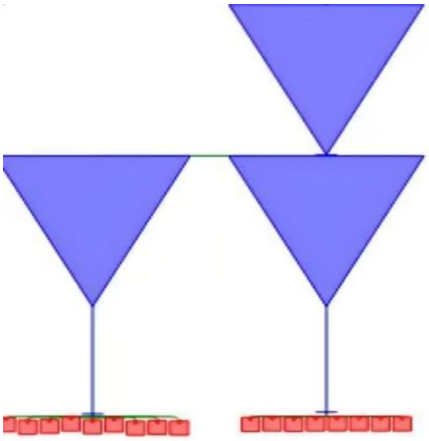
Since the clock is the most important component of a synchronous system design, all flops must be synchronized for the design to operate flawlessly and shift and store any data [24]. CTS buffers or inverters are used to balance clock trees since they require a clock that is always on. If the clock path's wire width is not increased, cross-talk will result. Consequently, use NDR on the clock net. When creating clock trees, use the CTS buffer and inverter cell since their rise and fall times are the same. In addition, the wire length is affected by the location of the Flop during the installation step. It also impacts the setup and hold times.

The addition of positive clock skew helps with design setup violations by increasing the time it takes for data to get from the launch flop to the capture flop [25]. On the other hand, skew added to the data would impact the difficult need for a more stable time and might result in design metastability, which will complicate the hold analysis. A cluster group is made up of virtually identical sorts of insertion delay flops.

The software also makes an effort to create cluster groups based on the given skew and utilizes clustering to aid in tree balancing.



(a)



(b)

Fig. 8: Illustrations of: a) A clock port on the left boundary connecting all sequential cells b) A debugger window identifying problematic sequential cell groups with similar skew for timing issue resolution.

It is essential to fix timing issues, and the port clocks connecting sequential cells, and the debugger indicating timing concerns, play a crucial role in this scenario.

17 DFF flip-flops are linked to the clock port, as seen in the above image. Additionally, it can be seen that the design includes 17 DFF from the synthesis report.

E. Routing

Routing is the technique of creating a physical link between each instance by utilizing the metal layer to connect all of the instances with ports or instance to instance.

Prior to estimating the parasitic value, the tool completed global routing. In order to maintain the continuity of the N-well, a filler cell is also placed on this step. Once placed DEF is inserted, the tool reads LEF.

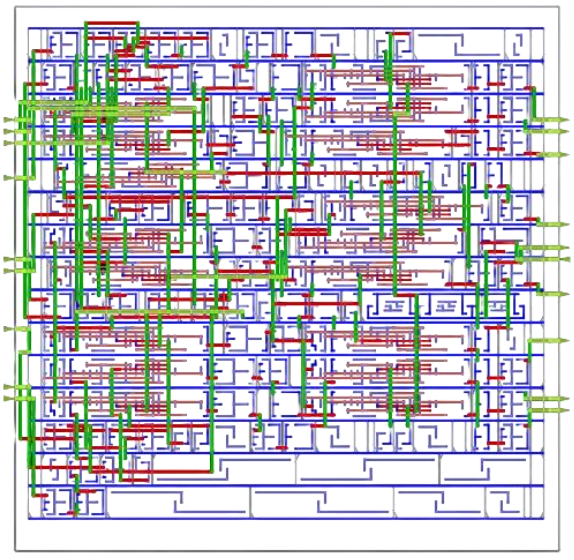


Fig. 9: Signal routing indicating physical connections between all the communicating objects within the design.

Signal routing also has the critical task of routing the signals between all the object in the design such that they are connected correctly in the end layout.

In order to accomplish global routing, it defines global routing cells, or gcells, and minimizes wire length and vias while limiting congestion and overflow within the cells [9].

Afterward, the tool finishes the detailed routing as efficiently as feasible, making use of the routing truck and being aware of the basic DRC rule of the design.

Result and Discussion

There are three primary (PPA) focuses in this study. First is performance, often known as timing, in a VLSI digital design. Next are power and area. The battery lifetime and manufacturing cost will be impacted by increased power and area. However, if the timing is right, the device will function as intended.

Moreover, the design will cease to function if the timing is violated. Primarily focused on using this concept to solve the time issue. Furthermore, this design must not have any time violations or negative slack. Contrasted with industry-level comparisons. With less than 1% IR loss, area utilization is also about 80% without filler, according to the industry level.

