Design and Characterization of a 4 MS/s 8-bit Flash ADC in 180 nm CMOS for Dual-Channel Image Sensor Readout

By:
Francesco Marrocco

A Thesis
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Master of Science
in
Electrical and Computer Engineering

May 2026

APPROVED:

Professor Suat Ay, Major Advisor

Ibrahim Bozyel, Co-Advisor

Professor Ulkuhan Guler, Thesis Committee Member

Professor John McNeill, Thesis Committee Member

Abstract

The rapid scaling of global semiconductor manufacturing has made device characterization an essential pillar of modern supply chains. Among the most ubiquitous mixed-signal building blocks, the analog-to-digital converter (ADC) demands rigorous performance validation to ensure the reliability of the systems that use them.

This thesis presents the design, implementation, and characterization of a 4 MS/s 8-bit Flash ADC fabricated in 180 nm CMOS process for use as a readout converter in a CMOS image sensor. Two identical ADC instances are implemented to serve even and odd pixel columns respectively, providing the imager with a frame rate of approximately 40 FPS. The architecture follows the standard Flash topology: a sample and hold circuit with a slew rate of about 3.6V/µs, a comparator with a pre-charged PMOS and latch structure, and typical one-hot ROM encoder.

Device characterization strictly follows the IEEE 1241 standard for ADC testing, utilizing standard laboratory equipment, Verilog control logic on a custom PCB, and a Python GUI for automatic data acquisition and processing. Measured static performance yielded a peak DNL of +3.15/-1.0 LSB and INL of +2.68 / -3.19 LSB. Dynamic characterization at 4 MS/s produced an SINAD of 31.52 dB, THD of -32.9 dBFS, and an ENOB of 4.94 bits. Beyond ADC characterization, this work establishes a reusable test framework and documented procedure to support future integrated circuit validation efforts.

Acknowledgments

Center for Education of Microchip Designers (CEMiD)	For their sponsorship of the WCIS tapeout.
Professor Suat Ay	For his expertise, mentorship, and for fostering an environment where I could explore freely and grow as a student, researcher, and engineer.
Ibrahim Bozyel	For his collaboration, guidance, and for the many late nights spent working together in the lab.
Professor Ulkuhan Guler **Professor John McNeill**	For serving on my thesis committee and for their time and feedback at my defense.
Rosemarie Ivaldi **Dennis Marrocco** **Vanessa Marrocco**	For their unconditional love and encouragement throughout my academic journey.
Charlotte Leavitt **Akhilesh Yarlagadda** **Neil Kale**	For their friendship and companionship this work, and many others, would not have been possible without them.

Contents

List of Figures

1 Introduction

In May 2025, in completion of my undergraduate Major Qualifying Project, the Worcester CMOS Image Sensor (WCIS) was taped out at TSMC through funding from the Center for Education of Microchip Designers (CEMiD). I was responsible for the ADC, encryption, and serializer modules of the imager [1]. In CMOS image sensors (CIS), the readout stage is typically one of the most critical steps between capturing light with the photodiode array and producing the final digital image. Semiconductor manufacturers need to be able to characterize the sampling speeds, linearity, effective resolution, and power consumption, among other parameters, of these stages to produce reliable devices and sell them at their listed specifications.

In an imager, if the ADC cannot be independently accessed, the only alternative is to drive it through the pixel array. This approach is problematic, however, as the measured results would reflect a combination of both pixel and ADC characteristics, making it impossible to isolate the source of any errors in the ADC itself. Morishita et al. of Renesas Electronics Corporation [2] demonstrate this challenge directly, presenting a test setup for characterizing a single-slope column parallel ADC in a CIS chip. Using an optical input source, dark characteristic measurements could be performed reliably, but brightness-dependent and column interference measurements were fundamentally limited by the controllability and resolution of the optical signal. In particular, isolating a single column for interference testing was unachievable due to optical diffraction, which prevented precise column-edge patterning. Their solution was an on-chip dual-path circuit with a multiplexer that could switch the input of the ADC to either the output of the pixel array or an external electrical input, enabling accurate, isolated ADC characterization under real operating conditions. Similarly, the WCIS has the same approach, ensuring the ADC

can be properly characterized independent of pixel behavior and in line with established best practices for imager testing.

The characterization methodology in this work follows the IEEE 1241: Standard for Terminology and Test Methods for Analog-to-Digital Converters [3], covering both static and dynamic measurements. Static metrics include gain, offset, differential nonlinearity (DNL), and integral nonlinearity (INL), extracted via a code-density histogram test. Dynamic metrics include signal-to-noise-and-distortion ratio (SINAD), effective number of bits (ENOB), signal-to-noise ratio (SNR), total harmonic distortion (THD), spurious-free dynamic range (SFDR), dynamic range (DR), Walden figure of merit ($\mathrm{FOM_W}$) and Schreier figure of merit ($\mathrm{FOM_S}$).

At a large scale, companies like Teradyne develop sophisticated automated test equipment capable of high-throughput production characterization. This work takes a parallel approach at the academic scale, designing a custom PCB, FPGA-based digital controller, and Python GUI capable of full IEEE 1241-compliant characterization of the WCIS and serving as a reusable platform for future student tapeouts at WPI.

The remainder of this thesis is organized as follows. Section 2 provides background on fundamental performance metrics and setup requirements. Section 3 describes the WCIS chip architecture and the Flash ADC circuit design elements. Section 4 details the test setup and characterization methodology. Section 5 presents and discusses the measured results. Lastly, Section 6 discusses the findings, proposes improvements, and outlines directions for future work.

2 Background

This section provides the theoretical foundation for the ADC characterization methodology used in this work. The relevant static and dynamic performance metrics are defined, followed by a discussion of the coherent sampling requirements imposed by IEEE 1241 for accurate results.

2.1 Fundamental ADC Characteristics

The IEEE 1241 Standard for Terminology and Test Methods for Analog-to-Digital Converters [3] defines the characterization methodology used in this work. The standard divides characterization into two categories: static and dynamic. Static errors are found by injecting a low frequency signal (like a linear ramp from 1V to 2V over 1ms) and can be characterized by the non-idealities between code transition levels (CTL). Dynamic errors are found by injecting a higher frequency signal (like a 100kHz sine wave) and can be characterized by the non-idealities in the frequency domain [3].

After injecting a high-frequency sine wave into the ADC, the output will be a quantized version of this input. The Fast Fourier Transform (FFT) then maps the output waveform into the frequency domain. The resulting spectrum plot provides insight into the presence of noise and distortion in the signal. First, noise is represented by the signal-to-noise ratio (SNR) and dynamic range (DR), where DR measures how far the signal amplitude is above the noise floor (dBFS) and SNR represents that same ratio using the RMS value of the signal amplitude.

$$\mathrm{SNR} = 20\log\left(\frac{V_{signal,RMS}}{V_{noise,RMS}}\right) \tag{1}$$

$$\mathrm{DR} = 20\log\left(\frac{A_{signal}}{V_{noise,RMS}}\right) \tag{2}$$

Next, distortion is captured by the signal-to-noise-and-distortion ratio (SINAD) and the spurious-free dynamic range (SFDR), where SFDR measures how far the signal amplitude is above the largest harmonic spur and SINAD folds distortion into the RMS noise floor to obtain the combined noise and distortion.

$$\text{SINAD} = 20\log\left(\frac{V_{signal,RMS}}{V_{noise+distortion,RMS}}\right) \tag{3}$$

$$\text{SFDR} = 20\log\left(\frac{A_{signal}}{A_{spur,max}}\right) \tag{4}$$

Additionally, the total harmonic distortion (THD) represents the ratio of the RMS amplitude of all harmonics to the RMS amplitude of the fundamental signal.

$$\text{THD} = 20\log\left(\frac{V_{harmonic,RMS}}{V_{signal,RMS}}\right) \tag{5}$$

Lastly, the effective number of bits (ENOB) can be calculated to represent the true resolution of the ADC accounting for all errors introduced in the quantization process.

$$\text{ENOB} = \frac{\text{SINAD} - 1.76}{6.02} \tag{6}$$

After applying a low-frequency ramp signal, a line representing the code transition level (CTL) transfer function, $T[k]$, is defined. Section 4.3 provides more detail on how to calculate this function. When comparing $T[k]$ to the ideal CTL function, a straight line from V_{reflo} to V_{refhi}, there will be significant error in the slope at certain points, the distance away from the line, and the start and end points of the two functions. The static parameters quantify those sources of error. First, the terminal gain (A) and offset (V_{OS}) are defined by how much the measured CTL function must be scaled and shifted in order to match the start and end points of the ideal CTL function.

$$A = \frac{\left(2^N - 2\right) Q}{T[2^N - 1] - T[1]} \tag{7}$$

$$V_{OS} = T_1 - A \cdot T[1] \tag{8}$$

Next, the differential nonlinearity (DNL) measures the error in step width between each code transition and compares it to the ideal bin width $Q = \frac{V_{FSR}}{2^N}$. For example, in an ideal ADC each code, 001 010 011 etc, would represent one step in voltage. If an 8-bit ADC had a voltage range of 2V, the bin width would be about 8mV per step. In a real ADC, this bin width can vary. Sometimes the bin is too wide and sometimes it is skipped entirely, which would be a missing code. The following equation is used to determine the DNL from the histogram plot where $W[k] = G \times (T[k+1] - T[k])$ is the measured bin width.

$$DNL[k] = \frac{W[k] - Q}{Q} \tag{9}$$

Lastly, the integral nonlinearity (INL) measures the cumulative deviation between the measured CTL function and the ideal CTL function.

$$INL[k] = \frac{A \cdot T[k] + V_{OS} - T_{\text{ideal}}[k]}{V_{FS}} \tag{10}$$

2.1.1 Figure of Merit

The Walden and Schreier figure of merit (FOM) is one way to condense the wide variety of ADC performance metrics into a single value that captures the tradeoff between power dissipation, sampling rate, and resolution [4]. With the Walden FOM we can calculate how much energy it costs to produce one conversion step, expressed in pJ/step. Starting with the basic equation for power, $P = dE/dt$, and substituting $dt = 1/f_s$ gives $P = E \cdot f_s$. Solving for energy per sample yields $E = P/f_s$. To account for the resolution of the ADC, this energy is divided by 2^{ENOB}, giving the energy spent per code step:

$$\text{FOM}_W = \frac{P}{f_s \cdot 2^{\text{ENOB}}} \tag{11}$$

where P is the power consumption, f_s is the sampling frequency, and 2^{ENOB} is the effective number of output codes.

Since the Walden FOM captures energy per conversion step, it becomes a limited metric when comparing extremely high-speed ADC's at GS/s sampling rates. The Schreier FOM addresses this by measuring how much SINAD a design achieves relative to its power consumption and Nyquist bandwidth; penalizing designs whose SINAD degrades as sampling rate increases.

$$\text{FOM}_S = \text{SINAD} + 10\log_{10}\left(\frac{f_s/2}{P}\right) \tag{12}$$

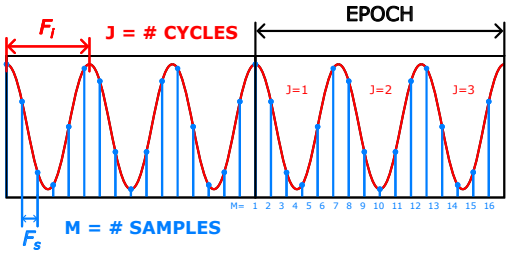
Unlike the Walden FOM where a lower value indicates better performance, a higher Schreier FOM indicates a more efficient design, and is particularly useful for comparing designs across different process nodes and architectures. Note that SINAD can be replaced with DR in Equation 12 to evaluate noise performance independently of harmonic distortion.

2.2 Coherent Sampling

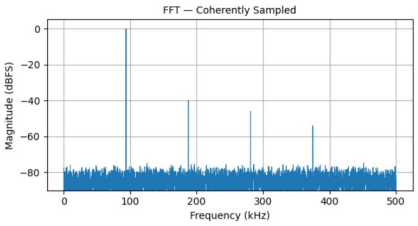
IEEE 1241 defines coherent sampling as "sampling of a periodic signal such that there is an integer number of signal cycles in the waveform epoch" [3]. Coherent sampling occurs when the following condition is satisfied:

$$F_i \cdot M = F_s \cdot J \tag{13}$$

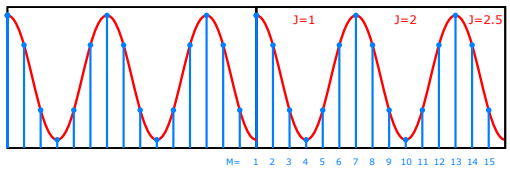
where F_i is the input frequency, M is the number of samples in the epoch, F_s is the sampling frequency, and J is the number of complete input signal cycles within the epoch, required to be relatively prime to M. Figure 1a and 1b shows an example of a coherently sampled sinusoidal signal.



