FLOW COMPUTING FAQ

1. What problem is Flow trying to solve?

We believe there has been only incremental improvements in CPU performance during recent decades. In our opinion, it has led to a situation where the CPU has actually become the weakest link in computing due to its suboptimal sequential architecture. A new era in CPU performance has become a necessity to meet the continuously-increasing demand for more computing performance (driven to large extent by needs in AI, edge and cloud computing). Flow intends to lead this revolution through its radical new Parallel Processing Unit (PPU) architecture, enabling up to 100X the performance boost in parallel functionalities. PPU can be applied to any CPU architecture and still maintain full backwards software compatibility.

2. What are the core benefits Flow offers better than its competitors?

Flow's architecture, referred to as Parallel Processing Unit (PPU), boosts the CPU performance 100-fold to enable a totally new era of "superCPUs". These superCPUs are fully backwards compatible with existing software and applications - parallel functionality in legacy software and applications can be accelerated by recompiling them for the PPU without any code changes. The more parallel functionality there is, the more performance boost will be subsequently gained. Our technology is also complementary in nature - while it boosts the CPU, all other connected units (such as matrix units, vector units, NPUs and GPUs) will indirectly benefit from the performance of PPU and get a boost from the more capable CPU.

3. How long has Flow been in business?

The company was established in January 2024 as a spin-off from VTT Technical Research Center of Finland.

4. How much money has the company raised to date?

In total, €4M.

5. What round of funding is the company currently on?

Flow closed the first funding round (pre-seed) on January 31st, 2024.

6. How many people does Flow employ?

The company has three co-founders and is aggressively increasing the headcount - we are trying to hire 10-12 technical experts to help build the product even further out than it already is.

7. Which angels and venture firms have invested in Flow to date?

Butterfly Ventures (Finland), Sarsia (Norway), Stephen Industries (Finland), Superhero Capital (Finland), and FOV Ventures (Finland/UK).

8. What were some of the key reasons VC's decided to invest in Flow?

Our investors were especially excited about the innovativeness and uniqueness of Flow's technology, its strong IP portfolio and the potential to enable a new era of superCPUs (CPU 2.0) for the AI revolution.

9. What are some major clients Flow is currently working with?
As we just came out of stealth mode, we are now looking to engage with the leading-edge CPU companies such as AMD, Apple, ARM, Intel, NVIDIA, and Qualcomm to co-develop the future era of advanced CPU computing. We are also looking to partner with smaller CPU startups like Tenstorrent and others.

10. Are there any geographies where Flow is especially concentrating - and if so, why?
We are working on a global scale, as the industry is truly global in nature and not limited to certain geographies.

11. Are there any target markets / products which Flow plans to especially address?
Flow's unique PPU architecture excels in general-purpose parallel computing and in the most demanding applications such as locally-hosted AI. We can also turbocharge embedded systems and data centers, for uses such as edge and cloud computing, AI clouds, multimedia codecs across 5G/6G, autonomous vehicle systems, military-grade computing and more.

12. What is Flow's business model, making it almost unique among semiconductor companies?
Our business model is to license our technology, just as ARM does, to various licensees around the world. PPU is totally independent of instruction set design, so it can be used in any CISC or RISC CPU and it can be integrated into any current or pending design architecture using any process geometry.

13. Who are the company's primary competitors?
Since PPU is a unique and one-of-a-kind product that we have invented, patented and trademarked, we do not have any direct competitors. In a sense, our biggest competitor is the fact that the industry and CPU manufacturers keep on using the current ways of improving their performance - e.g. adding more processor cores and increasing the clock frequency instead of utilizing new alternatives offering FAR higher performance.

14. How do PPU compare to existing chip units inside the die?
The blocks inside the CPU die are optimized and meant for different purposes - vector units for vector calculation, matrix units for matrix calculation - Parallel Processing Unit is optimized for parallel processing.

15. What are the core characteristics of Flow's new architecture?
To highlight the hardware advantages of Flow over traditional SMP/NUMA CPU (and GPU) computing, let us have a more detailed look at a number of differences between them:

A. **Nonexistent cache coherence issues.** Unlike in current CPU systems, in Flow's architecture there are no cache coherence issues in the memory systems due to the memory organization excluding caches in the front of the intercommunication network.

