Hybrid Temporal Computing for Lower PowerHardware Accelerators

Maliha Tasnim, Sachin Sachdeva, Yibo Liu, Sheldon Tan

VLSI Systems and Computation Lab

Department of Electrical and Computer Engineering

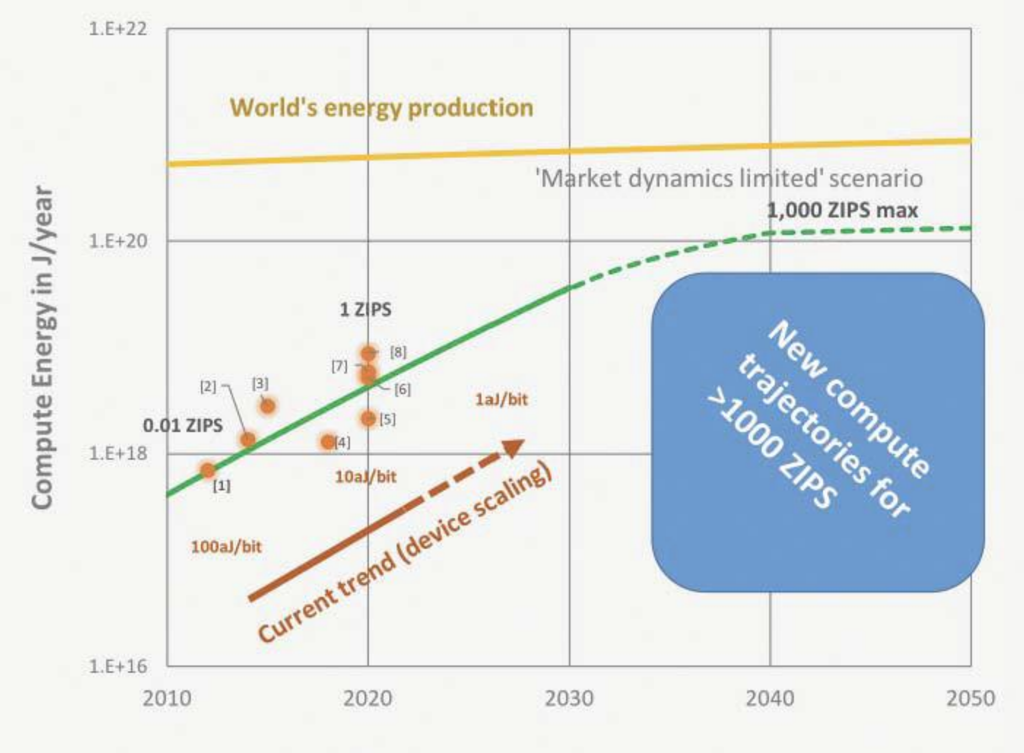
University of California, Riverside

Outline

- Introduction and challenge
  - Energy consummation for Gen AI is fundamental challenging
- Review of Temporal Computing and Stochastic Computing
- The proposed Hybrid Temporal Computing
  - The HTC Data Coding
  - The HTC Multiplication and Addition operations
  - The HTC MAC and 4-bit MAC design
- Experimental results and discussions
- Conclusion and takeaway

The Growing Energy Challenge

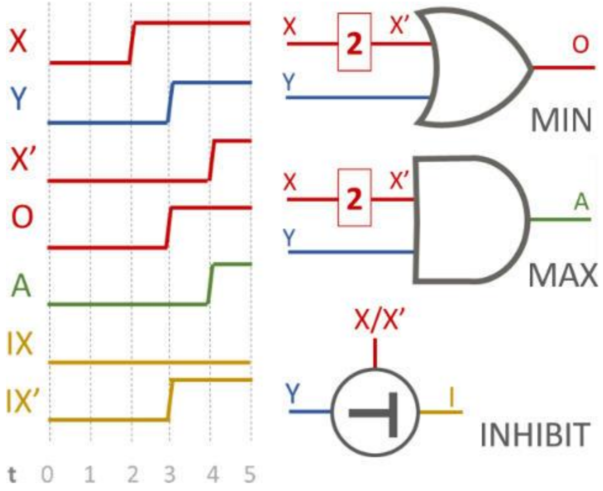
- Exponential growth in computing power demands due to emerging Gen AI.
- Linear growth in power supply (approx. 2% per year).
- Computing energy consumption projected to double every three years
- Need for new, ultra-low-energy computing paradigms



