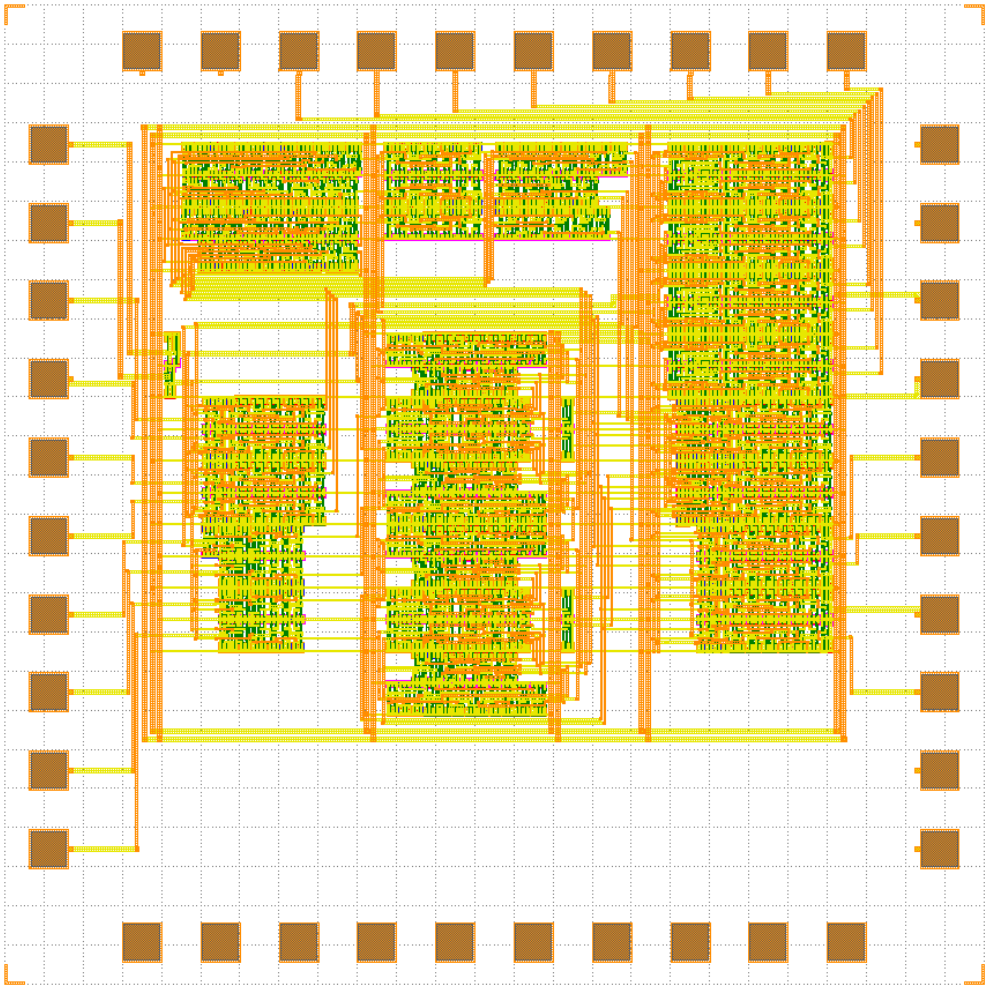
Design of a 4-bit RISC Processor in Silicon Carbide Technology

Master's Thesis



Jianeng Xu

Design of a 4-bit RISC Processor in Silicon Carbide Technology

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Jianeng Xu
born in Wuxi, China

The Electronic Components, Technology and Materials Group
Department of Electrical Engineering
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Design of a 4-bit RISC Processor in Silicon Carbide Technology

Author: Jianeng Xu
Student id: 5940311

Abstract

Silicon carbide (SiC) is a wide-bandgap semiconductor with excellent thermal stability, high breakdown voltage, and robustness under harsh environments, making it well-suited for high-temperature digital logic applications. Compared with conventional silicon, SiC devices maintain reliable operation at significantly higher temperatures due to their strong electric field tolerance and reduced leakage characteristics. However, as an emerging technology, its integration level for digital circuits is still in a preliminary stage and requires validation through a full chip design and fabrication process.

This thesis presents the design and simulation of a fully functional 4-bit RISC processor in a 2 μm SiC CMOS process developed by Fraunhofer IISB. The processor includes the implementation of the ALU, program counter, control unit, and memory blocks, together with validation of its functionality and performance through circuit-level simulations across a wide temperature range. Results demonstrate stable operation and robustness up to 500°C, as well as reliable logic performance at clock frequencies up to 750 kHz. The work further establishes a foundation for future digital systems based on the Fraunhofer IISB SiC CMOS process, paving the way toward more complex architectures such as 16-bit processors for real-world harsh-environment applications.

Thesis Committee:

Chair:	Dr. Ir. Sten Vollebregt, Faculty EEMCS, TU Delft
University supervisor:	Dr. Ir. Sten Vollebregt, Faculty EEMCS, TU Delft
Committee Member:	Dr. Chang Gao, Faculty EEMCS, TU Delft

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Rapid advancements in fields such as aerospace, nuclear energy, and automotive industries demand electronic systems capable of reliable operation under extreme environmental conditions. These conditions include high temperatures, high radiation levels, and severe vibrations. However, conventional silicon-based semiconductor technology faces its physical limits. In terms of breakdown voltage, thermal tolerance, and operating frequency, silicon technology struggles to meet the requirements of these harsh applications.

Among various wide-bandgap (WBG) semiconductors, silicon carbide (SiC) attracts significant attention due to its excellent properties. SiC exhibits outstanding thermal stability, high breakdown voltage, and robustness in harsh environments [30]. Crucially, SiC is also unique among WBGs for having an established complementary transistor (CMOS) platform [30]. These characteristics make it an ideal choice for high-temperature digital logic applications. Compared with traditional silicon, SiC devices can maintain reliable operation at much higher temperatures. This is thanks to their stronger electric field tolerance and significantly reduced leakage current characteristics.

Despite SiC's clear theoretical advantages and some successful demonstrations of SiC-based integrated circuits (ICs), it remains an emerging technology. This is particularly true for digital circuit integration, which is still in a preliminary stage. Although the feasibility of SiC CMOS technology has been proven, aspects like device performance, process maturity, and the associated design toolchain lag behind mature silicon technologies. Therefore, validation through a complete chip design, fabrication, and testing process is essential. Such validation is needed to ascertain the true potential and reliability of SiC CMOS technology for realizing complex digital systems, like processors.

Designing and implementing a SiC-based processor serves not only as an effective vehicle for validating the technology but also lays the groundwork for deploying more advanced computation and control systems in extreme environ-

ments. Potential applications include sensor interfaces for deep-space exploration missions, downhole electronics for geothermal exploration, or high-temperature controllers within the powertrain of electric vehicles. This thesis aims to address this challenge by designing a 4-bit RISC processor, thereby providing practical evidence for the feasibility of SiC digital system-level integration.

1.2 Research Objective

While SiC CMOS technology offers significant potential for high-temperature digital applications, its current development faces practical challenges. Specific to the Fraunhofer IISB 2 μm 4H-SiC CMOS process used in this research, its status as an emerging technology presents notable limitations for digital circuit integration. Firstly, constrained by process capabilities, the technology features a relatively low transistor density, restricting the available logic resources within a limited die area. Secondly, it lacks a mature automated digital design flow and comprehensive Process Design Kit (PDK) support. This means standard automated RTL-to-layout toolchains cannot be directly applied, complicating design complexity and verification. Under these constraints, designing a functional processor to validate the technology's system-level integration capability is inherently challenging.

To address these challenges and verify the potential of SiC CMOS technology in digital systems, this thesis sets out the following specific tasks:

- Explore and define an "ultra-minimalist" processor architecture suitable for the limitations of this SiC process.
- Establish a hybrid design and verification flow combining existing open-source tools with manual circuit-level design.
- Develop and characterize a basic SiC CMOS digital standard cell library, optimizing for high-temperature performance and area efficiency.
- Verify the processor's functional correctness and timing performance across a wide temperature range, from room temperature up to 500 °C, through detailed circuit-level simulations.
- Complete the full-custom physical layout design of the processor with process design rules compliance, adherence to die area constraints, and targeted layout optimization.

Therefore, the overall objective of this study is to design, simulate, and complete the layout implementation of a 4-bit RISC processor based on the Fraunhofer IISB 2 μm 4H-SiC CMOS process, verifying its potential for reliable operation at extreme temperatures up to 500 °C. Accomplishing this objective will establish a solid technical foundation for the future development of more complex digital systems based on this SiC process.

1.3 Thesis Outline

The thesis comprises seven chapters:

1. Introduction: This chapter states the research motivation, defines the problem, and sets the research objectives.
2. Literature Review: This chapter reviews SiC material and process characteristics, along with the state-of-the-art in SiC digital circuits and processors.
3. Processor Architecture Design: This chapter introduces the ultra-minimalist design philosophy, defines the ISA, and describes hardware modules including the ALU, registers, PC, and FSM control unit.
4. Logical Design and Functional Verification: This chapter covers RTL implementation, behavioral simulation verification, logic synthesis, and timing optimization.
5. Circuit-Level Design and Simulation: This chapter discusses transistor sizing, standard cell library design, module circuit implementation, high-temperature simulation, and performance analysis.
6. Layout Design and Verification: This chapter details standard cell and module layouts, top-level place and route strategies, and DRC/LVS verification.
7. Conclusions and Future Work: This chapter summarizes the main contributions and looks ahead to future research directions such as fabrication testing, flow optimization, and architecture expansion.

Chapter 2

Literature Review

The literature review establishes the technological and research background for this work. It first introduces the material and device properties of 4H-SiC relevant to CMOS integration, then surveys the evolution of SiC digital circuit design, and finally summarizes previous processor-level demonstrations that form the foundation for the 4-bit RISC processor developed in this thesis.

2.1 Properties of SiC Devices for CMOS Integrated Circuits

Rapid advancements in aerospace, nuclear, and automotive technologies have created a growing demand for electronic systems capable of reliable operation under extreme environmental conditions. However, conventional silicon–based semiconductor technology is approaching its physical limits in breakdown voltage, thermal tolerance, and operating frequency, restricting its suitability for such harsh applications. Among various wide-bandgap semiconductors, silicon carbide (SiC) uniquely combines strong theoretical advantages with a relatively mature CMOS platform that has demonstrated reliable IC operation at elevated temperatures. This section reviews the fundamental material and device characteristics underlying SiC CMOS circuit design and introduces the specific process technology used in this work—the Fraunhofer IISB 4H-SiC CMOS process.

2.1.1 Fundamental Material Properties of 4H-SiC

Silicon Carbide is a compound semiconductor composed of silicon (Si) and carbon (C), classified as a wide bandgap (WBG) material. Among its more than 250 crystalline forms, or polytypes, the 4H-SiC variant has become the material of choice for high-power, high-frequency, and high-temperature electronic devices due to its superior electrical characteristics [21, 30]. Compared to conventional Si, 4H-SiC demonstrates significant advantages across several key physical properties.

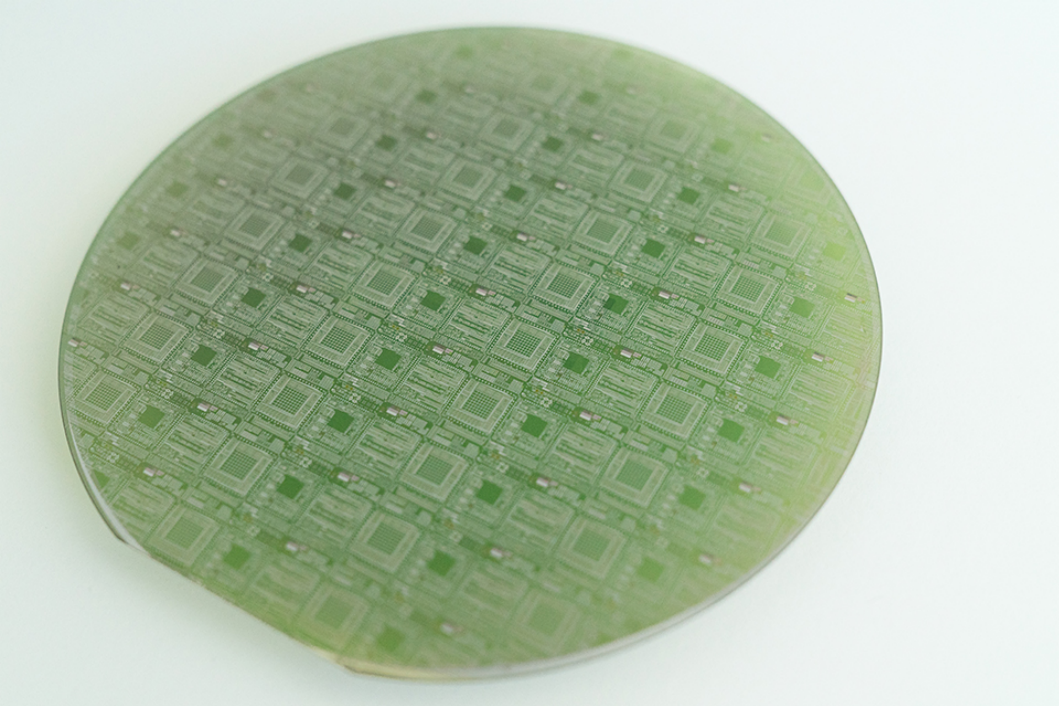


Figure 2.1: 4H-SiC wafer with sensor arrays and integrated circuits in the Fraunhofer IISB technology [3]

Wide Bandgap

The most fundamental advantage of 4H-SiC is its wide bandgap, approximately 3.26 eV, which is nearly three times that of silicon's 1.12 eV [21, 30]. This property is the cornerstone of its high-temperature capability. A wider bandgap results in an extremely low intrinsic carrier concentration, meaning that far fewer charge carriers are generated by thermal energy. This dramatically suppresses leakage currents, which are a primary failure mechanism for silicon devices as temperature rises. Consequently, SiC devices can operate reliably at temperatures far exceeding silicon's typical limit of 150-175°C, with demonstrated functionality at 600°C and beyond [22]. Furthermore, a wider bandgap means a higher electric field is required to cause avalanche breakdown, contributing to the material's superior high-voltage performance [1, 8].

High Critical Electric Field

4H-SiC possesses a critical electric field strength of about 2.2 MV/cm, nearly ten times higher than that of silicon ($\sim$0.3 MV/cm) [30]. This has a profound impact on the design of power devices. For a device to block a certain voltage, it requires a drift region of a specific thickness. Because SiC can withstand a much higher electric field, it requires a significantly thinner drift region to block the same voltage

compared to a silicon device. This directly leads to a lower specific on-resistance (R_{on}) and, consequently, a smaller required chip area for a given current and voltage rating. This reduction in size and resistance enables higher power density and faster switching speeds [7].

High Thermal Conductivity

In addition to its electrical properties, 4H-SiC exhibits excellent thermal conductivity of approximately 3.7 W/cm·K, more than double that of silicon (∼1.5 W/cm·K) [21, 30]. This property is crucial for reliability in high-power and high-temperature applications. It allows heat generated during device operation to be dissipated much more efficiently, lowering the junction temperature of the chip. This superior thermal management capability not only enhances the device's long-term reliability and operational lifetime but also reduces the dependency on bulky external cooling systems, facilitating system miniaturization [11].

Table 2.1: Comparison of key material properties between 4H-SiC and Silicon.

Property	**4H-SiC**	**Silicon (Si)**
Bandgap (eV)	∼3.26	∼1.12
Critical Electric Field (MV/cm)	∼2.2	∼0.3
Thermal Conductivity (W/cm·K)	∼3.7	∼1.5
Electron Sat. Drift Velocity (10^7 cm/s)	∼2.0	∼1.0
Channel Electron Mobility (cm^2/V·s)	< 50	> 400

Disadvantages and Processing Challenges

Despite its advantages, SiC technology is significantly less mature than silicon, presenting several key challenges in material growth and device fabrication.

- **Material Quality and Cost**: SiC substrate manufacturing is more complex and costly. SiC wafers have a higher density of crystallographic defects that can degrade device performance. Furthermore, SiC wafers are smaller, limiting manufacturing scalability and increasing per-device cost [30].

- **Poor Oxide Interface Quality**: The interface between SiC and its thermally grown oxide (SiO_2) suffers from a high density of interface traps (D_{it}). These traps scatter electrons, leading to significantly lower channel mobility in SiC MOSFETs (< 50 cm^2/V·s) compared to silicon devices (> 400 cm^2/V·s). This is a primary obstacle for high-performance SiC CMOS circuits [22, 30].

- **Threshold Voltage Instability**: The high trap density at the SiC/SiO_2 interface can also lead to threshold voltage (V_{th}) instability due to charge trapping, which compromises long-term reliability [12, 30].

- **Difficult Processing**: SiC is an extremely hard and chemically inert material, which makes processes like etching challenging. High temperatures (>1600°C) are required for ion implantation activation, adding complexity to the fabrication process [30].

From a device-level perspective, the fundamental properties of 4H-SiC translate into a unique performance profile that makes it the ideal choice for ICs in high-temperature environments. The wide bandgap ensures that even at several hundred degrees Celsius, the transistors maintain an extremely low off-state leakage current, a critical feature that silicon cannot provide. While the high critical electric field is most famous for enabling smaller, more efficient high-voltage power devices, it also contributes to the overall robustness of the transistors. Crucially, the high thermal conductivity allows the integrated circuit to effectively dissipate its own heat, ensuring stable operation and preventing thermal runaway. Although SiC's performance at room temperature is hampered by low channel mobility and other processing challenges, its unparalleled reliability and functionality at high temperatures make it the superior and often only viable technology for demanding applications in aerospace, automotive, and energy exploration.

2.1.2 The Fraunhofer IISB 4H-SiC CMOS Technology

The processor presented in this thesis is designed and implemented using the advanced 4H-SiC CMOS platform developed by the Fraunhofer Institute for Integrated Systems and Device Technology (IISB). This process is available through the Europractice Multi-Project Wafer (MPW) service and is specifically tailored for IC research in harsh environments, particularly at high temperatures [3]. The technology features a double-well structure on an n-type substrate, a nominal operating voltage of 20 V, and a minimum gate length of 2 μm [14]. Key fabrication steps include ion implantation, high-temperature activation annealing exceeding 1700 °C, the use of an n-type polysilicon gate, and distinct optimized ohmic contact processes for n-type and p-type regions [14].

As illustrated by the cross-section of the SiC MOSFET in Fig. 2.2, the process is built on an n-type epitaxial layer grown on an n-type substrate and involves multiple implantation steps to form the n-well, p-well, and shallow n^+/p^+ regions. A thermally grown oxide layer of approximately 55 nm is used, followed by nitric oxide (NO) annealing to reduce the SiC/SiO_2 interface trap density. The polysilicon gate exhibits a sheet resistance of approximately 15–17 $\Omega/\square$. Ni–Al and Ti–Al contacts are employed to achieve low-resistance ohmic contacts for n-type and p-type regions, respectively [15].

Temperature-Dependent Device Characteristics

MOSFETs fabricated in this technology have been experimentally characterized across a wide temperature range from room temperature to 500 °C. As shown in

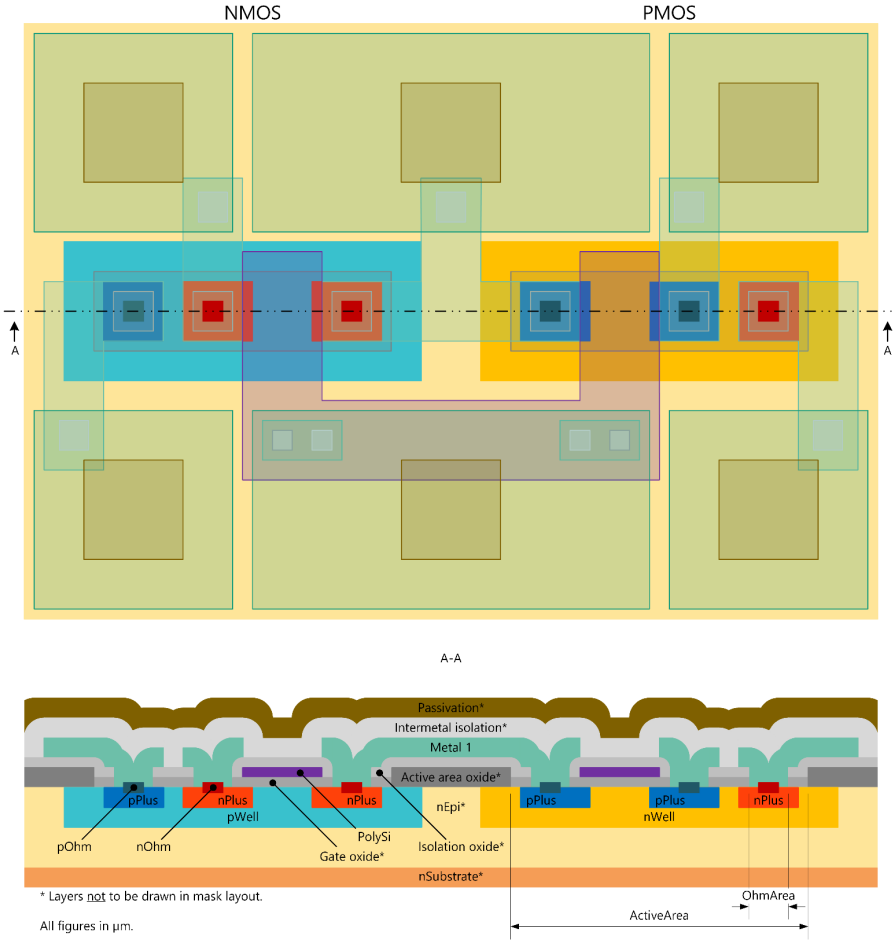


Figure 2.2: Top view and cross section of the SiC MOSFET with metal layer in the Fraunhofer IISB 4H-SiC CMOS process [14].

Fig. 2.3, both n-channel and p-channel devices exhibit a strong temperature dependency in their drain current (I_{DS}) versus drain–source voltage (V_{DS}) behavior. The drain current increases significantly with temperature and gradually saturates at higher temperatures. This phenomenon can be attributed to several physical effects: at low temperatures, limited dopant ionization and a high density of interface traps restrict carrier mobility; as temperature rises, carriers gain sufficient energy to overcome interface traps and contact resistance barriers, resulting in enhanced mobility and drive current. At temperatures above 400 °C, phonon scattering becomes dominant, leading to mobility saturation.

The threshold voltage also shows a pronounced temperature dependence, as illustrated in Table 2.2. For NMOS transistors, V_{th} decreases from approximately +4.8 V at room temperature to around 0 V at 500 °C. For PMOS devices, it shifts from about –8.2 V to –5.8 V over the same range [15]. The NMOS devices are

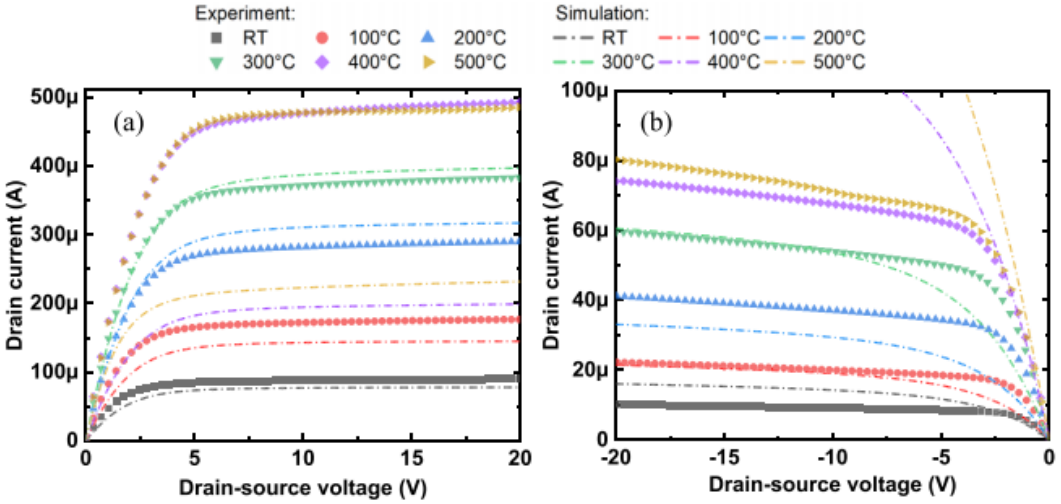


Figure 2.3: Measured and simulated I_{DS}–V_{DS} characteristics of (a) NMOS and (b) PMOS transistors in the Fraunhofer IISB 4H-SiC CMOS process. The drain current increases notably with temperature, demonstrating enhanced carrier mobility and improved drive strength at elevated temperatures [15].

more sensitive to temperature. This heightened sensitivity stems from the higher doping concentration in the p-well combined with the more significant impact of partial dopant ionization characteristic of p-type dopants in SiC. These factors lead to larger temperature-induced variations in the Fermi potential compared to the n-well. Consequently, a 20 V supply voltage is adopted in circuit design to ensure sufficient logic-level swing and robust noise margins across the entire temperature range.

In addition, the wide bandgap of SiC ensures that off-state leakage currents remain negligible even above 300 °C [30]. This low-leakage operation, combined with the aforementioned enhancement of carrier mobility and drive current at elevated temperatures, supports highly reliable operation in harsh environments. Experimental and simulated results show good agreement up to 400 °C, while deviations appear beyond this point, mainly due to model limitations in the current PDK temperature range [15].

Overall, the Fraunhofer IISB 4H-SiC CMOS process enables reliable device operation at supply voltages of 20 V and temperatures up to 500 °C. Its combination of low leakage current, temperature-enhanced carrier mobility, and verified high-temperature reliability makes it a mature and accessible platform for SiC-based circuit research. The technology forms the foundation for the processor design presented in this thesis, serving as the fabrication platform on which all circuit-level design, simulation, and validation are based.

Table 2.2: Extracted values of the threshold voltage (V_{th}), channel-length modulation parameter (λ), and peak field-effect mobility ($\mu_{FE,peak}$) from room temperature to 500°C. Mobility data extracted from the report by Mo et al. [15].

$100 \times 6\ \mu m^2$ PMOS			
Temp. (°C)	V_{th} (V)	λ (V^{-1})	$\mu_{FE,peak,p}$ ($cm^2V^{-1}s^{-1}$)
RT	-8.20	1.21×10^{-2}	∼3.5
100	-7.60	1.26×10^{-2}	∼5.5
200	-6.90	1.23×10^{-2}	∼7.0
300	-6.40	1.17×10^{-2}	∼7.5
400	-6.05	1.12×10^{-2}	∼7.5
500	-5.80	1.43×10^{-2}	∼7.0
$100 \times 6\ \mu m^2$ NMOS			
Temp. (°C)	V_{th} (V)	λ (V^{-1})	$\mu_{FE,peak,n}$ ($cm^2V^{-1}s^{-1}$)
RT	4.80	2.66×10^{-3}	∼18
100	3.55	2.81×10^{-3}	∼20
200	2.30	2.76×10^{-3}	∼21
300	0.70	3.10×10^{-3}	∼21
400	0.25	3.24×10^{-3}	∼20
500	0.05	1.40×10^{-3}	∼19

2.2 Digital Circuit Design in SiC Technology

The realization of reliable digital circuits in silicon carbide (SiC) CMOS technology represents a crucial step toward enabling fully integrated systems that can operate in harsh environments beyond the limits of conventional silicon electronics. While the previous section introduced the material properties of 4H-SiC and the process characteristics of the Fraunhofer IISB CMOS platform, this section focuses on how those underlying physical and process parameters influence circuit-level design considerations, challenges, and performance. Over the past three decades, continuous progress in SiC fabrication and circuit methodologies has expanded the technology's capability from fundamental logic gates to complex architectures, including data converters and processor-level implementations. This section reviews the evolution of SiC digital circuit design, beginning with early pioneering efforts in 6H- and 4H-SiC technologies, followed by recent advancements achieved in the Fraunhofer IISB 4H-SiC CMOS platform, and concluding with a summary of the key design characteristics and insights that guide modern SiC digital circuit development.

2.2.1 Pioneering Work in Early SiC Processes

The development of silicon carbide (SiC) digital integrated circuits has advanced from elementary logic gates to complex, high-temperature systems over the past three decades. Before the establishment of the Fraunhofer IISB CMOS platform, several institutions worldwide demonstrated pioneering implementations of SiC-based digital electronics, establishing the foundation for modern wide-bandgap CMOS design. This section reviews representative milestones achieved using various early SiC processes.

Early 6H- and 4H-SiC Demonstrations (Purdue University, USA)

The first digital circuits realized in SiC were based on 6H-SiC NMOS technology. Cooper *et al.* [31] implemented inverters and ring oscillators operating up to 300 °C, marking the earliest functional demonstration of digital logic in SiC. Subsequently, Ryu *et al.* [23] fabricated the first complementary SiC CMOS logic circuits, including logic gates and D-flip-flops operating at 5 V. Although these works proved the feasibility of digital integration in SiC, the devices suffered from low PMOS mobility, large threshold-voltage asymmetry, and high contact resistance, which constrained circuit complexity and switching speed.

GE Global Research 6H-SiC Digital Modules

At GE Global Research, Ghandi *et al.* [4] developed a 4-bit counter and an 8-bit timing generator using a custom SiC CMOS process. The circuits maintained continuous operation for more than 1,000 hours at 300 °C and survived 20 G random vibration and 215 G mechanical shock. These results represented the first verified demonstration of long-lifetime and mechanically robust SiC CMOS digital circuits, confirming their potential for harsh-environment telemetry and control applications.

Raytheon HiTSiC® 4H-SiC CMOS Process

Raytheon UK developed the 1.2 µm HiTSiC® 4H-SiC CMOS process, enabling systematic design of complex digital logic. As shown in Fig 2.4, Kuhns *et al.* [10] implemented seventeen digital circuits in this technology, including finite-state machines (FSMs), ripple-carry adders (RCAs), shift registers, ring oscillators, and asynchronous NULL Convention Logic (NCL) circuits. All designs operated correctly up to 300 °C, demonstrating stable functionality and timing robustness in high-temperature environments. In a subsequent tape-out, a Digital-to-Analog Converter (DAC) Controller shown in Fig 2.5a was also fabricated to demonstrate digital control for mixed-signal applications. According to Fig 2.5b, the 18-state FSM-based DAC Controller operated reliably up to 300 °C with a maximum clock frequency of about 4.1 MHz [10]. This design further verified the process's ca-

pability to support moderately complex synchronous digital systems for high-temperature environments.

