# Microchip and UHCL

# Methodology Lab Feasibility Study

## Purpose

Determine Feasibility of a Methodology Lab in a University setting. Output of this study will be a validation report on measurement techniques, and an implementation plan for the permanent lab.

## Background

In data storage changing from spinning media (HDDs) to flash (SSDs) requires changes in measurement methodology. Advanced design processes also require that new ASIC architectures are fully validated in virtualized environments. Virtualized environments suffer from stretches time domains, where legacy storage measurements often use single data points that take seconds or minutes. A new statistical based methodology is needed that derives necessary data from many small samples.

Microchip wishes to commission a study and validate our approach to both measurement and improving the system design process.

## Staged Plan

This is a staged study designed to evolve towards an independent **Fairchild Systems Architecture Lab** as described in this proposal. Microchip proposes an initial staged feasibility and demonstration project as a starting point for this effort. We would like to consider a two-year plan.

Each stage in the plan is a semester.

### Lab Operational Costs

Microchip chip would like to have flexibility from semester to semester with regards to the size of the team. We would like agreement to specify lab operating costs on semester basis, based on a rate per student worker per semester.[1] We expect the initial two-year study, at maximum, to involve no more than ten students in any given semester.[2]

### Lab Overhead Costs

Lab fixed overhead (rent, power, IT costs) should be broken separately and specified either on yearly or semester basis.

### Lab Built Out

One time cost for lab build out should be specified separately and cover the two-year feasibility phase.

### Stages

#### *Stage 1: Feasibility*

Described in detail in the next section. Validate basic tool set and ready lab for future stages.

[1] Or may specify in units of two students. We want fully engaged students with expectation of working no more than 20 hours per week.

[2] Ten or the maximum under supervision under single professor, whichever is less.

### *Stage 2: Correlation*

Students will correlate the most basic storage application, file copy, to synthetic benchmarks, using statistical analysis techniques.

### *Stage 3: Quantification and Optimization*

Students will quantify measurements into single metric set, and then determine optimal values for application performance.

### *Stage 4: Modeling*

Students will model their optimizations using a programable RAM drive and measure actual performance against their predictive optimizations.

## Partner Sponsors

The intent is to evolve to an independent, self-sustaining Methodology Lab. With this goal in mind, it benefits Microchip and UHCL to bring in other industry partners as soon as possible. We would like the proposal to be amendable, and partners to join at any of the stages. Partners contribute not only funding, but expertise, as well as software and tools, and will work to correlate the lab's findings in their own labs.

The University should maintain a clear accounting of all material commitment by sponsors. This is so that responsibility for the governance of an independent lab can be divided proportional to the commitment of the sponsored lab.

## Detailed Breakdown, Stage 1: Feasibility

### Measurement Methodology

Statistical methodology requires small samples of short duration. These studies are initial steps in validation of the methodology. Two aspects of measurement will be considered, and these will be correlated to and compared against traditional measurement techniques.

This study should be performed by students with experience with a strong background in statistics and data visualization. Rudimentary C and Linux skills will be helpful.

#### *Isolation*

Data points are taken back-to-back, with a small delay for software cleanup and initialization. Compare this with deterministic "settle" times between data points.

#### *Duration*

Traditional benchmarks specify duration in terms of seconds. New methodology requires much shorter duration data points, specified in the number of operations. Compare short tests of various counts and determine if a deterministic value can be extrapolated from short tests. Evaluate the value of an incremental count and measure drift by count.

### Test Methodology

This study will plan and implement basic lab infrastructure. The long-term goal is a REDS digital design lab, but we need to start with a framework for the students to recurse and optimize upon. The study will

plan lab infrastructure and come up with a starting point for students. Must Evaluate resources needed to open-source toolset and define limited support model.

*Engineers are responsible for the tools they use.* The lab is an engineer's most basic tool. Establishing a modern process for creating a lab for defining a methodology might be just as important and insightful as establishing the methodology.

### Team (3)

**Professor**. Should have background in statistical analysis of data, visualization (charting) and previous hands-on lab experience. They will supervise students, set schedules and interface with sponsor.

**Student, Measurement**. Have basic C programming experience, make minor modifications to existing toolset. May need to transform data sets for analysis. Maintains process documentation.