Courtesy of SRC *The decadal plan for semiconductors*

Introduction to Temporal Computing and Race Logic

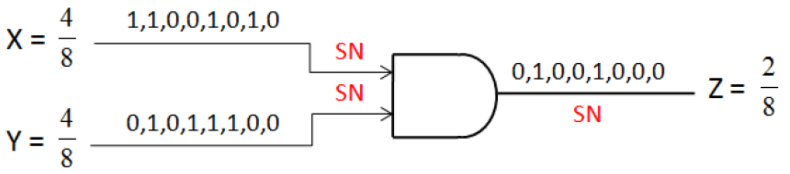
- **Temporal computing** is a promising approach for reducing energy consumption.
- It's rooted in the concept of **race logic**, where information is encoded in the timing of voltage transitions rather than individual bits.
- Multiple bits of information can be encoded on a single wire.
- Race logic sacrifices some precision but offers advantages in speed, energy efficiency, and reduced area.
- However **pure temporal computing (TC)**, which uses race logic, struggles with performing general arithmetic operations like multiplication and addition due to causality and waveform format restrictions
- TC for general-purpose computing remains a **challenging** problem.
    - Despite recent efforts to develop temporal state machines and temporal memory structures, race logic-based design for general computing still remains challenging



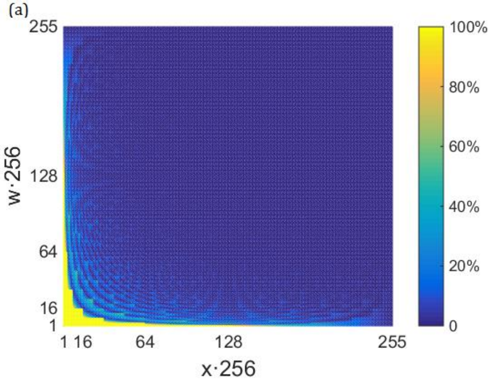
Courtesy of NIST, In “race logic” information is not encoded in these bits but rather in the **time or delay** at which a voltage changes from low to high.

Stochastic Computing (SC) and CBSC

- **Stochastic computing (SC)** represents values as probabilities in a bit stream or pulse rate.
  - SC offers a trade-off between accuracy and latency/energy/area.
  - Traditional SC implementations can suffer from long latency and large area overhead.
- **Counting-based SC (CBSC)** is a more efficient and accurate SC multiplier that addresses some limitations of traditional SC.
  - CBSC replaces the AND operation with a counting process, and bit streams no longer need to be random.
- **Limitations:** Traditional SC can suffer from long latency and large area overhead, while CBSC still performs addition in binary format



