# ASIC Chip Design For Agriculture Application

by

ABHISHEK PANDYA
202011037

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY


May, 2022

# Declaration

I hereby declare that

(i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at DA-IICT and has not been submitted elsewhere for a degree,

(ii) due acknowledgment has been made in the text to all the reference material used.

A.J. Pandya

Signature of Student

# Certificate

This is to certify that the thesis work entitled ASIC Chip Design for Agriculture Application has been carried out by Abhishek Pandya (202011037) for the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology under my/our supervision.

Rutu M Parekh.

DR. Rutu Parekh

Thesis Supervisor

# Acknowledgements

I want to thank my thesis guide, Dr. Rutu Parekh, for her invaluable assistance and insights leading to this work. My sincere thanks also go to my co-guide, Dr. Vinay Palaparthy, whose paper became a reference to me in this thesis. With their noble guidance, support, encouragement, and enthusiasm, they have helped me continue and successfully finish this research. I want to express my deepest gratitude to the defence committee members, Dr. Sreeja Rajendran, Dr. Yash Agrawal, and Dr. Biswajit Mishra, for their valuable feedback on the betterment of the research work.

Special thanks to my colleague Mr. Nishit Nathwani for helping me with the analog aspects of the mix signal design. I would also like to thank the whole help desk team at DAIICT for assisting us throughout the year regarding the installation and availability of different tools. Their prompt reply made our work easier. Finally, I would also like to thank all my friends and family members for encouraging and supporting me whenever I needed them.

# Contents

# Abstract

This paper presents an application-specific integrated circuit (ASIC) implementation suitable for agriculture applications, which employ RISC-V as a digital processing unit and the sensor interfacing circuits. The motivation is early plant disease detection and providing control measures to improve yield. The design consists of three sensors for collecting the temperature, humidity, and wetness of crops. The design targets an edge computing application where data is processed and analyzed closer to the point where it is obtained. Because of this, the latency and the required bandwidth are reduced considerably. The design of analog circuits is done using the specifications obtained from the sensors. The data obtained can be processed with the computing device to extract information and take desired actions. The RTL-based design of a processor is implemented using Verilog HDL. Logic equivalence is verified using Xilinx ISE. Physical realizations of the design are obtained using RTL to GDSII design flow. The analog design consists of a unity gain buffer, sample and holds circuit, and flash type ADC. We have tested our ASICs with AMS verification methodology using Cadence CAD tools. The area of the designed RISC core is 50927 $um^2$, the critical path delay is 361 ps, and the power consumption is 420 uW after the placement and route stage. The designed analog ASIC contains all the signal processing components, and the combined length of all the routed layers comes out to be 23358 um. Vertical length is almost 70%, and the remaining is Horizontal length. The total area of the analog ASIC is 2.366 $mm^2$.

# List of Principal Symbols and Acronyms

| | |
|---|---|
| Kn′ | Process transconductance parameter (Electron) |
| Kp′ | Process transconductance parameter (Hole) |
| $\mu_n$ | Mobility of a hole |
| $\mu_p$ | Mobility of an electron |
| $g_m$ | Transconductance of a transistor |
| $A_v$ | Voltage Gain |
| V | Voltage |
| R | Resistance |
| C | Capacitance |
| S | W/L ratio |

Other minor symbols are defined at first occurrence, and where necessary some symbols are redefined in the text.

| | |
|---|---|
| ASIC | Application Specific Integrated Circuit |
| HDL | Hardware Description Language |
| RISC | Reduced Instruction Set Computing |
| ADC | Analog to Digital Converter |
| C to F converter | Capacitance to Frequency converter |

# List of Tables

# List of Figures

# Chapter 1
# Introduction

## 1.1 General

Crop disease detection at early stages and providing control measures have become highly required to improve crop yield. The agricultural field's plant diseases, weeds, and pests lead to almost 36% crop loss. To reduce such loss, sensor-based technology is vital and plays a crucial role in the early detection of plant diseases [1]. Different parameters are responsible for the growth of the plant. By controlling that parameter, the production and development of plants can be improved rapidly compared to traditional agriculture. A precision agriculture-based network can quickly reduce problems with conventional agriculture [2]. Many sensor nodes collect information from the environment and then transmit this data to a base station that forms the sensor network. A wireless sensor network can be designed to sense and process temperature, humidity, and wetness [3]. Precision agriculture is innovative, based on the latest technologies which target to streamline the agricultural process. Parameters such as humidity, temperature, soil characteristics, and nutrients all play a role in plant development. It is essential to measure and monitor many interacting physical variables in greenhouse crops to improve their quality and productivity. Only control systems with built-in applications are capable of completing these tasks. It is costly to build a greenhouse. The high cost of greenhouse technology prevents many farmers from adopting it [5]. The resources in the field can be managed for maximum utilization of available resources. In developing nations, the advantage of the IoT in agriculture and farming is the economic technology for developing the agriculture process. The low power consumption makes it work on batteries or solar for stationary nodes [6].

The progress in the sensors used is also a significant part of this development. E.g., In reference 7, they have used a parallel-plate capacitor instrument to detect soil moisture based on frequency response characteristics of the soil. Also, A measuring method based on the model's frequency characteristic is proposed, the edge effect of a parallel-plate capacitor is analyzed, and the functional relation of the test capacitor on the relative dielectric constant of medium and probe insertion is established. A capacitance calibration method that multiplies the capacitance formula by a coefficient is proposed [7].

The data gathered from the sensors must be processed for extracting information. The Machine learning algorithms can be used here. The data collected from the sensors vary significantly in terms of the range of values. To better insight and introduce uniformity, we perform a scaling operation on them. In the last twenty years, exponential growth in IoT has been witnessed. In the last few years, IoT has been used in conjunction with Machine learning Algorithms for different domains like agriculture, finance, engineering, and healthcare [9-10]. Accurate measurements and cost-effective fabrication processes for large-volume and high-performance sensors with flexible form factors are essential to meet the stringent performance requirements of the emerging application areas [11]. The analytics and decision support system that provides advice to the farmer about the fertilization strategy can be presented. The decision support model based on a machine learning Algorithm offers advice to farmers based on the conceptual plant model (the stage for the plant growth), the multi-sensor values, and meteorological prognosis, considering possible environmental aspects. Examples of recommendations cover classical farming actions, e.g., using a specific type of fertilizer or the need for irrigation [12]. The critical moments identified were the harvest season, with farming often affected by un-seasonal precipitation, and the germination stage, with an additional high risk for low temperatures at high altitudes. A good understanding of the differentiated risks and effectiveness of in-season coping strategies could promote sustainable crop production in similar agro-ecologies.

Moreover, the effectiveness of the present-day coping strategy, rather than the use of coping approaches itself, could signal a potential ability to adjust to future climate change [13-14]. Many signal processing applications demand highly energy-efficient flexible implementations. The Domain-Specific Instruction-set Processor (DSIP) architecture was proposed, which can be used to deploy in the domain of online surveillance. The system is heavily optimized ASIC realization for the same application benchmark [17].

The proposed work checks the suitability of the soil for urban terrace farming. This work helps the user decide the best plant and fertilizers to grow plants in the given soil conditions. Implementing the proposed system using FPGA is to design it as an Application Specific Integrated Circuit (ASIC) to ensure the features of field re-programmability, parallel processing, faster time-to-market, low cost, and power consumption [18]. Recently, a miniature electronic system was described to interact specifically with plant-based electrochemical

biosensors. And it was proposed that the cost be further reduced by converting the system to an ASIC using VLSI technology. We have implemented a similar strategy which improves the design matrix [19]. Agricultural mechanization impacts agricultural productivity and society development far-reaching. The emergence of ASIC (Application Specific Integrated Circuit) provides the possibility for full intelligence and automation of agricultural products [20].

In related work, a fully integrated application-specific integrated circuit (ASIC) for the readout of multiple electrochemical sensor signals. It consists of an analog router that routes to various sensors, SAR ADC, two DACs, and a multiplexer [21]. We have also developed the ASIC with the signal processing circuitries in our work. The implemented work is the ASIC implementations of all such components that can be integrated to make an SoC (System on Chip). Smart agriculture based on new sensors, data analytics, and automation is an essential enabler for optimizing yields and maximizing efficiency to feed the world's growing population while limiting environmental pollution. The implemented work is the ASIC implementations of all such components that can be integrated to make an SoC (System on Chip).

## 1.2 Organisation of the Thesis

This paper is organized in the following way. In section 2, Soc-ASIC design's top-level design is presented and discussed, focusing on its various parts and the layout design. Both analog and digital components that are internal to ASIC are analyzed. The details of the sensors used for it are also mentioned here. Further, sections 3 and 4 discuss the design presented using detailed descriptions with their analytical equations. Use of the specifications is done for designing that are obtained from the different sensors that are used. Section 5 shows experimental results concerning the digital processor and analog components. Finally, a summary is given in section 6.

# Chapter 2
# Technology and Functionality of ASIC for Agriculture

## 2.1 Block Diagram of proposed ASIC

The top module of the design is shown in figure I; all the components are designed as the ASIC Implementation. Figure II shows the detailed diagram of the implementation. RV32I is a RISC-V processor. It has two memories which are instruction and data memory. Instruction memory contains the program that must be executed. It is unidirectional memory. Data memory includes the data that we load or store in memory. Data memory is bidirectional.

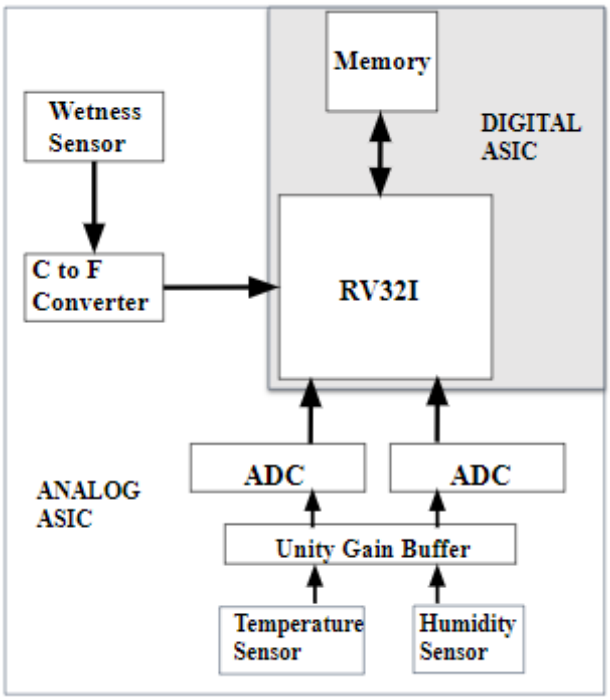


Figure 2.1 Block Diagram of ASIC

These implementations are in the digital domain (RTL to GDSII). Analog components are ADC (Analog to Digital Converter), C to F Converter, and UGB (unity Gain Buffer). They are for the sensors and can be used for communication with the processor. The ADC is used for the sensors with the voltage outputs, and the C to F converter is used for the sensor with the capacitance output. The sensors used here are temperature, humidity, and leaf wetness sensors. The following sections have detailed information on all the implementations done. The memory decoder passes the desired information to the processor's data bus.

It controls the data that the processor will read. The data can be from data memory, ADCs, or the counter.

