Power, Performance and Portability: System Design Considerations for Micro Air Vehicle Applications

Yakun Sophia Shao*, Judson Porter*,
Michael Lyons*, Gu-Yeon Wei*,
David Brooks*,[1]

* *Harvard University, 33 Oxford St., Cambridge, MA 02138*

ABSTRACT

Recent years have seen an increased interest in Micro Air Vehicles (MAVs) with applications ranging from search-and-rescue to mimicking insect behavior. MAVs have several challenging design requirements that impact processor design. These include real time processing demands and severe power/weight budgets. In this paper, we describe the characteristics of MAV applications and propose hardware acceleration to improve the power, performance, and portability of MAV system designs.

KEYWORDS: Micro Air Vehicles; Hardware Acceleration;

1 Introduction & Background

Micro Air Vehicles (MAVs) are a subclass of Unmanned Air Vehicles (UAVs) with severe size and weight restrictions. The most extreme scale MAVs are the size of insects[Woo07]. The size and flight attributes of MAVs provide advantages in flying into hazardous environments which might be dangerous or inaccessible to humans. For example, MAVs could be used to conduct aerial power line inspection or to search for disaster victims. The scale of MAVs has also inspired researchers to consider applications that mimic insect behavior. One compelling example is motivated by a recent phenomenon of large-scale honeybee depopulation called Colony Collapse Disorder (CCD). Our research team at Harvard University is developing robotic bee MAVs to pollinate flowers as a short-term CCD solution [RB]. MAV research has flourished in the last several years due to developments in microtechnology, sensors, energy-storage techniques, and low-power computation.

However, limited power and weight budgets constrain the use of MAVs. A MAV requires a power source to drive all aspects of the system, including flight apparatus, sensors, computation and communication. The power source must be compact and light as it is part of the MAV flight payload severely limiting system energy capacity. All components of MAV

---

[1] E-mail: {shao, jlporter, mjlyons, guyeon, dbrooks}@eecs.harvard.edu

systems need to be power efficient. Computation in current MAV designs is typically provided with off-the-shelf general purpose microprocessors. However, these generic microprocessors are not specifically designed for MAVs, which limits the energy efficiency of the system. To address this limitation, we highlight several characteristics of MAV applications and propose an accelerator-based approach to achieve power-performance efficient MAV processor design.

2 Characteristics of MAVs

From the computer architecture perspective, we must explore design tradeoffs of processors for MAV applications. During the initial phase of the design process, we need to understand the typical computational needs of the MAV workloads. Given the functional requirements of MAVs, the processor needs to perform the following the three broad classes of tasks: Flight Control, Visual Guidance, and Target Detection.

2.1 Flight Control

Designing an autonomous flight system requires accurate and stable control algorithms implemented in the MAV processor. It reads environmental data from sensors, estimates the current flight state, compares the state with target movement, and outputs the actuation signals. MAVs deploy a diverse set of sensors including inertial sensors, visual sensors, and accelerometers, leading to different control algorithms. Given the critical timing requirements of MAV flight, the flight controller needs to process the incoming data in real time.

2.2 Visual Guidance

Autonomous MAV applications require environmental sensing to regulate and avoid obstacles during flight. Inspired by the compound eyes of insects, computer vision researchers have developed optical flow to describe the relative movement of the environment. Optical flow is the pattern of apparent motion of objects in a visual scene. Based on successive images from visual sensors, optical flow can estimate the movement of surrounding objects. Optical flow algorithms have high computational requirements, including Sobel filters, matrix multiplication, and matrix inverse. Moreover, MAVs require real time optical flow processing as a prerequisite for stable flight control.

2.3 Target Detection

Many MAV applications require the identification of interesting objects. For example, robotic bee MAVs will detect flowers using UV sensors coupled with visual detection algorithms. Visual detectors read in a series of images from optical sensors and identify if a particular object or condition is present. Similar to optical flow algorithms, visual target detection generally employs matrix operations, Gaussian filters and gradient computation.

2.4 Characteristics

The above discussion presents the functional requirements of MAV applications. In general, characteristics of MAV designs include:

1. Repetitive tasks: Typical MAV applications can be broken down into repetitive tasks like flight control and target detection, which are essential for functionality.

