



SCU: An SFQ Co-Design for Bit-Vector Approximate Matching in Genome Alignment

Peiyao Qu; Haihang You; Guangming Tang

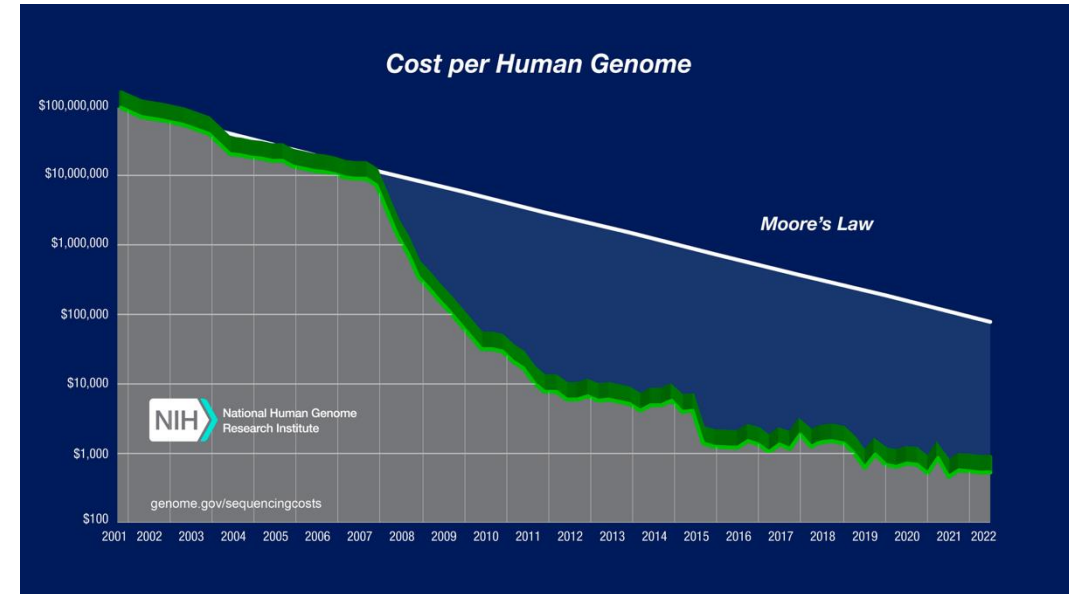
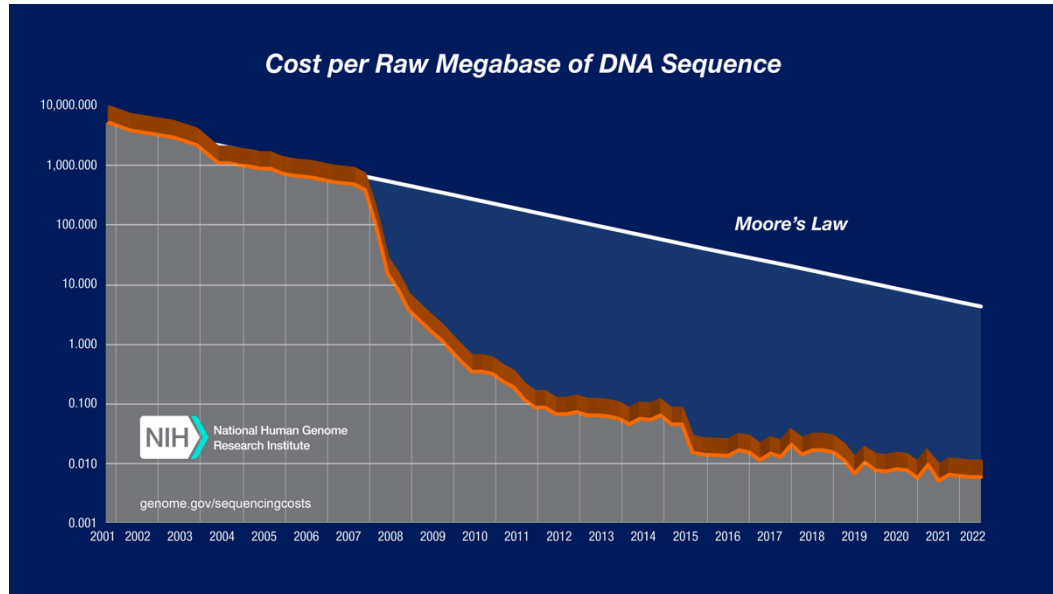
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

University of Chinese Academy of Sciences, Beijing, China

Email: qupeiyao@ict.ac.cn

Motivation: Sequencing costs have fallen faster than Moore's law

- Lower sequencing cost is shifting the bottleneck from data generation to genome analysis.