The tool checked the design rules after finishing the detailed route, attempted to resolve any violations, and produced a design with the fewest possible timing and physical violations. Next comes the sign-off check, which includes a final, physical verification check, RC parasitic check, sign-off time check, and power analysis. The design may then be tapped out for manufacturing.

In the final design stage, there are 476 single-cut vias and no DRC, connectivity, or DRV violations are present.

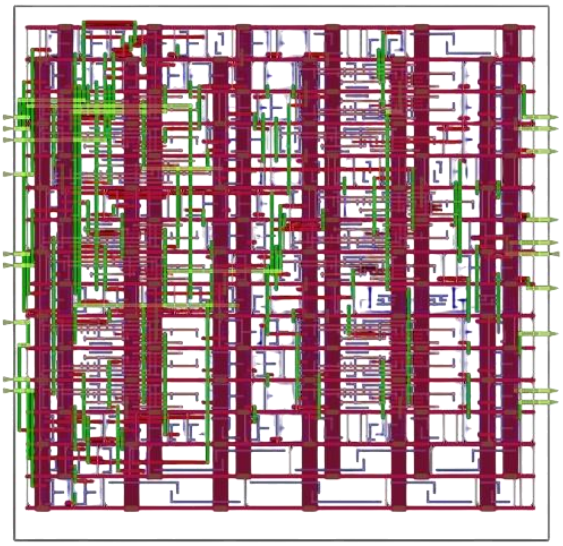


Fig. 10: Design final physical layout — completed place and route after optimizations.

The final physical layout, following optimization, is

given to show that the design process was accomplished effectively. It displays the results of the placement, routing, and optimization processes, indicating that the design fits the requirements and is ready for manufacture. This value is important because it demonstrates the efficacy of the design process and the overall operation of the IoT device.

The overall design area is 14 um2 with an aspect ratio of 1, of which 80 percent is eventually used. The overall locations of all 193 cells. There are 476 single-cut vias in the final design stage, and there are no DRC, connection, or DRV violations.

Table 2: Total number of cells used in the final designed layout

Cell Type	Count
Filler	56
Tap	30
Tie	18
Clock Buffer	3
Timing repair buffer	50
Inverter	18
Clock Inverter	1
Sequential cell	17
Total	193

Timing is a more important component of design as it affects functionality if it is done incorrectly. Thus, there shouldn't be any timing violation in the design. Timing violations mostly fall into two categories: setup and hold. The clock period mostly determines setup. The equations for Setup and Hold violation check:

$$Tc2q + Tcomb + Tsetup \leq Tclk + Tskew \quad (1)$$

$$Tc2q + Tcomb \geq Thold + Tskew \quad (2)$$

The setup analysis uses the worst data path delay if the path delay is maximal since the chip will pass other setup analyses if it operates with the largest delay

Timing analysis is done with OpenSTA, a tool that is also integrated with OpenROAD flow scripts. Also, KLayout is utilized for Physical Verification Checks.

Tool performed hold analysis using best case as in best scenario there is minimal data path delay and if chip functions here flawlessly, it will function for other analysis views [26].

Table 3: Final Design Timing Report

	Required Time	Arrival Time	Slack
Setup	100	92.78	7.22
Hold	84.6	94.22	9.62

The final complete timing report for the setup and hold design, including cell and net delay, is provided [26]. WNS and TNS are both greater than 0 because this design does not have any negative slack; for setup and hold analysis, these are 7.32 ps and 9.62 ps, respectively. Negative slack causes functional mismatches, whereas positive slack affects overall performance with regard to power but has no influence on device functionality. More positive slack will cause the device to consume more power.