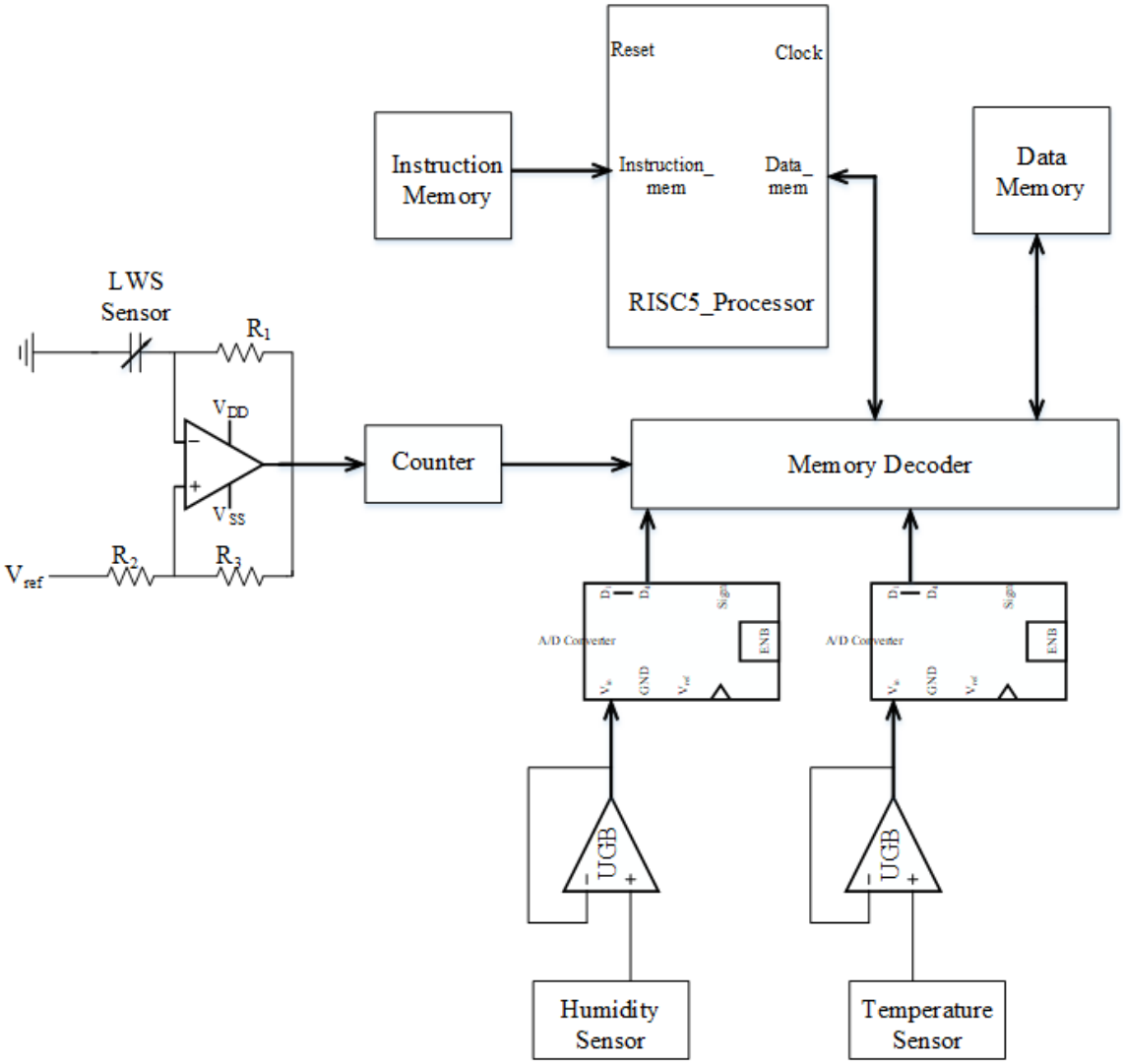


Figure 2.2 Data flow of the design

The designed ASIC SOC for agriculture has two different technology nodes. The design of the RISC core is done using a 45 nm technology node. In contrast, all the analog components are created on 180 nm technology. The reason behind these choices is that the sensors targeted for these applications produce voltages that are more compatible with the higher technology nodes than smaller nodes. The processor is designed at a smaller node because the computing element can be more efficient in all design matrices. The proposed sensors are HIH 5030 (Humidity Sensor), MCP 9700A (Temperature Sensor), and a custom-designed leaf wetness sensor. The overall output voltage range for these sensors is 0.1 V to 1.575 V. The leaf wetness sensor gives output in terms of capacitance in 0 to 25 nF, converted to frequency, and then read by a counter.

# Chapter 3
# Digital ASIC Implementation

The five-stage pipeline processor is designed with the RISC-V Instruction set architecture (ISA). RISC-V is an open standard instruction set architecture (ISA) based on established reduced instruction set computer (RISC) principles. The version of ISA used for the design is RV32I, a Base Instruction set of 32 bits. It is a simple instruction set comprising just 47 instructions. Eight instructions are system instructions (system calls and performance counters) that can be implemented as a single trapping instruction, reducing the number of mandatory instructions to 40. As with many RISC instruction sets, the remaining instructions fall into three categories: computation, control flow, and memory access. RISC-V is a load-store architecture in which arithmetic instructions operate only on the registers and only loads and stores transfer data to and from memory. 32 general-purpose integer registers in RV32I, named R0–R31, each 32 bits wide. The only additional register is the program counter, pc, which holds the byte address of the current instruction. Instructions in RV32I are 32 bits long and must be stored naturally aligned in memory, in little-endian byte order. Six instruction formats, which Figure 3.1 shows, comprise the 47 instructions: four major formats, R, I, S, B, U, and J. Opcode decides the type of instruction. Instructions in these formats source up to two register operands, identified by rs1 and rs2, and produce up to one register result, identified by rd. An essential feature of this encoding is that these register specifiers, when present, always occupy the same position in the instruction. This property allows register fetch to proceed parallel with instruction decoding, facilitating a critical path in many implementations. Another feature of this encoding scheme is that generating the immediate

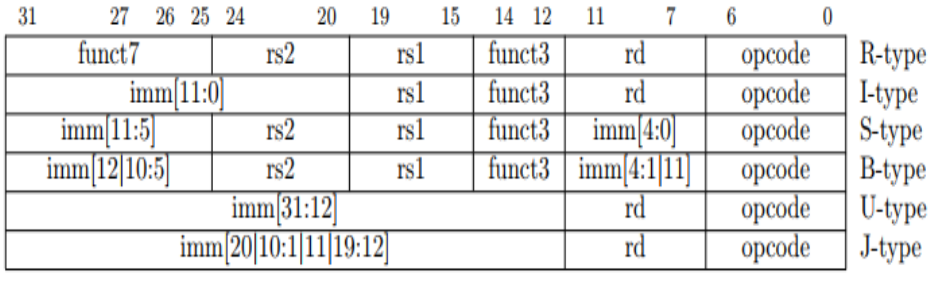


Figure 3.1 Formats of Instructions

operand from the instruction word is inexpensive. Seven of the 32 bits in the immediate operand always come from the same position in the instruction,

including the sign bit, which, due to its high fan-out, is the most critical. 24 more bits come from one of two positions, and the final immediate bit has three sources. This section describes the entire implementation of the processor from RTL to GDSII.

## 3.1 Stages of a Pipeline Processor

There are five stages of a pipeline processor. The stages include Fetch, Decode, Execute, memory, and Write back.

### 3.1.1 Fetch stage

The fetch unit comprises an Instruction Fetch Register (32 bits). It takes PC as input and loads instruction in IF Register. For all instructions, for those who do not include branch and jump operation, PC is incremented by 4. Instruction memory holds different instructions to be performed by the processor. The size of an Instruction memory is 4K bytes. The size of an Instruction is 32 bits. PC takes value according to the taken_branch signal. If taken_branch = 1, PC = branch_address generated by the branch unit, otherwise PC = NPC [PC + 4]. For the Jump instructions, if the j flag is set, we'll not take any further instructions until two clock cycles. It means that we'll not fetch the following two instructions written just after the jump instruction because we have to jump to a newer location. When the fetch unit encounters the jump instructions, it'll set the j flag.

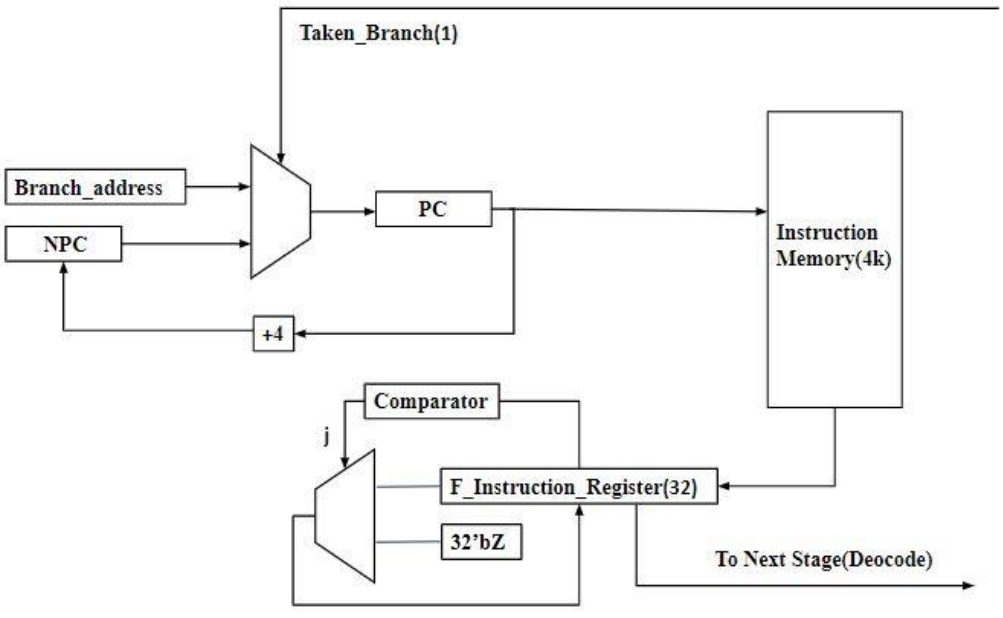


Figure 3.2 (a) Fetch Stage

### 3.1.2 Decode stage

The decode unit reads the fetched instruction and decodes the address of two source operands and destination register for register-based arithmetic, logical and shift instructions, immediate data (data or effective address) for immediate data-based instructions like arithmetic, logical, and branch type instructions. Register Bank reads the value of source operands (Rs1 and Rs2) at the negative edge of a clock and writes the data in the destination register (Rd) at the positive edge of the clock. We also have a data forwarding unit to tackle the problem of hazards in case of data dependency. The forwarding unit forwards the data from the following stages (e.g., Execute, memory) and decides whether to take operand for ALU operation (In the execute stage) from ALU result or memory Data to eliminate hazards like RAW(Read after Write).

This stage decodes the instruction available in the instruction register and generates signals (i.e., alu_op, type, etc.). The decoder circuit will give source register addresses (i.e., rs1_address and rs2_address), destination register address (i.e., rd_address), and immediate data. The forwarding unit will compare different destination register addresses from the following stages with source operands address, and based on this, it will generate d_forward_ctrl, which will be used in the operand selector. alu_op will decide the operation to be performed in ALU, and the type signal will define the type of instruction.

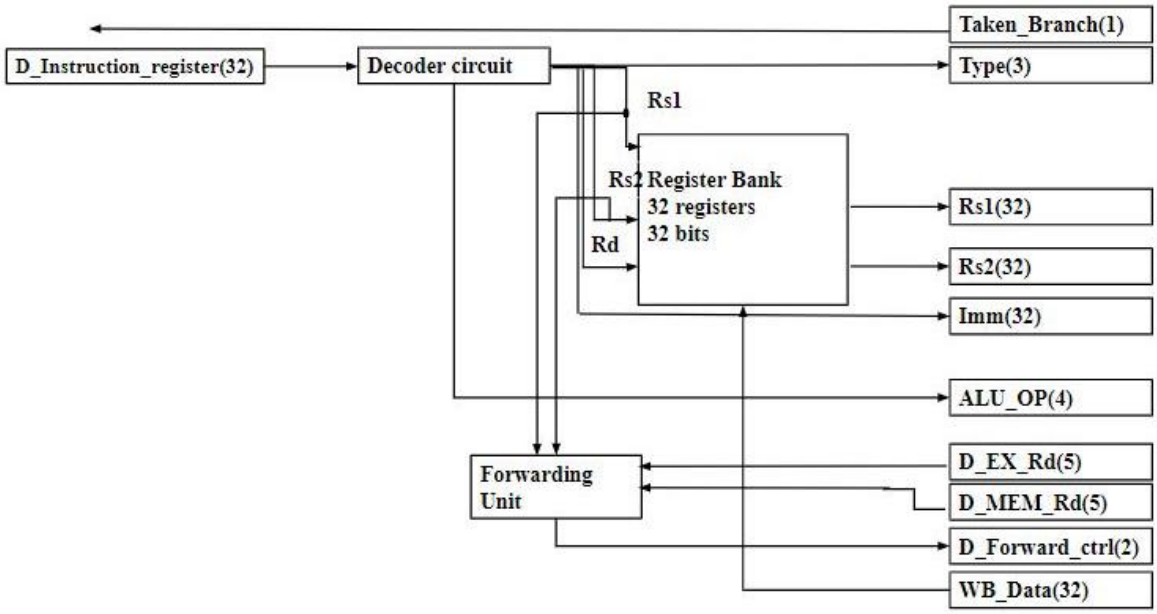


Figure 3.2 (b) Decode Stage

### 3.1.3 Execute stage

ALU Module uses the aluop generated by the ALU Control unit to operate on source operands. It generates the result and stores it in the ex_alu_result register. The operand Selector unit decides which data to take in the operand to perform different operations. PSR (Program Status Register) shows the status according to the operations performed on the operands. The width of the PSR is 3 bits. The branch unit decides whether to take the branch according to the status of PSR.