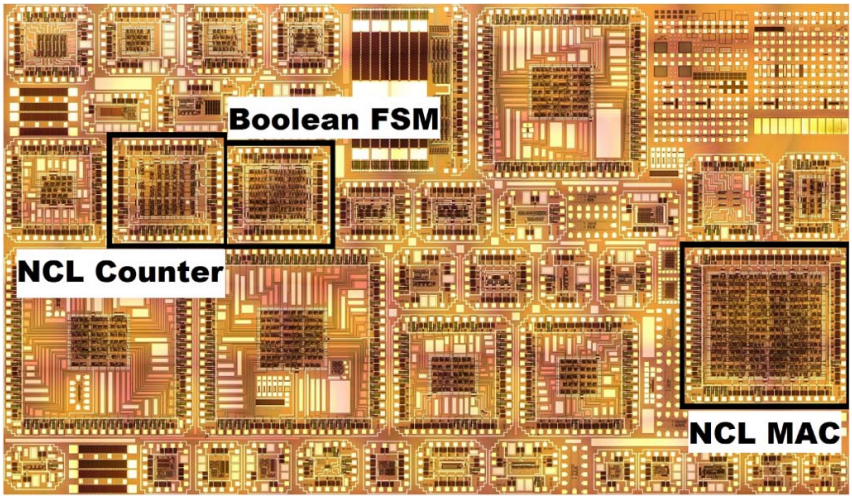
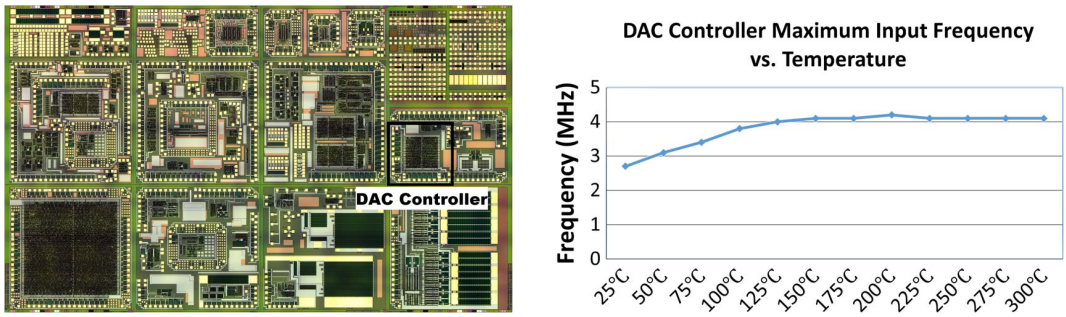


Figure 2.4: Tapeout 1: Digital Circuits in HiTSiC® 4H-SiC CMOS [10]



(a) (b)

Figure 2.5: (a) DAC controller layouts in HiTSiC® 4H-SiC CMOS and (b) corresponding graphical performance data of DAC controller [10].

Extended High-Temperature Operation and SoC Integration

Building on the HiTSiC platform, Holmes *et al.* [6] demonstrated the extended operation of SiC CMOS circuits at 470–500 °C, corresponding to the Venus-surface environment. They tested ring oscillators, clock generators, and D-flip-flop registers for more than 100 hours of continuous operation, confirming the long-term functionality of the technology. Based on these results, the authors synthesized a 16-bit OpenMSP430 microcontroller core and a 1 kB SRAM using a custom high-temperature digital cell library, providing the first conceptual step toward a programmable SiC processor architecture. Subsequent reports indicated that

the HiTSiC platform later became unavailable [15], leading to increased focus on currently accessible technologies.

Table 2.3: SiC Digital Circuits in Early SiC Processes

Year	Process / Organization	Digital Circuit Type	Max Temp (°C)
1994	6H-SiC NMOS (Purdue University, USA)	Inverters, Ring Oscillators [31]	300
1997	6H-SiC CMOS (Purdue University, USA)	Logic Gates, D-Flip-Flops [23]	300
2014	Custom SiC CMOS (GE Global Research)	4-bit Counter, 8-bit Timing Generator [4]	300
2016	HiTSiC® 4H-SiC CMOS (Raytheon UK)	FSMs, RCAs, SRs, NCL Counters, MAC Unit, DAC [10]	300
2016	HiTSiC® 4H-SiC CMOS (Ozark IC / Univ. of Arkansas)	Ring Oscillator, Clock Generator, DFF Register, 16-bit MCU [6]	470–500

Summary

Table 2.3 summarizes representative early SiC digital circuits fabricated by different organizations and their maximum validated operating temperatures. These pioneering efforts collectively established the feasibility of SiC CMOS digital design and proved that reliable high-temperature operation is achievable far beyond the limits of silicon technology. These foundational efforts ultimately led to currently accessible technologies like the Fraunhofer IISB platform discussed in Section 2.2.2, which became increasingly important as some earlier pioneering processes faced availability challenges.

2.2.2 Digital Circuits in IISB 4H-SiC CMOS Process

The 4H-SiC CMOS platform developed by the Fraunhofer IISB serves as the process foundation for our processor design. Its detailed fabrication flow, including the double-well structure, 20 V nominal supply, and minimum gate length of 2 µm, has been described previously in Section 2.1.2. This platform is distributed through the Europractice Multi-Project Wafer (MPW) service and is specifically tailored for the research and development of integrated circuits operating in harsh environments [22].

Early fabrication runs in the IISB 4H-SiC CMOS technology focused on validating the fundamental digital building blocks such as inverters, NAND/NOR gates, multiplexers, and D-type flip-flops [22]. These circuits confirmed that complementary logic can be stably implemented under a 20 V supply, maintaining consistent switching thresholds and voltage transfer characteristics across temperature up to 200 °C. The results established the technological readiness of the SiC CMOS platform for reliable high-temperature logic design.

To explore the scalability of sequential circuits and on-chip memory, a six-transistor (6T) static random-access memory (SRAM) cell shown in Figure 2.6 was implemented and experimentally characterized [21]. The cell maintained correct bistable operation and data retention at 200 °C, verifying that matched pull-up and

pull-down transistor pairs can be achieved despite the inherent mobility asymmetry between NMOS and PMOS devices. This achievement marks a major step toward embedding digital storage in SiC-based systems.

Beyond logic and memory, the Fraunhofer IISB process has also been employed for data-conversion circuits. A two-bit flash analog-to-digital converter (ADC) was first realized and tested successfully [22], followed by higher-resolution designs—a four-bit flash ADC and a six-bit successive-approximation (SAR) ADC—implemented on the same platform [17]. The measured and simulated results demonstrated good linearity and functionality up to 200–300 °C, highlighting the process's capability for mixed-signal integration and control logic at high temperature. Comparable investigations on SiC-based converters were also reported in [20, 28], confirming the advantage of wide-bandgap CMOS for data conversion beyond the thermal limits of silicon.

More recently, the same technology has enabled large-scale integration of functional subsystems. Romijn *et al.* realized a 64-pixel ultraviolet (UV) image sensor with on-chip readout circuitry fabricated entirely in SiC CMOS [21]. The chip integrates pixel-level photodiodes, analog amplifiers, digital scanning logic, and timing generators on a single die, operating stably up to 200 °C. Such system-level integration demonstrates the transition from discrete logic elements toward full system-on-chip (SoC) architectures within the IISB 4H-SiC CMOS technology.

Through successive design iterations, the Fraunhofer IISB platform has evolved from basic logic validation to complex digital and mixed-signal subsystems, forming a scalable and reliable foundation for future high-temperature processors and sensor systems. Representative fabricated circuits are summarized in Table 2.4.

Table 2.4: Digital Circuits in the Fraunhofer IISB 4H-SiC CMOS Process

Year	Circuit Type	Key Specifications	Max Temp (°C)
2022	Inverter, NAND/NOR, MUX, D-FF [22]	20 V supply, 85% yield	200
2022	6T SRAM Cell [22]	SNM 7.1/4.4/7.5 V	200
2022	2-bit Flash ADC [22]	68 devices, linear conversion	200
2022	64-Pixel UV Image Sensor SoC [21]	Digital readout array integration	200
2023	4-bit Flash ADC, 6-bit SAR ADC [15]	266 transistors, 8.8×1.9 mm^2; 821 transistors, 4.9×4.9 mm^2	300
2024	Sensor Interface and ADC Readout Circuits [15]	Fully integrated readout chain	200

2.2.3 Design Characteristics and Considerations of SiC Digital Circuits

Silicon carbide (SiC) CMOS technology exhibits a set of distinctive design characteristics that fundamentally differ from conventional silicon-based logic circuits, offering both remarkable advantages and unique design challenges. Its most prominent strength lies in maintaining reliable functionality under extreme environmental conditions.

Experimental demonstrations conducted using the Fraunhofer IISB 4H-SiC CMOS process have verified stable operation of digital and mixed-signal circuits

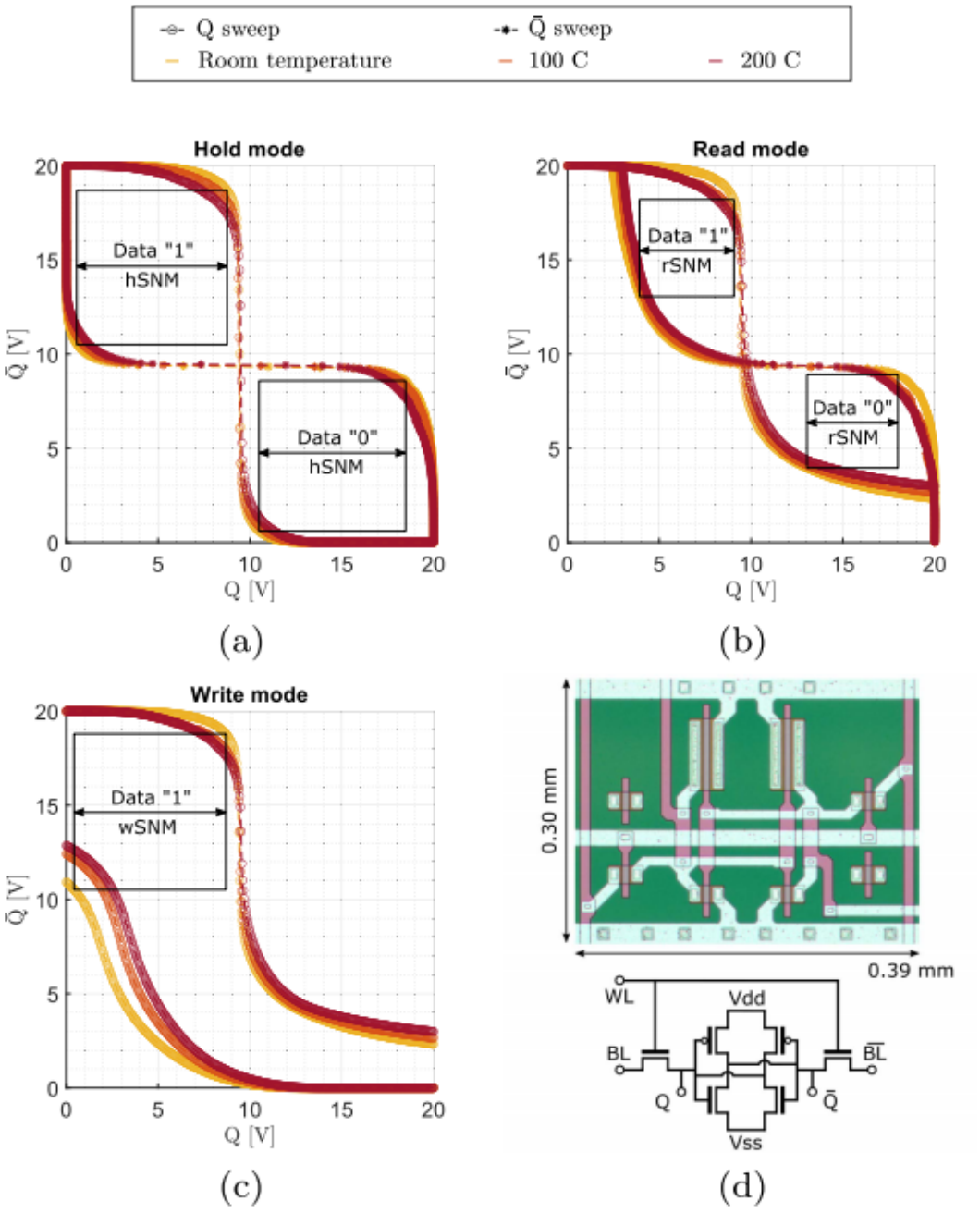


Figure 2.6: Measured Q/$\overline{Q}$ SRAM cell butterfly curves for different temperature levels of (a) hold, (b) read, and (c) write mode including (d) circuit and micro-photograph on die (0, 2). A sweep of Q/$\overline{Q}$ and the maximum size squares are drawn for SNM derivation and a single sweep is depicted for the write mode in favor of readability [21].

including counters, FSMs, ADCs, and SRAMs at temperatures exceeding 300 °C, a performance level unattainable with silicon CMOS [15, 22]. This outstanding high-temperature reliability simplifies system-level design by reducing or even eliminating the need for bulky cooling systems and thermal shielding, enabling more compact and efficient electronic architectures suitable for aerospace, automotive, and geothermal applications. Furthermore, the thermal and electrical stability of SiC devices allows digital and mixed-signal circuits to be co-located with high-power components or sensors, minimizing parasitic effects and improving signal integrity and response speed [22].

However, these advantages are accompanied by several circuit-level limitations that are also evident in the same technology process. A pronounced channel mobility asymmetry between NMOS and PMOS transistors results in highly unbalanced drive strengths, requiring large device width ratios to achieve proper logic-level symmetry [21]. This inevitably increases circuit area, input capacitance, and dynamic power consumption. Additionally, the relatively high and asymmetric threshold voltages—typically above +2 V for NMOS and below –6 V for PMOS—necessitate operation at higher supply voltages in the range of 15–20 V, which further exacerbates static and dynamic power dissipation [15]. Timing closure in synchronous systems is also complicated by the strong temperature dependence of both threshold voltage and propagation delay. As temperature increases, carrier mobility degradation is counterbalanced by a reduction in threshold voltage, often resulting in faster switching at elevated temperatures. This behavior can lead to setup and hold-time violations across wide thermal ranges if design margins are not properly considered in digital circuits.

Futhermore, the Fraunhofer IISB 4H-SiC CMOS process remains in an early stage of technological maturity. Due to the limited number of metal layers, high p-type contact resistance, and partially developed process design kits (PDKs), the design flow still relies heavily on manual optimization and conservative timing margins [22]. Consequently, SiC digital circuits are typically realized with simplified architectures and robust design margins to ensure functional stability under high-temperature operation. Nevertheless, continuous advances in device fabrication and circuit design have steadily increased the level of integration—progressing from basic logic gates to memory arrays, data converters, and even processor-level implementations. These advancements demonstrate that, despite inherent material and device constraints, the Fraunhofer IISB 4H-SiC CMOS technology employed in this work has evolved into a reliable foundation for high-temperature digital systems. It provides the technological foundation for the processor architecture described in the following chapter.

2.3 RISC Processor Design and Integration in SiC Technology

Having established the design principles for SiC digital circuits, this section shifts the focus to system-level integration by exploring the design of a RISC processor in SiC technology. It reviews minimalist RISC architectures that serve as practical validation platforms for emerging semiconductor technologies and examines existing SiC processor demonstrations that provide design insights for the 4-bit processor implemented in this work.

2.3.1 Minimalist RISC Processors as Validation Vehicles

The concept of the minimalist processor originated in computer architecture education, where the main objective was to reveal the fundamental working principles of a processor through an extremely simplified hardware structure. Early didactic CPUs, typically 4-bit or 8-bit, were not designed for performance but to demonstrate instruction fetch, decode, and execution processes using the smallest possible logic scale. This design philosophy emphasized de-complexification by reducing the instruction set, minimizing register count, and limiting addressing modes, resulting in a functionally complete yet structurally transparent processor architecture [13].

As research needs expanded, such architectures evolved from teaching models into valuable experimental platforms for emerging technologies. Compared to commercial microprocessors, minimalist RISC architectures offer clear structural transparency and controllability, enabling focused studies of timing stability, power characteristics, and high-temperature reliability. These architectures allow researchers to directly evaluate the impact of device and process characteristics on system-level performance without interference from complex pipeline stages, branch prediction, or multi-port register files [29].

In contrast to complex instruction set (CISC) architectures, the essence of RISC lies not in having fewer instructions, but in employing a fixed instruction length and simplified addressing modes that enhance decoding efficiency and timing regularity [32]. RISC processors typically adopt hardwired or finite-state-machine (FSM) control units, dividing execution into fixed stages of *fetch*, *decode*, and *execute* to ensure deterministic timing and minimize logic depth [16]. Such control organization improves both reliability and timing predictability, which is an essential advantage for implementation in immature or constraint-heavy fabrication technologies.

Typical minimalist RISC processors exhibit the following key features as shown in the block digram of a typical RISC processor (Figure 2.7) [13, 16, 29]:

- **Harvard architecture:** Program and data memories are physically separated, avoiding bus contention and allowing parallel instruction fetch and data operations.

- **Accumulator-based design:** Arithmetic and logical operations share a single accumulator as both operand and result register, eliminating the need for a complex multi-port register file.
- **Finite-state-machine control:** The control unit generates sequential timing signals based on instruction decoding. With a compact instruction set, the FSM requires only a few states, leading to simple, transparent logic that is easily verifiable.
- **Fixed-cycle execution:** Each instruction completes in a predefined number of clock cycles (typically three or four), facilitating deterministic timing analysis and system verification.

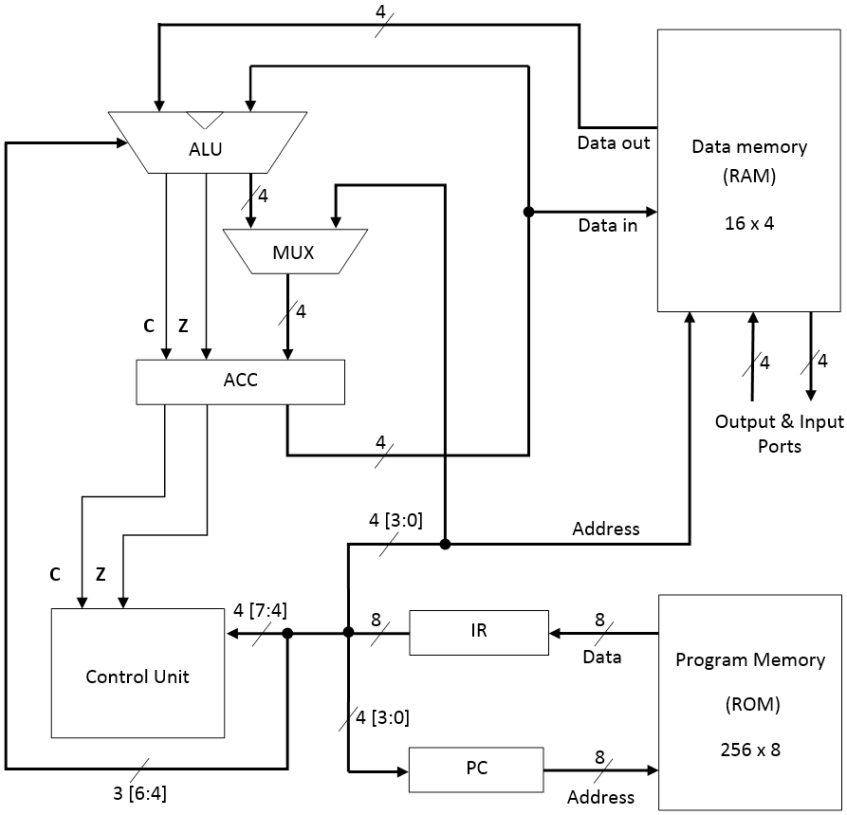


Figure 2.7: The block diagram of a minimalist RISC processor showing the core modules and data flow between the ALU, ACC, control unit, ROM, and RAM [13].

The relationship between processor performance and its key parameters can be expressed as:

$$T_{\text{exec}} = N_{\text{instr}} \times \text{CPI} \times T_{\text{critical}} \text{ [32]}$$

where T_{critical} is the critical-path delay determining the maximum operating frequency. Minimalist RISC architectures reduce T_{critical} by minimizing combina-

tional depth, shortening signal paths, and enforcing a fixed control sequence. This design approach sacrifices instruction complexity in exchange for improved timing reliability and process robustness, which are particularly beneficial for wide-bandgap semiconductor technologies such as SiC CMOS.

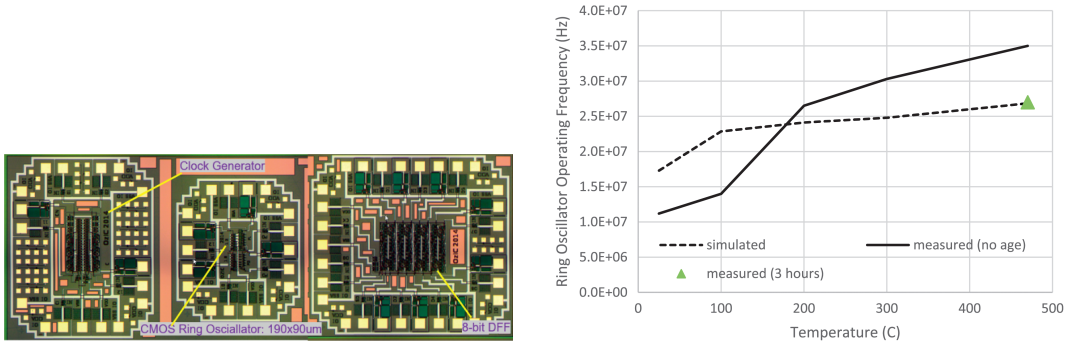
Such structural simplicity not only minimizes area and logic overhead but also enhances design verifiability and portability, making minimalist processors ideal as first-silicon validation vehicles in emerging technologies [29]. In wide-bandgap processes such as 4H-SiC CMOS, process rules, parasitic models, and EDA tools are still under development. With their shallow logic depth and compact interconnect structure, minimalist RISC processors enable rapid system-level demonstration and characterization [13]. Consequently, they have been widely employed as system-level validation chips, serving as foundational testbeds for assessing logic reliability, timing consistency, and high-temperature operation in new semiconductor technologies.

2.3.2 Existing SiC Processor Demonstrations and Integration Insights

Early studies on silicon carbide (SiC) digital circuits mainly focused on verifying the high-temperature functionality of basic logic modules, as discussed previously in Section 2.2. As the 4H-SiC CMOS process matured, researchers began to construct the first SiC digital system prototypes capable of performing both control and arithmetic operations, marking an initial step toward system-level integration. Kuhns *et al.* implemented a set of complex digital circuits in the Raytheon HiTSiC® CMOS process, including finite-state machines, asynchronous counters, multiply–accumulate units, and DAC controllers [10]. These circuits operated reliably at 300 °C, demonstrating that SiC CMOS technology was capable of supporting higher levels of digital integration and control logic functionality.

Holmes *et al.* further advanced SiC CMOS technology toward processor-level verification, aiming to achieve long-term stable operation of SiC microcontroller systems under a 470 °C environment [6]. Based on the HiTSiC® CMOS process, the study established high-temperature device models and developed a digital logic design flow applicable to 470 °C operation. Long-duration testing of CMOS ring oscillators, clock generators, and register files was performed, and their corresponding wafer layouts are shown in Figure 2.8a. As a representative example, the simulation analysis of ring oscillators at elevated temperatures (Figure 2.8b) confirmed stable circuit operation, confirming stable behavior and verifying the reliability of the SiC CMOS devices up to 470 °C [6]. Using the extracted experimental data, the team derived temperature-dependent parameters such as threshold voltage shift, mobility degradation, and aging effects, thereby constructing SPICE and timing libraries at 470 °C. High-temperature simulation analyses based on these models revealed that the conventional 5:1 PFET-to-NFET width ratio caused excessive delay mismatch and overdriving above 300 °C, whereas a 3:1 ratio provided superior drive balance and timing consistency at extreme temperatures [6]. Consequently, a new digital standard-cell library was developed, including ba-

sic elements such as logic gates, latches, and flip-flops, which provided a unified foundation for subsequent processor synthesis and system-level design.



(a) (b)

Figure 2.8: (a)Die micrographs of CMOS clock generator, CMOS and NMOS ring oscillators, and D-type flip-flop and (b) Operation of ring oscillator using simulation models from room temperature to 470 °C [6].

With the optimized device models and logic library, the team synthesized and simulated a 16-bit SiC microcontroller based on the open-source OpenMSP430 architecture [5]. The system adopted a Von Neumann structure, integrating 1 kB SRAM, a clock generator, and standard I/O interfaces within an active area of approximately 250 mm^2 (Figure 2.9a). Corresponding simulation results shown in Figure 2.9b indicated stable operation at around 800 kHz under a 12 V supply and 470 °C [6]. Additional lifetime tests on the ring oscillator, non-overlapping clock generator, and 8-bit register file confirmed stable functionality during 100 hours of continuous high-temperature operation. These findings validated the effectiveness of the 3:1 sizing strategy and high-voltage biasing for maintaining logic integrity and timing stability, providing essential guidelines for future system-level processor design [6].

The work by Holmes and colleagues further demonstrated the scalability of SiC CMOS microcontroller integration. Several peripheral modules, including an 8-bit ADC, comparator, communication interface, and gate-driver circuits, were proposed and partially verified to interface with the SiC microcontroller [6, 10]. These results illustrate that SiC CMOS technology can support not only digital logic and memory modules but also mixed-signal peripherals, paving the way toward high-temperature SiC-based system-on-chip (SoC) platforms. Overall, the study established a significant milestone in the evolution from discrete SiC digital modules to a complete programmable microcontroller system, confirming the feasibility of wide-bandgap CMOS technology for processor-level integration in extreme environments. These developments provide valuable insights and technical foundations for the minimalist 4-bit RISC processor design presented in this work.

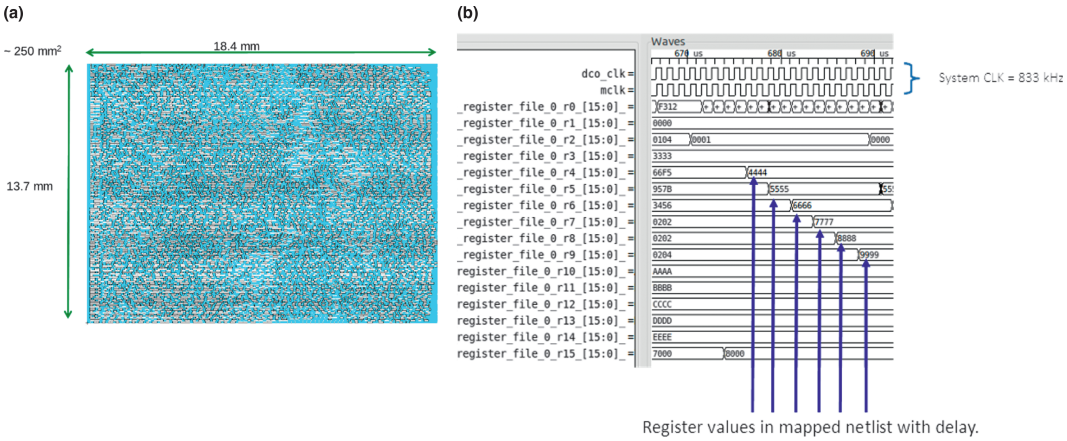


Figure 2.9: (a) Layout of SiC-CMOS microcontroller core implemented with placed and routed SiC-CMOS digital standard cells and (b) digital simulation (ADD operation) using 470°C extracted timing models [6].

2.3.3 Summary and Outlook

In summary, minimalist RISC architectures, characterized by structural simplicity, deterministic timing, and design transparency, have proven highly suitable for validating emerging semiconductor technologies. Over the past decade, SiC CMOS research has progressed from demonstrating basic logic circuits to exploring processor-level system integration. The 16-bit microcontroller proposed by Holmes *et al.* [6] represented a major step in this evolution, providing a complete design flow and high-temperature simulation validation up to 470 °C. However, the design remained at the simulation and low-frequency verification stage without fabrication, indicating that SiC processor research is still primarily focused on feasibility-level evaluation.

Building upon these developments, the present work implements a minimalist 4-bit RISC processor in the Fraunhofer IISB 4H-SiC CMOS process. This design provides a compact and fabricable platform for system-level validation, enabling further investigation of integration capability and performance reliability of SiC CMOS digital circuits under extreme temperature conditions.

Chapter 3

Processor Architecture Design

This chapter aims to design and describe in detail a minimalist 4-bit processor, with the core objective of serving as a test vehicle to validate the feasibility and reliability of SiC CMOS technology for digital systems. The chapter begins by outlining the 'ultra-minimalist' design philosophy adopted to mitigate the constraints of the emerging SiC process. It then details the custom Instruction Set Architecture (ISA) and the overall top-level architecture of the processor, including its core modules and external interfaces. A significant portion is dedicated to the Control Unit, explaining its Finite State Machine (FSM) based design. The chapter concludes by discussing the functional implementation possibilities and summarizing the key design accomplishments.

3.1 Design Philosophy

This design of the minimalist 4-bit Processor is based on the Fraunhofer IISB 2 μm 4H-SiC CMOS process. This technology is developed for applications in harsh environments, such as high operating temperatures [30]. However, as an emerging technology, its integration level for digital circuits is still in a preliminary stage. Limited by the capabilities of process equipment such as lithography tools, the minimum gate length of this process is 2 μm , which results in a transistor density significantly lower than that of mainstream silicon-based processes[14]. Consequently, within the limited chip area of 5x5 mm², the available logic resources are highly constrained, accommodating an estimated maximum of only around 3000 transistors. To minimize design risks under this strict area constraint and to maximize the probability of a successful first-pass tape-out with this emerging process, this design adopts an "ultra-minimalist" architectural philosophy.

To achieve this goal, several key decisions were made: First, the program memory (ROM) was moved entirely off-chip to maximize the available die area for the processor core. Second, area-intensive on-chip general-purpose RAM was eliminated and replaced with basic input and output registers. Finally, and most critically, an "instruction set-hardware co-design" strategy was employed. By design-

ing a highly optimized and concise instruction set, we shifted hardware complexity to the software (instruction set encoding) layer, thereby minimizing the gate count and area of the control unit.

This design enables the processor to efficiently execute predefined, linear arithmetic and logic operations, bit manipulations, and simple timing control tasks. It is precisely this inherent simplicity and predictable linear execution flow that make it a very robust design, ideal for executing high-reliability, fixed sequential control tasks in extreme physical environments such as aerospace.

