Memory-Centric Accelerators for High-Frequency Space-Based Signal Processing: A High-Order Wavefront Sensing and Control Case Study

Barry Lyu, Nathaniel Bleier
Computer Science and Engineering
University of Michigan
Ann Arbor, MI, USA
{barrylyu, nbleier}@umich.edu

***Abstract*—A growing class of space-based signal processing applications—including radar imaging, hyperspectral analysis, and real-time optical control—demands computational capabilities far exceeding radiation-hardened processor performance, yet power constraints prohibit deploying datacenter-class hardware in orbit. We present two memory-centric accelerator architectures designed to bridge this gap. Using high-order wavefront sensing and control (HOWFSC) for the Habitable Worlds Observatory as a case study, we demonstrate that these workloads exhibit extremely low operational intensity, making memory bandwidth—not compute throughput—the limiting factor. We propose an HBM-based process-near-memory architecture achieving $100\,\mathrm{Hz}$ HOWFSC operation at $928\,\mathrm{W}$, and a distributed SRAM architecture reaching the same performance at just $90\,\mathrm{W}$—a $33\times$ reduction compared to GPU-based alternatives. Combined with reduced-precision storage of J and M, the SRAM design reaches $51\,\mathrm{W}$ ($59\times$ reduction), reducing co-flying satellite mass from $1110\,\mathrm{kg}$ to $193\,\mathrm{kg}$ and saving over $430M across a 25-year mission. These memory-centric design principles generalize beyond HOWFSC to the broader class of matrix-dominated space signal processing workloads.**

***Index Terms*—space computing, custom accelerators, memory-centric architecture, high-bandwidth memory, SRAM, wavefront control**

I. Introduction

A growing class of space-based signal processing applications demands computational capabilities that exceed radiation-hardened processor performance by orders of magnitude. These workloads—spanning synthetic aperture radar processing, hyperspectral image analysis, on-board machine learning inference, and real-time optical control—share a common computational signature: they are dominated by dense matrix operations with extremely low arithmetic intensity. The fundamental bottleneck is not floating-point throughput but rather the rate at which data can be streamed from memory to compute units.

This insight has profound implications for accelerator design in space. Commercial GPUs, optimized for compute-intensive AI workloads, provision enormous arithmetic capability that sits idle when executing memory-bound kernels. The resulting power consumption—often exceeding a kilowatt—is impractical for spacecraft thermal and power budgets. Radiation-hardened processors offer reliability but deliver performance several orders of magnitude below requirements. Ground-in-the-loop architectures remove compute constraints but introduce latencies of $10\,\mathrm{s}$ or more from deep space [1].

We propose a fourth path: custom accelerators designed from first principles around the memory access patterns of space signal processing. Rather than adapting general-purpose architectures, we architect systems where compute capability precisely matches memory bandwidth, eliminating wasted silicon area and power consumption.

To ground this architectural exploration, we adopt high-order wavefront sensing and control (HOWFSC) for the Habitable Worlds Observatory (HWO) as a detailed case study. HOWFSC represents an extreme point in the design space: maintaining coronagraph contrast at 10^{-10} requires control rates potentially exceeding $100\,\mathrm{Hz}$ [2], streaming 100+ GB matrices through compute units every iteration. As shown in Figure 1, operating at the optimal control cadence yields orders-of-magnitude better contrast than what ground-in-the-loop or radiation-hardened processors can achieve—motivating the custom accelerator architectures developed in this paper. The architectural solutions we develop for this demanding application generalize to the broader class of memory-bound space signal processing workloads.

This paper makes the following contributions:

- A roofline-based characterization demonstrating that representative space signal processing workloads exhibit operational intensities far below the compute-memory balance point of modern processors
- An HBM-based process-near-memory architecture that reduces power consumption by $3\times$ compared to GPU solutions while maintaining equivalent performance
- A distributed SRAM architecture achieving $33\times$ power reduction through elimination of off-chip memory access, extending to $59\times$ with reduced-precision storage
- A system-level cost analysis quantifying spacecraft mass and mission cost implications
- A comparative analysis of architectural tradeoffs including power scaling, radiation tolerance, and integration considerations

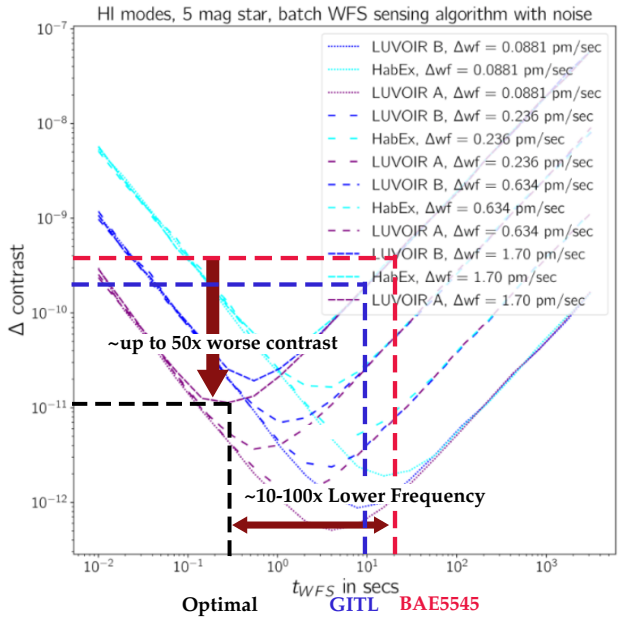


Fig. 1. Impact of wavefront sensing cadence on achievable contrast for LUVOIR-class telescopes, adapted from Pueyo et al. [2]. The optimal cadence (0.1–1 s) is orders of magnitude faster than ground-in-the-loop (GITL) or radiation-hardened processor (BAE5545 [3]) capabilities, motivating co-flying custom accelerators.

Related work. Pogorelyuk et al. [1] characterized the computational complexity of HOWFSC and identified the memory bandwidth bottleneck, concluding that radiation-hardened processors fall orders of magnitude short of requirements; Belsten et al. [4], [5] further evaluated HOWFSC on embedded and space-relevant hardware. Denby and Lucia [6], [7] introduced orbital edge computing, demonstrating that nanosatellite constellations can process remotely sensed data on-orbit rather than downlinking to ground stations; Bleier et al. [8], [9] proposed space microdatacenters—dedicated computational satellites for in-space processing—and developed system-level sizing models for co-flying compute platforms. Process-near-memory and process-in-memory architectures have been explored commercially through Samsung's HBM-PIM [10] and academically through various DRAM-logic integration proposals, but these target AI inference workloads with higher operational intensity than HOWFSC. Algorithm-based fault tolerance for matrix operations was introduced by Huang and Abraham [11]; a companion study [12] extends ABFT to the ill-conditioned matrices arising in HOWFSC and demonstrates that reduced-precision storage can significantly shrink the memory footprint without compromising control accuracy. Our work bridges these threads by co-designing the memory architecture, compute datapath, and fault tolerance strategy for the specific demands of space signal processing.

II. Co-Flying Compute Architecture

Before examining specific accelerator designs, we first address a fundamental architectural question: where should the computation physically reside? Three options exist for missions operating at the Sun-Earth L2 point: onboard the primary spacecraft, on Earth via ground-in-the-loop (GITL), or on a dedicated co-flying compute satellite.

A. The Case Against Onboard Processing

Radiation-hardened processors suitable for flagship observatory integration deliver performance three to five orders of magnitude below HOWFSC requirements. The BAE RAD5545, representative of current space-qualified hardware, achieves approximately $1.6\,\mathrm{GFLOP/s}$ [1], [3]. Even with aggressive parallelization across multiple processors, the power and thermal burden of deploying sufficient compute capability directly on the telescope would compromise instrument stability—the very problem HOWFSC aims to solve.