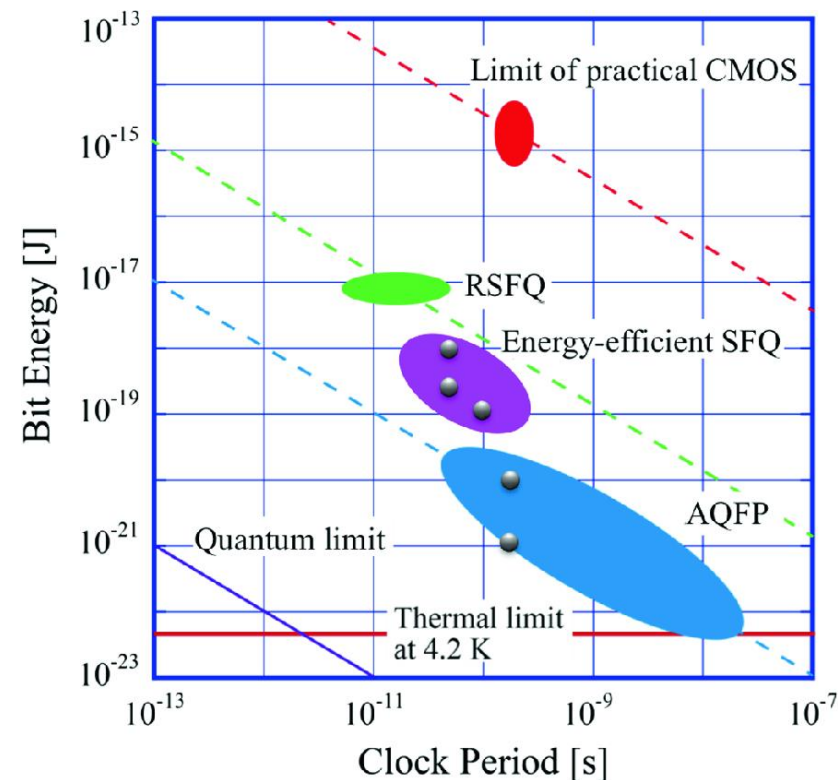
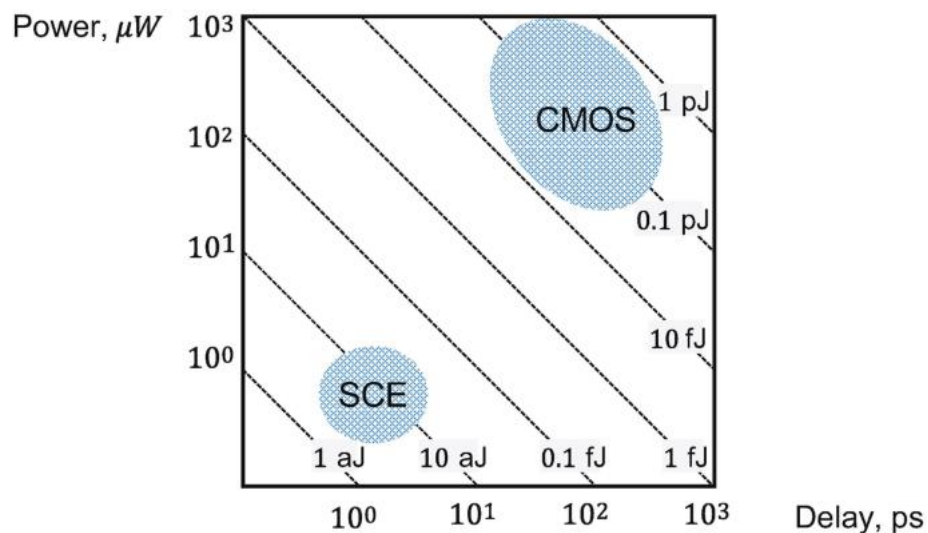
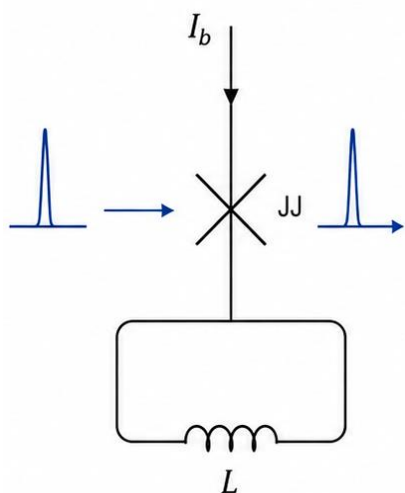


- More genomes can now be sequenced—but every additional read must still be aligned.
- Genome analysis therefore needs substantially higher throughput and energy efficiency.

Source: NHGRI, [DNA Sequencing Costs: Data \(genome.gov\)](https://www.genome.gov/sequencingcosts).

Motivation: SFQ logic occupies a high-speed, low-energy design space

- Superconducting electronics can achieve picosecond-scale switching and very low switching energy than CMOS.



