11-Bit 200MSps Pipeline Analog-to-Digital Converter

ECEN 607 - Advanced Analog Circuit Design Techniques
Final Project Report
Spring 2024

Leo Predanic
Alex Anderson
Mitchell Clark

Table of Contents

1. Motivation and Project Overview..... 3
2. Literature Review..... 3
3. Architecture..... 4
4. Simulation Results..... 11
5. Conclusion..... 17
6. References..... 18
7. Appendix..... 20

1. Motivation and Project Overview

As data rates continue to grow in communication and serial bus applications, high resolution and high sample rate analog to digital converters (ADCs) are a necessity. Pipeline ADCs fall nicely into this category compared to other ADC architectures. While Flash ADCs are competitive in speed, their bit precision requires 2^N comparators as opposed to the pipeline ADCs' linearly scaling need for comparators. On the other end, SAR ADCs are similar in bit resolution to pipeline ADCs, but require an immensely faster clock to resolve N-times as many cycles as pipeline ADCs to match in speed. SAR ADCs are also very sensitive to comparator parasitics and offsets. As a result, Pipeline ADCs have maintained their dominance in medium to high-speed, high-resolution applications. Several variations on Flash and SAR ADCs have been implemented using pipeline techniques (i.e. Pipeline-SAR) to optimize their performance in Flash and SAR-dominated applications. This project consists of a complete implementation of a Pipeline ADC with the performance goals of 11-bit resolution at 200MSps, consuming less than 10mW of power.

2. Literature Review

Reference [2] implements the ADC using a 1.5-bit/stage architecture. This is done mainly to take advantage of power reduction using low-voltage techniques as GBW is compromised, a stringent specification for extracting a large number of bits per stage. One challenge with the reduced supply constraint is the linearity of the amplifier. A cascode compensated (Ahuja) with a Class A output stage is used to increase the linear range of the amplifier. A highly linear bias generator is also employed. Since pipeline stages with bit redundancy are used, comparator offset requirements are relaxed and a dynamic comparator is used to minimize power.

Since amplifiers consume the most power in a pipeline ADC, [4] uses a dual supply amplifier to reduce power. The low supply domain amplifier is optimized for power whereas the high supply domain amplifier is optimized for gain and bandwidth. It also takes advantage of partial positive feedback to boost the gain. Within the pipeline stage, a dithering circuit is added to improve performance. An additional comparator sampling capacitor is used to reduce charge redistribution time and relax the MDAC settling time requirement.

Other implementations have tried to solve the linearity issues in alternating ways, [6] uses an open-loop amplifier to reduce the gain and bandwidth requirement in the conventional feedback residue amplifier topologies, however the price is higher amplifier nonlinearity. As a result, a 3rd order nonlinearity cancellation calibration algorithm is employed. Additional post-processing is employed after the alignment and recombination logic to tune the calibration.

There are many other calibration methods for ADCs. Reference [8] implements a forecasting technique in the sub-ADCs of the pipeline cell and uses a Class C amplifier to boost performance

at lower power consumption. The forecasting technique adds an additional comparator in each cell which can perform an early detection of the MSB at that stage. This determines which comparators in the cell are connected to the input which reduces the capacitive loading on the amplifier and reduces dynamic power losses.

3. Architecture

The ADC is based on a 2.5-bit/stage design that uses a front-end SHA to buffer the input from the pipeline stages. Five stages are used to generate the 11-bit digital word and digital logic is used to align the 3-bit output from each stage before combining them. Non-overlapping clocks are used to prevent charge losses from capacitor switching between the sample and hold phases.

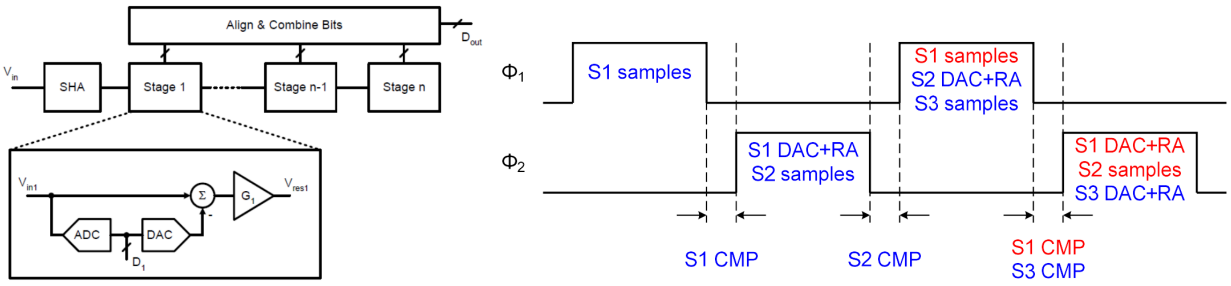

Figure 1: Pipeline ADC Architecture and Timing Diagram [14]

The SHA is implemented using a unity-gain flip-over topology and the 2.5-bit/stage cell is shown. A fully differential implementation of the cell and the transfer function is shown.

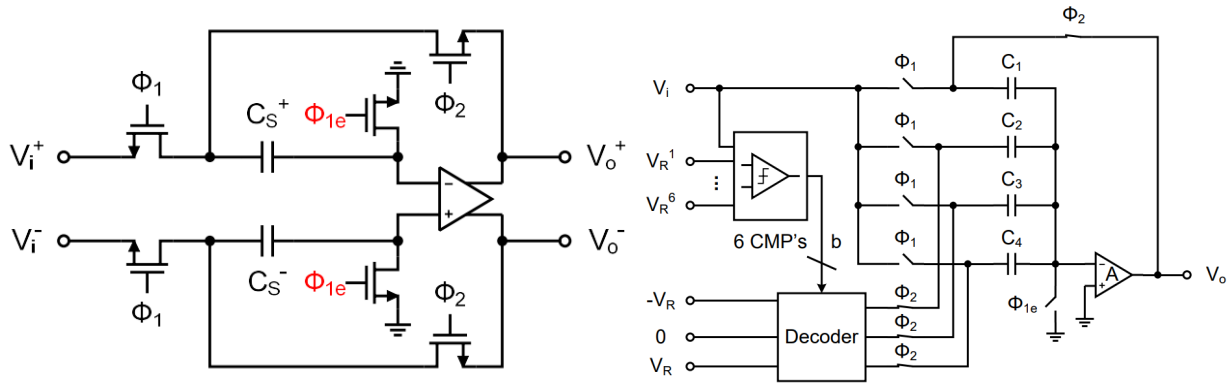

Figure 2: Flip-Over SHA [11] and 2.5-Bit/Stage Cell [14]

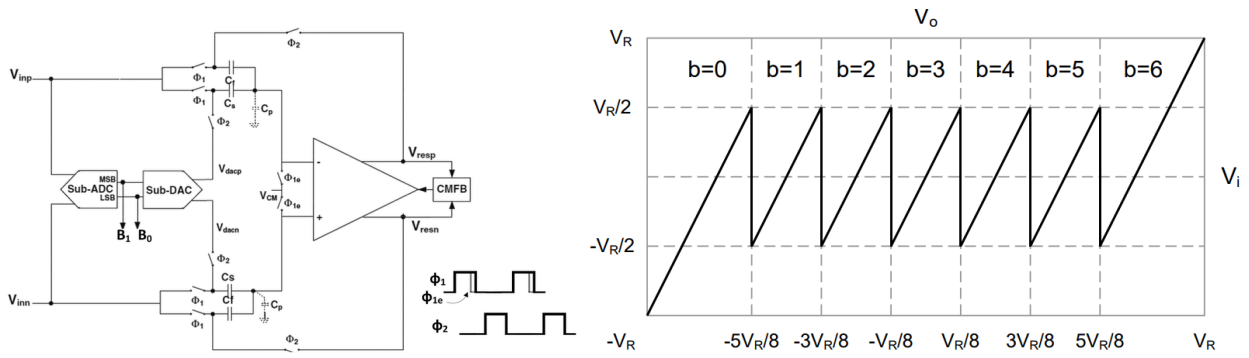

Figure 3: Simplified Fully-Differential Cell and 2.5-Bit/Stage Transfer Function [14]

The sub-DAC logic functions to properly subtract the reference voltage from the amplifier input once a specific comparator threshold is reached. The digital recombination logic uses full and half adders to take advantage of the redundancy of a 2.5-bit/stage implementation.

b	0	1	2	3	4	5	6
b-3	-3	-2	-1	0	+1	+2	+3
b_1	-1	-1	-1	0	+1	+1	+1
b_2	-1	-1	0	0	0	+1	+1
b_3	-1	0	0	0	0	0	+1
C_2	$+V_R$	$+V_R$	$+V_R$	0	$-V_R$	$-V_R$	$-V_R$
C_3	$+V_R$	$+V_R$	0	0	0	$-V_R$	$-V_R$
C_4	$+V_R$	0	0	0	0	0	$-V_R$

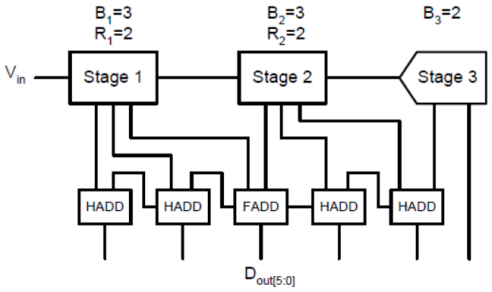

Figure 4: Sub-DAC Truth Table [14] and Recombination Logic Example [11]

A bootstrap switch improves linearity over a standard switch by adding a ‘floating’ battery between source and gate. This makes the voltage between the two terminals constant and hence, the triode resistance is constant across a large input voltage range. The bootstrap switch is best used at the input of the SHA. This way, the sampling capacitor follows the input signal very closely which helps to set how linear the remainder of the pipeline stage will be. The bootstrap switch was designed after [10]. This architecture works analogously to a typical NMOS and PMOS pass gate switch, with the addition of both devices being bootstrapped. This provides a significant linearity improvement over a single-ended bootstrap switch while still keeping the design relatively simple.

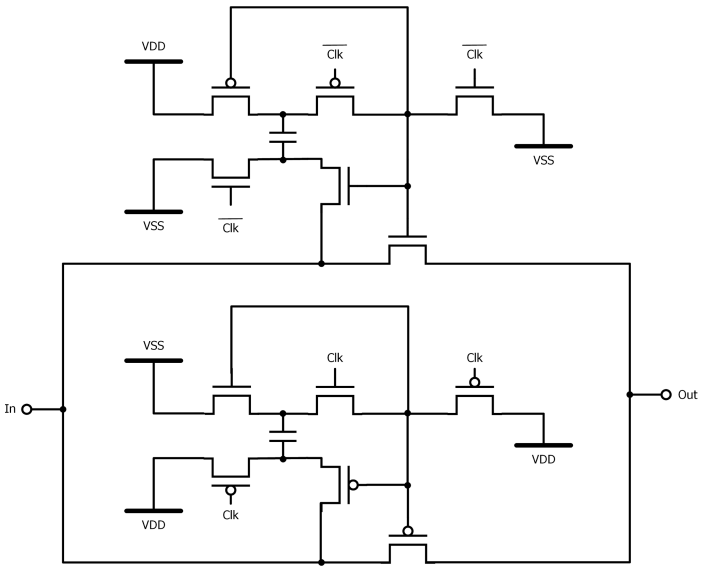

Figure 5: Bootstrap Switch

The sample and hold amplifier (SHA) is implemented as a recycling folded cascode (RFC) amplifier with gain-boosting. The gain boosting amplifier was implemented as a simple common-source pair with a current mirror PMOS load. RFC is advantageous for its high power efficiency, while still providing a high output swing and decent bandwidth. For use in the sample and hold cell, the amplifier must be fully-differential, thus a wideband CMFB engine is necessary to ensure the common-mode stays at the desired level. The use of gain-boosting allows for a higher DC gain, while minimally affecting the bandwidth and stability, provided that the gain-boosting amplifier's unity gain frequency is well positioned relative to the core amplifier [15].

