ELECTRICAL & COMPUTER ENGINEERING
TEXAS A&M UNIVERSITY

A Novel Mixed-Signal Flash-based Finite Impulse Response (FFIR) Filter for IoT Applications

Cheng-Yen Lee[◊]
Sunil P. Khatri[◊]
Ali Ghrayeb[+]

[◊] Texas A&M University, College Station, USA
[+] Texas A&M University at Qatar, Doha, Qatar

Outline

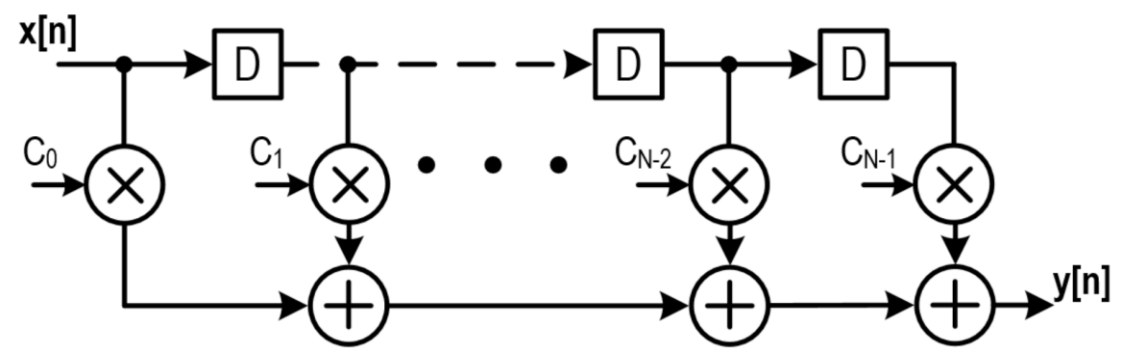
- Introduction
- Prior Works and Motivation
- Background: Flash Transistor
- Proposed Structure
  - Flash-based Finite Impulse Response (FFIR) Filter
  - Flash-based Coefficient Multiplier (FCM)
- Experimental Results
  - Simulation Environment Setup
  - Performance of FCM
  - Performance of FFIR
- Comparison
- Conclusions

Introduction

- Low-power Internet of Things (IoT) devices have recently gained significant attention.
  - These IoT devices **collect** and **process** information with **extremely limited computing resources**.
- For IoT devices implementing Digital Signal Processing (DSP) applications, the power budget for the filtering operation can be significant.
  - **Achieving optimal performance** while ***minimizing power consumption***, ***energy usage, and chip area*** becomes a key design requirement.
- Among the different types of filters in DSP applications, the **Finite Impulse Response (FIR) filter** is widely used , due to its **simplicity**, **stability**, and **linear phase response**.

Introduction: Digital FIR Filter

- A direct-form N-tap Digital FIR (DFIR) filter comprises
  - A delay line: Stores input signal samples.
  - N coefficient multipliers: Multiply samples with filter coefficients ($C_{0\text{-}N\text{-}1}$).
  - N-1 summation circuits: Sum multiplier results to produce the final output.
- To achieve rigorous filter specifications, such as **narrow transition band** and **high stopband attenuation**, **DFIR filters require a large number of taps**.
- The use of DFIR filters with a large number of taps becomes impractical for IoT devices.
  - Increases power consumption



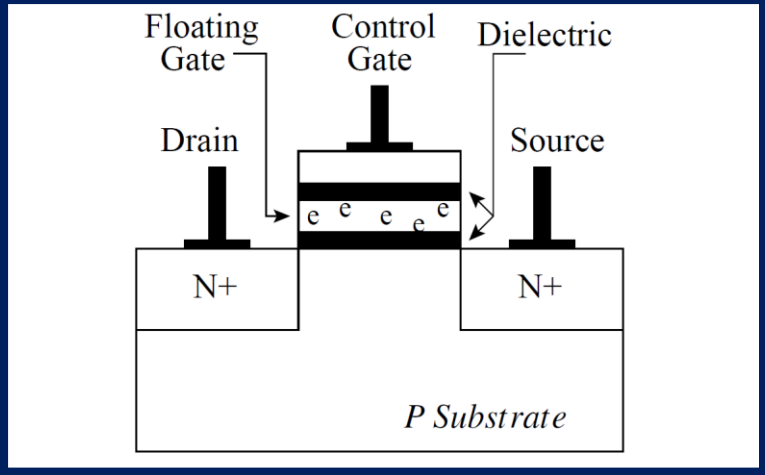
Prior Works and Motivation

- Several prior works have focused on the development of **Analog FIR (AFIR) filters**, leveraging **analog computation to improve energy efficiency**. However, these works face challenges such as:
  1. Complex control logic
  2. Dependence on size ratios for generating output signals
  3. Low operational speed
  4. Sensitivity to PVT variations and transistor aging effects

- To address these limitations, we propose a Flash-based FIR (FFIR) filter that
  - Utilizes the identical-sized components (flash transistors) to attain the same functionality.
  - Allows for adjustment of filter coefficients by reprogramming the threshold voltages (V_t) of flash transistors in the FCMs.
  - Mitigates PVT variations and aging effects through the use of flash transistors.