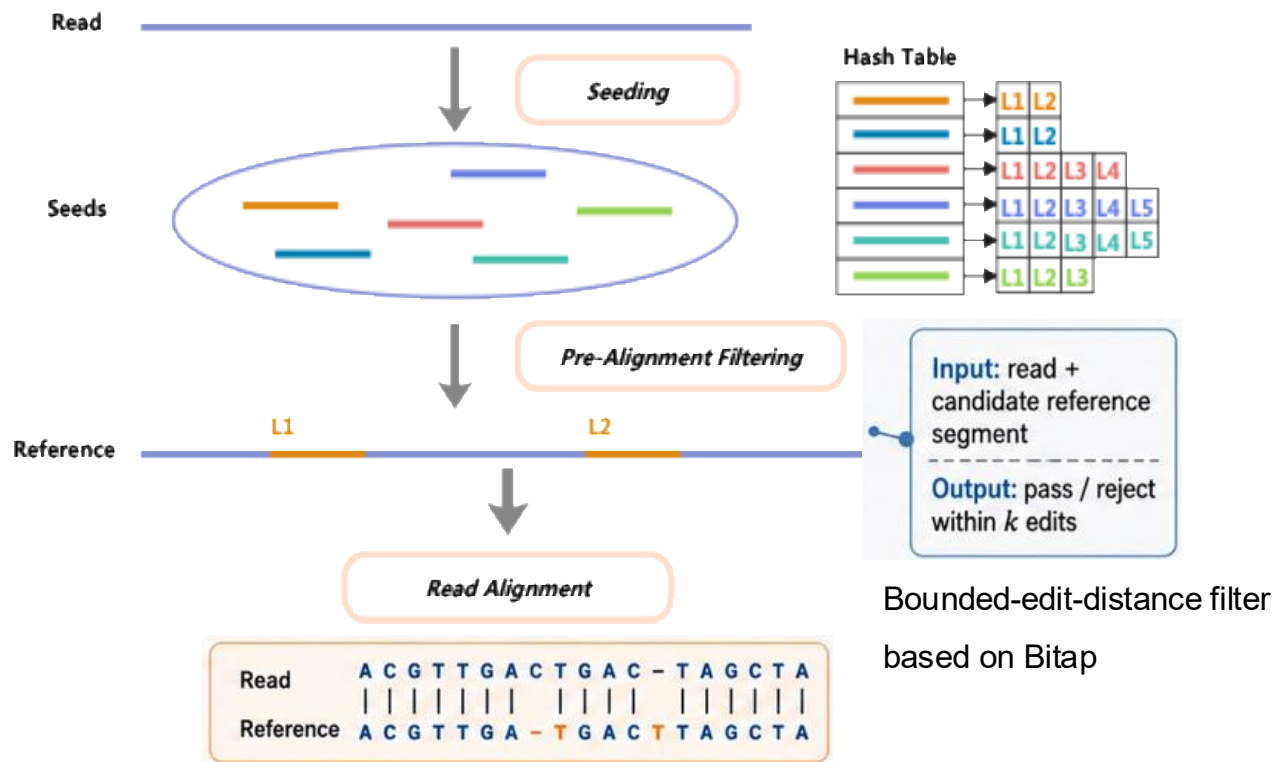
- Superconducting electronics shift the operating point toward both lower delay and lower power.

Source: Tafuri, Francesco, ed. Fundamentals and frontiers of the Josephson effect. Vol. 286. Springer Nature, 2019.
Krylov, G., Friedman, E.G. (2022). Introduction. In: Single Flux Quantum Integrated Circuit Design. Springer, Cham.

Motivation: Where Can SFQ Help Genome Alignment?

We target a kernel, not the entire aligner.

- A read-mapping pipeline mixes irregular memory/control with a regular compute kernel.



SFQ opportunity

- Repeated bit-vector state updates
- Dense, local bitwise operations
- Fixed and predictable dataflow
- Bounded state determined by edit distance

SFQ Target: Bitap-based bounded-edit-distance filtering

Conventional Bitap Algorithm

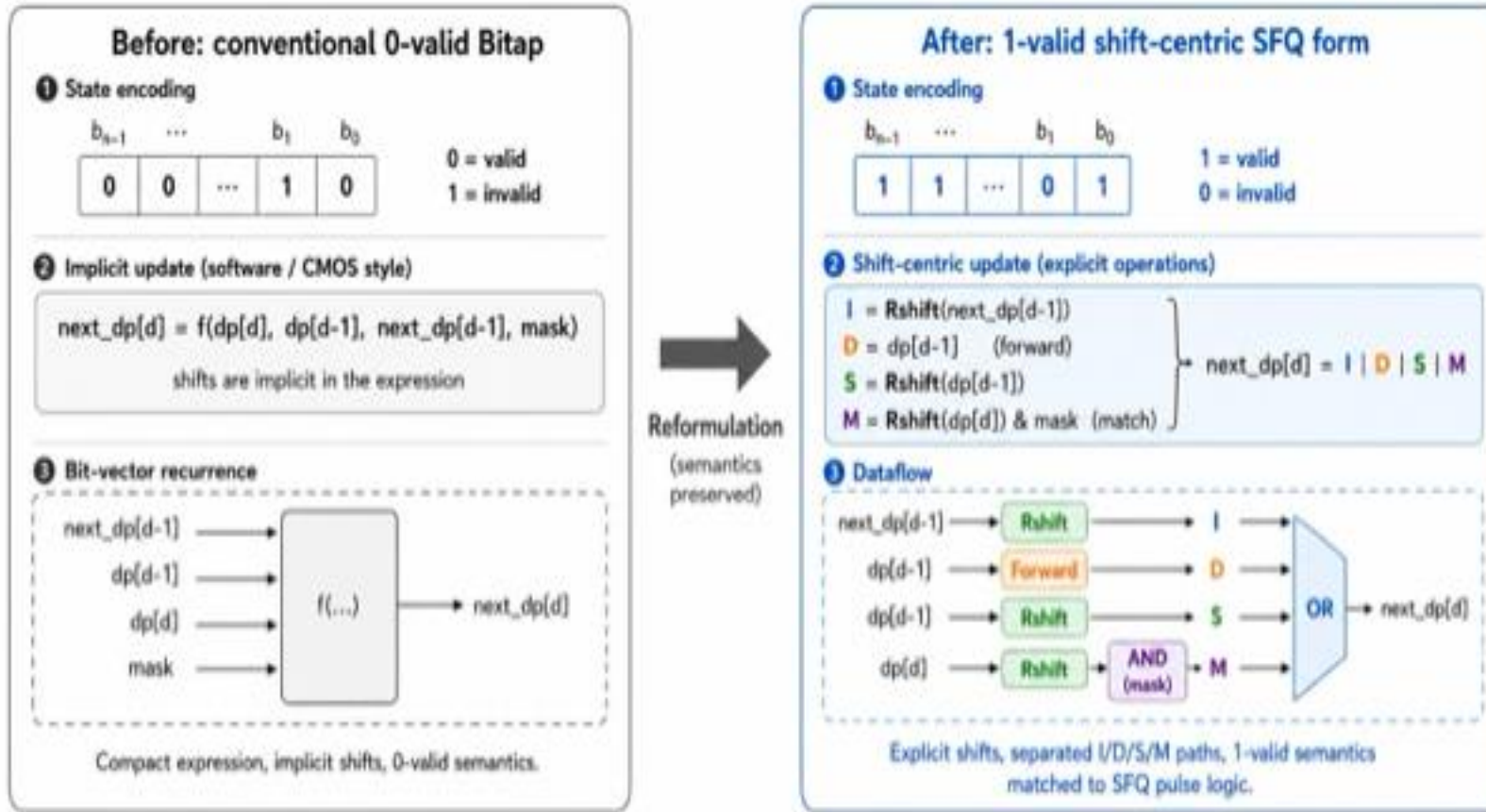
- For each new input character, Bitap updates one state vector per edit budget.