B. The Case Against Ground-in-the-Loop

GITL processing eliminates spacecraft compute constraints entirely but introduces insurmountable latency limitations. The round-trip light time from Earth to L2 exceeds $10\,\mathrm{s}$, establishing a hard floor on control loop period regardless of ground compute capability or link bandwidth. For HOWFSC targeting $100\,\mathrm{Hz}$ operation, GITL is fundamentally incompatible with mission requirements.

The Nancy Grace Roman Space Telescope will demonstrate GITL coronagraph operation, but at contrast levels (10^{-8}) and control rates (sub-Hz) far below HWO requirements [13]. Scaling GITL to HWO parameters is not feasible.

C. Co-Flying Compute Satellites

A dedicated compute satellite co-located at L2 resolves both constraints [8], [9]. The concept of offloading computation to co-located spacecraft extends the orbital edge computing paradigm [6], [7] from LEO Earth observation to deep-space science missions. Proximity to the telescope—separations of $100\,\mathrm{km}$ or less—reduces round-trip latency to under $1\,\mathrm{ms}$, enabling control frequencies limited only by compute throughput and inter-satellite link bandwidth. Meanwhile, physical separation from the primary observatory allows deployment of high-performance hardware without thermal or electromagnetic interference concerns.

Modern free-space optical links readily achieve $25\,\mathrm{Gbit/s}$ at these distances [14]. A brief link budget confirms this is sufficient: each HOWFSC iteration transfers a $206\,\mathrm{kB}$ detector image to the compute satellite and receives $206\,\mathrm{kB}$ of DM commands (25,736 actuators $\times$ 64 bits), totaling $412\,\mathrm{kB}$ per iteration or $330\,\mathrm{Mbit/s}$ at $100\,\mathrm{Hz}$—less than 2% of the $25\,\mathrm{Gbit/s}$ link capacity. Round-trip propagation latency at $100\,\mathrm{km}$ separation is $0.67\,\mathrm{ms}$, consuming only 7% of the $10\,\mathrm{ms}$ control period. The compute satellite can be designed around radiation-tolerant rather than radiation-hardened components, accepting higher upset rates in exchange for dramatically improved performance, with the understanding that the satellite is replaceable over the multi-decade mission lifetime.

This architectural approach motivates our focus on power-efficient accelerator designs: every watt saved in the compute subsystem translates directly to reduced satellite mass, simplified thermal management, and lower mission cost.

III. Workload Characterization

We characterize the computational requirements of memory-bound space signal processing using high-order wavefront sensing and control (HOWFSC) as a representative application. HOWFSC maintains coronagraph dark zones by estimating focal-plane electric fields and computing deformable mirror corrections through dense linear algebra operations [15]–[17].

A. Dominant Computational Kernels

The HOWFSC pipeline consists of three primary computational stages, each dominated by dense linear algebra operations that reduce to general matrix-vector multiplication (GEMV) as the fundamental kernel.

Electric Field Estimation converts focal-plane detector measurements into estimates of the complex electric field. Two approaches dominate: pairwise probing (PWP) and extended Kalman filtering (EKF). PWP applies known deformable mirror probe patterns and solves a small regularized least-squares problem independently for each pixel:

$$\hat{E} = (\mathbf{H}^T\mathbf{H} + \mu\mathbf{I})^{-1}\mathbf{H}^T\boldsymbol{\delta} \tag{1}$$

where $\boldsymbol{\delta}$ contains difference measurements between positive and negative probes. EKF maintains running state estimates updated each iteration via Kalman gain computation, requiring access to the full Jacobian matrix $\mathbf{J}$ to propagate control perturbations.

Control Computation determines corrective deformable mirror commands given the estimated electric field. Electric field conjugation (EFC) with precomputed matrix $\mathbf{M}$ computes:

$$\Delta\mathbf{u} = -\mathbf{M}\mathbf{J}^T\mathbf{E} \tag{2}$$

where $\mathbf{M} = (\mathbf{J}^T\mathbf{J} + \alpha\mathbf{I})^{-1}$ can be precomputed since the Jacobian and regularization parameter evolve slowly compared to the control rate.

Jacobian Application propagates the previous iteration's control command through the optical model to predict the induced electric field change, enabling the EKF state update. This requires computing $\mathbf{J}\Delta\mathbf{u}$, another GEMV operation.

Each HOWFSC iteration thus follows a three-phase control loop: (1) capture a focal-plane image and estimate the residual electric field (PWP or EKF), (2) compute corrective DM commands via EFC ($\Delta\mathbf{u} = -\mathbf{M}\mathbf{J}^T\mathbf{E}$), and (3) apply the Jacobian to predict the field change for the next iteration's state update ($\mathbf{J}\Delta\mathbf{u}$). Phases (2) and (3) are each a single GEMV against a precomputed matrix and dominate runtime by 2–3 orders of magnitude over all other operations [1]. The Jacobian $\mathbf{J}$ and matrix $\mathbf{M}$ evolve slowly relative to the control rate and can be updated on timescales of minutes to hours using the general-purpose processor, while the per-iteration GEMVs must execute at cadences of tens to hundreds of hertz.

For a LUVOIR-class coronagraph with 25,736 active actuators, 51,316 active pixels, and 5 spectral channels, the Jacobian matrix requires 106 GB of storage in double precision [1]. The precomputed matrix $\mathbf{M}$ adds another 5.3 GB. Each control iteration must stream these matrices through the compute units, fundamentally limiting achievable frequency to memory bandwidth rather than arithmetic throughput.

TABLE I
Operational Intensity of HOWFSC Kernels in Double Precision

Kernel	Op. Intensity	Regime
Pairwise Probing	0.33 FLOP/byte	Memory-bound
EKF State Update	0.25 FLOP/byte	Memory-bound
EFC Control	0.25 FLOP/byte	Memory-bound

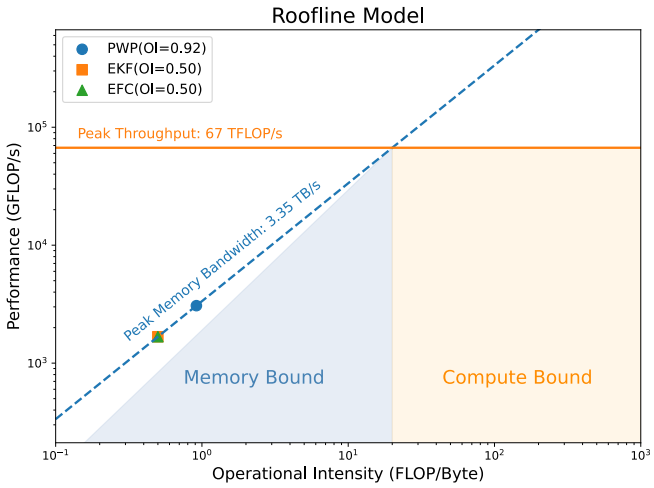


Fig. 2. Roofline analysis showing HOWFSC kernels on an NVIDIA H100 GPU using FP32 (H100 HBM capacity is insufficient for double precision Jacobian). Despite 67 TFLOPS of peak throughput, effective performance is limited by memory bandwidth to below 5 TFLOPS for these low-intensity workloads.

B. Operational Intensity Analysis

Operational intensity—the ratio of floating-point operations to bytes transferred from memory—determines whether a workload is compute-bound or memory-bound on a given architecture. For GEMV with an $M \times N$ matrix, the operation count is $2MN$ floating-point operations while the memory traffic is $8(MN + M + N)$ bytes in double precision, yielding an operational intensity of approximately 0.25 FLOP/byte.

Table I summarizes the operational intensity of key HOWFSC kernels. All values fall well below the compute-memory balance point of modern processors, which typically ranges from 10–100 FLOP/byte for high-performance GPUs and CPUs.

C. Roofline Analysis

The roofline model provides a visual framework for understanding the performance ceiling of memory-bound workloads [18]. Figure 2 maps the HOWFSC kernels onto the roofline for several hardware platforms. All kernels fall deep within the memory-bound region, indicating that achievable performance scales linearly with memory bandwidth regardless of peak compute capability.