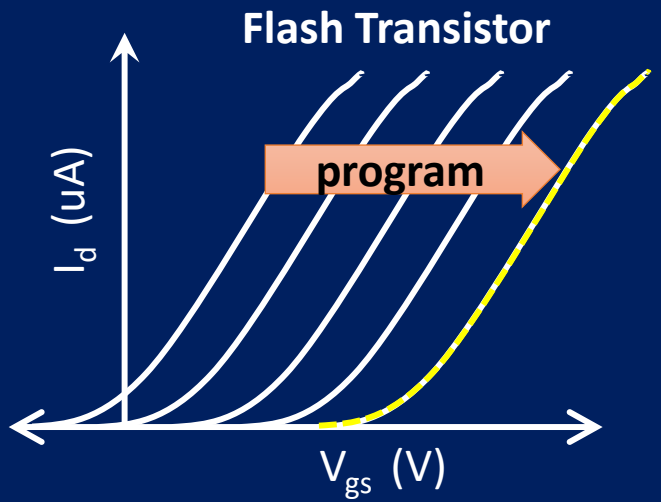
Background: Flash Transistor

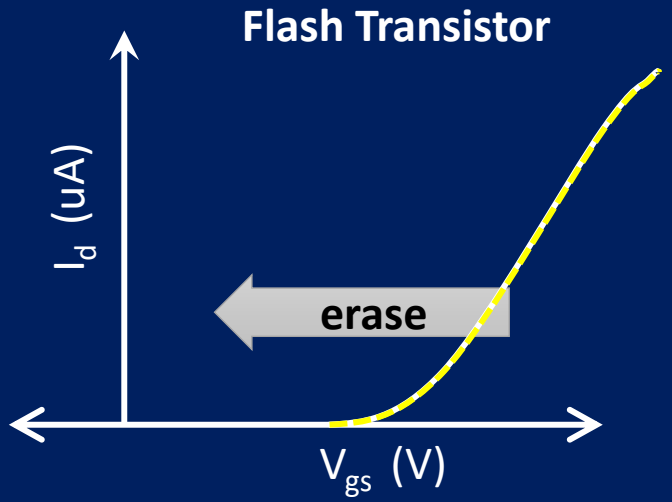
- A flash transistor is a field effect transistor (FET) with two gates.
  1. **Control gate**, which is structurally and functionally similar to the gate of a CMOS transistor.
  2. **Floating gate**, which is physically located between the first gate and the channel.
- The threshold voltage (V_t) of the flash transistor can be adjusted by causing electrons to tunnel out of (or into) the floating gate **by Fowler-Nordheim (FN) tunneling**, or **hot carrier injection (HCI)**.



Background: Flash Transistor

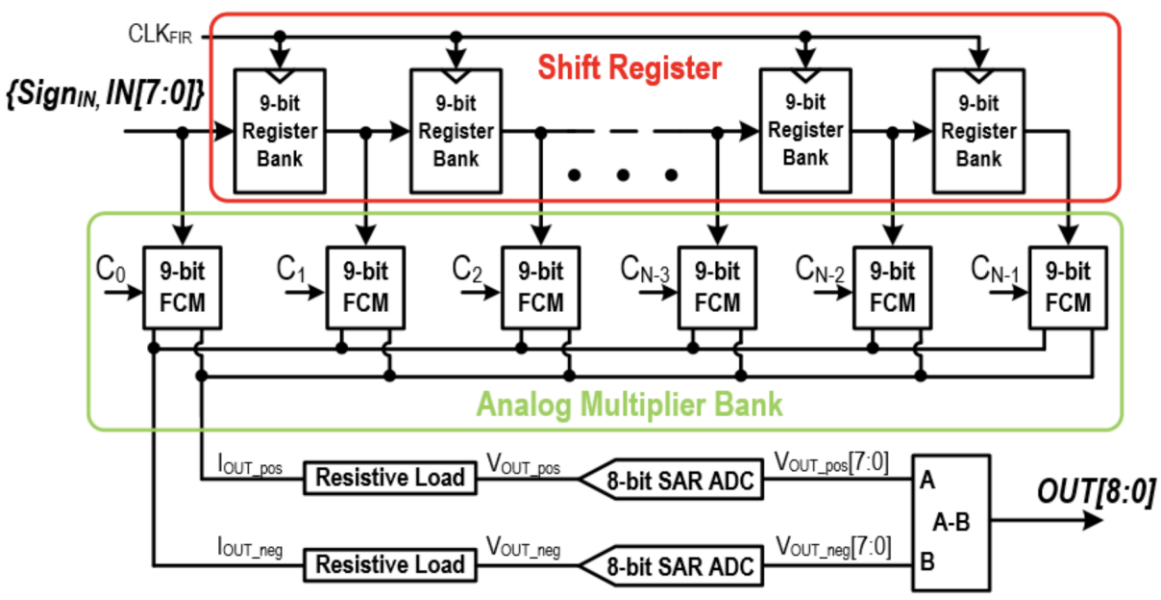
- Program (increase V_t):
  - Electrons are injected into the floating gate from substrate.
  - The control gate is driven high (10V~20V), while driving the source, drain, bulk to ground.
  - The V_t value can be precisely controlled by the programming pulse duration, typically achieved through a series of 'program-verify' cycles.
- Erasure (reduce V_t):
  - Electrons are removed from the floating gate.
  - The bulk voltage is driven high, the control gate is grounded, and the source and drain terminals are left floating.