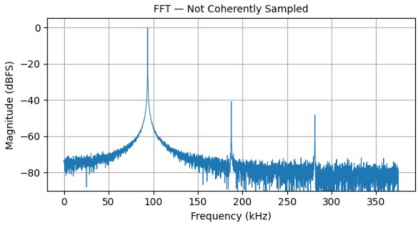
(a) Coherently sampled time-domain signal satisfying Equation 13.



(b) FFT of coherently sampled signal showing energy concentrated at the fundamental bin.



(c) Non-coherently sampled time-domain signal where J is non-integer, resulting in an epoch discontinuity.



(d) FFT of non-coherently sampled signal showing spectral leakage.

Figure 1: Comparison of coherent and non-coherent sampling in the time and frequency domains.

Coherent sampling is critical for dynamic ADC testing because it eliminates spectral leakage, seen in Figure 1d, in the FFT output. If Equation 13 is not satisfied, the waveform epoch does not end at the same phase it began, producing a discontinuity when epochs are appended, Figure 1c. This discontinuity manifests as spectral leakage in the FFT, spreading energy from the fundamental tone into adjacent frequency bins and degrading the accuracy of SNR, SINAD, and THD measurements.

The requirement that J be relatively prime to M ensures that no two samples in the epoch coincide at the same phase of the input signal. If J and M share a common factor, the sampled points repeat a subset of phase positions rather than covering all M unique points, reducing the effective resolution of the test.

3 Architecture

This section describes the architecture of the WCIS chip and the circuits within the Flash ADC, including the sample-and-hold circuit, comparator array, and encoding logic. The WCIS chip has an area of $4\,\mathrm{mm}^2$ with a dual-channel rolling shutter readout. For the purposes of this thesis, the pixel control and sampling path is bypassed entirely and the Flash ADC is driven directly through the V_{test_in} pin, isolating the converter for characterization.

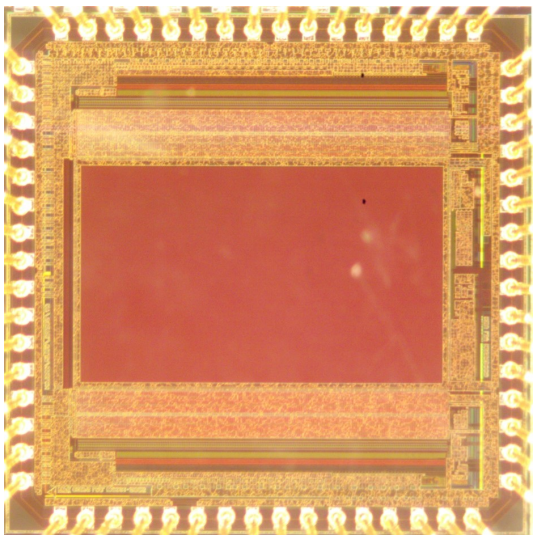


Figure 2: Image of the Worcester CMOS Image Sensor (2x2mm, 180nm 1P6M 3v3/1.8V CMOS)

3.1 WCIS

The WCIS, Figure 2, contains a 420×256 active pixel sensor (APS) array, where each pixel contains a photodiode and control transistors for reset, buffering, and selection. The chip has a top and bottom readout module. Each module contains a column decoder, where the top module accesses the even columns and the bottom accesses the odd columns. Then the Global ASP module can adjust the gain of the pixel output signal before it is sampled by the Flash ADC.

The analog signal path can be bypassed entirely via the analog input pin V_{test_in}, which was used exclusively for ADC characterization in this work. Figure 3 shows two highlighted blocks, the Flash ADC and the SERDES. Between them is an encryption engine that takes a 16-bit key and X-OR's 8-bits of the key at a time to the 8-bit data, and shifts the key for each sample. The only way to retrieve the data would be to know the exact key and the rotation of the key during that sample. This module was also excluded from analysis in this thesis. The supporting circuits in gray provide configurable power settings to the chip.

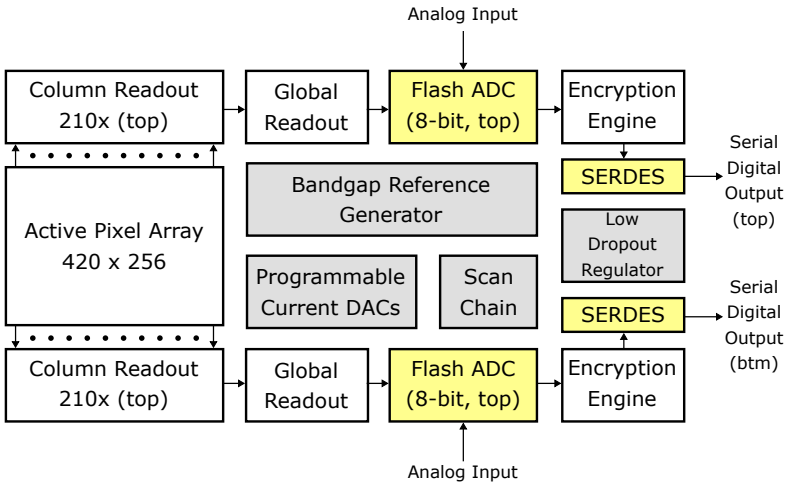


Figure 3: WCIS block diagram

3.2 Flash ADC

The readout chain of the WCIS chip has 3 power pins, 1 analog input pin, 6 digital input pins, and 2 digital output pins. The typical Flash ADC topology implemented in WCIS and shown in Figure 4, has three stages: the sample and hold circuit, comparison circuit, and encoding logic.

The sample-and-hold circuit consists of a 330 fF capacitor buffered by a unity-gain op-amp driving the resistor-comparator ladder. The settling time is controlled by the SH pin, which is active high. The reference voltages $V_{ref-high}$

and $V_{ref-low}$ can be supplied by either a DAC or a low-noise power supply, and must be set within the ICMR of the ADC's comparators, 0.6V - 3.3V.

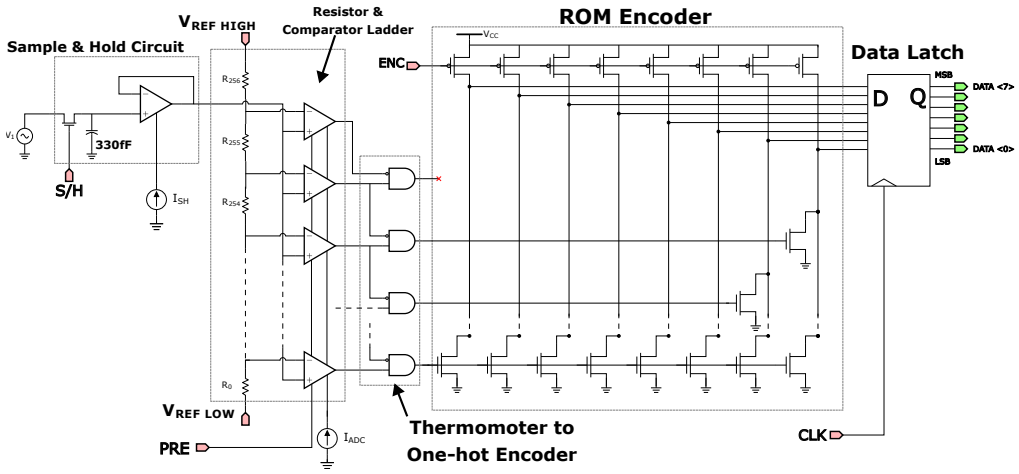


Figure 4: High level schematic of WCIS 8-bit flash ADC

The Flash ADC layout is shown in Figure 5. Two identical instances are placed at the top and bottom of the chip, mirrored along their long edge to serve the even and odd pixel columns respectively. Each instance measures 1.3 mm × 0.1 mm and is organized as a horizontal array of 256 steps at a 4 μm pitch, where each step contains a unit resistor, comparator, and encoder cell stacked vertically. The 8-bit parallel output latches sit on the left side feeding into the serializer, while the current biasing, scan chain, and sample-and-hold circuits are placed on the right at the analog input.

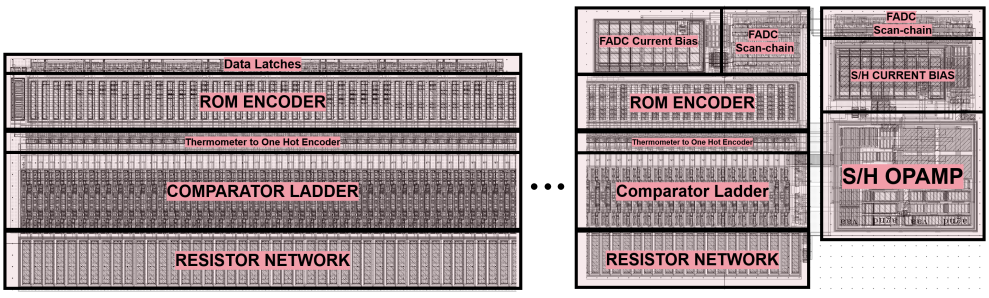


Figure 5: Layout of Flash ADC, 1.3mm x 0.1mm, 4μm pitch

The sample and hold circuit shown on the left side of Figure 4 can be characterized through simulation to determine how quickly it can sample a waveform. In Figure 6, a sinusoidal signal is applied to the input (GREEN). When the ADC_SH pin (RED) is high, the sampled output (sh_out, YELLOW) tracks the input. When ADC_SH goes low the waveform output holds that sampled voltage level. When ADC_SH rises back to high, there is a period where the capacitor in the S/H circuit has to charge up again. Measuring the time it takes for the capacitor to reach back to the voltage level, we get a slew rate of about $3.6\,\mathrm{V}/\mu\mathrm{s}$.

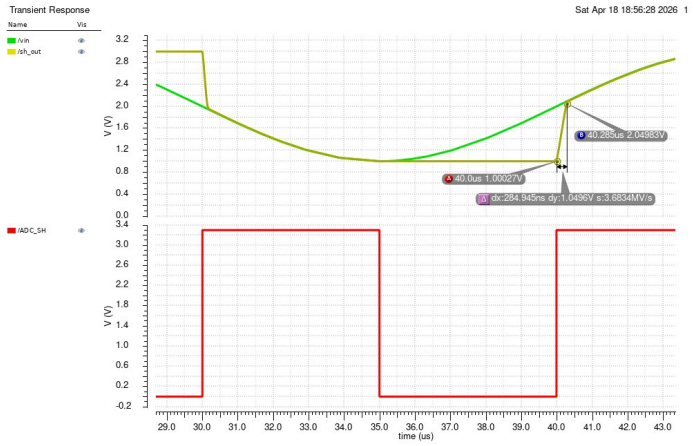


Figure 6: Simulation of sample and hold circuit

The next stage is the resistor-comparator ladder network. The size of the ladder, the number of comparators, is equal to 2^N where N is the number of bits in the ADC. Since this is an 8 bit ADC, there are 256 comparators. Additionally, there is a resistor between each comparator and a resistor before each reference voltage, as shown in Figure 4. Each resistor has a resistance of about $40\,\Omega$, which is about the smallest that can be made in 180 nm CMOS, and gives a total ladder resistance of about $10\,\mathrm{k}\Omega$. These resistors create the DC reference voltage steps between reference low and reference high. The resolution of 1 LSB can be determined by dividing the voltage range by the number of code steps: $V_{FSR}/2^N$. In the case of my tests, the full scale range

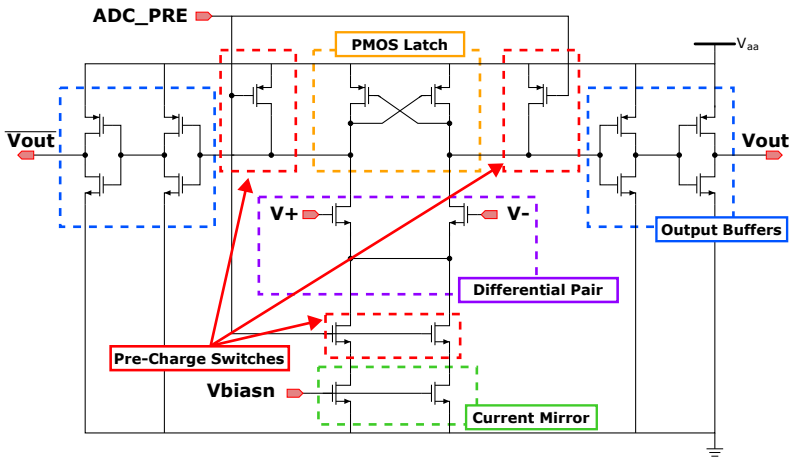


Figure 7: Transistor level schematic of comparator used in FADC

was about 2 V, giving 8 mV of resolution per LSB.