For this design, an IEEE 1481-1999 SPEF file was also created. By supplying the parasitic values of each net, the SPEF file aids the STA tool in accurately calculating the delay [27].

```
=============================================================================
finish report_checks -path_delay max
-----------------------------------------------------------------------------
Startpoint: data_out[0]$_DFF_PP0_
          (rising edge-triggered flip-flop clocked by clk)
Endpoint: data_out[0] (output port clocked by clk)
Path Group: clk
Path Type: max

Fanout     Cap    Slew   Delay    Time   Description
---------------------------------------------------------------------------
                          0.00    0.00   clock clk (rise edge)
                          0.00    0.00   clock source latency
     1    1.01    0.00    0.00    0.00 ^ clk (in)
                                         clk (net)
                0.29    0.09    0.09 ^ clkbuf_0_clk/A (BUFx2_ASAP7_75t_R)
     2    1.28    7.04   11.10   11.19 ^ clkbuf_0_clk/Y (BUFx2_ASAP7_75t_R)
                                         clknet_0_clk (net)
                7.05    0.07   11.26 ^ clkbuf_1_1_0_clk/A (BUFx2_ASAP7_75t_R)
     9    5.32   19.29   19.03   30.30 ^ clkbuf_1_1_0_clk/Y (BUFx2_ASAP7_75t_R)
                                         clknet_1_1_0_clk (net)
               19.30    0.28   30.57 ^ data_out[0]$_DFF_PP0_/CLK (DFFASRHQNx1_ASAP7_75t_R)
     1    1.39   18.68   44.24   74.81 v data_out[0]$_DFF_PP0_/QN (DFFASRHQNx1_ASAP7_75t_R)
                                         _16_ (net)
               18.68    0.11   74.92 v _36_/A (INVx2_ASAP7_75t_R)
     1    0.68    7.16    6.45   81.37 ^ _36_/Y (INVx2_ASAP7_75t_R)
                                         net18 (net)
                7.16    0.06   81.43 ^ output10/A (BUFx2_ASAP7_75t_R)
     1    0.10    3.82   11.34   92.78 ^ output10/Y (BUFx2_ASAP7_75t_R)
                                         data_out[0] (net)
                3.82    0.00   92.78 ^ data_out[0] (out)
                                92.78   data arrival time

                          200.00  200.00   clock clk (rise edge)
                            0.00  200.00   clock network delay (propagated)
                          -40.00  160.00   clock uncertainty
                            0.00  160.00   clock reconvergence pessimism
                          -60.00  100.00   output external delay
                                  100.00   data required time
---------------------------------------------------------------------------
                                  100.00   data required time
                                  -92.78   data arrival time
---------------------------------------------------------------------------
                                    7.22   slack (MET)
```

(a)

```
=============================================================================
finish report_checks -path_delay min
-----------------------------------------------------------------------------
Startpoint: data_in[5] (input port clocked by clk)
Endpoint: internal_data[5]$_DFF_PP0_
          (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: min

Fanout     Cap    Slew   Delay    Time   Description
---------------------------------------------------------------------------
                          0.00    0.00   clock clk (rise edge)
                          0.00    0.00   clock network delay (propagated)
                         60.00   60.00 v input external delay
     1    0.59    0.00    0.00   60.00 v data_in[5] (in)
                                         data_in[5] (net)
                0.09    0.03   60.03 v hold15/A (BUFx2_ASAP7_75t_R)
     1    0.46    4.59   10.28   70.31 v hold15/Y (BUFx2_ASAP7_75t_R)
                                         net59 (net)
                4.59    0.03   70.34 v input6/A (BUFx2_ASAP7_75t_R)
     1    0.49    4.66   11.79   82.14 v input6/Y (BUFx2_ASAP7_75t_R)
                                         net14 (net)
                4.66    0.04   82.17 v hold16/A (BUFx2_ASAP7_75t_R)
     1    0.61    4.92   12.00   94.18 v hold16/Y (BUFx2_ASAP7_75t_R)
                                         net60 (net)
                4.93    0.05   94.22 v internal_data[5]$_DFF_PP0_/D (DFFASRHQNx1_ASAP7_75t_R)
                                94.22   data arrival time


                          0.00    0.00   clock clk (rise edge)
                          0.00    0.00   clock source latency
     1    1.01    0.00    0.00    0.00 ^ clk (in)
                                         clk (net)
                0.29    0.09    0.09 ^ clkbuf_0_clk/A (BUFx2_ASAP7_75t_R)
     2    1.28    7.04   11.10   11.19 ^ clkbuf_0_clk/Y (BUFx2_ASAP7_75t_R)
                                         clknet_0_clk (net)
                7.05    0.08   11.27 ^ clkbuf_1_0_0_clk/A (BUFx2_ASAP7_75t_R)
     9    5.19   18.89   18.89   30.16 ^ clkbuf_1_0_0_clk/Y (BUFx2_ASAP7_75t_R)
                                         clknet_1_0_0_clk (net)
               18.91    0.36   30.52 ^ internal_data[5]$_DFF_PP0_/CLK (DFFASRHQNx1_ASAP7_75t_R)
                         40.00   70.52   clock uncertainty
                          0.00   70.52   clock reconvergence pessimism
                         14.09   84.60   library hold time
                                 84.60   data required time
---------------------------------------------------------------------------
                                 84.60   data required time
                                -94.22   data arrival time
---------------------------------------------------------------------------
                                  9.62   slack (MET)
```

(b)

Fig. 11: GBA report for: (a) Setup analysis and (b) Hold analysis, showing timing checks for signal synchronization.