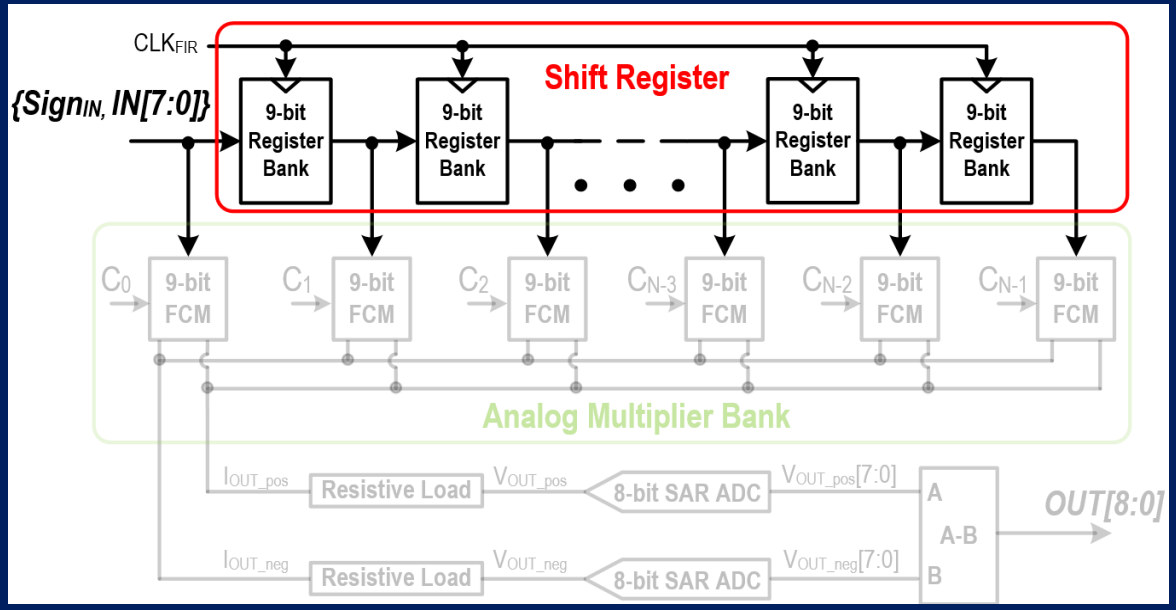
Proposed Structure

- The Proposed N-tap Flash-Based Finite Impulse Response (FFIR) Filter consists of three parts.
  1. A shift register
  2. An analog multiplier bank
  3. An A/D block (2 SAR ADCs and a digital subtractor)
- The FFIR filter implements the FIR computation in the **direct form structure**.



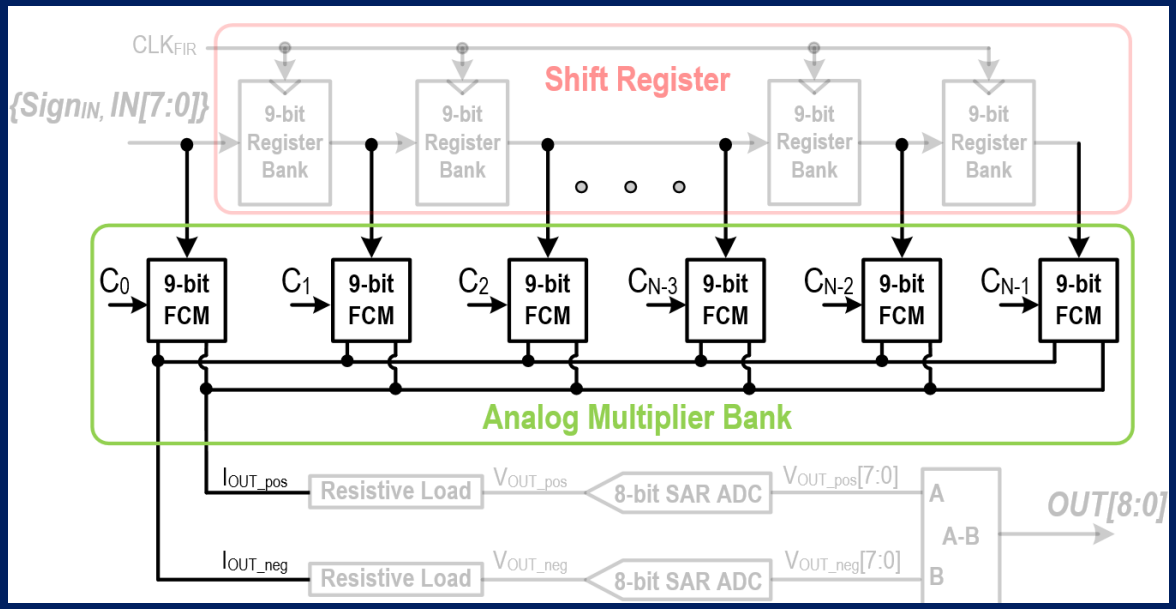
Flash-Based Finite Impulse Response (FFIR) Filter

- The shift register is composed of N-1 9-bit register banks that are connected in a cascade configuration.
  - The register banks are synchronized by a shared clock signal (CLK_{FIR}).
- The shift register stores input samples ({$Sign_{IN}$, IN[7:0]}) and shifts the samples from one stage to the next.



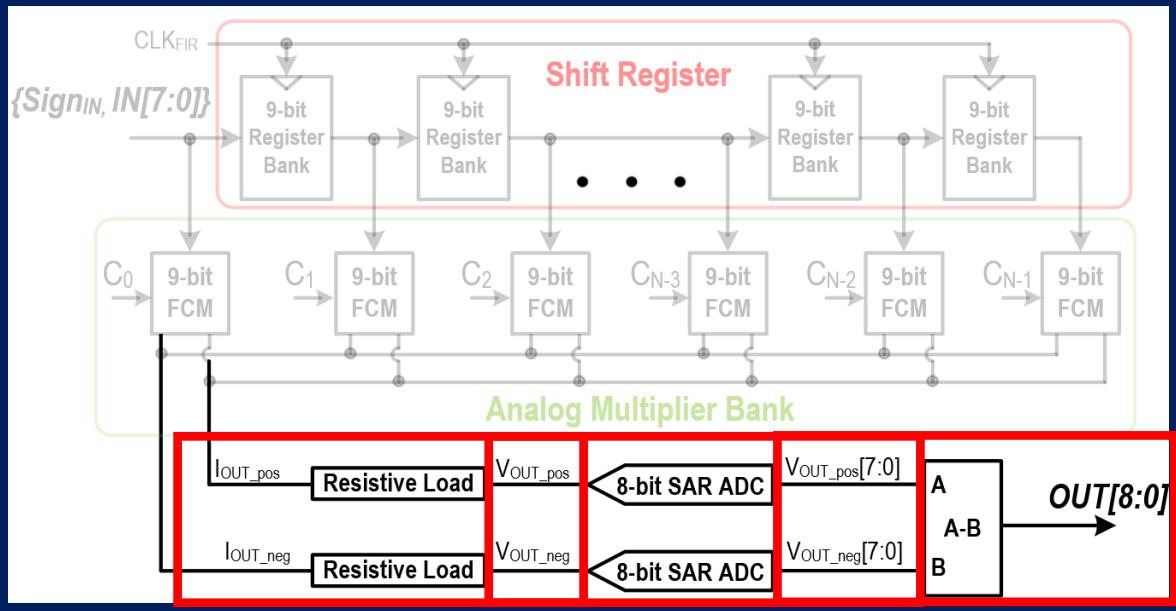
Flash-Based Finite Impulse Response (FFIR) Filter

