Team Project Sprint 1

Devin Kot-Thompson & Chih Han “Josh” Yeh

EC 601 Osama Alshaykh

Black Parrot and Its Use In Hardware Accelerator Concepts

Project Definition

The Black Parrot RISC-V Processor is an open source project that provides a heavily verified processor to the public for incorporation in various applications. The open source of the RTL and the hardware design allows for a deeper understanding of the microarchitecture. This is particularly useful for learning and performing experimentation in hardware design. In this project, the Black Parrot processor will be studied extensively to develop a strong understanding of its operation and to investigate potential applications that utilize its strengths. One such application is the use of the processor as a host for hardware accelerators. The goal of this project is to incorporate the Black Parrot Processor in a hardware accelerator design.

Target Users

RISC-V is a very popular topic in the world of hardware and processor architecture. It is gaining interest because of the capabilities and open source nature of the ISA. Being open source allows for a high level of accessibility when designing custom hardware solutions. Due to these advantages over proprietary ISAs it is gaining traction in both research and corporate markets. One popular area in both research and corporate worlds is hardware acceleration. This is the process of offloading intensive tasks from the processor in a digital design. The advantages of this approach is that the intensive tasks are performed with much greater efficiency due to the specified nature of the hardware accelerator. Additionally, the processor is free to manage the system without being hampered by intensive operations. Black Parrot provides the opportunity to develop one's own hardware accelerator for intensive streaming applications. Users in both research and in industry can use this design in a modular approach to create solutions for specific applications.

User Stories

A user of Black Parrot for hardware acceleration could be anyone who wants a design for an intensive data processing application without the overhead cost of developing every intellectual property from the ground up. Black Parrot provides a silicon verified modular solution that can be interfaced with a custom hardware accelerator. For example, someone interested in an application specific digital design wants an open source processor that allows them to create a product free of proprietary considerations and that can be customized to a high level. Black Parrot serves as a good fit to this story as it follows the RISC-V ISA and has a BSD-3 license.

Minimum Viable Product

To begin the project, Black Parrot needs to be well understood. This will be necessary to develop it into the host core for a hardware accelerator SoC. A minimum viable product for this hardware accelerator SoC will include a review explaining the operation in detail. Additionally, simulations, block diagrams, and potentially some HDL for different sections of the hardware accelerator could be contributions toward a proof of concept.

Literature review

Black Parrot is a single cycle pipelined RISC-V processor with 8 stages [1]. Single cycle means one instruction is executed every clock cycle and pipelined means the datapath is broken into stages so multiple instructions can be buffered through to increase throughput. This is because the clock period is roughly divided by the number of stages. In a 5 stage pipeline design the stages are typically Fetch, Decode, Execute, Memory and Writeback. Fetch is the stage in which an instruction is fetched from the Instruction Cache after being provided an address by the program counter. Decode is the stage in which source operands to the register file are decoded to activate appropriate control lines, access registers and prepare the architectural state to execute the encoded operation. Execute is the stage containing the ALU. Memory contains the Data Cache which is read to or written to depending on the instruction. Finally, Writeback is the stage where the result of the operation can be written into the Register File, if applicable. The separate instruction and data caches allow instructions and data to be accessed in the same cycle. The advantages of this architecture come at the cost of logic and non architectural pipeline registers to handle the dependencies of simultaneous instruction execution[10]. Black Parrot has all these stages but with increased complexity.

The Instruction Fetch is broken into 3 stages. The Memory stage has sections for a Multiplication Pipe and Floating Point Pipe[5]. Despite this increased complexity Black Parrot follows the similar operation to the 5 stage pipelined design. It can be categorized into 3 sections: Front End, Back End and Memory End. The Front End (FE) is responsible for PC generation and instruction access through the cache. It performs branch prediction and the architectural state is speculative here[1].

Branch prediction is a process in which the CPU would guess the outcome of conditional operations before they are executed, this is to keep the instruction pipeline full and avoid stalls. Speculative execution is an optimization process where the CPU chooses whether or not to advance in certain tasks based on branch predictions. If the prediction is correct, the results are committed and executions will continue, on the contrary, if it is incorrect, the speculative result is discarded and the CPU will roll back to the correct execution path. BlackParrot's FE uses several components for branch prediction[9]. Branch History Table(BHT), Branch Target Buffer (BTB), and Return Address Stack (RAS), with the BHB and BTB making an initial prediction, then backing prediction using partial instruction decode BHT and RASm this allows BlackParot to fetch instructions along predicted paths.