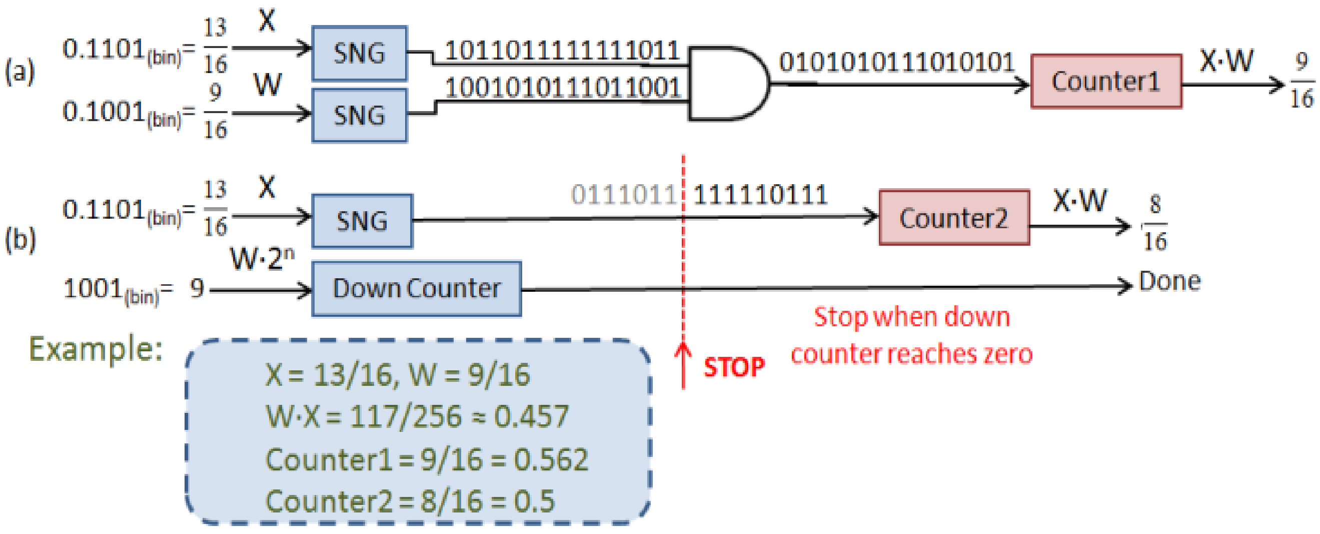
SC Multiplication core is a simple *AND* gate



SC suffer large errors for small numbers

Review of Counter-based SC

Binary data converted to binary bitstream



$X = 13/16$, $W = 9/16$

$W \cdot X = 117/256 \approx 0.457$

$\text{Counter1} = 9/16 = 0.562$

$\text{Counter2} = 8/16 = 0.5$

Multiplication is performed in AND/XNOR, addition is performed with OR/MUX

CBSC- Eliminates the necessity of randomness of bitstream

Replaces expensive SNG with a FSM based bitstream generator

Has evolved beyond SC – Deterministic computing

- A special form of temporal computing

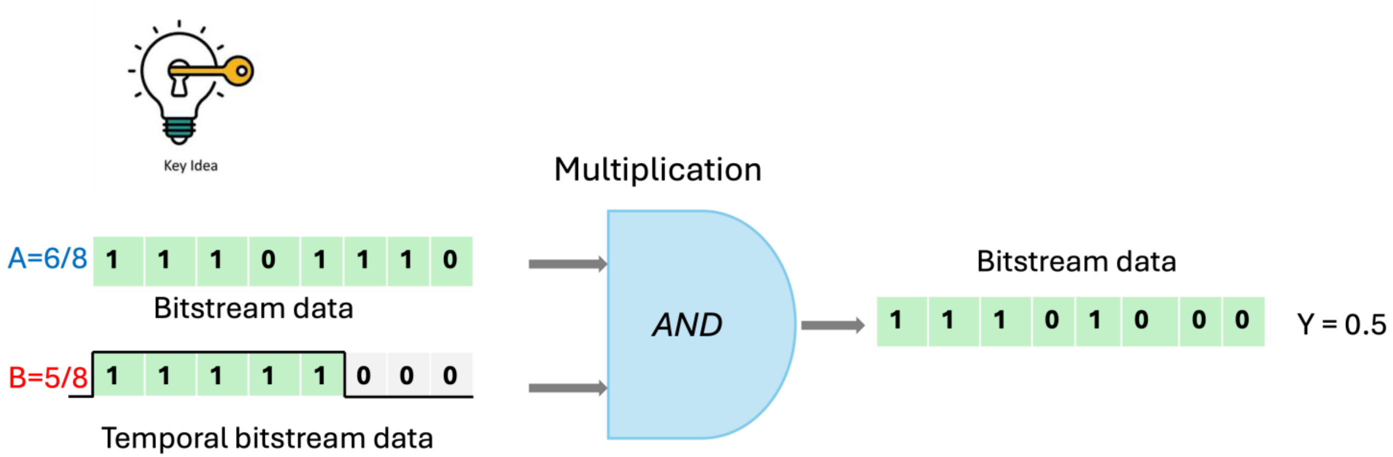
Hybrid Temporal Computing: A New Framework

- Leverages both bitstream (pulse rate) and temporal data encoding.
  - Both pulse rate and temporal encoding data are still bitstreams suitable for SC

- Encodes data in temporal and traditional bitstream (pulse rate) formats.
  - Minimizes switching activities while retaining energy efficiency of stochastic computing.