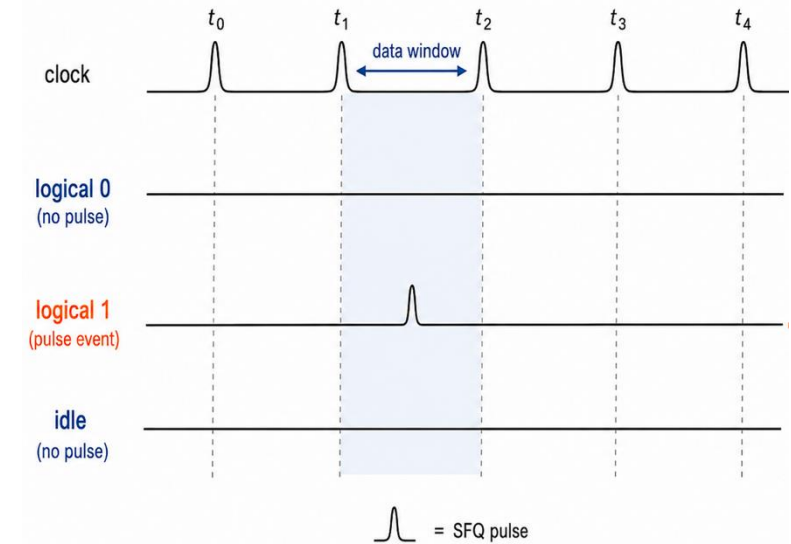
- Bitap merges them by bitwise AND under 0-valid encoding.

Algorithm reformulation for SFQ

- Same matching semantics, but a more explicit dataflow and a pulse-friendly state encoding.



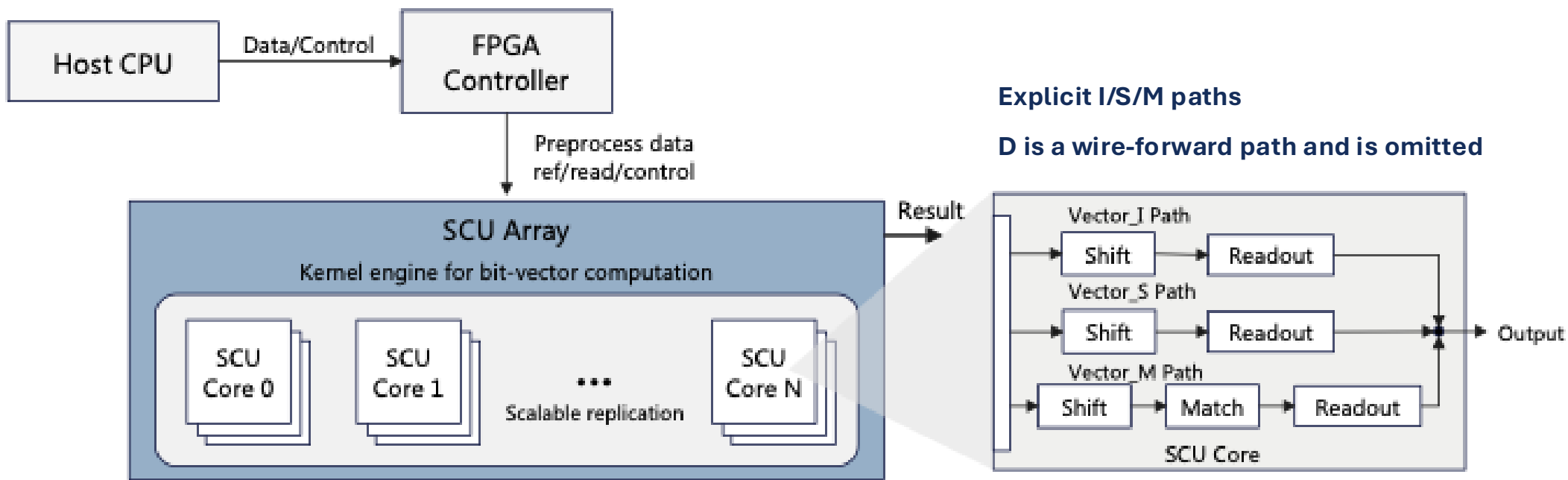
- '1' - valid state
- explicit *Rshift* operations



SCU Architecture: Replicated SCU Cores for Bitap State Updates

- The host and FPGA handle irregular orchestration, while the cryogenic SCU array executes regular bit-vector updates.