This stage executes different instructions according to the alu_op signal in the ALU unit. Operand1 to the ALU will be selected according to the forward_ctrl signal, and Operand2 will be chosen according to the type and forward_ctrl signals. ex_rsk (k = 1,2) defines data of rsk register, ex_alu_result defines data of previously executed instruction, mem_result defines data from the memory that is received in previous instruction, and imm defines immediate data from the instruction. PSR is a program status register. It contains 3 bits, which will be used to decide the taken_branch signal in conditional branch instructions and jump type of instructions.

| Sr. No. | PSR bits (3) | Condition |
|---|---|---|
| 1 | 000 | rs1 = rs2 |
| 2 | 001 | rs1 != rs2 |
| 3 | 011 | rs1 >= rs2 (signed) |
| 4 | 010 | rs1 < rs2 (signed) |
| 5 | 100 | rs1 < rs2 (unsigned) |
| 6 | 101 | rs1 >= rs2 (unsigned) |
| 7 | 110 | Jump and link |
| 8 | 111 | Jump and link with target address |

Table 3.1 PSR Values for different Conditions

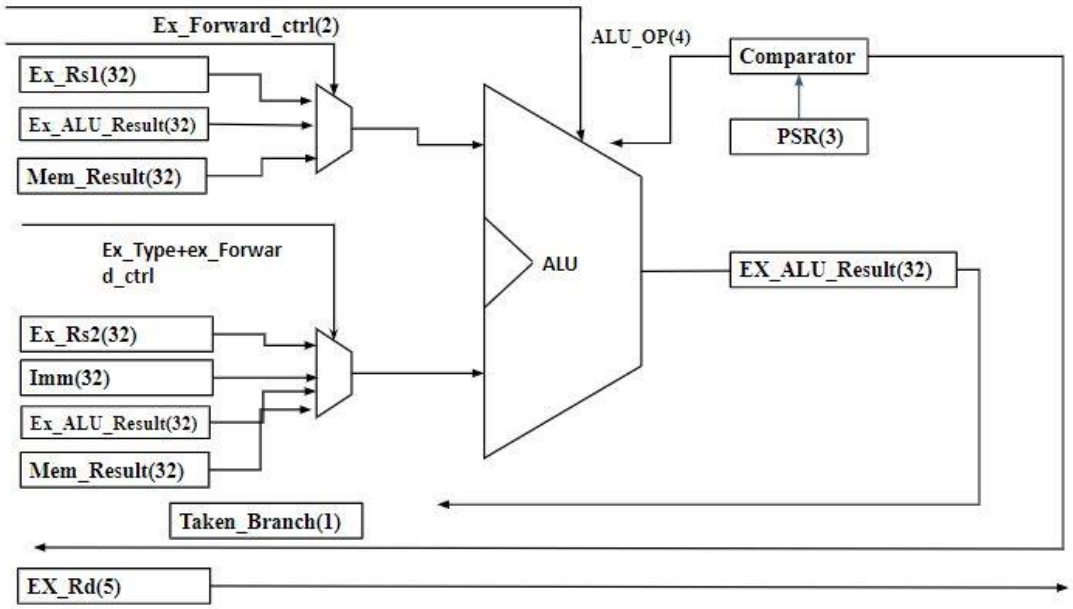


Figure 3.2 (c) Execute Stage

### 3.1.4 Memory stage

Data memory stores the data from and to the processor. Data Memory is selected to be 8KB in size. It is Bi-directional Memory. As in RISC architectures, only this stage is given the privilege to communicate with memory. We have created one signal, which allows either read or write operation based on the instruction. The instructions which access this stage are the load and store instructions. The data can be read according to the operation identified in the instruction decoding at the decode stage. Here, the store operation will allow data from register to the memory. The data here can be signed, half-wordd, or byte. The instruction set allows different variations of it. The loading operation will happen here, however,r the writing to the registers will happen in the write-back stage only.

memory unit will either write to the data memory or will read from the data memory. This stage is used to load and store data from register_bank to data_memory and vice-versa. mem_rd_wr signal defines load or store operation (i.e. mem_rd_wr = 0 , load and mem_rd_wr = 1, store). mem_address_result will contain either result or address. mem_type defines the type of instruction. mem_write_data register defines the data that must be written in data_memory in a store instruction. mem_result and mem_rd_adderess are used for data forwarding.

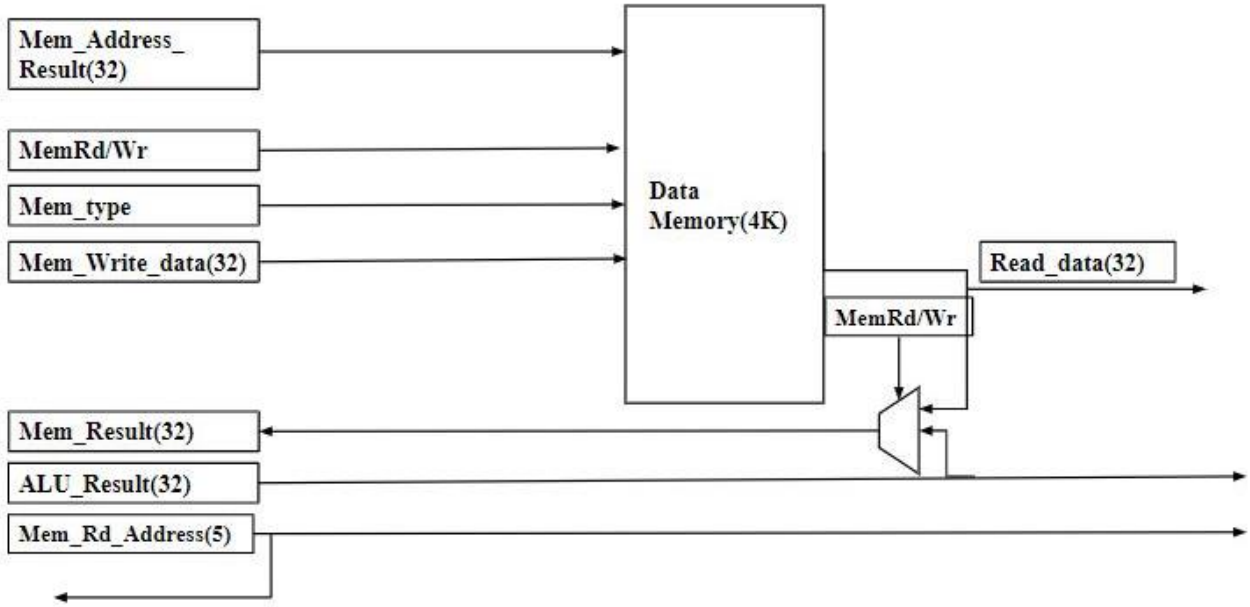


Figure 3.2 (d) Memory Stage

### 3.1.5 Write Back stage

Write back unit will select the data and write the data to register the bank. alu_op signal is used for defining the number of Bytes to be moved in the register bank. wb_alu_result is the data to be written in the register bank, which is the result from the ALU, and wb_read_data is the data to be written in the register bank, which is from the memory stage (i.e., load). No. of bits stored in the register of register bank will depend upon the instruction (i.e., byte, half word, unsigned).

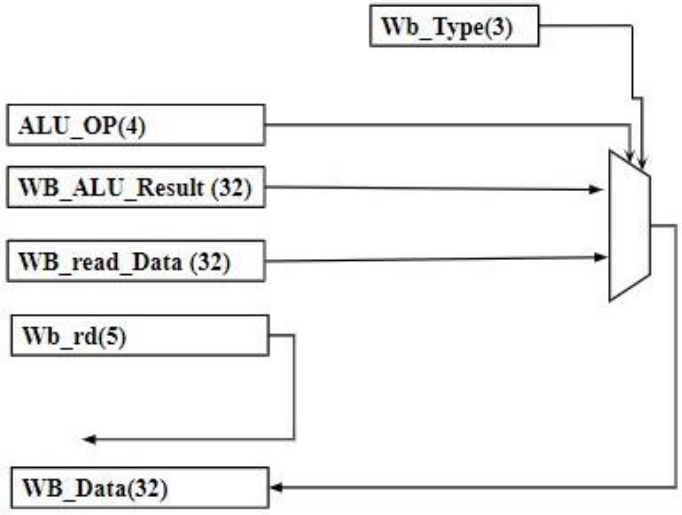


Figure 3.2 (e) Execute Stage

## 3.2 Physical Design Flow

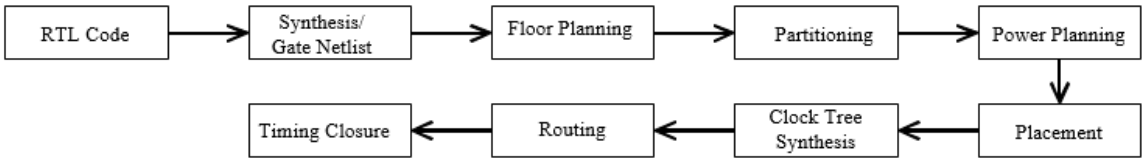


Figure 3.3 Physical Design Flow

Physical design flow converts the HDL code into technology mapped, placed, and routed design. It includes several steps, which are shown in figure 3.2.

### A) Gate Level Netlist stage

The files required for the netlist stage are mentioned below.

i) Gate level Netlist:

A synthesis tool will translate RTL into a collection of interconnected logic gates that define the logic. The most common format is Verilog.

ii) Standard Cell Library:

The standard cell library will have a layout model and timing model information for the standard cells.

iii) Technology file

The rules of the specific process selected must be given to PNR tools. The file includes widths, and spacing, via definitions of metal layers.

iv) Timing Constraints:

SDC files define the timing constraints of our design. We will have the clock definitions, false paths, input, output delay constraints, etc.

### B) Floor Planning, Partition, and Power Planning

The floor plan determines chip quality. At this step, we define the size of the chip/block, allocate power routing resources, place the hard macros, and reserve space for standard cells.

i) Core Boundary:

The floor plan defines the size and shape of your chip/block. Floor planning can be controlled by various parameters like aspect ratio, core utilization, and boundary.

ii) IO Placement/Pin Placement:

We need to place IO pads and IO buffers on the chip. According to the package, we will also get a maximum and minimum die size.

iii) Macro Placement:

Once the size & shape of the floor plan are ready, we can place macro. The creation of the power Rings and straps happens in this step.

## C) Placement

Standard cell placement happens in this stage after floor planning, i.e., creating the core area, placing the macros, and deciding the design's power network structure.

## D) Clock Tree Synthesis

At this stage, buffer insertion, gate sizing, and any other optimization technique are employed on the data paths, but no change is made to the clock net.

## E) Routing

After CTS, the routing process determines the exact paths for interconnections, including the macro and standard cell pins, the pins on the block boundary, or pads at the chip boundary.

## F) Timing Closure

It optimizes circuit performance through specialized placement or routing techniques.

# Chapter 4
# Analog ASIC Implementation

The Analog ASIC contains ADCs, C to F converter, and buffers. We have designed each component from its basic building blocks and various configurations. The two basic building blocks are Op-Amp and comparator. The following section contains the Implementation details of Op-Amp and Comparator. After that, the discussion of different components designed is presented.

## 4.1 Operational Amplifier Design

Operational amplifiers (Op-Amps) are integral to many analog and mixed-signal designs. Op-Amps are realized for functions ranging from dc bias generation to high-speed amplification or filtering. Here we have designed an Op-Amp based on the specification according to the design requirements. Our designed Op-Amp specifications are mentioned in Table III. We have designed a standard CMOS two-stage Op-amp and verified its functionality. The two-stage Op-Amp consists of the first stage as the differential amplifier and the second stage as a common source amplifier, as shown in the below figure. The biasing circuit is not shown in figure 5. According to the Specification, we have calculated the W/L ratios of the transistors present in the design, and those are shown in Table IV.