**Student, Test.** Basic Linux system administration, familiar with lab environment. Will manage data sets and create skeleton process for test execution.

### Equipment

The feasibility study will focus on the toolset and process, so testing can be done on legacy system, thus reducing cost. Microchip would like to provide (gift) raw servers and SSDs.

## Stage One Estimated Budget – Lab Operation

| Item | Unit |
|---|---|
| Weeks (per semester) | 18 |
| Hours | 20 |
| People | 2 |
| Wage Per Hour (mid-point) | $25.00 |
| Base Wages | $18,000.00 |
| Professor Supervision | $20,000.00 |
| Raw Total | $38,000.00 |
| University Markup | 30.00% |
| Estimated Subtotal | $49,400.00 |

# Fairchild Systems Architecture Lab

V0.15

"*Engineers are responsible for the tools they use."- Steve Fairchild*

## Purpose

A laboratory to define and validate open measurement methodologies and tools for digital systems.

### Background

Successful architecture of a digital system requires a proven measurement methodology. In complex open systems, this can be difficult, as no sub-system is completely independent.

Subsystem architects often use synthetic benchmarks to isolate and optimize their components. Without correlation to complete system applications, it can only be a design measurement, not a measurement of architectural performance.

Scale-out architectures dominate hyperscale and enterprise compute. This trend will continue and accelerate as silicon designs reach the geometry limit. In scale-out, the most important factors aren't raw performance of a subsystem, but the efficient operation of a node under a controlled load. There's an adage in the PC industry, "*You can never add too much memory*." That's not true in scale-out – Too much of anything is as bad as too little. Overprovisioning a node is a waste of capital, power, and time.

Digital architecture is very simple. Broken down into components, architects use data-flow analysis to size and connect the components. It's very much like designing a system to efficiently load railcars or a container ship. It can be modeled and visualized in this manner.

Fairchild Lab's primary mission is to develop a recursive methodology and toolset, one which self-adjusts and demands constant integration as parts of the system change.

## Architectural Breakdown

Since the lab is funded and supported by the Storage Industry, we'll begin a system breakdown on high-performance SSD storage.

### Recursive Engineering Design System (REDS)

The industry trend in chip design over the past fifty years has led to large design teams and longer design cycles. The trend for the future is smaller design teams and shorter design cycles. To get there, engineering teams must fully leverage Moore's law and constantly optimize architectures, designs and most importantly, the design process.

Key to rapid integration are tools and practices that allow teams to effectively debug and communicate complex designs. Unfortunately, larger design teams have led to more specialization. This has led to

specialization of tools and processes, creating separate engineering abstraction domains. Teams using different abstraction domains must constantly reframe their understanding of problems into their own abstraction. This is inefficient and error prone.

Fairchild Labs will develop a fully recursive data and test management system, with cohesive and modern data visualization. It should also include integrated scheduling and management of lab resources and staff. It will use a common abstraction domain to relay all data and visual interpretation.

Lab sponsors encourage student workers to develop their own team structure and work processes, including terminology and leadership. We believe superior results will come from small teams of five or less working together on a common task set and objectives.

## Characterization

The storage subsystem, or "stack" consists of many software and hard components. An architectural breakdown uses SW and HW tools to take measurements of the same load at various measurement points in the stack. By "breaking down" the stack, we can measure the efficiency of the components using traditional benchmarks, isolating the bottlenecks and resource constraints.

The lab will characterize production SSDs, controllers, and storage fabrics and produce a publicly available data set on baseline reference hardware.

The lab may also perform private characterization of prototypes under NDA for sponsor members.

## Correlation

The lab and sponsor companies will work with software vendors, hyperscale data-centers, as well as end user customers in business and academia to define application reference baselines. By using the same hardware and software instrumentation we use for characterization, we can also characterize application loads.

## Modeling

Using characterization and correlation data, we can build accurate models of subsystems, as we know the inputs and outputs rates and latency of each component. First, we'll build models of existing components on a simulation and visualization platform. We'll validate the models by running the same characterization and correlation benchmarks. At that point, we'll have the tools to model new devices based on creative input from the students, as well as directed models from the sponsors.

## Emulation

Models the prove worthy can be instantiated as software or hardware emulation and validated. The lab will maintain funds for student projects.