Candidate scheduling, data preparation, and control

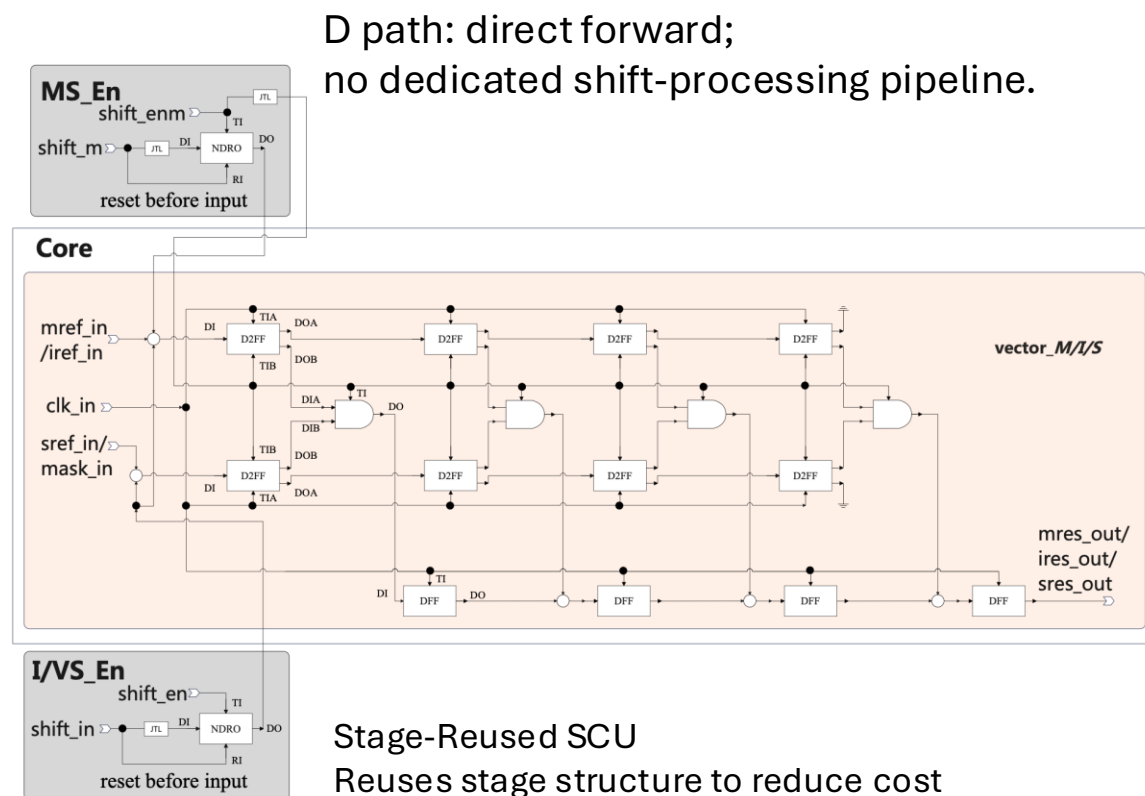
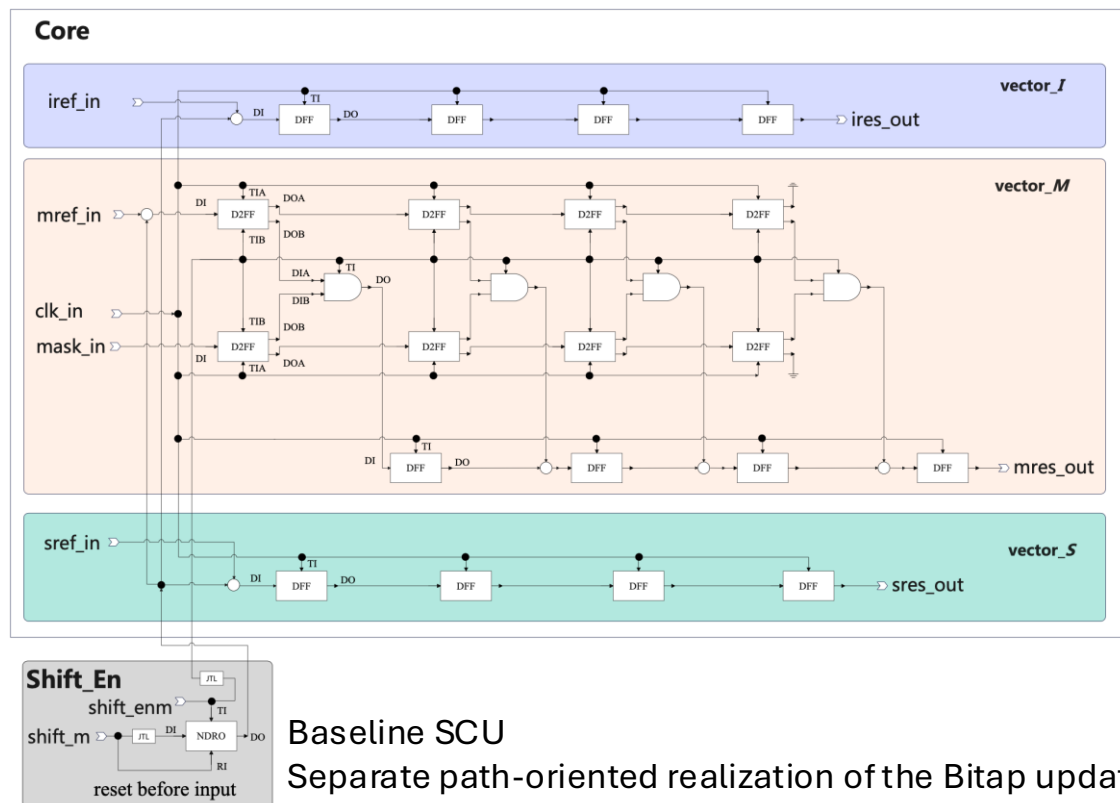


Replicated cores process multiple candidate regions in parallel

- Throughput scales by replicating a compact, fixed-dataflow kernel

Circuit Design Evolution: Baseline SCU vs. Stage-Reused SCU

- Explicit Bitap operations map to synchronized, stage-local SFQ pipelines.