B. **Cost efficient synchronization.** Flow synchronization cost is roughly 1/Tb (Tb=fibers per PPU core) whereas in SMP/NUMA CPU systems it can be hundreds to thousands of clock cycles and in GPUs it can be from thousands to hundreds of thousands of clock cycles.

C. **Support for parallel computing primitives.** Flow's architecture provides unique and specific techniques/solutions for executing concurrent memory access (both read and write) operations, multi-operations for executing reductions, multi-prefix operations, compute-update operations and fiber mapping operations in the most efficient manner possible. These primitives are not available in current CPUs. Implementing these primitives in Flow-computing involves active memory technologies in the SRAM-based on-chip shared cache level providing potentially better performance and greater flexibility than current DRAM-based processing in memory (PIM) solutions, but not preventing the use of both techniques in the same design.

D. **Flexible threading/fibering scheme.** Flow-computing technology allows an unbounded number of fibers at the model level, which can also be supported in hardware (within certain bandwidth constraints). In current-generation CPUs, the number of threads is - in theory - not bounded, but if the number of hardware threads is exceeded in the case of interdependencies, the results can be very bad. In addition, the operating systems typically limit the number of threads to a few thousand at most. The mapping of fibers to backend units is a programmable function allowing further performance improvements in Flow.

E. **Low-level parallelism for dependent operations.** In Flow-enabled CPUs, it is possible to execute dependent operations with the full utilization within a step (with the help of chaining of functional units), whereas in current CPUs the operations executed in parallel need to be independent due to parallel organization of the functional units.

F. **Non-existent context switching cost.** In Flow-enabled CPUs, the fiber switching cost is zero whereas in current CPUs it is more than 100 clock cycles.

G. **Intercommunication traffic congestion avoidance.** In Flow-enabled CPUs, the probability of inter-communication network traffic congestion is low due to hardware support for hashing, concurrent memory access, multi-operations and multi-prefix operations. In current CPUs, congestion can happen frequently if the access patterns are non-trivial.

H. **Scalable latency tolerance.** Flow's technology provides a scalable latency hiding mechanism for constellations up to thousands of backend units, whereas in current CPUs the latency tolerance (with snooping/directory-based cache coherence maintenance mechanisms) appears to be relatively weakly scalable. There is also evidence that in high-end Flow-enabled systems, even the latency of DRAM -based memory systems can be hidden in many cases with suitable memory organization and sufficient bandwidth.

I. **No need for locality-maximizing memory data partitioning.** Flow-enabled CPUs are not prone to alternative memory partitioning schemes, whereas current CPUs are highly-sensitive to data placement.

J. **Sufficient intercommunication bandwidth.** Flow's PPU has an intercommunication network that is designed to provide sufficient bandwidth for random communication, whereas current CPUs are limited only to cases where most references are local. This kind of locality maximization is not always possible since in the general case there is no algorithm to maximize locality.

K. **Dual unit organization.** Current multicore CPUs were built by replicating processors originally designed for sequential computing and therefore optimized for low latency. As a result, they are relatively good for executing sequential workloads but have substantial performance issues with non-trivial parallel functionalities.

To support high-speed execution of parallel code, Flow introduces the PPU that utilizes the slack of parallelism to reorganize operations so that the requirement for low-latency is turned to the need for high throughput and combines it with a CPU.

The resulting dual unit CPU-PPU combines the best of both worlds to achieve the best performance for modern workloads containing a lot of parallelism but also some sequential parts while providing backwards compatibility via the CPU.

L. **Minimal disadvantages of superpipelining while having all benefits of that and full support for long latency, floating point and application-specific operations.** Flow-computing PPU is a fully superpipelined design with regular structure and patented support for long-latency operations, floating point operations and optional application-specific operations. In current CPUs, superpipelining increases the degree of pipeline delays that may cannibalize the performance benefits.

M. **Parametric design and instruction set independence.** Flow is not limited to single instances only, as it features parametric blocks with design time adjustable numbers of CPU cores, numbers of PPU cores, numbers and types of functional units per PPU core, size and organization of step caches, scratchpads, on-chip shared caches, latency compensation unit length, instruction set etc. Current CPU designs are typically tied to certain instruction sets and may require partial redesign if these kinds of parameters are altered.