- The analog multiplier bank consists of N 9-bit Flash-based Coefficient Multipliers (FCMs).
  - The FCMs perform coefficient multiplication of the **input data (digital domain)** and **the coefficient (stored in the flash transistor)**.
    - The output is in the analog current domain.
  - The direction of the current flow for each FCM is determined by the sign of the corresponding coefficient and the shifted input.
- Currents from each of the N FCMs are summed towards I_{OUT_pos} or I_{OUT_neg} using Kirchhoff's Current Law (KCL).



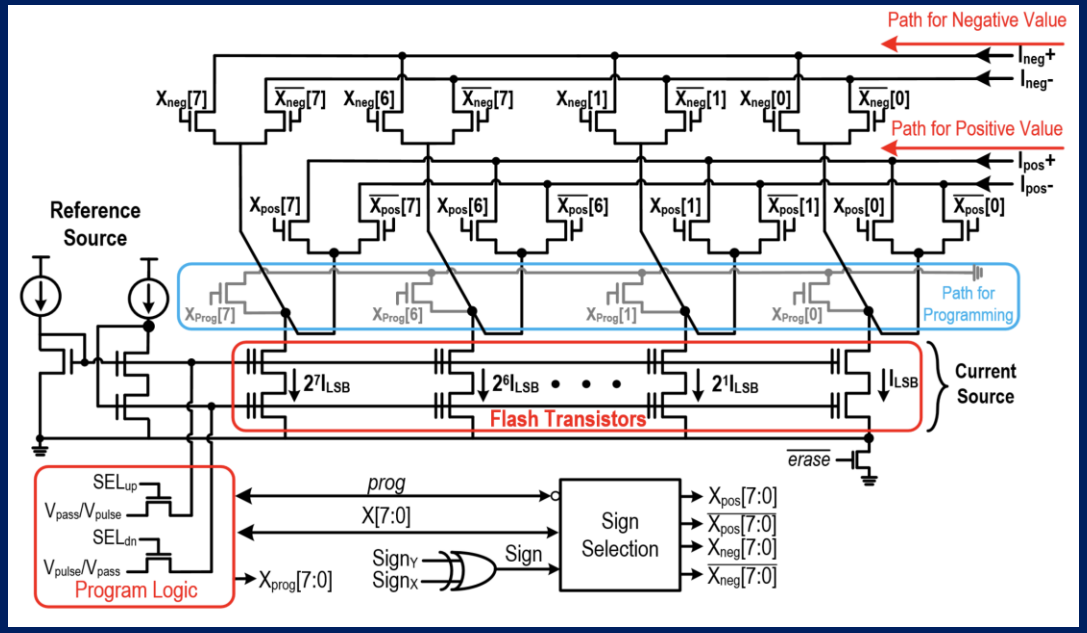
Flash-Based Finite Impulse Response (FFIR) Filter

- The A/D block converts the analog current results into digital domain.
  1. **Current-to-Voltage Conversion:**
     - Analog currents (I_{OUT_pos} and I_{OUT_neg}) are converted into analog voltages (I_{OUT_pos} and V_{OUT_neg}) through **resistive loads.**
  2. **Voltage-to-Digital Conversion:**
     - V_{OUT_pos} and V_{OUT_neg} are converted into digital signals (V_{OUT_pos}**[7:0]** and V_{OUT_neg}**[7:0]**) by 8-bit SAR ADCs.
  3. **Final Output Derivation:**
     - **OUT[8:0]** = V_{OUT_pos}**[7:0]** − V_{OUT_neg}**[7:0]** (in digital domain).