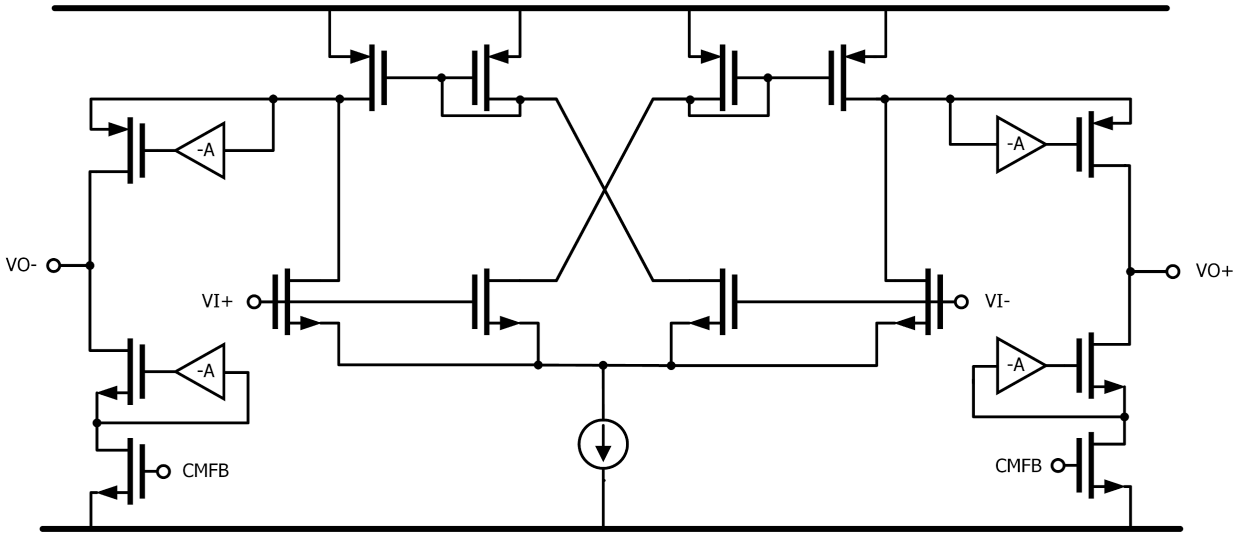

Figure 6: Recycling Folded Cascode (RFC) Amplifier with Gain-Boosting

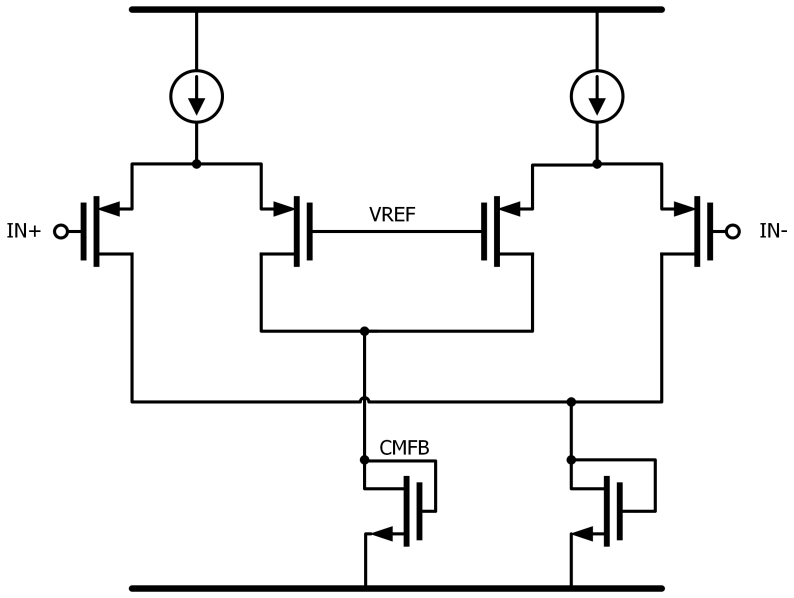

Figure 7: CMFB Engine

An RFC without gain-boosting [13] was implemented as the residue amplifier core with similar advantages as the RFC with gain-boosting. Since the SHA has a higher gain and a smaller bandwidth requirement, the RFC alone can increase the bandwidth in exchange for gain as the residue amplifier has a more stringent bandwidth requirement to meet the desired settling time for a given sampling rate. The linearity is slightly improved as gain-boosting requires more headroom.

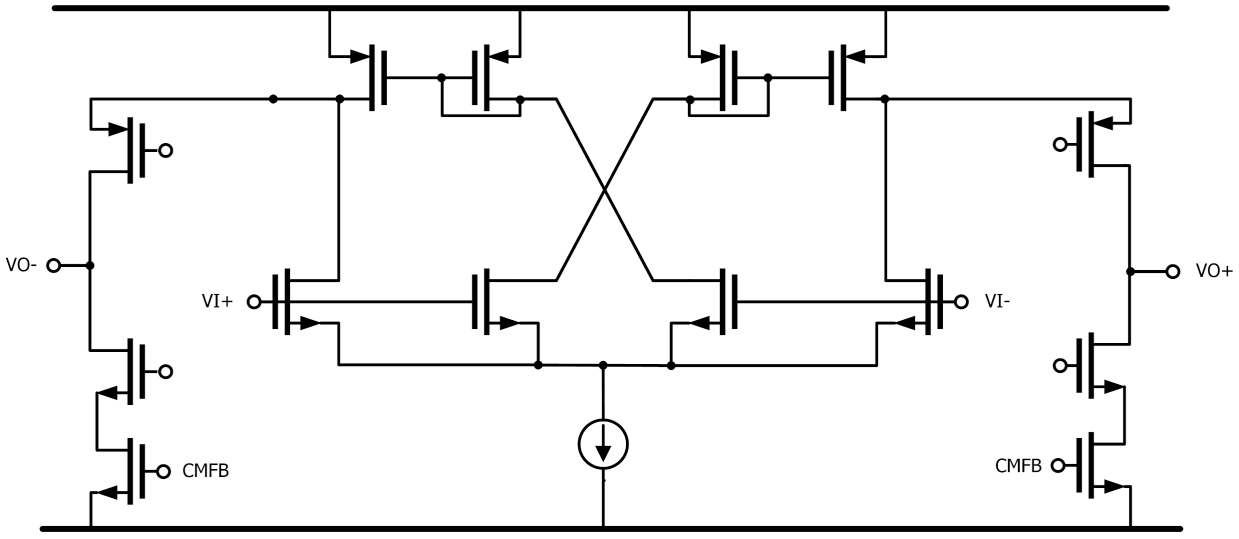

Figure 8: RFC Amplifier Core Used as a Residue Amplifier

A feedforward amplifier [13] was attempted as higher gain and bandwidth than the RFC can be achieved. The feedforward amplifier adds a ‘fast’ path which skips the ‘slow’ high gain main path. When designed properly, at high frequencies, the fast path provides moderate gain at higher bandwidth introducing a ‘phantom’ LHP zero in the transfer function. GBWs in 180nm processes in excess of 1.5GHz with DC gains of 60dB can be obtained using this technique. This significantly reduces the per-cell error due to nonidealities but comes at a higher power cost which doesn’t make this amplifier a worthwhile option.

The comparators used within the sub-ADC are based on a Strong-ARM latch design. This design is a compact and fast way to resolve the thermometer-encoded bits per stage. The Strong-ARM is preceded by a preamplifier to reduce the effect of kickback noise and offset. The Strong-ARM is followed by an RS latch to hold the decision while the Strong-ARM is reset, performing a flip-flop function.