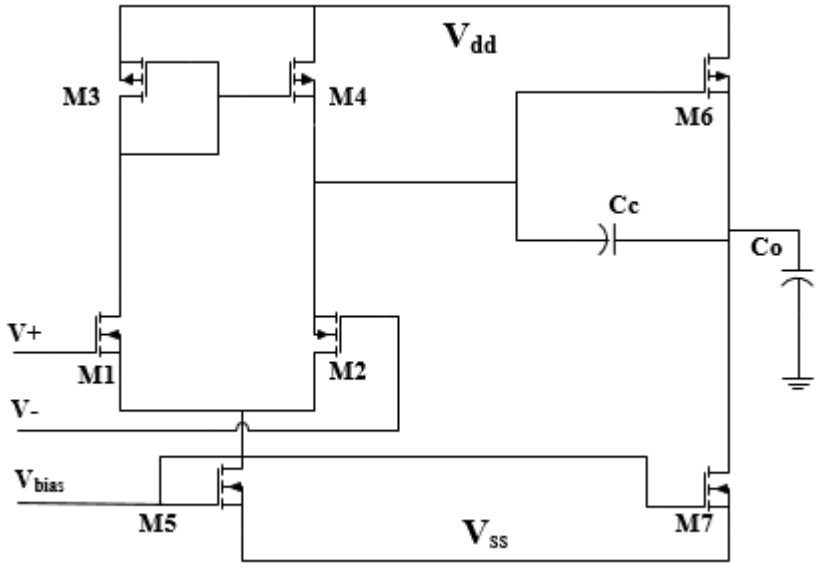


Figure 4.1 Op-Amp Circuit

| Sr. No. | Design parameters | Value |
|---|---|---|
| 1 | Supply Voltage | 1.8 V |
| 2 | Biasing Current | 90 - 100 uA |
| 3 | Open-loop Gain | > 80 dB |
| 4 | Gain - Bandwidth Product | > 5 MHz |
| 5 | Slew Rate | 10 V / us |
| 6 | ICMR | 0 to 1.6 V |
| 7 | CMRR | > 60 dB |
| 8 | PSRR | > 60 dB |

Table 4.1 Design Specifications of an Op-Amp

## 4.2 Design Procedure of two-stage Op-Amp

To design a two-stage CMOS Op-Amp, various design steps need to be considered. These design steps help achieve the proper sizing of the transistors used in an op-amp. It has been supposed that all the transistors are in a saturation state. Then we choose the device length to be used throughout the circuit, which keeps the channel length modulation parameter constant. Below are the steps for designing a two-stage CMOS op-amp. For the desired phase margin of 60 degrees, choose the minimum value of Cc by assuming z 10 GB using the equation

$$C_C = 0.22C_L \quad (1)$$

Determine the tail current I5 using the equation

$$I_5 = SR.C_C \quad (2)$$

Design for S3 from maximum input voltage specifications,

$$S_3 = S_4 = \frac{2I_3}{K_3'\left[V_{dd} - V_{in(max)} - V_{to3(max)} + V_{t1(min)}\right]^2} \quad (3)$$

Verify that the pole and zero due to Cgs3 and Cgs4 will not dominate by assuming P3 > 10 GB,

$$\frac{gm_3}{2C_{gs3}} > 10\ \text{GB}$$

Design for S1 and S2 so as to achieve desired GB

$$g_{m1} = GB.C_C$$

$$S_1 = S_2 = \frac{(g_{m2})^2}{K_2'I_5} \quad (4)$$

$$V_{ds5(sat)} = V_{in(min)} - V_{ss} - \sqrt{\frac{I_5}{B_1}} - V_{t1(max)} \geq 100\ mV$$

$$S_5 = \frac{2.I_5}{K_5\left(V_{ds5(sat)}\right)^2} \quad (5)$$

$$P_2 \geq 2.2GB$$

$$g_{m6} = 2g_{m2}\left(\frac{C_L}{C_C}\right)$$

which gives

$$S_6 = S_4\left(\frac{g_{m6}}{g_{m4}}\right) \quad (6)$$

Now we solve for $I_6$ as given by the equation

$$I_6 = \frac{{g_{m6}}^2}{2K_6'S_6}$$

Design for $S_7$ as given by the equation

$$S_7 = S_5\left(\frac{I_6}{I_5}\right) \quad (7)$$

Check for gain and power dissipation as given by equation

$$A_v = \frac{2g_{m2}g_{m6}}{(I_5I_6)(\lambda_2 + \lambda_4)(\lambda_6 + \lambda_7)}$$

and thus,

$$P_{diss} = (I_5 + I_6)(V_{DD} + V_{SS}) \quad (8)$$

| Sr. No. | Transistor | W/L ratio |
|---|---|---|
| 1 | M1, M2 | 2 |
| 2 | M3, M4 | 10 |
| 3 | M5, M8 | 44 |
| 4 | M6 | 48 |
| 5 | M7 | 100 |

Table 4.2 W/L ratio of MOSFET

## 4.3 Comparator Design

A comparator is an essential building block for many electronic circuits. This design is used for the creation of an ADC. They are also used in peak detectors, zero-crossing detectors, BLDC operating motors, and switching power regulators. We have designed a comparator according to the design requirements

based on the specifications in this implementation. Our designed comparator specifications are mentioned in Table IV.

We have designed a standard CMOS two-stage comparator and verified its functionality. This configuration is based on the open-loop version of an Op-Amp. The technology file used for the design was gpdk 180nm, provided inside the cadence virtuoso. Various input specifications include the propagation delay, the peak-to-peak output voltage swing, the resolution, and the ICMR. If the comparator is not slewing, the pole locations play an important role. But, if the comparator slews, the capability to charge or discharge capacitors becomes crucial.

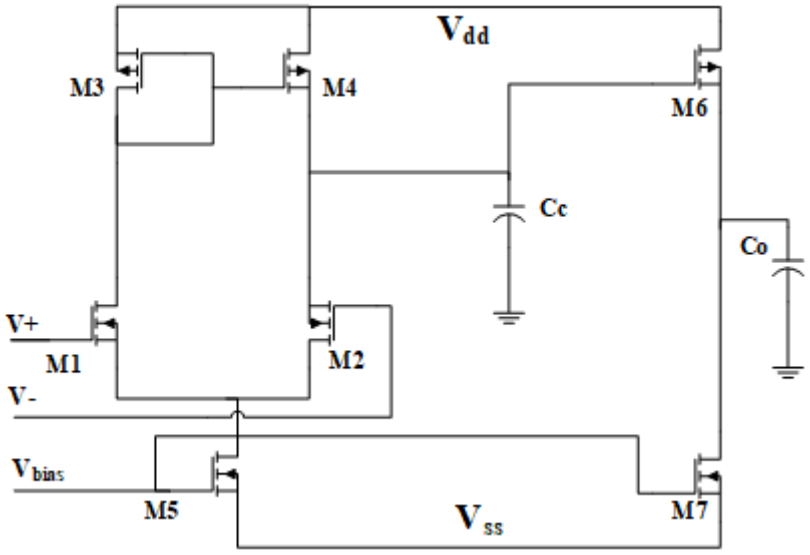


Figure 4.2 Comparator Circuit

| Sr. No. | Design parameters | Value |
|---|---|---|
| 1 | Supply Voltage ($V_{dd}$, $V_{ss}$) | (1.8, -1.8) V |
| 2 | Propagation Delay | 200 ns |
| 3 | $V_{OH}$ | 1.6 |
| 4 | $V_{OL}$ | -1.6 |
| 5 | $V_{ICM+}$ | 1.6 |
| 6 | $V_{ICM-}$ | -0.8 |
| 7 | $V_{in(min)}$ | 1 mV |

Table 4.3 Design Specifications of a Comparator

## 4.4 Design Procedure of two-stage Comparator

The steps below can accomplish a minimum propagation delay time for the two-stage, open-loop comparator. This procedure attempts to design the pole locations to correspond to a desired propagation delay time. The obtained W/L ratios are shown in Table V. $P_I$ and $P_{II}$ are the poles of the comparator. $t_p$ is the propagation delay.

$$|P_I| = |P_{II}| = \frac{1}{t_P\sqrt{mk}}$$

$$I_7 = I_6 = \frac{|P_{II}|C_{II}}{\lambda_N + \lambda_P} \quad (1)$$

$$\frac{W_6}{L_6} = \frac{2.I_6}{K_P'(V_{SD6}(sat))^2}$$

and

$$\frac{W_7}{L_7} = \frac{2.I_7}{K_N'(V_{DS7}(sat))^2} \quad (2)$$

Guess $C_1$ as 0.1-0.5 pF;

$$I_5 = I_7\frac{2C_I}{C_{II}}$$

$$\frac{W_4}{L_4} = \frac{W_3}{L_3} = \frac{I_5}{K_p'\left(V_{sg3} - |V_{tp}|\right)^2} \quad (3)$$

and

$$g_{m1} = \frac{Av(0)(g_{ds2} + g_{ds4})(g_{ds6} + g_{ds7})}{g_{m6}}$$

thus,

$$\frac{W_1}{L_1} = \frac{W_2}{L_2} = \frac{{g_{m1}}^2}{K_N I_5} \quad (4)$$

$$V_{DS5}(sat) = V_{icm-} - V_{GS1} - V_{SS}$$

$$\frac{W_5}{L_5} = \frac{2.I_5}{K_N'(V_{DS}(sat))^2} \quad (5)$$

| Sr. No. | Transistor | W/L ratio |
|---|---|---|
| 1 | M1, M2 | 8 |
| 2 | M3, M4 | 10 |
| 3 | M5, M8 | 15 |
| 4 | M6 | 272 |
| 5 | M7 | 136 |

Table 4.4 W/L ratio of MOSFET

## 4.5 Unity Gain Buffer Design

The unity gain buffer is made from the Op-Amp, which is designed. It is used in the non-inverting configuration with zero feedback resistance. The UGB outputs the same voltage as input, but it boosts the current capability of the design. That is why it is used before the voltage is given to ADC. The output of the UGB is given to ADC.

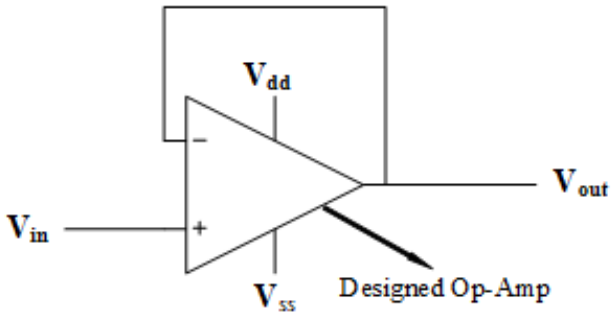


Figure 4.3 UGB Circuit

## 4.6 Design Procedure of UGB

As we do not have any feedback resistance in the design of UGB, we can take it as 0. The overall voltage gain can be calculated with the equation below. The output follows the input as it is.

$$V_{out} = V_{in}\left(1 + \frac{R_f}{R_1}\right)$$

Here, Rf = 0.

$$V_{out} = V_{in}$$

## 4.7 Sample and Hold Circuit

The sample and Hold Circuit are used for taking the samples from the sensors. It holds them for a particular time and then outputs the sampled part of the input signal. A Sample and Hold circuit consists of switching devices, a capacitor, and an operational amplifier. The capacitor is the heart of the Sample and Hold Circuit because it is the one that holds the sampled input signal and provides it at output according to command input. The figure is shown below, where we have used the op-amps designed for making sample and hold circuits.

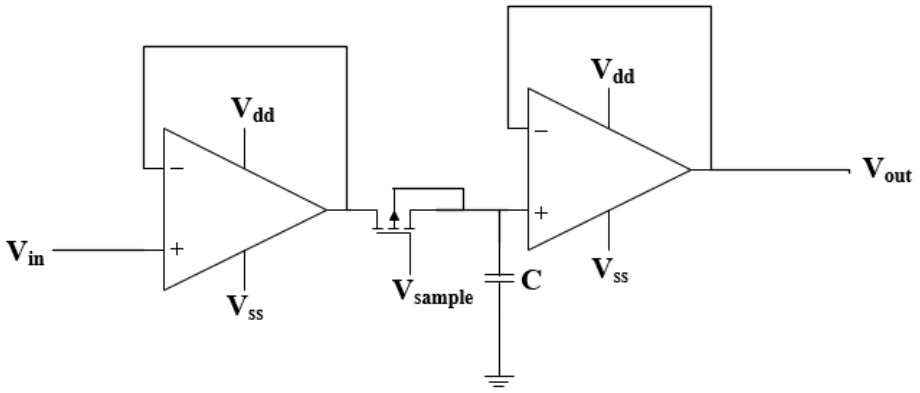


Figure 4.4 Sample and Hold Circuit

Three principal factors will control the acquisition time of the above circuit.
1) RC time constant where R is the rds (ON), i.e., on-resistance of MOSFET and C is the holding capacitance $C_H$.
2) Maximum output current, which can be the source or sunk by the operational amplifier.
3) Slew rate of the Op-Amp.
According to the above parameters, the capacitance and ON resistance value of MOSFET are adjusted, and thus acquisition time is managed to get proper sampling.