The GBA report for setup and hold analysis is key for ensuring the design meets timing requirements.

Usually, the sum of the circuit's dynamic and static power results in the overall power dissipation.

$$P_{total}=P_{dynamic}+P_{leakage} \quad (3)$$

$$P_{dynamic}=P_{switching}+P_{short_circuit\ (internal)} \quad (4)$$

Dynamic and static power loss are the two basic forms that occur in designs.

Static power loss, often referred to as leakage power loss, and dynamic loss due to switching and short circuit loss. The design's switching activity is the primary determinant of switching power.

In reality, logic transitions from 0 to 1 and 1 to 0 in a design resulting in dynamic power [28]. Furthermore, short circuit power is mostly impacted by a cell's transition time. Here is the equation for Dynamic Power loss in a design.

$$P_{switch}=a\times f\times C_{load}\times V_{dd}^2 \quad (5)$$

$$P_{sc}=t_{sc}\times V_{dd}\times I_{peak} \quad (6)$$

Leakage power, however, is the most crucial component. Considering that loss of data happens while a chip or gadget is not in use. Therefore, this loss must be reduced [29]. A device's lifespan will be shortened otherwise.

Table 4: Breakdown of the total power in terms of switching, leakage, and internal power for each individual type of cell

	Internal Power (mW)	Switching Power (mW)	Leakage Power (mW)	Total Power (mW)
Sequential	0.0938	0.00258	1.68E-06	0.0964
Combinational	0.016	0.00728	2.92E-06	0.0232
Clock	0.0212	0.0349	1.41E-07	0.0562

The design's overall power dissipation is 0.1758 mW, while most IoT devices consume 0.1 to 1 mW power [30].

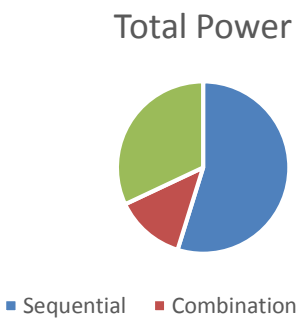


Fig. 12: A graphic representation of the percentages of total power used in a certain type of cell.

A maximum IR loss of less than 1% can be attained with wider stripes. When the layer becomes wider, it will reduce the total IR loss since R will also get smaller. If this design encounters an IR drop because of cell congestion in a specific area, the cell must be divided via placement blockage.

The following table contains the power analysis (IR) report for this design, with IR drop percentages equal to the ratio of the worst-case voltage to the IR drop.

Table 5: Table on IR droop analysis for VDD and VSS

Net	VDD	VSS
Supply Voltage	770 mV	0 mV
Worst-case voltage	769 mV	0.473 mV
Average voltage	770 mV	0.0719 mV
Average IR drop	0.0732 mV	0.0719 mV
Worst-case IR drop	0.604 mV	0.473 mV
Percentage drop	0.08%	0.06%

Due to the tiny size of the design, there isn't much IR drop; nevertheless, if it does exceed the limit, it varies from design to design. For example, in a moderate instance count design, the dynamic allowable IR drop is less than 10% and the static drop is 2% [31]. If the limit is crossed, the problem can be resolved by distributing the cell by putting cell blockages where IR problems arise, or by decreasing the R by widening the stripe layer or adding more stripes to the IR congested area.

Conclusion

The development, assembly, and implementation of a physical prototype for a cutting-edge Internet of Things device that can recognize cars are all covered in this study. Combining the ASAP7 PDK with Yosys for synthesis, OpenROAD for physical design, and OpenSTA timing analysis, a focused high-performance Internet of Things device was created. These attest to the design's true operation at maximum efficiency. The project primarily examines a chip's RTL-GDSII flow and the steps a designer takes to bring a chip from RTL to production. Additionally, using an open-source program, I completed the task and saw that the device's power, performance, and area (PPA) were all satisfactory.