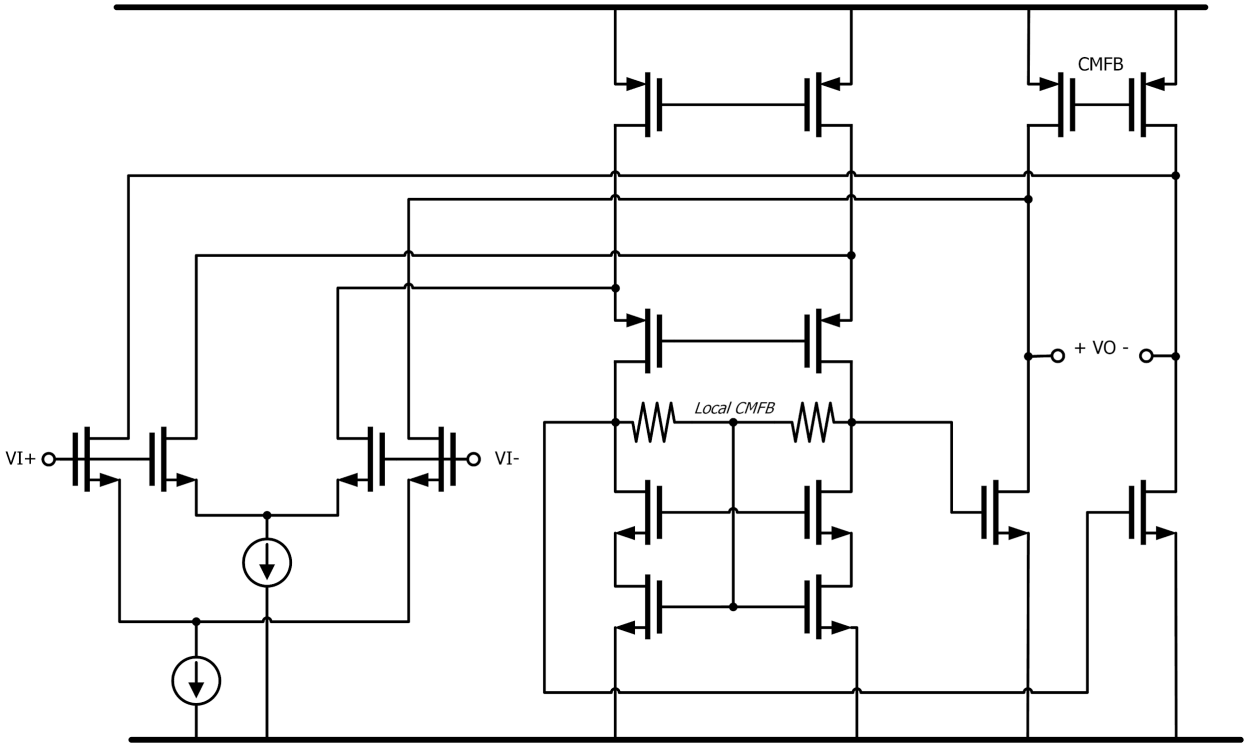

Figure 9: Feedforward Amplifier

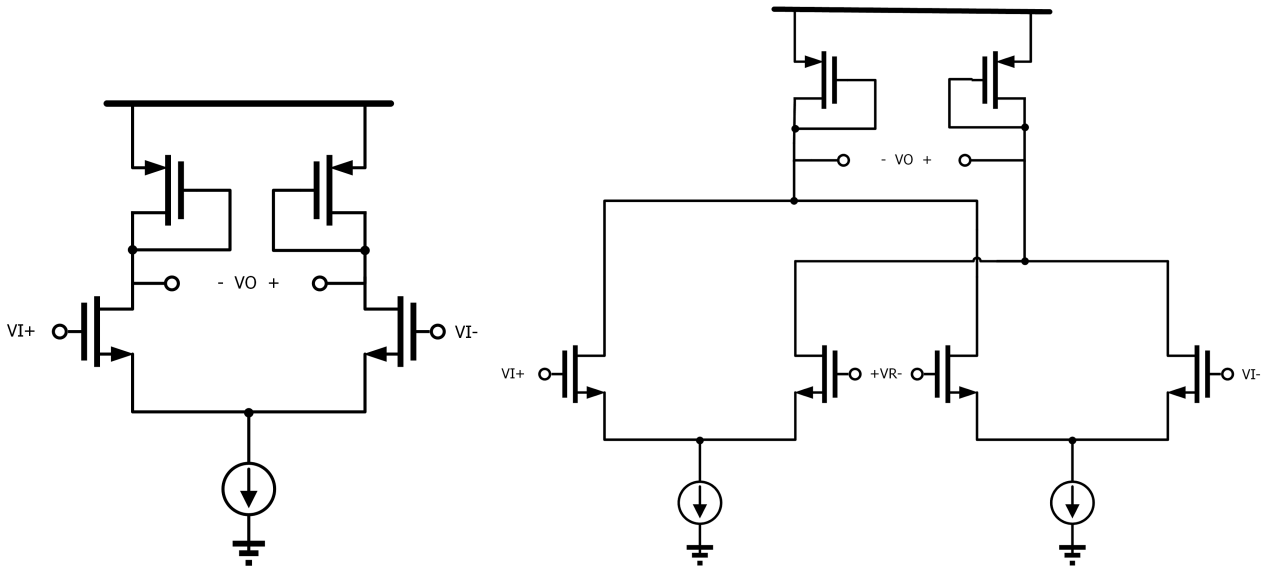

Figure 10: Preamplifier Designs

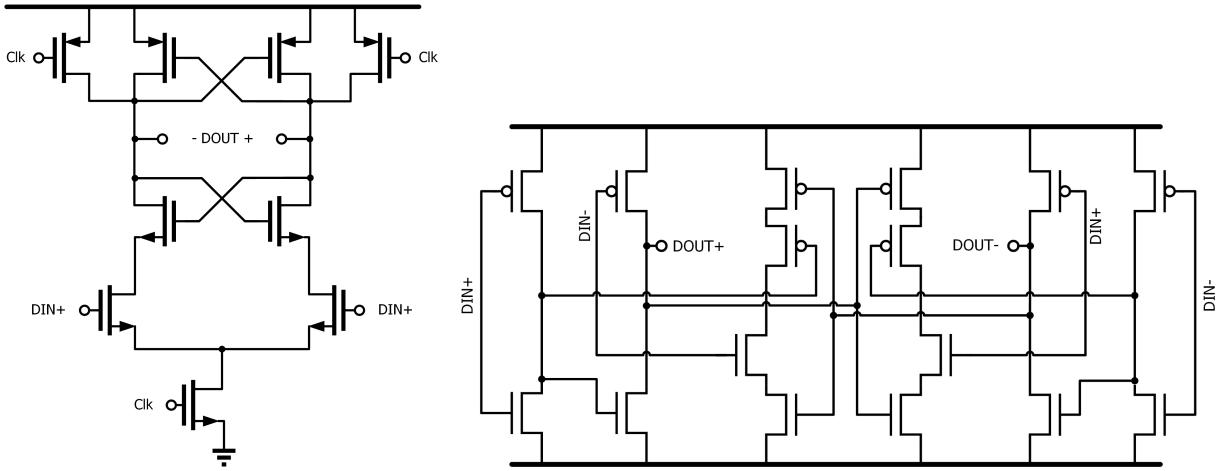

Figure 11: Strong-ARM Latch (left) and RS-Latch (right)

In each pipeline cell, sub-DACs and thermometer-to-binary digital logic are needed. The sub-DAC serves to take the digitized input and convert it to an analog signal to be subtracted through the residue amplifier, and the thermometer-to-binary provides conversion of the 6-comparator output to a 3-bit binary signal. Both cells were implemented in Verilog-A.

```
module sub_dac (comp6, comp5, comp4, comp3, comp2, comp1, c2, c3, c4);
electrical comp6, comp5, comp4, comp3, comp2, comp1, c2, c3, c4;
parameter real vref  = 0.3;
parameter real trise = 0 from [0:inf);
parameter real tfall = 0 from [0:inf);
parameter real tdel  = 0 from [0:inf);
parameter real vtrans  = 2.5;

    real out_scaled; // output scaled as fraction of 256

    analog begin
        out_scaled = -3;
        out_scaled = out_scaled + ((V(comp6) > vtrans) ? 1 : 0);
        out_scaled = out_scaled + ((V(comp5) > vtrans) ? 1 : 0);
        out_scaled = out_scaled + ((V(comp4) > vtrans) ? 1 : 0);
        out_scaled = out_scaled + ((V(comp3) > vtrans) ? 1 : 0);
        out_scaled = out_scaled + ((V(comp2) > vtrans) ? 1 : 0);
        out_scaled = out_scaled + ((V(comp1) > vtrans) ? 1 : 0);
        V(c2) <+ transition((abs(out_scaled) > 0 ? vref*1 : 0)*(out_scaled < 0 ? -1 : 1), tdel, trise, tfall );
        V(c3) <+ transition((abs(out_scaled) > 1 ? vref*1 : 0)*(out_scaled < 0 ? -1 : 1), tdel, trise, tfall );
        V(c4) <+ transition((abs(out_scaled) > 2 ? vref*1 : 0)*(out_scaled < 0 ? -1 : 1), tdel, trise, tfall );
    end
endmodule
```

Figure 12: Sub-DAC Verilog-A

```
module therm_to_bin (comp6, comp5, comp4, comp3, comp2, comp1, vout_0, vout_1, vout_2);
electrical comp6, comp5, comp4, comp3, comp2, comp1, vout_0, vout_1, vout_2;
parameter real vref  = 0.9;
parameter real trise = 0 from [0:inf);
parameter real tfall = 0 from [0:inf);
parameter real tdel  = 0 from [0:inf);
parameter real vtrans  = 2.5;

    real out_scaled; // output scaled as fraction of 256

    analog begin
        out_scaled = 0;
        out_scaled = out_scaled + ((V(comp6) > vtrans) ? 1 : 0);
        out_scaled = out_scaled + ((V(comp5) > vtrans) ? 1 : 0);
        out_scaled = out_scaled + ((V(comp4) > vtrans) ? 1 : 0);
        out_scaled = out_scaled + ((V(comp3) > vtrans) ? 1 : 0);
        out_scaled = out_scaled + ((V(comp2) > vtrans) ? 1 : 0);
        out_scaled = out_scaled + ((V(comp1) > vtrans) ? 1 : 0);
        V(vout_2) <+ transition(((out_scaled % 8 >= 4) ? vref*1 : vref*-1), tdel, trise, tfall );
        V(vout_1) <+ transition(((out_scaled % 4 >= 2) ? vref*1 : vref*-1), tdel, trise, tfall );
        V(vout_0) <+ transition(((out_scaled % 2 >= 1) ? vref*1 : vref*-1), tdel, trise, tfall );
    end
endmodule
```

Figure 13: Thermometer-to-Binary Verilog-A

In addition, an ideal 11-bit DAC was used for testing purposes to visualize the quantized output and determine the SNDR. This was used to calculate the ENOB of the design.

```
module dac_11bit_ideal (vd10, vd9, vd8, vd7, vd6, vd5, vd4, vd3, vd2, vd1, vd0, vout);
electrical vd10, vd9, vd8, vd7, vd6, vd5, vd4, vd3, vd2, vd1, vd0, vout;
parameter real vref  = 1 from [0:inf);
parameter real trise = 0 from [0:inf);
parameter real tfall = 0 from [0:inf);
parameter real tdel  = 0 from [0:inf);
parameter real vtrans  = 0;

    real out_scaled; // output scaled as fraction of 2048

    analog begin
        out_scaled = 0;
        out_scaled = out_scaled + ((V(vd10) > vtrans) ? 1024 : 0);
        out_scaled = out_scaled + ((V(vd9) > vtrans) ? 512 : 0);
        out_scaled = out_scaled + ((V(vd8) > vtrans) ? 256 : 0);
        out_scaled = out_scaled + ((V(vd7) > vtrans) ? 128 : 0);
        out_scaled = out_scaled + ((V(vd6) > vtrans) ? 64 : 0);
        out_scaled = out_scaled + ((V(vd5) > vtrans) ? 32 : 0);
        out_scaled = out_scaled + ((V(vd4) > vtrans) ? 16 : 0);
        out_scaled = out_scaled + ((V(vd3) > vtrans) ? 8 : 0);
        out_scaled = out_scaled + ((V(vd2) > vtrans) ? 4 : 0);
        out_scaled = out_scaled + ((V(vd1) > vtrans) ? 2 : 0);
        out_scaled = out_scaled + ((V(vd0) > vtrans) ? 1 : 0);
        V(vout) <+ transition( 2*vref*(out_scaled/2048)-vref, tdel, trise, tfall );
    end
endmodule
```

Figure 14: Ideal 11-bit DAC Verilog-A

4. Simulation Results

The ADC was modeled in Simulink to determine the amplifier specifications. The best performance requires a 70dB 500MHz GBW SHA amplifier and a 60dB 1.5GHz residue amplifier. The amplifier specifications in Cadence were adjusted based on performance.

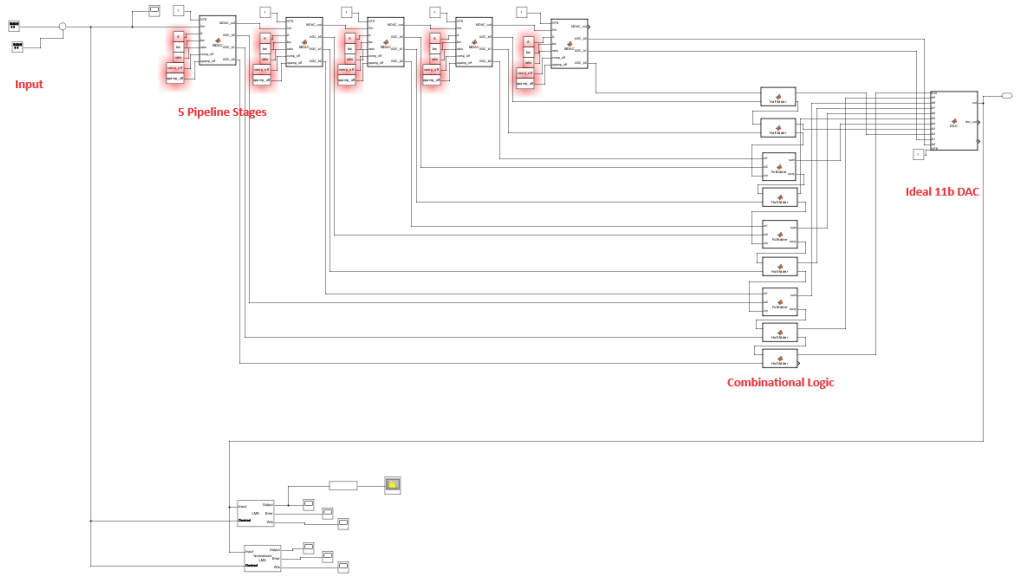

Figure 15: Simulink System Level Testbench

The unit capacitor sizing was based on thermal noise considerations - 200fF was used. Due to the high power consumption of the feedforward amplifier, it was not used.