The comparator shown in Figure 7 consists of a differential input stage, a cross-coupled PMOS latch similar to a strongARM latch in [5], pre-charge switches, and output buffers for the inverted and non-inverted output. During the precharge phase (PRE = 0V), the PMOS precharge switches pull both output nodes to V_{aa}. During the comparison phase (PRE = 3.3V), the tail current mirror is enabled and the differential pair steers current based on the difference between the sampled input $V+$ and the resistor ladder reference $V-$. If $V+ > V-$, V_{out} gets pulled high by the differential pair and latched from the PMOS latches at the top.

The 256 comparators produce a thermometer code, in which all comparators below the input voltage output a logic high and all comparators above it output a logic low. Similar to how a mercury thermometer rises as temperature increases. This thermometer code is first converted to a one-hot code using a bank of AND gates, producing a single logic high on exactly one of the 256 lines. The active line corresponds to the quantized input voltage.

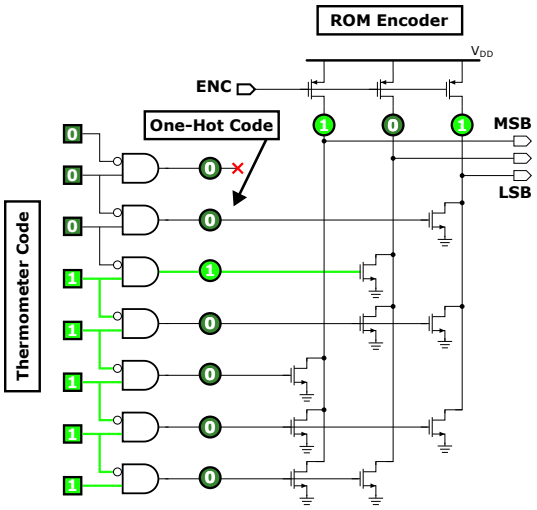


Figure 8: Example of encoder logic shown in a 3-bit ADC

The one-hot code then drives a hardwired ROM encoder, shown in Figure 8. Each column of the ROM represents one output bit, and the PMOS devices at the top of each column are precharged to V_{DD} when ADC_ENC is low. When ADC_ENC goes high, each column is either pulled low by the NMOS device connected to the active one-hot line, or remains high due to parasitic capacitance, producing the corresponding binary output bit. The result is the 8-bit binary representation of the input voltage. The example in Figure 8 shows a simplified 3-bit version of this circuit, where the active one-hot line produces a binary output of 101.

3.3 Serializer and Scan Chain

The SERDES, or serializer, takes the 8 parallel lines from the D-latches and "serializes" them into one wire that is sent out of the chip and into the FPGA. To do this, the data needs to be loaded into the shift register (LOAD pin) and then shifted out (SHIFT pin). LOAD is attached to the select pin on a two input MUX, when it is high the MUX selects the new bit to be shifted into the register. As LOAD is high, the rising edge on SHIFT will shift the new byte of data into the eight shift registers. When LOAD goes low, selecting

the previous bit in the shift register chain, eight rising edges on the SHIFT pin will shift the data out of the chip.

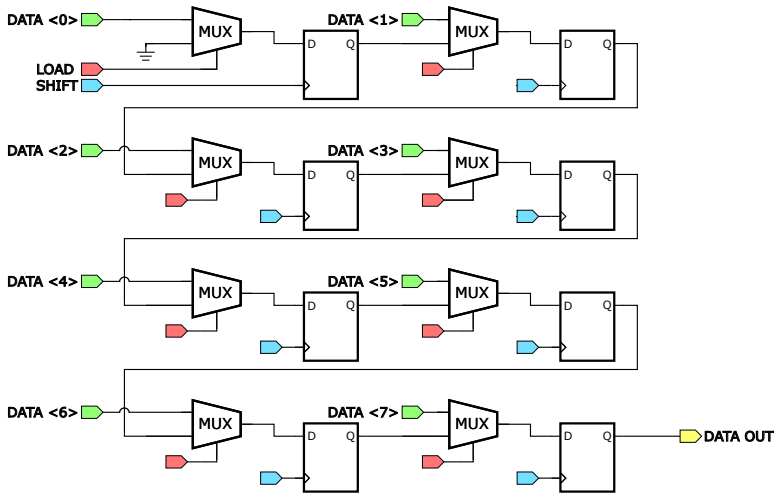


Figure 9: Schematic of 8-bit serializer

The scan chain operates in the same exact way, but instead of shifting data out, it shifts data in. Throughout the different modules of the chip, like the pixel array, global ASP, or ADC, there are the same MUX's and shift registers. In the Flash ADC there are eight registers, four for the S/H circuit, and four for the comparator network. The binary value of these registers maps to different power settings for the ADC. These are described in Table 1 along with the highlighted power modes used in testing.

ADC S/H Register Values ($I_{in} = 10\,\mu$A)						FADC Register Values ($I_{in} = 5\,\mu$A)					
1	**2**	**3**	**4**	**Current (μA)**	**Hex**	**1**	**2**	**3**	**4**	**Current (μA)**	**Hex**
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	5	1	0	0	0	1	2.5	1
0	0	1	0	10	2	0	0	1	0	5	2
0	0	1	1	15	3	0	0	1	1	7.5	3
0	1	0	0	20	4	0	1	0	0	10	4
0	1	0	1	25	5	0	1	0	1	12.5	5
0	1	1	0	30	6	0	1	1	0	15	6
0	1	1	1	35	7	0	1	1	1	17.5	7
1	0	0	0	40	8	1	0	0	0	20	8
1	0	0	1	45	9	1	0	0	1	22.5	9
1	0	1	0	50	A	1	0	1	0	25	A
1	0	1	1	55	B	1	0	1	1	27.5	B
1	1	0	0	60	C	1	1	0	0	30	C
1	1	0	1	65	D	1	1	0	1	32.5	D
1	1	1	0	70	E	1	1	1	0	35	E
1	1	1	1	75	F	1	1	1	1	37.5	F

Table 1: Scan chain power settings for S/H and FADC registers. Red = LPM, Yellow = RPM, Green = HPM.

3.4 Operation

Figure 10 demonstrates the timing order for each of the digital pins in the ADC and serializer modules. It is important to note that the space between each of the rising edges in the diagram were adjusted in real implementation to satisfy the appropriate settling time for the chip under test. When fabricated, chips exhibit variation in transistor speeds across process corners (FF, SS, FS, SF), which affects these timing requirements. One sample cycle requires 16 steps, where the bottom two rows in the diagram show the LOAD and SHIFT pins connected to the MUX's and shift registeres of the serializer. The speed of your base clock multiplied by those 16 steps will determine your sampling rate. There are eight rising edges of the SHIFT pin for each sample cycle, one for each bit, where the dotted red lines show the shifted out data bit from the chip. The eighth rising edge both shifts the last bit out and also shifts the new byte into the shift registers. For easier configurability, the time between rising edges of the PRE, ENC, and CLK pins can be adjusted relative to the rising and falling edges of the SHIFT clock, but the order must remain the

same. First, SH rises, and is held high until the voltage has settled. When SH is pulled low, the signal is held, and PRE is pulled high to enable the comparators. Once the thermometer code has settled, the ENC pin converts it to one-hot encoding and then to binary via the ROM encoder. Finally, the CLK pin latches the eight parallel data bits to be shifted out.

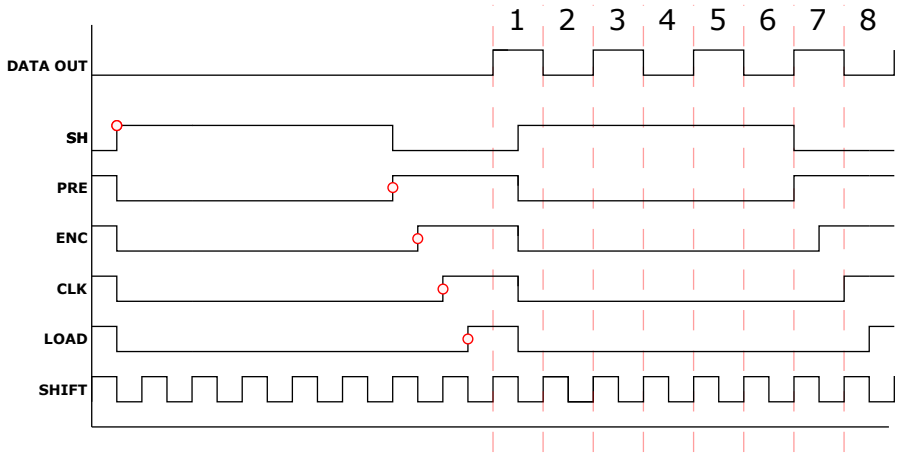


Figure 10: Timing diagram of FADC control logic

4 Methods

This section describes the experimental procedures used to characterize the ADC's dynamic and static performance, as well as its power consumption. Dynamic testing analyzes the output waveform in the frequency domain to extract key figures of merit, while static testing uses a histogram-based approach on a low-frequency ramp signal to measure linearity. Power isolation techniques, including PCB reworks and shunt resistance measurements, are also described to ensure accurate per-channel power estimation. The full GitHub repository with all the Verilog modules, Python GUI, and Jupyter Notebooks can be found here: https://github.com/Vision314/FLASH-ADC-CHARACTERIZATION.

4.1 Test Setup

The test setup for this experiment is shown in Figure 11. Power was supplied by a low-noise Agilent E3630A, and power consumption was measured via a 1.02 Ω shunt resistor using a Tektronix DMM4050 6.5-digit multimeter. The input signal was generated by a Tektronix AFG3021B function generator, capable of producing sine and ramp waveforms with a total harmonic distortion of about 0.2% [6].

4.1.1 PCB Carrier Board

The carrier board PCB shown in Figure 12, was designed by my co-advisor, Ibrahim Bozyel, with the intention to be used with different student tape-out projects with the swappable chip-on-board (COB) module. The board includes two external ADCs (LTC1407), two DACs (DAC53204RTER), two video difference amplifiers (LT6552), and two op-amps (LPV542DGKR). The 5V supply connects to three LDO's (LT1963) driving the Vaa (3.3V) rail