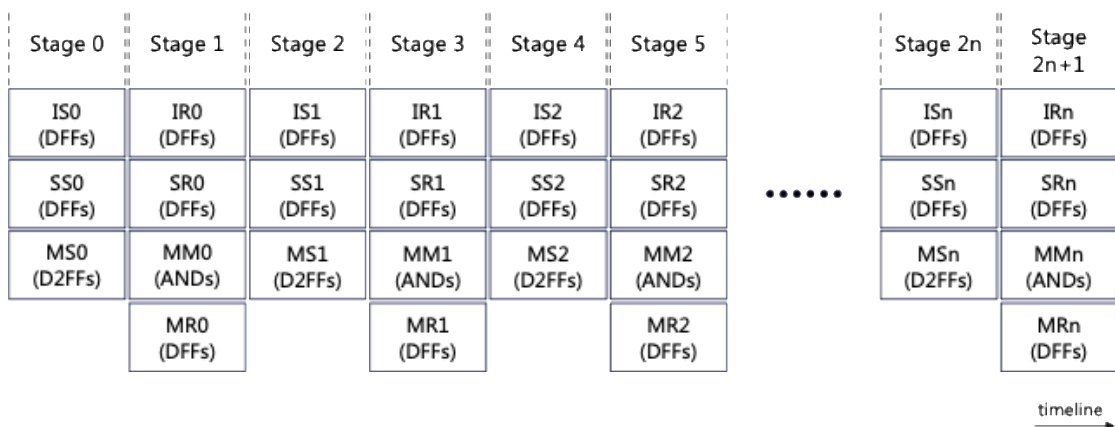


- Why it matters: Stage reuse improves circuit regularity and can reduce implementation.

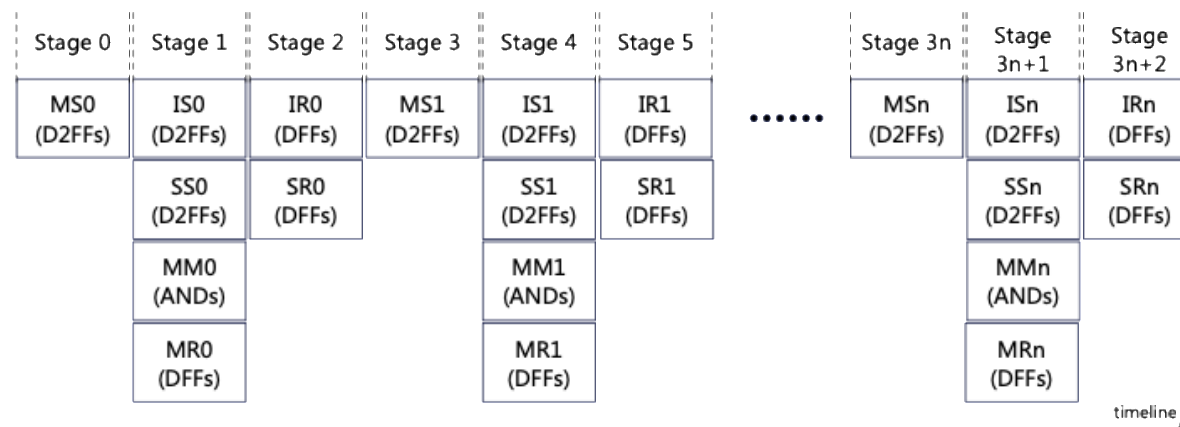
Stage Reuse Reduces the Per-Position Hardware Cost

- Both SCUs implement the same Bitap recurrence, but the Stage-Reused organization reduces replicated storage and logic.