N. **Support key patterns in parallel computing.** Flow supports the key patterns of parallel computation, such as parallel execution, reduction, spreading and permutation. Current-generation CPUs support only the parallel execution pattern.

16. What are the key differences between PPUs and GPUs?

PPU is optimized for parallel processing, while the GPU is optimized for graphics processing. PPU is more closely integrated with the CPU, and you could think of it as a kind of a co-processor, whereas the GPU is an independent unit that is much more loosely connected to the CPU.

Improving the CPU's parallel processing capability with PPU brings significant benefits. In GPU programming, the width of parallelism is fixed within a kernel, while the width in PPU can vary. This often brings inefficiency to GPU kernels, which are avoided in Flow Computing's PPU architecture. Starting a kernel involves some overhead, i.e. there is a minimum amount of work before outsourcing to the GPU will be profitable. In contrast, PPU works for a significantly wider range of programs because PPU can be utilized as an integral part of the code without creating a separate kernel.

17. Could a GPU take advantage of PPU architecture?
GPU's will most definitely take advantage (indirectly) from PPU usage in the CPU. While CPU performance is greatly improved due to PPU, **ALL** computations that happen between CPU and GPU will also benefit from this improvement. The overall performance of CPU plus GPU configurations will significantly benefit from PPU.

18. How does Flow's performance compare to Quantum Computing?
Comparing these two is like comparing apples to oranges. We think that Flow and Flow's PPU will significantly improve both the performance and effective life cycle of CPU-centric, von Neumann computing architectures. CPUs are small, relatively power- and cost-efficient and come with strong programmability - they are used for computation in various devices and embedded systems.

Quantum architectures, on the other hand, are meant for scientific large-scale calculations that do not scale for use in personal computing or mobile devices in the foreseeable future. Data centers could conceivably benefit from quantum, but they are EXTREMELY EXPENSIVE DEVICES that need to be kept at close to Absolute Zero degrees Kelvin - not an environment found in the average data center.

19. Isn't Quantum Computing the ultimate in State-of-the-Art today for performance?
Again we are speaking about apples and oranges here. Flow and Flow's PPU will enable a new era in CPU-centric, von Neumann computing architectures and power up the superCPUs of **TODAY's** designs. Quantum represents the state-of-the-art for certain classes of computation for scientific purposes, while Flow Computing is applicable to both general purpose personal ***plus ALL*** scientific computing.

20. What types of devices can Flow's PPU architecture most benefit from?
All devices that require high-performance CPUs will hugely benefit from Flow's PPU licensing.

21. Is Flow's architecture dependent on using State-of-the-Art manufacturing geometries?
Flow's PPU can be integrated into any current or pending design architecture or process geometry.

22. Is Flow foundry- or architecture dependent?
Flow is totally foundry and architecture independent - no changes to tools or processes are required. Also, it can be used with all instruction sets.

23. How much die space does adding a PPU require to achieve 100X performance over standard architectures?
It depends on the system configuration. In case the number of processor cores is high, it is expected that several CPU cores could be substituted by the PPU. Then PPU uses the leftover die space without the need to add any extra silicon area.

Our initial silicon area estimation model is based on legacy silicon technology parameters and public scaling factors. For the 64-core PPU that achieves 38X - 107X speedup in laboratory tests, the initial silicon area estimate is 21.7 mm^2 area in 3 nm silicon process. The silicon area estimate for a 256-core PPU achieving 148X - 421X speedup is 103.8 mm^2, respectively.

24. Is there a theoretical limit on what kind of PPU can be added to a typical mobile, PC or server CPU?
PPU is a parametric - it can be configured for any desired use case: number of PPU cores (e.g. 16, 64, 256…), number & type of functional units (e.g. ALUs, FPUs, MUs, GUs, NUs), size of on-chip memory resources (caches, buffers, scratchpads), etc. The performance boost scales up with the number of PPU cores - for very small devices (e.g. a smart watch) a PPU with 16 cores would work well, PPU with 64 cores would be suitable for smartphones and PCs, while PPU with 256 cores would likely be the most suitable configuration for AI and edge computing servers.

25: How much additional power draw does a pipeline of PPUs typically draw?
PPU's can be actually configured to REDUCE power consumption! Due to the parametric nature of PPU, the uptake in performance can be used to save power used for processing - 100x performance boost could be traded for 10x performance boost with 10x less power consumption, as just one example.