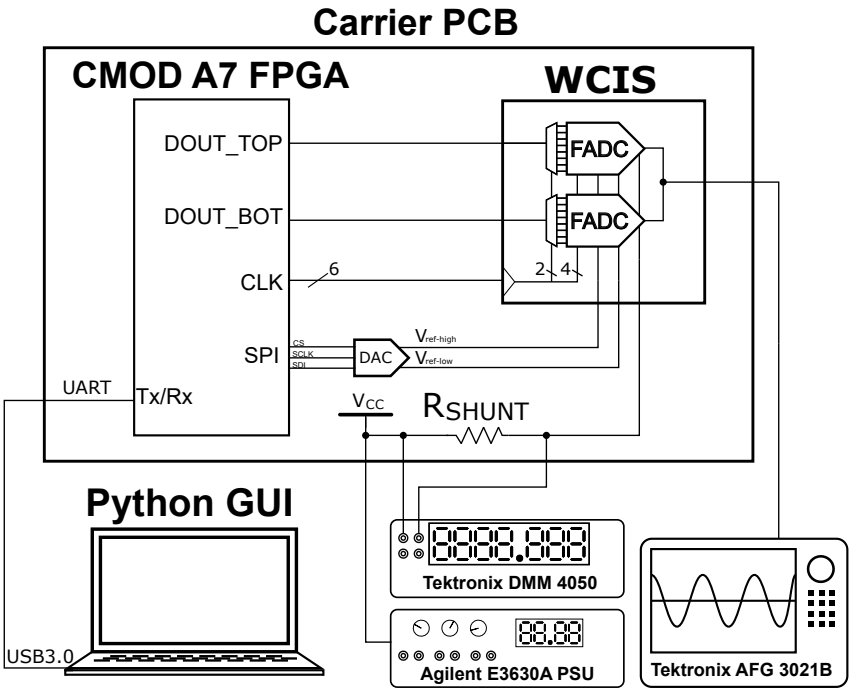


Figure 11: Diagram of test setup

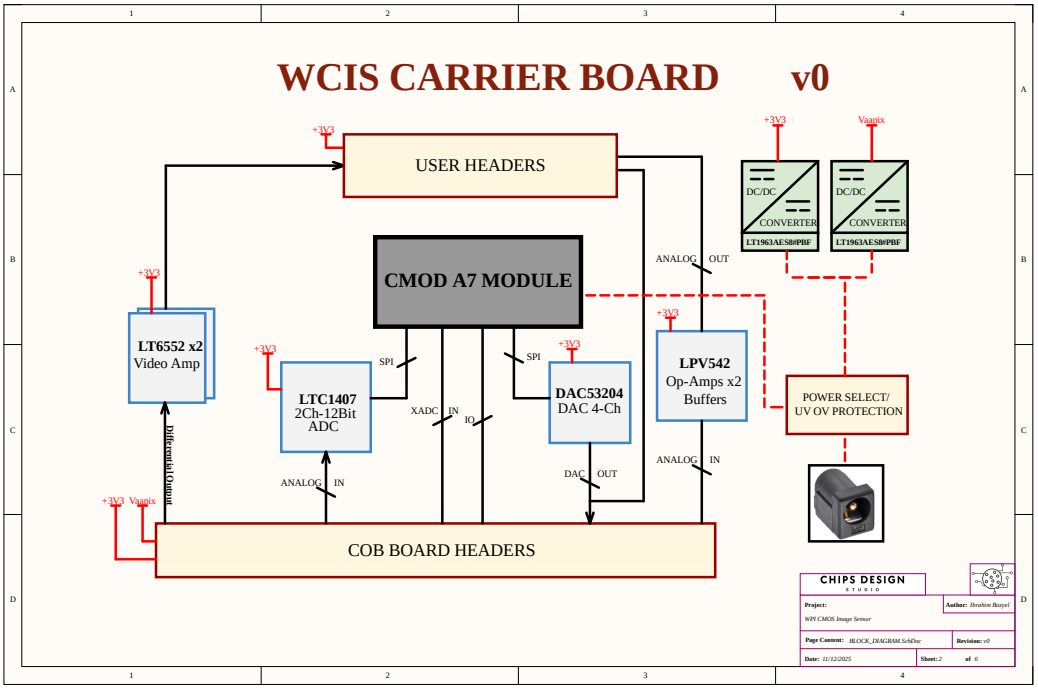


Figure 12: WCIS carrier board block diagram

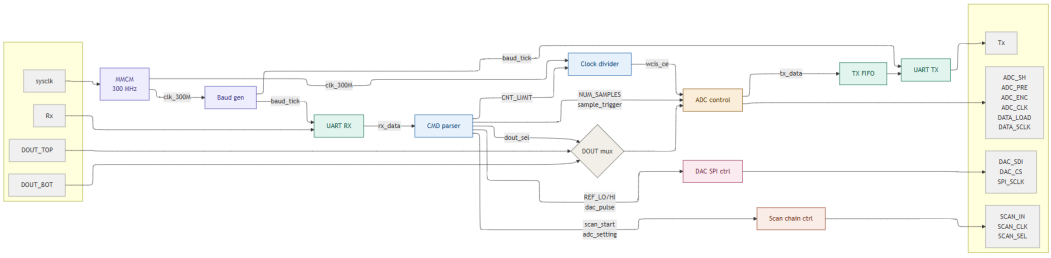


Figure 13: High level diagram of Verilog modules

and two other adjustable supply voltages. A more detailed schematic can be found in Appendix 2.

The Digilent CMOD A7-35T FPGA, was used to control the WCIS and all the peripherals because it provided sufficient GPIO pins, appropriate clock speeds, a compact form factor, at an accessible price point. UART communication with the Python GUI is handled with the on-board FT2232HQ USB-to-UART bridge, eliminating the need for additional hardware. The FPGA was clocked at 300 MHz using the onboard PLL, configured via the `clk_wiz_0` IP core.

Figure 13 shows a block diagram for the Verilog modules that control the test setup. The command parser receives 3 bytes at a time through the Rx line wire from the GUI, where the first byte is the instruction byte, and the last two are data bytes for that instruction. A clock-enable generator (`wcis_ce_gen`) divides the 300 MHz clock down to an adjustable rate used to drive the ADC control logic. The ADC control FSM follows the timing described in Figure 10 exactly. The DAC SPI controller programs the DAC53204 with the desired reference voltages over SPI, and the scan chain controller shifts configuration bits into the chip to control different modes of power consumption. Captured ADC samples are written into a TX FIFO and streamed back to the host PC over UART.

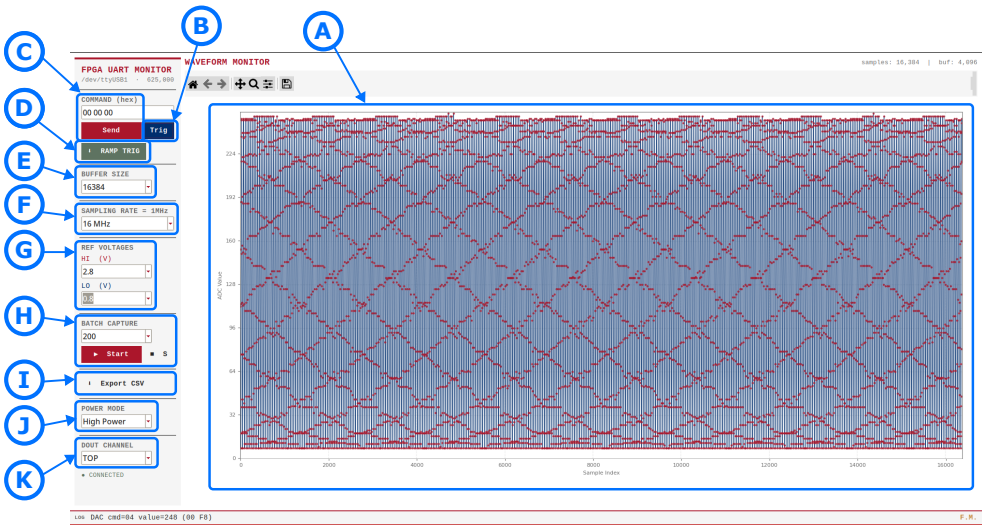


Figure 14: Screenshot of custom Python GUI highlighting important functionality

4.1.2 Python GUI

The Python GUI, shown in Figure 14, was developed to interface with the WCIS Flash ADC over UART and served as the primary control and data acquisition tool for all characterization measurements. The following describes each labeled component.

The GUI is organized into several functional controls, labeled in Figure 14. The waveform monitor (A) displays ADC output codes in real time as data arrives over UART. A single acquisition epoch is initiated via the Trigger button (B), while the Buffer Size control (E) sets the number of samples per epoch between 1024 and 16384. The sampling rate (F) is configured by selecting the FPGA base clock frequency from a dropdown, and fine-grained division is available via OP-CODE command `01 XX XX`. Where `XX XX` represents the clock divisor minus 1. Reference voltages (G) are set over SPI to the on-board DAC, between 0 V to 3.3 V. Batch Capture (H) automates multi-epoch collection and saves results to a structured folder hierarchy. The OP-CODE command interface (C) exposes direct low-level control for configurations not available through the standard GUI controls. Additional controls include a

Ramp Trigger (D), CSV export (I), power mode selection (J), and DOUT channel selection (K).

4.2 Dynamic Characterization Methods

Dynamic characterization was performed by applying a sinusoidal input to the ADC at a target frequency of $F_i \approx 100\,\text{kHz}$ between 0.8V and 2.8V across six sampling frequencies ranging from 500 kHz to 5 MHz. Test parameters were selected to satisfy the coherent sampling condition discussed in Section 2.2, with $M = 8192$ samples per epoch. The resulting parameters are summarized in Table 2, where F_i is the signal generator's achieved output frequency, M is the number of samples in the epoch, J is the number of cycles, and F_s is the sampling rate.

Table 2: Dynamic Test Parameters

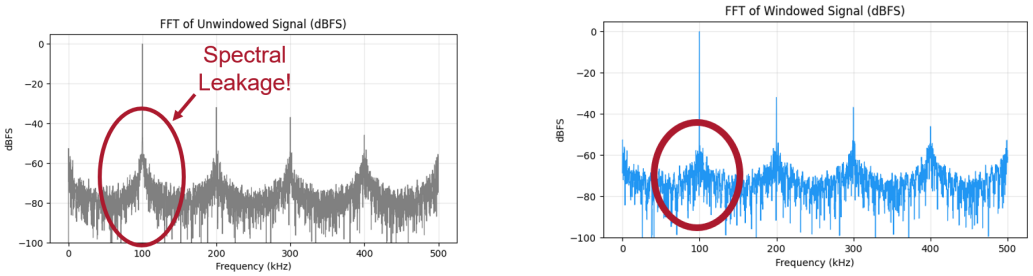
Parameter	**Units**	**Value**					
F_s	MS/s	0.5	1.0	2.0	3.0	4.0	5.0
J	—	1637	819	409	273	205	163
M	—	8192	8192	8192	8192	8192	8192
F_i	Hz	99.91k	99.98k	99.85k	99.98k	100.10k	99.49k

The target of $F_i \approx 100\,\text{kHz}$ was chosen for two reasons. First, a high input frequency ensures adequate spectral separation between signal, harmonics, and noise floor bins. Second, since $J = F_i \cdot M/F_s$, a higher F_i produces a larger J, increasing the number of candidate prime values near the desired coherent frequency and making fine frequency tuning on the function generator reliable. It is also important to keep the same input frequency for all the tests, as changing it between sampling rates could produce different distortion or noise affects, making it harder to compare the performance of different rates.

Since only the magnitude spectrum is needed for the metrics defined in Section 2.1, the phase of each captured epoch is irrelevant. To reduce variance in the spectral estimate, 100 epochs of 8192 samples were captured at each

sampling frequency and their FFT magnitudes averaged. Data acquisition was automated using the batch capture function in the Python GUI described in Section 4.1.2.

Despite careful frequency selection, residual spectral leakage, Figure 15a remains around the fundamental tone and harmonics. A Hanning window was applied to each epoch prior to the FFT to suppress this leakage, with the result shown in Figure 15b.



(a) Initial FFT spectrum showing spectral leakage around fundamental tone

(b) FFT spectrum after applying Hanning window

Figure 15: Effect of Hanning windowing on spectral leakage

After the window is applied, the IEEE 1241 standard recommends using three or four-parameter sine wave fitting to estimate the amplitude, frequency, and phase of the fundamental tone. An alternative approach presented in [7], referred to as the Tabei and Ueda method, was used here and gave similar results to other signal estimation techniques.

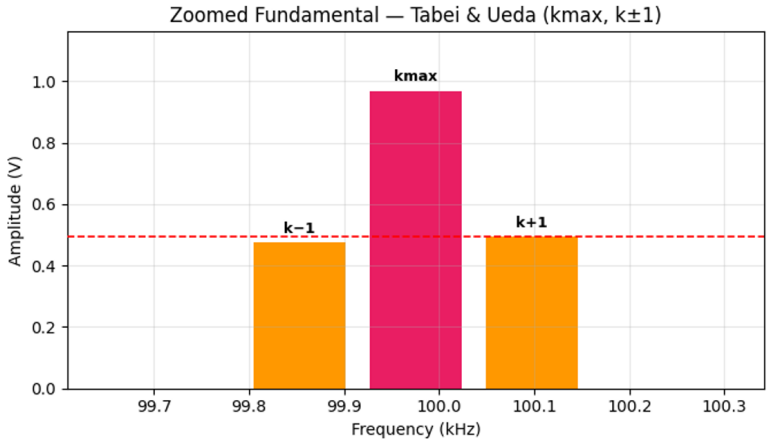


Figure 16: Peak bin k_{max} and neighboring bins $k \pm 1$ used by the Tabei and Ueda method to estimate the fractional bin offset and recover the true fundamental signal

The index of the maximum magnitude bin in the FFT spectrum is first identified as k_{max}. The ratio r is then formed using the adjacent bin with the next highest magnitude, Figure 16:

$$r = \frac{|G(k_{max} \pm 1)|}{|G(k_{max})|} \tag{14}$$

where the -1 bin is used if $|G(k_{max} - 1)| > |G(k_{max} + 1)|$, and the $+1$ bin otherwise. Defining r accordingly, the frequency estimate f is:

$$f = k_{max} + \frac{1 - 2r}{1 + r} \tag{15}$$

when the $k_{max} - 1$ bin is used, or

$$f = k_{max} - \frac{1 - 2r}{1 + r} \tag{16}$$

when the $k_{max}+1$ bin is used. The amplitude and phase of the fundamental are then:

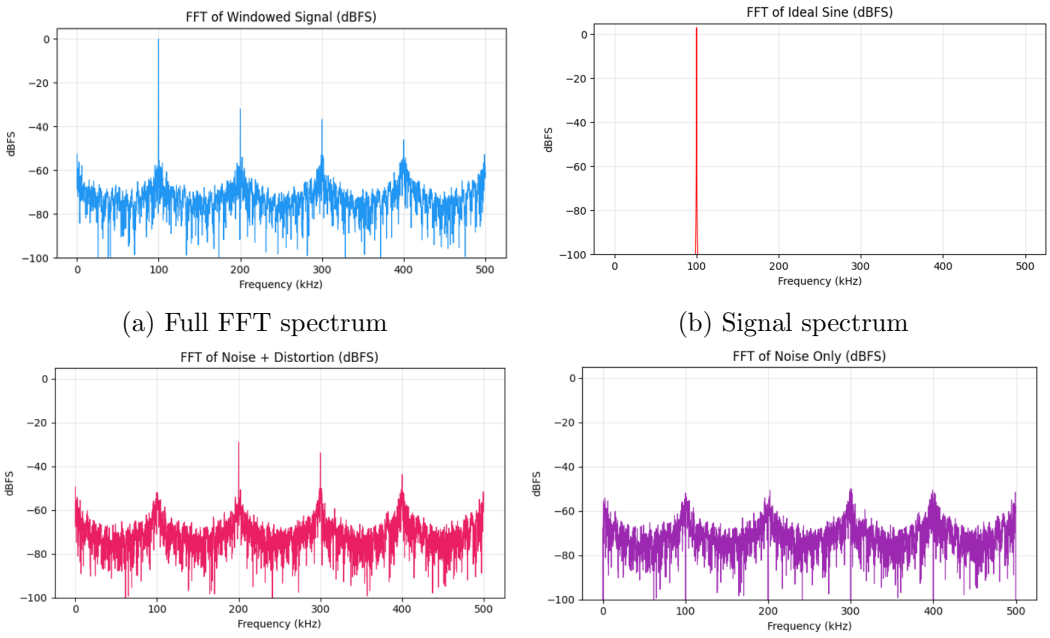
$$A = \frac{-2|G(k_{max})|}{N \cdot C_G} \cdot \frac{\pi(f - k_{max})}{\sin(\pi(f - k_{max}))} \cdot (f - k_{max} - 1)(f - k_{max} + 1) \tag{17}$$

$$P_o = \frac{1}{2\pi} \arg\left(G(k_{max})\, e^{-j\pi(f-k_{max})}\right) \tag{18}$$

where G is the complex FFT spectrum, N is the number of samples, and $C_G = \frac{1}{N}\sum_{n=0}^{N-1} w(n)$ is the coherent gain of the Hanning window. The code below shows the implementation of this signal estimation in Python:

```python
def tabei_ueda(fft_full, fft_mag, N, hann_window):
    # Returns (A, fx, Po) - amplitude (V), fractional bin, phase (rad).
    fft_pos = fft_mag[:N//2]
    kmax    = np.argmax(fft_pos[1:]) + 1
    # find the next highest bin and calculate r and f
    if fft_mag[kmax - 1] > fft_mag[kmax + 1]:
        r  = fft_mag[kmax - 1] / fft_mag[kmax]
        fx = kmax + (1.0 - 2.0*r) / (1.0 + r)
    else:
        r  = fft_mag[kmax + 1] / fft_mag[kmax]
        fx = kmax - (1.0 - 2.0*r) / (1.0 + r)
    dx   = fx - kmax
    dy   = np.pi * dx
    sinc = (dy / np.sin(dy)) if abs(dx) > 1e-10 else 1.0

    # coherent gain
    CG   = np.sum(hann_window) / N
    # amplitude
    A    = (-fft_mag[kmax] / N) * sinc * (dx - 1.0) * (dx + 1.0) / CG * 2.0
    # phase
    Po   = (1 / (2*np.pi)) * np.angle(fft_full[kmax] * np.exp(-1j * dy))
    return A, fx, Po
```



(a) Full FFT spectrum

(b) Signal spectrum

(c) Noise and distortion spectrum

(d) Noise only spectrum

Figure 17: Separated signal, noise, and distortion spectrum

Once the fundamental tone is estimated using the Tabei & Ueda method, the signal, noise, and noise + distortion spectrum can be created. The noise and distortion spectrum, Figure 17c, is calculated by subtracting the estimated tone, Figure 17b, from the full FFT, Figure 17a. The multiples of the index of the fundamental tone can then be subtracted from the noise + distortion spectrum to calculate the noise only spectrum, Figure 17d.

Now that the signal, noise, and distortion are separated from each other, all the fundamental dynamic characteristics and figures of merit can be calculated using the equations from Section 2.1.

4.3 Static Characterization Methods

Similar to the dynamic tests, the static characterization tests require coherently sampled waveforms. However, instead of a high frequency sine wave, the static tests use a low frequency ramp signal, shown in Figure 18, to determine nonlinearities in the ADC code transition levels. The histogram method,

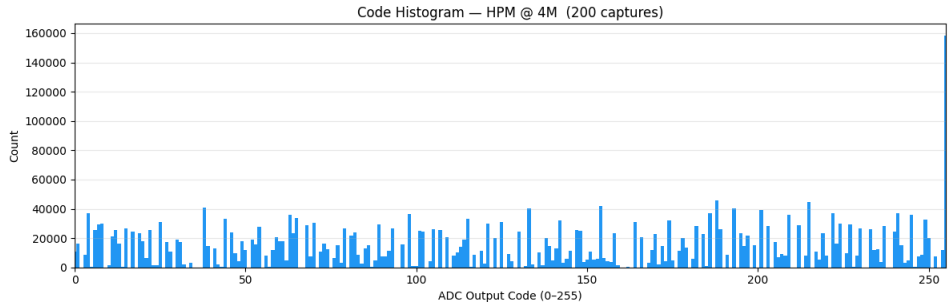


Figure 19: Histogram of low frequency ramp signal

described in [8], was used in this analysis.

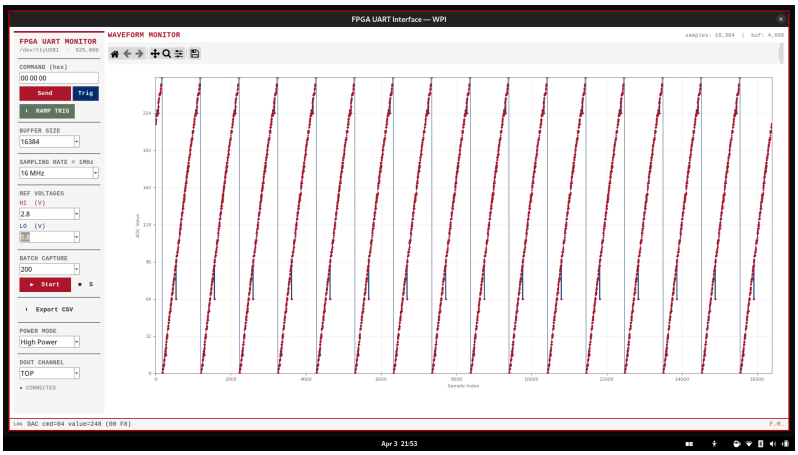


Figure 18: Low frequency ramp signal capture

To use this method efficiently, it is important to have a large amount of data samples. Two hundred epochs of 16384 each samples each were collected, over 3 million data points. Creating a histogram, Figure 19, of each code step will show insight into the linearity and missing code steps of the ADC.

From this histogram, the Code Transition Level (CTL) function can be calculated. The CTL function maps each ADC output code to its corresponding input voltage, revealing nonlinearities and missing codes across its input range. The CTL function takes the same form as a linear equation $y = C + Gx$, where $C = V_{\text{REF-LOW}}$ is the y-intercept and $G = (V_{\text{REF-HIGH}} - V_{\text{REF-LOW}}) \,/\, M$ is the slope. The independent variable x is replaced by the cumulative histogram sum $\sum_{i=0}^{k-1} H[i]$, which counts the total number of samples that fell at

or below code $k-1$. Since the ramp sweeps the input range linearly, this cumulative count is proportional to the corresponding input voltage. The measured CTL function is therefore:

$$T[k] = C + G \sum_{i=0}^{k-1} H[i], \quad \text{for } k = 1, 2, \ldots, (2^N - 1) \tag{19}$$

Additionally, since the ramp input range slightly exceeds the ADC reference voltage span, the outermost transition levels $T[1]$ and $T[2^N - 1]$ are discarded and clamped to $V_{\text{REF-LOW}}$ and $V_{\text{REF-HIGH}}$, respectively [3].

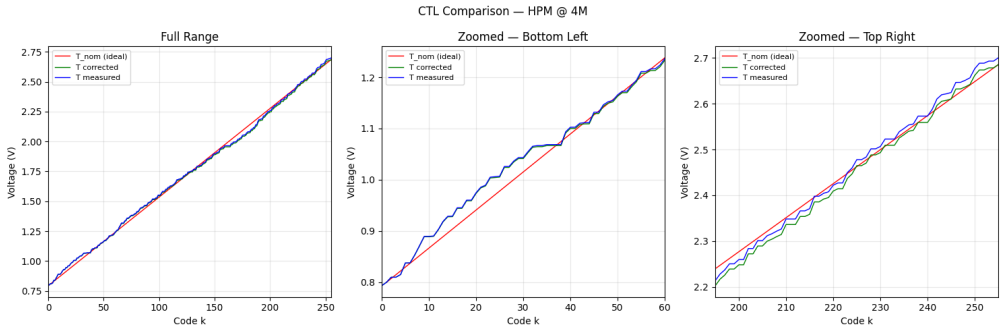


Figure 20: Terminal based CTL transfer function correcting T[k] based on the first and last CTL

With the CTL function established, the terminal-based gain and offset of the ADC can be extracted. Terminal based gain and offset was used rather than independently based gain and offset because the Flash ADC topology generally is less accurate near the reference voltages rather than the total RMS of the CTL function. The gain and offset are calculated as follows:

$$A = \frac{\left(2^N - 2\right) Q}{T[2^N - 1] - T[1]} \tag{20}$$

$$V_{OS} = T_1 - A \cdot T[1] \tag{21}$$

where Q is the ideal code bin width, $\frac{V_{FSR}}{2^N}$.

To calculate INL and DNL, the residual error corresponding to the k$^{\text{th}}$ code transition, $\varepsilon[k]$, is calculated with Equation 22.

$$\varepsilon[k] = \frac{A \cdot T[k] + V_{OS} - T_{\text{ideal}}[k]}{Q} \tag{22}$$

Finally the INL and DNL can be calculated using the following equations with their results shown in Figure 21.

$$INL[k] = \frac{\varepsilon[k]}{2^N Q} = \frac{\varepsilon[k]}{V_{FS}} \tag{23}$$

$$DNL[k] = \frac{W[k] - Q}{Q} \tag{24}$$

where $W[k] = G \times (T[k+1] - T[k])$ is the measured code bin width.

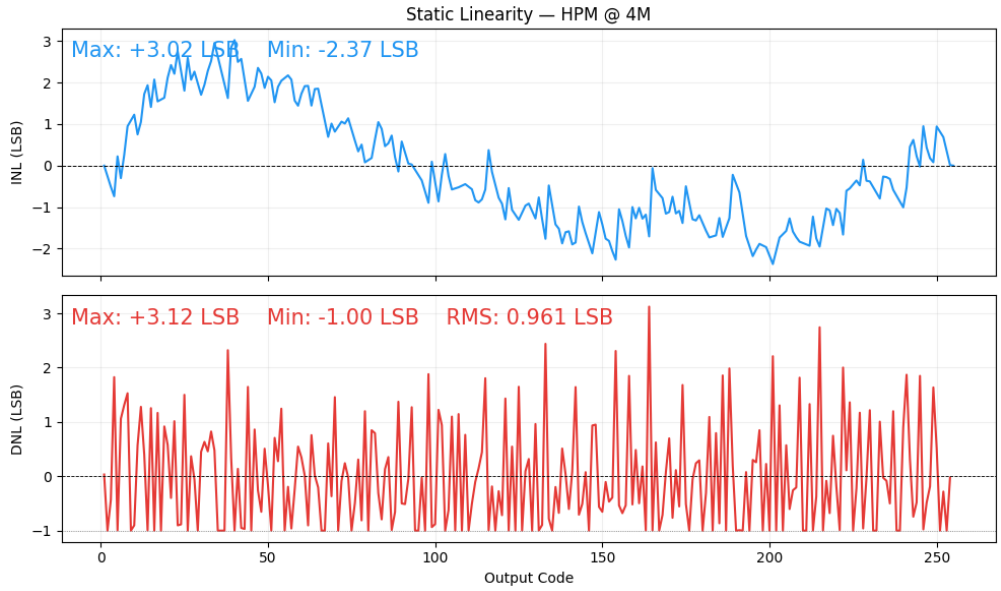


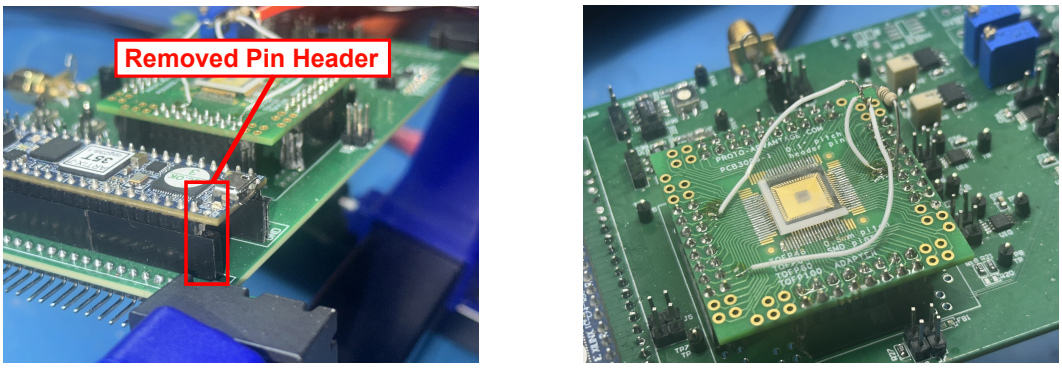
Figure 21: Measured INL (+2.86/-4.83) and DNL (+2.76/-1.00)

4.4 Power Analysis

Initial power measurements were taken by placing a benchtop multimeter in series with the power supply and the PCB, measuring the current through the multimeter. This approach quickly proved inadequate, as the measurement captured the combined power consumption of the FPGA, DACs, and other on-

board peripherals rather than the ADC alone.