## Publication

The lab will publish characterization reports openly after a review period, where sponsors may review and rebut methodology or correct deficiencies in their product. All characterization reports will fully document methodology and open source all tools required. The lab will also encourage students to integrate findings and innovations into their course work, practicing recursive integration on the curriculum.

### Summer/Semester Co-Op

The lab will work with sponsors to develop semester and summer work programs for lab students. These programs should directly leverage student lab experience and should be goal oriented with measurable results.

## Governance

The Lab shall be formed as an independent non-profit entity and managed by a board.

- Founding Sponsor Company – one permanent seat (propose 5 total)
- University Trustee – one permanent seat
- Alumina Trustee – one permanent seat
- Student Trustee – active participant selected/elected seat per semester
- Rotating Trustee(s) – one seat for every three non-founding sponsors

All Sponsors are committing to the educational and vocational improvement of the student-engineers as the primary mission of the board.

### Budget

<Brandon: This is for open discussion; I just want to write something down to begin the conversation with both university and partners. I need to have more discussion with university and consult with existing labs >

The lab should be fully funded for a minimum sustainable period of five years. This is to ensure we can measure the effectiveness of working student-engineers over a typical college enrollment period. We propose that a five-year budget be committed by the Founding sponsor and split evenly, to fairly distribute commitment to the responsibility of the board seats. 50% of the budget for each sponsor is reserved for direct student tasks and managed as a credit for priority of future task assignments.

| Semester Cost | | Unit |
|---|---|---|
| Weeks (per semester) | | 17.2 |
| Hours | | 20 |
| People | | 5 |
| Wage Per Hour | | $25.00 |
| Admin overhead % | | 5.00% |
| Professor supervision % | | 20.00% |
| Base Wages | | $43,000.00 |
| Wage Overhead | | $53,750.00 |
| Baseline Equipment | | $50,000.00 |
| DUTs (SSD/IOC/HDD - sponsor product) | IN KIND | |
| Total Per Semester | | $103,750.00 |

| 5 Year Breakdown | Unit |
|---|---|
| Founding Sponsors | 5 |
| Cost of 1 Team per semester | $103,750.00 |

| | |
|---|---|
| Number of Teams | 2 |
| Years | 5 |
| Total Semesters | 20 |
| 5 Year Minimum Budget | $2,075,000.00 |
| Founding Sponsor 5 Year Commitment | $415,000.00 |

## Non-Founding Members

The board should debate new sponsorships. We should add sponsors only with multi-year commitments and in line with the long-term capabilities and goals of the Lab itself.

## Report Publication Disputes

The board should be hands off and the lab self-directed with guidance to accomplish their macro-tasks. We expect student-engineers to prove their findings with data. We will set a timeline for publication, where sponsors and the board have a right to suggest better techniques or methodology. The board should specify these rules in writing.

We expect that all sponsor companies should be able and willing to correlate the lab's finding using the lab's tools and methodology. If disputing finding, sponsors should provide counter-data and help with debug. In all cases, the direction should be a formal adjudication process, with publication held as long as necessary to resolve the issue of objection.

## Confidentiality

Sponsors wishing for characterization of prototypes may wish to keep their activities and results confidential from public and other sponsors. The board shall devise a process of supervising and administrating these activities.

## Compensation

The lab shall have competitive pay, based on skill, experience and leadership. Tasks are given a skill rating by sponsors. Student engineers may advance through the ranking and pay levels as fast as they are able to demonstrate competency. Specific tasks may have course prerequisites.

| Skill Level | Competencies |
|---|---|
| 1 | Run tests, manage data sets, prepare visualization of data for review |
| 2 | Test infrastructure, debug of results, lead reviews, task development |
| 3 | C development of tools, models and infrastructure. Team leadership. NDA Tasks. |
| 4 | Advanced modeling, emulation. Sponsor Engineer I expectations [Engineer I] |
| 5 | Solicitation of work from sponsors, engaging with non-sponsor institutions. [Engineer II] |

## Plan of Action

- Spring 2023
  - Identify key professors.
  - Identify underclassmen with skillset and leadership potential.
  - Get 3 to 5 students working immediately (participants for fall 2023)