TABLE II
MEMORY TECHNOLOGY COMPARISON

Technology	Bandwidth	Density	Energy
DDR5 DRAM	Moderate	High	Moderate
HBM3e	Very High	High	Higher
SRAM	Highest	Low	Lowest

This analysis reveals a fundamental mismatch between modern accelerator architectures and space signal processing requirements. GPUs provision enormous compute capability that remains idle during GEMV execution, while the memory subsystem—designed for workloads with significant data reuse—becomes the sole performance limiter.

IV. DESIGN SPACE EXPLORATION

Given that memory bandwidth dominates performance, accelerator design for space signal processing becomes primarily a memory system architecture problem. We explore the tradeoffs across memory technologies to identify promising design points.

A. Memory Technology Landscape

Table II summarizes the key characteristics of candidate memory technologies. Each presents distinct tradeoffs between bandwidth, capacity, energy efficiency, and radiation tolerance.

High Bandwidth Memory (HBM) achieves exceptional bandwidth through 3D stacking and wide interfaces, with current HBM3e devices delivering over $1\,\mathrm{TB/s}$ per stack [19]. The primary limitations are elevated power consumption ($20\,\mathrm{W}$ per stack) and sensitivity to radiation-induced errors in the stacked DRAM dies.

Static RAM (SRAM) offers the lowest access energy and highest bandwidth per bit but at significantly reduced density compared to DRAM [20]. Modern SRAM at $2\,\mathrm{nm}$ achieves approximately $38\,\mathrm{Mbit/mm^2}$, requiring substantial die area to store large matrices. However, SRAM can be fabricated with radiation-hardened cell designs and integrated directly with compute logic.

Emerging memories including ReRAM and MRAM offer intrinsic radiation tolerance through their non-volatile storage mechanisms, but current density and bandwidth limitations preclude their use as primary storage for large-scale signal processing [21], [22].

B. Architectural Design Principles

Our analysis motivates several key design principles for memory-centric space accelerators:

Match compute to bandwidth. Rather than provisioning excess compute capability, size the arithmetic units to exactly consume the available memory bandwidth at the target operational intensity. This eliminates wasted area and power from idle compute resources.

Minimize data movement. Every byte transferred consumes energy proportional to the distance traveled. Process-near-memory and process-in-memory architectures reduce this overhead by placing compute adjacent to storage.

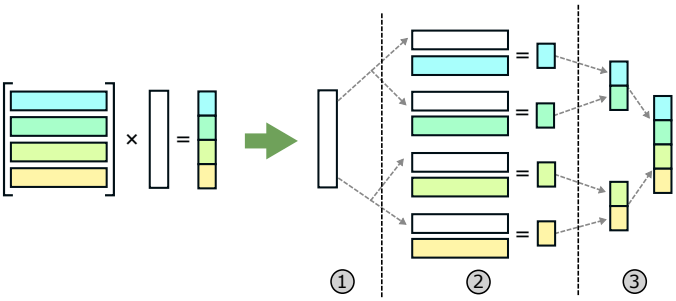


Fig. 3. Parallel GEMV decomposition. The matrix is partitioned by rows across processing elements. (1) Input vector broadcast. (2) Independent local GEMV. (3) Partial result gather.

Exploit parallelism through distribution. GEMV admits efficient parallel decomposition by partitioning the matrix across multiple memory-compute units, with only the input vector broadcast and output vector reduction requiring inter-unit communication.

Design for the memory hierarchy. The 110+ GB matrix storage requirement exceeds on-chip SRAM capacity for any practical single die. Architectures must either accept off-chip memory access (HBM) or distribute storage across multiple chiplets (SRAM).

C. Parallel GEMV Execution

GEMV operations admit efficient parallel decomposition that scales to arbitrary processor counts with minimal communication overhead. The key insight is that matrix-vector multiplication can be partitioned by rows: each processing element stores a horizontal slice of the matrix and computes the corresponding elements of the output vector independently.

For N processing elements executing GEMV with matrix $\mathbf{A}$ and input vector $\mathbf{x}$ (illustrated in Figure 3):

1) **Broadcast**: The input vector $\mathbf{x}$ is distributed to all processors
2) **Compute**: Each processor i computes $\mathbf{y}_i = \mathbf{A}_i\mathbf{x}$ for its local partition $\mathbf{A}_i$
3) **Gather**: Partial results are collected to form the complete output vector

The communication cost is dominated by the broadcast phase, which transfers $\mathcal{O}(M)$ elements where M is the vector dimension. The gather phase transfers only $\mathcal{O}(M/N)$ elements per processor. Since the matrix dimensions vastly exceed the vector dimensions in HOWFSC (millions of matrix elements vs. tens of thousands of vector elements), communication represents a negligible fraction of total execution time.

A hierarchical tree (H-tree) network topology enables efficient broadcast and gather operations. For a degree-d tree connecting N processors, broadcast latency scales as $\mathcal{O}(\log_d N)$ while per-link bandwidth requirements remain modest. Analysis shows that for 128 processors operating at $100\,\mathrm{Hz}$, point-to-point bandwidth requirements are approximately $31\,\mathrm{GB/s}$—well within the capability of modern chip-to-chip interconnects [23].

This favorable scaling behavior motivates distributed architectures where memory capacity and bandwidth are partitioned across multiple chiplets or memory stacks, with the interconnect fabric sized to support the relatively lightweight vector traffic.

V. HBM-Based Architecture

Our first accelerator design leverages HBM to provide the memory bandwidth necessary for 100 Hz HOWFSC operation. The architecture prioritizes minimizing engineering complexity by building on commercially available memory technology.

A. System Organization

The HBM architecture consists of multiple HBM3e stacks connected to a custom compute die containing double-precision multiply-accumulate (MAC) units. The compute die is sized to exactly balance the aggregate memory bandwidth of the attached HBM stacks, avoiding the over-provisioning characteristic of GPUs.

At 100 Hz operation with the LUVOIR-A parameter set, the minimum required memory bandwidth is:

$$BW = 100 \times 111 \text{ GB} = 11.1 \,\text{TB/s} \tag{3}$$

With HBM3e stacks providing 1.125 TB/s each [19], approximately 10 stacks would theoretically suffice. However, physical interface constraints, communication and processing latencies, and the need to partition data across stacks motivate a design with 27 HBM stacks, providing comfortable bandwidth margin and enabling efficient parallel access.

B. Compute Organization

The compute die implements approximately 7,400 double-precision MAC units operating at 500 MHz. These units are organized into clusters, with each cluster serving a subset of HBM stacks. The precomputed matrices are striped across all HBM stacks, and each compute cluster performs its portion of the GEMV independently.

The input vector is broadcast to all clusters through a hierarchical distribution network. Partial results are gathered and reduced to produce the final output vector. This communication overhead is minimal compared to the GEMV computation itself, as the vectors are orders of magnitude smaller than the matrices.

C. Power Analysis

Table III breaks down the power consumption of the HBM architecture. Memory access dominates, consuming over 70% of total power. The compute units themselves contribute modestly due to the small number required to match the memory-limited throughput.

A radiation-tolerant general-purpose processor (Teledyne LX2160A Space) [24] handles non-GEMV operations including EKF state updates, dither generation, and system control. These operations are far less demanding and execute efficiently on conventional processors.

TABLE III
HBM Architecture Power Breakdown

Component	Power
HBM Stacks (27×)	804 W
HBM PHY Interfaces	58 W
Compute (7,400 MACs)	11 W
Interconnect	8 W
Control + Misc	25 W
LX2160A Processor	25 W
Total	928 W

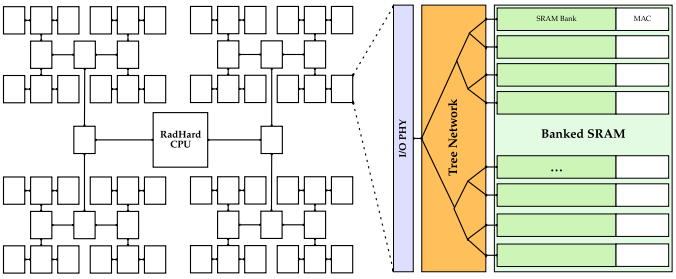


Fig. 4. Distributed SRAM architecture with 56 chiplets arranged in a hierarchical tree interconnect. Each chiplet contains SRAM storage and co-located MAC units.