- Temporal data format further reduces energy consumption for signal propagation.

- Can be viewed as a generalized CBSC framework.



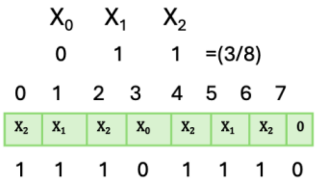
A=6/8: 1 1 1 0 1 1 1 0

B=5/8: 1 1 1 1 1 0 0 0

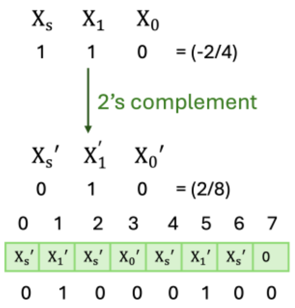
Y: 1 1 1 0 1 0 0 0

Data Encoding in HTC

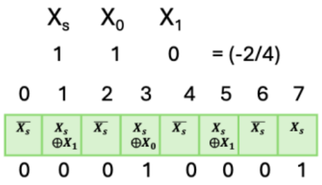
- Two formats: **General Bitstream (GB)** and **Temporal Bitstream (TB)**.
- **GB:** Value represented by number of '1' bits, similar to SC.
- **TB:** Value represented by the time period or delay relative to a reference signal.
- Data values can be unipolar ([0,1]) or bipolar ([-1,1]).
- **Regulated Bitstream (RB)**: Generated using a finite state machine, evenly distributing bits.



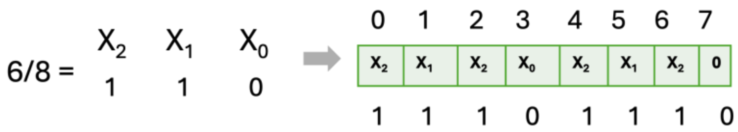
RB : Unipolar Regulated bitstream



RB : Bipolar Regulated bitstream


X_s X_0 X_1
1 1 0 = (-2/4)
0 1 2 3 4 5 6 7
$\overline{X_s}$ $X_s \oplus X_1$ $\overline{X_s}$ $X_s \oplus X_0$ $\overline{X_s}$ $X_s \oplus X_1$ $\overline{X_s}$ X_s
0 0 0 1 0 0 0 1