        - Work directly for Microchip until Lab is legally formed; or
        - Work for school, funded by sponsors
    - Prepare base toolset for open-source.
    - Identify student leadership for Fall/Summer
- Summer 2023
    - Develop budget for continuous operation for 5 years
        - Base budget for 10 students
        - Founding sponsor commit for five years
        - Goal is 5+ founding sponsors
    - Form lab as non-profit governed by trustees appointed by sponsor companies.
    - Staff/outsource administrative functions
    - Build out initial "lab"
        - 1 rack and 1 bench is all space needed initially
    - Microchip will host student leadership and train directly.
- Fall 2023
    - Begin recursive characterizations.
    - Aim for 10 students
        - Attract at least 5 freshmen to gauge 4 year effectiveness
    - Publish first Report
    - Open source base toolset

# Implementation

<Brandon: The rest of this technical and is WIP>

## Astros

Astros is a storage performance tool developed by Microchip. It was designed for statistical analysis and small burst sampling. Small burst sampling is important to chip designers, as it allows us to use the same techniques in simulation, where the time and compute resources required is expensive.

We would like to donate the source code to Fairchild Labs and for the staff of the lab commit to open sourcing all tools and providing limited support (release and bug tracking). The Lab is expected to source or design the tools necessary for the job. The lab's motto shall be *Engineers are responsible for the tools they use*. If the right tool doesn't exist, the lab will accept the challenge and built it.

## Fairchild Metric

For decades, raw storage performance has been expressed as a rate while driving a load. The metrics are expressed as Operations per second (IOP/s) or as bandwidth, a multiplier of bytes per second. The loads would typically be described as queue depth, operation (read or write or combination), a block size, and a type of access (random or sequential).

Typically, vendors pick a small block size for random, and a large block size for sequential and express the max rates. This is simplistic and not useful for modeling. Why? Because real applications don't behave like that at all.

We propose the lab define a new metric, the Fairchild.

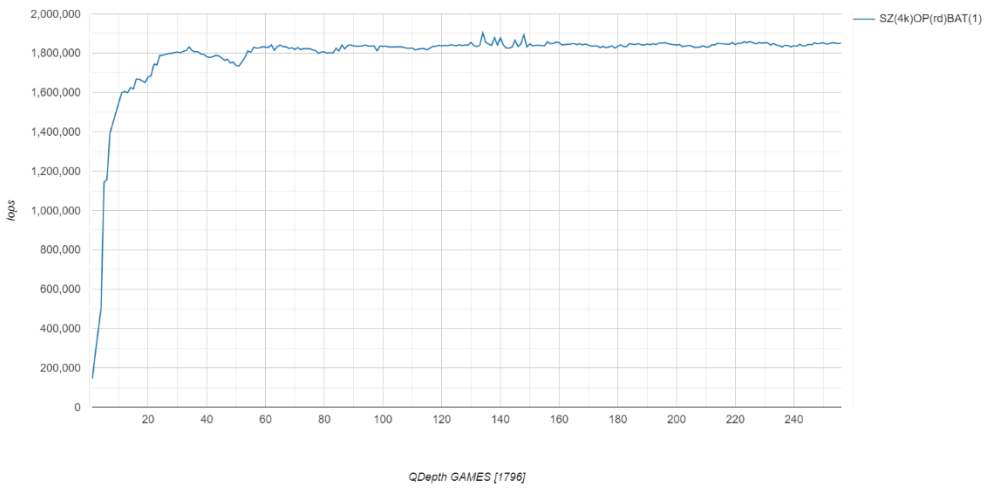


*Figure 1- single IOP/s curve*

This is a sweep of a queue depth load at a fixed block size. As queue depth grows, so does the raw IOP/s. A simple way to quantify this mathematically would be the area under the queue depth curve. This gives us a single factor metric that is more descriptive than a maximum.