D. Comparison to GPU Baseline

The HBM architecture achieves equivalent 100 Hz performance to a three-GPU cluster (NVIDIA B200 [25]) while reducing power consumption from 3000 W to 928 W—a 3.2× improvement. This reduction stems entirely from eliminating unused compute capability; the memory subsystem power is comparable between approaches.

VI. SRAM-Based Architecture

To further reduce power consumption, we explore a distributed SRAM architecture that eliminates off-chip memory access entirely. While requiring significantly more silicon area, this approach offers substantial energy efficiency advantages and improved radiation tolerance.

A. Distributed Memory Organization

The SRAM architecture distributes the full 111 GB of matrix storage across 56 custom chiplets, each containing approximately 2 GB of SRAM and co-located compute units. At 38 Mbit/mm^2 density, each chiplet requires roughly 425 mm^2 of SRAM area—within the manufacturing limits of current process technology.

The chiplets are connected through a hierarchical tree network that supports efficient broadcast of input vectors and gathering of partial results. Figure 4 illustrates the system organization.

B. Compute Organization

Each chiplet contains 135 double-precision MAC units, for a system total of 7,560 MACs. The per-chiplet compute capability is sized to exactly consume the SRAM read bandwidth, maintaining the memory-centric design philosophy.

TABLE IV
SRAM ARCHITECTURE POWER BREAKDOWN

Component	FP64	BFP-23
SRAM Dynamic	38 W	14 W
SRAM Leakage	11 W	4 W
Compute	11 W	3 W
Interconnect	5 W	5 W
LX2160A Processor	25 W	25 W
Total	90 W	51 W

Within each chiplet, SRAM banks are organized to enable parallel read access, with the MAC units arranged in a dataflow pattern that minimizes internal data movement. The matrix partition stored on each chiplet remains stationary; only the input and output vectors traverse the interconnect.

C. Power Analysis

Table IV presents the power breakdown for the SRAM architecture. The dramatic reduction compared to HBM stems from two factors: SRAM read energy is approximately $25\times$ lower than HBM per bit, and the elimination of off-chip interfaces removes a significant power overhead.

SRAM leakage contributes a non-negligible baseline power that is independent of operating frequency. At very low control rates, this leakage dominates and the SRAM architecture becomes less efficient than HBM. However, for the 100 Hz target, dynamic power dominates and SRAM offers clear advantages.

D. Area and Cost Considerations

The total SRAM area across all chiplets exceeds $23\,000\,\mathrm{mm}^2$—roughly equivalent to 20 large processor dies. While this represents significant silicon cost, the expense is modest compared to overall spacecraft development and operation costs for flagship missions.

The distributed chiplet approach also offers advantages for radiation tolerance. Individual chiplets can be fabricated with radiation-hardened SRAM cells, and the system can potentially continue operation with reduced capability if individual chiplets fail.

VII. COMPARATIVE EVALUATION

We evaluate the proposed architectures against a GPU baseline across multiple dimensions relevant to space deployment.

A. Power-Frequency Scaling

Figure 5 shows how each architecture scales across the range of control frequencies relevant for HOWFSC. The GPU baseline exhibits poor efficiency at low frequencies due to high idle power, and requires multiple units to reach 100 Hz. Both custom architectures scale more gracefully, with the SRAM design maintaining low power across the full frequency range.

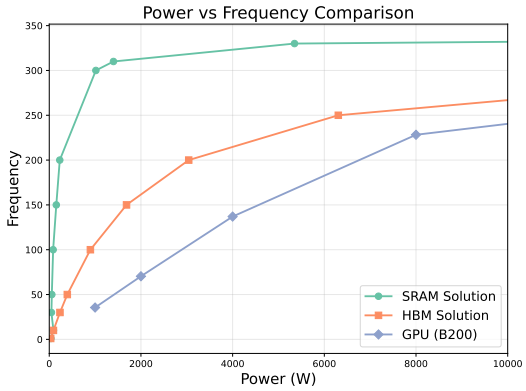


Fig. 5. Power consumption versus compute-limited control frequency for the three architectures. Inter-satellite transfer latency is excluded; at $25\,\mathrm{Gbit/s}$ and separations under $300\,\mathrm{km}$, link overhead is negligible relative to compute time. Custom designs achieve equivalent performance at dramatically lower power.

TABLE V
ARCHITECTURE COMPARISON AT 100 Hz

Metric	GPU	HBM	SRAM	SRAM-BFP
Power	3000 W	928 W	90 W	51 W
Silicon Area	Low	Moderate	High	Moderate
Dev. Complexity	Low	Moderate	High	High
Rad. Tolerance	Low	Low	Moderate	Moderate

B. Architectural Tradeoffs

Table V summarizes the key tradeoffs between architectures. The optimal choice depends on mission-specific constraints including power budget, radiation environment, and development timeline.

GPU baseline offers the fastest path to deployment through commercial availability, but power consumption makes spacecraft integration extremely challenging. A 3 kW thermal load requires substantial radiator area and constrains orbital configurations.

HBM architecture provides a middle ground with manageable power consumption and moderate development complexity. The primary concern is HBM's sensitivity to radiation; error-correcting codes and scrubbing can mitigate soft errors, but the stacked die structure limits shielding options.

SRAM architecture offers the lowest power consumption and the best path to radiation hardening, but requires the largest silicon investment and most complex system integration. For flagship missions with long development timelines, this approach may be most appropriate.

VIII. SYSTEM-LEVEL COST ANALYSIS

The power reduction achieved by custom architectures has cascading effects throughout the satellite system, ultimately determining mission feasibility and total cost.

A. Satellite Sizing Model

Spacecraft mass and cost scale super-linearly with payload power requirements. Higher compute power demands larger

TABLE VI
SATELLITE COST SCALING WITH COMPUTE POWER

Architecture	SRAM-BFP	SRAM	HBM	GPU
Compute Power	51 W	90 W	928 W	3000 W
BOL Power	198 W	237 W	1400 W	4418 W
Spacecraft Mass	193 kg	231 kg	476 kg	1110 kg
Unit Cost	$32M	$38M	$65M	$111M

solar arrays, bigger batteries for eclipse operation, more massive thermal radiators, and structural reinforcement to support these subsystems. Using parametric cost models validated against historical small satellite missions [9], we estimate the relationship between compute power provisioning and total satellite cost.

Table VI presents satellite parameters for the four compute architectures assuming a 3-year operational lifetime and $25\,\mathrm{Gbit/s}$ inter-satellite link capability.

Beginning-of-life (BOL) power exceeds compute power due to housekeeping loads, communication systems, and efficiency losses in power conversion. The non-linear relationship between power and mass reflects the compounding nature of spacecraft subsystem sizing.

B. Mission Lifetime Cost

Over HWO's projected 25-year operational lifetime, multiple compute satellite replacements will be required due to radiation degradation and component aging. Assuming 3-year satellite lifetimes, approximately 9 units must be manufactured and launched over the mission duration.

Total mission cost including non-recurring engineering (NRE) and recurring unit costs:

- **SRAM-BFP**: $170M total ($17M NRE + 9×$17M units)
- **SRAM**: $203M total ($17M NRE + 9×$21M units)
- **HBM**: $357M total ($29M NRE + 9×$36M units)
- **GPU baseline**: $602M total ($49M NRE + 9×$61M units)

The SRAM architecture saves approximately $400M over the mission lifetime compared to GPU-based alternatives, with reduced-precision storage extending savings to over $430M—funds that could support additional science instrumentation or extended operations.