## 4.8 Flash ADC Design

We have used the Flash-type ADC for the design. The sample and hold circuit output goes to the ADC's input, and we get the output in bit format. We have designed 5-bit ADC. The choice of 5-bit ADC is made according to the sensors' voltage ranges. However, larger ADCs can be designed if a finer resolution is required. It will increase the area of the design but will provide better accuracy. Flash type ADC takes the least amount of time for conversion from Analog to Digital data. It converts the data in a single clock cycle. Only delays present in the system are propagation delay of a comparator and logic circuit delay of the priority encoder.

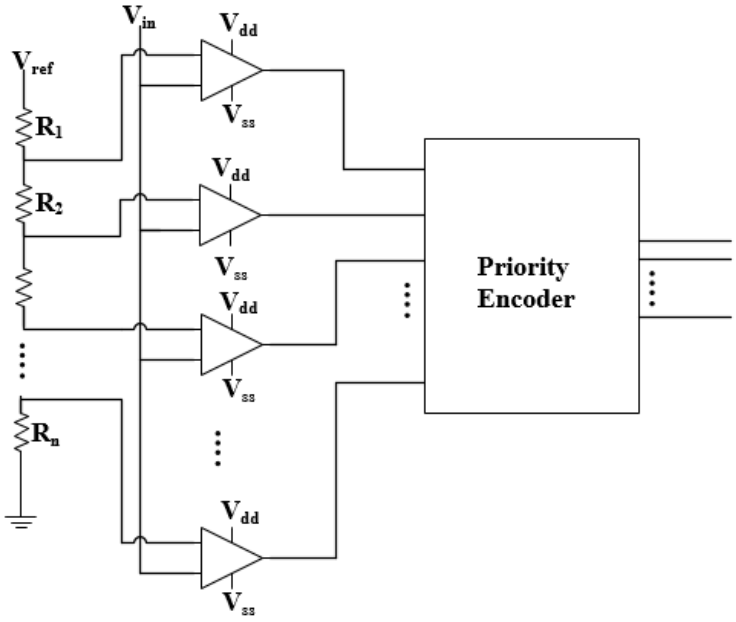


Figure 4.5 n-bit Flash ADC Circuit

## 4.9 Design Procedure of Flash Type ADC Circuit

As we have 5-bit ADC. So, in total $2^5$ = 32 different values. The values of the registers will be equal. There will be 32 comparators. Let's take R = 1K. With the value of the $V_{ref}$, we can calculate, what bits will appear at the output.
The total range of bits at the output will be 00000 to 11111. The lowest value corresponds to the input voltage greater than $V_{ref}/32$. The Highest value corresponds to the input voltage greater than $V_{ref}$ (31/32).

## 4.10 C to F Converter

The Design of the C to F converter is done for the Wetness sensor, which produces the data in terms of Capacitance variation and converted to frequency (e.g., voltage pulse), which can be given to counter for further processing. The counter counts the frequency, and thus the processor can read that to extract the value of wetness.

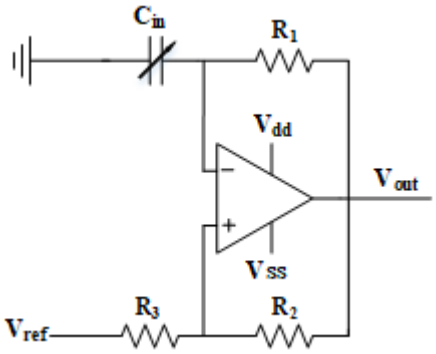


Figure 4.6 C to F converter

The design of a C to F converter is done using the assumption of all register values to be equal, and thus the overall frequency of the output pulse can be given via the equation below.

$$\mathbf{F} = \frac{\mathbf{1}}{\mathbf{2.2}\boldsymbol{RC}}$$

So, to make the base frequency 1 kHz. We have taken the value of the R = 30 K and C = 15 nF, respectively, using the equation.

## 4.11 Analog IC Design Flow

Analog IC design flow is the process of making the Analog circuit's final layout from the specification. It includes various steps as shown in fig 4.7.

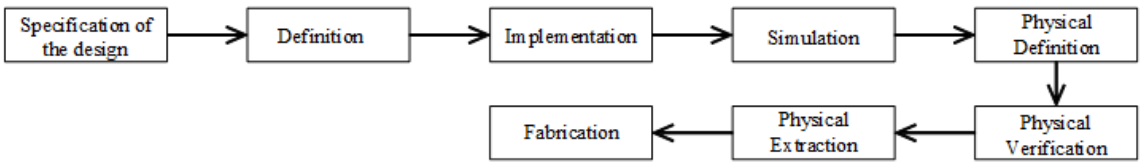


Figure 4.7 Analog IC Design Flow

## A) Specification of the Design

Analog components are designed according to the external connections. The different requirements of the connection can be anything. (i.e., Sensor Interfacing, Gain, Current boost, etc.) According to those requirements, we arrive at the specifications of the design. The specification will be supply voltage, delay, power consumption, etc.

## B) Definition

According to the specification, we decide which design structure we will implement. As there are several ways by which we can get the same functionality. But, with increasing requirements, we can go for better topologies or designs. We also decide which technology node we will use for the design.

## C) Implementation

After deciding on the structure, this is one of the most critical parts of the design flow. In this stage, we calculate the W/L rations of the design. The design equations for analog circuits are complex. Here, we need to have all the physical parameters of the technology node and transistors we will use to implement the design.

## D) Physical Definition

Once the design is successfully simulated, we check for the Physical aspects. The physical definition is a step where we enter the layout aspects of the design. For the simulation, we may have put the current source, resistors, and capacitors, which cannot be converted into the layout form directly; here, we must change/replace some components which can be put in the layout. We must do the placement and route process for the different connections, power, ground, clock, and pins.

## E) Physical Verification

After the routing process is completed, we check for any physical parameter violations that may have happened. The two primary checks are DRC and LVS. Where in DRC check, we verify that the physical lengths and widths of the metal layer, Via, clocks are according to the design rules.

In the LVS step, we confirm that the netlist was obtained from the layout and schematic matches. This step includes all the other checks for the correct functionality according to the specification and the design.

## F) Physical Extraction

In this step, we perform parasitic extraction. That refers to the information about the parasitic devices, which is not included as a part of the original circuit design. But these parasitic devices affect the circuit performance in several ways.

# Chapter 5
# Results and Conclusion

## 5.1 Discussion of Results

Two ASICs available in the design are verified on functionality and layout level. Extracted final results are shown here. The design and functionality check of the RISC-V design is done in the Xilinx ISE tool using Verilog HDL. The functionality for instructions was verified, and the code was made synthesizable. The Verilog design was tested for custom test benches. The simulation of the Verilog code can be seen in the figure.


Figure 5.1 Simulation result for RISC core

In the above figure, the unsigned decimal version of the four different instructions (given as a vector file to the Verilog code in initial block) is shown. The instructions are load, logic, store, and branch type. They perform all the operation as required. Implementation of the RISC-V processor's microarchitecture design gives the results shown in table 6. The results are presented in stages, the first data represent the results after the synthesis process, and the second data is of the core after the placement and route process. The cadence tools were used for the Physical Design flow. Cadence has a GUI interface to simplify the whole process of running synthesis, place, route, and all other related processes. The implementation is done using the gsclib045 file that is based on the 45 nm technology.

| Sr. No. | Parameter | Value (Synthesis) | Value (PNR) |
|---|---|---|---|
| 1 | Area ($um^2$) | 50402.59 | 50927.90 |
| 2 | Critical path delay (ps) | 335 | 361 |
| 3 | Power (uW) | 411.98 | 420.11 |

Table 5.1 Results of RISC core after synthesis and PNR

The difference between the data can be seen as area, power, and delay; all the parameters increase at the end because of the added interconnects, and their delay increases the total delay. The overall critical path delay is increased, and because of that, the core frequency is reduced. However, these changes are as expected.

The analog ASIC components are designed using the cadence library for 180 nm designs (gpdk180). The frequency response of the design is shown in fig. 17. AC analysis of the Op-Amp with capacitive load is shown in the figure below. Approximately 78 dB of gain is obtained, and the phase margin comes out to be 62.50 degrees. The transient analysis is also shown in fig. 18 for the designed Op-Amp, where we can see a very high gain output signal (1.5 V peak to peak) for the small-signal (5 mV peak to peak) at the input.

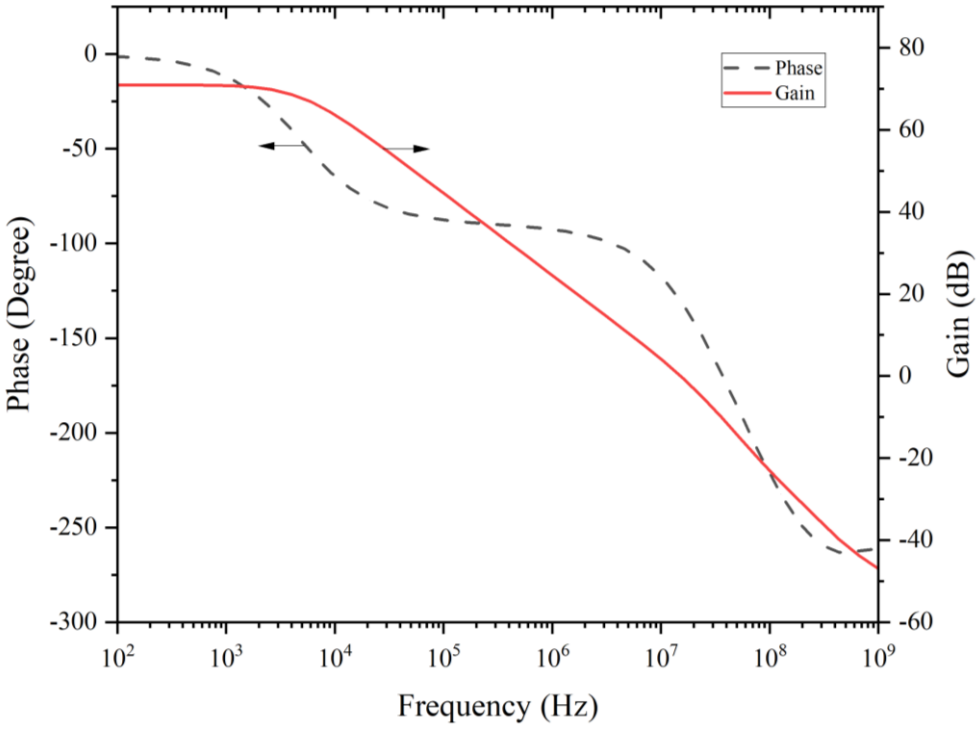


Figure 5.2 Frequency response of an Op-Amp

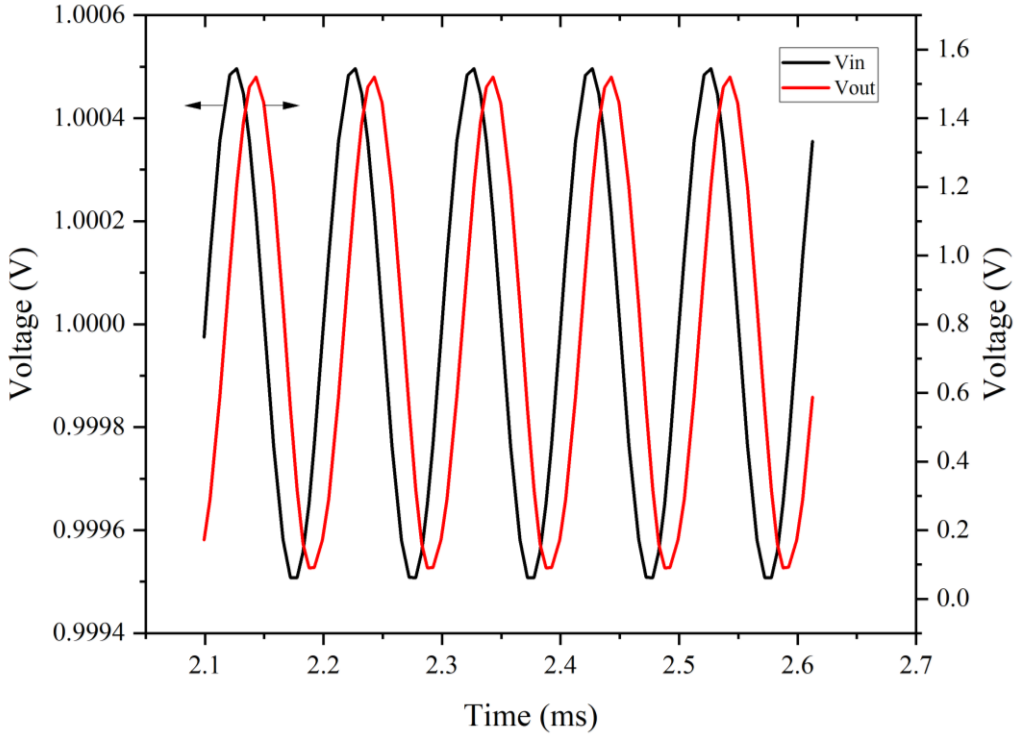


Figure 5.3 Transient response of an Op-Amp

The output of a comparator is shown in fig. 19. The constant voltage at an input gives rail to rail voltage and square wave output when the input sinusoidal signal crosses the fixed voltage. It generates output according to specifications. The output square wave is peak to peak 3.45 V (V+ = 1.68 V, V- = -1.77 V).