Future Work

Starting at 10 GHz and satisfying the criteria with 1 mW of power, less area, and sufficient operating without timing violations were the main challenges. A significant setup and hold violation occurred on the 10 GHz frequency. Additionally, there was a brief infraction when an additional buffer and inverter were inserted to balance the clock tree to match the design at a 10GHz frequency. Thus, the area also grows. In addition, restricting IR drop to less than 1% and selecting power stripe width, which may supply power to the cell, were the other obstacles. This design can use a frequency of 5GHz to operate the device within the parameters.

But there are further issues as well; just 80% of the available space is used. Future studies could attempt to employ fewer filler cells and more device space. In order to improve system performance overall, future work will focus on developing vehicle identification algorithms,

investigating power-aware design, and investigating advanced design methodologies. Will also continue to work on constructing the entire gadget, including the MCU and LED sections. For design, the industry-standard tool can be utilized to get precise results.

Author Contributions

S. H. Rakib was responsible for planning and structuring the research, conducting the experiments, collecting the data, analyzing the results, and writing the manuscript.

S. N. Biswas reviewed the manuscript and provided necessary feedback for improvements.

Acknowledgment

The author gratefully acknowledges Neural Semiconductor Limited for the knowledge and support that contributed to this work.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

ASAP7	A design kit for 7-nm FinFET predictive processes
HPWL	Half Perimeter Wire Length
GDSII	Graphic Design System

References

[1] N. B. Soni, J. Saraswat, "A review of IoT devices for traffic management system," in Proc. International Conference on Intelligent Sustainable Systems, 2017.

[2] H. Jayakumar, A. Raha, Y. Kim, S. Sutar, W. S. Lee, V. Raghunathan, "Energy-efficient system design for IoT devices," in Proc. 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), 2016.

[3] E. Nasr, E. Kfoury, D. Khoury, "An IoT approach to vehicle accident detection, reporting, and navigation," in Proc. IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), 2016.

[4] C. V. S. Babu et al., "IoT-based smart accident detection and alert system," in Handbook of Research on Deep Learning Techniques for Cloud-Based Industrial IoT, 2023.

[5] R. K. Kodali, G. Swamy, B. Lakshmi, "An implementation of IoT for healthcare," in Proc. IEEE Recent Advances in Intelligent Computational Systems (RAICS), 2015.

[6] G. Punyavathi, M. Neeladri, M. K. Singh, "Vehicle tracking and detection techniques using IoT," Mater. Today Proc., 51(1): 909-913, 2021.

[7] A. Hosny, A. B. Kahng, "Open-source EDA and machine learning for IC," in Proc. International Conference on VLSI Design and Embedded Systems, 2020.

[8] "GitHub-The OpenROAD Project," [Online]. Available: https://github.com/TheOpenROAD-Project.

[9] T. Ajayi et al., "INVITED: Toward an open-source digital flow: First learnings from the openroad project," in Proc. 2019 56th ACM/IEEE Design Automation Conference (DAC), 2019.

[10] A. B. Kahng et al., "The OpenROAD Project: Unleashing Hardware Innovation,".

[11] "OpenROAD Flow Scripts documentation!," [Online]. Available: https://openroad-flow-scripts.readthedocs.io/en/latest/index2.html.

[12] V. Vashishtha, L. T. Clark, "ASAP5: A predictive PDK for the 5 Nm node," Microelectron. J., 126: 105481-105481, 2022.

[13] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, G. Yeric, "ASAP7: A 7-nm finFET predictive process design kit," Microelectron. J., 53: 105-115, 2016.

[14] D. Velenis, M. C. Papaefthymiou, E. G. Friedman, "Reduced delay uncertainty in high performance clock distribution networks," in Proc. 2003 Design, Automation and Test in Europe Conference and Exhibition. 2003.

[15] P. Maurine, M. Rezzoug, N. Azemard, D. Auvergne, "Transition time modeling in deep submicron CMOS," IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 21(11): 1352-1363, 2002.

[16] "GitHub-Yosys," [Online]. Available: https://github.com/YosysHQ/yosys.

[17] S. N. Adya, I. L. Markov, "Fixed-outline floorplanning: enabling hierarchical design," IEEE Trans. Very Large Scale Integr. VLSI Syst., 11(6): 1120-1135, 2004.

[18] P. Kadarkarai et al., "Implementation of a PnR flow at block level to achieve the high utilisation rate," in Proc. International Conference on Inventive Computation Technologies (ICICT), 2024.