C. Custom Silicon Development Cost

Custom accelerator development requires non-recurring investment in chip design and fabrication. Unlike complex general-purpose processors where software development dominates NRE, our proposed architectures consist primarily of regular memory arrays and simple floating-point datapaths, dramatically reducing verification complexity.

Photomask fabrication at leading-edge nodes represents the dominant NRE component, with estimates of $30M for a complete 5nm-class mask set. However, this one-time cost is amortized across all satellite units and remains small compared to the cumulative savings in spacecraft mass and launch costs. The recurring per-chip manufacturing cost is negligible relative to satellite-level expenses.

IX. DISCUSSION

A. Radiation Environment and Tolerance

The Sun-Earth L2 point presents a moderate radiation environment compared to other orbital regimes. Spacecraft operate outside Earth's trapped radiation belts, with total ionizing dose (TID) dominated by galactic cosmic rays and solar particle events [26]. Studies for JWST estimated approximately $2.4\,\mathrm{krad}$ per year behind standard aluminum shielding [27], [28]—manageable for multi-year satellite lifetimes with appropriate component selection.

Modern FinFET and gate-all-around transistors exhibit improved single-event effect (SEE) tolerance compared to planar CMOS, with upset rates reduced by approximately an order of magnitude at each technology generation [29]. The reduced sensitive volumes in advanced nodes, combined with silicon-on-insulator substrates where available, provide inherent hardening without specialized fabrication processes.

B. Layered Fault Tolerance Strategy

We adopt a defense-in-depth approach combining multiple protection mechanisms. ECC protects memory at rest, ABFT protects computation in flight, and TMR guards control logic.

Error-Correcting Codes (ECC) protect all memory structures. SRAM banks implement single-error-correction, double-error-detection (SECDED) codes with minimal area overhead. For HBM, chipkill-level protection guards against complete die failures within stacks. Since HOWFSC accesses all matrix elements every iteration, normal memory traffic triggers error correction without dedicated scrubbing cycles.

Triple Modular Redundancy (TMR) protects critical control paths including state machines, configuration registers, and the inter-satellite communication interface. These structures are small relative to the datapath and memory, making TMR overhead acceptable.

Graceful Degradation allows continued operation at reduced performance if radiation events cause persistent failures. The distributed SRAM architecture can bypass failed chiplets, accepting proportionally reduced control frequency. The system maintains dark zone stability at degraded rates while ground operators assess recovery options.

C. Algorithm-Based Fault Tolerance

ECC protects $\mathbf{J}$ and $\mathbf{M}$ in storage, but transient faults during the GEMV computation itself—SEUs in MAC registers, adder trees, or data buses—are not covered by ECC. Algorithm-based fault tolerance (ABFT) [11] addresses this gap by embedding checksum relationships directly into the matrix computation, with overhead proportional to matrix *dimensions* rather than total elements.

1) ABFT Mechanism: For each protected matrix A, ABFT encodes K checksum rows $c_k[j] = \sum_i w_k[i] \cdot A[i,j]$ at load time. After each GEMV $\mathbf{y} = A\mathbf{x}$, the syndrome

$$d_k = \mathbf{w}_k^T \mathbf{y} - \mathbf{c}_k^T \mathbf{x} \tag{4}$$

is zero in exact arithmetic; $|d_k| > \tau_k$ indicates an error. The weights $\mathbf{w}_k$ and checksums $\mathbf{c}_k$ are computed once when the

matrices are loaded; per-iteration cost is $2K$ dot products plus n multiplies for vector scaling, totaling less than 0.2% of the GEMV at LUVOIR scale with K=1 [12].

2) The Ill-Conditioning Challenge: The composed EFC gain matrix $\mathbf{G} = \mathbf{M}\mathbf{J}^T$ is severely ill-conditioned: SVD analysis of a FALCO-generated vortex-coronagraph model reveals singular values spanning approximately 66 decades, with row norms varying by a factor of 10^{32} [12]. This extreme dynamic range causes the ABFT syndrome d_k to be nonzero even on clean data due to floating-point roundoff, potentially masking actual faults. A companion study [12] addresses this with *row-scaling preconditioning*: the scaled gain matrix $\tilde{G} = D\mathbf{G} = (D\mathbf{M})\mathbf{J}^T$ (where $D = \mathrm{diag}(1/\|\mathbf{G}_{i,:}\|)$) normalizes all active rows to unit norm, preventing faults in small-norm rows from being masked by the checksum accumulation noise of large-norm rows. Scaling is applied once at encode time; the per-iteration cost is n multiplies on the output vector.

The detection threshold τ_k is derived analytically from the probabilistic rounding model of Higham and Mary [30], bounding the syndrome noise as:

$$|d_k| \leq \varepsilon_{\mathrm{m}} \left(\sqrt{m}\, \|\mathbf{w}_k\| \|\tilde{G}\|_F \frac{\|\mathbf{x}\|}{\sqrt{n}} + \sqrt{n}\, \|\mathbf{c}_k\| \|\mathbf{x}\| \right) \quad (5)$$

where $\varepsilon_{\mathrm{m}} = 2.22 \times 10^{-16}$ is the FP64 machine epsilon. All quantities are known at encode time, so no empirical tuning or calibration runs are required.

3) Dangerous Fault Threshold: Not all faults matter: a bit-flip in a low-significance mantissa bit may change $\Delta\mathbf{u}$ imperceptibly. A fault is *dangerous* only if the resulting DM command error degrades contrast beyond a tolerable budget. The Jacobian $\mathbf{J}$ maps command errors $\mathbf{e}$ to focal-plane field perturbations, bounding the contrast degradation as $\delta C \leq \|\mathbf{J}\|_2^2 \|\mathbf{e}\|^2$. Allocating a fraction $f = 0.1$ of the target contrast $C_{\text{target}} = 10^{-10}$ as the fault budget yields the dangerous-fault threshold:

$$\varepsilon_{\mathrm{d}} = \frac{\sqrt{f \cdot C_{\text{target}}}}{\|\mathbf{J}\|_2 \cdot \|\Delta\mathbf{u}\|} \quad (6)$$

which evaluates to approximately 1.5%–9% depending on actuator count and input E-field [12]. This threshold is large because the Jacobian spectral norm is small: the coronagraph's optical response to individual actuator perturbations is weak, so a substantial command error is needed before the focal-plane field change degrades contrast.

4) Tolerance Gap: The three quantities—roundoff noise floor, detection threshold, and dangerous-fault threshold—occupy well-separated regimes:

$$\underbrace{|d_k|_{\text{noise}}}_{\sim 10^{-13}} \ll \underbrace{\tau_k}_{\sim 10^{-8}} \ll \underbrace{\varepsilon_{\mathrm{d}}}_{\sim 10^{-2}} \quad (7)$$

providing approximately six orders of magnitude of margin between the detection threshold and the dangerous-fault threshold [12]. This gap is the reason ABFT works despite the extreme ill-conditioning: even though the matrix is numerically challenging, the physics of the coronagraph ensures that only large command errors degrade contrast, and large errors produce large syndromes. Experimental validation across 10,000 fault injection trials with five physically motivated fault types (single bit-flip, multi-bit flip, bitline inversion, full randomization, and burst error) achieves 100% detection of all dangerous faults with zero false positives [12].

ABFT also supports matrix sharding naturally: each partition carries local checksums, and per-partition conditioning ($\kappa_{\text{tile}} < \kappa_{\text{global}}$) improves noise-floor separation—complementing the distributed memory organizations described in Sections V and VI.

D. Reliability Analysis

We quantify the reliability benefit of ABFT combined with reduced-precision storage using the synthesis-derived MAC areas from Section IX-F. The dangerous computational fault rate depends on three factors: the SEU flux (upsets/mm^2/hour, set by the radiation environment), the active datapath area exposed per iteration, and the fraction of faults that are dangerous.