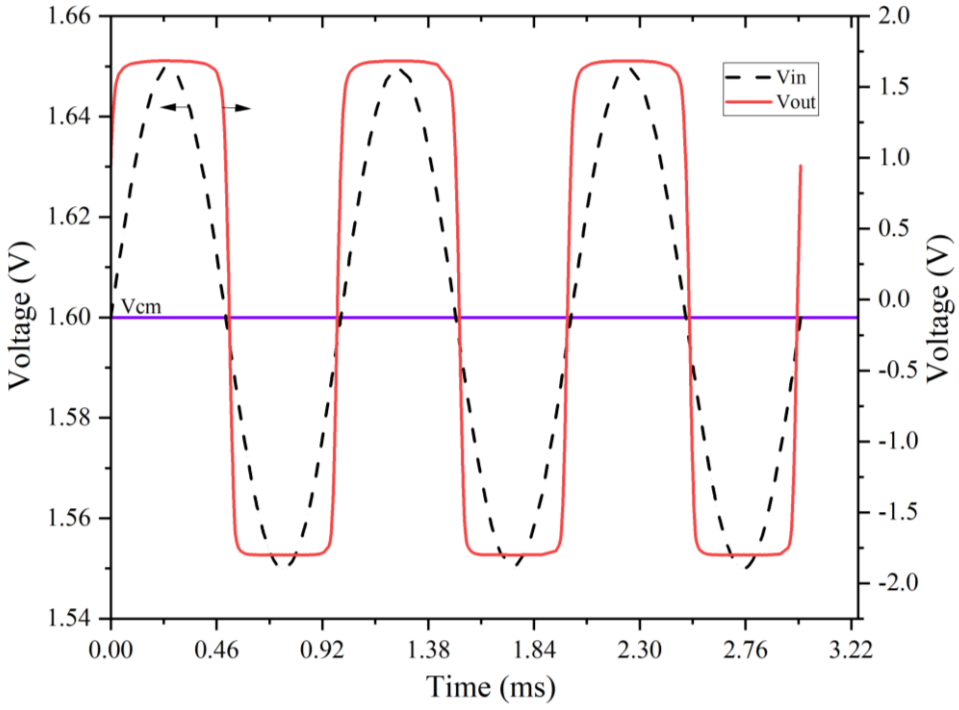


Figure 5.4 Transient response of a Comparator

The UGB and Sample and Hold circuits outputs are shown, and the designed Op-Amp is only used to make them. They also generate results as expected. The UGB generates the same signal as input with high current driving strength. The sample and hold circuit generates a sampled input version according to the sampling frequency(fs). For a very high sampling frequency, the output follows the input.

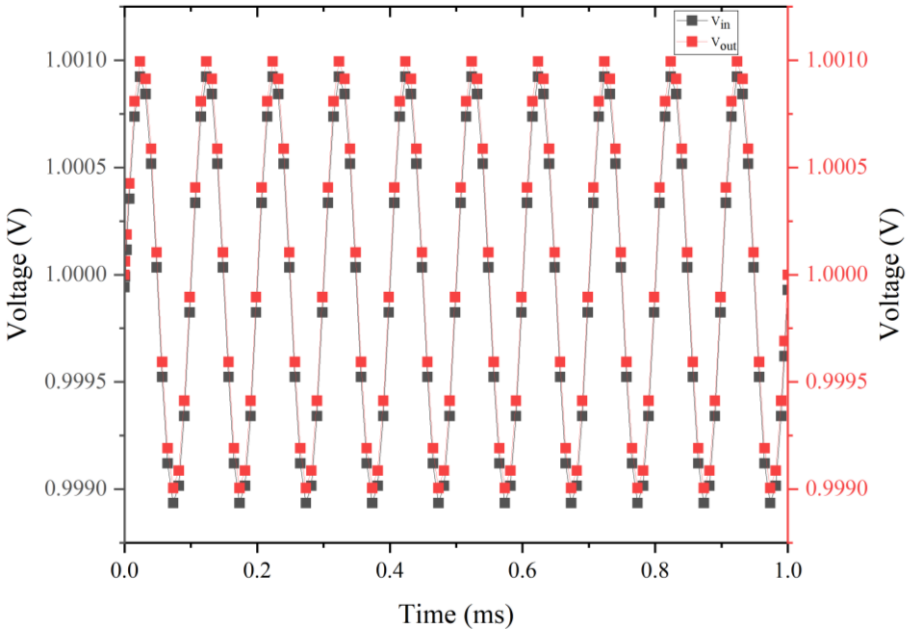


Figure 5.5 Transient Response of a UGB

According to the calculation we have done for selecting the resistance and capacitance value, the C to F converter generates the output in pulses with a frequency of 1.1 kHz. The final layouts of the design are also shown for all the designed parts. Table 7 shows design parameters of the analog ASIC.

| Sr. No. | Parameter | Value |
|---|---|---|
| 1 | Area ($mm^2$) | 2.365 |
| 2 | Propagation delay (us) | 1.92 |
| 3 | Power (mW) | 102.24 |

Table 5.2 Results of analog ASIC

The final layout after placement and route process for the RISC core is shown. The core pins can be shown at the boundary. The power and ground rings and stripes are shown. The total standard cell count in the design is 11K.