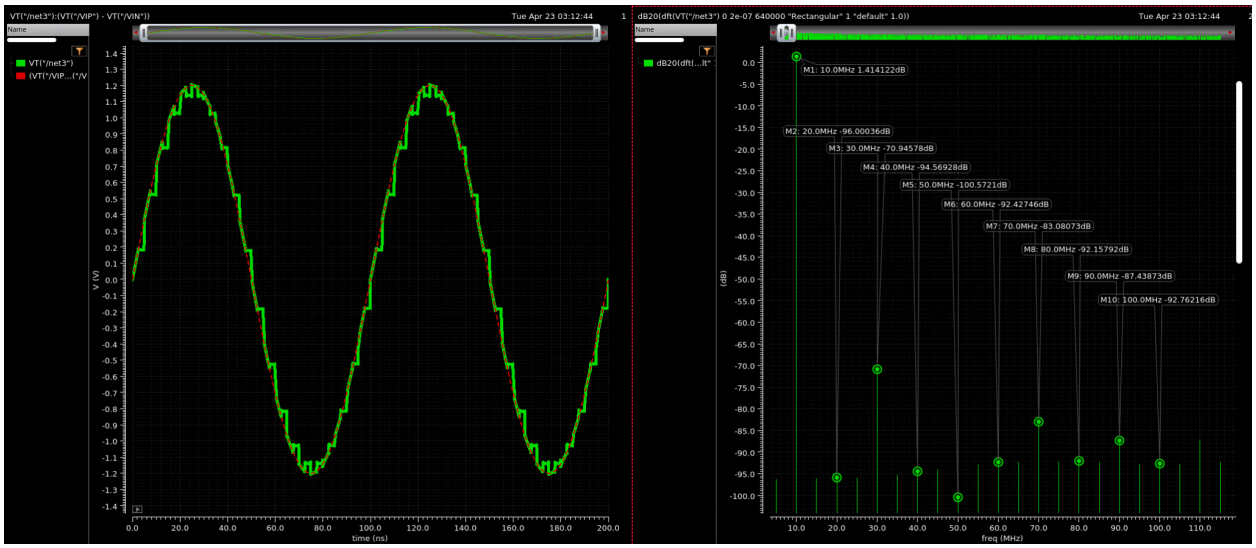

Figure 16: Bootstrap Switch Transient Response and Harmonic Content

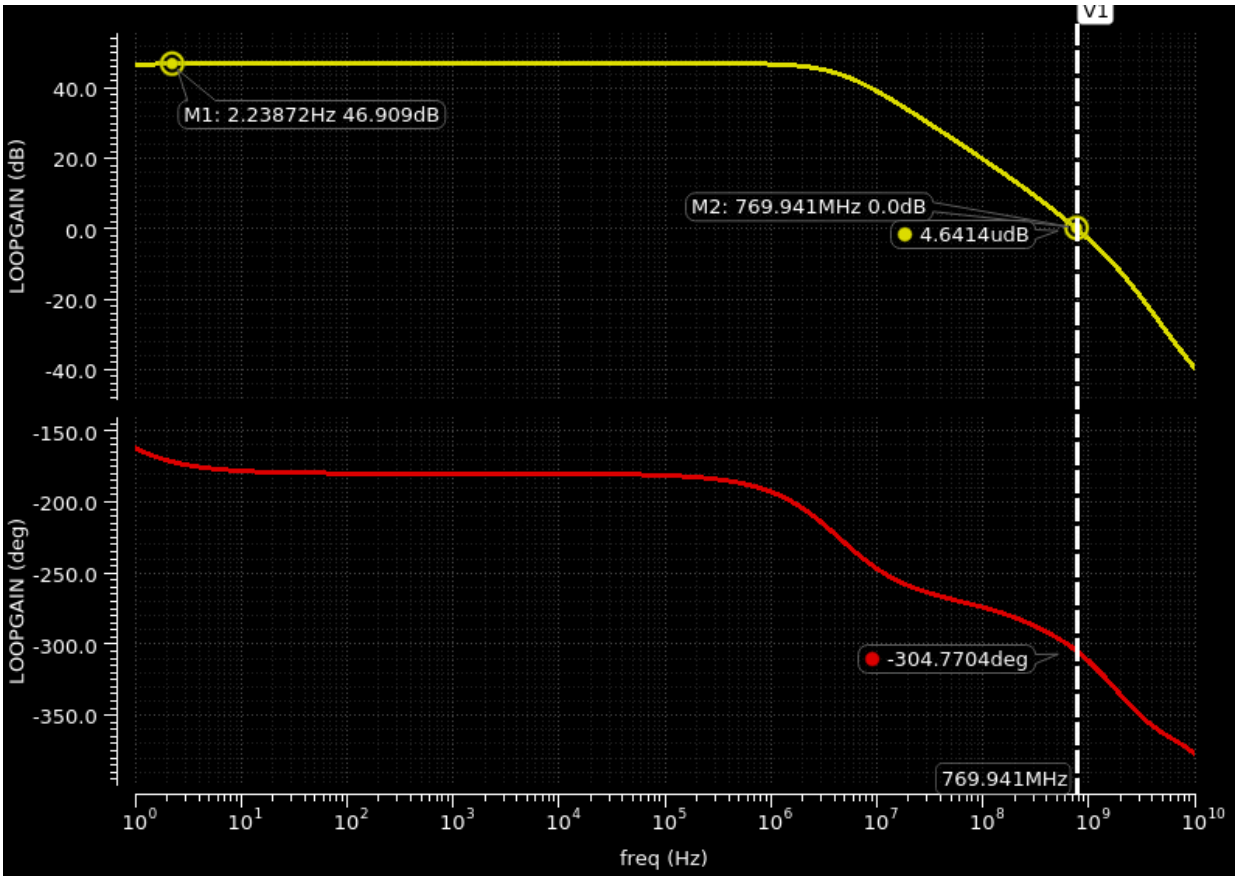

Figure 17: SHA Hold Phase Loop Characteristics

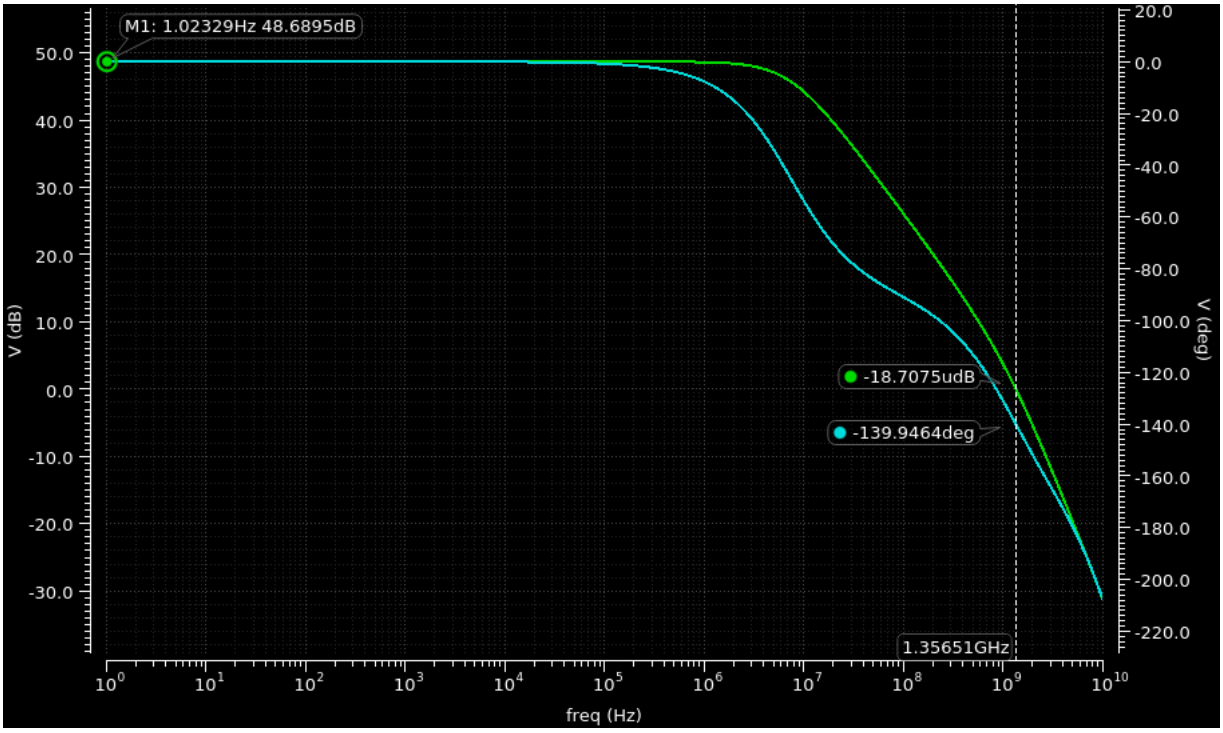

Figure 18: Residue RFC Amplifier Core Open Loop AC Response

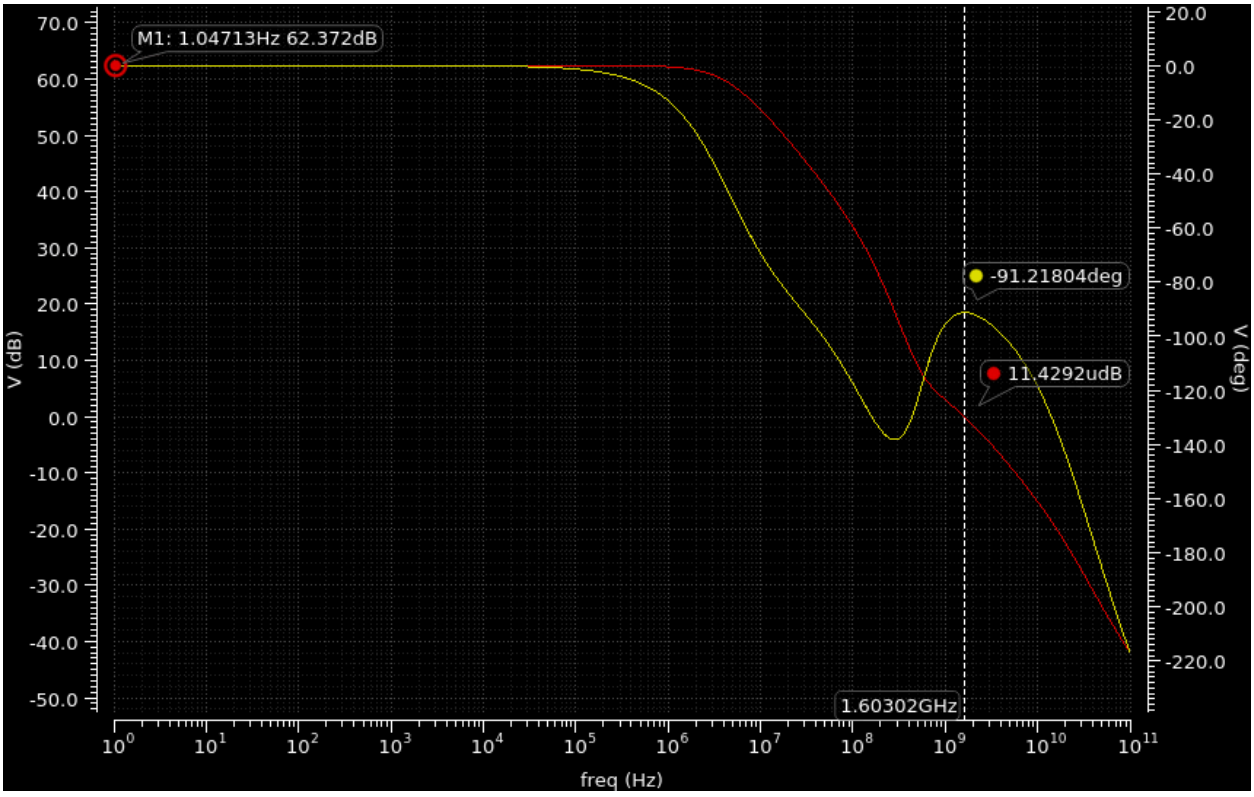

Figure 19: Feedforward Amplifier Core Open Loop AC Response

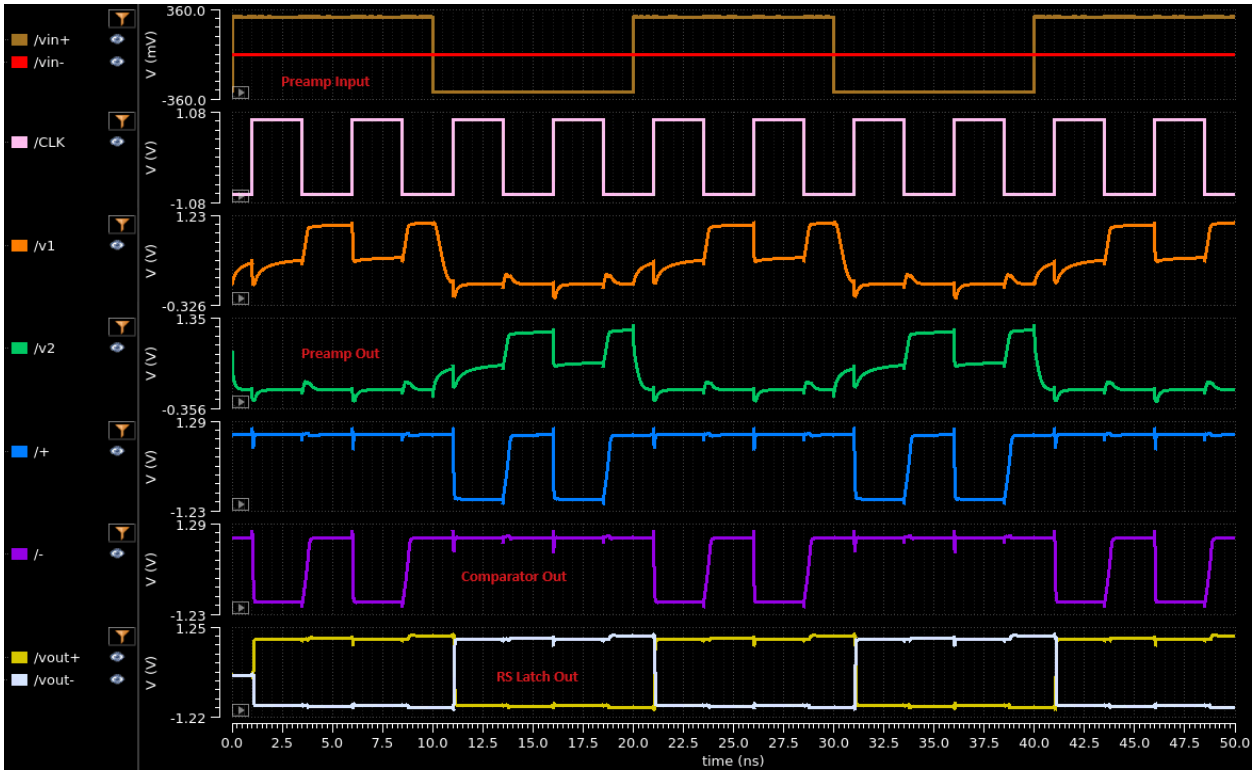

Figure 20: Sub-ADC Transient Waveforms

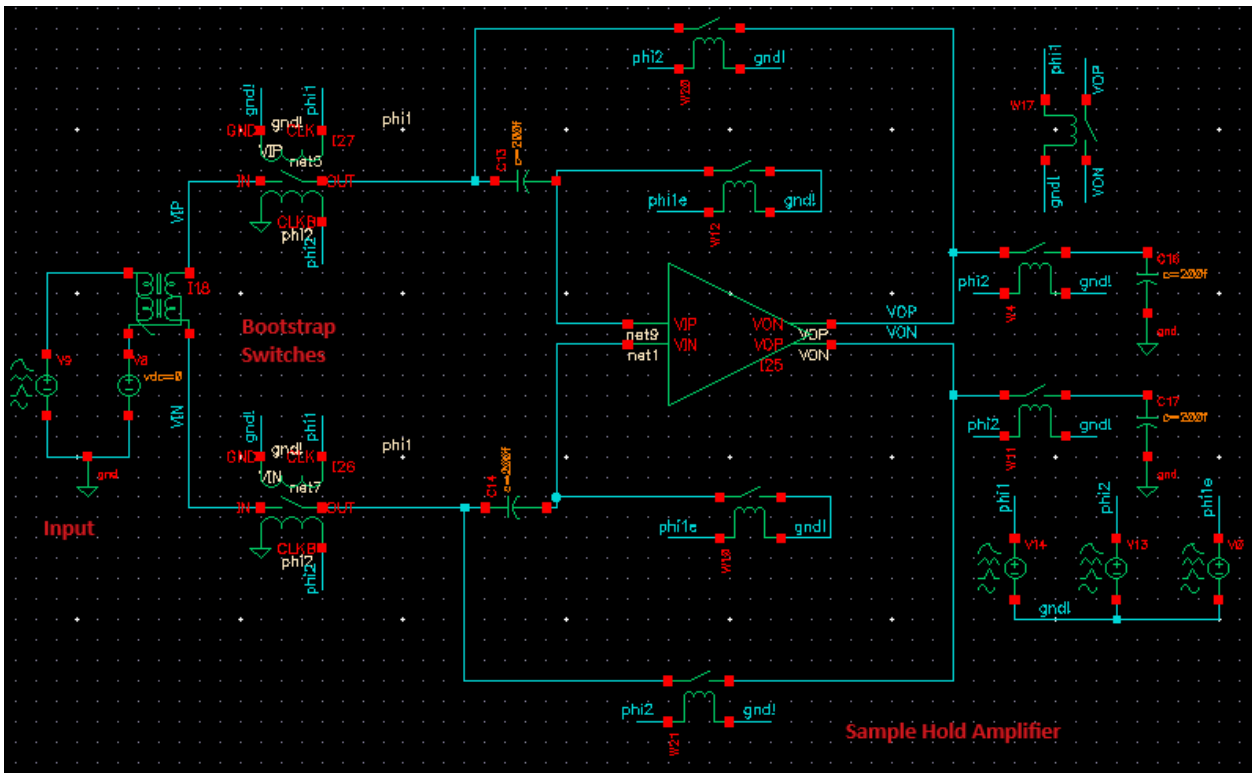

Figure 21: SHA Testbench

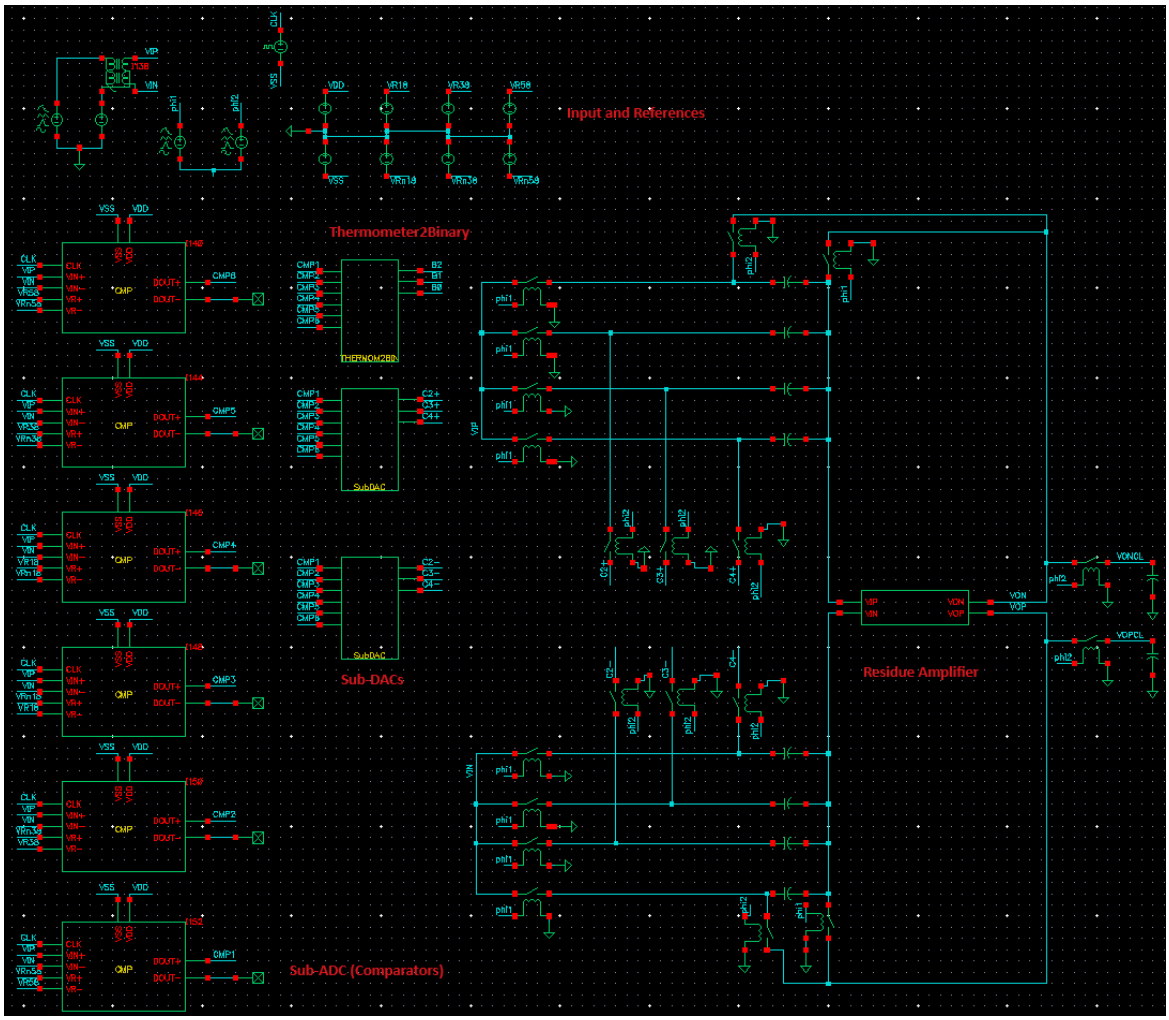

Figure 22: Pipeline Cell Testbench

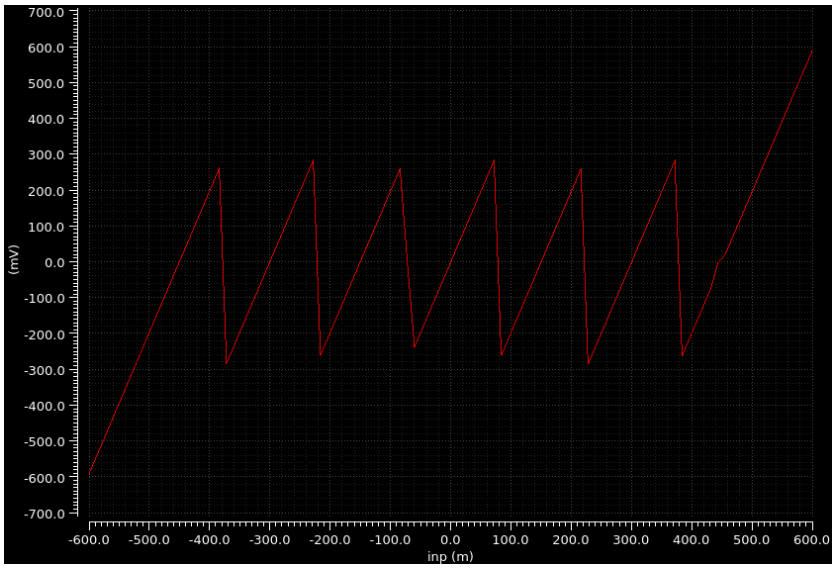

Figure 23: Simulated Residue Transfer Function

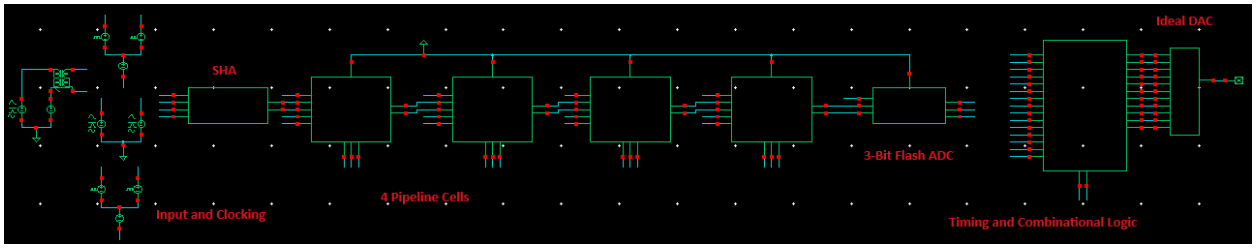

Figure 24: Full Pipeline ADC Testbench

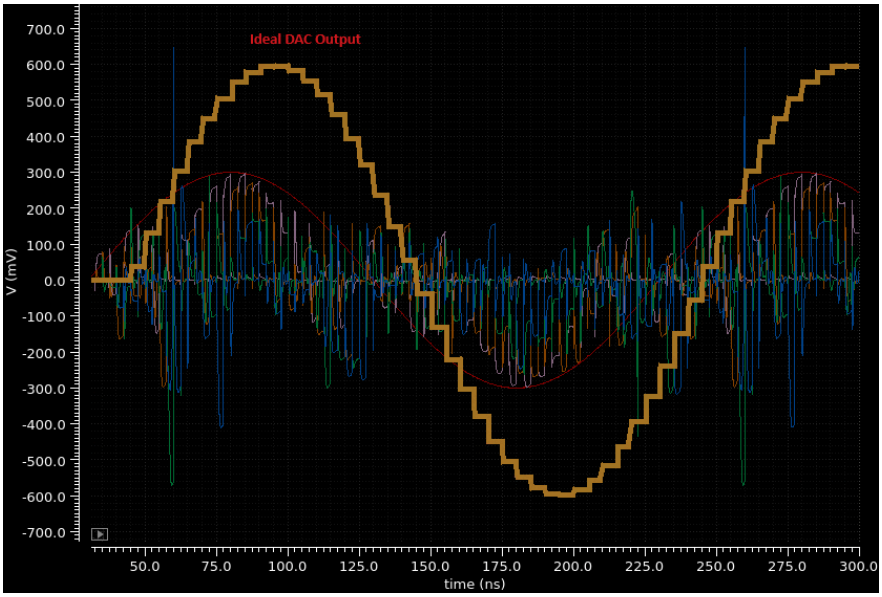

Figure 25: Input, Residue Amplifier Output, and Testing DAC Waveforms

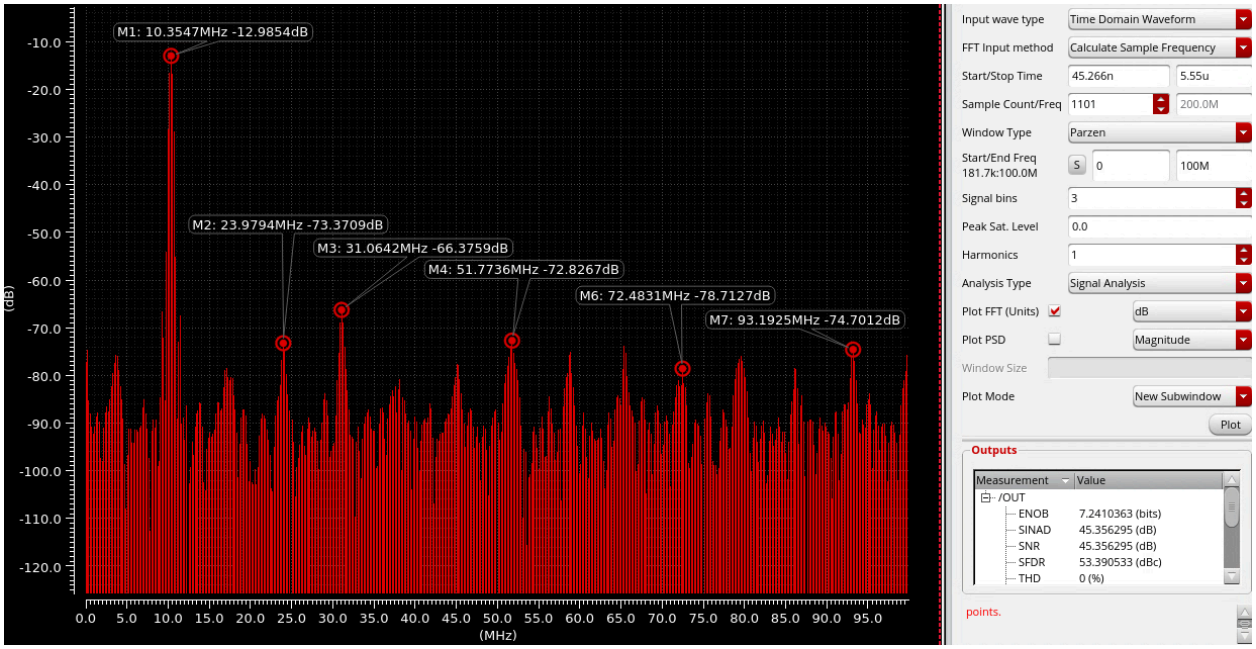

Figure 26: FFT Plot with Incommensurate Input and Sampling Frequency

5. Conclusion