3.2 Instruction Set Architecture (ISA)

The Instruction Set Architecture (ISA) is the interface between the processor's hardware and software, defining all the operations the processor can understand and execute. While drawing inspiration from the minimalist processor for teaching purposes presented in [13], this processor's ISA is meticulously adapted for ultimate simplicity. As shown in table 3.1, it consists of a minimal set of 11 instructions covering basic arithmetic, logic, and data transfer functions. The Arithmetic Logic Unit (ALU) is the core hardware module that executes the computational parts of these instructions. The first eight instructions can be categorized as ALU execution group.

Table 3.1: Instruction Set and Coding for 4-bit processor

Instruction	Opcode	Function	Description
ADD	0000	ACC ← Input + ACC	Addition
SUB	0001	ACC ← Input - ACC	Subtraction
AND	0010	ACC ← Input AND ACC	And
OR	0011	ACC ← Input OR ACC	Or
XOR	0100	ACC ← Input XOR ACC	Exclusive or
NOT	0101	ACC ← NOT ACC	1´s Complement
IN	0110	ACC ← Input	ACC load in
NOP	0111	ACC ← ACC	No operation
OUT	1000	Output ← ACC	Load out results to output port
CLR	1001	ACC ← 0	Clear data inside the processor
HALT	1010	Pause all modules	Terminate the processor

ALU Operations

The Arithmetic Logic Unit (ALU) is responsible for executing the processor's core computations, including unsigned arithmetic operations such as addition (ADD) and subtraction (SUB), and logical operations such as NOT, AND, OR, and exclusive-OR (XOR). The unit has two 4-bit data inputs (A from the input latch and B from the accumulator) and one data output. A 3-bit selection signal

determines which operation is performed. The key optimization of this design lies in its decoding and execution logic. The lower three bits of the instruction register, IR[2:0], are directly wired to the ALU's 3-bit operation select port, ALU_Sel [13]. This means the ALU continuously calculates a result based on the lower three bits of the current instruction. The table 3.2 shows all possible operations of the ALU.

Table 3.2: Operations of the 4-bit ALU

ALU_Sel	Output	Description
000	A + B	Addition
001	A - B	Subtraction
010	A AND B	Bitwise AND
011	A OR B	Bitwise OR
100	A XOR B	Bitwise XOR
101	NOT B	Bitwise NOT on B
110	A	Pass-through A
111	B	Pass-through B

3.3 Top-Level Architecture

This processor is an accumulator-based Reduced Instruction Set Computer (RISC) with a Harvard architecture which separate program and data interfaces. To achieve a minimal implementation, the data bus width was constrained to 4 bits and the storage capacity was reduced to its lowest practical level. Its entire design is aimed at minimizing internal logic complexity. The schematic design is shown in Figure 3.1. In the diagram, the arrows with blue text represent the control signals from the Control Unit to various modules inside the processor, while the arrows with red text indicate the signals that interface with off-chip components.

Core Modules

- **Program Counter (PC)**: An 8-bit register capable of addressing an off-chip program memory with 256 locations.
- **Instruction Register (IR)**: A 4-bit register that temporarily stores the current instruction opcode fetched from ROM.
- **Accumulator (ACC)**: A 4-bit core working register used to store intermediate and final results of ALU operations.
- **Input Latch**: A 4-bit latch used to instantly capture transient data from external pins during the instruction execution phase, adopted in place of a conventional register. The design rationale and timing behavior are further discussed in Chapter 4.

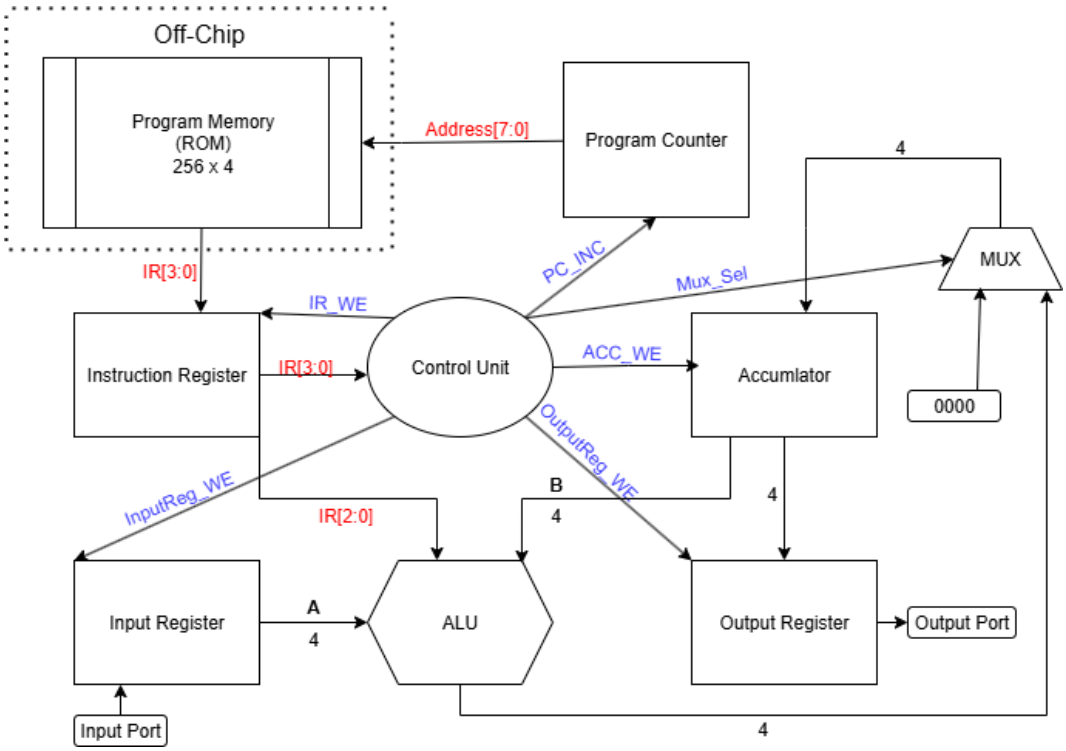


Figure 3.1: Processor architecture block diagram

- **Output Register**: A 4-bit register used to provide stable output data to external pins.
- **Arithmetic Logic Unit (ALU)**: A 4-bit operational unit capable of performing 8 different arithmetic and logical operations.
- **Control Unit (CU)**: A Finite State Machine (FSM)-based controller responsible for generating all timing and control signals. All instructions are completed in fixed 4 clock cycles.

Chip External Interface (I/O)

According to 3.2, the processor core is packaged as a 24-pin chip. Its external interfaces, as shown in the figure below, are mainly divided into three categories:

System Signals & Power

- **Clock Source (CLK):** Provides the operating clock signal for the processor.
- **Reset Signal (RST):** Resets the processor to its initial state.
- **Power Supply (VCC, GND):** Provides the operating voltage for the chip.

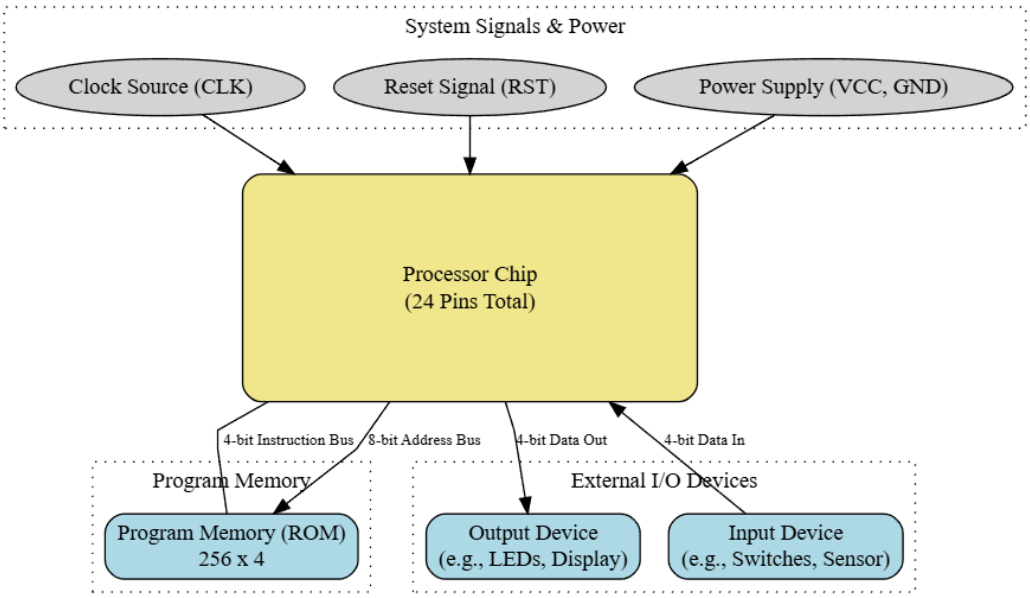


Figure 3.2: Input and Output Ports of Processor

Program Memory Interface

- **8-bit Address Bus:** Outputs an 8-bit address to the external program memory (ROM) to select an instruction from 256 locations.
- **4-bit Instruction Bus:** Reads a 4-bit instruction opcode from the external ROM.

External Device Interface

- **4-bit Data In:** Reads 4-bit data from external input devices (e.g., switches, sensors).
- **4-bit Data Out:** Writes 4-bit data to external output devices (e.g., LEDs, displays).

Data Processing and I/O Path

The processor's data processing path is the core datapath where all calculations and data manipulations occur. The flow begins at the Input Port, where external data is captured by the Input Latch when the Control Unit issues the latch enable signal. This latched data is then continuously supplied as one of the inputs to the ALU, while the other input is provided by the ACC, which holds the result from the previous operation. As a purely combinational circuit, the ALU constantly generates a new result based on these inputs and the current instruction's ALU select signal (IR[2:0]). This result is routed to a Multiplexer (MUX) that determines

what will be written into the ACC. The MUX selects either the ALU's output or a constant '0' (for the clear instruction). During instructions that modify the ACC, the Control Unit asserts the write enable signal, and the value selected by the MUX is loaded into the Accumulator on the next clock edge, completing the processing loop. Finally, for output operations, the value stored in the ACC is captured by the Output Register when the OUT instruction is executed. This ensures the final result is stable and available on the Output Port.

3.4 Control Unit (FSM)

The Control Unit is the heart of the processor. It is a controller based on a Finite State Machine (FSM) that ensures each instruction is completed in a fixed sequence of 4 clock cycles. The FSM's state transitions, as shown in the 3.3, primarily include Fetch, Decode, and multiple Execute states.

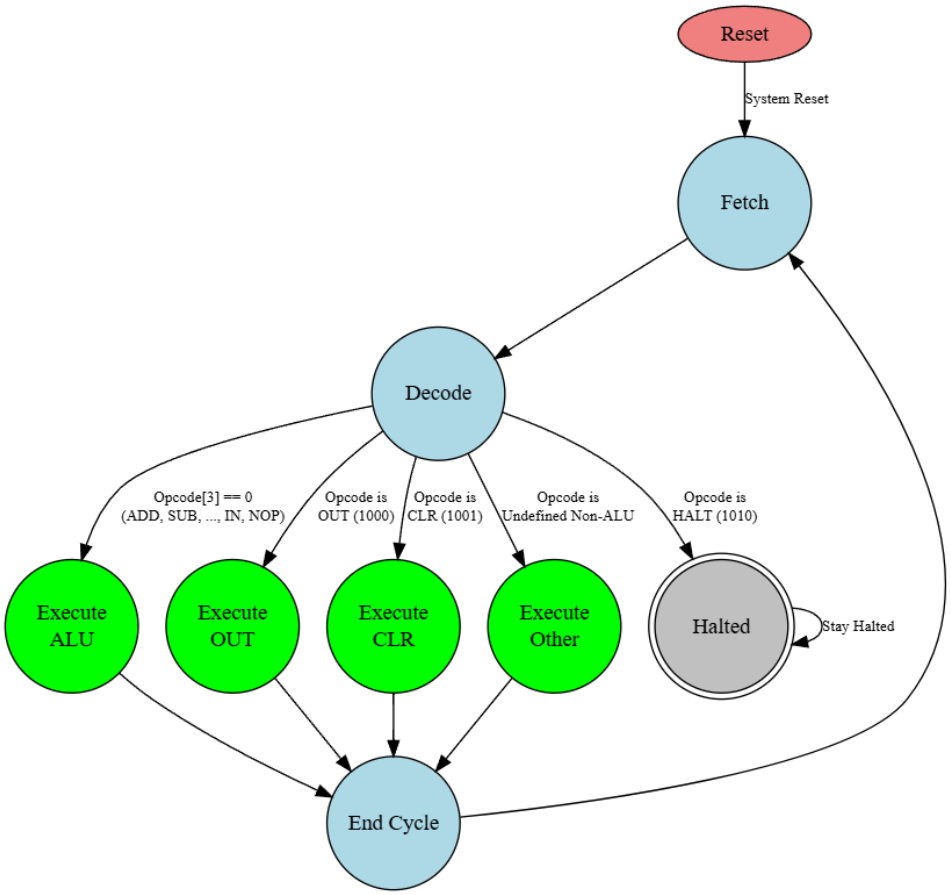


Figure 3.3: Finite state machine of Control unit

1. **Fetch:** The FSM generates control signals to load the instruction pointed to by the Program Counter (PC) from external ROM into the Instruction Register (IR) and then increments the PC.
2. **Decode:** The FSM enters the Decode state and determines the next state based on the opcode in the IR. This is the key decision point of the FSM.
3. **Execute:** Based on the decoding result, the FSM transitions to the corresponding execution state, such as *Execute_ALU*, *Execute_OUT*, or *Execute_CLR*.
4. **End Cycle:** After execution, the FSM enters a unified end state and then returns to the Fetch state to begin the next instruction cycle. For the HALT instruction, the FSM will enter and remain in the Halted state until the next system reset.

One of the core task of the Control Unit (CU) FSM is to decide the subsequent jumps and operations based on the most significant bit, I[3], during the Decode state:

- When I[3] is 0 (ALU instructions): The FSM will jump to the Execute_ALU state and issue a load signal in that state, allowing the ALU's output result to be written into the Accumulator (ACC).
- When I[3] is 1 (non-ALU instructions): The FSM will prevent the ALU's result from being written to the ACC and, based on IR[2:0], decode and jump to specific states like Execute_OUT, Execute_CLR, or Halted to perform the corresponding operations.

This design minimizes the need for decoding logic, resulting in a more streamlined hardware implementation for the control unit and aligning perfectly with the core goal of this design. The logic design and behavior verification for the control unit will be presented in the next chapter.

State Encoding

The FSM consists of 8 states, requiring 3 state bits (S2,S1,S0) for encoding

Table 3.3: FSM States and Encoding

State	Encoding (S2 S1 S0)	Description
Fetch	000	Fetch instruction from ROM
Decode	001	Decode opcode in IR
Execute_ALU	010	Execute ALU-type instruction (write ACC)
Execute_OUT	011	Perform OUT
Execute_CLR	100	Perform CLR
Execute_Other	101	Undefined non-ALU
End_Cycle	110	End instruction cycle / PC increment
Halted	111	HALT state

Control Signals Output

The Table 3.4 lists every command signal that the Control Unit sends out to the other parts of the processor (like the PC, Accumulator, etc.) with a description of what action occurs when that signal is activated or turned "on".

As shown in Table 3.5, the output table defines the output value for each control signal in every state. All outputs are Moore-type.

Table 3.4: Control Signals Definition

Signal Name	**Type**	**Description**
PC_INC	Output	Active-high. Enables the Program Counter (PC) to increment.
IR_WE	Output	Active-high. Write enable for the Instruction Register (IR).
ACC_WE	Output	Active-high. Write enable for the Accumulator (ACC).
InputReg_LE	Output	Active-high. Latch enable for the Input Register.
OutputReg_WE	Output	Active-high. Write enable for the Output Register.
ACC_Mux_Select	Output	MUX select signal. 0 = ALU result; 1 = constant 0000.

Table 3.5: Control Unit Output Table

State	**State Name**	**PC_INC**	**IR_WE**	**ACC_WE**	**InputReg_LE**	**OutputReg_WE**	**ACC_Mux_Select**
000	Fetch	0	1	0	0	0	0
001	Decode	0	0	0	0	0	0
010	Execute_ALU	0	0	1	1	0	0
011	Execute_OUT	0	0	0	0	1	0
100	Execute_CLR	0	0	1	0	0	1
101	Execute_Other	0	0	0	0	0	0
110	End_Cycle	1	0	0	0	0	0
111	Halted	0	0	0	0	0	0

3.5 Functional Implementation

Although this processor architecture is streamlined, its instruction set and 256-instruction program capacity allow it to perform a variety of meaningful functions. Its core capability lies in performing pre-defined, linear arithmetic and logical processing on 4-bit external inputs and outputting the results.

- **Basic Arithmetic & Data Operations:** The processor can execute a series of mathematical calculations, handling 4-bit unsigned integers from 0 to 15.
  - **Multi-stage Adder/Subtractor:** A program can be written to continuously read multiple numbers and perform cumulative addition or subtraction. Example: calculating $A + B + C - D$.
  - **Multiplication via Repeated Addition:** Multiplication can be performed for small numbers. Example: calculating 3×4 by accumulating 3 four times.
  - **Data Clearing:** The CLR instruction can be used to conveniently clear the accumulator before starting a new calculation, ensuring accuracy.
- **Logical Operations & Bit Manipulation:** The processor can perform bit-wise operations on data, which is highly useful in low-level control and data processing.
  - **Masking Operations:** It can check, isolate, or clear specific bits within an input data word. Example: checking if the second bit (D_1) of a 4-bit input is set by performing an AND operation with the mask `0010`.
  - **Setting/Clearing a Specific Bit:** It can force a specific bit of an input data word to 1 or 0. Example: forcing the most significant bit of any input to 1 by performing an OR operation with the mask `1000`.

3.6 Summary of Architectural Design

This chapter's work culminates in the architectural design of a 4-bit processor core targeting the Fraunhofer IISB 2 μm 4H-SiC CMOS process. To ensure a high success rate for this implementation, the overall architecture follows an "ultra-minimalist" design philosophy. Through key strategies such as hardware-software co-design, off-chip memory, and a simplified datapath, the design achieves an exceptionally low logic gate count and a highly optimized structure, thus minimizing on-chip logic complexity and area. Therefore, the architecture proposed in this chapter serves as a crucial foundation for the logic design, circuit implementation, and simulation detailed in the following chapters, paving the way for eventual physical implementation and process validation.

Chapter 4

Logical Design and Functional Verification

This chapter details the logical implementation and verification of the processor, focusing first on the Control Unit. It covers the FSM hardware design, Register-Transfer Level (RTL) implementation in Verilog, and behavioral simulation. The chapter then discusses the digital synthesis flow using Yosys and the subsequent optimization of the control unit's logic for the SiC process. Following this, the chapter moves to the processor-level verification, including modifications to the input module to resolve timing issues. It concludes with a comprehensive functional verification of the entire processor and the synthesis of its top-level schematic.

4.1 Control Unit Logic Design and Verification

This section provides a detailed description of the Control Unit's low-level logic design and verification. Building upon the high-level FSM architecture introduced in Chapter 3, it focuses on the derivation of the state transition logic, RTL implementation in Verilog, and the translation into the final hardware circuit.

4.1.1 FSM Hardware Architecture and State Transition Design

The control unit is implemented as a classic Moore-type FSM because its outputs depend only on the current state, providing stable, glitch-free control signals for the entire clock cycle. This predictable timing is essential for reliably commanding the synchronous registers in the processor's datapath. From a hardware implementation perspective, it consists of three core logic components as shown in Figure 4.1:

- **State Register:** Consists of a set of flip-flops used to store the machine's current state. In our design, as there are eight distinct states, we use three D-type flip-flops. This register is updated on every rising edge of the clock,

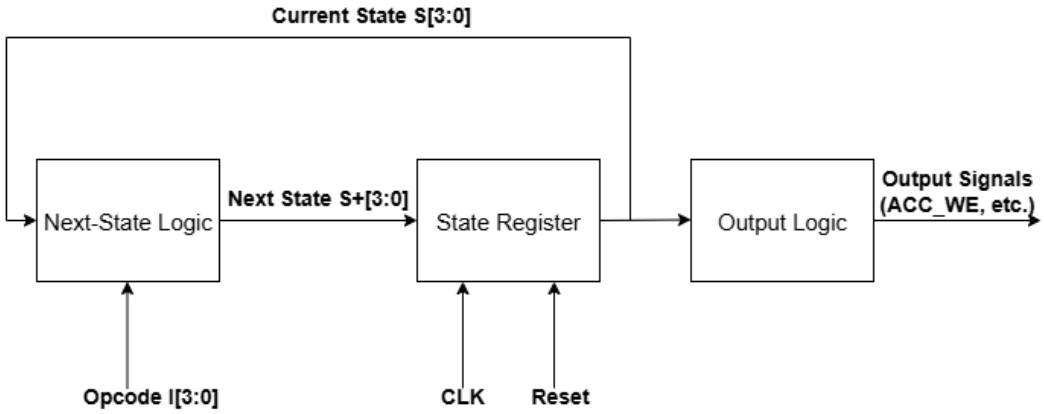


Figure 4.1: Control Unit Block Diagram

transitioning the machine from the current state to the next state. An asynchronous reset is also included to force the FSM into the known initial `FETCH` state.

- **Next-State Logic:** A combinational logic block that determines the FSM's next state based on the current state and external instruction opcode.

- **Output Logic:** A second combinational logic block that generates control signal outputs based solely on the current state. This is a characteristic of a Moore FSM, where outputs are associated with states rather than transitions.

The division into these three components forms the foundation of the FSM logic design, mapping the abstract state transition behavior to a concrete hardware structure. This behavior is formally defined in Table 4.1, which precisely dictates the next state for every possible combination of current state and input conditions.

Table 4.1: State Transition Table

Current State	State Encoding	Input Condition	Next State	State Encoding
Fetch	000	Don't Care	Decode	001
Decode	001	I[3] == 0 (ALU-type instruction)	Execute_ALU	010
		I[3:0] == 1000 (OUT)	Execute_OUT	011
		I[3:0] == 1001 (CLR)	Execute_CLR	100
		I[3:0] == 1010 (HALT)	Halted	111
		Others (Undefined)	Execute_Other	101
Execute_ALU	010	Don't Care	End_Cycle	110
Execute_OUT	011	Don't Care	End_Cycle	110
Execute_CLR	100	Don't Care	End_Cycle	110
Execute_Other	101	Don't Care	End_Cycle	110
End_Cycle	110	Don't Care	Fetch	000
Halted	111	Don't Care	Halted	111

4.1.2 RTL Design and Behavioral Simulation

To aid the hardware implementation of the control unit, a more detailed FSM flowchart was derived from the State Encoding and Control Signals Output definitions in Chapter 3. As shown in the figure below, this diagram not only illustrates the transition relationships between states but also clearly indicates the specific control signals to be generated in each key state or transition (e.g., IR_WE=1 for DECODE State). This diagram serves as the core design philosophy for the module's Verilog code.

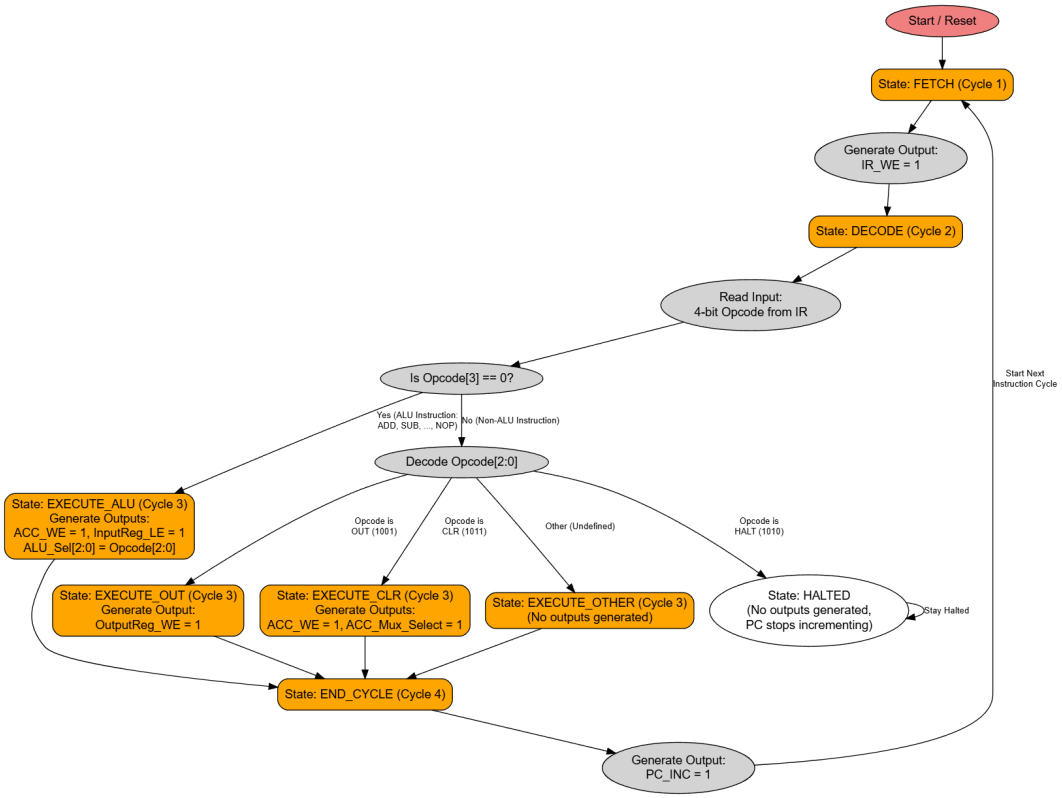


Figure 4.2: FSM Flowchart with Control Signals and States

Guided by this flowchart, our initial approach was a standard Register-Transfer Level (RTL) design, using standard behavioral Verilog HDL to design and model the control unit (see Appendix A.1 for the complete code). Through this high-level abstraction, we were able to quickly build a functional prototype of the FSM and operate behavioral simulations, testing whether its logical functionality met the design requirements completely.

As the simulation waveform in Figure 4.3 and 4.4 shows, the behavioral model was thoroughly verified. The control unit correctly processes a sequence of instructions, including ALU-type instructions (I=0-7) and other commands like `OUT`, `CLR`, and `HALT`. Each instruction is executed in a fixed 4-cycle sequence, with the

module asserting the appropriate control signals in the correct clock cycle. The control signal outputs in the simulation perfectly match the behavior described in the Detailed FSM Flowchart. For instance, during the Fetch state (000), `IR_WE` is asserted high; during the `Execute_ALU` state (010), both `InputReg_LE` and `ACC_WE` are asserted. Furthermore, upon receiving the HALT instruction (I=10), the control unit correctly transitions to and remains in the `Halted` state (111), which aligns with our design goal of stopping the processor's operation.

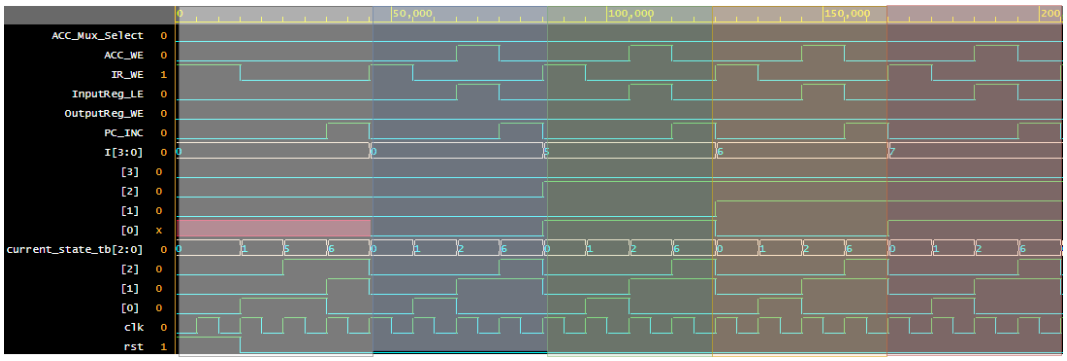


Figure 4.3: Behavioral Simulation Waveform of ALU Execution. Color-coded intervals represent the execution periods of each instruction.

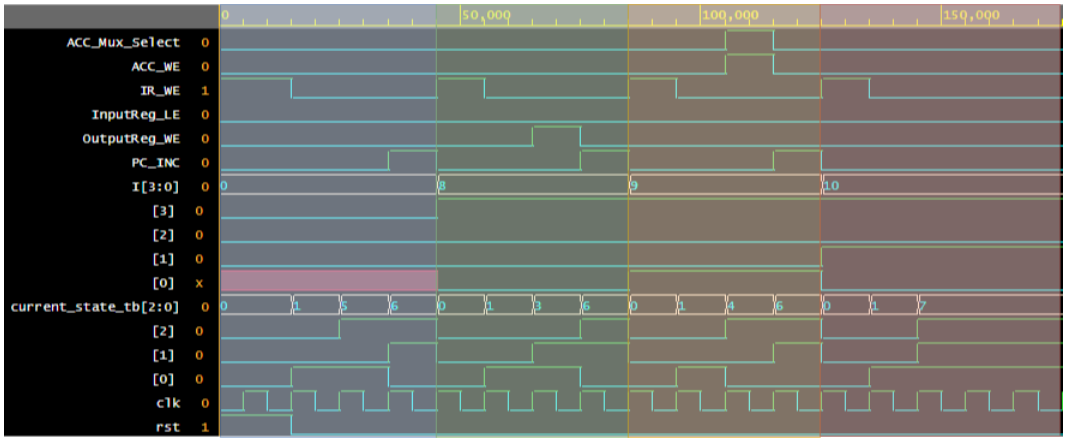


Figure 4.4: Behavioral Simulation Waveform of Non-ALU Execution. Color-coded intervals represent the execution periods of each instruction.

4.1.3 Digital Synthesis Flow

As the project progressed to the physical implementation stage, a critical challenge emerged: the Fraunhofer IISB 2 μm 4H-SiC CMOS technology used in this work is still in an early stage of development. While its Process Design Kit (PDK) supports circuit-level design and simulation, it lacks a mature digital back-end synthesis

flow and corresponding tool support. Consequently, the behavioral Verilog code could not be automatically synthesized into a low-level gate-level circuit, as is standard practice in silicon-based CMOS processes.