The power consumption depends heavily on the desired configuration. Our initial power consumption estimation model is based on legacy silicon technology parameters and public scaling factors, which indicate 43.4W consumption for a 64-core PPU giving 38X - 107X speedup in laboratory tests if 3nm silicon process would be used and 235W consumption for a 256-core PPU giving 148X - 421X speedup, respectively.

26. Don't GPU's already provide parallel processing capabilities in both the rasterization and geometry pipelines? Why add to the CPU?
Improving the CPU's parallel processing capability with PPU brings significant benefits. In GPU programming, the width of parallelism is fixed within a kernel, while the width in PPU can vary. This often brings inefficiency to GPU kernels, which are avoided in Flow Computing's PPU architecture. Starting a kernel involves some overhead, i.e. there is a minimum amount of work before outsourcing to the GPU will be profitable. In contrast, PPU works for a significantly wider range of programs because PPU can be utilized as an integral part of the code without creating a separate kernel. Moreover, CPU and GPU memories are normally separate, which leads to memory consistency challenges.

27. What is the expected cost delta (SRP) between adding a PPU to a CPU vs. buying a standard GPU card?
The PPU architecture will be integrated into the CPU and thus cannot be bought as a separate unit. In addition, licensing costs are dependent on the number of cores and other variables that make the answer impossible to quantify in a general sense.

28. What kind of PPU would need to be added to a CPU to equal the performance of a 2023-era NVIDIA RTX 4090, the expected performance of the new NVIDIA Blackwell?
Our target is not to replace the GPU, but to improve the performance of the weakest link of computation - CPU. Better CPUs powered by Flow's PPU will benefit the performance of the overall configuration, including GPU. CPUs, and future superCPUs featuring Flow's PPU are aimed mostly for different functionalities than GPUs such as RTX 4090. When the functionality requires non-trivial access patterns or contains inter-thread dependencies, superCPUs with Flow's PPU will be much faster than GPUs.

GPU vs. CPU is a bit beside the point - the more interesting reference point is current CPU and next generation superCPU with PPU. NVIDIA does have its own Grace CPU Superchip with 144 cores (that is a 2x72 core). If they would have a CPU with 72 cores coupled with 64 cores of PPU, it would likely have much better performance than the current version of Grace CPU. And Grace CPU with Flow's PPU coupled to configurations with powerful GPUs like Blackwell would raise the performance bar tremendously.

29. What CPU vendors are most likely to consider licensing the Flow PPU architecture and why?
Our vision is that PPU is used in all future high-performance CPUs. The most likely CPU vendors to license Flow's PPU are the leading-edge CPU companies such as ARM, AMD, Intel, Apple, NVIDIA, and Qualcomm. We also expect to be very interesting for companies that use custom CPUs for specific use cases that need the utmost from CPU performance. Most RISC-V CPU companies are successfully developing these types of custom CPUs.

30. Can Flow really be used in anything from a mobile phone to a Supercomputer?
As PPU is both configurable and parametric, it suits a wide range of different use cases. The number of PPU cores (e.g. 4, 16, 64, 256…), the number & type of functional units (e.g. ALUs, FPUs, MUs, GUs, NUs), the size of on-chip memory resources (caches, buffers, scratchpads) are all parametric.

The performance boost scales up with the number of PPU cores - for very small devices (e.g. a smart watch) a PPU with 4 cores is eminently suitable, while a PPU with 16 cores is suitable for smartphones and 64 cores for PCs. A PPU matrix with 256 cores or more would likely be the most suitable configuration for a Supercomputer.

31. What are the use cases for Flow in AI?
Data pre- and post processing currently chews up 50% of the total time when teaching an LLM a new language. This can be **significantly** reduced with high-performance PPU-powered CPUs. In addition, locally-hosted AI would now become feasible.

32. What are the use cases for Flow in base stations?
Multimedia codecs used over 5G/6G networks would become **significantly** faster, allowing new use cases and applications. In addition PPU can be configured to **REDUCE** power consumption! Due to the parametric nature of PPU, the uptake in performance can be used to save power used for processing - 100x performance boost could be traded for 10x performance boost with 10x less power consumption.