[19] M. Chakraborty, D. Saha, A. Chakrabarti, S. Bindai, "A CAD approach for pre-layout optimal PDN design and its post-layout verification," Microprocess. Microsyst., 65: 158- 168, 2019.

[20] C. K. Cheng, A. B. Kahng, I. Kang, L. Wang, "RePlAce: Advancing solution quality and routability validation in global placement," IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 38(9): 1717-1730, 2018.

[21] S. M. Das, S. M. Rafi, "Reducing cell density congestion issue in chip design," J. Microelectron. Solid State Dev., 6(3): 11-17, 2020.

[22] Sarrafzadeh, M. Wang, "NRG: Global and detailed placement," in Proc. International Conference on Computer Aided Design, 1997.

[23] N. Patel, "A novel clock distribution technology multisource clock tree system (MCTS)," Int. J. Adv. Res. Electr. Electron. Instrum. Eng., 2(6): 2234-2239, 2013.

[24] E. G. Friedman, "Clock distribution networks in synchronous digital integrated circuits," Proc. IEEE, 89(5): 665-692, 2001.

[25] D. Harris, M. Horowitz, D. Liu, "Timing analysis including clock skew," IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 18(11): 1608-1618, 1999.

[26] B. Rebaud, M. Belleville, C. Bernard, Z. Wu, M. Robert, P. Maurine, "Setup and hold timing violations induced by process variations, in a digital multiplier," in Proc. 2008 IEEE Computer Society Annual Symposium on VLSI, 2008.

[27] J. H. Kim, W. Kim, Y. H. Kim, "Efficient statistical timing analysis using deterministic cell delay models," IEEE Trans. Very Large Scale Integr. VLSI Syst., 23(11): 2709-2713, 2015.

[28] M. Mostafa, M. Watheq El-Kharashi, M. Dessouky, A. M. Zaki, "A novel flow for reducing dynamic power and conditional performance improvement," IEEE Trans. Circuits Syst. I Regul. Pap., 68(5): 2003-2016, 2021.

[29] N. P. Bose, N. Santhi, "Efficient leakage reduction approach for low power VLSI design using modified feedback sleeper stack technique," Int. J. Electron. Commun. Eng., 11(3), 2024.

[30] P. Mayer, M. Magno, L. Benini, "Smart power unit—mW-to-nW power management and control for self-sustainable IoT devices," IEEE Trans. Power Electron., 36(5): 5700-5710, 2020.

[31] V. G. Menon, S. Jacob, S. Joseph, P. Sehdev, M. R. Khosravi, F. Al-Turjman, "An IoT-enabled intelligent automobile system for smart cities," Internet Things, 18, 2022.

Biographies



Saeed Hossen Rakib received his bachelor's degree in Electrical and Electronic Engineering from Ahsanullah University of Science and Technology in Dhaka in June 2022. After that, he began working as a physical design engineer in Neural Semiconductor Limited. ASIC Design, Low Power IC Design, VLSI Circuit and System Design, Microelectronics and Nanotechnology, Electronics, SoC Design are among his areas of interest in study.

- Email: s.h.rakib.153@gmail.com
- ORCID: 0009-0008-3727-6699
- Web of Science Researcher ID: N/A
- Scopus Author ID 58920493500
- Homepage: NA



Satyendra Nath Biswas (M'98) received his B.Sc. from BUET and M.Sc. & Ph.D. from Yamaguchi University, Japan. He has worked as an R&D Engineer in Dhaka and Canada and was a Research Assistant at the University of Ottawa. Currently, he is an Assistant Professor at Georgia Southern University. His research focuses on VLSI design, built-in self-testing and reconfigurable computing. He is a Professional Engineer (P.Eng.) and a member of IEICE.

- Email: sbiswas.eee@aust.edu
- ORCID: 0000-0002-5334-0784
- Web of Science Researcher ID: N/A
- Scopus Author ID 14048019400
- Homepage: NA

How to cite this paper:
S. H. Rakib, S. N. Biswas, "Implementing yosys & openroad for physical design (PD) of an IoT device for vehicle detection via ASAP7 PDK," J. Electr. Comput. Eng. Innovations, 13(2): 463-472, 2025.

DOI: 10.22061/jecei.2025.11258.785

URL: https://jecei.sru.ac.ir/article_2315.html