To overcome this limitation, Yosys was employed as the front-end synthesis framework. Yosys is an open-source logic synthesis tool capable of transforming RTL-level Verilog descriptions into gate-level representations while generating visualized circuit schematics that illustrate the interconnections and data flow among logic modules. The behavioral-level Verilog was synthesized into the gate-level schematic shown in Figure 4.5. Based on the synthesized netlist and schematic structure obtained from Yosys, the circuits were manually implemented and migrated into Cadence Virtuoso, where circuit-level simulations and performance evaluations were subsequently performed.

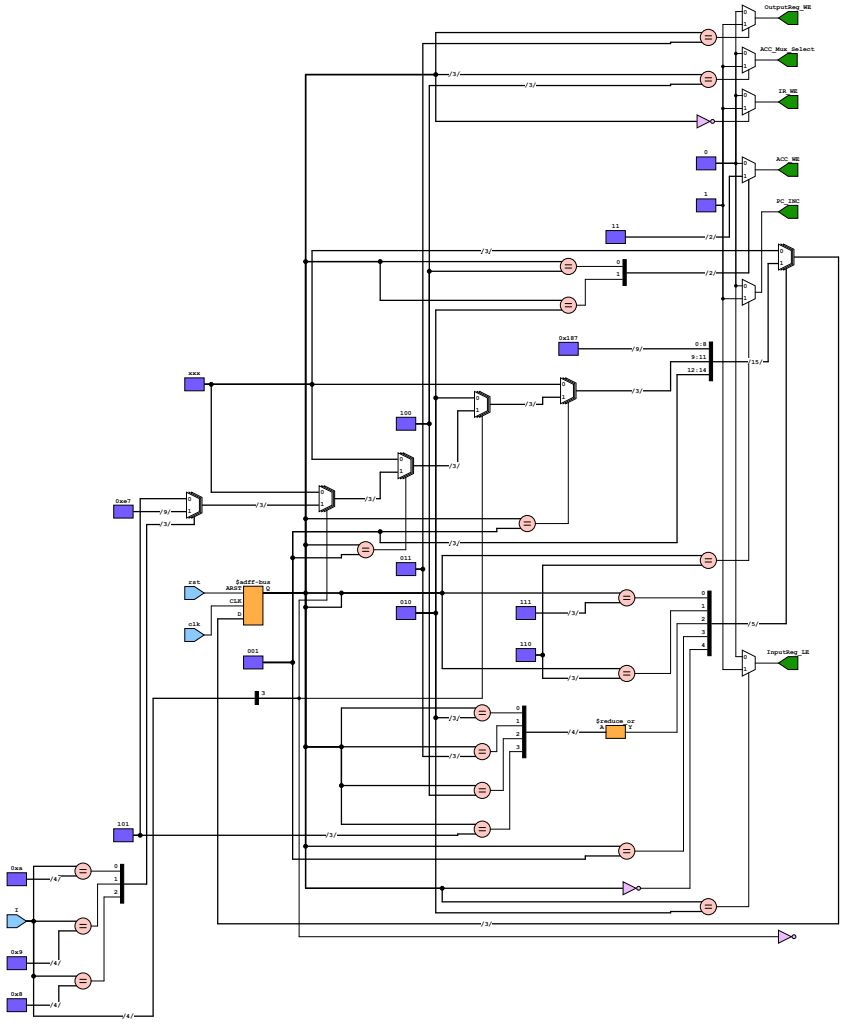


Figure 4.5: Schematic of the Control Unit (Behavioral-level design).

4.1.4 Control Unit Optimization

The hardware implementation scheme of the Control Unit was refined to improve feasibility under the SiC CMOS process. The initial behavioral-level design 4.5, when synthesized using Yosys, resulted in a complex logic structure involving multi-bit conditional comparisons and multiplexing networks. The notations on the data lines (e.g., "/4/") indicate the signal bit width. Such implementations resemble FPGA-style synthesis patterns—functionally correct but not area-efficient for transistor-level realization. When manually reproduced in Cadence Virtuoso, these generated circuits, particularly the multi-bit logic comparators marked with "=" in Figure 4.5 and the corresponding multi-bit multiplexers, result in a substantial increase in transistor count and layout area.

To achieve a more process-compatible implementation, the behavioral approach was replaced with a manually derived Boolean-expression-based design. Based on the state transition and output tables defined in Chapter 3, minimized Boolean equations were obtained for each next-state and output signal. These simplified expressions were then directly written into Verilog (Appendix A.2) using a dataflow modeling style, mapping the control logic to a concise gate-level structure. This Boolean-expression-based implementation was synthesized into the final gate-level schematic using Yosys, as shown in Figure 4.6. Although this approach sacrifices some abstraction and reconfigurability, it significantly improves layout compactness and ensures the design's practicality under the constraints of the current SiC CMOS technology.

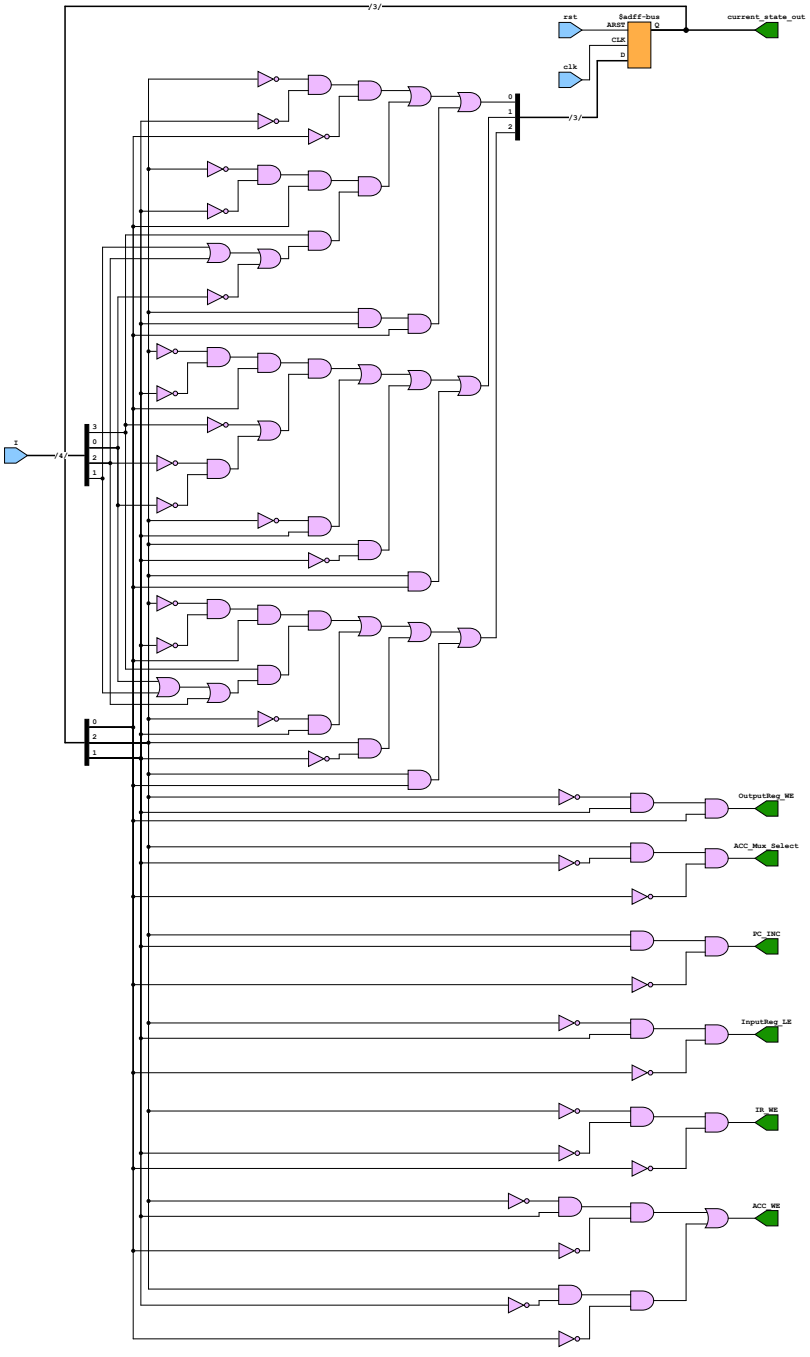


Figure 4.6: Schematic of the Control Unit (Structural-level design).

4.2 Processor Logic Design and Verification

With the Control Unit fully verified, this section advances to the processor-level logic verification. It focuses on integrating all core modules including the Control Unit, ALU, and memory components into the complete processor system. The objective is to validate the correctness of inter-module interactions and ensure that the processor executes instruction sequences accurately under a unified timing environment. The section also introduces timing optimizations in the input module and presents the final functional verification results together with the synthesized top-level schematic.

4.2.1 Input Module Modification and Timing Improvement

During behavioral simulation, it was observed that the accumulator (ACC) register consistently updated its value one instruction cycle later than expected. Although the arithmetic logic unit (ALU) produced the correct computational results during the *EXECUTE* phase, these results were not loaded into the ACC until the subsequent instruction. An example of this behavior is shown in the simulation waveform (Fig 4.7), when executing the first instruction *IN* (`address_out = 0`, `ir_reg = 6`), the ALU output (`alu_out = 5`) was not loaded into the accumulator; instead, the ACC still retained the previous value (`alu_out = 0`). It was not until the EXECUTE phase of the second instruction that the earlier ALU result was finally written into the accumulator.

Detailed waveform analysis revealed that this delay originated from the edge-triggered implementation of the input register (`input_reg`). In the original design, `input_reg` was implemented as a positive-edge D flip-flop (DFF), which only sampled the external input (`external_data_in`) on the rising edge of the clock when `InputReg_LE` was asserted. Since both the input register and the accumulator (`acc_val`) were updated on the same clock edge, the ALU could only access the previously latched input during its current *EXECUTE* phase. As a result, the correct ALU output became available one cycle later, introducing a timing misalignment between input sampling and ACC update.

To eliminate this delay without modifying the instruction set architecture or the overall timing scheme, the input register was redesigned as a level-sensitive latch. In this configuration, when the control signal `InputReg_LE` is high, the latch becomes transparent, allowing `external_data_in` to propagate directly to the ALU within the same clock cycle. When `InputReg_LE` goes low, the latch holds the last stable value. This modification effectively removes the additional one-cycle delay: the ALU now operates on the latest input, and the accumulator can be updated correctly within the same instruction's *EXECUTE* phase. As shown in the behavioral simulation results after the modification Fig 4.8, this adjustment realigns the timing between the input and accumulator, restoring proper synchronization of instruction execution.

From a system-level perspective, this adjustment ensures that each program

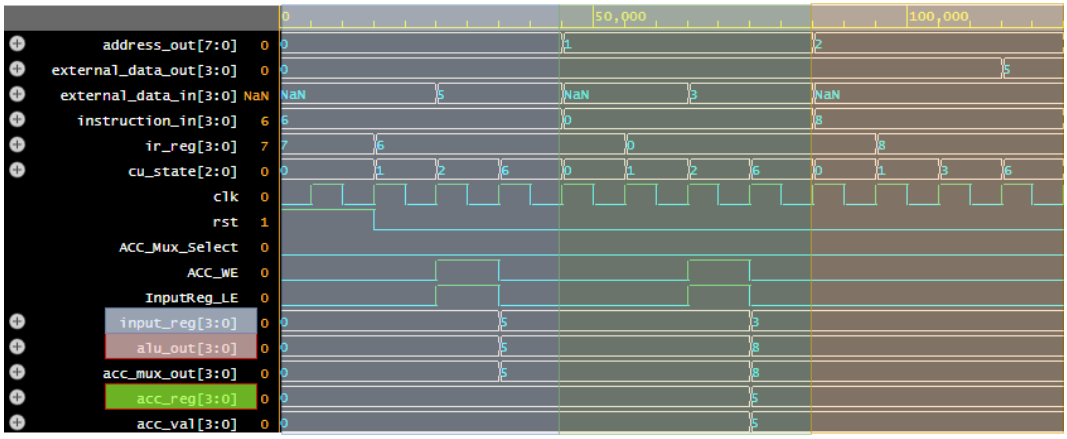


Figure 4.7: Behavioral simulation showing the one-cycle delay between ALU output and accumulator update caused by the edge-triggered input register. Different color-coded regions represent the execution intervals of each instruction. Signals in the key path are also marked.

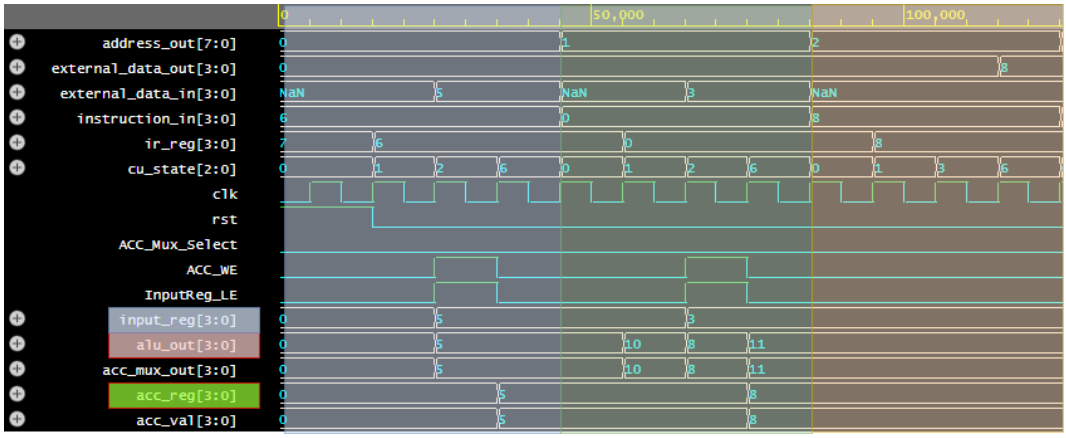


Figure 4.8: Behavioral simulation result of the processor after input latch modification, showing correct four-cycle instruction execution. Different color-coded regions represent the execution intervals of each instruction. Signals in the key path are also marked.

instruction completes within four clock cycles, thereby maintaining the intended operational rhythm and functional consistency of the processor. Although the use of a level-sensitive latch introduces greater sensitivity to signal variations compared to an edge-triggered register, the effect remains well-controlled and acceptable in this context, since the input signal is stable within the latch enable window and precisely managed by the control unit. Consequently, this modification successfully resolves the timing deviation observed during behavioral simulation.

4.2.2 Functional Verification of Processor Operation

To verify the correctness of the overall processor operation after timing modification, a behavioral simulation was performed using the following test program sequence: `IN 5`, `ADD 3`, `OUT`, `NOP`, `CLR`, `OUT`, and `HALT`. As shown in the simulation waveform (Fig. 4.9), each instruction executes within four clock cycles. The control unit successfully generates the corresponding enable signals (`InputReg_LE`, `ACC_WE`, `OutputReg_WE`, etc.) in the correct clock phases, consistent with the results of the control unit's standalone verification presented in Section 4.1.

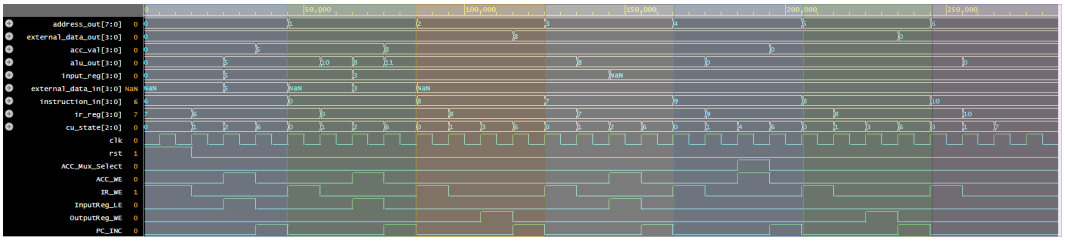


Figure 4.9: Behavioral Simulation Waveform of Processor. Different color-coded regions represent the execution intervals of each instruction.

During the execution of the `IN 5` and `ADD 3` instructions, the ALU performs the addition operation correctly, and the ALU result is stored into the accumulator register (`acc_val = 8`) in the same instruction cycle. When the `OUT` instruction is executed, the stored value is successfully transferred to the output port, resulting in `external_data_out = 8`. Subsequently, when the `CLR` instruction is issued, the accumulator value is reset to zero, and the following `OUT` instruction outputs `external_data_out = 0`. Finally, upon reaching the `HALT` instruction, the processor ceases operation as expected, and the control unit stops generating any further enable signals. These results demonstrate that the final version of the processor's logic design achieves fully correct functional operation and maintains proper timing synchronization throughout all instruction cycles.

4.2.3 Schematic Synthesis of TOP-Level

As shwon in Figure 4.10, the top-level schematic of the processor represents the concrete hardware realization of the processor architecture introduced in Chapter 3, where the functional connections and control signal paths are fully translated

into gate-level logic. This synthesized structure clearly reflects how the processor's data flow and control logic are integrated at the circuit level. It also serves as an essential reference for the top-level circuit integration discussed in Chapter 5.

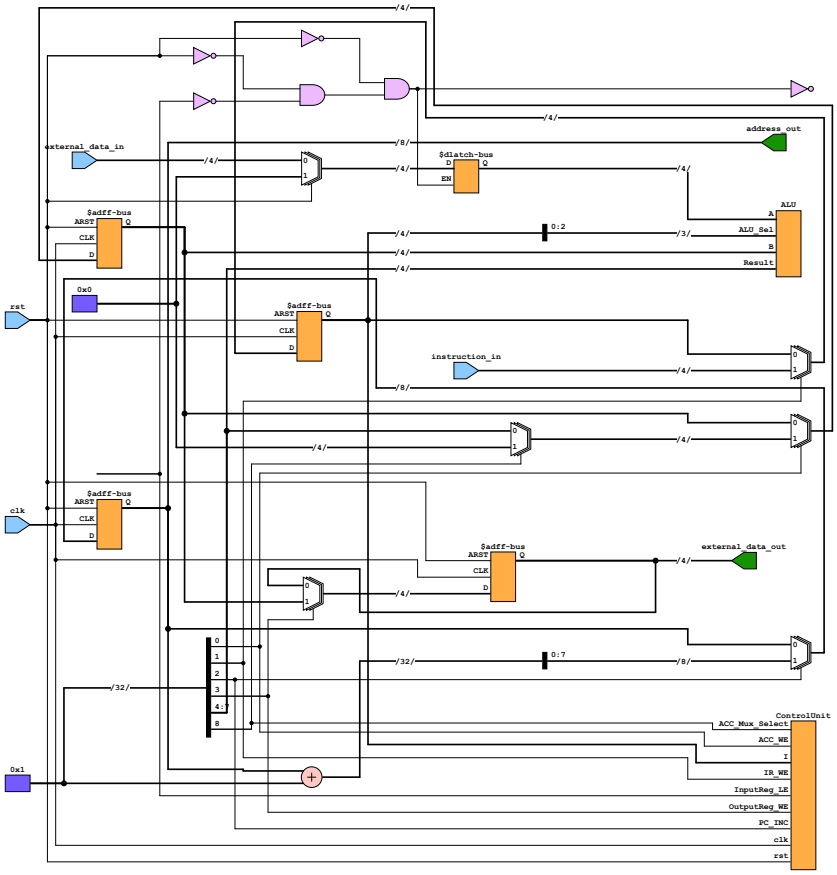


Figure 4.10: TOP-level schematic of 4-bit RISC Processor synthesized by Yosys.

4.3 Summary of Logical Design

This chapter successfully transitioned the processor's design from a high-level architectural concept to a concrete, gate-level implementation. Following the successful integration of all processor modules, comprehensive behavioral simulations demonstrated the processor's complete functional correctness, confirming that the overall logic design performs as intended. The chapter's work culminated in a verified, top-level gate-level schematic, establishing a solid and validated logical foundation for the subsequent circuit-level implementation and simulation detailed in Chapter 5.

Chapter 5

Circuit-Level Design and Simulation in SiC

Building upon the logical design established in the previous chapter, this chapter delves into the circuit-level implementation and simulation verification of the 4-bit SiC RISC processor. The content covers the detailed circuit design of fundamental components, ranging from basic logic gates, key operational and storage units, to the control logic, culminating in the integration of the complete top-level processor circuit. All designs are based on the Fraunhofer IISB 2 µm SiC CMOS process, and their functional correctness and performance are verified through simulation across the temperature range from room temperature up to 500°C.

5.1 Logic Gate Design

The transistor sizing for the standard logic cells in this work is constrained by the fabrication limits of the Fraunhofer IISB 4H-SiC CMOS process, in which the die area is restricted to 5 mm $\times$ 5 mm with an estimated transistor budget of only about 3000 devices. To improve logic density and overall processor speed while maintaining functionality and high-temperature reliability, the standard cells of logic gates must be carefully optimized at the device level. To achieve this, this section will first establish the fundamental transistor sizing—including channel length (L), NMOS width (W_n), and the W_p/W_n ratio—through a detailed analysis of the CMOS inverter. Subsequently, it will demonstrate how these baseline parameters are used to derive other essential logic gates.

5.1.1 SiC CMOS Inverter Characteristics and Transistor Sizing

The sizing of transistors follows a bottom-up procedure derived from both the process design kit (PDK) constraints and analysis of simulation results from a reference inverter. In the standard-cell design of SiC CMOS circuits, this methodology ensures that all logic gates exhibit consistent electrical characteristics and design portability across wide thermal ranges. Since the CMOS inverter represents

the simplest and most fundamental logic element, its simulated electrical behavior under various bias and temperature conditions provides a reliable foundation for parameter optimization throughout the entire standard-cell library. Based on this reference, circuit-level simulations at room temperature (27 °C), 200 °C, and 500 °C are conducted to determine key parameters such as channel length, transistor width, and the W_p/W_n ratio.

First, the channel length (L) is selected according to the minimum geometry and reliability constraints defined in the PDK, ensuring stable operation, limited leakage, and consistent threshold behavior at elevated temperatures. Next, a reference NMOS width (W_n) is chosen to provide sufficient drive strength, and the PMOS width (W_p) is adjusted to obtain a nearly symmetric inverter transfer characteristic with a switching threshold close to $V_{DD}/2$ over the target temperature range. The influence of PMOS drive variation with temperature is inherently reflected in these inverter simulations, allowing the W_p/W_n ratio to be tuned to achieve balanced rise and fall times. After the reference cell is established, this configuration represents the standard (X1) drive strength. Higher drive-strength variants (e.g., X2, X3) are then derived by scaling the transistor widths according to fan-out and load requirements, providing scalable building blocks for subsequent logic-gate and circuit implementations.

This sizing procedure ensures that all transistor parameters comply with PDK design constraints and maintain a consistent trade-off between functionality, speed, and temperature resilience. Based on this methodology, other digital cells such as AND, NOR, MUX, and buffer structures are also derived, ensuring uniform electrical performance and process consistency across the SiC CMOS standard-cell library.

Transistor Length Selection

In digital circuit design, the selection of the channel length (L) directly determines the speed, area, and high-temperature reliability of logic cells. According to the long-channel approximation, the saturation current of a MOS transistor is inversely proportional to the channel length according to equation (5.1) and the propagation delay of a logic gate can be approximated as equation (5.2). A shorter channel length therefore provides higher drive current per unit width and lower intrinsic delay, while enabling smaller device dimensions under the same performance target. This leads to improved energy efficiency and higher integration density of the standard-cell library.

$$I_{\text{on}} \approx \frac{1}{2}\mu C_{\text{ox}}\frac{W}{L}(V_{\text{GS}} - V_{\text{th}})^2, \tag{5.1}$$

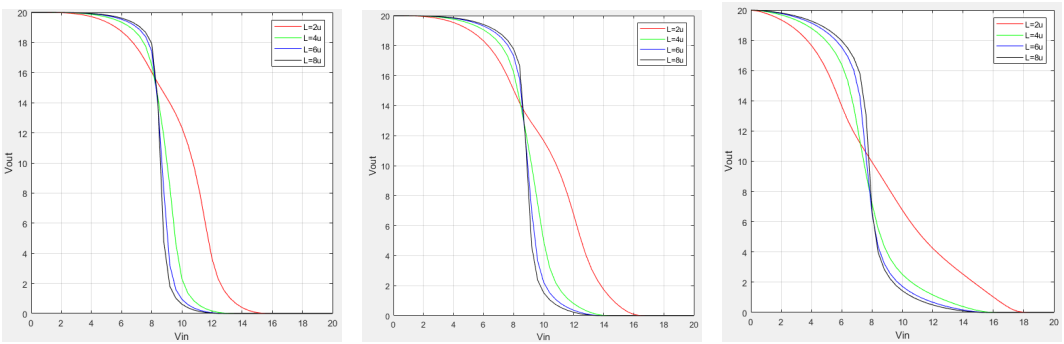
$$t_p \approx \frac{C_L V_{\text{DD}}}{I_{\text{on}}} \propto L. \tag{5.2}$$

To evaluate the influence of channel length on device and circuit behavior, inverter transfer characteristics (VTC) were simulated for $L = 2\,\mu\text{m}, 4\,\mu\text{m}, 6\,\mu\text{m}$, and

$8\,\mu$m at room temperature (27 °C) and at elevated temperatures of 200 °C and 500 °C. All simulations were performed using the final standard-cell transistor parameters ($W_p/W_n = 3{:}1$, $W_n = 8\,\mu$m, $W_p = 24\,\mu$m), with a 20 V supply voltage and BSIM4SIC device models provided by the Fraunhofer IISB 4H-SiC CMOS PDK. The results are shown in Figs. 5.1.

At room temperature (27 °C), shortening L from 8 µm to 4 µm only slightly shifts the switching point toward higher input voltage, while maintaining a steep transition and full logic swing. However, at $L = 2\,\mu$m, the VTC slope becomes noticeably shallower and the threshold voltage increases, indicating the onset of short-channel effects (SCE). Although the inverter still operates, the reduced noise margins and VOH/VOL degradation already signal limited robustness.

At 200 °C and 500 °C, the degradation becomes more pronounced. For $L = 2\,\mu$m, the output no longer reaches full logic levels and the transfer curve becomes almost linear, showing a loss of digital behavior. At $L = 4\,\mu$m, the inverter still switches but exhibits decreased slope. In contrast, $L = 6\,\mu$m and $L = 8\,\mu$m maintain steep transitions and nearly complete output swings ($\approx 20\,\text{V}/0\,\text{V}$) across the temperature range, confirming their superior high-temperature stability.



(a) (b) (c)

Figure 5.1: DC transfer characteristics of inverters with different channel lengths ($L = 2,\ 4,\ 6,\ 8\ \mu$m) under 20 V supply: (a) 27 °C, (b) 200 °C, and (c) 500 °C.

Based on the model validity specifications of the Fraunhofer IISB 4H-SiC CMOS PDK, it indicates accurate matching only for channel lengths $L \geq 5\,\mu$m [14]. Considering that cells with $L = 4\,\mu$m failed to converge in subsequent system-level simulations above 300 °C, this study ultimately selected a uniform channel length of $L = 6\,\mu$m for the standard cell library. This value strikes a good balance between performance and high-temperature reliability, ensuring the circuit's stable operation at extreme temperatures while also meeting timing speed requirements.

Transistor Width Selection

In the standard-cell transistor design, the NMOS width (W_n) directly affects the circuit area, load capacitance, and power consumption. Under the fixed channel

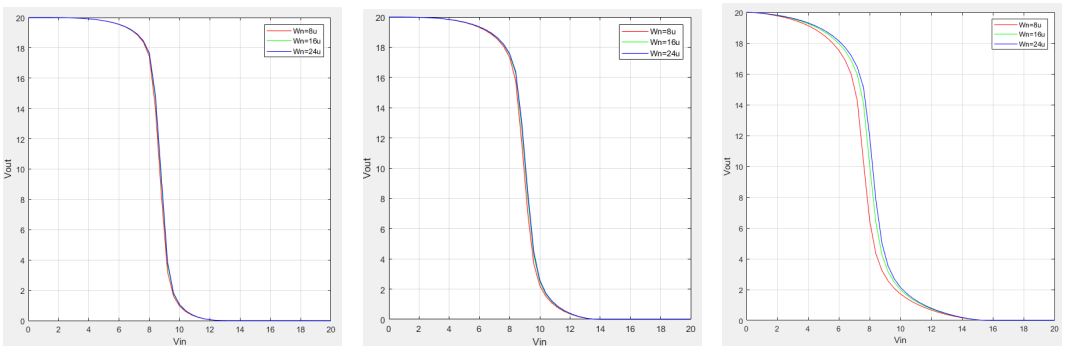
length of $L = 6$ μm and a supply voltage of 20 V, a series of simulations were performed to determine an optimal reference width that balances area efficiency and robustness across the high-temperature operating range.

Reducing transistor width effectively decreases both cell area and gate capacitance, thereby lowering parasitic capacitances and dynamic power. According to equation (5.3) and (5.4), the total load capacitance C_L scales approximately with the device width W; thus, reducing W directly decreases power consumption under identical voltage and frequency conditions. Smaller devices also reduce diffusion and coupling capacitances, improving both area utilization and overall energy efficiency.

$$P_{\text{dyn}} = \alpha C_L V_{DD}^2 f \tag{5.3}$$

$$C_L \approx C_{\text{gate}} + C_{\text{diff}} + C_{\text{wire}} \propto W \tag{5.4}$$

The inverter DC transfer characteristics were simulated for $W_n = 8$ μm, 16 μm, and 24 μm across the previously defined temperature range. The results in Figure 5.2 show that at room temperature and 200 °C, all curves nearly coincide, indicating that the switching voltage and full voltage swing are insensitive to NMOS width. At 500 °C, a slight separation appears, where the inverter with $W_n = 8$ μm exhibits a marginally earlier transition point. This difference originates from the width-dependent drive capability of the NMOS devices, yet the improvement from wider transistors is limited because carrier mobility degradation and the reduction of threshold voltage dominate at high temperature. Consequently, even the minimum-width device maintains complete voltage swing and stable switching behavior.



(a) (b) (c)

Figure 5.2: DC transfer characteristics of inverters with different NMOS width ($W_n = 8$, 16, 24 μm) under 20 V supply: (a) 27 °C, (b) 200 °C, and (c) 500 °C.

From a layout perspective, according to the PDK design rules and the generated transistor PCell in Virtuoso, the minimum active-region width is defined as

10 μm [14]. Therefore, further reducing the drawn width below this value would not yield additional area benefits in practice.

Considering area efficiency, dynamic power, and high-temperature stability, the minimum width of $W_n = 8$ μm is adopted as the reference NMOS dimension for the standard-cell library, with the channel length fixed at $L = 6$ μm. For nodes requiring higher drive capability, such as the program counter (PC) address bus, local scaling is applied only at the final stage using higher drive classes (e.g., X2 or X3), rather than globally upsizing all gates. This localized strengthening approach maintains overall processor area and power efficiency while ensuring reliable signal drive at critical interconnects.