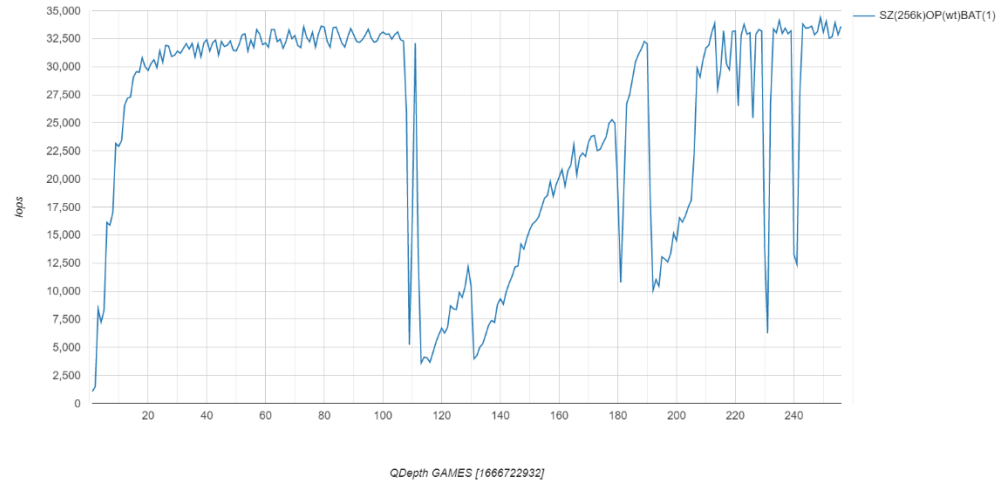


*Figure 2 - incoherent curve*

Consider curves like this. Scale-out solutions demand components that scale with a reasonable expectation of positive linearity.  Non-linearity may be derived from a smaller than expected area, but not always.

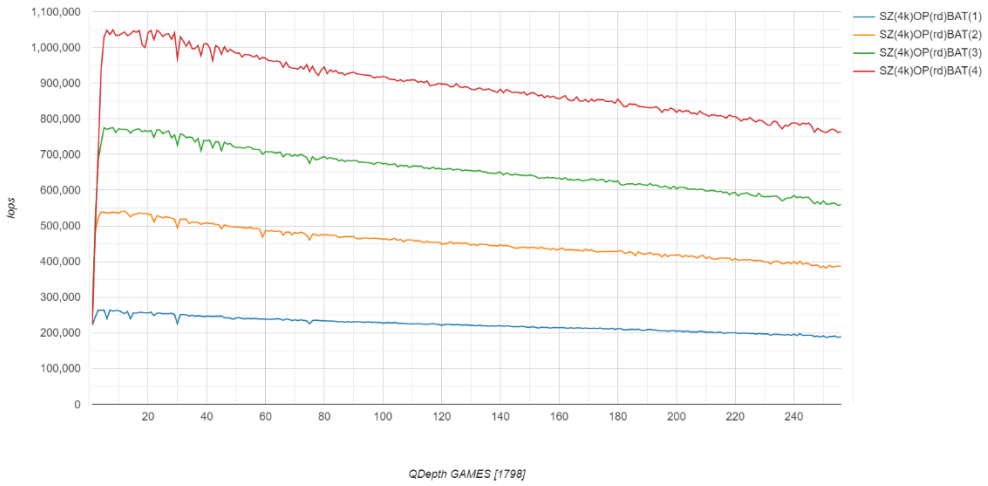


*Figure 3 - CPU Scaling*

Each series in this chart represents another dimension of the load, the number of “batters” or CPU threads that are used.  It’s a good measure of the amount of Host CPU resources to “drive” a load.  By splitting the load across more CPUs, we can scale performance to higher levels under otherwise identical loads.