33. What are the use cases for Flow in Supercomputers and the defense industry?
Our technology can **dramatically** improve standard Supercomputer performance! In addition PPU can be configured to **REDUCE** power consumption! Due to the parametric nature of PPU, the uptake in performance can be used to save power used for processing - 100x performance boost could be traded for 10x performance boost with 10x less power consumption.

In the defense industry - missiles, drones and missile and drone defense devices are the most lucrative use cases, alongside military aviation. Whoever processes the data and calculates the fastest will win in warfare - Flow does have a major geopolitical defense impact.

34. Why would compute powerhouses like AMD, Apple, ARM, Intel, NVIDIA, and Qualcomm ever consider licensing Flow PPU when they have billions of dollars/Pounds Sterling already invested in their own designs?

The mentioned companies are constantly looking for the next big thing to improve their products. Technologies used in current multicore CPUs by these companies can't be used to solve inefficiencies popping up when executing parallel functionalities. Flow is developing a unique product in PPU that will enable these computing powerhouses to step into a new era of superCPUs - "**CPU 2.0**". PPU is complementary in nature, benefiting all CPU computations and instruction sets and usable in independent fitting for all existing fabs, foundries, tools and processes, so PPU fits in easily into their own designs. We are now looking to engage with these leading-edge CPU companies to co-develop the future era of CPU computing. nd elegant. Parallel portions of the code can be expressed as natural parallel statements without having to worry about race conditions, deadlocks and synchronizations.

37. What are the tradeoffs between maintaining full backwards software compatibility with an existing architecture such as x86 or Snapdragon or ARM vs. going for maximum performance?

ALL existing software is compatible with CPUs that have PPU matrices built-in. The performance boost level depends on the amount of parallelism in the software - the more parallelism, the more boost PPU will generate without any software code changes using only recompiling. If the libraries are already optimized for Flow, even more performance gains are achieved without any additional steps.

For maximum performance gains, it is possible to refactor the critical parts of the code or rewrite it entirely as a native parallel. We will be developing AI tools to help companies to identify which parts of their software can be parallelized.

38. Licensing a core architecture from a new, unknown startup is a BIG ask for a company like Intel (or any other one, for that matter) - how can Flow assure a potential licensee that the architecture will be around for the foreseeable future?

This is a legitimate concern - we are planning to increase resourcing and scaling-up capabilities such as technical customer support and other variables to provide rock-solid assurances to our customers. Our investors are fully committed to backing us in our growth journey and we are obviously looking to bring new investors onboard in future financing rounds. With positive market traction, we are certain that we are able to continue growing to meet the rigorous demands of our licensees. Our customers with an architecture license will have full technical control over their design.

39. It all sounds great - but if it's so amazing, why haven't the multi-billion dollar chip companies already done it? What's the catch?

Parallel processing is not a "new kid on the block" in computing - it's actually been around conceptually since the 1970's, culminating with the Inmos Transputer from the 1980's! <https://www.wikiwand.com/en/Transputer>. It was never mainstream in these earliest days of the PC due to its programming complexity and inefficient architecture - as a result, architectural selections in the past led to the current multicore CPUs replicating processors that were originally optimized for sequential computing.

As a result, the industry has learned to settle for incremental performance gains that are achieved by adding cores and clock-frequency. Moreover, the industry has become complacent with cumbersome and unproductive programming techniques. We have been driven by these and have therefore meticulously continued researching and developing parallel processing technology. With PPU and our technology stack, we are able to finally combine all the benefits of current CPUs and parallel processing.

40. Who are Flow's founders and what makes them qualified to be creating all of this?
The Flow founders are Dr. Martti Forsell, Jussi Roivainen and Timo Valtonen. Dr. Forsell has been researching parallel processor architectures and programming for several decades, first at the University of Joensuu (later called University of Eastern Finland) and later at the VTT Technical Research Center of Finland where Jussi Roivainen joined his research team. Timo Valtonen joined the team to drive and plan the commercialization of the research and technology.

The initial idea was to develop the fastest CPU in the world! Alongside this original idea, the team started in parallel (bad pun, sorry) to explore the possibility to instead make a product that could be used by **all** CPU manufacturers. PPU and Flow were born from this - alongside the vision that Flow's PPU would be used in all high-performance CPUs to start a new era in CPU computing. Years of joint work has led to this point and the founders now have a funded company to fulfill that vision.