RB : Bipolar Regulated bitstream



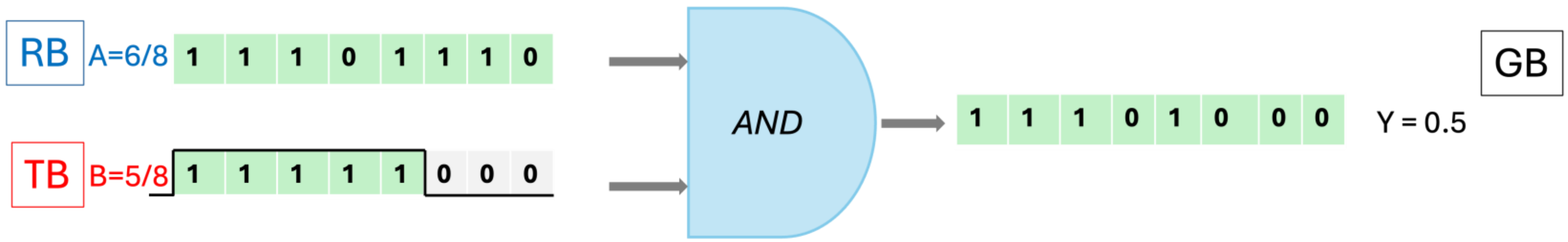
RB : Regulated bitstream

GB : General bitstream

RB : Regulated bitstream

TB : Temporal bitstream

Multiplication

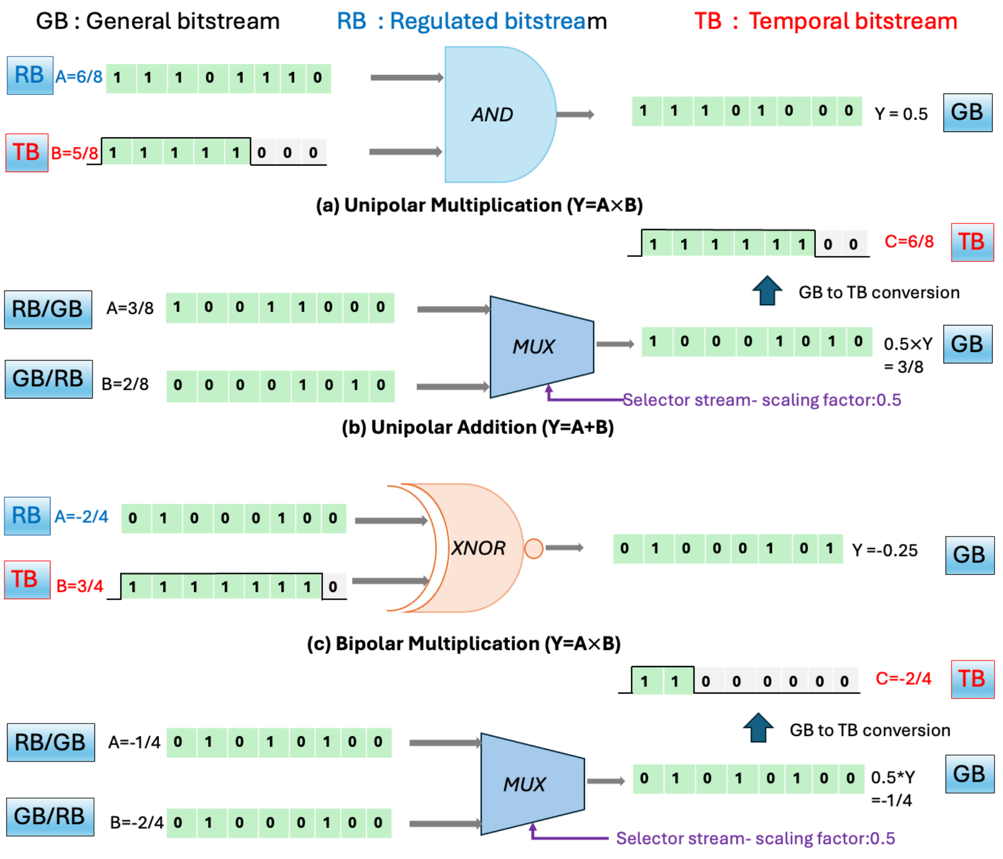


HTC Multiplication and Addition

Multiplication: Performed using AND (unipolar) or XNOR (bipolar) gate with RB and TB data.

Addition: Scaled addition with a MUX gate.

Temporal data can be naturally converted back to binary with a counter or shift register



(a) Unipolar Multiplication (Y=A×B)

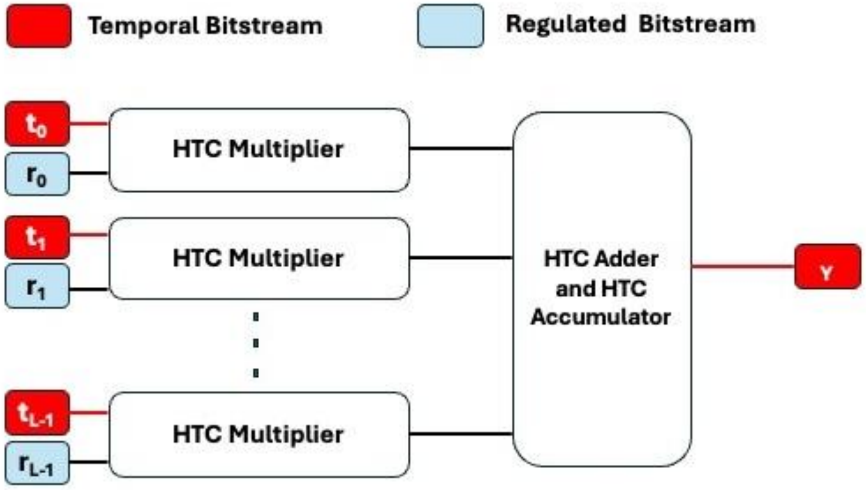
(b) Unipolar Addition (Y=A+B)