To isolate the ADC's power consumption, two reworks were made to the PCB. First, the VU power pin was removed from the FPGA board so that the chip is powered exclusively by the low-noise supply. Second, a shunt resistor was inserted between the 3.3 V supply and the chip's power pins on the COB module, as shown in Figure 22.



(a) Removed VU pin rework. (b) 1.02 Ω shunt resistor rework.

Figure 22: PCB reworks for power measurement.

The shunt resistance was measured at 1.02 Ω using four-wire mode on the benchtop multimeter. During operation, the voltage drop across the shunt is measured and used to compute the current draw, giving the power consumed from the 3.3 V supply.

However, there is one additional source of power that must be accounted for: the static current drawn through the internal resistor ladder between V_{refhi} and V_{reflo}. The reference voltages are set by the DACs on the PCB; for all tests $V_{refhi} = 2.8\,\text{V}$ and $V_{reflo} = 0.8\,\text{V}$, giving a 2 V drop across the ladder. The resistance between the $V_{ref-high}$ and $V_{ref-low}$ pins was measured at 6.1 kΩ using the same four-wire method. Since the two ADCs share the reference voltages in parallel, each ADC's ladder network presents a resistance of approximately 12.2 kΩ. The resulting static power draw per channel is then:

$$P_{ref} = \frac{V^2}{R} = \frac{(2)^2}{12.2\,\text{k}\Omega} \approx 327\,\mu\text{W} \tag{25}$$

This value is added to the shunt-measured supply current to arrive at the total per-channel power consumption. While this estimate does not account for kickback current or other secondary effects, it provides a reasonable approximation of the reference voltage power contribution during operation.

5 Results

This section presents the measured static and dynamic performance of the WCIS Flash ADC. Results are reported for both the top and bottom ADC instances and compared against three commercial devices and one recent academic design to provide context for the measured figures of merit.

5.1 Dynamic Characteristics

Figure 23 compares the ENOB of the top and bottom ADC instances across all sampling rates and power modes. The bottom ADC achieves slightly higher resolution in most operating conditions, though the difference was on the order of 0.1 to 0.2 bits which is unlikely to produce a visible difference in image quality between even and odd pixel columns. Since the test setup was not physically altered between measurements, this deviation is attributed to device mismatch on the die rather than test procedure error. Full dynamic characterization results for both instances are provided in Table 4 and 5 in Appendix 1.

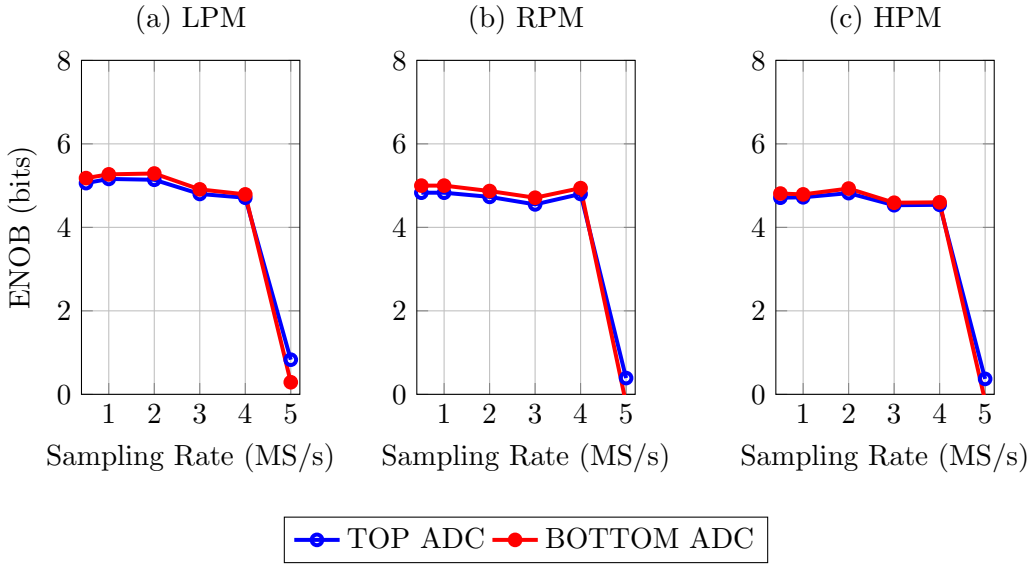


Figure 23: ENOB versus sampling rate comparison between TOP and BOTTOM ADCs for (a) LPM, (b) RPM, and (c) HPM.

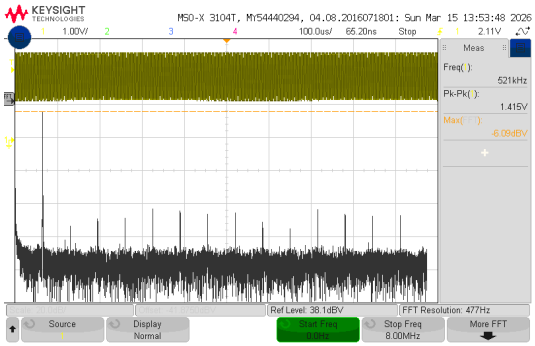


Figure 24: FFT of input waveform from function generator, showing harmonic content at integer multiples of the fundamental frequency.

The bottom FADC instance achieved its best performance at 4 MS/s in RPM, yielding an ENOB of 4.94 bits, a SINAD of 31.52 dB, and a power consumption of 342 µW. At lower sampling rates (0.5 MS/s to 3 MS/s), the ADC performs best in LPM, reaching a peak ENOB of 5.29 bits at 2 MS/s. At 5 MS/s, ENOB collapses to near zero across all power modes.

A consistent gap between SNR and SINAD is observed across all operating points. At 4 MS/s in RPM, the SNR is 37.17 dB while the SINAD is 31.52 dB, a difference of approximately 5.65 dB, with a measured THD of −32.9 dBFS. Figure 24 shows the FFT of the function generator output, confirming that harmonic content is present in the input signal prior to reaching the ADC.

5.2 Static Characteristics

The top ADC exhibited 48 missing codes with an INL of +3.02/−2.37 LSB and DNL of +3.12/−1.00 LSB, while the bottom ADC exhibited 42 missing codes with an INL of +2.68/−3.19 LSB and DNL of +3.15/−1.00 LSB, shown in Figures 25 and 26. Both instances measured a terminal gain of 0.9922 and an offset of 6.25 mV, indicating that the reference voltage endpoints are well matched to the ideal transfer function. The higher missing code count in the top instance is consistent with its slightly lower ENOB observed in the dynamic results, as missing codes reduce the effective number of usable output

levels.

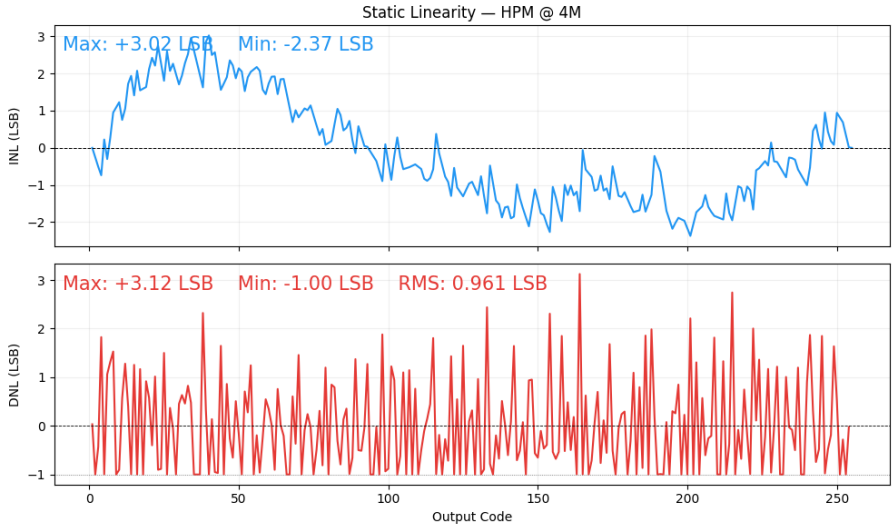


Figure 25: Measured INL and DNL for TOP ADC instance.

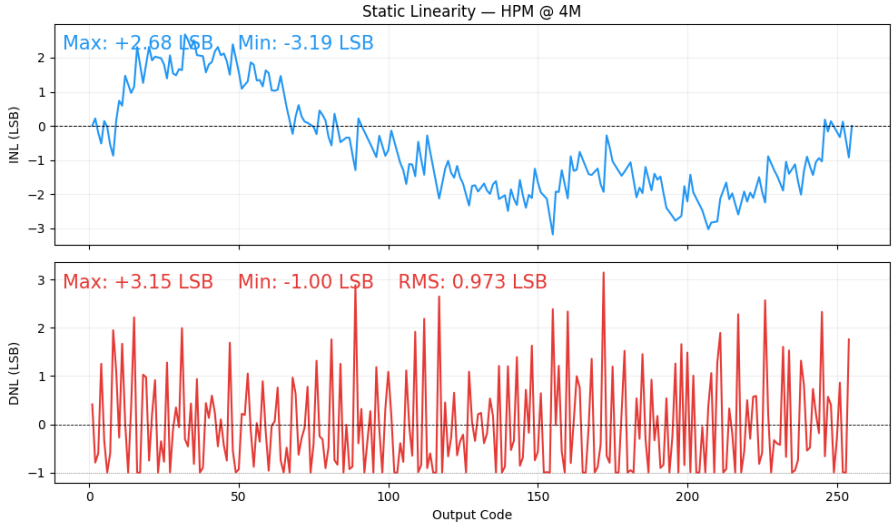


Figure 26: Measured INL and DNL for BOTTOM ADC instance.

5.3 Comparison

Table 3 compares the WCIS Flash ADC against three commercial parts and one recent academic design. The commercial devices, the MAX153 [9], AD9057 [10], and CA3318 [11], are well-established designs targeting video and communications applications, while Lotfi et al. [12] represent a state-of-

Table 3: ADC Comparison: WCIS Flash ADC vs. Commercial and Academic Parts

Parameter	**Units**	**WCIS FADC**	**[9]**	**[10]**	**[11]**	**[12]**
Resolution	Bits	8	8	8	8	5
Application	—	Video	Telecom	Video / Comms	Video	—
Year	—	2025	1994	2003	1992	2023
Topology	—	Flash, 180 nm	Half-Flash	Flash	Flash	Flash, 22 nm
f_{in}	kHz	100	195.8	10,000	100	4,700
f_s	MS/s	4	1	40	15	18,500
ENOB	Bits	4.94	7.7	6.7	7.2	3.9
SNR	dB	37.17	49.1	46.5	47.0	31.67
SINAD	dB	31.52	45.0	42.0	45.0	25.0
THD	dB	−32.9	−50.0	—	−46.0	−30.58
DNL	LSB	+3.15 / −1.0	+1.0 / −1.0	+2.0 / −1.0	+1.0 / −0.8	+1.5 / −0.95
INL	LSB	+2.68 / −3.19	+1.0 / −1.0	+2.0 / −2.0	+1.5 / −1.5	+0.76 / −0.95
Power	mW	0.342	40	260	150	3.2
FOM_W	pJ/step	6.00	192.37	62.52	68.01	0.01
FOM_S (SINAD)	dB	129.2	115.97	120.86	121.99	149.61

the-art 5-bit 22 nm Flash ADC operating at 18.5 GS/s. The WCIS achieves the lowest power consumption of any device in the table at 0.35 mW, roughly two orders of magnitude below the commercial parts. However, the ENOB of 4.1 bits is the lowest among the 8-bit designs, falling short of the 6.7 to 7.7 bit range achieved by the commercial parts. The DNL of $+3.15/-1.00$ LSB is comparable to the AD9057 ($+2.0/-2.0$ LSB) but exceeds the ± 1.0 LSB of the MAX153 and CA3318.