On the flip side, BlackParrot's Back End (BE) handles non-speculative execution of instructions. It will process the stream of speculative instructions from the FE in order through severe key components: the Director which holds the speculative Next Program Counter (Next PC or NPC) and communicates with the FE to correct the fetching path on the occurrence that the branch mispredicts or other similar cases, the Control Pipeline which executes the control flow instructions and can perform early writeback of the speculative NPC to the Director, and lastly the System Pipeline which is responsible for modifying architectural states and handling asynchronous events like interrupts.

In its current version, BlackParrot supports up to 5 types of predictor implementations, bimodal: using the lowest bits of the branch address as a hash function, gshare: XORs the lowest bits of the branch address with branch history, gselect: concatenates the lowest bits of the address with recent branch history bits, tournament: a hybrid approach where one branch predictor uses the branch address only as a hash function, while the other one uses the branch history shift register only, two-level local: which uses a correlation table and branch history table for its predictions. Having multiple branch predictor options, BlackParrot allows designers to choose the best implementations based on their specific needs, an approach through their extensive FE and BE designs offering flexibility, and a balance between accuracy and hardware constraints.

Outside of branch predictions and speculative execution, the BE controls PC generation that is external from sequential fetching. Additionally, it executes instructions using the integer ALU, Multiplication Unit, Floating Point Unit, or System Unit containing Control and Status Registers, and Memory Unit for the Data Cache. The Memory End contains the cache coherence logic called BedRock Engine in a multicore configuration and a state machine for the Data Cache in a single core configuration[1]. Cache coherence is a critical aspect of digital design for multicore or distributed memory systems. This makes it important to hardware accelerators which may want to access main memory separately from the processor. Cache coherence becomes a problem when a piece of data from memory is shared across multiple processor caches and the data becomes modified and inconsistent across the caches and from main memory[2]. The BedRock CCE handles this problem using a Directory Based cache coherence approach. The Directory holds information on where copies of memory are stored, the current state of that copy, block address of the memory, and owner of the block. The states are Modified, Exclusive, Shared and Invalid. Modified is one cache that has the only valid copy and it has been modified. Exclusive is one cache that has the only valid copy and it has not been modified. Shared multiple caches have a copy and none are modified. Invalid is the block that is no longer valid in the cache. When the processor wants to read or write a memory block it sends a request to the directory. The directory then takes actions such as invalidating other copies and retrieving recent data[1].

BlackParrot's Microarchitecture

As the use and requirements of computational processes have increased exponentially through the years, technological development for computational power has also increased to meet that demand. One of those ways the demand is met is through the process of hardware acceleration; using specialized hardware[11] such as the graphics processing unit to offload certain demanding tasks away from the CPU, as the processor would typically sequentially execute tasks. Modern CPUs are now typically built with accelerator-host processing architecture, which means instead of just containing a general-purpose processor as the host, it would also contain single or multiple hardware accelerators, such as DSPs, iGPUs, FPGAs, and many more.

The BlackParrot processor's goal is to act as an accelerator host multi-core system. It is designed as a “scalable, heterogeneously tiled multi-core microarchitecture”[5], if you were to define BlackParrot by this statement term by term, it means the processor is designed to be capable of increases and decreases in the number of cores and other components to meet different demands (scalability), utilizes distinctly differently designed cores for different tasks (heterogeneous), and each processor component is organized into modular, repeatable units (tiles).