PMOS/NMOS Width Ratio Selection

The selection of the PMOS to NMOS width ratio (W_p/W_n) is a critical step in CMOS logic design, as it directly determines the symmetry of the voltage transfer characteristic (VTC) and, consequently, the noise margins of the logic gates. The primary goal is to center the switching threshold (V_m) at half the supply voltage ($V_{DD}/2$), which in this 20 V process is 10 V, to ensure balanced rise and fall times and maximize noise immunity.

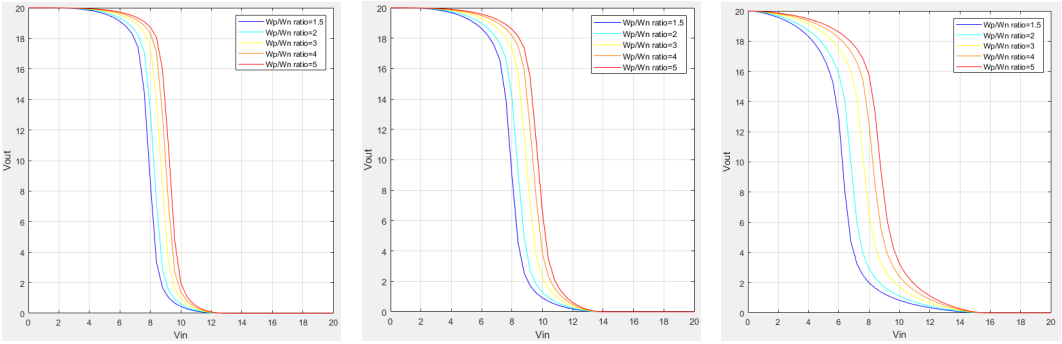
Due to the significantly lower hole mobility compared to electron mobility in SiC, a W_p/W_n ratio greater than one is required to balance the drive strengths of the PMOS and NMOS transistors. The theoretical geometric ratio required to achieve a specific switching threshold V_m can be calculated. Applying equation (5.5) [19] with the specific process parameters, previous research has calculated that a ratio of approximately 5.7 would be needed to perfectly center the switching point at room temperature [20].

$$\frac{W_p/L_p}{W_n/L_n} = \frac{k'_n V_{dsat,n}(V_m - V_{th,n} - \frac{V_{dsat,n}}{2})}{k'_p V_{dsat,p}(V_{dd} - V_m + V_{th,p} + \frac{V_{dsat,p}}{2})} \tag{5.5}$$

To experimentally determine the optimal ratio for this design, simulations of the inverter's DC transfer characteristics were performed for W_p/W_n ratios ranging from 1.5 to 5 at the three key temperature points. The results are shown in Figure 5.3. At room temperature (Figure 5.3a), a clear trend is visible: as the W_p/W_n ratio increases, the switching threshold shifts to the right, moving closer to the ideal 10 V center. This observation aligns with the theoretical calculations.

However, the primary application for this SiC technology is high-temperature operation. The VTC curves at 200 °C and 500 °C (Figure 5.3b, 5.3c) reveal a significant leftward shift in the switching threshold for all ratios as temperature increases. This behavior highlights that a ratio optimized for room temperature may not be suitable for extreme environments. As discussed in the literature review (Section 2.3.2), this trade-off has been a key consideration in similar SiC processes. Extensive studies by Holmes et al. on the HiTSiC® 4H-SiC CMOS process have shown that while a 5:1 ratio performs well up to approximately 300 °C, a 3:1 ratio

provides a superior drive balance and timing consistency at extreme temperatures (above 300 °C) [6]. It offers a better compromise for device performance across a wide thermal range, mitigating excessive delay mismatch at high temperatures.



(a) (b) (c)

Figure 5.3: DC transfer characteristics of inverters with different W_p/W_n ratio (8, 16, 24 μm) under 20 V supply: (a) 27 °C, (b) 200 °C, and (c) 500 °C.

Considering these factors, and prioritizing robustness in high-temperature environments over perfect symmetry at room temperature, a W_p/W_n ratio of 3:1 was selected for the standard cell library in this work. This choice represents an engineering trade-off that ensures reliable logic functionality across the entire target operating range from 27 °C to 500 °C, while also offering a more area-efficient design compared to cells with larger ratios.

Summary

Through detailed simulation and analysis of SiC CMOS inverters across various geometric dimensions and temperatures, this study has established an optimized set of fundamental transistor parameters for the processor design.

To achieve a balance between performance and high-temperature reliability, a channel length (L) of 6 µm was ultimately selected for the standard cells. Building on this, and with comprehensive consideration for area efficiency, drive capability, and the need for drive balance at extreme temperatures up to 500 °C, a W_p/W_n of 3:1 was adopted upon setting the reference NMOS width (W_n) to 8 µm. These parameters collectively define the baseline (X1) drive strength for the standard cell library. For nodes requiring higher drive capability, a local strengthening approach is employed, using cells with higher drive classes only at the final output stage.

Ultimately, the fundamental transistor dimensions adopted for this processor design are: L = 6 µm, W_n = 8 µm, and W_p = 24 µm. These carefully selected parameters will serve as the cornerstone for building the entire digital standard cell library, ensuring that all subsequent circuits—from logic gates to complex functional blocks—exhibit consistent, reliable electrical characteristics and high-temperature stability.

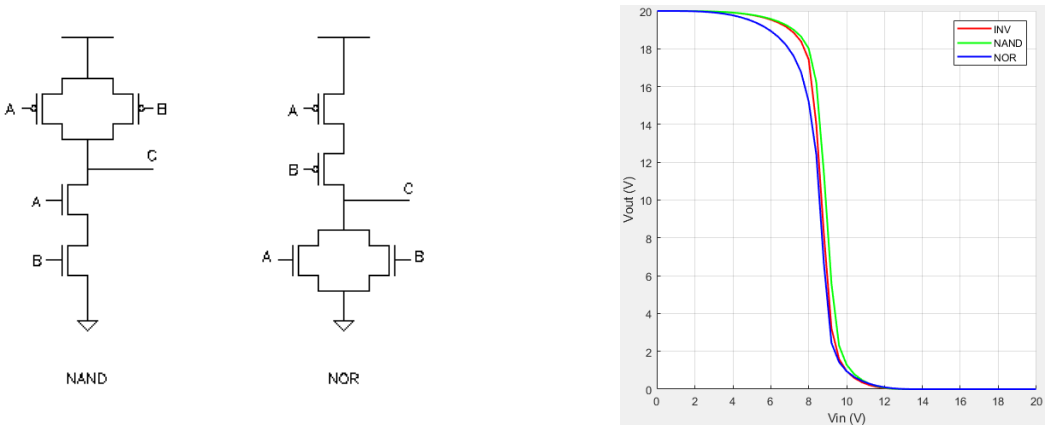
5.1.2 Derivation of Standard Logic Cells

With the fundamental sizing parameters of the CMOS inverter established in the previous section, the design process now extends to the derivation of other essential logic gates required for the standard cell library. This section presents the implementation of several key logic gates used throughout this design.

NAND and NOR Gates Design

The design of NAND and NOR gates follows the principle of matching their drive strength to that of the reference inverter defined in the previous section, thereby ensuring consistent timing characteristics across the entire standard cell library. To validate this sizing strategy, circuit-level simulations were performed on the designed logic gates.

Figure 5.4 shows the voltage transfer characteristics (VTC) of these logic gates at room temperature, overlaid with the curve of the reference inverter. As can be seen from the figure, the switching threshold voltages of the NAND and NOR gates are close to that of the reference inverter, which is approximately 8.7 V. Despite the slight mismatch, all logic gates exhibit high gain and a full logic swing. This discrepancy is within an acceptable range, which validates the overall effectiveness of the derived design methodology. Furthermore, high-temperature simulations also verified that these logic gates can operate stably at 500 °C, and their temperature drift characteristics are consistent with the trend of the reference inverter.



(a) (b)

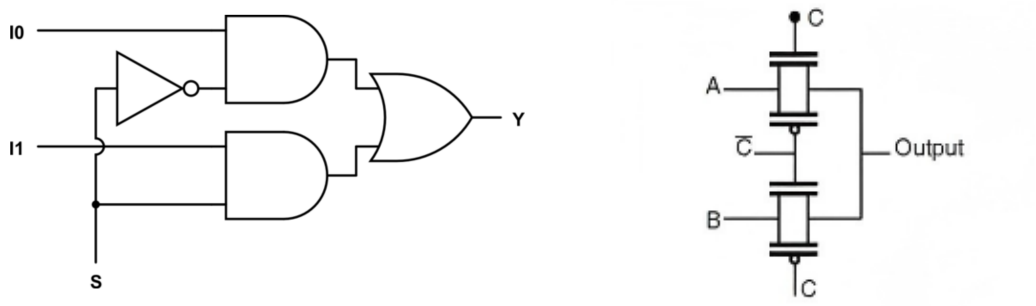
Figure 5.4: (a) Standard logic gate implementation of NAND and NOR [9] (b) DC transfer characteristics of the 2-input NAND and NOR gates at room temperature (27 °C), overlaid with the baseline inverter.

Multiplexer Design using Transmission Gates

The multiplexer (MUX) is a fundamental component in processor design, not only widely used in the Arithmetic Logic Unit (ALU) to select operands or results but also frequently employed in other modules' registers to control data signal inputs. In a standard digital design flow, a 2-to-1 MUX is typically implemented using a combination of AND, OR, and NOT gates, as shown in Figure 5.5a. This implementation, while logically straightforward, requires 20 transistors, leading to a significant layout area overhead for complex modules like the ALU that utilize numerous MUXs.

To optimize logic density, this design adopts a MUX structure based on Transmission Gates (TG), which requires only 6 transistors, as depicted in Figure 5.5b. Although this structure offers a substantial area advantage, it has theoretical drawbacks, such as weaker drive capability and the potential for output voltage level degradation.

To verify the reliability of this structure at high temperatures and evaluate its practical performance, a transient simulation was conducted at 500 °C. The resulting waveform is shown in Figure 5.6. The output signal Y correctly follows input A when the select signal S is low (e.g., 0–20 µs, 40–60 µs) and follows input B when S is high (e.g., 20–40 µs, 60–80 µs), perfectly matching the expected logical functionality. Besides, the simulation results demonstrate that even at the extreme temperature of 500 °C, the output signal Y maintains a full logic swing from 0 V to 20 V without any observable signal level degradation. This successful simulation proves that despite its theoretical drawbacks, the performance of the TG structure is sufficiently robust under the specific SiC process and load conditions of this design. Therefore, given its considerable area savings and proven high-temperature reliability, this structure was ultimately adopted.



(a) (b)

Figure 5.5: Comparison of (a) standard logic gate implementation of a 2-to-1 MUX versus [27] (b) transmission gate implementation of a 2-to-1 MUX [2].

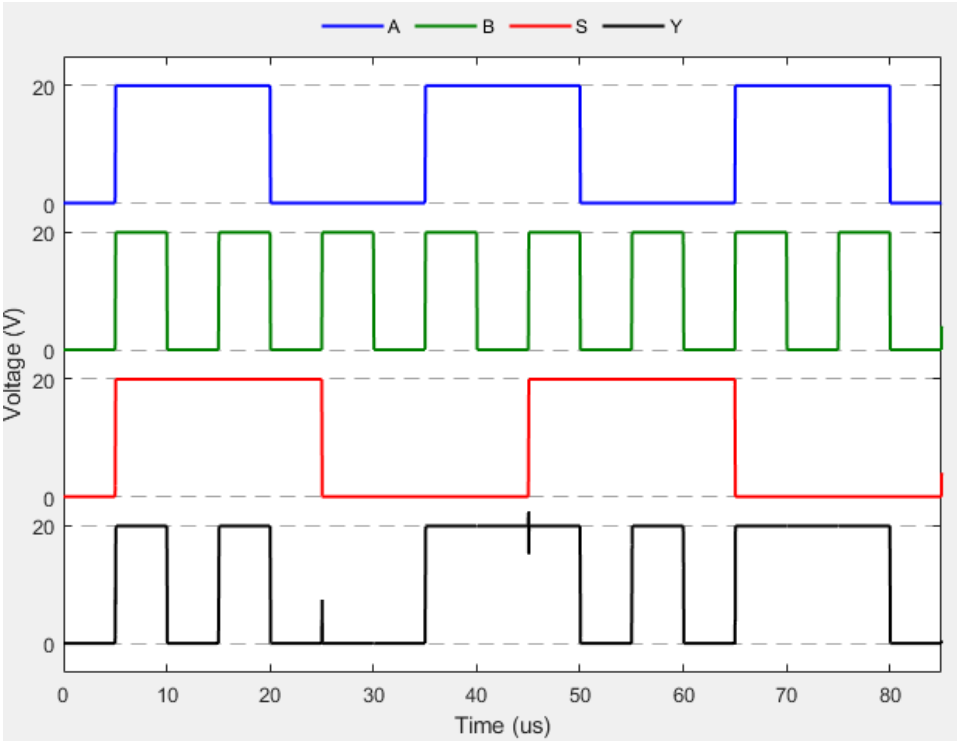


Figure 5.6: Transient simulation of the transmission gate MUX at 500 °C. The output (Y) switches between inputs A and B based on the select signal (S).

5.2 ALU Implementation

As architecturally defined in Chapter 3, the Arithmetic Logic Unit (ALU) is the computational core of the processor, responsible for executing all arithmetic and logical instructions. This section details its circuit-level implementation, translating the architectural concept into a physical structure built from the standard cells designed in Section 5.1.

The 4-bit ALU is constructed using a modular and scalable bit-slice design approach. A single 1-bit ALU slice is designed first and then instantiated four times to create the full 4-bit datapath. The slices are interconnected in a ripple-carry configuration, where the carry-out (c_{out}) of one slice serves as the carry-in (c_{in}) for the next more significant bit slice. This method simplifies the design and layout process. The circuit diagram for a 4-bit ALU is shown in Figure 5.7. It is composed of fundamental logic gates, a full adder, and multiplexers, all of which are implemented using the standard cells (e.g., Inverter, NAND, NOR, TG-MUX) defined in the previous sections. The operation of the slice is governed by the 3-bit `ALU_Sel` signal, which is broadcast to all four slices simultaneously.

- **Logical Operations**: For logical operations such as AND, OR, and XOR, the inputs A and B are directly fed into the respective logic gates. A large 4-to-1

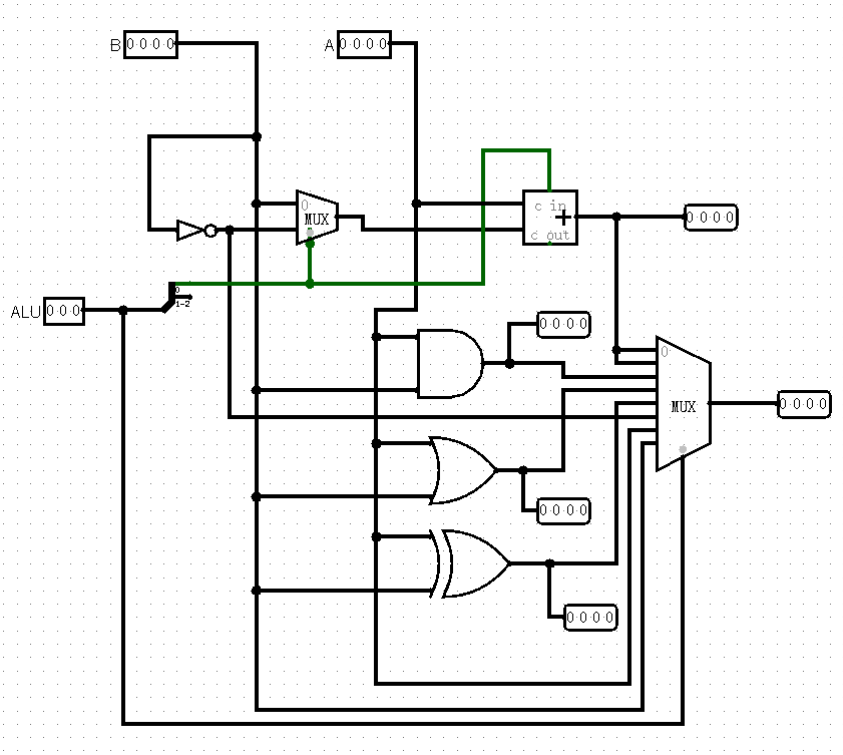


Figure 5.7: Circuit schematic of a 4-bit ALU.

MUX at the output stage then selects the appropriate result based on the `ALU_Sel` control signals (e.g., when `ALU_Sel` is `010`, the output of the AND gate is selected).

- **Arithmetic Operations**: Arithmetic functions are efficiently implemented by a unified binary adder-subtractor circuit. The mode switching of this circuit is directly controlled by a specific bit of the instruction opcode, namely `ALU_Sel[0]`. The circuit uses a set of XOR gates to switch between modes, as shown in Figure 5.8.
  - **Addition**: When the ADD instruction is executed, the control bit `ALU_Sel[0]` is 0. In this case, the XOR gates pass the B operand unchanged to the full adders, and the initial carry-in (C_{in}) is also 0. The circuit performs a standard $A + B$ addition.
  - **Subtraction**: When the SUB instruction is executed, the control bit `ALU_Sel[0]` is 1. In this case, the XOR gates perform a bitwise inversion of the B operand (obtaining the one's complement of B), and the initial carry-in (C_{in}) is also set to 1. The full adder actually performs the operation $A + B' + 1$, which is the standard implementation of two's complement subtraction ($A - B$).

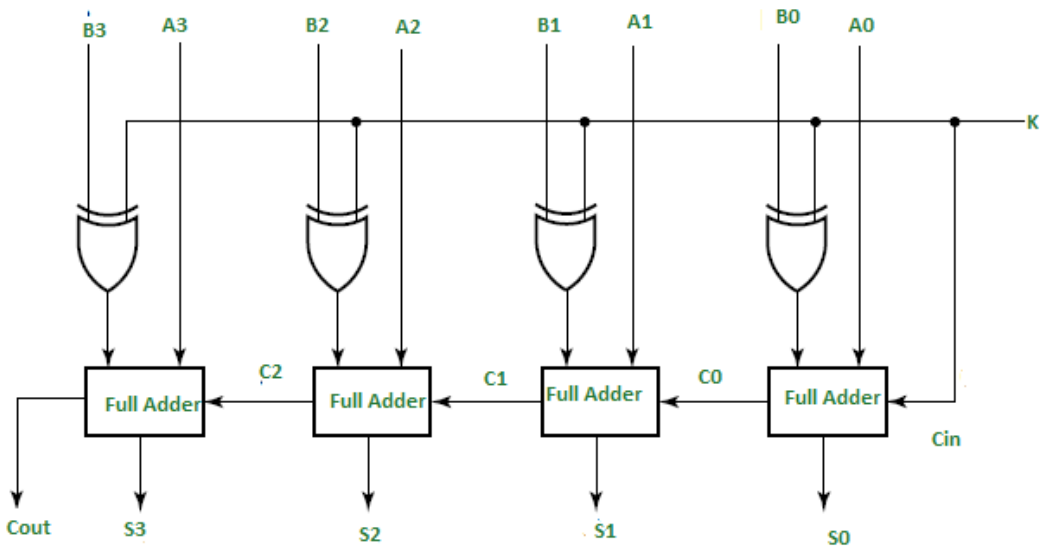


Figure 5.8: Structure of the 4-bit binary adder-subtractor. The control signal K corresponds to the `ALU_Sel[0]` bit from the instruction opcode [18].

Simulation and Verification

To verify the functional correctness and performance robustness of the fully constructed 4-bit ALU, a transient simulation was conducted at a temperature of 500 °C. In the simulation, the input operands A and B were held at constant values of 7 (binary `0111`) and 3 (binary `0011`), respectively. Meanwhile, the 3-bit ALU operation select signal, `ALU_Sel` (S2-S0), iterated through all eight possible combinations from `000` to `111` to test the full functionality of the ALU. The simulation results shown in Figure 5.9 are compared against the theoretical expected values in Table 5.1.

Table 5.1: Theoretical ALU outputs (Y3-Y0) for constant inputs A=7 and B=3.

Time (µs)	S2S1S0	Operation	Y3-Y0
0-10	`000`	A + B	`1010` (10)
10-20	`001`	A - B	`0100` (4)
20-30	`010`	A AND B	`0011` (3)
30-40	`011`	A OR B	`0111` (7)
40-50	`100`	A XOR B	`0100` (4)
50-60	`101`	NOT B	`1100` (12)
60-70	`110`	Pass A	`0111` (7)
70-80	`111`	Pass B	`0011` (3)

Through the comparison, it clearly demonstrates that the 4-bit output of the ALU perfectly matches the theoretical expected value for each operation. Even at

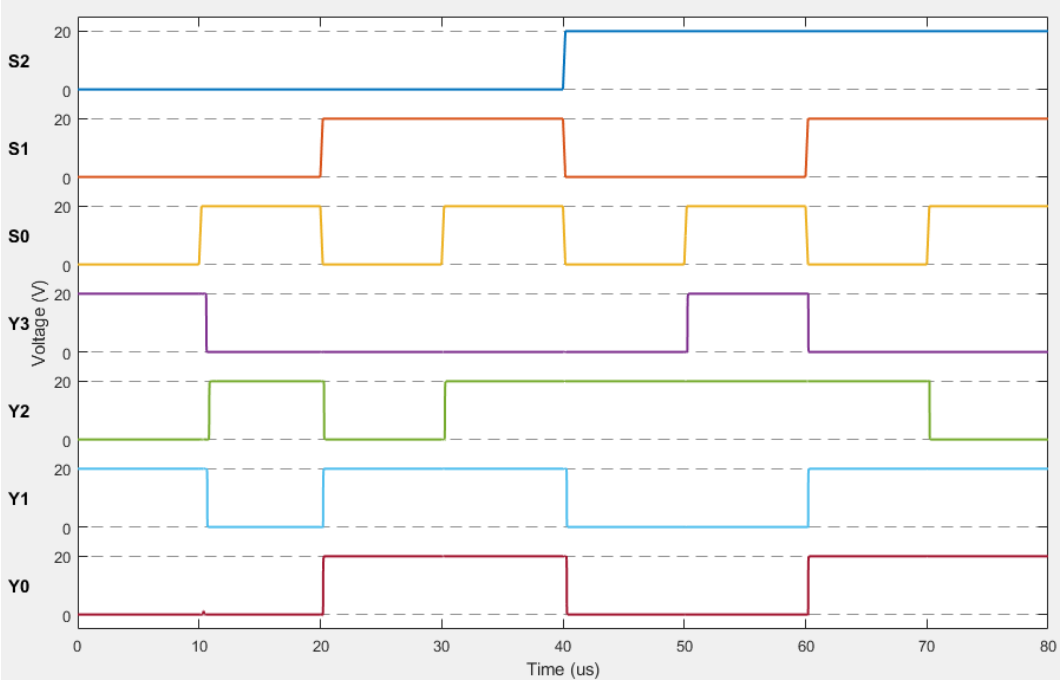


Figure 5.9: Transient simulation waveform of the 4-bit ALU at 500 °C. The resulting output (Y3-Y0) corresponds to the theoretical values listed in Table 5.1.

the high temperature of 500 °C, all output signals exhibit a full logic swing from 0 V to 20 V with clean transitions. This successful simulation not only verifies the correctness of the ALU's circuit-level design but also confirms its high reliability in an extreme temperature environment, further validating the robustness of the adopted standard cell library and design methodology.

5.3 Memory Components Implementation

This section details the circuit-level implementation of the processor's core memory and sequential logic components. These modules are fundamental for storing state, data, and instruction addresses, and include the Program Counter (PC), various registers (Instruction Register (IR), Accumulator (ACC), Output Register), and the Input Latch. All these components are built upon basic sequential logic units such as D-type Flip-Flops and Latches.

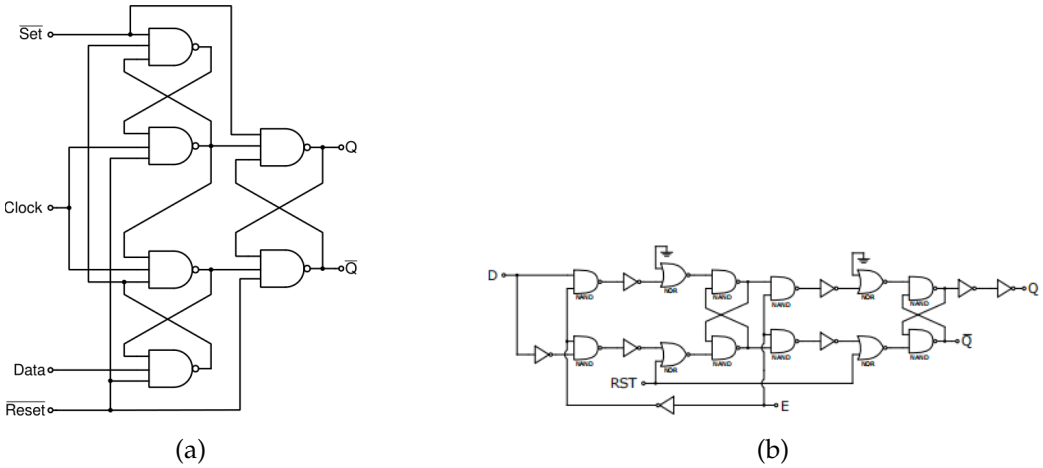
For the sake of clarity and consistency in this thesis, the functional verification for each module will be demonstrated using transient simulation results at 500 °C. However, it is important to note that based on multiple simulation verifications, all Memory Components designed in this chapter can operate stably and correctly over the full temperature range from room temperature (27 °C) to 500 °C.

5.3.1 D flip-flops and Registers

The main storage units in the processor, including the PC, IR, ACC, and the Output Register, are all constructed based on a fundamental sequential logic block: the D-type Flip-Flop (DFF). To ensure the absolute stability of register functions in high-temperature and complex operating environments, this design adopts a classic Edge-triggered Master-Slave D-Flip-Flop, which is based on NAND gates. Its circuit is shown in Figure 5.10a. This structure operates with a two-stage latching mechanism (master and slave), which effectively prevents race conditions in critical timing paths and offers very high reliability. Compared to the DFF structure used in prior work by Romijn et al. shown in Figure 5.10b within the same process, our design is optimized while maintaining high reliability, requiring only 34 transistors, which significantly reduces the module's area.

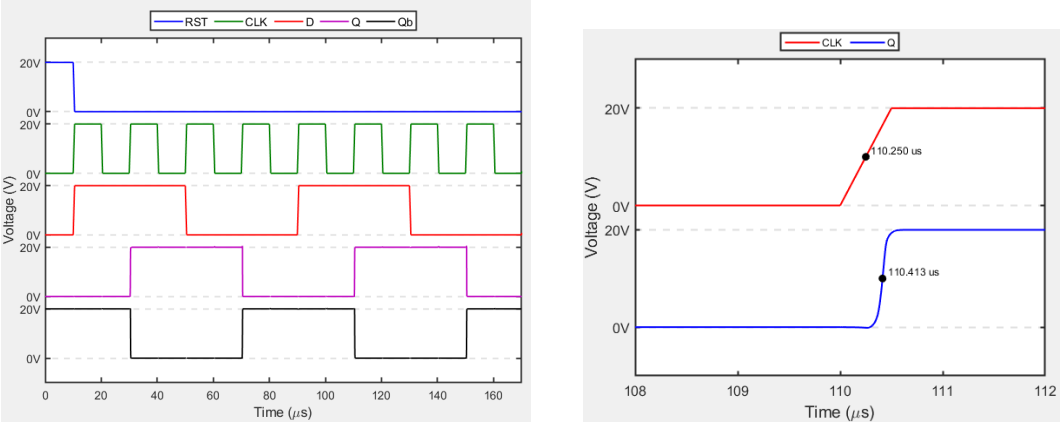
We use the transient simulation at 500 °C as an example to verify the functionality and reliability of the DFF circuit, with the waveform shown in Figure 5.11a. The simulation results demonstrate that at each rising edge of the clock (CLK), the outputs (Q and Qb) accurately sample and latch the level of the input (D) while maintaining a full logic swing, proving that the DFF circuit can operate normally and reliably at high temperatures.

Furthermore, according to several simulation results from 27 °C to 500 °C, this circuit can operate stably across the entire temperature range. There exist some variations in the clock-to-q delay (tcq) at different temperatures, which will be discussed in the following timing analysis. This successful simulation verifies the correctness of the chosen DFF structure and its excellent stability at high temperatures, providing a solid foundation for building all the register modules within the processor.



(a) (b)

Figure 5.10: Comparison of (a) Edge triggered D flip flop structure used in this work (34 transistors) [25] and (b) D flip flop structure from prior work by Romijn et al (66 transistors) [21].



(a) (b)

Figure 5.11: (a) Transient simulation of the adopted DFF at 500 °C and (b) measurement of DFF clock-to-output delay (T_{CQ}).

Timing Analysis

The clock-to-output delay (T_{CQ}) is a critical timing parameter for sequential logic elements like D flip-flops. It represents the time required for the flip-flop's output (Q) to change and stabilize after the triggering edge of the clock signal (CLK) arrives. Specifically, T_{CQ} is measured from the point where the clock signal reaches 50% of VDD to the point where the output Q reaches 50% VDD as shown in Figure 5.11b. The triggering rising edge of the CLK signal reaches 10V at 110.250 µs. In response, the output Q begins to transition and reaches 10V at 110.413 µs. Thus, the T_{CQ} at 500°C is approximately 0.163 µs.

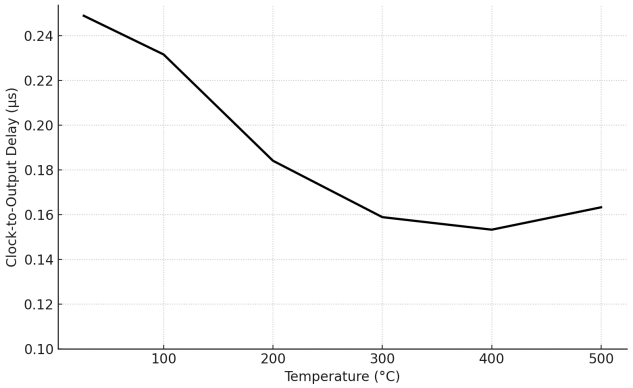


Figure 5.12: Simulated DFF T_{CQ} with temperature.