For FP64, the MAC cell area is 3755 (ASAP7 [31] area units); for BFP-23, it is 1423—a $2.6\times$ reduction in silicon area exposed to particle strikes per MAC. With 7,560 MACs operating at $100\,\mathrm{Hz}$, the BFP-23 datapath intercepts proportionally fewer ions per iteration simply by being physically smaller.

Without ABFT, every dangerous computational fault passes through to the DM command. With ABFT achieving 100% dangerous-fault detection, the only residual risk is a fault in the checker itself—the K syndrome dot products, which represent approximately 0.2% of the GEMV computation. This reduces the dangerous computational fault rate by approximately $500\times$ [12]. Combined with the $2.6\times$ area reduction from BFP-23 MACs, the total dangerous-fault rate improvement is approximately $1300\times$ compared to an unprotected FP64 datapath. At the L2 SEU flux of approximately 10^{-6} upsets/mm^2/hour [32], this translates to a dangerous computational fault rate well below one event per year of continuous $100\,\mathrm{Hz}$ operation. Figure 6 shows how the dangerous-fault rate (in FIT) scales with SEU flux for all four configurations.

This layered approach—ECC for storage, ABFT for computation, TMR for control, and graceful degradation for persistent failures—enables deployment of commercial-grade silicon in the L2 environment with reliability appropriate for supporting, but not directly endangering, the primary observatory mission.

E. Technology Roadmap

Both proposed architectures are designed around memory technologies available today or on near-term product roadmaps. Looking toward the HWO launch timeframe in the 2040s, continued improvements in memory density and energy efficiency will further enhance the viability of memory-centric accelerators.

Advances in chiplet integration, packaging, and high-density interconnects will reduce the complexity and cost of distributed multi-chiplet systems. Improved SRAM density at

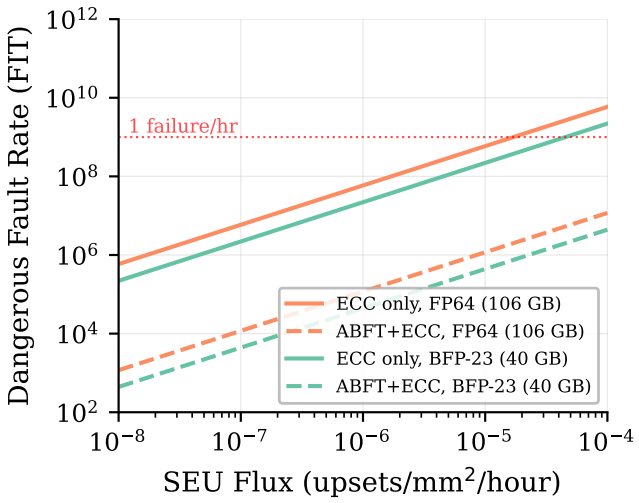


Fig. 6. Dangerous computational fault rate (FIT) vs. SEU flux. Solid lines: ECC only (computational faults unprotected). Dashed lines: ABFT+ECC (residual risk from checker faults only). BFP-23 MACs have 2.6× smaller area than FP64, reducing exposure to particle strikes.

future process nodes will also reduce the number of chiplets required, though even with reduced-precision storage shrinking the matrices to approximately 42 GB, distributed architectures will remain necessary for the foreseeable future.

We also note that, since the core workload is highly parallel, the proposed architectures do not require high inter-processor communication bandwidth. A multi-chip module (MCM), or multiple modules connected through board-level interconnects such as PCIe, can satisfy the communication requirements, although at the cost of increased interconnect energy.

F. MAC Synthesis Methodology

To quantify the power and area impact of reduced-precision datapaths, we synthesized Synopsys DesignWare `DW_fp_mac` (floating-point multiply-accumulate, $a \times b + c$) instances at two precisions using Synopsys Design Compiler (S-2021.06-SP1) with `compile_ultra` optimization. The target library is ASAP7 [31], a predictive 7nm FinFET PDK with regular-V_t cells at typical-typical corner (0.7V, 25°C). Both designs are purely combinational (no pipeline registers), providing a direct comparison of datapath complexity.

The FP64 MAC uses `sig_width=52`, `exp_width=11` (IEEE 754 double precision, 64-bit inputs). The BFP-23 MAC uses `sig_width=23`, `exp_width=11` (35-bit inputs), modeling the reduced-precision format where $\mathbf{J}$ and $\mathbf{M}$ are stored with 23-bit mantissas but the full FP64 exponent range is retained. Table VII summarizes the results.

The 3.6× power reduction exceeds the 2.6× area reduction because smaller multipliers have shorter critical paths and lower internal switching activity per operation. These per-MAC ratios are used in Table IV to estimate the BFP-23 compute power and in Section IX-D to estimate the reduced radiation cross-section of the narrower datapath.

TABLE VII
MAC SYNTHESIS RESULTS (ASAP7 7NM, COMPILE_ULTRA)

Metric	FP64	BFP-23	Ratio
Cell Area (units)	3755	1423	2.6×
Dynamic Power (uW)	2285	640	3.6×
Cell Count	18468	6707	2.8×

TABLE VIII
LUVOIR-SCALE JACOBIAN MEMORY FOOTPRINT

Format	Mantissa	Size (GB)	Savings
FP64 (baseline)	52	106	—
BFP, per-row exp.	36	61	42%
BFP, per-row exp.	30	51	52%
BFP, per-row exp.	23	40	62%

G. Reduced-Precision Storage Opportunity

The architecture sizing in Sections V and VI assumes the $\mathbf{J}$ and $\mathbf{M}$ matrices are stored in IEEE 754 double precision (FP64). However, constructing $\mathbf{M} = (\mathbf{J}^T\mathbf{J} + \alpha\mathbf{I})^{-1}$ and applying $\mathbf{J}$ and $\mathbf{M}$ in the per-iteration GEMVs impose fundamentally different precision requirements. The inversion involves ill-conditioned linear algebra that genuinely requires FP64, but the subsequent GEMVs—sequences of dot products—are far more tolerant of reduced precision because truncation errors are independent and average out as $\sqrt{n}$ rather than n [12].

A companion study [12] demonstrates that the gain matrix can be stored in block floating point (BFP) with a 23-bit mantissa. Motivated by this result, we use the same row-wise format for $\mathbf{J}$ and $\mathbf{M}$, where each matrix row shares a single exponent and elements store only a sign bit and mantissa, reducing their combined FP64 footprint from 111.3 GB to approximately 42 GB, a 62% reduction. Figure 7 shows the gain-matrix GEMV relative error as a function of mantissa width: even at 10 bits (FP16-class), the error remains 19× below the dangerous-fault threshold, with FP32-class precision offering over four orders of magnitude of margin. The GEMV error at 23-bit BFP consumes less than 0.03% of the contrast error budget, and ABFT detection capability is fully preserved. Row-scaling preconditioning maps directly to the BFP per-row exponent, making both techniques effectively free in hardware.

Table VIII shows the LUVOIR-scale Jacobian footprint across precision levels. BFP rows store one shared FP64 scale factor plus a sign bit and p-bit mantissa per element, yielding significant savings even at moderate truncation.