From the FFT, the measured SNDR is 45.4dB which corresponds to an ENOB of 7.24 bits. The power consumed by the full ADC is 9.78mW with the following breakdown: 2mW from the SHA, 5.32mW from the residue amplifiers, and 3.46mW from the switches and comparators (preamplifiers, Strong-ARM, and RS latch). While the digital logic was modeled in Verilog-A, since the clock frequency is slow relative to the ft of the process, the logic can be implemented at minimum sizing making the power consumption only an estimated few hundred microwatts. The FoM is 145.5dB. While the ENOB of the ADC is low, this is similar to the performance of uncalibrated ADCs in the literature. In order to bring the ENOB of a pipeline ADC to within 0.5 bits of the desired resolution, a post-calibration scheme such as the implementation of the LMS algorithm through adaptive filtering is necessary. Reference [9] shows how much improvement can be achieved by the addition of a calibration scheme. More time is needed to implement this.

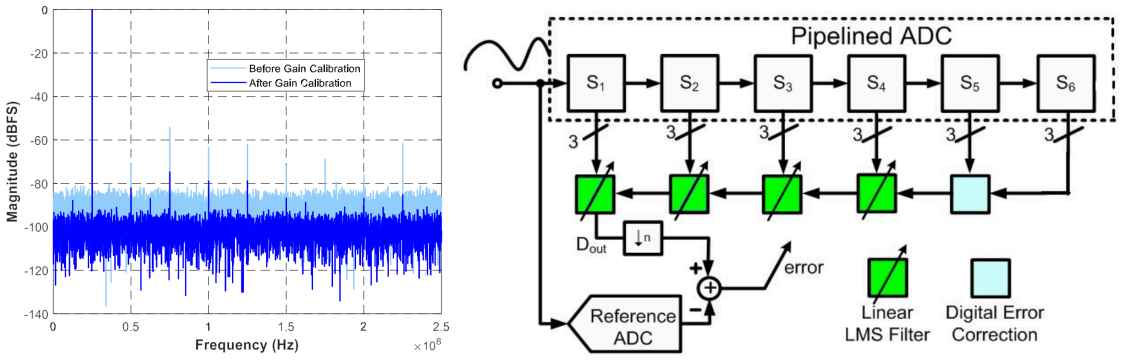

Figure 27: FFT with Calibration [9] and Common Adaptive Filter Scheme [8]