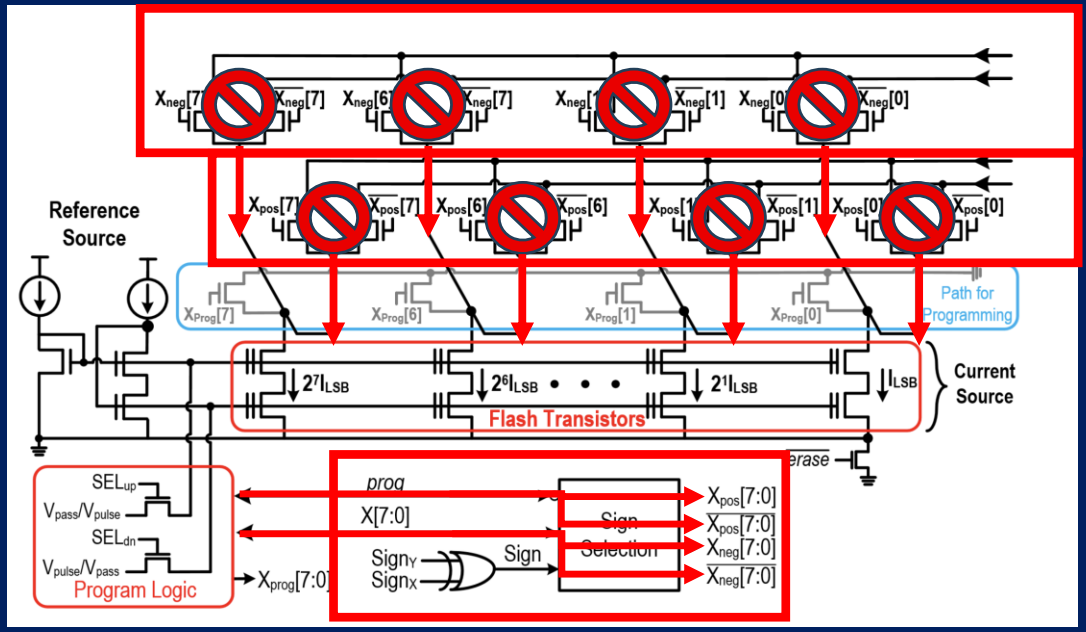
FCM - Operation

- The FCM consists of 8 identical current branches to implement **an 8-bit unsigned multiplication**.
  - Assume the FCM multiplies X[7:0] and Y[7:0], with **X[7:0] from register banks** and **Y[7:0] as the FCM coefficient stored in the flash transistors**.
- The FCM employs the same size of flash transistors to implement binary-weighted currents.
  - We adjust the V_t of the flash transistor in the i^{th} current branch to ensure its current is $2^i * I_{LSB}$.
  - To implement arbitrary coefficient values for **8-bit multiplication**, **256 I_{LSB} values** are required for the FCM.



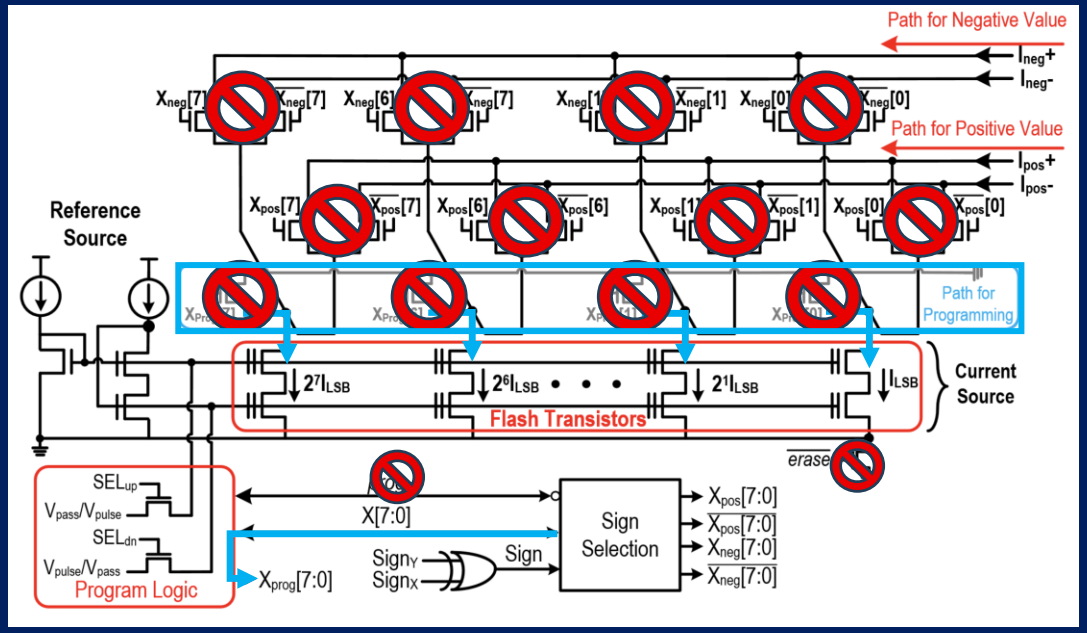
FCM - Operation

- To implement signed multiplication, we implement **another differential pair for each current branch** and use **a sign selection circuit** to enable the FCM to perform **signed multiplication**.
  - A signal Sign = $Sign_X \oplus Sign_Y$ to determine whether the output of the FCM is positive or negative.
    - If **Sign = 1**, the sign selection circuit **drives X_{neg}[7:0] to X[7:0]** and **drives X_{pos}[7:0] to 0.**
      - The FCM current flows towards I_{neg}+ and I_{neg}-.
    - If **Sign = 0**, the sign selection circuit **drives X_{pos}[7:0] to X[7:0]** and **drives X_{neg}[7:0] to 0.**
      - The FCM current flows towards I_{pos}+ and I_{pos}-.