This reduction would substantially shrink both proposed architectures. The HBM design, currently sized at 27 stacks to store and stream 111.3 GB at 100 Hz, could operate with roughly 11 stacks—approaching the theoretical minimum—with proportional reductions in memory power, currently the dominant contributor at 804 W. The SRAM design, which distributes storage across 56 chiplets at approximately 2 GB each, could achieve the same capacity with approximately 22 chiplets. We estimate the BFP-23 SRAM power budget by scaling SRAM dynamic and leakage power linearly with

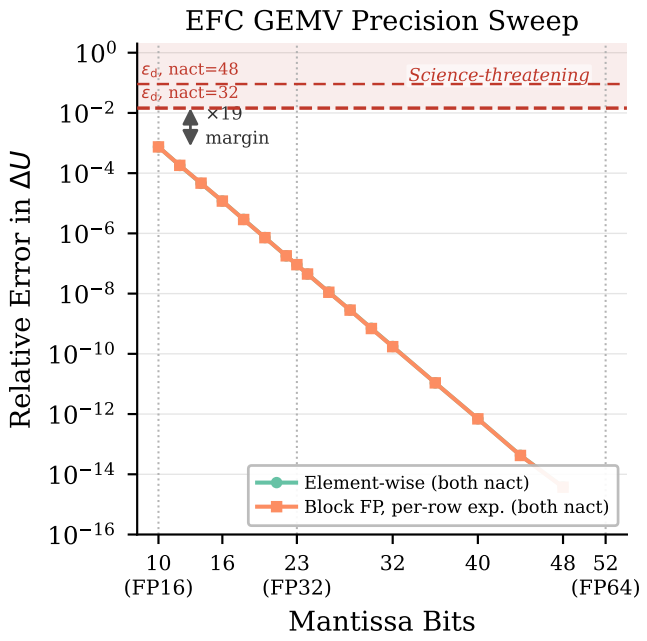


Fig. 7. EFC gain-matrix GEMV relative error vs. mantissa bit width. The shaded region above ε_d indicates science-threatening errors. Even FP16-class precision (10 bits) maintains 19× margin below the dangerous threshold. Element-wise and block floating-point formats perform identically [12]. N_{act}: actuators per DM side.

chiplet count (42/111.3 of baseline) and scaling compute power using synthesis results: a DesignWare floating-point MAC synthesized at the BFP-23 datapath width consumes 3.6× less power than an FP64 MAC (ASAP7 7nm, compile_ultra), yielding 3 W for compute versus the baseline 11 W. The resulting system estimate of 51 W (Table IV, BFP-23 column) represents a 59× reduction compared to the GPU baseline. These savings cascade through the satellite system model: lower power translates to smaller solar arrays, lighter thermal radiators, and reduced spacecraft mass and cost.

We retain FP64 as the baseline throughout the baseline architecture sizing to provide conservative, technology-independent architecture bounds. Detailed reworking of the HBM and SRAM designs under BFP storage is left to future work; the analysis in [12] confirms that such reworking would yield significant additional savings beyond those reported here.

H. Broader Space Computing Applications

While we have focused on HOWFSC as a driving application, the memory-centric design principles apply broadly to space signal processing workloads sharing low operational intensity and large working sets.

Synthetic Aperture Radar (SAR) focusing requires large FFTs and matched filtering over multi-gigabyte datasets. Current SAR satellites downlink raw data for ground processing; on-board processing would enable real-time targeting and reduced downlink bandwidth.

Hyperspectral Image Analysis involves covariance matrix computation and eigendecomposition across hundreds of spectral bands per pixel—operations dominated by memory bandwidth at megapixel scale. On-board analysis would enable intelligent data triage, transmitting only scientifically relevant observations.

On-Board ML Inference for autonomous spacecraft operation requires streaming weight matrices through compute units—precisely the memory-bound pattern our architectures optimize.

Space Domain Awareness applications including debris tracking and conjunction assessment require batch least-squares orbit determination over large observation catalogs.

In each case, the computational kernel reduces to dense matrix operations where memory bandwidth—not arithmetic throughput—limits performance, and radiation-hardened processors fall orders of magnitude short of requirements.

X. Conclusion

Memory bandwidth, not compute throughput, limits the performance of signal processing workloads relevant to future space missions. By designing accelerators around this insight, we can achieve dramatic improvements in power efficiency compared to general-purpose approaches.

Our HBM-based architecture reduces power consumption by 3× relative to GPU solutions while our distributed SRAM architecture achieves a 33× reduction at FP64 precision. Combined with reduced-precision storage [12]—block floating point with 23-bit mantissas, reducing **J** and **M** from 111.3 GB to approximately 42 GB—the SRAM architecture reaches 51 W, a 59× reduction that enables deployment of high-frequency control loops within practical spacecraft power budgets. Algorithm-based fault tolerance provides 100% detection of science-threatening computational faults with less than 0.2% overhead, and the reduced-precision datapath further improves reliability by shrinking the silicon area exposed to particle strikes. These architectures establish a viable path toward co-flying compute platforms for flagship missions requiring computational capabilities beyond what radiation-hardened processors can deliver. Future work will refine these architectures through detailed physical design and validate radiation tolerance through simulation and testing.

Acknowledgment

The authors thank Jamila Taaki, Leonid Pogorelyuk, Laurent Pueyo, and Kerri Cahoy for discussions on wavefront sensing and control algorithms.

References

[1] L. Pogorelyuk, C. Haughwout, N. Belsten, E. Cady, and K. Cahoy, "Computational complexities of image plane algorithms for high contrast imaging in space telescopes," *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 8, no. 4, p. 049003, 2022. [Online]. Available: https://doi.org/10.1117/1.JATIS.8.4.049003

[2] L. A. Pueyo, L. Pogorelyuk, I. Laginja, R. Soummer, A. Sahoo, E. Por, K. Cahoy, L. Coyle, and S. Knight, "Architecture trades to optimize wavefront stability requirements for exoplanet imaging in space." in *Space Telescopes and Instrumentation 2022: Optical, Infrared, and Millimeter Wave*, L. E. Coyle, S. Matsuura, and M. D. Perrin, Eds., vol. 12180, International Society for Optics and Photonics. SPIE, 2022, p. 121802J. [Online]. Available: https://doi.org/10.1117/12.2630757

[3] BAE Systems, *RAD5545 SpaceVPX SBC v1.2 Flight – Data Sheet*, BAE Systems, Jan. 2025, spaceVPX single-board computer for space applications. [Online]. Available: https://www.baesystems.com/en-us/dam/jcr:cd804739-19fa-43f6-b47a-c58eb18e3bba/17-c09-RAD5545-SpaceVPX-SBC-v1-2-Flight-ds-2025-web.pdf

[4] N. Belsten, K. Milani, L. Pogorelyuk, B. Eickert, S. Rao, E. S. Douglas, and K. Cahoy, "Evaluating embedded hardware for high-order wavefront sensing and control," in *Techniques and Instrumentation for Detection of Exoplanets XI*, G. J. Ruane, Ed., vol. 12680, International Society for Optics and Photonics. SPIE, 2023, p. 126801N. [Online]. Available: https://doi.org/10.1117/12.2677816

[5] N. Belsten, "Embedded computing for wavefront control on future space telescopes," Ph.D. dissertation, Massachusetts Institute of Technology, 2025.

[6] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proceedings of the 25th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2020, pp. 939–954.

[7] ——, "Orbital edge computing: Machine inference in space," *IEEE Computer Architecture Letters*, vol. 18, no. 1, pp. 59–62, 2019.

[8] N. Bleier, M. H. Mubarik, G. R. Swenson, and R. Kumar, "Space microdatacenters," in *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2023.

[9] N. Bleier, R. Eason, M. Lembeck, and R. Kumar, "Architecting space microdatacenters: A system-level approach," *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2025. [Online]. Available: https://crucible.eecs.umich.edu/papers.html?id=3

[10] S. J. Lee, K.-Y. Kim, S. Oh, J. Park, G. Hong, D. Y. Ka, K.-D. Hwang, J. Park, K. P. Kang, J. Kim *et al.*, "A 1ynm 1.25v 8gb, 16gb/s/pin GDDR6-based accelerator-in-memory supporting 1tflops MAC operation and various activation functions for deep-learning applications," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, 2022, pp. 1–3.

[11] K. H. Huang and J. A. Abraham, "Algorithm-based fault tolerance for matrix operations," *IEEE Transactions on Computers*, vol. C-33, no. 6, pp. 518–528, 1984.

[12] B. Lyu and N. Bleier, "Algorithm-based fault tolerance for high-order wavefront sensing and control," in *Proceedings of the 63rd ACM/IEEE Design Automation Conference (DAC)*, 2026, invited paper.