Table 1: Literature Comparison

Specification	This Work	[2]	[4]	[6]	[9]
Sample Rate (MSps)	200	14.3	1000	75	500
Power (mW)	9.78	36	350	314	18.15
SNDR (dB)	45.4	58.5	62	68.2	66
SFDR (dB)	53.4	-	72.6	80	64.1
ENOB (bits)	7.24	9.43	10	11	10.67
FoM* (dB)	145.5	141.5	153.5	149	167

6. References

[1] B. Razavi, "The Bootstrapped Switch [A Circuit for All Seasons]," in *IEEE Solid-State Circuits Magazine*, vol. 7, no. 3, pp. 12-15, Summer 2015, doi: 10.1109/MSSC.2015.2449714.

[2] A. M. Abo and P. R. Gray, "A 1.5-V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter," in *IEEE Journal of Solid-State Circuits*, vol. 34, no. 5, pp. 599-606, May 1999, doi: 10.1109/4.760369.

[3] B. Razavi, "The StrongARM Latch [A Circuit for All Seasons]," in *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12-17, Spring 2015, doi: 10.1109/MSSC.2015.2418155.

[4] M. Liu, D. Li and Z. Zhu, "A Dual-Supply Two-Stage CMOS Op-amp for High-Speed Pipeline ADCs Application," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 4, pp. 650-654, April 2020, doi: 10.1109/TCSII.2019.2926133.