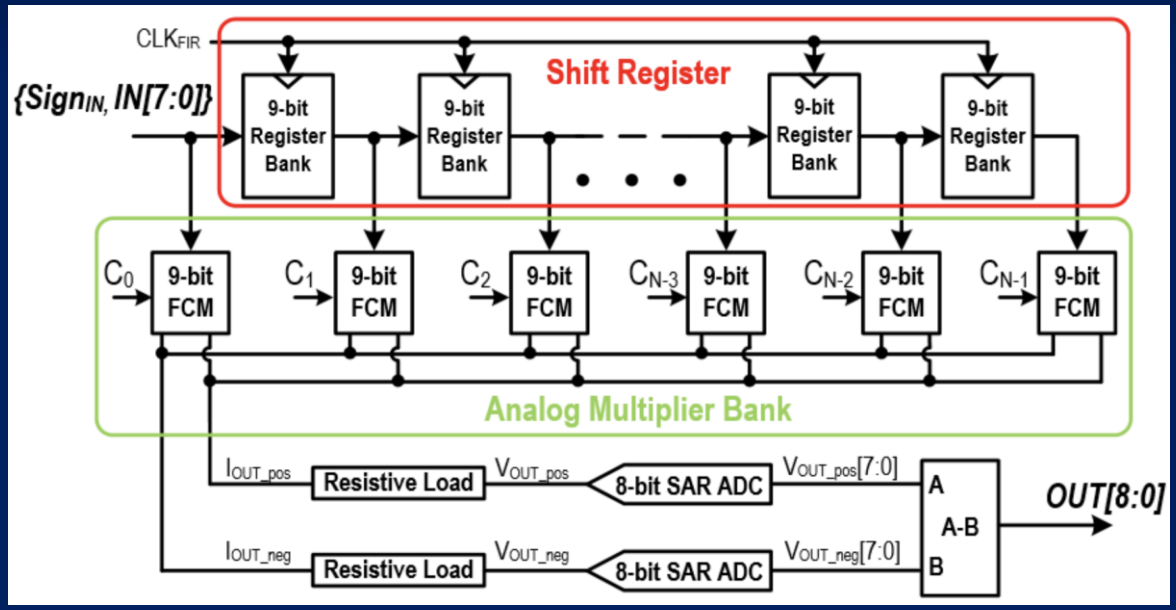
FCM - Programming

- In order to adjust the drain current of any current branch to the desired current level, we add another path for programming.
- **Program (*prog* = 1, *erase* = 0)**
  - Drives $X_{prog}[7:0]$ to X[7:0], while $X_{pos}[7:0]$, $X_{neg}[7:0]$, $\overline{X_{pos}[7:0]}$, and $\overline{X_{neg}[7:0]}$ are driven to 0.
  - Applies V_{pulse} (V_{pass}) to the gate of the selected (unselected) flash transistors.
- **Erasure (*prog* = 0, *erase* = 1)**
  - $X_{prog}[7:0]$, $X_{pos}[7:0]$, $X_{neg}[7:0]$, $\overline{X_{pos}[7:0]}$, and $\overline{X_{neg}[7:0]}$ are driven to 0.
  - Applies a high voltage (10 V – 20 V) to the bulk terminals.



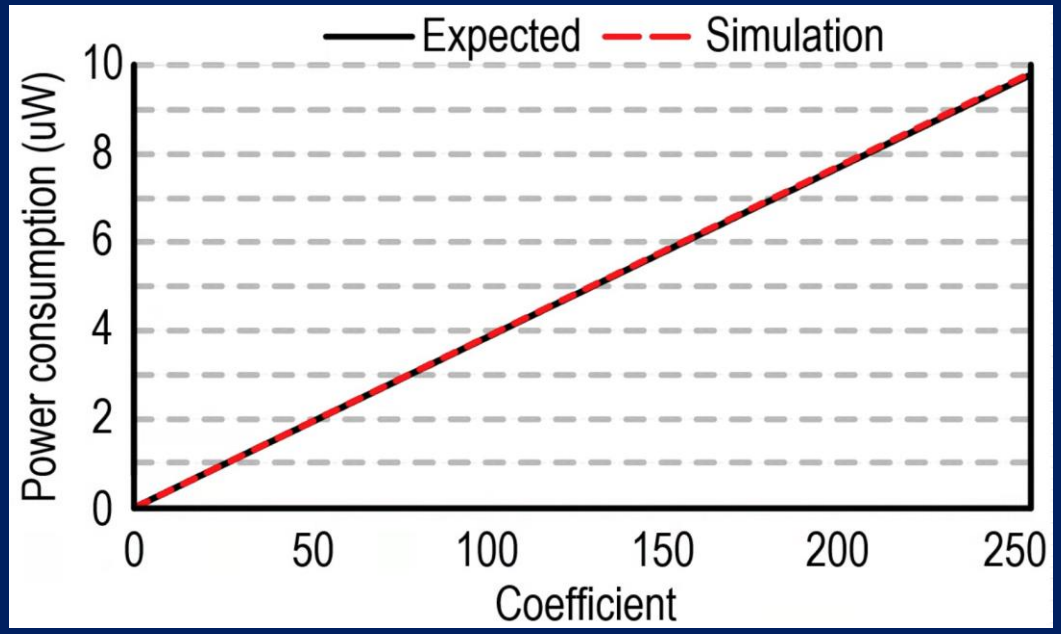
Simulation Environment Setup

- 45nm PTM CMOS and Flash process
- Synopsys HSPICE for circuit simulations
- Synopsys Design Compiler for synthesizing the digital circuits
- The coefficients for the filters we used in our designs are generated using MATLAB