In the case of BlackParrot, their microarchitectural tiles fall into four different categories[5]. The **BlackParrot Core Tile**, a tile that contains a processor with coherent caches (data consistency caches), a directory shard that keeps track of caches with specific memory blocks, and a portion of the Level 2 cache (L2 cache), allowing the state of cache lines across the system to be tracked. An **L2 Extension Tile**, a modular component allowing for easy expansion of the Level 2 cache within the BlackParrot system, contains a directory and a non-inclusive, non-exclusive L2 cache slice; allowing scalability to the L2 cache capacity, and non-interfering with critical paths in the cores or the Network-on-Chip (NoC). Next is the **Coherent Accelerator Tile** which is completely modular, and in simple terms viewed as a local cache engine (LCE). Lastly, there is the **Streaming Accelerator Tile** which has no locally cached memory and does not control any physical memory, they are used for basic I/O devices, network interfaces, or even heavier stream-based accelerators like GPUs.

The BlackParrot processor utilizes three different classes of NoCs, the communication system that connects the various components within its SoCs. The aforementioned **BedRock** network is a cache-coherent fabric that connects all of the tiles in the systems using three logical channels (request, command, and response) to manage communication between Local Cache Engines (LCE) and Cache Coherence Engine (CCE), it is implemented as a 2-D mesh network. **DRAM Network** is a lightweight 0.5-D network that connects the CCEs to memory devices like DRAM, Flash, on-chip ROMs, optimized for wormhole routing and designed to handle memory requests efficiently. The **IO Network** is a 1-D wormhole network in the I/O complex that connects the processor to various peripherals and external devices, serving as a transducer between the processor’s protocols and standard I/O protocols.

References

[1] S. V. Ranga, “ParrotPiton and ZynqParrot: FPGA Enablements for the BlackParrot RISCV Processor,” _Bespoke Silicon Group_, 2021. https://www.bsg.ai/papers/theses/Shashank_Thesis_Final.pdf (accessed Sep. 23, 2024).

[2] “Cache Coherence I – Computer Architecture,” _www.cs.umd.edu_. https://www.cs.umd.edu/~meesh/411/CA-online/chapter/cache-coherence-i/index.html

[3] Z. Azad, Sadullah Canakci, S. Davidson, and A. Joshi, “BlackParrot: An Open-Source RISC-V Multicore Processor A core for and by the world!,” ResearchGate, Jan. 31, 2020. https://www.researchgate.net/publication/366177341_BlackParrot_An_Open-Source_RISC-V_Multicore_Processor_A_core_for_and_by_the_world (accessed Sep. 23, 2024).

[4] M. Taylor, “BLACKPARROT, AN OPEN SOURCE RISC-V MULTICORE, AND ITS ECOSYSTEM See additional restrictions described on inside pages STINFO COPY AIR FORCE RESEARCH LABORATORY SENSORS DIRECTORATE WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320 AIR FORCE MATERIEL COMMAND UNITED STATES AIR FORCE,” 2023. Accessed: Sep. 23, 2024. [Online]. Available: https://apps.dtic.mil/sti/trecms/pdf/AD1211452.pdf

[5] D. Petrisko _et al._, “BlackParrot: An Agile Open-Source RISC-V Multicore for Accelerator SoCs,” _IEEE Micro_, vol. 40, no. 4, pp. 93–102, Jul. 2020, doi: https://doi.org/10.1109/mm.2020.2996145.

[6] M. Wyse _et al._, “The BlackParrot BedRock Cache Coherence System,” Nov. 2022, Accessed: Sep. 23, 2024. [Online]. Available: https://arxiv.org/pdf/2211.06390

[7] A. Waterman, Y. Lee, D. Patterson, and K. Asanović, “The RISC-V Instruction Set Manual, Volume I: User- Level ISA, Version 2.1,” 2016. Available: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-118.pdf

[8] RISC-V International, “RISC V ISA & Foundation Overview,” _YouTube_. May 20, 2018. Accessed: Nov. 21, 2023. [Online]. Available: https://www.youtube.com/watch?v=GWmo8y-eR0c

[9] black-parrot, “BlackParrot: A Linux-Capable Accelerator Host RISC-V Multicore,” GitHub, Jan. 29, 2022. https://github.com/black-parrot/black-parrot

[10] S. L. Harris and D. Harris, Digital Design and Computer Architecture, RISC-V Edition. Morgan Kaufmann, 2021.

[11] R. Merritt, “What Is Accelerated Computing?,” NVIDIA Blog, Sep. 01, 2021. https://blogs.nvidia.com/blog/what-is-accelerated-computing/