Baseline SCU
Dedicated shift resources
Pipeline depth $\approx 2n + 1$



Stage-Reused SCU
Reused shift-stage resources
Pipeline depth $\approx 3n + 1$

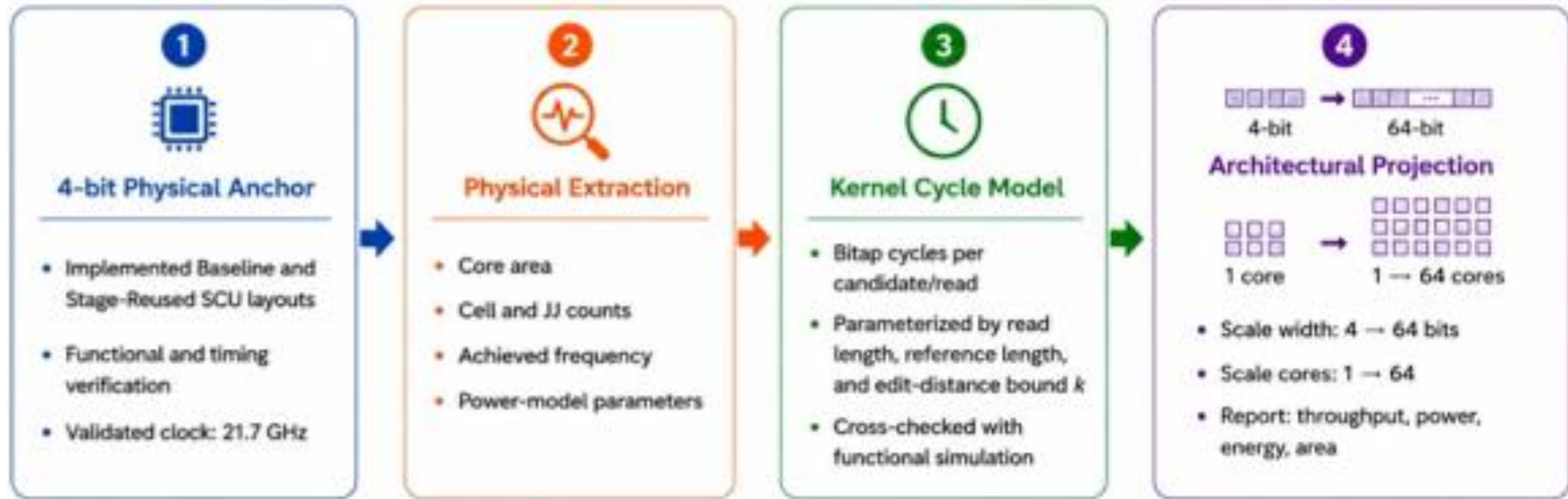


- IS and IR : the shift and readout stages of *vector I*
- SS and SR : denote the corresponding stages of *vector S*
- MS , MM , and MR : denote the shift, match, and readout stages of *vector M*

area–throughput tradeoff

Layout-calibrated evaluation methodology

- Implemented 4-bit SCUs calibrate the physical cost and timing of larger 64-bit multicore designs.



IMPLEMENTED / VERIFIED
4-bit layouts

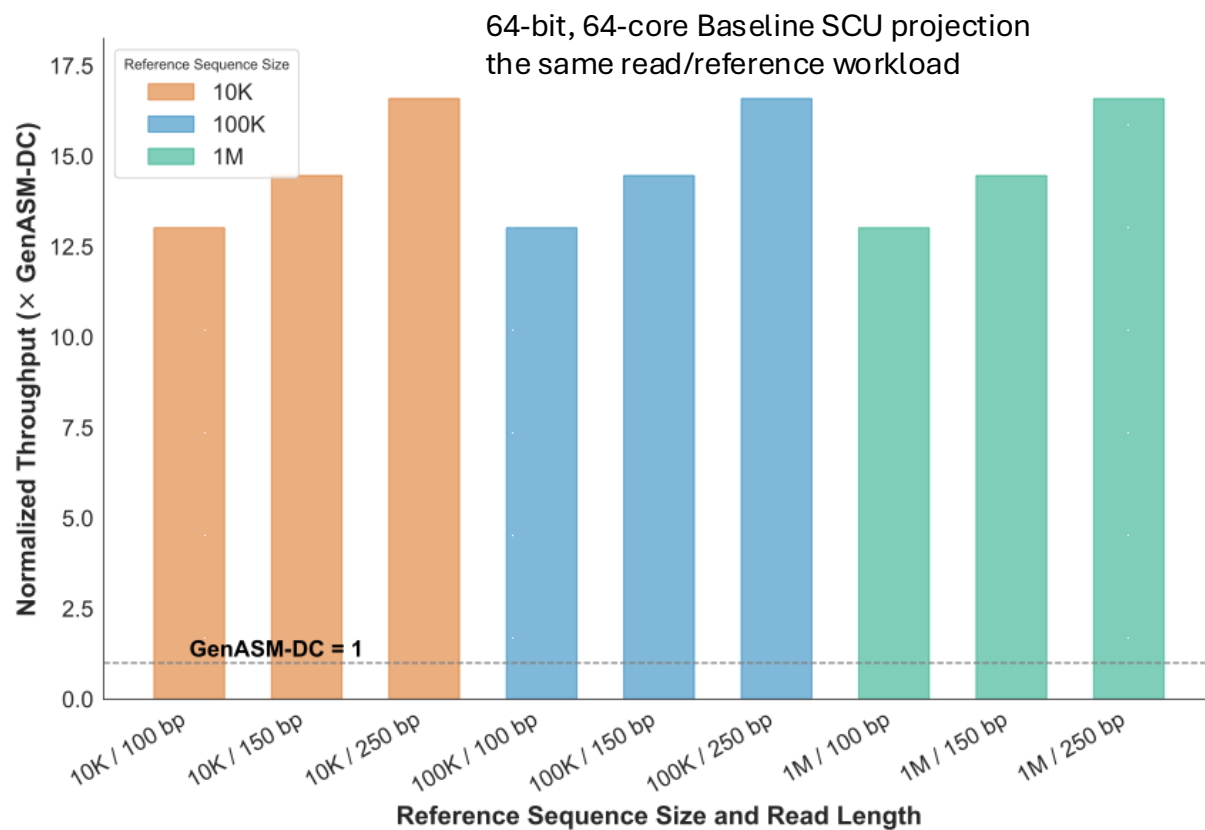
calibrated model

PROJECTED
64-bit / multicore kernel

- Physical cost and timing come from implemented layouts; only width and core-level scaling are projected.

SCU Achieves up to 16.6× Higher Kernel Throughput than GenASM-DC

- Speedup increases with read length and remains stable across 10K–1M reference sizes.