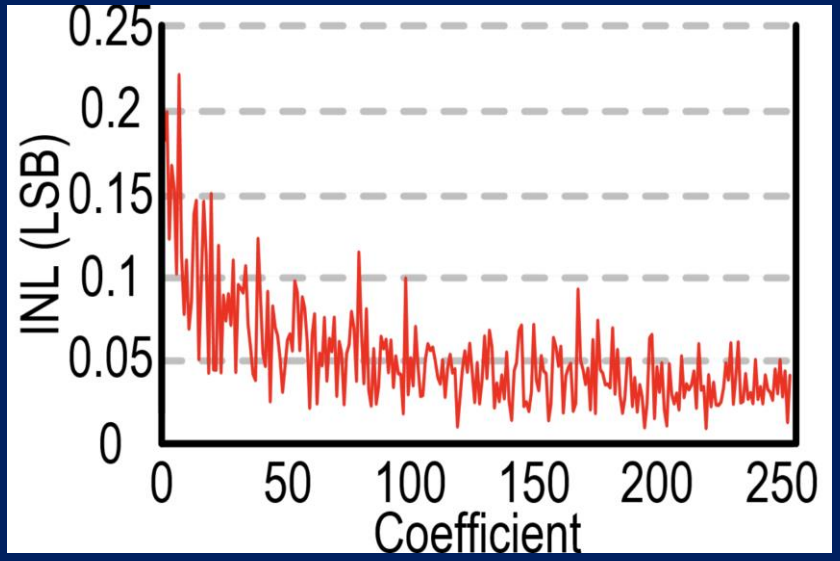
Performance of FCM - Power

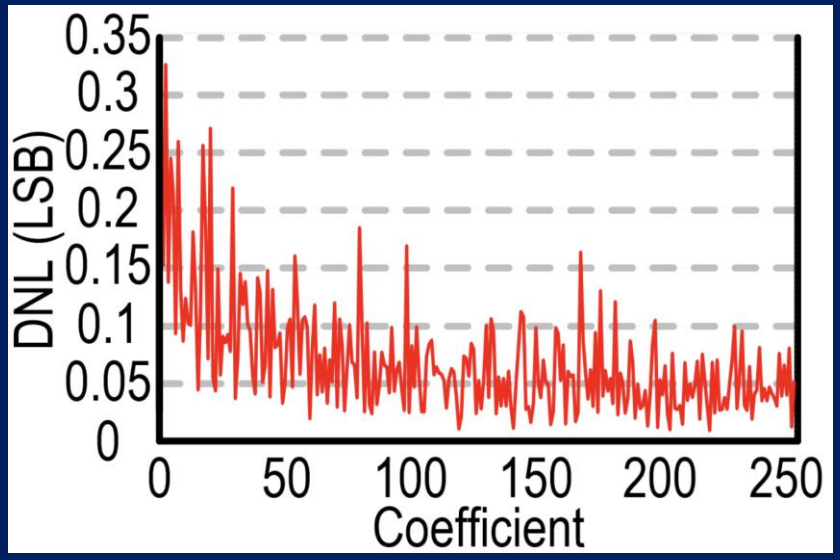
- The FCM utilizes 256 (8 bits) I_{LSB} values ranging from 0 to 31.875 nA in 0.125 nA increments, representing coefficients from 0 to 255.
- Power Consumption:
  - Increases linearly with the coefficient value.
  - The maximum power consumption is 31.875 nA×255×1.2 V=9.75 µW.
- The simulation results are very close to the expected results, with the maximum and average errors being **3.44%** and **0.12%**, respectively.



Performance of FCM – Linearity

- We evaluate the linearity of our FCM by measuring INL and DNL.
  - INL: **Maximum deviation** between the ideal and measured output **for any input code**.
  - DNL: **Maximum deviation** from the ideal LSB step between consecutive input codes.
- Results
  - Maximum INL (DNL) is 0.222 LSB (0.323 LSB).
  - Average INL (DNL) is is 0.052 LSB (0.069 LSB) with standard deviations of 0.033 LSB (0.048 LSB).
- These results indicate that our FCM behavior is **highly linear**.



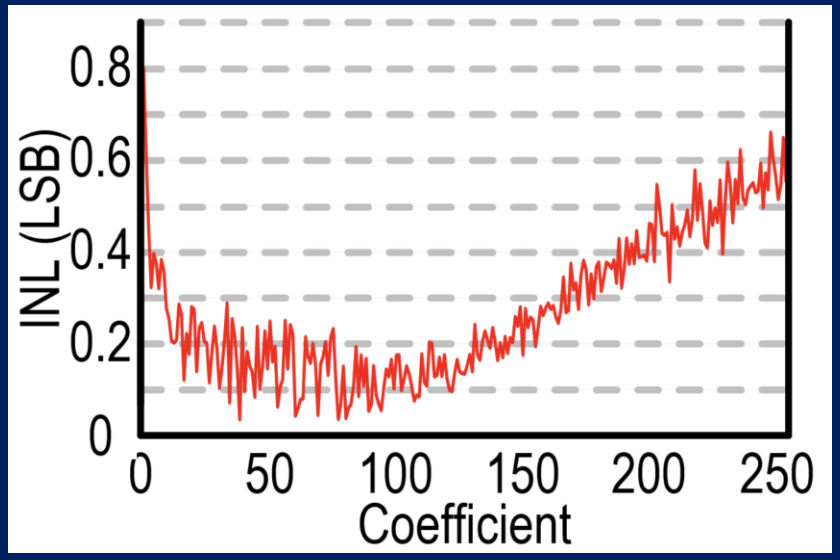


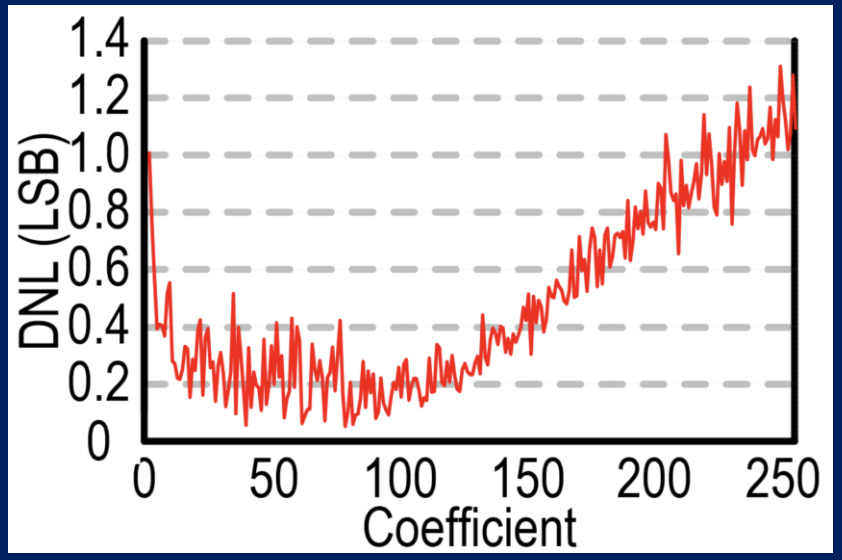
Performance of FCM – Linearity

- We perform Monte Carlo simulations to verify the functionality of the FCM under process and supply voltage (VDD) variations.