Figure 5.12 shows the simulated clock-to-output delay (T_{CQ}) of the D flip-flop over temperature. The delay decreases from room temperature to about 400 °C, then slightly increases at 500 °C. This non-monotonic trend originates from the

competing effects influencing the transistors' drive strength and switching speed. Below 400 °C, factors contributing to faster switching dominate: the significant decrease in threshold voltages and the reduction in contact resistance of PMOS devices enhance the effective drive strength, leading to a shorter gate delay [15], leading to faster switching. At higher temperatures, phonon scattering becomes dominant, and the resulting carrier mobility degradation begins to limit the current, counteracting the benefits from lower threshold and contact resistance [6], thus causing a small rise in T_{CQ}.

The observed trend agrees with previously reported results of SiC CMOS logic and sequential circuits [6, 22], confirming that the simulation accurately reflects the intrinsic high-temperature behavior of the Fraunhofer IISB SiC CMOS process.

5.3.2 Program Counter

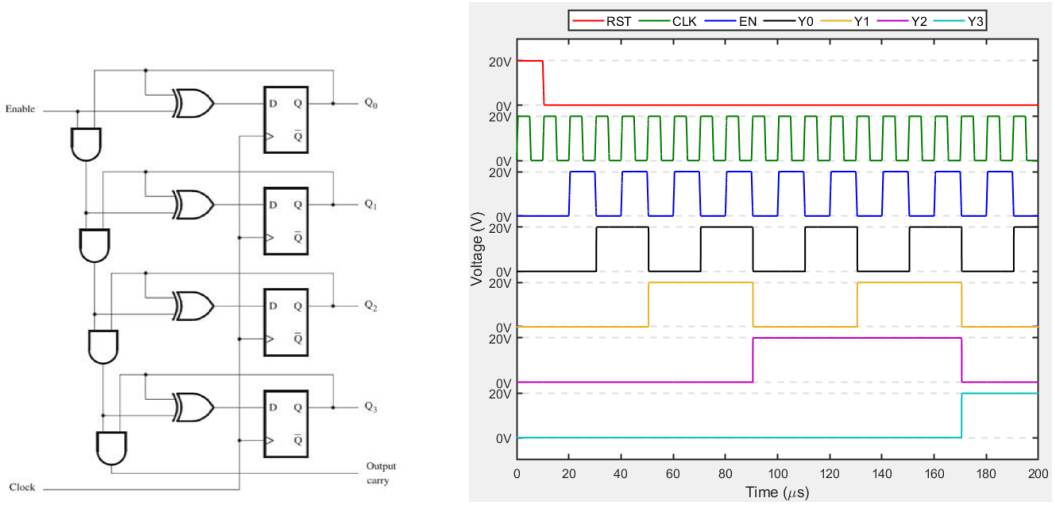
The program counter (PC) serves as a representative example of the register modules in the processor that are constructed from DFFs. It sequentially generates instruction addresses during program execution and is implemented as a synchronous binary counter. Each bit of the counter is realized using a DFF, while XOR and AND gates form the combinational logic required for incrementing. When the enable signal (EN) is high, the PC increments its value on each rising clock edge; when the reset signal (RST) is asserted, all output bits are cleared to zero.

The structural concept follows the 4-bit synchronous counter illustrated in Fig. 5.13a, which is extended to an 8-bit design in this work to accommodate a larger instruction addressing range. The circuit shares a common clock across all stages to ensure synchronized counting and minimize skew.

Figure 5.13b shows the transient simulation results of the program counter at 500 °C with a clock period of 10 µs. At the start of the simulation, the RST signal is asserted, initializing all counter bits to logic low. Once released, the counter begins to increment synchronously on each clock rising edge when EN is enabled. The output waveforms Y_0–Y_3 clearly exhibit a standard binary counting sequence, while the count is held constant whenever EN is low. All outputs maintain full voltage swings and stable edge transitions, confirming the reliability and robustness of SiC CMOS–based DFF register structures under extreme temperature operation.

5.3.3 Input Latch

The input latch module maintains stable input data during the *EXECUTE* phase of processor operation, ensuring that external input signals remain unchanged throughout instruction execution. The module is constructed using a D latch structure, as shown in Fig. 5.14a. When the enable signal is high, the output follows the input; when the enable is low, the latch holds the previous value. This



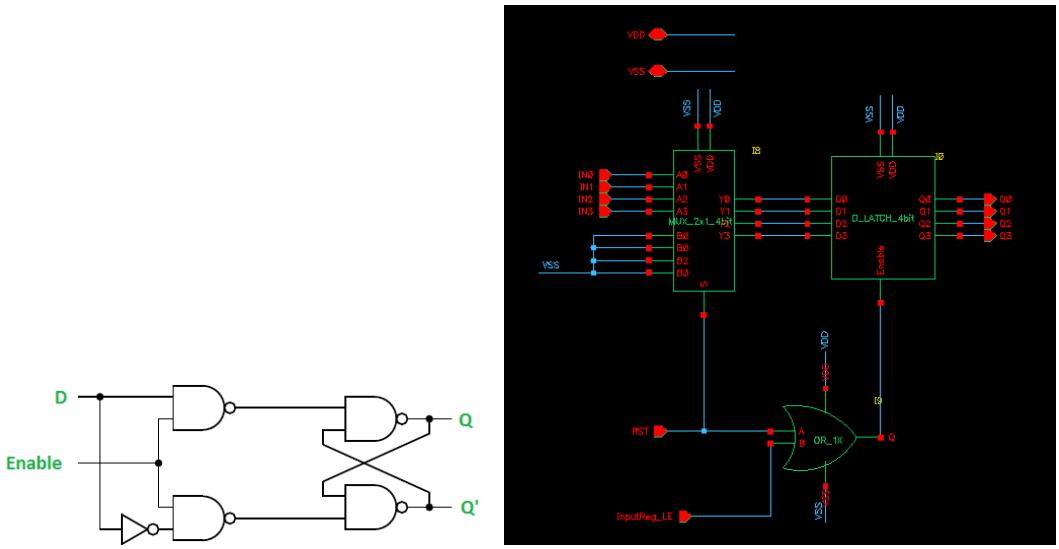
(a) (b)

Figure 5.13: (a) 4-bit synchronous counter schematic [24] and (b) transient simulation of the program counter at 500 °C.

transparent behavior makes it well suited for temporarily storing external inputs and synchronizing them with internal control signals.

At the system level, the input latch is implemented as a 4-bit latch circuit, whose schematic is shown in Fig. 5.14b. External input signals (IN0–IN3) are first routed through a multiplexer before being passed to the D latch. The latch enable signal is generated through an OR gate combining the input enable (*InputReg_LE*) and reset (RST) signals. When RST is high, the enable signal is driven high and a ground-level "0" value is written into the latch, thereby resetting all outputs. When RST is low and the input enable is active, the external input signals are sampled and held at the outputs Q0–Q3. This configuration ensures mutual exclusivity between the reset and sampling conditions, effectively avoiding race conditions and timing uncertainty.

The transient simulation at 500 °C is shown in Fig. 5.15. In this test, input channels D3 and D0 are selected to demonstrate the functional behavior. When the *InputReg_LE* signal is asserted high, the corresponding Q outputs follow the D inputs; when *InputReg_LE* returns low, the outputs remain latched. At 130 µs, the RST signal is driven high, and all outputs are immediately cleared, confirming the correct operation of the reset logic. The results verify that the input latch performs accurate sampling, holding, and reset behavior under high-temperature conditions, meeting the timing requirements of the input module in the designed processor.



(a) (b)

Figure 5.14: (a) Basic D latch structure used for the input module and (b) 4-bit input latch implementation.

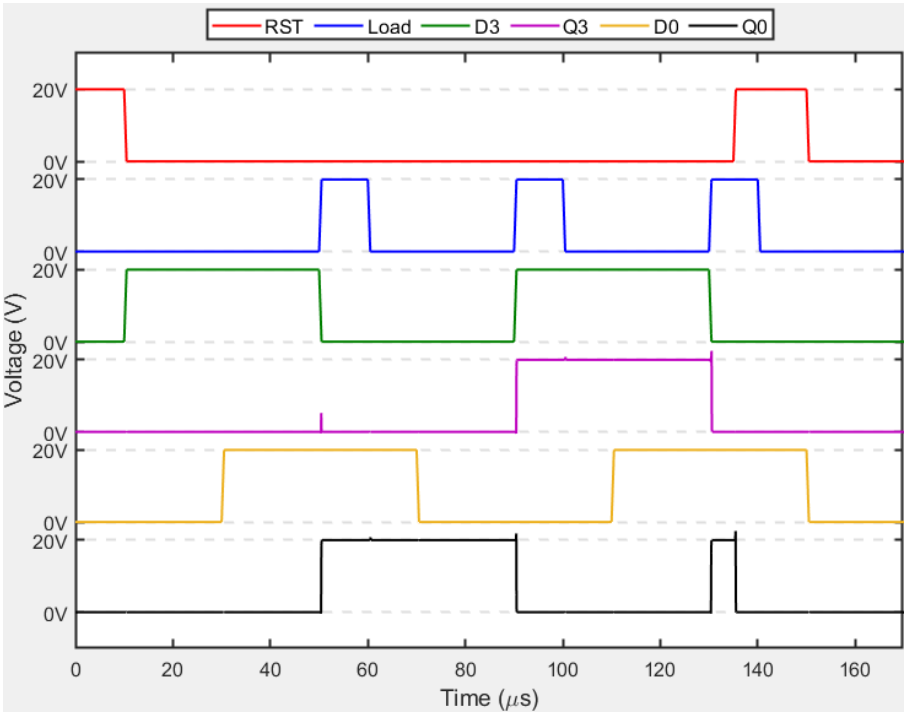


Figure 5.15: Transient simulation of the input latch at 500 °C showing sampling, holding, and reset operations.

5.4 Control Unit Implementation

The control unit (CU) generates the internal control signals of the processor according to the decoded instruction and current timing state. Its logical behavior and state transition have been detailed in Chapter 4, and this section focuses on the circuit-level implementation and verification. The circuit is derived from the gate-level schematic synthesized by Yosys in Chapter 4 and was reconstructed and simulated using the Fraunhofer IISB 4H-SiC CMOS PDK, as referenced in Fig. 4.5.

Figure 5.16 shows the transient simulation results at 500 °C, obtained with a clock period of 10 µs. Each instruction therefore required a total execution time of 40 µs, consisting of four internal stages: *Fetch*, *Decode*, *Execute*, and *End_Cycle*. The simulated 5 instruction sequence two ALU instructions followed by three non-ALU instructions. All control signals (*ACC_WE*, *PC_INC*, etc.) are correctly generated within each instruction phase, achieving the timing coordination among the memory and arithmetic modules in the processor. Their transitions fully match the behavioral simulation results presented in Section 4.1.2, confirming the functional consistency between the gate-level and behavioral designs. The simulation results demonstrate that the circuit-level control unit maintains stable operation at elevated temperatures and correctly performs the sequential timing control and state switching required by the processor.

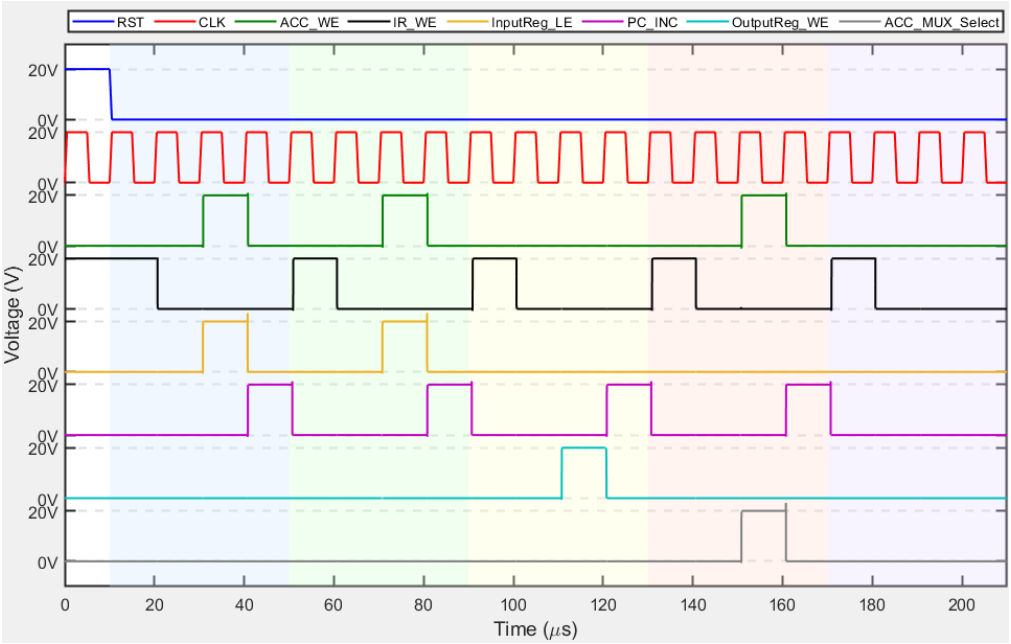


Figure 5.16: Transient simulation of the control unit at 500 °C showing the five-instruction sequence (*IN*, *NOP*, *OUT*, *CLR*, *HALT*). Color-coded intervals represent the execution periods of each instruction.

5.5 Processor Implementation and Performance Analysis

Having completed the circuit-level design and verification of key modules such as the ALU, memory components, and control unit, this section focuses on integrating these blocks into the complete top-level circuit of the 4-bit SiC processor. It first details the final processor circuit implementation, followed by a comprehensive transient simulation to verify its functional correctness at 500°C, and concludes with an analysis of its performance across different temperatures.

5.5.1 Processor Implementation

The processor top-level circuit was derived from the Yosys-synthesized schematic of the processor (Fig. 4.10) presented in Chapter 4. Similar to the control unit implementation, the top-level circuit was then built and simulated in Virtuoso as shown in Fig. 5.17.

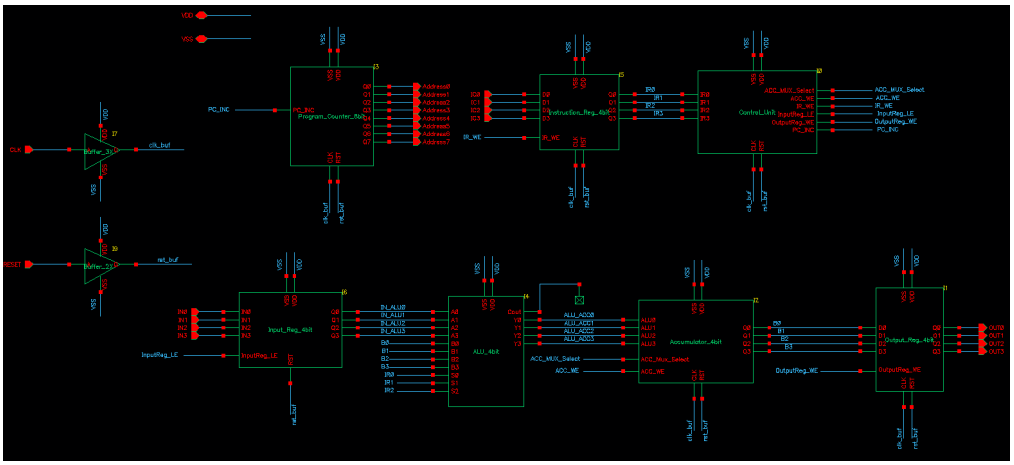


Figure 5.17: Top-level schematic of the 4-bit SiC processor in Virtuoso.

During reconstruction, several structural refinements and optimizations were applied to the top-level circuit to improve structural clarity and signal integrity. For example, the MUX circuits controlling the enable signals for each register module were moved inside their respective submodules, reducing top-level interconnect complexity. Similarly, the bitwise increment logic, represented as "+" in the synthesized schematic, was directly embedded within the program counter module, streamlining the datapath connection as described previously. These structural adjustments enhance the design's modularity and simplify inter-module timing analysis.

Furthermore, circuit-level optimizations were implemented to bolster signal integrity and robustness. Buffers were added to critical global signals, namely the clock (CLK) and reset (RST), to reduce fan-out loading, equalize propagation delays across the chip, and ensure reliable distribution vital for synchronous operation under varying conditions. Buffers were also added to the output signals,

including the 8-bit Address Bus and the 4-bit Data Out, to enhance their drive strength, improve noise immunity, and better isolate the processor core from external load variations or interference.

Simulation conditions were set as follows: the supply voltage was 20 V, temperature 500 °C, and clock period 4 µs. This clock period was chosen to verify if the processor can operate correctly near the theoretically calculated minimum clock period (Tclk). A detailed analysis regarding the maximum operating frequency (Fmax) will be presented in the following Processor Performance Analysis section. Each instruction therefore required a total execution time of 16 µs. To ensure proper initialization, a 20 µs global reset phase was applied at the beginning of the simulation. Afterward, the program counter began execution from zero, following the instruction sequence listed in Table 5.2.

Figure 5.18 presents the transient simulation results at 500 °C. During each instruction period, the instruction register (IR3–IR0) correctly loads the opcode, and the accumulator outputs (B3–B0) exhibit the expected arithmetic and logic results. When the *OUT* instruction is executed, the accumulator value is correctly transferred to the output register (OUT = 11); during *CLR*, the accumulator resets to zero; and after the *HALT* instruction, the processor stops operation and the instruction register no longer updates subsequent instructions. All observed behavior matches the theoretical results summarized in Table 5.2.

Table 5.2: Theoretical instruction sequence and expected execution results for processor verification.(*) indicates that the operand is not used for this instruction.

PC	Instruction	Opcode	Input Data	Operation	ACC Result	Output Reg.
0	IN	0110	5	ACC ← Input	5	0
1	SUB	0001	7	ACC ← Input - ACC (7–5)	2	0
2	AND	0010	2	ACC ← ACC & Input (2 & 2)	2	0
3	OR	0011	3	ACC ← ACC | Input (0 | 3)	3	0
4	XOR	0100	4	ACC ← ACC ⊕ Input (3 ⊕ 4)	7	0
5	NOT	0101	0*	ACC ← NOT ACC (NOT 7)	8	0
6	ADD	0000	3	ACC ← ACC + Input (8 + 3)	11	0
7	NOP	0111	1*	ACC ← ACC	11	0
8	OUT	1000	2*	Output ← ACC	11	11
9	CLR	1001	3*	ACC ← 0	0	11
10	HALT	1010	4*	Halt Processor	0	11
11	ADD	0000	5*	Halted	0	11

The simulation results confirm that the processor operates stably at 500 °C, demonstrating its capability to perform arithmetic, logical, and data operations as designed. All signal transitions are clean, instruction cycles are clearly separated, and no race or glitch behavior is observed, validating the correctness and timing consistency of the circuit-level implementation. Furthermore, according to the simulation interface information, the entire processor utilized a total of 2092 transistors. This figure is well within the initial estimated budget constraint of approximately 3000 devices, confirming the feasibility of integrating the complete processor layout within the specified $5\ mm \times 5\ mm$ die area.

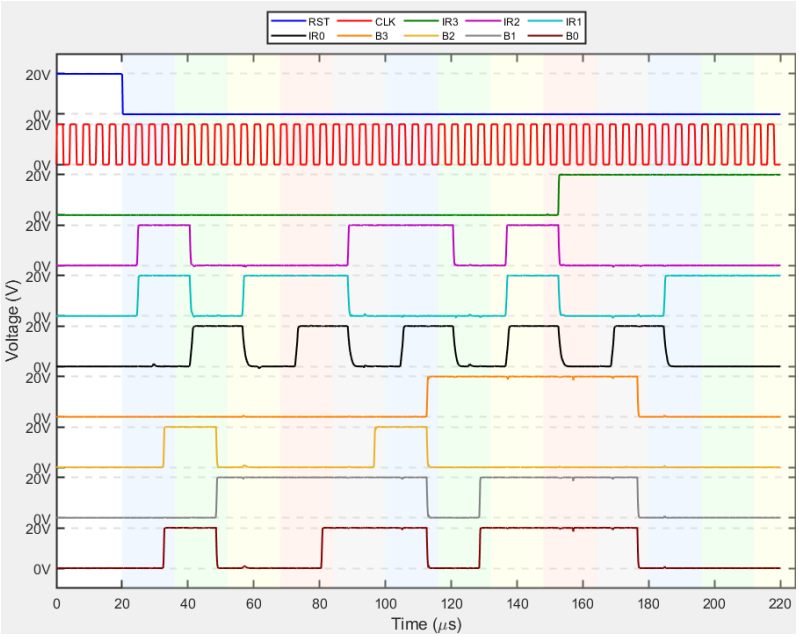


Figure 5.18: Top-level transient simulation of the instruction register (IR3–IR0) and accumulator (B3–B0) at 500 °C. Different color-coded regions represent the execution intervals of each instruction. The simulated sequence matches the expected functional behavior listed in Table 5.2.

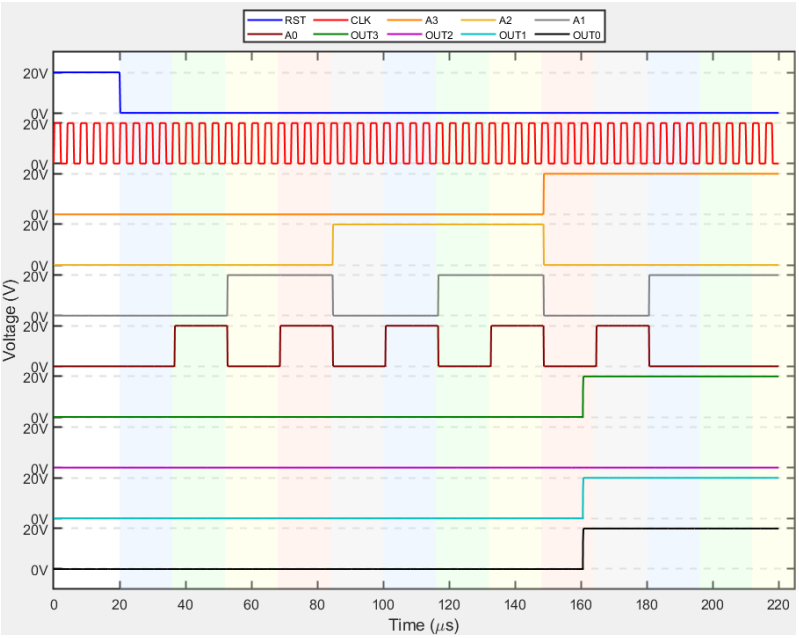


Figure 5.19: Top-level transient simulation of the output register and external ROM addressing (A3-A0) during the same instruction sequence at 500 °C.

Addressing Signal Integrity at the Accumulator Input

During initial top-level transient simulations, a signal integrity issue was identified at the input to the accumulator (ACC). The logic-high voltage level failed to reach the full 20 V supply level and was instead clamped at approximately 15 V. This degradation was observed at the output of the multiplexer responsible for selecting between the ALU result and the grounded '0' input used during the CLR instruction, as illustrated in the simulation waveform prior to correction (Figure 5.20).

The root cause was traced to the use of a transmission-gate (TG) based multiplexer implementation at this critical node as illustrated in Fig. 5.5b. When the grounded input was selected, the PMOS transistor within the TG structure experienced incorrect body biasing, where its source was tied to VSS while its body remained connected to VDD. This resulted in forward biasing of the source-body junction, creating an unintended leakage path that effectively clamped the output voltage.

To rectify this issue, the TG-based MUX was replaced with a conventional CMOS logic multiplexer, whose structure inherently avoids such body biasing problems under these connection conditions. Subsequent simulations confirmed the effectiveness of this solution. As shown in Figure 5.21, the accumulator input signal now exhibits a clean and full 0–20 V voltage swing, ensuring reliable data capture by the accumulator register and validating the correct operation of the modified ALU-ACC signal path.

5.5.2 Processor Performance Analysis

This section provides a quantitative evaluation of the designed 4-bit SiC processor under different temperature conditions, focusing on its timing characteristics and maximum operating frequency in high-temperature environments. By extracting the propagation delay of the critical path and applying a consistent timing margin, the minimum clock period $T_{clk,min}$ and corresponding maximum clock frequency F_{max} can be calculated. These results verify the functional feasibility and design robustness of the processor when operating under extreme thermal conditions.

The analysis aims to determine the processor's temperature-dependent performance limits and assess its timing robustness up to $500°C$. Circuit-level simulations were performed to extract the propagation delay of the longest internal path, expressed as $(T_{cq} + T_{logic})$, and the total clock period was computed as:

$$T_{clk} = (T_{cq} + T_{logic} + T_{setup} + T_{skew}) \times (1 + T_{margin}) \tag{5.6}$$

In Equation 5.6, T_{setup} is conservatively approximated as $\approx 0.2 \times T_{cq(max)}$. Based on Section 5.3.1, the maximum T_{cq} is 0.249 µs at room temperature, yielding a T_{setup} budget of 50 ns. T_{skew} is also conservatively budgeted at 50 ns, and T_{margin} is set to 15%. The selected critical path corresponds to the arithmetic logic execution path:

$$Input \rightarrow ALU \rightarrow ACC$$

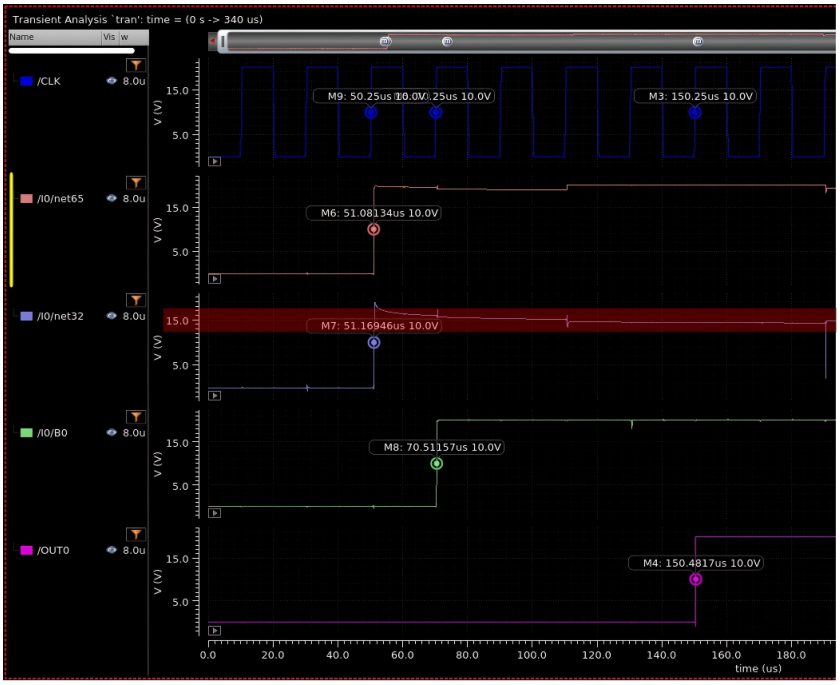


Figure 5.20: Simulated waveform showing voltage degradation at the accumulator input before MUX replacement. The accumulator input signal (net32, highlighted red) is clamped at approximately 15 V for logic high due to incorrect TG-MUX biasing.

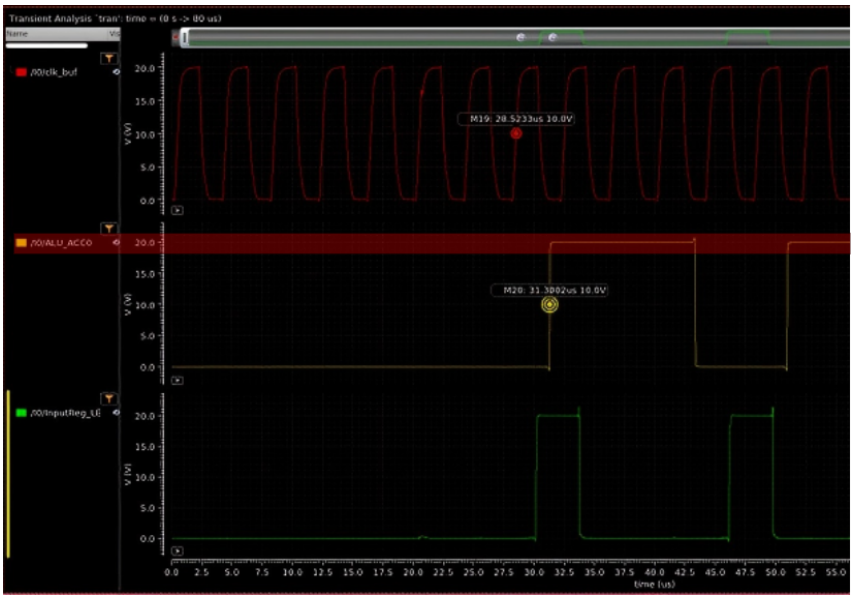


Figure 5.21: Simulated waveform after replacing the TG-MUX with a CMOS logic MUX. The accumulator input signal (ALU_ACC0, highlighted red) now exhibits a full 0–20 V voltage swing.

The total delay for this path is calculated as shown in Figure 5.22. It is the time difference between the triggering CLK signal (red) rising to 50% VDD (10V) at 28.391 µs, and the final signal at the accumulator's input, ALU_ACC[0] (blue), flipping and reaching 50% VDD at 29.449 µs. The ACC_WE signal is the accumulator's write enable, indicating the execution phase. From this, the total sum of T_{logic} and T_{cq} at 500°C is calculated to be 1.058 µs. This path includes the propagation from the input register through the ALU to the accumulator update. It contains multiple cascaded logic levels and represents the slowest internal signal transition in the processor core, thereby defining the minimum feasible clock period.

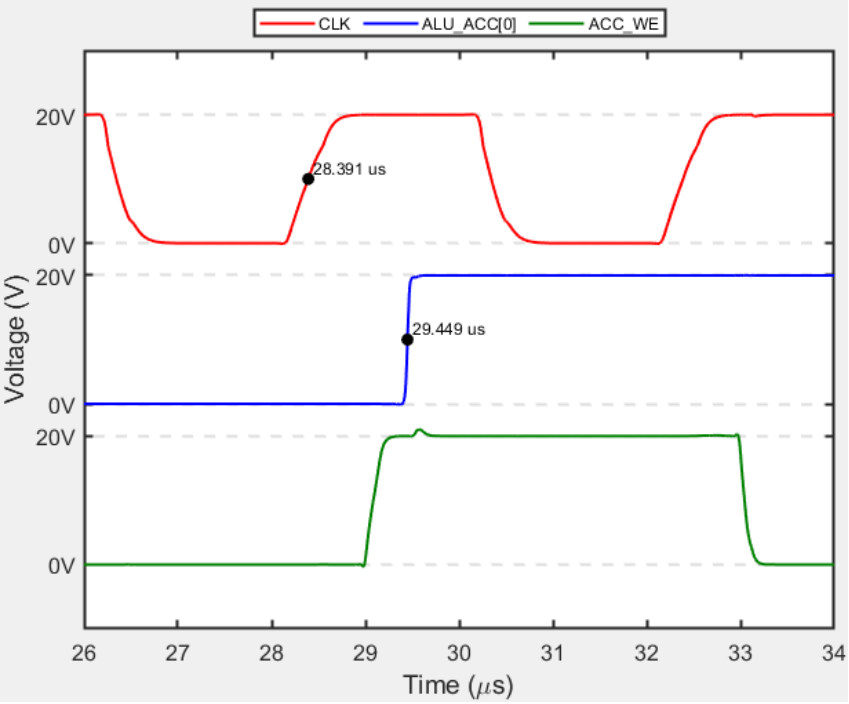


Figure 5.22: Transient simulation of the arithmetic logic execution path delay at 500°C.