**Higher state-update throughput
across all evaluated configurations**

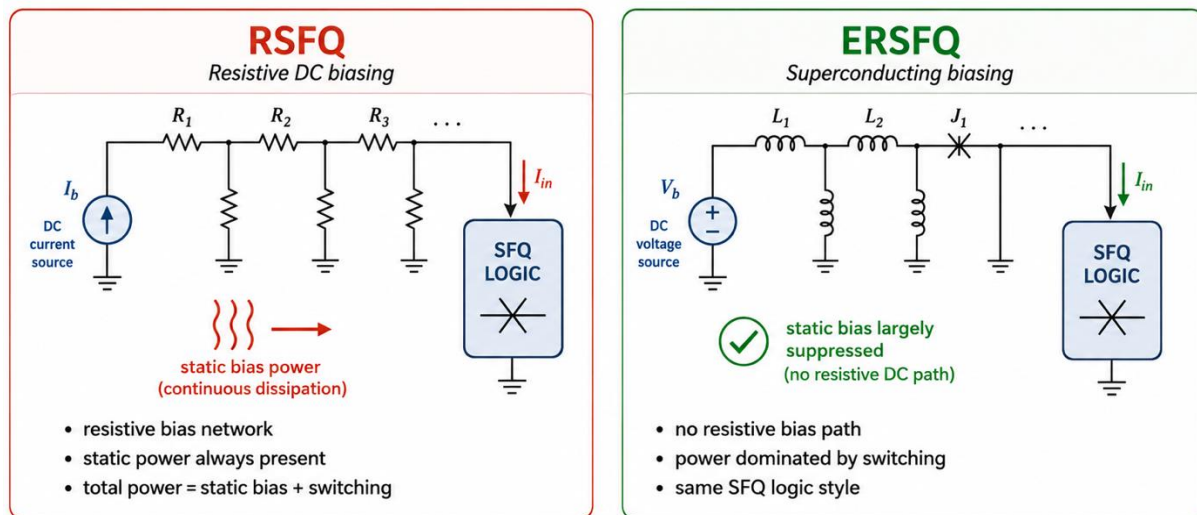
Larger gain for longer reads
14.0 $\times$ at 100 bp; 16.6 $\times$ at 250 bp

- Kernel-only comparison. End-to-end acceleration also depends on candidate generation, scheduling, traceback, and SFQ-CMOS I/O.

From RSFQ to ERSFQ: Static Bias Determines Energy Efficiency

- The SCU dataflow is unchanged; the biasing technology changes its power profile.

RSFQ vs. ERSFQ: Biasing and Static Power



Configuration	Norm. power	Norm. comp. energy efficiency
GenASM-DC	1.0×	1.0×
RSFQ-based SCU	~13.8×	~1.1×
ERSFQ-oriented SCU	~0.2×	~77.3×

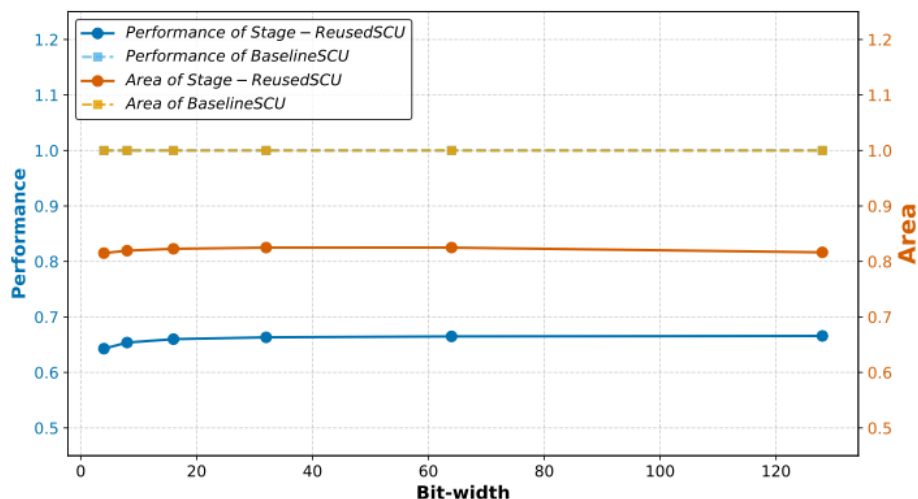
- The SCU dataflow provides the throughput; suppressing static-bias power determines whether that throughput translates into energy efficiency.

SCU Design Space: Peak Throughput vs. Lower Physical Cost

- All prior GenASM-DC comparisons use the Baseline SCU; Stage-Reused explores a lower-cost design point.

Relative throughput $\approx 0.67\times$

Relative area $\approx 0.81\times$



4-bit SCU variants

Metric	Baseline	Stage-Reused	Change
Update schedule	$2n+1$	$3n+1$	$\approx +50\%$
Rel. throughput	1.00×	$\sim 0.67\times$	-33%
JJ count	1064	865	-18.7%
Footprint (mm × mm)	1.98×0.87	1.95×0.72	-18.5%

- Baseline SCU: peak kernel throughput
- Stage-Reused SCU: $\approx 19\%$ lower JJ count and layout area, at $\approx 33\%$ lower throughput

Takeaways: An SFQ Co-Design for Bit-Vector Approximate Matching

What the co-design demonstrates

1 Algorithm–logic co-design

- Regular Bitap kernel for genome alignment
- 1-valid, shift-centric reformulation
- Explicit I / D / S / M pulse-driven SFQ dataflow

2 Layout-grounded performance

- 4-bit layouts anchor 64-bit multicore projection
- Baseline SCU: 14–16.6× higher kernel throughput vs. GenASM-DC
- Comparison uses the peak-throughput Baseline SCU

3 Design space and system

- Stage reuse: $\approx 19\%$ lower JJ count and footprint
- At $\approx 33\%$ lower throughput
- SCU is a kernel accelerator inside a heterogeneous mapper; seeding, traceback, and cryogenic I/O remain outside

- SCU co-designs the algorithmic representation, SFQ datapath, and physical implementation of a regular genome-alignment kernel.



Thank You !

SCU: An SFQ Co-Design for Bit-Vector Approximate Matching in Genome Alignment

Peiyao Qu; Email: qupeiyao@ict.ac.cn