Device	Parameter	Standard Deviation (σ)
NMOS/PMOS	W, L [40]	2.6nm
	V_t [41]	$3.19 * 10^{-8} * \frac{t_{ox} * N_a^{0.4}}{\sqrt{L_{eff} W_{eff}}}$
V_{DD}	V	3.3% of nominal V_{DD}

- Results
  - Maximum INL (DNL) is 0.802 LSB (1.323 LSB).
  - Average INL (DNL) is 0.276 LSB (0.489 LSB) with standard deviations of 0.158 LSB (0.327 LSB).
- Though INL and DNL values increase due to process and VDD variations, our FCM is still **monotonic** for most coefficients.
  - The FCM demonstrates **robustness and tolerance** to process and VDD variations.





Performance of FFIR

- We compare 12 FFIR filter designs comprising lowpass, bandpass, and highpass designs with the corresponding synthesized DFIR filters.
  - The DFIR filters are synthesized using the fastest operating frequency.
- FFIR filters demonstrate significant improvements in **power**, **energy**, and **area.**
  - **Minimum Improvements**
    - Power: at least **3.29×** better
    - Energy: at least **1.34×** better
    - Area: at least **5.88×** better
  - **Average Improvements**
    - Power: **4.05×** better
    - Energy: **1.95×** better
    - Area: **6.06×** better

#taps	Filter Type	FFIR Filter				DFIR (Fastest Operating Frequency)				Ratio (DFIR/FFIR)			
		Latency (ns)	Power (μW)	Energy (pJ)	Active Area (μm^2)	Latency (ns)	Power (μW)	Energy (pJ)	Active Area (μm^2)	Latency	Power	Energy	Active Area
31 taps	Lowpass	5.4	1166	6.30	294	2.2	3835	8.44	1727	0.41×	3.29×	1.34×	5.88×
	Bandpass		1163	6.28							3.30×	1.34×	
	Highpass		1167	6.30							3.29×	1.34×	
71 taps	Lowpass	5.4	1845	9.96	646	2.6	8017	20.8	3959	0.48×	4.35×	2.09×	6.13×
	Bandpass		1842	9.95							4.35×	2.10×	
	Highpass		1846	9.97							4.34×	2.09×	
111 taps	Lowpass	5.4	2523	13.6	998	2.7	10944	29.6	6061	0.50×	4.34×	2.17×	6.08×
	Bandpass		2520	13.6							4.34×	2.17×	
	Highpass		2523	13.6							4.34×	2.17×	
128 taps	Lowpass	5.4	2810	15.2	1147	2.8	11866	33.2	7082	0.52×	4.22×	2.19×	6.17×
	Bandpass		2808	15.2							4.23×	2.19×	
	Highpass		2811	15.2							4.22×	2.19×	
Average										0.48×	4.05×	1.95×	6.06×

Performance of FFIR – PSNR & ENOB

- We quantify the usefulness of our FFIR filters with two metrics.
  - **Peak signal-to-noise ratio (PSNR)**
  - **Effective number of bits (ENOB)**
- Among 12 designs:
  - PSNR exceeds **31.56 dB** (Average PSNR is **38.04 dB**)
  - ENOB surpasses **8.76 bits** (Average ENOB **is 8.87 bits**)
- These results meet real-world application requirements (PSNR ≥ 30 dB).

#taps	Filter Type	PSNR (dB)	ENOB (bit)
31 taps	Lowpass	41.57	8.92
	Bandpass	40.79	8.92
	Highpass	34.54	8.83
71 taps	Lowpass	45.0	8.95
	Bandpass	34.64	8.83
	Highpass	39.15	8.90
111 taps	Lowpass	42.72	8.93
	Bandpass	32.41	8.78
	Highpass	36.82	8.87
128 taps	Lowpass	41.28	8.92
	Bandpass	31.56	8.76
	Highpass	36.03	8.86
Average		38.04	8.87

Comparison

- We compare our FFIR filters with 4 prior works aimed at low-power IoT applications
  - **Latency**: at least **18.5× reduction**
  - **Energy per Tap**: at least **1.3× reduction**
  - **Area**: at least **5.3× smaller**

	This work				JSSC'20 [7]	SSC-L'24 [6]	CICC'18 [5]	A-SSCC'20 [4]
Type	Mixed-Signal				Mixed-Signal	Mixed-Signal	Mixed-Signal	Mixed-Signal
Process	45nm				22nm	28nm	65nm	40nm
Input/Output Signal Type	Digital				Analog	Analog	Analog/Digital	Analog/Digital
Voltage	1.2V(FCM, Logic) 1.5V(Comp, S/H)				0.7V	0.8V	1.0V(ADC) 1.2V/1.6V (Filter)	1.1V
Coefficients	Tunable				Tunable	Tunable	Tunable	Tunable
Design	31-tap	71-tap	111-tap	128-tap	128-tap	96-tap	15-tap	13-tap
Total Power (mW)	1.17	1.85	2.52	2.81	0.092*	0.356*	0.039	0.092
Latency (ns)	5.4				2000*	2000*	100	100
Energy/tap (pJ)	0.20	0.14	0.12	0.12	1.44*	7.42*	0.26	0.71
Area (mm^2)	0.0003	0.0006	0.0010	0.0011	0.09*	0.05*	0.0061	0.067

* Excludes ADC operation.

Conclusion

- We propose **a mixed-signal Flash-based Finite Impulse Response (FFIR) filter** for IoT applications, achieving **low power**, **energy**, and **area** compared to previous approaches.
- Flash-based Coefficient Multipliers (FCMs):
  - Coefficients are stored in the form of threshold voltages (V_t) of the flash transistors.
  - The use of flash transistors eliminates the need for binary-weighted transistor sizing.
  - PVT variations and aging effects can be mitigated by adjusting the Vt of flash transistors.
- Performance Compared to Digital FIR (DFIR) Filters:
  - Reduces power by 4.05×
  - Reduces energy by 1.95×
  - Reduces area usage by 6.06×
- Performance Compared to the best of 4 recently Analog FIR (AFIR) Filters:
  - Decreases latency by 18.5×
  - Reduces energy per tap by 1.3×
  - Reduces area by 5.3×

Thank You!