5.4 Figure of Merit

Despite its limited resolution, the WCIS ADC achieves a Walden FOM of 6.00 pJ/step, which compares well against the commercial parts in Table 3. The commercial Flash ADCs were designed in the early 1990s and 2000s and consume significantly more power, ranging from 40 mW to 260 mW, resulting in Walden FOMs of 62 pJ/step to 192 pJ/step. The low power consumption of

the WCIS FADC, at only 342 µW, gives it a competitive energy per conversion step despite the low ENOB.

The 2023 design by Lotfi et al. achieves a Walden FOM of 0.01 pJ/step, surpassing all other entries in the table. This is expected, as modern processes enable dramatically lower power and higher speed compared to the older designs. Because the Walden FOM rewards high sampling rate and low power regardless of resolution, it is less informative when comparing ADCs across different technology nodes and performance tiers.

The Schreier FOM accounts for the achievable signal bandwidth relative to power consumption and is therefore better suited for cross-topology and cross-generation comparisons. Under this metric, the WCIS ADC scores 129.2 dB, outperforming the commercial parts (116 dB to 122 dB) and consistent with the efficiency level of early 2000s designs. Lotfi et al. scores 149.61 dB, reflecting the substantial efficiency gains afforded by modern process technology and circuit design expertise [12]. The WCIS result confirms that while the ADC's resolution is limited, its overall power efficiency is comparable to commercial Flash ADCs of the same technology generation.

6 Discussion

The WCIS Flash ADC demonstrates competitive power efficiency relative to commercial designs of similar technology, achieving a Walden FOM of 6.00 pJ/step at 4 MS/s, well below the 62 pJ/step to 192 pJ/step range of the commercial parts surveyed in Table 3. While the measured ENOB of 4.94 bits falls short of the nominal 8-bit resolution, this result is consistent with expectations for a first-generation Flash ADC in a 180 nm process without foreground or background calibration. The WCIS represents the first exposure to microelectronic circuit design for its undergraduate designers, and the fact that it operates correctly and produces characterizable results is itself a meaningful outcome.

At lower sampling rates (0.5 MS/s to 3 MS/s), the ADC performs better in LPM than HPM. At reduced sampling rates, comparator regeneration time is sufficient even at lower bias currents, meaning the speed advantage of HPM goes unused. Instead, the dominant performance limiter becomes thermal noise. Since LPM reduces the current through the comparator input pair, it directly reduces the thermal noise, improving resolution. HPM's higher bias current increases switching speed but at the cost of greater thermal noise, which explains why it underperforms LPM at sampling rates where speed is not the bottleneck.

The sudden collapse in ENOB at 5 MS/s is caused by the ADC operating beyond its design limits. As shown in Figure 6, the S/H output requires over 100 ns to fully acquire and settle to the input voltage after the hold phase ends. At 4 MS/s the available sampling window of approximately 80 ns is sufficient for marginal operation, but the further reduction to 60 ns at 5 MS/s pushes both the S/H acquisition time and the control signal timing margins past their limits simultaneously, resulting in the observed performance collapse.

The primary source of static nonlinearity is comparator offset mismatch across the 255-comparator array. Each comparator has a slightly different input-referred offset voltage due to transistor mismatch introduced during fabrication. When the offset of a comparator exceeds one LSB, its transition level overlaps with an adjacent comparator, causing the corresponding output code to become inaccessible and producing a missing code. Resistor ladder mismatch is a secondary contributor, introducing non-uniform step widths across the reference voltage range. In fact, in production it is common for manufacturers to physically trim the size of the resistors after fabrication to account for linearity errors. That is likely how they achieved such a high ENOB with the Flash ADC topology.

The SNR/SINAD gap observed across all operating points is partly due to harmonic distortion in the input signal chain rather than the ADC itself, as confirmed by the FFT of the function generator output in Figure 24. Inserting a bandpass filter centered at F_{in} between the function generator and the ADC input would attenuate these harmonics before they reach the device, producing a cleaner stimulus and more accurate dynamic measurements. An additional improvement to the test setup would address ground loop noise on the PCB, which contributes to the noise floor.

There are many more advanced techniques that could have improved the performance of the ADC and will be tested in future research. For example, foreground calibration is a common technique to improve non-linearity errors in Flash ADCs. [13] presents a look-up-table (LUT) based calibration to reduce the effects of comparator offset mismatch. Additionally, the dual-instance architecture also enables low-cost jitter characterization, as the shared clock path between the two ADCs can be exploited to separate aperture jitter from clock jitter and obtain accurate SNR and ENOB estimates [14]. Additionally, some long term goals include characterizing the remaining modules on the WCIS, like the encryption module and an in depth analysis of the imager.

7 Conclusion

This thesis presented the design, implementation, and characterization of an 8-bit Flash ADC embedded in the Worcester CMOS Image Sensor, fabricated in a 180 nm CMOS process. Static characterization via the code-density histogram method yielded a peak DNL of $+3.15/-1.00$ LSB and INL of $+2.68/-3.19$ LSB, with 42 missing codes attributed to comparator offset mismatch across the resistor-comparator ladder. Dynamic characterization at 4 MS/s in RPM produced an ENOB of 4.94 bits and a SINAD of 31.52 dB, falling short of the nominal 8-bit resolution primarily due to comparator offset and harmonic distortion in the input signal chain. Despite these limitations, the ADC achieved a Walden FOM of 6.00 pJ/step and a Schreier FOM of 129.2 dB, demonstrating power efficiency that outperforms commercial Flash ADCs of the same technology generation.

Equally important to the ADC results, a reusable testing framework was developed for the WPI CHIPS Design Studio. The Python GUI and carrier PCB will be iterated upon continuously, becoming more capable with each revision. By demonstrating that the undergraduate designed Worcester CMOS Image Sensor operates correctly as a characterizable mixed-signal circuit, future experiments can explore other parts of the chip like the pixel array and encryption engine along with more advanced calibration and testing techniques. Backed by funding from organizations like CEMiD and WPI's expanding microelectronics program, this framework is built to support the next generation of chip design and hardware characterization courses.

Appendices

1 ADC Dynamic Results

Table 4: Dynamic characterization results for TOP configuration.

Power Mode	F_s (MSps)	ENOB (bits)	SNR (dB)	SINAD (dB)	THD (dBFS)	SFDR (dBFS)	DR (dBFS)	Power (μW)	FOM$_W$ (pJ/step)	FOM$_S$ (DR, dB)	FOM$_S$ (SINAD, dB)
HPM	0.5	4.71	36.33	30.13	-31.33	32.82	33.14	338	25.8	121.8	118.8
	1	4.72	36.09	30.15	-31.43	32.97	33.16	340	12.9	124.8	121.8
	2	4.82	35.96	30.80	-32.37	35.06	33.81	343	6.1	128.4	125.4
	3	4.53	35.01	29.03	-30.29	32.21	32.04	346	5.0	128.4	125.4
	4	4.54	35.12	29.11	-30.36	34.65	32.12	350	3.8	129.7	126.7
	5	0.37	4.57	3.98	-12.9	18.37	6.99	354	54.9	105.5	102.5
RPM	0.5	4.83	36.83	30.85	-32.11	32.74	33.86	333	23.4	122.6	119.6
	1	4.83	36.65	30.87	-32.2	32.89	33.88	334	11.7	125.6	122.6
	2	4.73	36.09	30.25	-31.56	32.74	33.26	337	6.3	128.0	125.0
	3	4.55	35.27	29.12	-30.33	31.9	32.13	339	4.8	128.6	125.6
	4	4.8	36.47	30.64	-31.95	36.62	33.65	342	3.1	131.3	128.3
	5	0.39	4.61	4.13	-13.89	17.56	7.14	345	52.5	105.7	102.7
LPM	0.5	5.06	37.7	32.23	-33.68	34.11	35.24	331	19.8	124.0	121.0
	1	5.16	37.63	32.84	-34.59	35.37	35.85	332	9.3	127.6	124.6
	2	5.14	36.7	32.73	-34.95	36.11	35.74	334	4.7	130.5	127.5
	3	4.8	36.21	30.67	-32.1	33.48	33.68	336	4.0	130.2	127.2
	4	4.71	34.09	30.12	-32.34	35.2	33.13	337	3.2	130.9	127.8
	5	0.83	7.66	6.75	-13.99	16.51	9.76	340	38.3	108.4	105.4

Table 5: Dynamic characterization results for BOTTOM configuration.

Power Mode	F_s (MSps)	ENOB (bits)	SNR (dB)	SINAD (dB)	THD (dBFS)	SFDR (dBFS)	DR (dBFS)	Power (μW)	FOM$_W$ (pJ/step)	FOM$_S$ (DR, dB)	FOM$_S$ (SINAD, dB)
HPM	0.5	4.81	37.14	30.69	-31.81	33.72	33.7	338	24.2	122.4	119.4
	1	4.79	36.88	30.61	-31.78	33.75	33.62	340	12.3	125.3	122.3
	2	4.93	37.29	31.43	-32.73	36.13	34.44	343	5.6	129.1	126.1
	3	4.59	35.75	29.4	-30.54	32.92	32.41	346	4.8	128.8	125.8
	4	4.6	35.16	29.48	-30.85	35.2	32.49	350	3.6	130.1	127.0
	5	-0.16	0.94	0.79	-15.54	16.58	3.8	354	79.2	102.3	99.3
RPM	0.5	5	38.11	31.84	-33.01	33.59	34.85	333	20.9	123.6	120.6
	1	5	38.25	31.88	-33.01	33.83	34.89	334	10.4	126.6	123.6
	2	4.87	37.74	31.1	-32.16	33.22	34.11	337	5.7	128.8	125.8
	3	4.71	36.22	30.11	-31.32	33.04	33.12	339	4.3	129.6	126.6
	4	4.94	37.17	31.52	-32.9	37.79	34.53	342	2.8	132.2	129.2
	5	-0.22	0.66	0.46	-13.9	17.01	3.47	345	80.2	102.1	99.1
LPM	0.5	5.18	39.53	32.93	-34	34.71	35.94	331	18.3	124.7	121.7
	1	5.27	39.23	33.5	-34.86	35.72	36.51	332	8.6	128.3	125.3
	2	5.29	39.28	33.63	-35.01	36.7	36.64	334	4.3	131.4	128.4
	3	4.91	37.36	31.32	-32.57	34.14	34.33	336	3.7	130.8	127.8
	4	4.79	34.66	30.6	-32.77	36.14	33.61	337	3.0	131.3	128.3
	5	0.29	3.79	3.53	-15.85	21.24	6.54	340	55.5	105.2	102.2