In actual operation, another potential limiting path is the instruction fetch path: $PC \rightarrow ROM \rightarrow IR$, which involves off-chip memory access and could contribute additional latency. However, as the I/O interfaces and external memory were modeled as ideal components (zero access delay) in the simulation, such external effects are not reflected in the timing results. Therefore, the internal ALU execution path is used as the representative basis for determining $T_{clk,min}$ and F_{max}, and the result represents the theoretical maximum frequency of the processor core logic in this analysis.

Two SiC CMOS device models were adopted in this work: the BSIM4SiC300 model from the Fraunhofer IISB SiC CMOS PDK v1.2 with the simulation temperature limit of 300°C, and the SiC_CMOS_500dC model from the updated PDK

v1.3 with the simulation temperature limit of 500°C. Both models are based on the same fabrication process.

During the top-level processor simulations, which comprise over 2000 transistors, convergence issues were observed for both models at different temperature ranges. Specifically, the SiC_CMOS_500dC model exhibited convergence errors at low temperatures below 150°C, while the BSIM4SiC300 model failed to converge at temperatures above 250°C. To ensure simulation stability across the full temperature spectrum, the BSIM4SiC300 model was employed for the low-temperature range from 27°C to 150°C, and the SiC_CMOS_500dC model was used for above 150°C. As summarized in Tables 5.3 and 5.4, the results show that at 150°C and 200°C, the SiC_CMOS_500dC model produces shorter propagation delay than the BSIM4SiC300 model, indicating improved device speed and drive capability in the updated version of the model at elevated temperatures. The simulation data differences and convergence issues from both models will be fed back to the foundry for future model accuracy improvement and verification.

Table 5.3: Propagation delay of ALU path ($T_{cq} + T_{logic}$) and estimated T_{clk} and F_{max} using the BSIM4SiC300 model.

Temperature (°C)	$T_{cq} + T_{logic}$ (µs)	T_{clk} (µs)	F_{max} (kHz)
27	2.780	3.312	301.9
100	2.020	2.438	410.2
150	1.633	1.993	501.8
200	1.390	1.713	583.6

Table 5.4: Propagation delay of ALU path ($T_{cq} + T_{logic}$) and estimated T_{clk} and F_{max} using the SiC_CMOS_500dC model.

Temperature (°C)	$T_{cq} + T_{logic}$ (µs)	T_{clk} (µs)	F_{max} (kHz)
150	1.209	1.505	664.3
200	1.100	1.380	724.6
250	1.046	1.318	758.8
300	0.972	1.233	811.2
350	0.965	1.225	816.5
400	0.963	1.222	818.0
450	1.028	1.297	770.9
500	1.058	1.332	750.9

The calculated T_{clk} and F_{max} results across temperatures are summarized in Tables 5.3 and 5.4 and illustrated in Figure 5.21. These results show F_{max} increasing from 301.9 kHz at 27°C and continuing to rise with the SiC_CMOS_500dC model to a peak of 818 kHz at 400°C. At 500°C, F_{max} remains high at 750.9 kHz. Notably, the updated SiC_CMOS_500dC model predicts faster device speed than the older BSIM4SiC300 model (e.g., 664.3 kHz vs. 501.8 kHz at 150°C). Despite this absolute

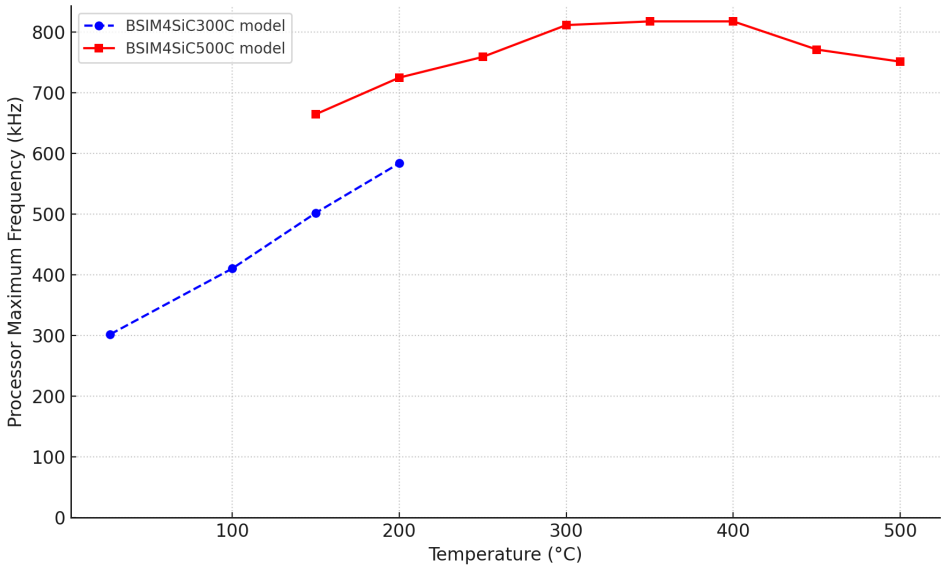


Figure 5.23: Operation of the RISC processor from the room temperature to 500°C using different models.

value difference from the model update, both models show a consistent trend of performance improving with temperature in their respective valid ranges, aligning with the decreasing T_{CQ} trend of the D flip-flop shown in Figure 5.11b.

From a physical standpoint, the variation in processor timing performance with temperature arises from the same competing mechanisms responsible for the T_{CQ} trend discussed in Section 5.3.1. At lower to moderate temperatures from 27°C to 400°C, factors such as the decrease in threshold voltage, increase in carrier concentration due to more complete dopant ionization, and the significant reduction of p-type contact resistance collectively improve SiC transistor drive strength [15, 30], thereby reducing delay. Conversely, at temperatures exceeding 400°C, the reduction in carrier mobility due to increased phonon scattering becomes the dominant factor, resulting in a slight increase in propagation delay.

Compared to prior work by Holmes et al. on the HiTSIC CMOS microcontroller ($F_{max} \approx 800$ kHz at 470°C) [6], the proposed processor demonstrates comparable high-temperature performance. The designed SiC processor core achieves a peak internal operating frequency of 818 kHz at 400°C and maintains a stable maximum frequency of 750.9 kHz under a 20 V supply and at 500°C. These results confirm the robustness and timing reliability of the implemented core logic under extreme temperature, establishing a solid basis for subsequent tape-out validation and experimental verification.

5.6 Summary of Circuit Design and Simulation

This chapter presented the circuit-level design and simulation of the 4-bit SiC RISC processor in the Fraunhofer IISB 2 µm technology. Based on inverter analysis across temperatures, optimal transistor sizing (L=6 µm, Wn=8 µm, Wp/Wn=3:1) was determined for the standard cell library. Core modules, such as logic gates, the ALU, registers, and the control unit, were implemented using these cells and verified via transient simulation for robust functionality up to 500°C. The integrated top-level processor, comprising 2092 transistors, demonstrated correct functional operation and timing coordination in simulations across the target temperature range from 27°C up to 500°C. Based on critical path delay analysis, the maximum stable operating frequency of the processor core was estimated to be approximately 750 kHz at 500°C. This verified circuit design provides a solid foundation for the physical layout implementation detailed in Chapter 6.

Chapter 6

Layout Design and Verification

Following the successful circuit-level design and simulation presented in Chapter 5, this chapter addresses the physical implementation phase: the layout design and verification of the 4-bit SiC RISC processor. The primary goal is to translate the validated schematics into a manufacturable layout adhering to the specific design rules and constraints of the Fraunhofer IISB 2 μm 4H-SiC CMOS process. This involves establishing a robust digital standard cell library, meticulously implementing the layouts for critical functional blocks like the ALU, registers, and control unit, and finally integrating these modules into a complete top-level floorplan within the target 5 $mm \times 5$ mm die area. The chapter details the standard cell design methodology, routing strategies, power distribution network, and concludes with the final verified layout of the processor core.

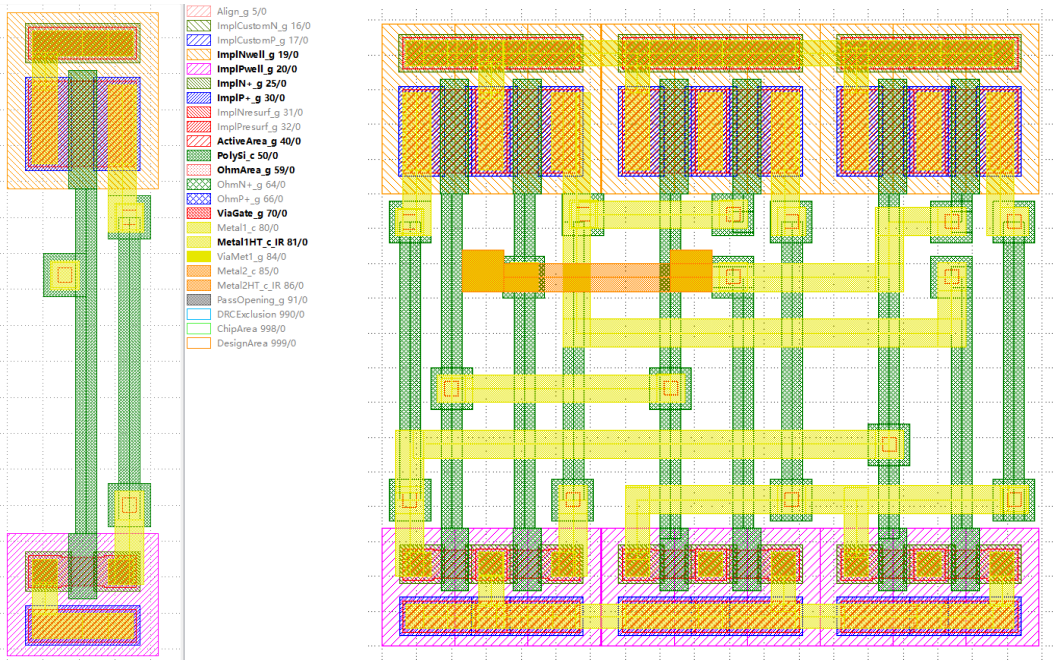
6.1 Design of Digital Standard Cell

The layout design of digital standard cells follows a regular and grid-based structure to ensure consistent geometry, routing hierarchy, and power distribution across all logic elements. This uniformity facilitates automated placement and interconnection in the top-level floorplan. Among all logic primitives, the CMOS inverter is selected as the reference cell for defining the layout style, routing conventions, and height standard for the entire standard-cell library.

Each inverter consists of one PMOS and one NMOS transistor arranged in a vertically mirrored configuration. The two transistors share a common polysilicon gate and are locally interconnected to form the pull-up and pull-down networks. As illustrated in Fig. 6.1a, the inverter layout demonstrates the physical arrangement of the transistors.

A spacing of 96 µm is reserved between the P-well and N-well regions to accommodate the routing channel for signal interconnection and power access. This design choice originates from a previous tape-out observation: when the Metal 2 (M2) layer was routed directly over the active region, it occasionally caused short-circuit issues with the underlying Metal 1 (M1) layer connected to the transistor

source/drain terminals. Following this finding, Fraunhofer IISB revised the DRC rules of the 4H-SiC CMOS process to prevent M2 overlap above transistor pcell areas. Consequently, all standard-cell routing must now be completed within the inter-device region. To ensure sufficient margin for higher-level cross-cell connections, several layout iterations were performed, and a total spacing of six routing tracks between the PMOS and NMOS devices was finally adopted. The number and height of routing tracks remain consistent across all standard cells, ensuring seamless abutment and alignment during cell placement.



(a) (b)

Figure 6.1: Standard-cell layouts implemented in the IISB 4H-SiC CMOS process: (a) Inverter (INV) standard cell layout showing the 6-track routing channel and layer information; (b) XOR standard cell layout, where Metal 2 is employed to relieve M1 congestion.

The core design philosophy of the standard cell is to achieve a compact layout. Under the constraints of design rules and electrical reliability, the transistor placement and interconnects are optimized to minimize cell area, thereby reducing the overall chip footprint and improving logic density. The total cell height is unified to 179 µm, enabling straightforward row-based arrangement and cell reuse during higher-level layout implementation. When arranged in standard-cell rows, the substrate and well regions can be continuously extended and shared among adjacent cells, achieving uniform body biasing and power distribution while simplifying the power network design, as illustrated in Fig. 6.2.

Within each standard cell, M1 is primarily used for signal routing, source/drain

connections, and local interconnects. When the M1 routing resources become limited, M2 is employed for auxiliary routing at the top of the cell and for cross-cell connections, thereby alleviating M1 congestion and enabling inter-cell signal extension. For example, in the XOR gate layout as shown in Fig. 6.1b, M2 is used to route overlapping signals and intermediate nodes that cannot be efficiently realized within a single metal layer, maintaining layout clarity and connectivity. Multiple via arrays are used to ensure robust vertical interconnects between M1 and M2. The layout strictly complies with the design rules regarding spacing, enclosure, and well isolation, ensuring reliable operation and electrical consistency at high temperatures.

By establishing unified cell height, routing track count, and power distribution guidelines during standard-cell design, this methodology provides a standardized foundation for the hierarchical implementation of more complex logic modules such as latches, registers, and multiplexers. It also significantly enhances layout reusability and scalability across the digital design hierarchy.

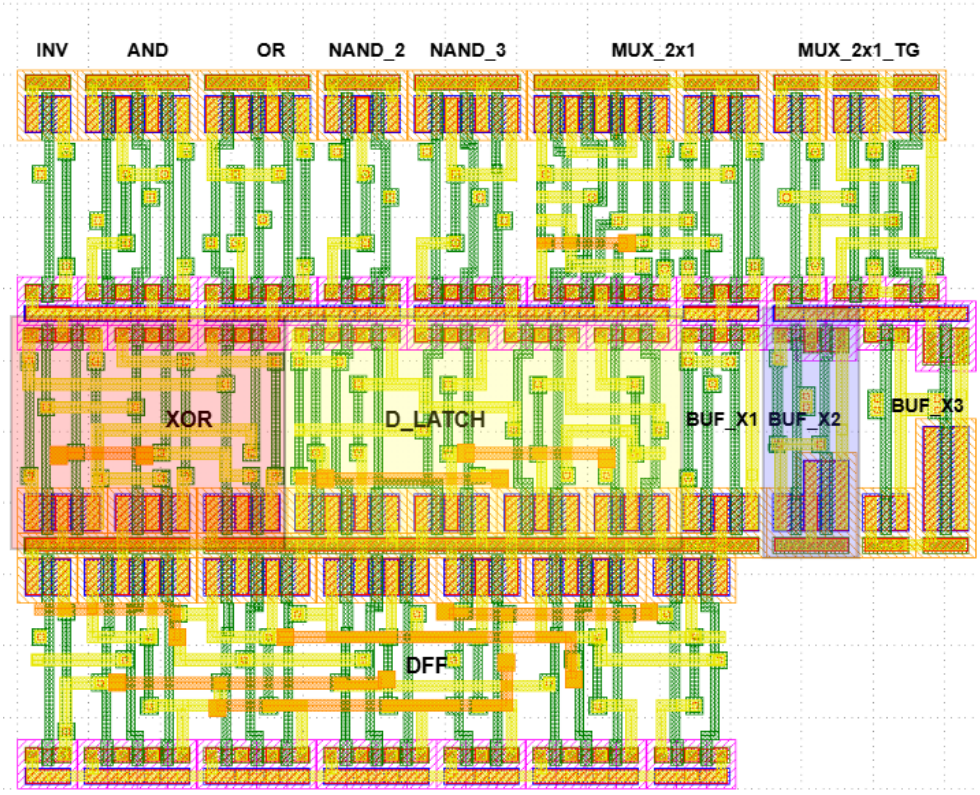


Figure 6.2: Complete library of digital standard cells implemented in the processor design. All cells have been verified through Layout Versus Schematic (LVS) and Design Rule Check (DRC) checks to ensure inter-cell connectivity consistency and design-rule compliance. (Note: MUX_2x1_TG represents a transmission-gate-based MUX, and BUF_1X denotes a single-stage buffer)

6.2 Critical Module Layouts

This section presents the layout implementation of representative functional modules in the 4-bit SiC processor. Each module was implemented using the standard-cell library introduced in Section 6.1 and verified through full DRC and LVS checks. Within each module, M1 is primarily used for horizontal intra-cell routing, while M2 is used for vertical and cross-row interconnections. To simplify internal wiring and power distribution, most modules adopt a vertical layout orientation, in which two adjacent standard-cell rows are mirrored to share the same power (VDD) and ground (VSS) rails. This approach enables compact area utilization and facilitates shared routing of key control and global signals, such as the output register enable signal `OutputReg_WE`, the global clock signal (`CLK`), and the reset signal (`RST`).

These modules were selected to represent the most layout-critical components in terms of area utilization, routing complexity:

- **4-bit Output Register** — a sequential storage block using vertically stacked D flip flop cells with shared control and power rails (Fig. 6.3).
- **8-bit Program Counter (PC)** — composed of two 4-bit register blocks with integrated incrementer logic (Fig. 6.4b).
- **4-bit Arithmetic Logic Unit (ALU)** — combining arithmetic and logic operations with dense M2 routing for carry and control signals (Fig. 6.4a).
- **Control Unit (CU)** — a logic-dominant FSM block coordinating instruction sequencing with control buses (Fig. 6.5).

Overall, the modular layouts demonstrate hierarchical consistency and efficient routing practices derived from the designed standard-cell library. By maintaining uniform cell height, shared bias regions, and orthogonal metal routing, all blocks achieve high layout regularity and compact integration suitable for 4H-SiC CMOS implementation.

6.3 Top-Level Floorplan and Layout Implementation

This section presents the top-level floorplan and layout implementation of the 4-bit SiC processor based on the standard-cell library described in Section 6.1. The chip was designed using the Fraunhofer IISB 4H-SiC CMOS process, with a total die area constrained within 5 mm $\times$ 5 mm. A hierarchical design methodology was adopted, in which standard cells are organized into functional modules and further integrated into the system-level layout. This approach ensures structural regularity, design scalability, and efficient verification.

According to the pad-frame specification [26], the processor adopts a 40-pad layout scheme, among which 24 pads are actively used for signal and power

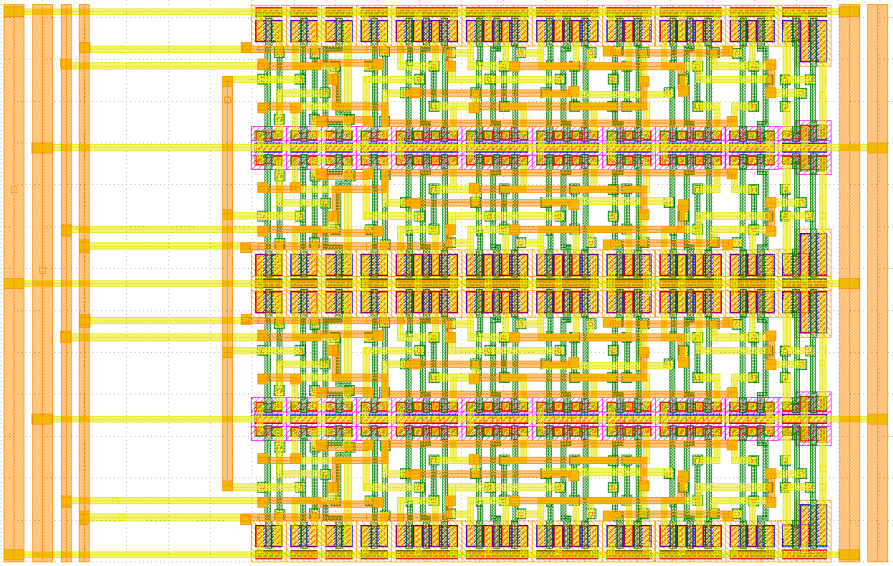


Figure 6.3: Layout of the 4-bit Output Register module.

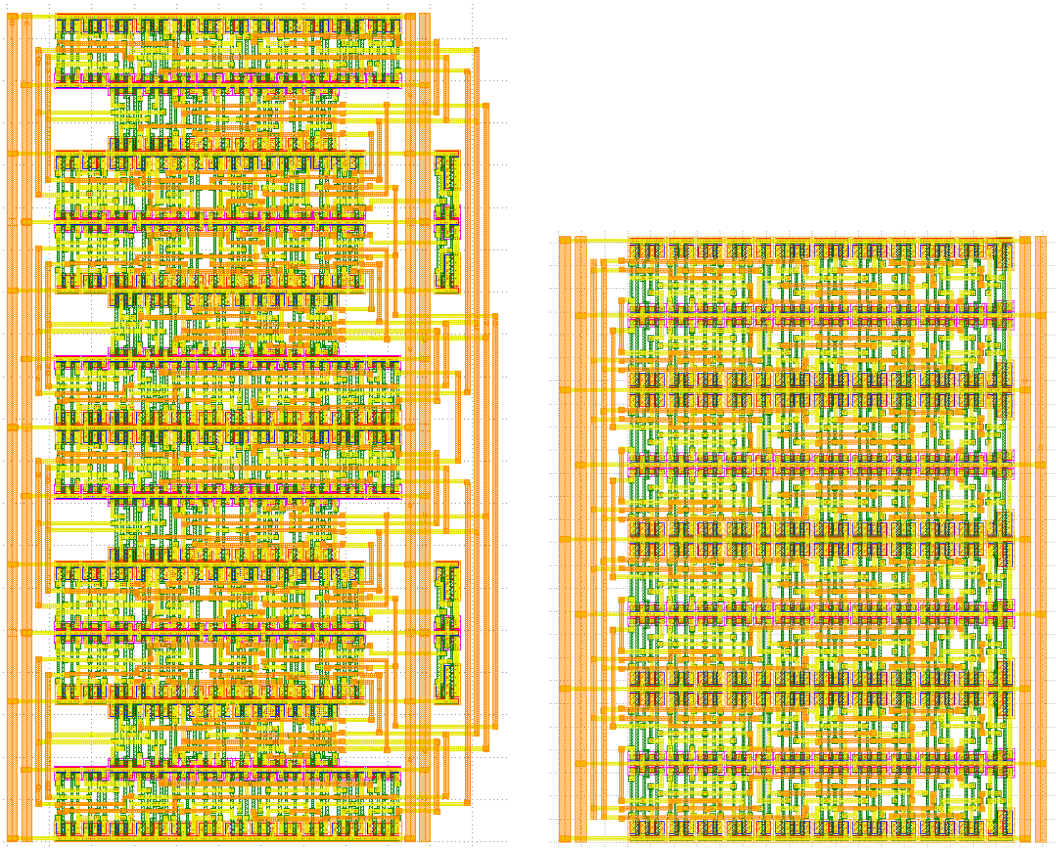
connections. Each pad measures 200 μm $\times$ 200 μm and is implemented using widened M2 layers with passivation openings to enable reliable wire bonding. A certain area is reserved at the bottom of the chip, where several pads remain unused to facilitate potential extensions for future test structures—such as registers with different transistor lengths or internal signal probing (e.g., extracting ALU outputs) for module-level testing.

As shown in Fig. 6.6, the processor modules are arranged according to their functional grouping to minimize critical path lengths and simplify routing:

- **Left**: Input Latch and Instruction Register (IR)
- **Center**: Arithmetic Logic Unit (ALU)
- **Right**: Program Counter (PC), Accumulator (ACC), and Output Register
- **Top**: Control Unit (CU)

The power distribution network combines both vertical and horizontal metal tracks. Continuous M2 rails form a closed loop surrounding the entire processor core, providing complete power enclosure. This ring-shaped configuration enhances power integrity and signal isolation, effectively reducing noise coupling between modules and minimizing voltage drop fluctuations, which is particularly beneficial for stable operation at elevated temperatures.

For signal routing, M1 is primarily used for horizontal interconnections, while M2 handles vertical and cross-module routing. This orthogonal routing scheme reduces parasitic coupling and simplifies layout visualization and debugging.



(a) 4-bit ALU (b) 8-bit Program Counter

Figure 6.4: Layout of the main computational blocks: (a) 4-bit ALU and (b) 8-bit Program Counter (PC).

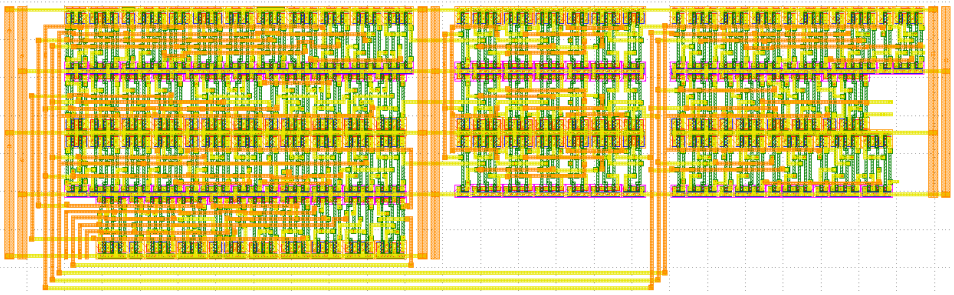


Figure 6.5: Layout of the Control Unit implementing the processor FSM.

The global clock (CLK) and reset (RST) signals are introduced vertically from the left edge of the chip and first buffered before entering the core logic. The inclusion of buffer stages helps to reduce clock fan-out loading, equalize propagation delays, and maintain signal integrity under high-temperature or large-capacitance conditions. Unlike ordinary signal lines, the clock routing employs a widened metal width to reduce resistance drop, mitigate electromigration risk, and improve high-frequency transmission robustness and noise immunity.

The final top-level layout, shown in Fig. 6.6, has successfully passed both DRC and LVS verification without any open or short errors. The resulting design achieves compact area utilization, clear signal hierarchy, and well-organized module connectivity, while leaving sufficient space for future test and verification structures.

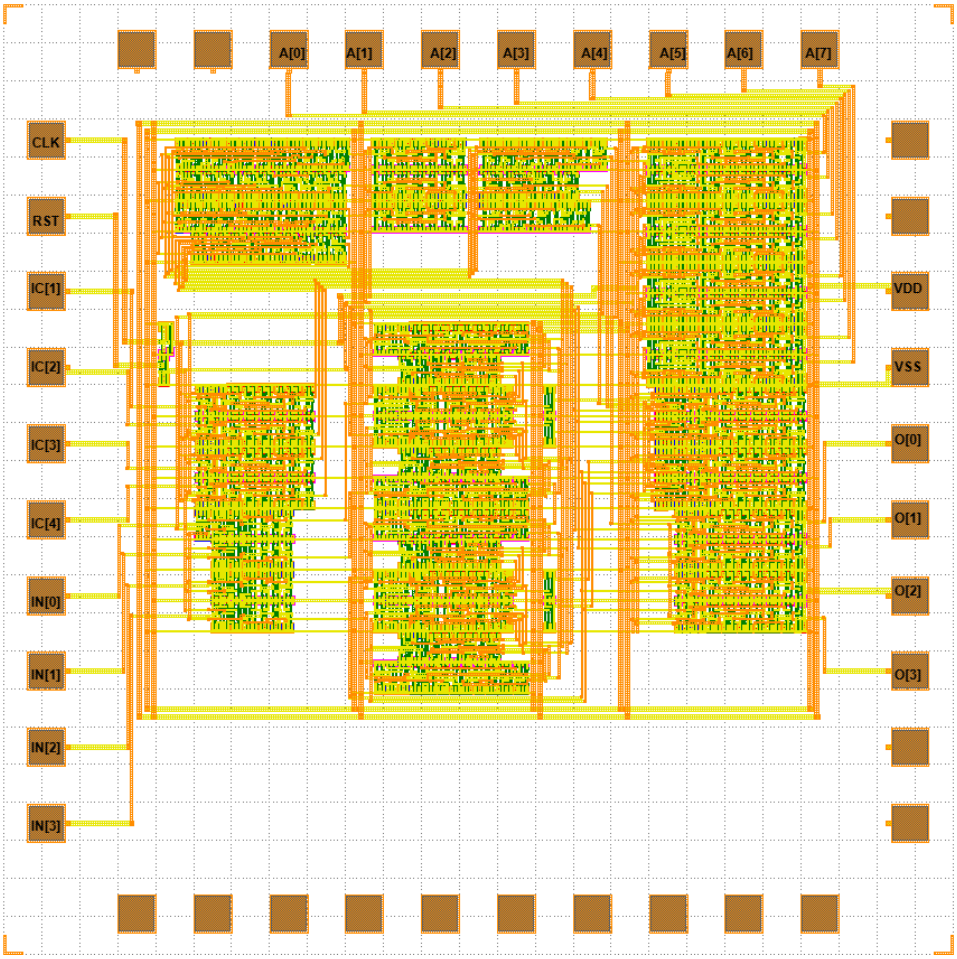


Figure 6.6: Top-level 4-bit RISC processor layout with labeled pad connections. Corresponding to the I/O interface shown in Fig. 3.2, IC[x] represents the instruction bus, IN[x] the data input bus, A[x] the address bus, and O[x] the data output bus. A 40-pad frame is used with 24 active I/O pads.

6.4 Summary of Layout and Verification

This chapter reviewed the successful transformation of the 4-bit SiC RISC processor's circuit schematics into a physical layout. A key achievement was establishing a standard cell library with unified height and routing specifications compliant with the Fraunhofer IISB 2 µm process rules, enabling hierarchical and automated layout. Based on this library, layouts for all core modules, including the ALU, registers, and control unit, were completed. Finally, these modules were successfully integrated within the 5x5 mm^2 die area; the top-level layout incorporated robustness strategies such as a ring power rail and buffered global signals. Crucially, the final layout passed DRC and LVS verification, confirming the design's manufacturability and yielding the GDSII file ready for fabrication.