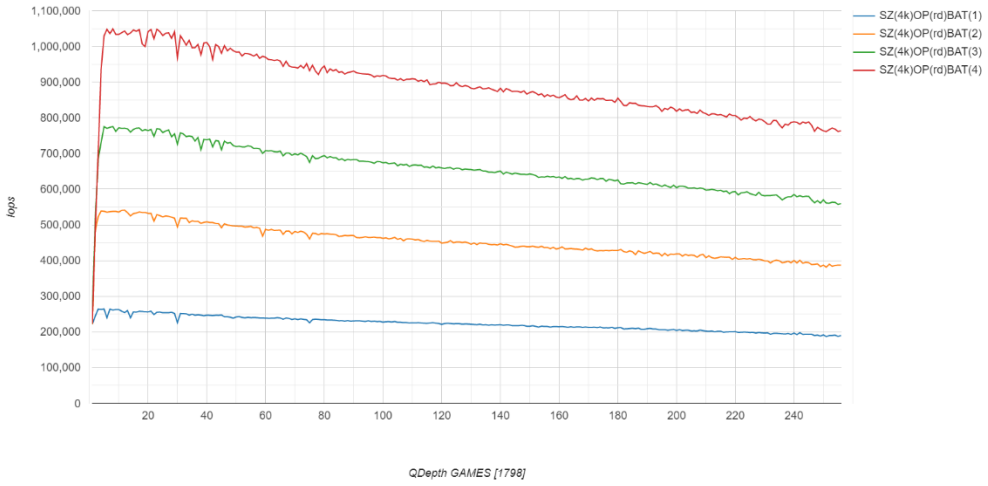


*Figure 4 – TODO: Brandon figure out how to do a 3-D volume Graph*

If you were to graph this in 3-D, instead of area, you could take a measurement of volume. Instead of graphing each batter as a separate series, you'd make batters the z-axis of your 3-D drawing. The volume of this would be a single factor metric of the gross performance.

Pilots use performance curves to calculate the fuel load on every single flight. They typically use software, but all pilots learn how to calculate this manually from printed performance curves. It's the same problem we face with scale-out. When planes carry too much fuel, the entire flight is inefficient. When planes carry too little fuel, the side-effects are much worse.

Pilots must consider many factors, such as wind direction, temperature, humidity, traffic when calculating performance. They use these factors as biases in their calculation, either manually, or with software. These calculations are very reliable. It's unusual to find a pilot hitchhiking to the next gas station.

We can apply this kind of analysis to digital systems. As we add dimensions of our load, we can extend the metric beyond the 3 dimensions, and still calculate a single factor. Area is the measure of a 2-D curve, volume is the measure under a 3-D curve. Hypervolume is the measure of a topography that has more than three dimensions. Visualizing hypervolume is difficult, as are most concepts in greater than three dimensions. Describing the inputs to the calculation in computing terms is simple – it's just a multi-dimensional array.

For calculating gross load, you first need to properly bias the dimensions of the load itself. Queue depth is an integer measurement, while block size can be expressed as bytes or kilobytes. Likewise, CPU count is an integer, while access type [random, sequential] is just a simple integer enumeration.

$$Fairchild_{grossload} = Fairchild_{gl} = Hypervolume\ of\ the\ tested\ load$$

$Fairchild_{gl}$ may have use cases. It is a full spectrum, allowing designers to measure gross scale of change when they are optimizing their designs. As a useful metric, it has problems. For instance, a high IOP/s rate at 4K reads may indicate strong transactional processing prowess, it's meaningless is the application has optimized on 32K and the bandwidth of the interconnect is the limiting factor. That excess of power is wasted potential in both the capital and power domains.

While not directly useful as a metric, the dataset is useful.

$$Faichild_{applicationbias} = Fairchild_{ab} = Fairchild_{gl}\ x\ Bias\ Matrix$$

Application bias is the measured relative bias of each data point in $Fairchild_{gl}$. This can be calucated by sampling profiles of applications loads, and correlating to data points measured in isolation. Where no correlation can be achieved, new values or dimensions of sampled data points can be added. The lab will recurse on a given application until correlation can be achieved with sampled benchmark data.

The next step is modeling a given $Fairchild_{ab}$ in software, emulation or hardware. With the model you perform the same profile. Once you have correlation with a model with both the application and sampling benchmarks, you can move to optimization for the targeted applications.

Once you have a spread of applications profiled, you can optimize an architecture or multiple architectures to a subset of the problems.

## Recursive Lab Management

The Lab shall design a fully integrated system for testing, data management, visualization, data-mining and debug. Every published report shall be replicated on a second set of hardware, by a second student-engineer. The Lab's core testing systems will be "locked-down" and all Hardware or Software changes logged and revalidated against existing baseline.

This might be the biggest challenge, but one where student-engineer's might be able to provide industry with the greatest leverage. Many sponsors have decades of legacy process and practice and often use in-house designed tools that are less than optimal. An engineer's most powerful tool is Moore's law – we need to make sure we always use it to leverage our designs. Just as important is to leverage the design process.

## Sustainable Path

The lab can become self-sufficient in terms of funding by contracting specific tasks. Some macro tasks the lab would be well suited for.

- Characterizing a sponsor's production products and calculating the Fairchild metrics for an array of applications.
- Profiling the $Fairchild_{ab}$ for a new application
- Optimizing a model for an application given a specific set of constraints
- NDA engineering studies of protypes and beta-level product