41. Who are Flow's advisory team members?
Flow doesn't yet have an advisory team. We are currently evaluating candidates to form an advisory team later this year, balanced between marketing and technical expertise.

42. Is Flow announcing immediate availability of this IP platform in its entirety? If not in May/June, when?
Flow is exiting stealth in May/June with the announcement of its incorporation and funding and basic details of its patented PPU architecture. Flow is still developing its IP platform and product further, so stay tuned for our future progress and the full details of our technical innovations in Q4 of 2024!

43. Are there any fabs currently aligned with Flow? If not, which are the ones most suited to deploy this?
Not at the moment. Flow is totally fabless and foundry and architecture independent - no changes to tools or processes are required. Also, it can be used with all instruction sets. Thus, CPUs with PPU can be deployed by **any** fab.

44. How does Flow provide its IP to a licensee? VHDL source, source code, final design schematics, compiled software?
Our target is to provide our IP to the RISC-V market as Soft IP, i.e. synthesizable HDL. For the ARM and X86 markets we will offer architecture type licensing, which allows the use of our patents and other IP for implementing customer's own PPU.

45. Are there any open-source components to Flow's compiler platform? If yes, under what license?
Not at the moment.

46. Is there a GitHub repository for code snippets and examples?
Not at the moment.

47. Does Flow plan to offer regional offices in other geographies?
Flow Computing is headquartered in Finland and the current Flow team is spread across Europe in several European countries. In the future, our plan is to have an office in the USA.

48. Where was Flow's original design architecture conceptualized?
Dr. Martti Forsell started researching parallel processor architectures and parallel computing at the University of Joensuu (later titled University of Eastern Finland) and continued his research in VTT Technical Research Center of Finland together with Jussi Roivainen. Flow Computing technologies and patents are the result!

49. Does VTT Research still own a stake in the company or rights to any of the IP?
VTT is a minority owner of Flow Computing after a significant IP apport to the company. We have full rights and ownership of the IP and are already generating new IP.

50. What is the estimated performance benefit of using Flow Computing technology, in particular how are the parallel computing performance gains likely to translate into gains at full application level?
The question of performance benefit is answered separately for software that is specifically written with Flow Computing technology in mind, and legacy software. For the latter, availability of source code allows programmers to re-compile with a compiler that is aware of Flow Computing technology and can provide performance improvements that are detected automatically by Flow's compiler. A definite advantage of Flow Computing is the possibility to run "classic code" on the CPU cores and exploit parallelism on the PPU cores whenever it occurs in an application.

Think of it as similar to the original PowerPC chip in 1990's Macs - older 68K software ran in compatibility mode while new software ran much faster using the PowerPC instruction set. A further performance benefit for legacy code can be obtained if operating system or programming system libraries (e.g. sorting via qsort() in the C library) can be re-compiled for Flow Computing and then run faster on the PPU's cores even if the application code itself is unmodified. There will be significant performance gains on most types of applications, especially for those that exhibit degrees of parallelism but which cannot be parallelized with current, thread-based schemes.

51. For what type of algorithms does Flow's technology likely work and how widely it can be applied, are there some specific areas it is not likely to work?
A currently growing field where Flow's technology is highly applicable is Artificial Intelligence. Both machine learning (e.g. training of neural networks) and symbolic AI (often searching through large graphs) are easily portable to Flow, as these applications currently are often used on GPUs. Yet, the regularization necessary in GPU computing limits parallelization on GPUs, something which can be overcome by the flexibility of Flow Computing.

In addition, fields such as (numeric) or (combinatoric) simulation and optimization, which are widely used in business computing (from logistics planning to forecasting investments) will greatly profit from Flow. Such applications tend to be heavily parallelized, often for GPU clusters, and will benefit from the flexibility of thick control flows over GPU thread blocks. Flow's technology also works for the classic numeric and non-numeric parallelizable workloads ranging from matrix or vector computations to sorting.

In code with parallelizable parts that are small, and which are currently not parallelized because runtime overhead is larger than runtime benefit from parallelization, it still brings a performance boost.