(c) Bipolar Multiplication (Y=A×B)

(d) Bipolar Addition (Y=A+B)

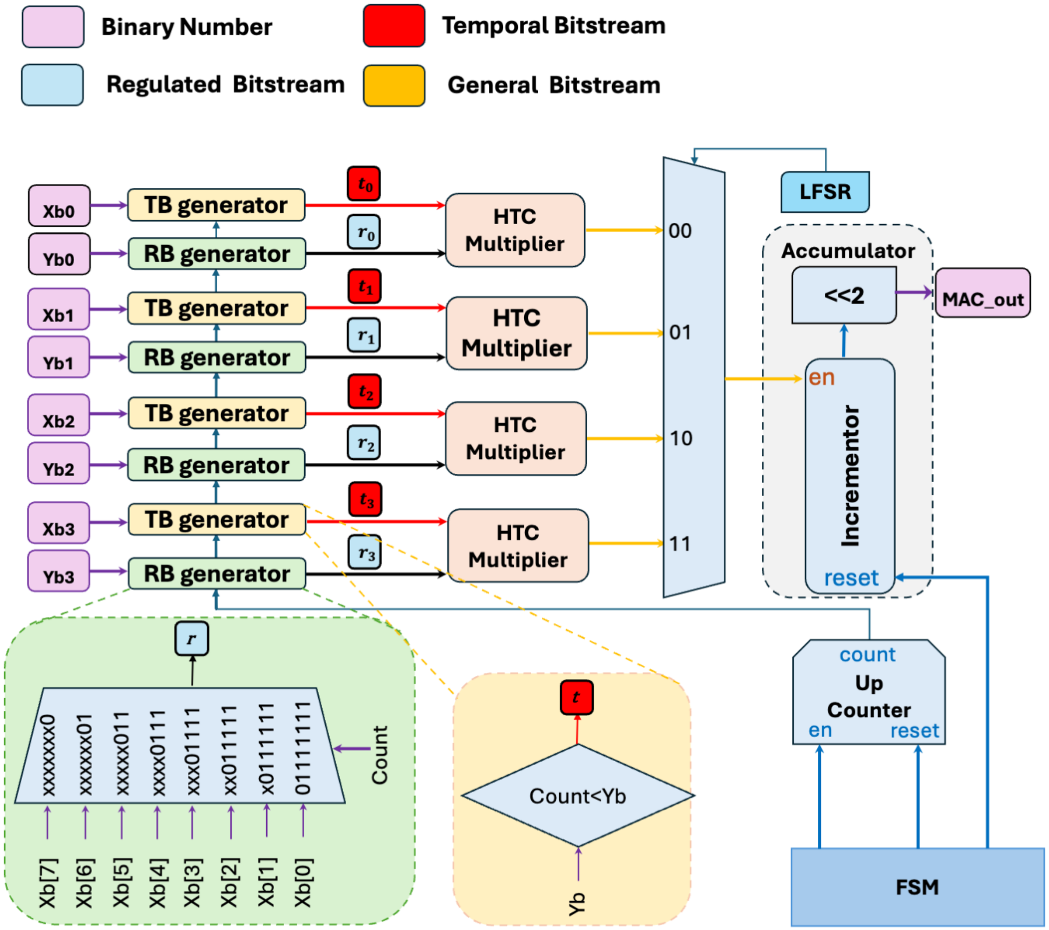
HTC Multiplier-Accumulator (MAC)

- MAC operation:
  - $a = \sum_{i=1}^{N}(b_i \times c_i)$
- Each multiplier takes one input in RB and another in TB format.
- Output is encoded in TB format for subsequent computations.