[13] B. Mennesson, V. P. Bailey, R. Zellem, S. Hildebrandt, M. Ygouf, J. Rhodes, N. Zimmerman, B. Nemati, G. Gonzalez, E. Cady, B. Kern, T. Koch, J. Krist, K. Heydorff, T. Luchik, F. Mok, P. Morrissey, I. Poberezhskiy, A. J. Riggs, F. Shi, F. Zhao, R. Akeson, L. Armus, A. Greenbaum, J. Ingalls, and P. Lowrance, "The Roman Space Telescope coronagraph technology demonstration: current status and relevance to future missions," in *Space Telescopes and Instrumentation 2022: Optical, Infrared, and Millimeter Wave*, L. E. Coyle, S. Matsuura, and M. D. Perrin, Eds., vol. 12180, International Society for Optics and Photonics. SPIE, 2022, p. 121801W. [Online]. Available: https://doi.org/10.1117/12.2629176

[14] C. M. Schieler, K. M. Riesing, A. J. Horvath, B. C. Bilyeu, J. S. Chang, A. S. Garg, J. P. Wang, and B. S. Robinson, "200 Gbps TBIRD CubeSat downlink: pre-flight test results," in *Free-Space Laser Communications XXXIV*, H. Hemmati and B. S. Robinson, Eds., vol. 11993, International Society for Optics and Photonics. SPIE, 2022, p. 119930P. [Online]. Available: https://doi.org/10.1117/12.2615321

[15] L. Pogorelyuk and N. J. Kasdin, "Dark hole maintenance and a posteriori intensity estimation in the presence of speckle drift in a high-contrast space coronagraph," *The Astrophysical Journal*, vol. 873, no. 1, p. 95, Mar. 2019. [Online]. Available: http://dx.doi.org/10.3847/1538-4357/ab0461

[16] S. F. Redmond, L. Pogorelyuk, L. Pueyo, E. Por, J. Noss, S. D. Will, I. Laginja, K. Brooks, M. Maclay, J. Fowler, N. J. Kasdin, M. D. Perrin, and R. Soummer, "Implementation of a dark zone maintenance algorithm for speckle drift correction in a high contrast space coronagraph," *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 8, no. 3, p. 035001, 2022. [Online]. Available: https://doi.org/10.1117/1.JATIS.8.3.035001

[17] L. Pogorelyuk, L. Pueyo, and N. J. Kasdin, "On the effects of pointing jitter, actuator drift, telescope rolls, and broadband detectors in dark hole maintenance and electric field order reduction," *Journal of Astronomical Telescopes, Instruments, and Systems*, vol. 6, no. 3, p. 039001, 2020. [Online]. Available: https://doi.org/10.1117/1.JATIS.6.3.039001

[18] S. Williams, A. Waterman, and D. Patterson, "Roofline: an insightful visual performance model for multicore architectures," *Communications of the ACM*, vol. 52, no. 4, pp. 65–76, 2009. [Online]. Available: https://doi.org/10.1145/1498765.1498785

[19] K. Chae, J. Song, Y. Choi, J. Park, B. Koo, J. Oh, S. Yi, W. Lee, D. Kim, K. Kang, E. Kim, J. Kim, S. Park, S. Park, M. Noh, H. G. Rhew, and J. Shin, "A 4-nm 1.15 tb/s hbm3 interface with resistor-tuned offset calibration and in situ margin detection," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 1, pp. 231–242, 2024.

[20] T.-Y. J. Chang, Y.-H. Chen, K. V. Reddy, N. Puri, T. Masina, K.-C. Lin, P.-S. Wang, Y. Lin, C.-Y. Lin, Y.-H. Nien, H. Fujiwara, K.-F. Lin, M.-H. Chang, C. W. Wu, R. Lee, Y. Wang, H.-J. Liao, Q. Li, P. W. Wang, and G. Yeap, "A 38.1mb/mm2 sram in a 2nm-cmos-nanosheet technology for high-density and energy-efficient compute," in *2025 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 68, 2025, pp. 492–494.

[21] Z. Jiang, S. Qin, H. Li, S. Fujii, D. Lee, S. Wong, and H.-S. P. Wong, "Next-generation ultrahigh-density 3-d vertical resistive switching memory (vrsm)—part ii: Design guidelines for device, array, and architecture," *IEEE Transactions on Electron Devices*, vol. 66, no. 12, pp. 5147–5154, 2019.

[22] T.-C. Chang, Y.-C. Chiu, C.-Y. Lee, J.-M. Hung, K.-T. Chang, C.-X. Xue, S.-Y. Wu, H.-Y. Kao, P. Chen, H.-Y. Huang, S.-H. Teng, and M.-F. Chang, "13.4 a 22nm 1mb 1024b-read and near-memory-computing dual-mode stt-mram macro with 42.6gb/s read bandwidth for security-aware mobile devices," in *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2020, pp. 224–226.

[23] D. De Sensi, L. Pichetti, F. Vella, T. De Matteis, Z. Ren, L. Fusco, M. Turisini, D. Cesarini, K. Lust, A. Trivedi, D. Roweth, F. Spiga, S. Di Girolamo, and T. Hoefler, "Exploring gpu-to-gpu communication: Insights into supercomputer interconnects," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*, ser. SC '24. IEEE Press, 2024. [Online]. Available: https://doi.org/10.1109/SC41406.2024.00039

[24] Teledyne e2v Semiconductors SAS, *LX2160-Space: Radiation-Tolerant 16× ARM Cortex-A72 2.2 GHz Microprocessor – Datasheet*, Teledyne e2v, Mar. 2023, preliminary specification. [Online]. Available: https://flir.netx.net/file/asset/71493/original/attachment

[25] NVIDIA Corporation, *NVIDIA Blackwell Architecture Data Sheet*, NVIDIA, 2025. [Online]. Available: https://nvdam.widen.net/s/wwnsxrhm2w/blackwell-datasheet-3384703

[26] R. A. Mewaldt, A. J. Davis, K. A. Lave, R. A. Leske, E. C. Stone, M. E. Wiedenbeck, W. R. Binns, E. R. Christian, A. C. Cummings, G. A. de Nolfo, M. H. Israel, A. W. Labrador, and T. T. von Rosenvinge, "Record-setting cosmic-ray intensities in 2009 and 2010," *The Astrophysical Journal Letters*, vol. 723, no. 1, p. L1, 2010.

[27] J. L. Barth, J. C. Isaacs, and C. Poivey, "The radiation environment for the next generation space telescope," in *Proceedings of the RADECS Conference*, 1999.

[28] T. J. Greenholt, D. F. Figer, B. J. Rauscher, M. Regan, S. H. Moseley, and J. Garnett, "Radiation environment performance of JWST prototype FPAs," in *Space Telescopes and Instrumentation 2008: Optical, Infrared, and Millimeter*, vol. 7010, 2008, p. 70103U.

[29] N. Seifert, B. Gill, S. Jahinuzzaman, J. Basile, V. Ambrose, Q. Shi, R. Allmon, and A. Bramnik, "Soft error susceptibilities of 22 nm tri-gate devices," *IEEE Transactions on Nuclear Science*, vol. 59, no. 6, pp. 2666–2673, 2012.

[30] N. J. Higham and T. Mary, "A new approach to probabilistic rounding error analysis," *SIAM Journal on Scientific Computing*, vol. 41, no. 5, pp. A2815–A2835, 2019.

[31] L. T. Clark, V. Vashishtha, L. Shifren, A. Gujja, S. Sinha, B. Cline, C. Ramamurthy, and G. Yeric, "Asap7: A 7-nm finfet predictive process design kit," *Microelectronics Journal*, vol. 53, pp. 105–115, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002626921630026X

[32] E. Normand, "Single event upset at ground level," *IEEE Transactions on Nuclear Science*, vol. 43, no. 6, pp. 2742–2750, 1996.