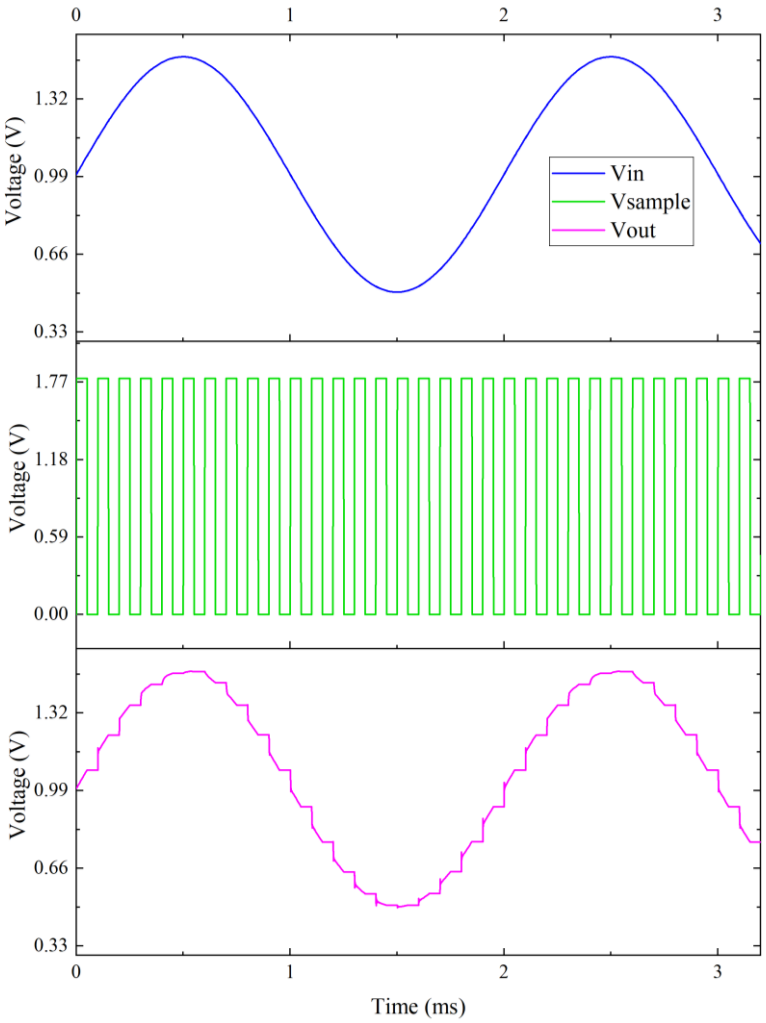


Figure 5.6 The output of a Sample and Hold Circuit

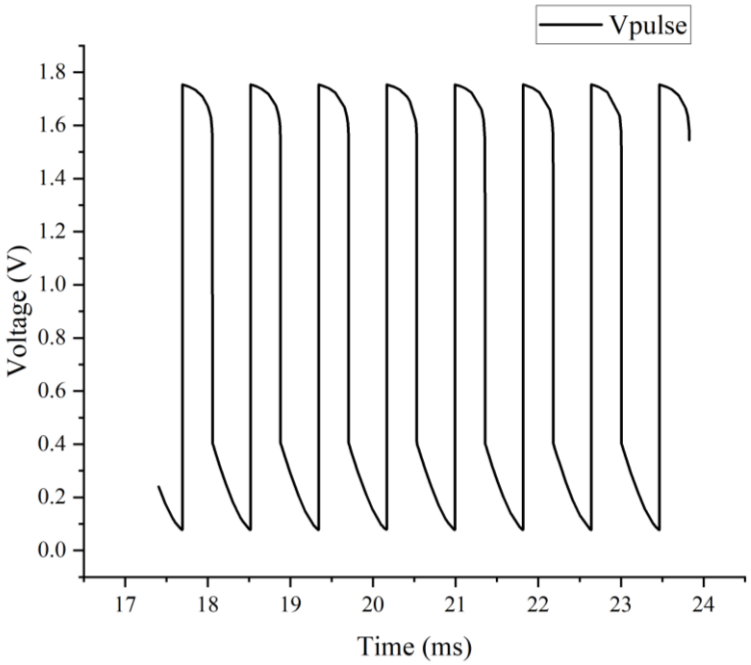


Figure 5.7 The output of the C to F converter

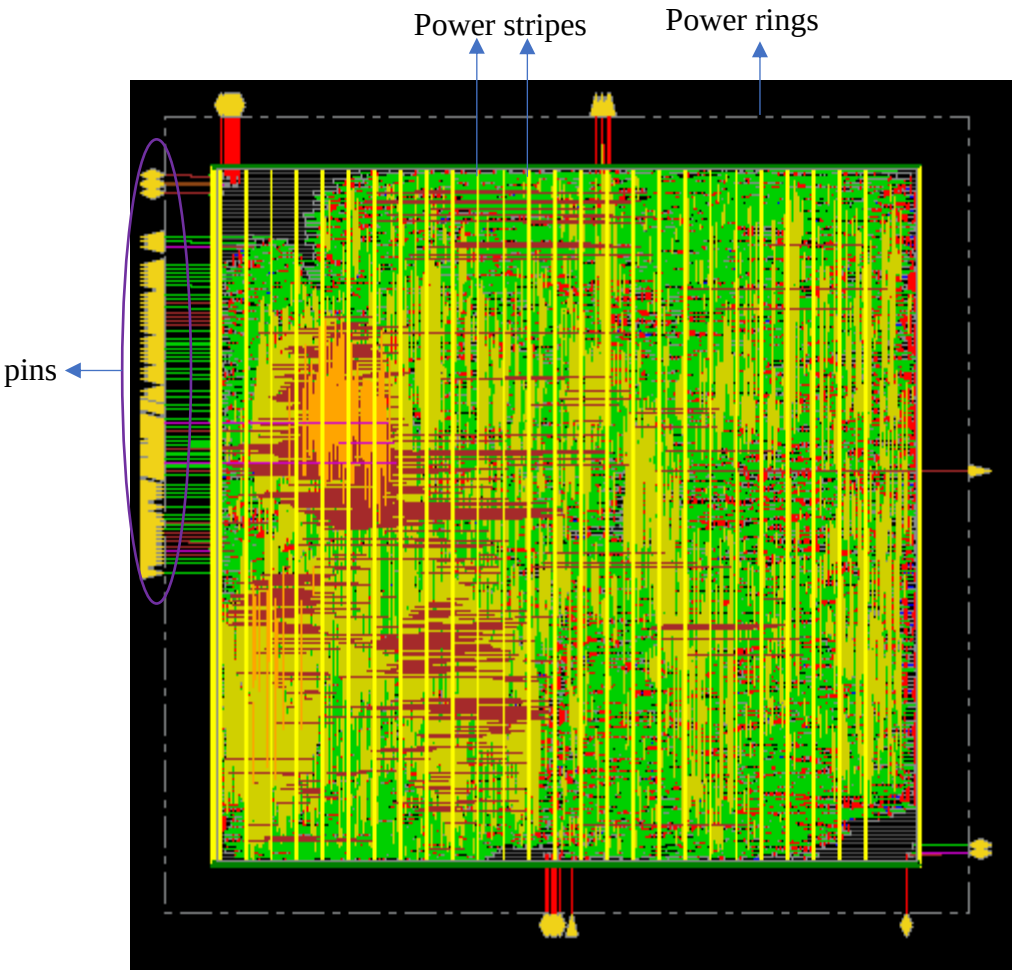


Figure 5.8 Layout of the RISC core

The layout for each analog component is developed. Each of the component's layouts is generated using a virtuoso Layout editor. They are auto placed and routed in the layout editor. After that, the DRC, LVS, and parasitic extraction steps are performed. Then the GDSII file is generated for all the components. Thus, the functionality and final layouts are verified. The final analog layout is shown here, where all the individually designed components are combined on the same chip. The total area of the analog layout is 2.365 $mm^2$. The horizontal width is 1385 um, and the vertical length is 1708.39 um. The full metal length used for the 75% utilization in the PR boundary is 23358.53 um. The horizontal metal length used is 6914.49, and the Vertical metal length used is 16444.04 um.

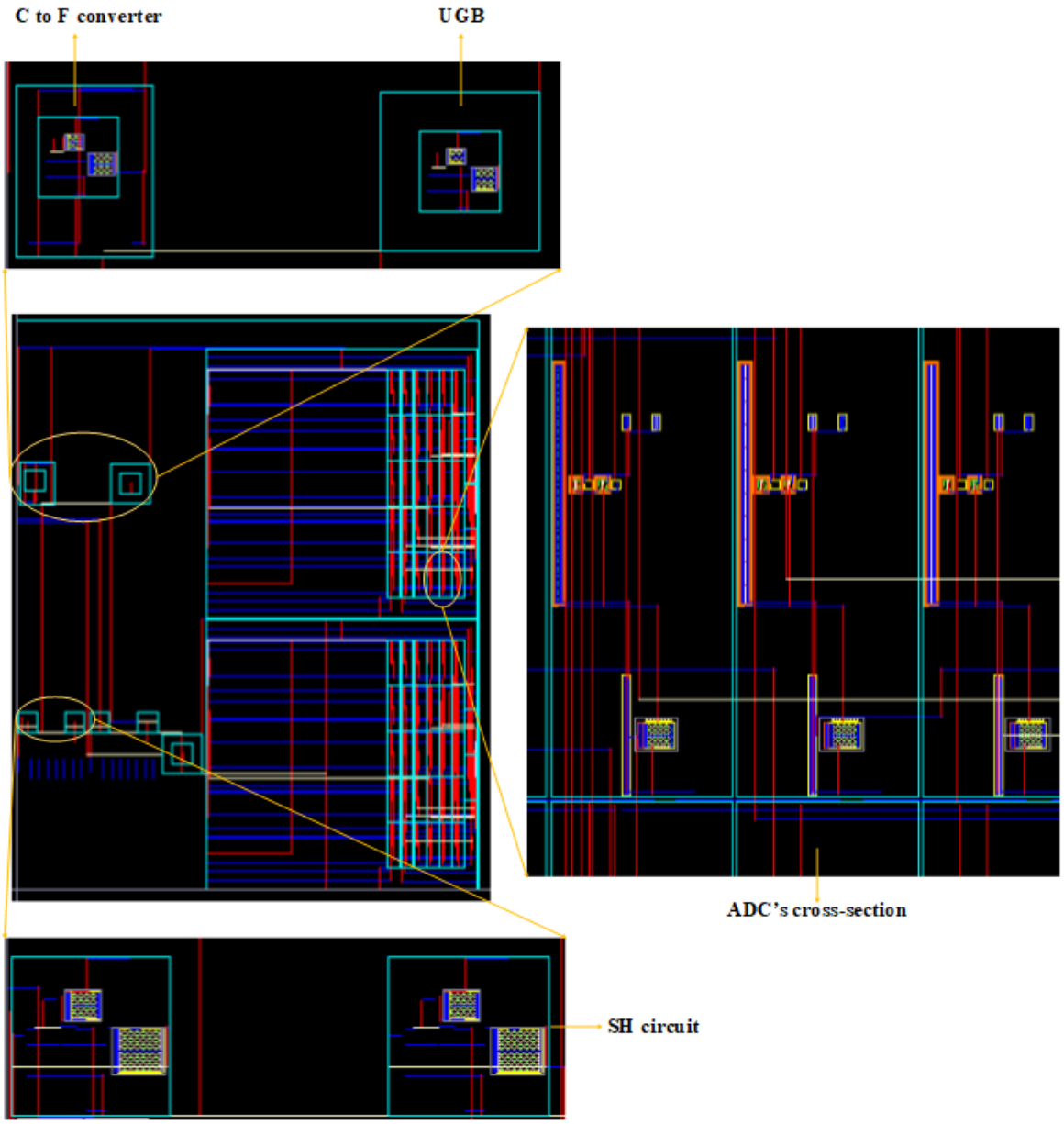


Figure 5.9 Layout of the Analog ASIC

## 5.2 AMS functionality implementation for ADC

The design of ADC is done using a mixed-signal design flow of cadence tools, where virtuoso and xcelium tools are used for analog and digital simulations, respectively. The comparator is used and designed, as shown in the above section. The priority encoder is developed using Verilog HDL. It is known as soft IP, which replicates the circuit's behaviour. The functionality is verified using the cadence virtuoso tool, which allows us to work with mixed-signal circuits. The figure below shows the schematic of the 5-bit implementation of Flash type ADC

to test the AMS functionality. The outputs of the ADC can be seen in the figure below. Five-bit ADC will have 32 comparators and 32 to 5 priority encoder.

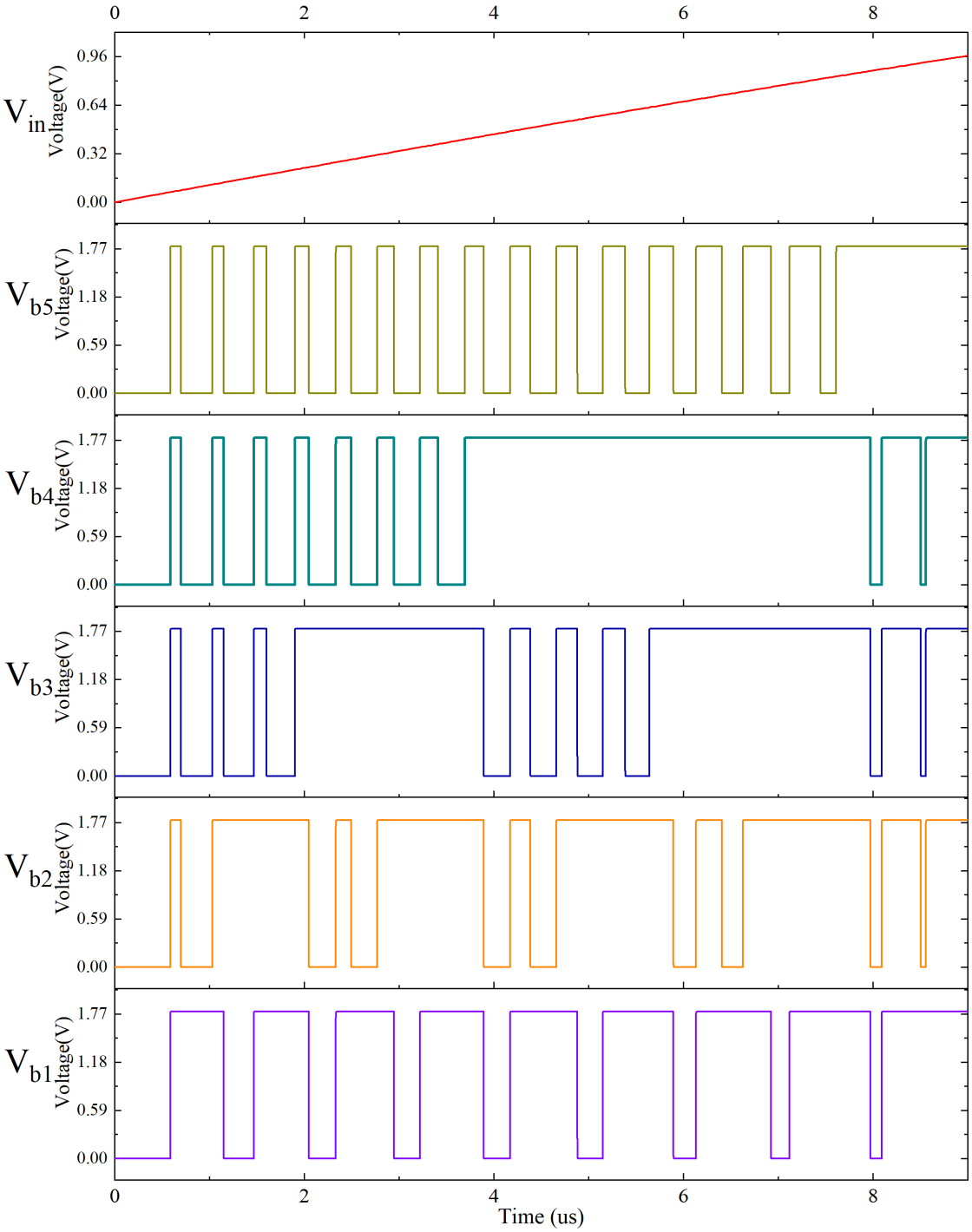


Figure 5.10 AMS functionality verification of ADC

## 5.3 Comparison of ASICs with other recent states of the art

Table 8 shows the comparison between the different RISC cores. The RISC core developed by us is the light as intended because we want to use it for ASIC applications.

| | This work | [23] | [23] | [24] | [25] |
|---|---|---|---|---|---|
| Technology | 45 nm | 28 nm FDSOI | 28 nm FDSOI | 65 nm | 45 nm |
| Instruction Set | RV32I | RV32IM | RV64G | 16-bit MSP430 | RISC-V Rocket |
| Pipeline | 5 | 3 | 5 | 3 | 6 |
| Standard Cell Count (K) | 11 | 18 | 196 | - | 192 |
| Area ($mm^2$) | 0.0509 | 0.053 | 0.461 | 0.42 | 1.45 |
| Frequency (Hz) | 2.77 G | - | 797 M | 25 M | 1.3 G |
| Power (W) | 420 u | - | 1200 u | 700 u | 20 m (multi core) |

Table 5.3 Comparison of RISC core with other recent states of the art

The other versions are shown where more extensive architecture or the implementation in the higher and Lower technology nodes are given. Our work is better in terms of all design matrices, Area, Power, and Delay. It, of course, is compared to larger architectures, but the lighter version was intended where we implement limited instructions that can be used for our application.

For the analog ASIC part, we could not find the exact same implementation as we have done in the recent state of the art to the best of our knowledge. However, the designed Op-Amp is compared to the recent work [26] and [27]. The gain-bandwidth product of our design is more than double in comparison (10 MHz), but the gain is 6 dB less in comparison (78 dB). That is because of the trade-off. However, the phase margin is approximately the same (difference of 2 degrees). Table 9 shows the comparison between different op-Amp designs.

| | This work | [26] | [27] |
|---|---|---|---|
| Gain (dB) | 78 | 84 | 84 |
| Gain bandwidth (MHz) | 10 | 4.95 | 4.53 |
| Phase Margin (Degree) | 62.5 | 64.5 | 66.3 |

Table 5.4 Comparison of an Op-Amp with other recent state of the art

## 5.4 Conclusion

In this work, we have successfully designed the proposed analog and digital ASICs. Our RISC core has a 0.0509 $mm^2$ area with a total power consumption of 420 uW. The maximum operating frequency obtained from the critical path delay is 2.77 GHz. Our analog ASIC has a total area of 2.365 $mm^2$. Future agriculture will become primarily dominated by electronics. The transition from traditional to modern farming is only possible by better controlling and understanding the factors affecting agriculture. And that can be supported by innovations in IC design. Chips are becoming more intelligent, smaller, faster, and can run longer on a single battery with increasing ASIC designs. To show that, we have developed the system with a mixed-signal ASIC design. Our ASICs are designed on two different technology nodes, i.e. (Analog ASIC is designed in 180 nm, and Digital ASIC is designed in 45 nm). It is a multinode mix-signal design. The two different ASICs can be fabricated, and then finally, integration can be done on the SoC design. Thus, proposed ASICs will enable us to create the SoC for the complete agriculture system monitoring and control through IoT.