Chapter 7

Conclusions and Future Work

7.1 Summary of Contributions

With the rapid advancements in fields like aerospace, nuclear energy, and automotive technology, there is a growing demand for electronic systems capable of reliable operation under extreme environmental conditions. This thesis aimed to address this challenge, with the primary objective being the design, simulation, and verification of a 4-bit RISC processor based on the Fraunhofer IISB 2 μm 4H-SiC CMOS process. This processor was intended as a feasibility validation platform for digital system-level integration using this emerging technology, targeting stable operation at extreme temperatures up to 500 °C.

To accommodate the unique constraints of the SiC process, particularly large size of transistor and strict transistor budget, this project adopted an "ultra-minimalist" architectural design philosophy. Key design decisions included placing the program memory off-chip to conserve core area, employing a highly streamlined ISA with only 11 instructions, and leveraging hardware-software co-design to minimize control logic complexity. Given the limitations in automated digital design flow support within the target SiC process PDK, a practical design methodology was established: initial RTL synthesis using the open-source tool Yosys generated gate-level netlists and schematics, which then guided manual circuit-level implementation, optimization, and simulation within the Cadence Virtuoso environment. A cornerstone of this effort was establishing the fundamental transistor sizing parameters ($L = 6\ \mu m$, $W_n = 8\ \mu m$, and a W_p/W_n ratio of 3:1), carefully determined through simulation to balance performance, area, and high-temperature reliability. Based on these parameters, a custom SiC standard cell library was developed and characterized; this library included essential logic units like inverters, NAND gates, NOR gates, transmission-gate multiplexers, D-type flip-flops, and D latches, all optimized for operation up to 500 °C and area efficiency within the 2 μm process.

The primary achievements and verification efforts of this project are summarized as follows: First, the functional completeness of the processor design was

demonstrated through comprehensive simulations spanning from the RTL behavioral level down to the transistor circuit level, confirming the correct execution of all defined arithmetic, logic, and data transfer instructions. Second, the capability for high-temperature operation was verified; transient simulation results showed that all individual modules as well as the complete top-level processor operate stably at the target extreme temperature of 500 °C, exhibiting clean signal transitions and correct timing. Further simulations confirmed reliable operation of these circuit components across the entire temperature range from 27 °C to 500 °C. Third, regarding performance verification, detailed timing analysis based on critical path simulations established the processor's F_{max}. This analysis indicated reliable logic performance peaking at approximately 817 kHz at 400°C and maintaining up to 750 kHz at the target temperature of 500°C. The characteristic non-monotonic temperature dependence of delays in SiC devices was also captured during characterization. Finally, on the physical design front, a complete custom layout of the processor was successfully implemented using the developed standard cell library, passing both DRC and LVS verification checks. The final layout comprised 2092 transistors, well below the initial budget estimate of approx. 3000, confirming the feasibility of manufacturing the design within the 5 $mm \times 5$ mm die area.

In conclusion, this thesis successfully delivered a complete and verified design for a 4-bit SiC RISC processor, covering the entire flow from architectural conception to final layout. Simulations confirmed its robust functionality up to 500 °C and validated its potential to operate reliably at frequencies reaching 750 kHz under these extreme conditions. This work validates the capability of the Fraunhofer IISB 4H-SiC CMOS process to support complex digital system-level integrated circuits beyond basic logic gates. It thereby establishes a foundation for the future development of more sophisticated SiC-based digital systems tailored for demanding harsh-environment applications.

7.2 Future Work

Although this thesis successfully demonstrates the feasibility of designing a 4-bit RISC processor in SiC CMOS technology and comprehensively verifies its functionality and performance at high temperatures through simulation, several avenues remain for further exploration to advance this technology towards practical applications.

Firstly, the crucial next step is fabrication and experimental validation. Taping out the completed processor layout through the MPW service offered by Fraunhofer IISB is essential. Once the actual chips are obtained, comprehensive experimental testing is necessary. This testing should particularly focus on functional verification and performance characterization under high-temperature conditions, ranging from room temperature up to 500 °C and potentially beyond. This includes measuring the actual F_{max}, power consumption, correctness of instruction execution, and long-term operational reliability. The acquired experimental data

will not only definitively confirm the design's effectiveness but also allow for comparison with the simulation results presented in Chapter 5. This comparison is invaluable for assessing the accuracy of the existing SiC PDK device models across a wide temperature range and providing crucial feedback for subsequent designs.

Secondly, there is room for improvement in the design flow and process optimization. The current flow, relying on a combination of Yosys synthesis and manual circuit implementation, while functional, is inefficient and difficult to scale to more complex designs. Future work should focus on developing a complete automated digital design flow. Specifically, establishing a SiC standard cell library that includes accurate timing information (.lib files) is critical. This would enable the use of standard P&R tools, facilitating fully automated synthesis from RTL code to GDSII layout, thereby significantly enhancing design efficiency and reliability. Furthermore, the experimental data obtained from chip testing should be fed back to the foundry. This feedback can help refine the SiC device models, particularly improving their predictive accuracy and addressing potential timing closure issues observed under <150 °C for system level simualtion, thus supporting more accurate performance prediction and timing convergence in future designs.

Finally, architecture enhancements and system-level integration represent important directions for increasing the processor's practicality. To make the processor more autonomous and powerful, integrating on-chip memory could be considered. For example, incorporating on-chip program ROM and data RAM, possibly based on the 6T SRAM cell designs previously validated, would transform it into a true single-chip microcontroller (MCU). The instruction set could also be expanded to include control flow instructions such as branches, jumps, and subroutine calls, although this would necessitate more complex designs for the control unit and program counter logic. More significantly, to handle the sensor data precision and algorithmic complexity required by real-world automotive and aerospace applications, expanding the 4-bit datapath to a 16-bit architecture represents a critical and logical progression. The ultimate goal could be to integrate this SiC processor core with other SiC functional blocks already developed on the same process platform such as ADCs or UV sensors, creating a complete SiC System-on-Chip (SoC) tailored for specific high-temperature sensing, control, or computation applications. Realizing these advanced architectures and levels of integration, however, will undoubtedly require concurrent advancements in the underlying SiC CMOS technology itself, particularly in achieving higher integration densities through smaller process nodes. Successfully achieving these technological and architectural milestones would fully leverage the unique advantages of SiC technology in extreme environments.

Bibliography

[1] M. Bhatnagar and B. J. Baliga. Comparison of 6h-sic, 3c-sic, and si for power devices. *IEEE Transactions on Electron Devices*, 40(3):645–655, 1993. doi: 10.1109/16.199375.

[2] Electronics-Tutorial.net. 2 : 1 mux using transmission gate, 2023. URL https://www.electronics-tutorial.net/Digital-CMOS-Design/Pass-Transistor-Logic/2-1-MUX-using-transmission-gate/. Accessed: 2025-10-19.

[3] Europractice IC Service. Fraunhofer iisb 4h-sic cmos process through europractice mpw service, 2024. URL https://europractice-ic.com/mpw-foundry-services/fraunhofer-iisb-sic-cmos/. Accessed: 2025-10-14.

[4] R. Ghandi, C.-P. Chen, L. Yin, R. Saia, T. Johnson, P. Sandvik, K. Fang, and R. W. Johnson. Reliability of sic digital telemetry circuits on aln substrate. *Journal of Microelectronics and Electronic Packaging*, 11(1):25–29, 2014. doi: 10.4071/imaps.397.

[5] Olivier Girard. Openmsp430 open-source 16-bit microcontroller core. https://opencores.org/projects/openmsp430.

[6] J. Holmes, A. M. Francis, H. A. Mantooth, and J. Di. Extended high-temperature operation of silicon carbide cmos circuits for venus surface application. In *Proceedings of the International Conference on High Temperature Electronics (HiTEC)*, pages 143–154, 2016.

[7] T. Kimoto. Material properties and physics of sic for high-power electronics. *Japanese Journal of Applied Physics*, 54(4S):040103, 2015. doi: 10.7567/JJAP.54.040103.

[8] T. Kimoto and J. A. Cooper. *Fundamentals of Silicon Carbide Technology: Growth, Characterization, Devices, and Applications*. John Wiley & Sons, 2014.

[9] kryptoknight. Preference of nand & nor gates, June 2013. URL https://electronics.stackexchange.com/questions/72806/preference-of-nand-nor-gates. Accessed: 2025-10-24.

[10] N. Kuhns, L. Caley, A. Rahman, S. Ahmed, H. A. Mantooth, A. M. Francis, J. Holmes, and J. Di. Complex high-temperature cmos silicon carbide digital circuit designs. *IEEE Transactions on Device and Materials Reliability*, 16(2):105–114, 2016. doi: 10.1109/TDMR.2016.2530664.

[11] M. E. Levinshtein, S. L. Rumyantsev, and M. Shur, editors. *Properties of Advanced Semiconductor Materials: GaN, AlN, InN, BN, SiC, SiGe*. John Wiley & Sons, 2001.

[12] C. Y. Liang, G. T. Dunne, K. S. Matocha, R. E. Green, D. J. Lichtenwalner, and B. J. Baliga. Reliability issues of sic mosfets: A technology for high-temperature environments. *IEEE Transactions on Device and Materials Reliability*, 10(4):418–426, 2010. doi: 10.1109/TDMR.2010.2081981.

[13] F. Martinez Santa, F. H. Martinez Sarmiento, and H. Montiel Ariza. Minimalist 4-bit processor focused on processors theory teaching. *Indian Journal of Science and Technology*, 10(14):1–9, 2017. doi: 10.17485/ijst/2017/v10i14/111328.

[14] Alexander May, Michael Schraml, and Mathias Rommel. IISB SiC CMOS PDK Package, Version 1.3. Technical report, Fraunhofer IISB, Erlangen, Germany, 2024. URL https://www.iisb.fraunhofer.de. EUROPRACTICE CMOS Run 3, Prototype High Temperature 4H-SiC CMOS Technology.

[15] J. Mo, Y. Niu, A. May, M. Rommel, C. Rossi, J. Romijn, G. Zhang, and S. Vollebregt. An analog to digital converter in a sic cmos technology for high-temperature applications. *Applied Physics Letters*, 124(15):152105, 2024. doi: 10.1063/5.0195013.

[16] S. Moslehpour, C. Puliroju, and A. Abu-aisheh. Design of risc processor using vhdl and cadence. Technical report, University of Hartford, West Hartford, CT, USA.

[17] Y. Niu, J. Mo, A. May, M. Romme, C. Rossi, J. Romijn, G. Zhang, and S. Vollebregt. Design and characterization of a data converter in a sic cmos technology for harsh-environment sensing applications. *IEEE Sensors Journal*, 23(9): 10325061, 2023. doi: 10.1109/SENSORS56945.2023.10325061.

[18] Tutorials Point. Binary adder-subtractor, 2024. URL https://www.tutorialspoint.com/digital-electronics/digital-electronics-binary-adder-subtractor.htm. Accessed: 2025-10-19.

[19] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolić. *Digital Integrated Circuits*. Prentice Hall Electronics and VLSI Series. Pearson Education, Upper Saddle River, NJ, USA, 2nd edition, 2003.

[20] A. Rahman, L. Caley, S. Roy, N. Kuhns, H. A. Mantooth, J. Di, A. M. Francis, and J. Holmes. High-temperature data converters in silicon carbide cmos. *IEEE Transactions on Electron Devices*, 64(3):1426–1432, 2017. doi: 10.1109/TED.2017.2651004.

[21] J. Romijn. *Integrated Silicon Carbide Sun Position Sensor System-on-Chip for Space Applications*. Phd dissertation, Delft University of Technology, Delft, The Netherlands, 2022. URL https://doi.org/10.4233/uuid:0dbff3c1-752b-4211-ab1d-c8ec71ffed9d.

[22] J. Romijn, S. Vollebregt, L. M. Middelburg, B. El Mansouri, H. W. van Zeijl, A. May, T. Erlbacher, G. Zhang, and P. M. Sarro. Integrated digital and analog circuit blocks in a scalable silicon carbide cmos technology. *IEEE Transactions on Electron Devices*, 69(1):4–10, 2022. doi: 10.1109/TED.2021.3125279.

[23] S. H. Ryu, K. T. Kornegay, J. A. Cooper Jr., and M. R. Melloch. 6h-sic cmos digital ics operating on a 5 v power supply. In *Proceedings of the Device Research Conference (DRC)*, pages 38–39, 1997. doi: 10.1109/DRC.1997.618222.

[24] S. V. Saravanan, S. A. Sivakumar, S. Sasipriya, and R. Sowmya. Design of low power flip flop and implementation in a 4-bit counter. *International Journal of Printing, Packaging & Allied Sciences*, 4(3):2110–2121, December 2016. URL https://www.researchgate.net/publication/345178504_Design_of_Low_Power_Flip_Flop_and_Implementation_in_a_4-bit_Counter. ISSN 2320-4387.

[25] Stunts1990. Edge triggered d flip-flop with set and reset. Wikimedia Commons, October 2020. URL https://commons.wikimedia.org/wiki/File:Edge_triggered_D_flip_flop_with_set_and_reset.svg. Created and uploaded at Wikimedia Commons under CC BY-SA 4.0 licence.

[26] PAD Technology. *CSB04075 Model (1) Datasheet*. PAD Technology Co., Ltd., 2024. URL https://www.padtechnology.com/. Device model specification sheet.

[27] Adele Thompson and David Bertuch. Analog multiplexer, April 2023. URL https://c2s2.engineering.cornell.edu/blogposts/SP23/AnalogMultiplexer. Accessed: 2025-10-19.

[28] R. F. Thompson, E. P. Ramsay, D. A. Smith, R. A. Young, and A. E. Murphy. High temperature silicon carbide cmos integrated circuits. In *Proceedings of the International Conference on High Temperature Electronics Network (HiTEN)*, pages 1–5, 2011.

[29] C. Venkatesan, M. Suriya, T. Sulthana, and M. G. Sumithra. Design of a 16-bit harvard structure risc processor in cadence 45 nm technology. In *Proc. 5th Int. Conf. on Advanced Computing and Communication Systems (ICACCS)*, pages 173–178, Coimbatore, India, 2019. IEEE.

[30] H. Wang, P. Lai, M. Z. Islam, A. S. M. K. Hasan, A. Di Mauro, N.-E.-A. Anika, R. Russell, Z. Feng, K. Chen, A. Faruque, T. White, Z. Chen, and H. A. Mantooth. A review of silicon carbide cmos technology for harsh environments. *Materials Science in Semiconductor Processing*, 178:108422, 2024. doi: 10.1016/j.mssp.2024.108422.

[31] W. Xie, J. A. Cooper Jr., and M. R. Melloch. Nmos digital integrated circuits in 6h silicon carbide. In *Proceedings of the 52nd Annual Device Research Conference (DRC)*, pages 23–24, 1994. doi: 10.1109/DRC.1994.585407.

[32] M. Zaid and P. Mustajab. Design and application of risc processor. In *Proc. International Conference on Advanced Communication and Computing Technologies (IMPACT 2017)*, Aligarh Muslim University, India, 2017.

Appendix A

Source Code

A.1 Control Unit Behavioral Implementation

```
// ================================================================================
// Control Unit (Direct-Connect Version)
// Based on the "Direct Connection + Write Suppression" philosophy.
// This CU does NOT generate the ALU_Sel signal.
// ================================================================================
module ControlUnit(
    // ---  ---
    input wire         clk,
    input wire         rst,
    input wire [3:0]   I, // 4-bit Opcode

    // ---  () ---
    output reg         PC_INC,
    output reg         IR_WE,
    output reg         ACC_WE,
    output reg         InputReg_LE,
    output reg         OutputReg_WE,
    output reg         ACC_Mux_Select
    // ALU_Sel is no longer an output of this module
);

    // State Assignments (8 states -> 3 state bits)
    localparam FETCH         = 3'b000;
    localparam DECODE        = 3'b001;
    localparam EXECUTE_ALU   = 3'b010;
    localparam EXECUTE_OUT   = 3'b011;
    localparam EXECUTE_CLR   = 3'b100;
    localparam EXECUTE_OTHER = 3'b101;
    localparam END_CYCLE     = 3'b110;
```

```
localparam HALTED        = 3'b111;

// State Registers
reg [2:0] current_state, next_state;

// Next-State Logic (Combinational)
always @(*) begin
    case (current_state)
        FETCH:      next_state = DECODE;
        DECODE:
            if (I[3] == 1'b0) begin // ALU-type instruction
                next_state = EXECUTE_ALU;
            end else begin // Non-ALU-type instruction
                case (I)
                    4'b1000: next_state = EXECUTE_OUT;
                    4'b1001: next_state = EXECUTE_CLR;
                    4'b1010: next_state = HALTED;
                    default: next_state = EXECUTE_OTHER;
                endcase
            end
        EXECUTE_ALU, EXECUTE_OUT, EXECUTE_CLR, EXECUTE_OTHER: next_state =
            END_CYCLE;
        END_CYCLE: next_state = FETCH;
        HALTED:    next_state = HALTED;
        default:   next_state = FETCH;
    endcase
end

// State Register Update (Sequential)
always @(posedge clk or posedge rst) begin
    if (rst)
        current_state <= FETCH;
    else
        current_state <= next_state;
end

// Output Logic (Combinational)
always @(*) begin
    // Default all signals to 0 (inactive)
    PC_INC         = 1'b0;
    IR_WE          = 1'b0;
    ACC_WE         = 1'b0; // CRITICAL: Default is 0 to suppress writes
    InputReg_LE    = 1'b0;
    OutputReg_WE   = 1'b0;
```

```
ACC_Mux_Select = 1'b0;

// Activate signals based on the current state
case (current_state)
    FETCH:       begin
    IR_WE = 1'b1;

        PC_INC         = 1'b0;
        ACC_WE         = 1'b0;
        InputReg_LE    = 1'b0;
        OutputReg_WE   = 1'b0;
        ACC_Mux_Select = 1'b0;

    end
    EXECUTE_ALU: begin
        // For ALU ops, we MUST enable the write to ACC
        ACC_WE      = 1'b1;
        InputReg_LE = 1'b1;

        PC_INC         = 1'b0;
        IR_WE          = 1'b0;
        OutputReg_WE   = 1'b0;
        ACC_Mux_Select = 1'b0;
    end
    EXECUTE_OUT: begin
        OutputReg_WE = 1'b1;

        PC_INC         = 1'b0;
        IR_WE          = 1'b0;
        ACC_WE         = 1'b0;
        InputReg_LE    = 1'b0;
        ACC_Mux_Select = 1'b0;
    end
    EXECUTE_CLR: begin
        // For CLR, we also enable the write to ACC (via MUX)
        ACC_WE         = 1'b1;
        ACC_Mux_Select = 1'b1;

        PC_INC         = 1'b0;
        IR_WE          = 1'b0;
        InputReg_LE    = 1'b0;
        OutputReg_WE   = 1'b0;
    end
    END_CYCLE: begin
```

```
            PC_INC = 1'b1;

            IR_WE          = 1'b0;
            ACC_WE         = 1'b0;
            InputReg_LE    = 1'b0;
            OutputReg_WE   = 1'b0;
            ACC_Mux_Select = 1'b0;
        end
        HALTED: begin
            PC_INC         = 1'b0;
            IR_WE          = 1'b0;
            ACC_WE         = 1'b0;
            InputReg_LE    = 1'b0;
            OutputReg_WE   = 1'b0;
            ACC_Mux_Select = 1'b0;
        end
        // Note: For EXECUTE_OUT and EXECUTE_OTHER, ACC_WE remains 0,
        // successfully suppressing the unwanted ALU calculation.
    endcase
  end
endmodule
```

A.2 Control Unit Structural Implementation

```
// This module implements the complete Control Unit,
// using a Finite State Machine (FSM) structure. The Boolean expressions
// have been updated according to the revised design.
module ControlUnit (
    // Sequential Logic Inputs
    input clk,          // System clock
    input rst,          // Asynchronous reset signal

    // FSM Inputs
    input [3:0] I, // 4-bit instruction opcode (from IR)

    // FSM Outputs
    output PC_INC,
    output IR_WE,
    output ACC_WE,
    output InputReg_LE,
    output OutputReg_WE,
    output ACC_Mux_Select,
    output [2:0] current_state_out // Optional debug output
);
```

```
// ----------------------------------------------------
// Internal Signal Declarations
// ----------------------------------------------------

// State Register: 3-bit D-Flip-Flop to store the current state S[2:0]
reg [2:0] current_state;

// Output of the next-state logic (Input to the D-Flip-Flop)
wire [2:0] next_state;

// ----------------------------------------------------
// Part 1: State Register
// This is the sequential logic part, responsible for updating and storing the
    current state.
// ----------------------------------------------------
always @(posedge clk or posedge rst) begin
    if (rst) begin
        // Asynchronous reset, forces the state back to 000 (Fetch)
        current_state <= 3'b000;
    end else begin
        // On the rising edge of the clock, latch the next state as the current
            state.
        current_state <= next_state;
    end
end

// ----------------------------------------------------
// Part 2: Next-State Logic
// This is the combinational logic part, calculating the next state based on the
    current state and opcode.
// ----------------------------------------------------

// Aliases for the current_state bits
wire S2 = current_state[2];
wire S1 = current_state[1];
wire S0 = current_state[0];

// Aliases for the opcode_in bits
wire I3 = I[3];
wire I2 = I[2];
wire I1 = I[1];
wire I0 = I[0];
```

```
// next_state[0] logic
assign next_state[0] = (~S2 & ~S1 & ~S0) |
                       (~S2 & ~S1 & S0 & (I3 & (I1 | I2 | ~I0))) |
                       (S2 & S1 & S0);

// next_state[1] logic
assign next_state[1] = (~S2 & ~S1 & S0 & (~I3 | (~I2 & ~I0))) |
                       (~S2 & S1) |
                       (S2 & ~S1) |
                       (S2 & S0);

// next_state[2] logic
assign next_state[2] = (~S2 & ~S1 & S0 & (I3 & (I0 | I1 | I2))) |
                       (~S2 & S1) |
                       (S2 & ~S1) |
                       (S2 & S0);

// -------------------------------------------------
// Part 3: Output Logic
// These output expressions remain unchanged as they are already in their
    simplified, final form.
// -------------------------------------------------

// PC_INC (Program Counter Increment)
assign PC_INC = S2 & S1 & ~S0;

// IR_WE (Instruction Register Write Enable)
assign IR_WE = ~S2 & ~S1 & ~S0;

// ACC_WE (Accumulator Write Enable)
assign ACC_WE = (~S2 & S1 & ~S0) | (S2 & ~S1 & ~S0);

// InputReg_LE (Input Register Latch Enable)
assign InputReg_LE = ~S2 & S1 & ~S0;

// OutputReg_WE (Output Register Write Enable)
assign OutputReg_WE = ~S2 & S1 & S0;

// ACC_Mux_Select (Accumulator MUX Select)
assign ACC_Mux_Select = S2 & ~S1 & ~S0;

// Assign the current state to the optional output
assign current_state_out = current_state;
```

```
endmodule
```

Behavioral simulation of the structural implementation confirmed that its functional behavior matched the original behavioral-level Verilog implementation, verifying the correctness of the manual logic derivation.

A.3 SiC 4-bit Processor Implementation

```
// ================================================================================
// Top-Level Processor Module (Direct-Connect Version)
// ================================================================================

module Processor(
    // --- Inputs ---
    input wire         clk,
    input wire         rst,
    input wire [3:0]   instruction_in,  // From external instruction memory (ROM)
    input wire [3:0]   external_data_in, // From external data input port

    // --- Outputs ---
    output wire [7:0]  address_out,      // Address to instruction memory (ROM)
    output wire [3:0]  external_data_out // To external data output port
);

    // --- Internal Control Signals ---
    wire PC_INC;
    wire IR_WE;
    wire ACC_WE;
    wire InputReg_LE;
    wire OutputReg_WE;
    wire ACC_Mux_Select;

    // --- Internal Datapath Signals ---
    wire [3:0] ir_out;
    wire [3:0] acc_out;
    wire [3:0] input_reg_out;
    wire [3:0] alu_out;
    wire [3:0] acc_mux_out;

    // 1. Control Unit (CU)
    ControlUnit cu (
        .clk(clk),
        .rst(rst),
        .I(ir_out),
        .PC_INC(PC_INC),
```

```
    .IR_WE(IR_WE),
    .ACC_WE(ACC_WE),
    .InputReg_LE(InputReg_LE),
    .OutputReg_WE(OutputReg_WE),
    .ACC_Mux_Select(ACC_Mux_Select)
);
// 2. ALU
ALU alu_inst (
    .A(input_reg_out),
    .B(acc_out),
    .ALU_Sel(ir_out[2:0]), // <-- Direct connection!
    .Result(alu_out)
);
// 3. Program Counter
reg [7:0] pc_reg;
assign address_out = pc_reg;
always @(posedge clk or posedge rst) begin
    if (rst)
        pc_reg <= 8'h00;
    else if (PC_INC)
        pc_reg <= pc_reg + 1;
end

// 4. Instruction Register
reg [3:0] ir_reg;
assign ir_out = ir_reg;
always @(posedge clk or posedge rst) begin
    if (rst)
        ir_reg <= 4'h7; // Reset to NOP (No Operation) instruction
    else if (IR_WE)
        ir_reg <= instruction_in;
end

// 5. Input Latch
reg [3:0] input_reg;
assign input_reg_out = input_reg;
always @(*) begin
  if (rst) begin
    input_reg = 4'h0;
  end else if (InputReg_LE) begin
    input_reg = external_data_in;
  end // This creates a level-sensitive latch
end
```

```
    // 6. Accumulator MUX
    assign acc_mux_out = ACC_Mux_Select ? 4'b0000 : alu_out;

    // 7. Accumulator (ACC)
    reg [3:0] acc_reg;
    assign acc_out = acc_reg;
    always @(posedge clk or posedge rst) begin
        if (rst)
            acc_reg <= 4'h0;
        else if (ACC_WE)
            acc_reg <= acc_mux_out;
    end

    // 8. Output Register
    reg [3:0] output_reg;
    assign external_data_out = output_reg;
    always @(posedge clk or posedge rst) begin
        if (rst)
            output_reg <= 4'h0;
        else if (OutputReg_WE)
            output_reg <= acc_out;
    end

endmodule

// This module implements the complete Control Unit,
// using a Finite State Machine (FSM) structure. The Boolean expressions
// have been updated according to the revised design.
module ControlUnit (
    // Sequential Logic Inputs
    input clk,          // System clock
    input rst,          // Asynchronous reset signal

    // FSM Inputs
    input [3:0] I, // 4-bit instruction opcode (from IR)

    // FSM Outputs
    output PC_INC,
    output IR_WE,
    output ACC_WE,
    output InputReg_LE,
    output OutputReg_WE,
    output ACC_Mux_Select,
    output [2:0] current_state_out // Optional debug output
```

```
);

// ------------------------------------------------
// Internal Signal Declarations
// ------------------------------------------------

// State Register: 3-bit D-Flip-Flop to store the current state S[2:0]
reg [2:0] current_state;

// Output of the next-state logic (Input to the D-Flip-Flop)
wire [2:0] next_state;

// ------------------------------------------------
// Part 1: State Register
// This is the sequential logic part, responsible for updating and storing the
    current state.
// ------------------------------------------------
always @(posedge clk or posedge rst) begin
    if (rst) begin
        // Asynchronous reset, forces the state back to 000 (Fetch)
        current_state <= 3'b000;
    end else begin
        // On the rising edge of the clock, latch the next state as the current
            state.
        current_state <= next_state;
    end
end

// ------------------------------------------------
// Part 2: Next-State Logic
// This is the combinational logic part, calculating the next state based on the
    current state and opcode.
// ------------------------------------------------

// Aliases for the current_state bits
wire S2 = current_state[2];
wire S1 = current_state[1];
wire S0 = current_state[0];

// Aliases for the opcode_in bits
wire I3 = I[3];
wire I2 = I[2];
wire I1 = I[1];
wire I0 = I[0];
```

```
// next_state[0] logic
assign next_state[0] = (~S2 & ~S1 & ~S0) |
                       (~S2 & ~S1 & S0 & (I3 & (I1 | I2 | ~I0))) |
                       (S2 & S1 & S0);

// next_state[1] logic
assign next_state[1] = (~S2 & ~S1 & S0 & (~I3 | (~I2 & ~I0))) |
                       (~S2 & S1) |
                       (S2 & ~S1) |
                       (S2 & S0);

// next_state[2] logic
assign next_state[2] = (~S2 & ~S1 & S0 & (I3 & (I0 | I1 | I2))) |
                       (~S2 & S1) |
                       (S2 & ~S1) |
                       (S2 & S0);

// ------------------------------------------------
// Part 3: Output Logic
// These output expressions remain unchanged as they are already in their
    simplified, final form.
// ------------------------------------------------

// PC_INC (Program Counter Increment)
assign PC_INC = S2 & S1 & ~S0;

// IR_WE (Instruction Register Write Enable)
assign IR_WE = ~S2 & ~S1 & ~S0;

// ACC_WE (Accumulator Write Enable)
assign ACC_WE = (~S2 & S1 & ~S0) | (S2 & ~S1 & ~S0);

// InputReg_LE (Input Register Latch Enable)
assign InputReg_LE = ~S2 & S1 & ~S0;

// OutputReg_WE (Output Register Write Enable)
assign OutputReg_WE = ~S2 & S1 & S0;

// ACC_Mux_Select (Accumulator MUX Select)
assign ACC_Mux_Select = S2 & ~S1 & ~S0;

// Assign the current state to the optional output
assign current_state_out = current_state;
```

```
endmodule

// ============================================================================
// Arithmetic Logic Unit (ALU) Module
// This is a purely combinational module that performs all arithmetic and logic
// operations for the processor.
// ============================================================================
module ALU(
    // --- Inputs ---
    input wire [3:0] A,          // Input A (from Input Register)
    input wire [3:0] B,          // Input B (from Accumulator)
    input wire [2:0] ALU_Sel,    // Operation select signal

    // --- Output ---
    output reg [3:0] Result      // 4-bit result of the operation
);

    // This combinational block continuously calculates the result
    // based on the current inputs.
    always @(*) begin
        case (ALU_Sel)
            3'b000: Result = A + B;       // ADD
            3'b001: Result = A - B;       // SUB
            3'b010: Result = A & B;       // AND
            3'b011: Result = A | B;       // OR
            3'b100: Result = A ^ B;       // XOR
            3'b101: Result = ~B;          // NOT (on B)
            3'b110: Result = A;           // Pass A
            3'b111: Result = B;           // Pass B
            default: Result = 4'h0;       // Default to 0 for safety
        endcase
    end

endmodule
```