2 WCIS Carrier Board Schematics

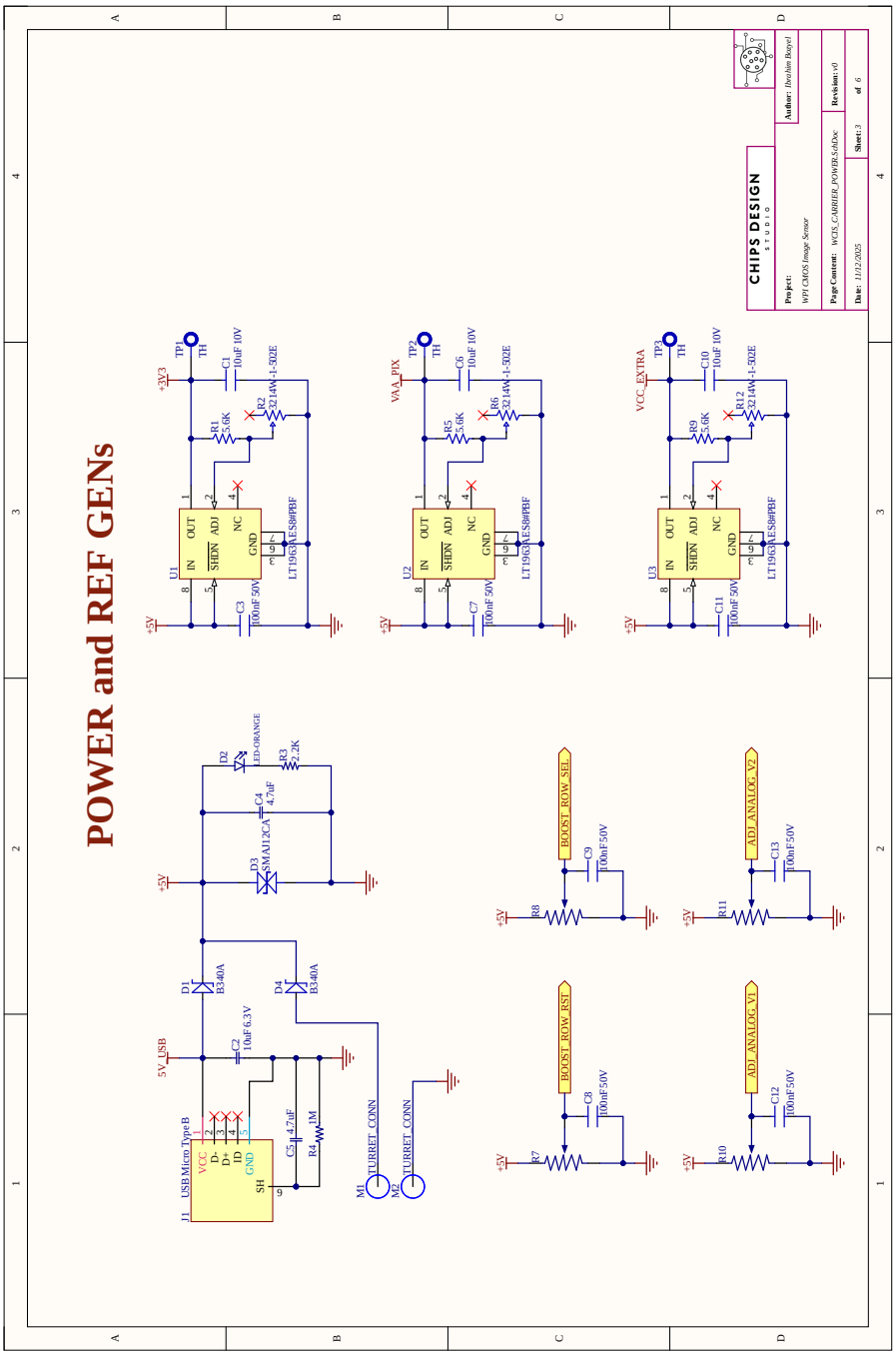


Figure 27: WCIS carrier board power schematic

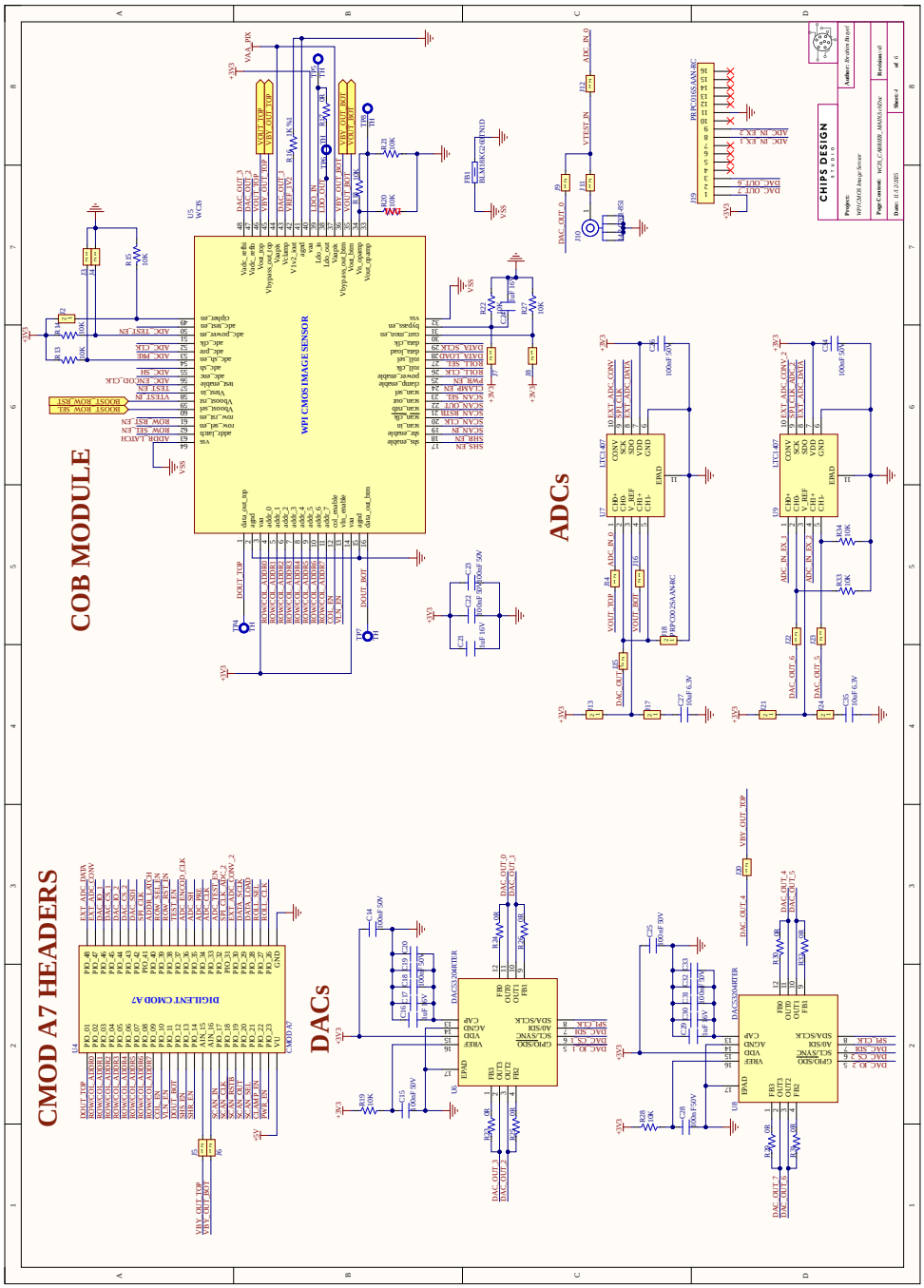


Figure 28: WCIS carrier board main schematic

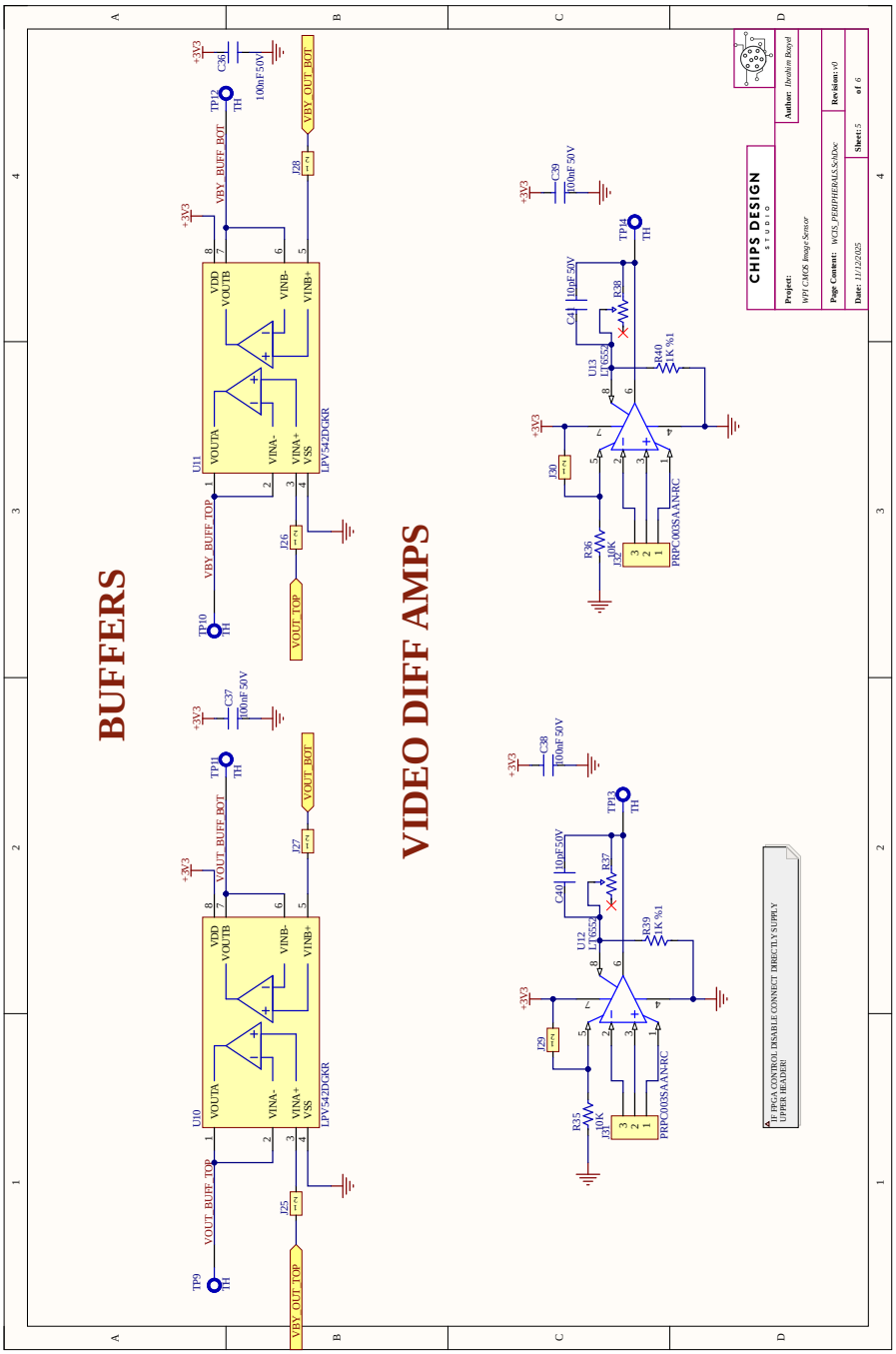


Figure 29: WCIS carrier board peripherals schematic

Bibliography

[1] H. Liu, F. Marrocco, K. Liu, Y. Xue, and P. Miu, "Development of a secure CMOS image sensor," Worcester Polytechnic Institute, Tech. Rep., 2025. [Online]. Available: https://digital.wpi.edu/concern/student_works/mk61rn717?locale=en.

[2] F. Morishita, M. Otsuka, and W. Saito, "An ADC Test Technique With Dual-Path/Multi-Functional Fine Pattern Generator Realizing High Accuracy Measurement for CMOS Image Sensor," in *2020 IEEE 29th Asian Test Symposium (ATS)*, Penang, Malaysia: IEEE, Nov. 2020, pp. 1–6. DOI: 10.1109/ats49688.2020.9301531. Accessed: Jul. 17, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9301531/.

[3] *IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters*, Piscataway, NJ, USA. DOI: 10.1109/ieeestd.2023.10269815. Accessed: Jul. 26, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10269815/.

[4] B. E. Jonsson, "An empirical approach to finding energy efficient ADC architectures," in *Proc. 2011 IMEKO IWADC & IEEE ADC Forum*, Orvieto, Italy, Jun. 2011. [Online]. Available: https://www.imeko.org/publications/iwadc-2011/IMEKO-IWADC-2011-28.pdf.

[5] B. Razavi, "The StrongARM latch," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12–17, 2015. DOI: 10.1109/MSSC.2015.2418155.

[6] *Tektronix AFG3021B Datasheet*, Datasheet. [Online]. Available: https://www.testequipmenthq.com/datasheets/TEKTRONIX-AFG3021B-Datasheet.pdf.

[7] H. Okawara, "Spectrum estimation," Verigy Japan, Tech. Rep. 12, Apr. 2009. [Online]. Available: https://www3.advantest.com/documents/11348/f641a733-fcbe-43e8-b020-7799cd4c82bc.

[8] H. Okawara, "Histogram method in adc linearity test," Verigy Japan, Tech. Rep. 18, Sep. 2009. [Online]. Available: https://www3.advantest.com/documents/11348/27fd03db-3c5d-49e7-afb9-e0bcb6861cee.

[9] Maxim Integrated, *MAX153 1Msps, μP-compatible, 8-bit ADC with 1μA power-down*, Rev. 2, Maxim Integrated Products, Inc., 2012. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX153.pdf.

[10] Analog Devices, *AD9057 8-bit 40 MSPS/60 MSPS/80 MSPS A/D converter*, Rev. D, Analog Devices, Inc., 2003. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD9057.pdf.

[11] Intersil Corporation, *CA3318 CMOS video speed, 8-bit, flash A/D converter*, File Number 3103.1, Intersil Corporation, 1997. [Online]. Available: https://www.alldatasheet.com/datasheet-pdf/download/66360/INTERSIL/CA3318.html.

[12] N. Lotfi, P. Scholz, and F. Gerfers, "The Fastest CMOS Single-Channel 5-bit Flash ADC Operating at 18.5 GS/s in 22 nm FD-SOI," in *2023 18th European Microwave Integrated Circuits Conference (EuMIC)*, Berlin, Germany: IEEE, Sep. 2023, pp. 121–124, ISBN: 978-2-87487-073-6. DOI: 10.23919/EuMIC58042.2023.10289098. Accessed: Apr. 9, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/10289098/.

[13] S. Chatterjee, M. Daimari, and S. Roy, "A fully digital foreground calibration technique of a flash adc," in *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2021, pp. 1–6. DOI: 10.1109/ISVLSI51109.2021.00012.

[14] S. K. Chaganti, L. Xu, and D. Chen, "A low-cost jitter separation and adc spectral testing method without requiring coherent sampling," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–5. DOI: 10.1109/ISCAS.2018.8351210.