[5] M. Liu, D. Li and Z. Zhu, "A Dual-Supply Two-Stage CMOS Op-amp for High-Speed Pipeline ADCs Application," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 4, pp. 650-654, April 2020, doi: 10.1109/TCSII.2019.2926133.

[6] Haitao Wang, Hui Hong, Lingling Sun and Zhiping Yu, "A sample-and-hold circuit for 10-bit 100MS/s pipelined ADC," *2011 9th IEEE International Conference on ASIC*, Xiamen, 2011, pp. 480-483, doi: 10.1109/ASICON.2011.6157226.

[7] B. Murmann and B. E. Boser, "A 12-bit 75-MS/s pipelined ADC using open-loop residue amplification," in *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2040-2050, Dec. 2003, doi: 10.1109/JSSC.2003.819167.

[8] C. Tsang, Y. Chiu, J. Vanderhaegen, S. Hoyos, C. Chen, R. Brodersen, B. Nikolic, “Background ADC Calibration in Digital Domain,” CICC 2008, San Jose, CA

[9] M. H. Naderi, C. Park, S. Prakash, M. Kinyua, E. G. Soenen and J. Silva-Martinez, "A 27.7 fJ/conv-step 500 MS/s 12-Bit Pipelined ADC Employing a Sub-ADC Forecasting Technique and Low-Power Class AB Slew Boosted Amplifiers," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 9, pp. 3352-3364, Sept. 2019, doi: 10.1109/TCSI.2019.2927383.

[10] Mehdi Sotoudeh, Farzan Rezaei, A new dual-network bootstrapped switch for high-speed high-resolution applications, Computers & Electrical Engineering, Volume 91, 202, 107125, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2021.107125. (https://www.sciencedirect.com/science/article/pii/S0045790621001294)

[11] S. Hoyos, “ECEN 610: Nyquist ADCs”

[12] J. Silva-Martinez, “Design Techniques for High Speed, Low Voltage, and Low Power Pipeline ADCs”

[13] J. Silva-Martinez, “Design of Operational Amplifiers for Advanced Analog-to-Digital Converters”

[14] V. Saxena, “Pipelined Analog-to-Digital Converters”

[15] Bult, Klaas, and Govert JGM Geelen. "The CMOS gain-boosting technique." Analog integrated circuits and signal processing 1 (1991): 119-135.

7. Appendix

The Cadence level schematics of the implemented circuits are shown here for reference.