Specific 4-bit MAC Design

- N-bits, 4x1 MAC
  - Adder scale factor- 0.25
    - Mux selector has inherent scaling of 0.25
    - One shifter is used at the end for rescale by 4
  - Bitstream generation - by FSM and multiplexer
  - One common counter for accumulator + FSM



Experimental Result – Multiplication & Accumulation

MAC PE	Vector size	Area (μm^2)	Power (μW)	Latency (ns)	Average Error (%)	RMSE(%)	SDE(%)
CBSC MAC[1]	4	1476.46	56.69	2560	0.49	0.65	0.42
Unary MAC[2]	4	249110.23	3085.00	870400	39.31	40.64	8.85
HTC MAC	4	736.95	31.07	2560	5.33	6.96	4.46

- Compared with Unary MAC and CBSC MAC.
- **HTC MAC reduces power consumption by 45.2% and area footprint by 50.13% compared to the CBSC MAC**.
- Orders of magnitude faster and significantly smaller power and area footprints compared to Unary MAC.
- HTC MAC is more accurate than Unary MAC.
- HTC MAC is less accurate than the CBSC design due to approximate scaled addition

Experimental Result – 6 Tap Finite Impulse Response Filter

- Implemented a 6-tap FIR filter using HTC, Unary, and CBSC MACs.

- **HTC design reduces power consumption by 36.61% and area cost by 45.85% compared to CBSC design**.

- HTC and CBSC significantly outperform the Unary design across all metrics.

- HTC delivers comparable PSNR and RMSE to CBSC design.

Original

Image	CBSC MAC[1]		Unary MAC[2]		HTC MAC	
	PSNR (dB)	RMSE	PSNR (dB)	RMSE	PSNR (dB)	RMSE
Boat	20.20	0.10	9.75	0.325	20.00	0.10
Man	21.75	0.08	12.14	0.247	21.31	0.09
Couple	20.59	0.09	10.39	0.301	20.08	0.10
Bridge	18.67	0.12	10.30	0.305	18.67	0.12
Clock	17.64	0.13	6.02	0.499	17.61	0.13
Hardware Cost	**Area** (μm^2)	**Power** (μW)	**Area** (μm^2)	**Power** (μW)	**Area** (μm^2)	**Power** (μW)
	2091.36	62.61	8.2× 10^5	10997	1216.21	39.96

Experimental Result – 8-Point Discrete Cosine Transform

- Implemented 8-point DCT using HTC and CBSC MACs with bipolar encoding.

- **HTC-based DCT filter consumes 23.34% less power and occupies 18.20% less area than CBSC MAC-based DCT filter**.

- HTC-based DCT filter retains the quality of the original image with a decent PSNR

Original

Image	CBSC MAC[1]		HTC MAC	
	PSNR(dB)	**RMSE**	**PSNR(dB)**	**RMSE**
Boat	38.96	2.89	22.19	19.89
Man	37.09	2.69	17.89	32.63
Couple	31.99	6.43	21.78	20.84
Bridge	37.06	3.59	21.39	21.80
Clock	30.95	7.25	21.25	21.70

Hardware Cost	**CBSC MAC[1]**		**HTC MAC**	
	Area (μm^2)	**Power (μW)**	**Area (μm^2)**	**Power (μW)**
	2532.71	81.64	2071.54	62.79

Conclusion and takeaway

- Hybrid Temporal Computing (HTC) is a novel approach that effectively combines temporal and pulse rate encoding to achieve low-power hardware acceleration.

- **HTC significantly improves energy efficiency by minimizing signal switching activity and simplifying hardware implementation**.
  - Compared to the CBSC MAC, the HTC MAC reduces power consumption by 45.2% and area footprint by 50.13%.
  - The HTC MAC is also significantly faster and smaller than the Unary MAC design.

- In Finite Impulse Response (FIR) filter design, **the HTC MAC-based FIR filter reduces power consumption by 36.61% and area cost by 45.85% compared to the CBSC design.**

- For Discrete Cosine Transform (DCT) filter design, **the HTC-based DCT filter consumes 23.34% less power and occupies 18.20% less area than the CBSC MAC-based DCT filter**, while maintaining a decent PSNR for image quality.

- The proposed HTC framework shows promising results in digital signal processing applications, including FIR filters and DCT/iDCT engines.