52. What is the applicability of Flow's technology: How widely Flow technology could be adapted?
Flow's technology is adaptable to any microprocessor-based system that currently uses multiple processor cores and/or massively-parallel accelerator devices (such as GPU, Intel Xeon Phi, or others), as the CPU's cores do the work of the processor cores and the PPU's cores play the role of the accelerator. In addition, PPU cores are able to take over work from the CPU's processor cores which is parallelizable, but currently is not outsourced to the accelerator because the parallel work is not regular enough or not large enough to pay off the overhead.

Thus, the technology becomes widely adaptable, starting from classic desktop and tablet computers to embedded systems and digital signal processors to smartphones, where a part of the workload is not from the user (such as video decoding) but is generated from the system itself (e.g. for software parts of the radio part, which are numeric such as digital signal processing applications or at least high throughput such as network stack processing). The current microprocessor-based systems face the unenviable situation where operating frequencies cannot be increased anymore, so the performance boost for a single application must come from the use of parallelism and no longer from faster devices. Further, as memory devices do not become correspondingly faster, access to main memory is quite slow when measured in processor core cycles (hundreds of cycles), so main memory access must be avoided as far as possible by the use of fast but small caches to store frequently accessed data items.

A good use of caches necessitates a programming style leading to particular memory access patterns, which often hinders parallelization. Flow technology brings advantages in both areas: PPU allows a more flexible, and thus more widespread use of parallelization than with parallel threads, which also supports programmability and thus programmer efficiency. Emulated shared memory hides the long latencies instead of avoiding them, by exposing them to the CPU's processor cores and thus allowing the CPU processor cores to better schedule other work for execution while PPU waits for a memory read.

53. In what way does Flow differ from and benefit over an architecture where CPU and GPU are combined on a single silicon chip?

PPU provides better utilization of compute resources than GPUs because in PPUs the amount of parallelism can be dynamically set to follow the optimum while in GPU it is more or less fixed. Processing in PPU starts immediately as a part of a CPU program, whereas in GPUs, one needs to launch a kernel that is executed outside the CPU. Starting a kernel involves some overhead, i.e. there is a minimum amount of work before offloading to GPU will be profitable. Moreover, CPU and GPU memories are normally separate, which necessitates explicit data transfers or implicit synchronization overhead in mapped memory regions.

In GPU programming, the width of parallelism is fixed within a kernel, while the width in PPU can vary. This causes inefficiency in GPU kernels that are avoided in Flow's PPU. Starting a kernel involves some overhead, i.e. there is a minimum amount of work before offloading to GPU will be profitable.

In contrast, PPU can directly start in the program execution which enables PPU usage for smaller workloads and for a wider range of programs. Moreover, CPU and GPU memories are normally separate,

which necessitates explicit data transfers or implicit synchronization overhead in mapped memory regions.

Unifying CPU and GPU memory in one physical memory will thus bring its own challenges both on the hardware design front and on the level of programming. As an example, the question as to when memory writes done by the CPU will be visible to the GPU and vice versa leads to memory consistency models that normally complicate the arguing about program correctness, or lead to a safe but inefficient programming style. To summarize, programming for CPU+PPU is more comfortable than programming CPU+GPU, with significantly added flexibility.

54. Some CPU accelerators introduce extra delays in computation. Does PPU suffer from this problem?

Flow's PPU is tightly connected to the CPU. There will be a minor latency in passing parameters to the PPU, but it rarely causes delays due to natural overlapping of PPU and CPU operation. Even if there would be a minimal delay, the performance gains of the PPU eliminates the possible minor slow down by a large margin. Sequential legacy code will be executed in the CPU without PPU involvement, thus the latency remains unaltered in this case.

When PPU is executing the parallel parts of the code, it has a different approach to latency than the CPU - rather than minimizing the latency of individual instructions, it exploits the slack of parallelism to maximize the throughput. This is, e.g., used to hide the latency of memory operations by executing other threads while accessing the memory. Need for cache coherence maintenance traffic is eliminated by placing no caches in the front of the intercommunication network. Scalability is provided via a high-bandwidth network-on-chip ultimately supporting memory access needs of general parallel computing.

-END-

FOR FURTHER INFORMATION, PLEASE CONTACT FLOW COMPUTING PR COUNSEL JONATHAN HIRSHON AT JH@HORIZONPR.COM