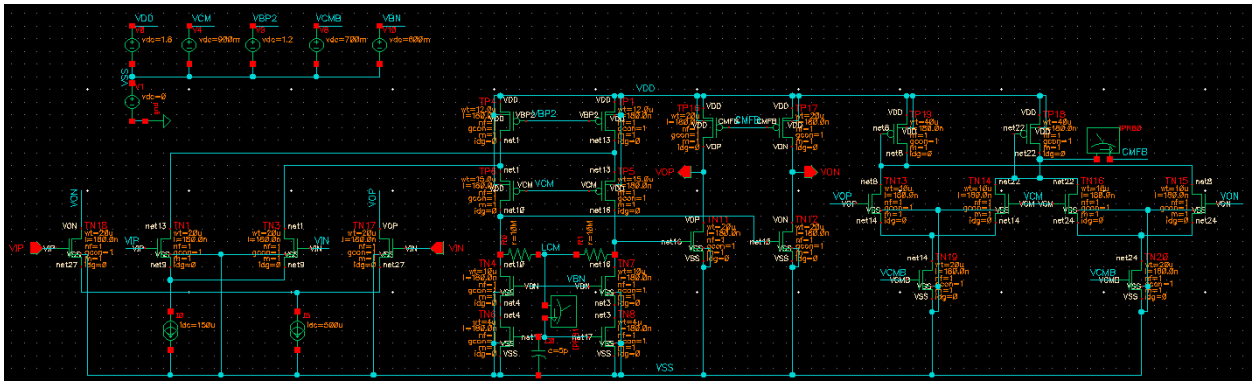

Figure 28: Feedforward Amplifier Core

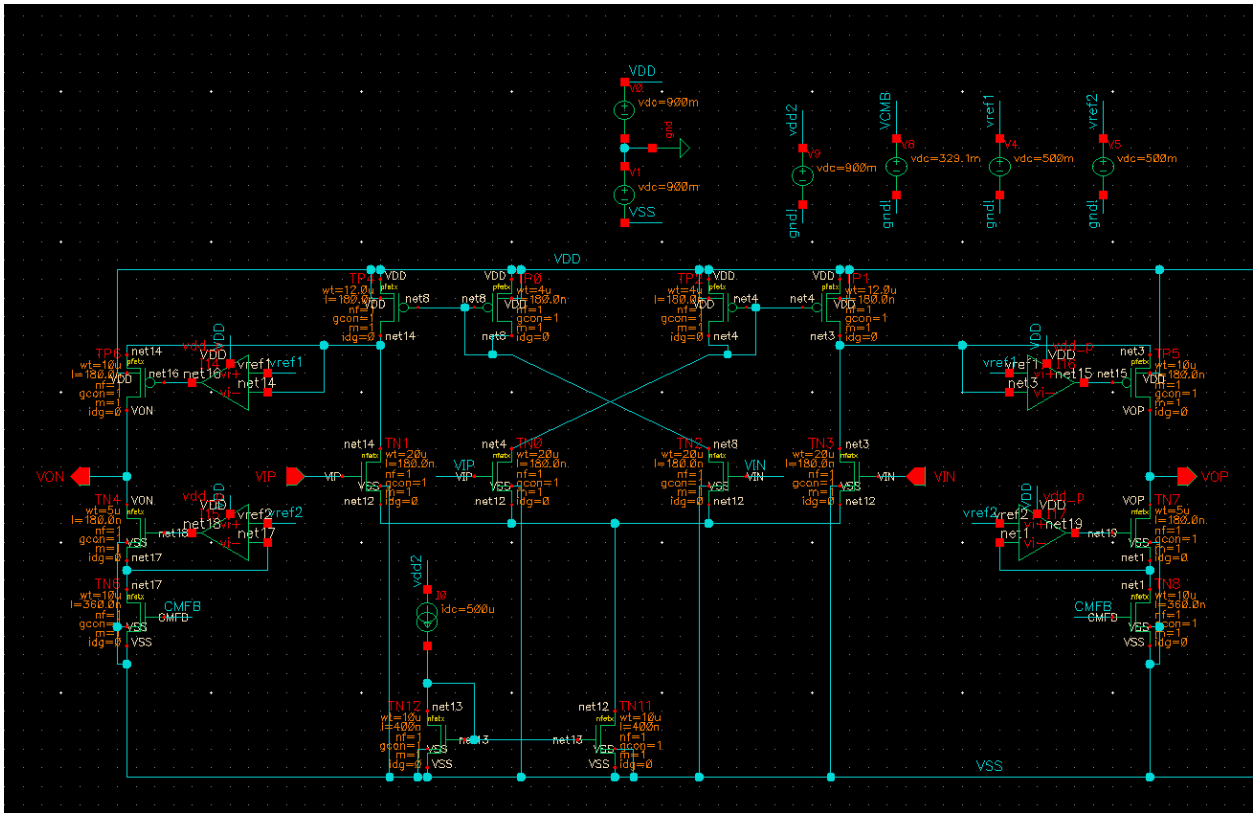

Figure 29: SHA Core

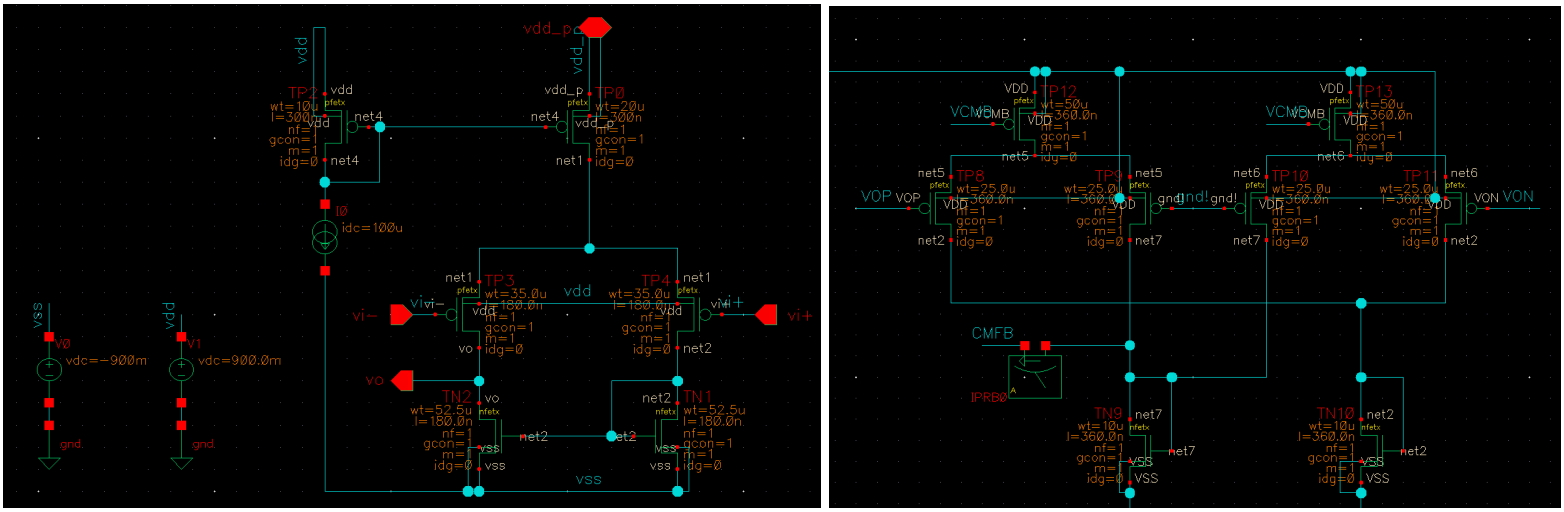

Figure 30: SHA Auxiliary Amplifier and CMFB Engine

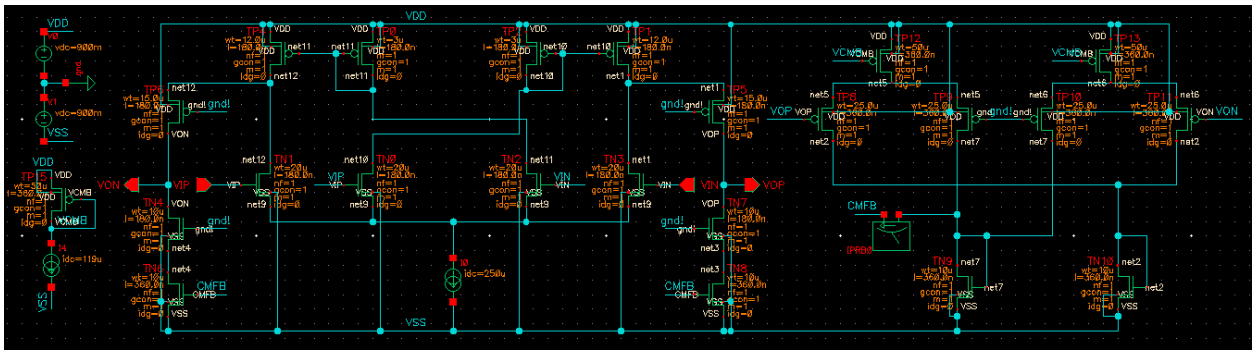

Figure 31: Residue Amplifier Core

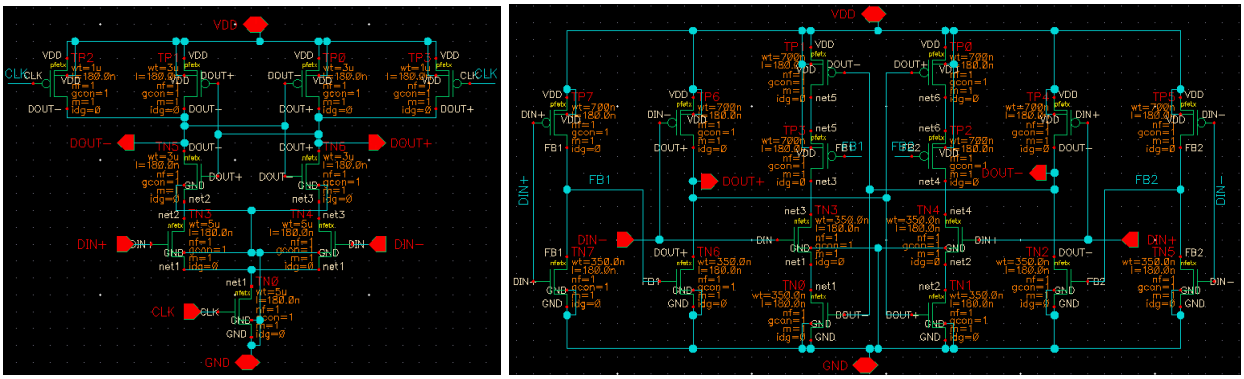

Figure 32: Comparator Strong-ARM and RS Latches

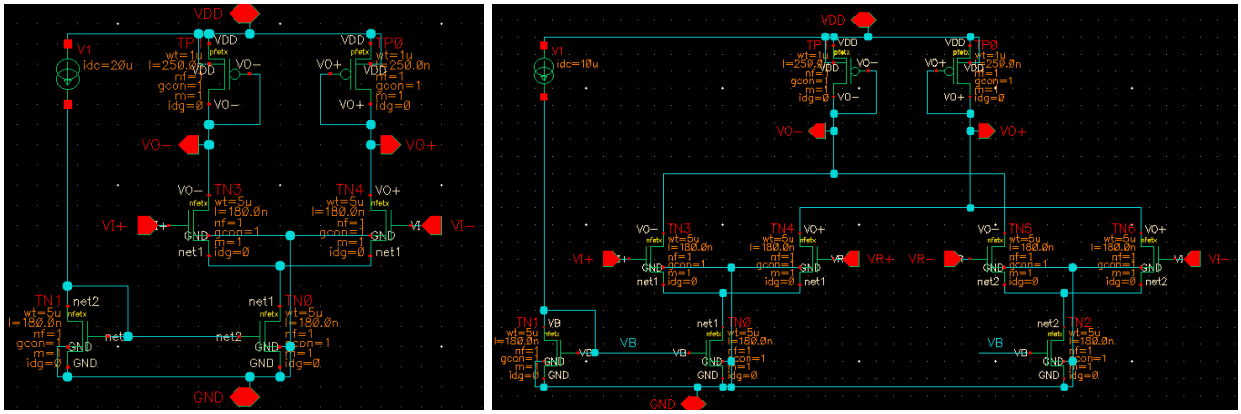

Figure 33: Comparator Preamplifiers (Single and Differential Reference)

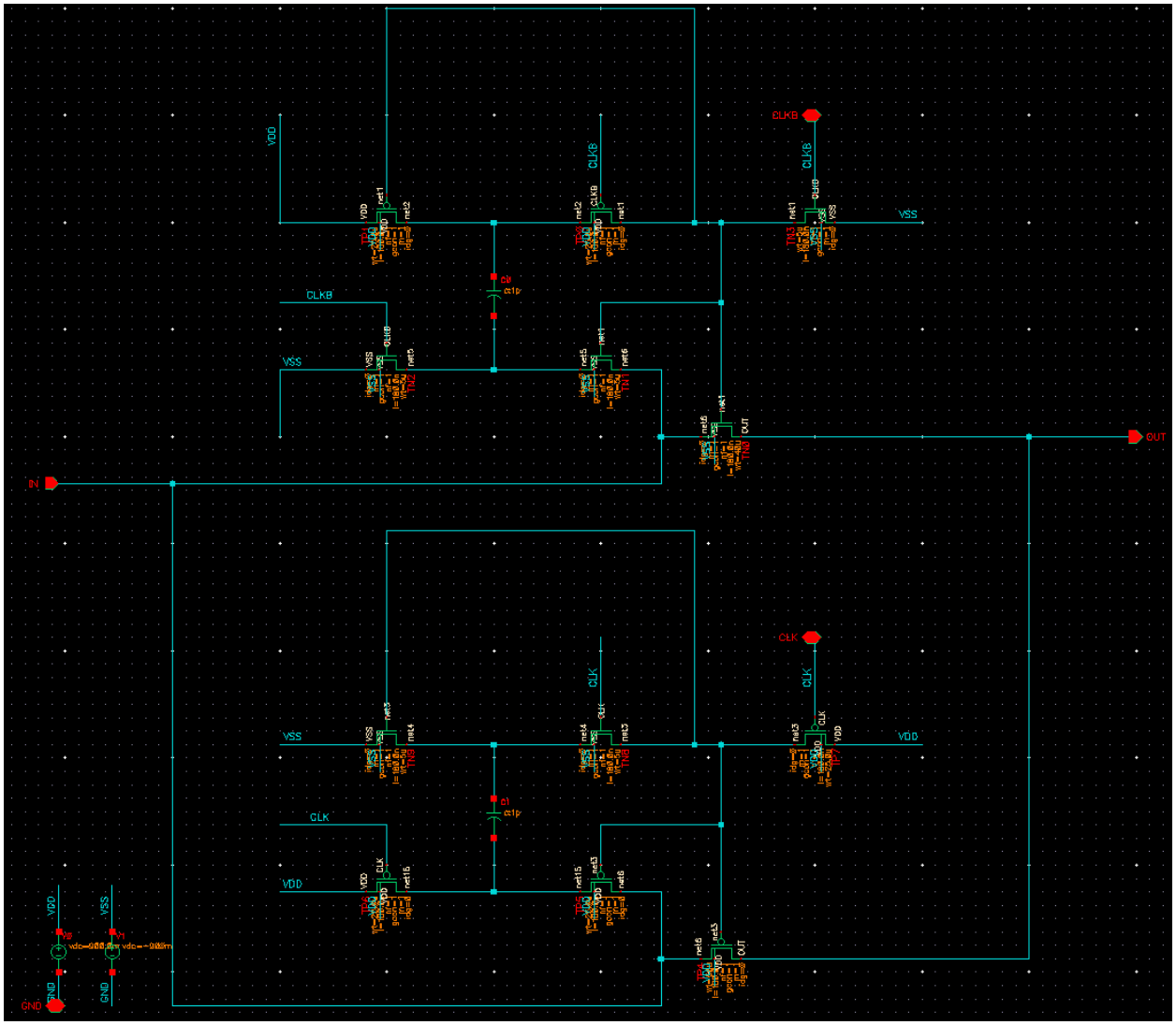

Figure 34: Bootstrap Switch

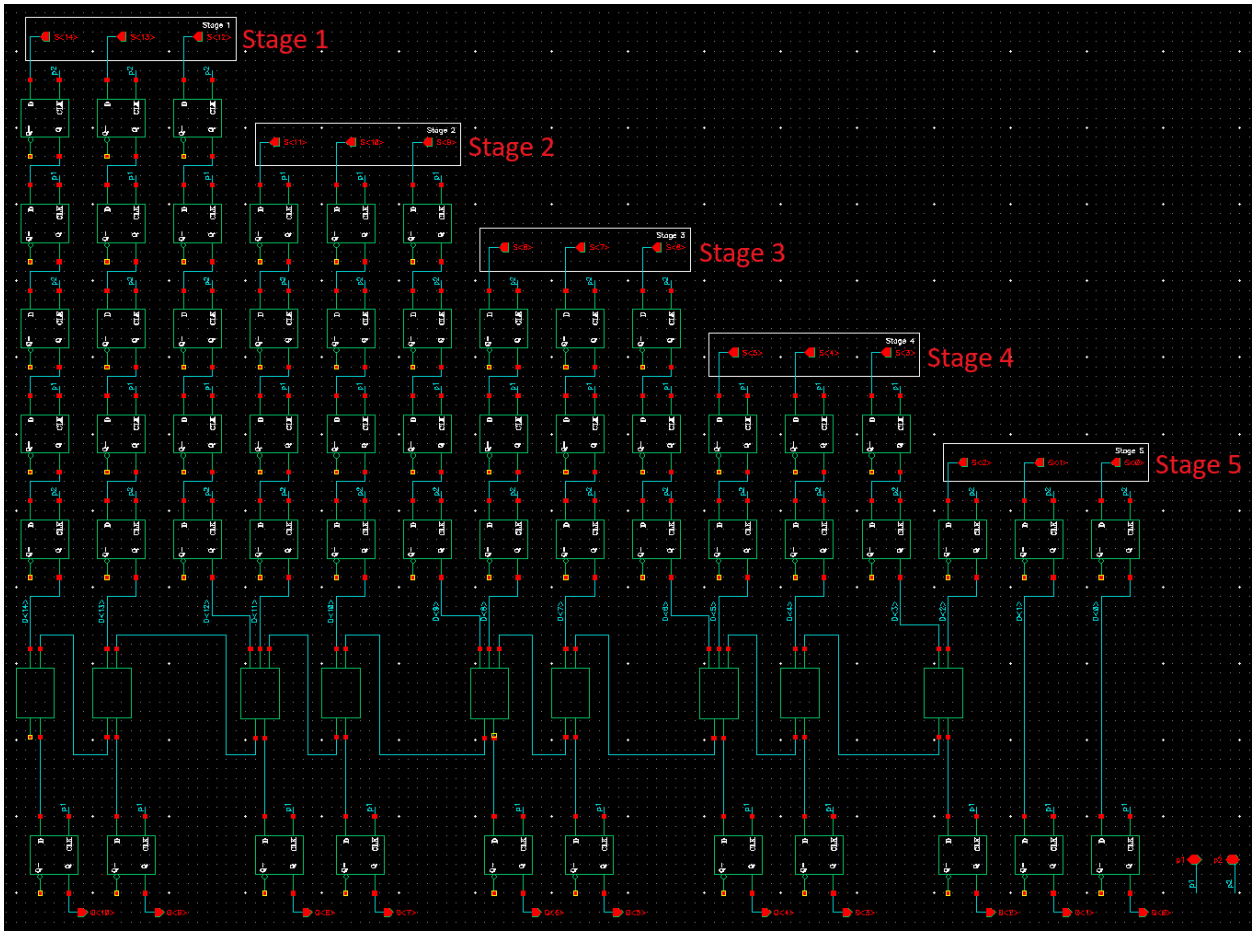

Figure 35: Digital Backend Logic and Timing