# References


1. K. S. Patle, R. Saini, A. Kumar, S. G. Surya, V. S. Palaparthy and K. N. Salama, "IoT Enabled, Leaf Wetness Sensor on the Flexible Substrates for In-Situ Plant Disease Management," in IEEE Sensors Journal, vol. 21, no. 17, pp. 19481-19491, 1 Sept.1, 2021, doi: 10.1109/JSEN.2021.3089722.
2. J. Patidar, R. Khatri and R. C. Gurjar, "Precision Agriculture System Using Verilog Hardware Description Language to Design an ASIC," 2019 3rd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), 2019, pp. 1-6, doi: 10.1109/IEMENTech48150.2019.8981128.
3. Madhukar BM, Reddy VP. Design and Development of FPGA Based Wireless SoC for Precision Agriculture. European Journal of Molecular & Clinical Medicine. 2021 Feb 11;8(1):1687-706.
4. Kumar PM, Kumar TG, Aravindhan R, Sairam D, Moorthi S. Design and Development of an Embedded Framework to test Soil for Urban
5. Terrace Farming. In2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE) 2020 Feb 7 (pp. 41-45). IEEE.
6. Vedulla T, Reddy YM, Kalyan A, Jenila R. VLSI Architecture for Smart and Precision Agriculture Using Sensors. International Journal of Advanced Science Computing and Engineering. 2021 Jun 28;3(1):18-27.
7. Jaiswal SP, Bhadoria VS, Agrawal A, Ahuja H. Internet of Things (IoT) for smart agriculture and farming in developing nations. International Journal of Scientific & Technology Research. 2019;8(12):1049-56.
8. ZHAN ZH, HAN NC, JIANG WQ, WANG ZF, LIU H. Study of Parallel-plate Capacitor Detecting Soil Moisture Based on frequency Characteristic. Science Technology and Engineering. 2017;2017:21.
9. Dhall R, Agrawal H. An improved energy efficient duty cycling algorithm for IoT based precision agriculture. Procedia Computer Science. 2018 Jan 1;141:135-42.
10. Abdellah NA, Thangadurai N. Real Time Application of IoT for the Agriculture in the Field along with Machine Learning Algorithm. In2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE) (pp. 1-6). IEEE.
11. Sharma M, Singh G. Need and Design of Smart and Secure Energy-Efficient IoT-Based Healthcare Framework. In Energy Conservation for IoT Devices 2019 (pp. 259-281). Springer, Singapore.

12. Delipinar T, Shafique A, Gohar MS, Yapici MK. Fabrication and Materials Integration of Flexible Humidity Sensors for Emerging Applications. ACS omega. 2021 Mar 25;6(13):8744-53.
13. Balan T, Dumitru C, Dudnik G, Alessi E, Lesecq S, Correvon M, Passaniti F, Licciardello A. Smart Multi-Sensor Platform for Analytics and Social Decision Support in Agriculture. Sensors. 2020 Jan;20(15):4127.
14. Ayaz M, Ammad-Uddin M, Sharif Z, Mansour A, Aggoune EH. Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk. IEEE Access. 2019 Aug 1;7:129551-83.
15. Shah H, Siderius C, Hellegers P. Cost and effectiveness of in-season strategies for coping with weather variability in Pakistan's agriculture. Agricultural Systems. 2020 Feb 1;178:102746.
16. Shanmukhi M, Tabassum N, Adilakshmi T, Rao AS. SUSTAINABLE AND INTELLIGENT IOT BASED PRECISION AGRICULTURE–SMART FARMING.
17. D. Novo, A. Kritikakou, P. Raghavan, L. Van der Perre, J. Huisken and F. Catthoor, "Ultra low energy Domain Specific Instruction-set Processor for on-line surveillance," 2010 IEEE 8th Symposium on Application Specific Processors (SASP), 2010, pp. 30-35, doi: 10.1109/SASP.2010.5521151.
18. P. M. Kumar, T. Ganesh Sai Kumar, R. Aravindhan, D. Sairam and S. Moorthi, "Design and Development of an Embedded Framework to test Soil for Urban Terrace Farming," 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), 2020, pp. 41-45, doi: 10.1109/ICETCE48199.2020.9091731.
19. A. Jog, Y. Avigal, A. Avital, J. Amble, A. Peled and Y. Shacham-Diamand, "An Integrated Electronic Interface for Bio-electrochemical Plant-based Sensors," 2020 IEEE SENSORS, 2020, pp. 1-4, doi: 10.1109/SENSORS47125.2020.9278777.
20. W. Gao, T. Gao, R. Zhao, L. Wang, and X. Liang, "Routability-driven legalization algorithm with minimum total movement for supporting agricultural machinery design," 2017 6th International Conference on Agro-Geoinformatics, 2017, pp. 1-5, doi: 10.1109/Agro-Geoinformatics.2017.8047058.
21. T. K. Yew, Y. Yusoff, L. K. Sieng, H. C. Lah, H. Majid and N. Shelida, "An electrochemical sensor ASIC for agriculture applications," 2014 37th International Convention on Information and Communication

Technology, Electronics and Microelectronics (MIPRO), 2014, pp. 85-90, doi: 10.1109/MIPRO.2014.6859538.
22. Samuel K. Moore. 3 Ways 3D Chip Tech Is Upending Computing. IEEE spectrum. 2022 March 16
23. B. Keller et al., "A RISC-V processor SoC With Integrated Power Management at Submicrosecond Timescales in 28 nm FD-SOI," in IEEE Journal of Solid-State circuits, vol. 52, no. 7, pp. 1863-1875, July 2017, doi: 10.1109/JSSC.2017.2690859.
24. D. Bol et al., "SleepWalker: A 25-MHz 0.4-V Sub- $\hbox{mm}^{2}$ 7-$\mu\hbox{W/MHz}$ Microcontroller in 65-nm LP/GP CMOS for Low-Carbon Wireless Sensor Nodes," in IEEE Journal of Solid-State circuits, vol. 48, no. 1, pp. 20-32, Jan. 2013, doi: 10.1109/JSSC.2012.2218067.
25. Y. Lee et al., "A 45nm 1.3GHz 16.7 double-precision GFLOPS/W RISC-V processor with vector accelerators," ESSCIRC 2014 - 40th European Solid State circuits Conference (ESSCIRC), 2014, pp. 199-202, doi: 10.1109/ESSCIRC.2014.6942056.
26. C. Chanapromma and J. Mahattanakul, "Improved Design Procedure for Two-Stage CMOS Op Amp Employing Current Buffer," 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2020, pp. 384-387, doi: 10.1109/ECTI-CON49241.2020.9158307.
27. S. Pal and P. Chosh, “Design and simulation of two-stage low-power CMOS op-amp in nanametre range,” Computational advanced in communication circuits and systems, Lecture notes in electrical engineering 335, Srpinger 2015, pp. 425-432, 2015.
28. N. Tanabe et al., "Preliminary evaluations of a FPGA-based-prototype of DIMMnet-2 network interface," Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA'05), 2005, pp. 9 pp.-, doi: 10.1109/IWIA.2005.38.
29. R. Kress and S. Schulz, "Robust mobile computing-a case study in architecture performance evaluation," Proceedings of 13th Annual IEEE International ASIC/SOC Conference (Cat. No.00TH8541), 2000, pp. 143-147, doi: 10.1109/ASIC.2000.880691.
30. H. Liu and J. Zhou, "Optimization of novel Interdigitated Electrode Structure for Capacitive Humidity Sensing," 2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), 2018, pp. 1-3, doi: 10.1109/ICSICT.2018.8565714.

31. Allen, Phillip E., and Douglas R. Holberg. CMOS Analog Circuit Design. New York: Oxford University Press, 2002.

| | | | | | | |
|---|---|---|---|---|---|---|
| imm[31:12] | | | | rd | 0110111 | LUI |
| imm[31:12] | | | | rd | 0010111 | AUIPC |
| imm[20\|10:1\|11\|19:12] | | | | rd | 1101111 | JAL |
| imm[11:0] | | rs1 | 000 | rd | 1100111 | JALR |
| imm[12\|10:5] | rs2 | rs1 | 000 | imm[4:1\|11] | 1100011 | BEQ |
| imm[12\|10:5] | rs2 | rs1 | 001 | imm[4:1\|11] | 1100011 | BNE |
| imm[12\|10:5] | rs2 | rs1 | 100 | imm[4:1\|11] | 1100011 | BLT |
| imm[12\|10:5] | rs2 | rs1 | 101 | imm[4:1\|11] | 1100011 | BGE |
| imm[12\|10:5] | rs2 | rs1 | 110 | imm[4:1\|11] | 1100011 | BLTU |
| imm[12\|10:5] | rs2 | rs1 | 111 | imm[4:1\|11] | 1100011 | BGEU |
| imm[11:0] | | rs1 | 000 | rd | 0000011 | LB |
| imm[11:0] | | rs1 | 001 | rd | 0000011 | LH |
| imm[11:0] | | rs1 | 010 | rd | 0000011 | LW |
| imm[11:0] | | rs1 | 100 | rd | 0000011 | LBU |
| imm[11:0] | | rs1 | 101 | rd | 0000011 | LHU |
| imm[11:5] | rs2 | rs1 | 000 | imm[4:0] | 0100011 | SB |
| imm[11:5] | rs2 | rs1 | 001 | imm[4:0] | 0100011 | SH |
| imm[11:5] | rs2 | rs1 | 010 | imm[4:0] | 0100011 | SW |
| imm[11:0] | | rs1 | 000 | rd | 0010011 | ADDI |
| imm[11:0] | | rs1 | 010 | rd | 0010011 | SLTI |
| imm[11:0] | | rs1 | 011 | rd | 0010011 | SLTIU |
| imm[11:0] | | rs1 | 100 | rd | 0010011 | XORI |
| imm[11:0] | | rs1 | 110 | rd | 0010011 | ORI |
| imm[11:0] | | rs1 | 111 | rd | 0010011 | ANDI |
| 0000000 | shamt | rs1 | 001 | rd | 0010011 | SLLI |
| 0000000 | shamt | rs1 | 101 | rd | 0010011 | SRLI |
| 0100000 | shamt | rs1 | 101 | rd | 0010011 | SRAI |
| 0000000 | rs2 | rs1 | 000 | rd | 0110011 | ADD |
| 0100000 | rs2 | rs1 | 000 | rd | 0110011 | SUB |
| 0000000 | rs2 | rs1 | 001 | rd | 0110011 | SLL |
| 0000000 | rs2 | rs1 | 010 | rd | 0110011 | SLT |
| 0000000 | rs2 | rs1 | 011 | rd | 0110011 | SLTU |
| 0000000 | rs2 | rs1 | 100 | rd | 0110011 | XOR |
| 0000000 | rs2 | rs1 | 101 | rd | 0110011 | SRL |
| 0100000 | rs2 | rs1 | 101 | rd | 0110011 | SRA |
| 0000000 | rs2 | rs1 | 110 | rd | 0110011 | OR |
| 0000000 | rs2 | rs1 | 111 | rd | 0110011 | AND |
| 0000 \| pred \| succ | | 00000 | 000 | 00000 | 0001111 | FENCE |
| 0000 \| 0000 \| 0000 | | 00000 | 001 | 00000 | 0001111 | FENCE.I |
| 000000000000 | | 00000 | 000 | 00000 | 1110011 | ECALL |
| 000000000001 | | 00000 | 000 | 00000 | 1110011 | EBREAK |

I RV32I Instructions

| | NMOS | PMOS |
|---|---|---|
| Lambda ( ∧ ) | 0.01 | 0.01 |
| Transconductance (K′) | 138 * 10-6 | 69 * 10-6 |
| Threshold Voltage (Vth) | 0.48 | 0.43 |

II Model Parameters for 180 nm process