2. Real time processing: Tasks related to flight behavior, for example obstacle avoidance and flight control, require real time processing.

3. Power budget: Achieving computational functionality and performance requirements under a limited power budget present significant challenges in architecture design.

3 Architectural Design Considerations

Given the characteristics of MAV workloads, general purpose computing is too inefficient to address power and performance requirements simultaneously. Instead, we propose an accelerator-based computing paradigm employing several specialized hardware resources. We envision MAV processors that consist of several hardware accelerators to handle repetitive tasks. The processor also includes a General Purpose CPU (GPCPU) to handle non-repetitive computations and to serve as "glue logic" between accelerators. Accelerator-based architectures can provide orders of magnitude gains in energy efficiency compared to general purpose processors for those tasks that run frequently and require a non-trivial amount of computation.

3.1 Case Study: Optical Flow Accelerator

As discussed in the previous section, optical flow is an essential algorithm for autonomous flight control. Optical flow can be utilized to identify the current MAV movement, and to navigate and avoid obstacles in the forward path. Lucas-Kanade is one of the most popular optical flow algorithms [LK81]. The algorithm begins by reading in two successive images from optical sensors. It then uses a Sobel filter to compute the horizontal and vertical gradients of each image. Finally, based on the brightness constraint function, it performs matrix inverse and multiplication to calculate the optical flow in horizontal and vertical directions. The algorithm is arithmetic-intensive with simple control flow. In addition, since many operations in the first and second stages are pixel-independent, significant parallelism exists.

Given the computational characteristics of optical flow and real time processing requirements, high-end general purpose microcontrollers or DSPs might meet the challenge. However, the power budget of MAV systems disqualifies these options. On the other hand, low-end microcontrollers are not able to meet the real time processing requirements due to lack of computational resources. In fact, researchers in this area have tried to use hardware acceleration to address the challenge. For example, one commercial MAV from Centeye implements a custom "vision chip" to perform front end image processing, and an off-the-shelf microcontroller to complete the optical flow computation [BYNL09]. Hardware acceleration for key MAV workloads would improve the power-performance efficiency of MAV systems.

3.2 Research Questions & Conclusion

Accelerator design naturally leads to several research questions. First, we must consider the best methodology for translating algorithmic needs to computing needs. We suspect this translation can be automated, both in terms of identifying partitioning algorithms into hardware and software, as well as implementing hardware accelerators from algorithms.

Second, we must address the best level of computational granularity when designing accelerators. When designing an accelerator for a task, we should tie several smaller accelerators together rather than developing one large accelerator. This approach will result in smaller accelerators that can be reused for other algorithms, but we must take care to ensure accelerators do not become too small and numerous to avoid unnecessary overheads.

Third, we must evaluate the tradeoff between GP-CPUs and hardware accelerator-based computation. As previously noted, accelerated hardware enhances power and performance efficiency, but is less flexible than general purpose logic. We therefore must identify which tasks are best suited for GP-CPUs, such as accelerator coordination and rarely executed tasks.

Fourth, we must also consider tradeoffs in specific accelerators. For example, optical flow performance and energy consumption is greatly affected by frame rates and video resolution. In addition, a focus on one stage of sensing can affect others. A more robust optical sensor would require a simpler vision detection algorithm, or vice versa.

Fifth, we must investigate various circuit implementation and design styles for each accelerator. Digital CMOS logic is traditionally used in most accelerator implementations due to clear design abstractions, but analog or neuronal circuit techniques may result in better performance or power. Precision, noise, and repeatability, and the ability to debug deserve consideration to ensure satisfactory functionality.

Micro Air Vehicle design is an exciting research area with many potential applications. For computer architects, it introduces research challenges in the design of power-performance efficient architectures to meet the performance requirements under a tight power budget. In this design space, we propose a hardware accelerator-based architecture to improve the power-performance efficiency of MAV workloads.

References

[BYNL09] G. Barrows, T. Young, C. Neely, and A. Leonard. Minimalist vision sensors for micro air vehicles. In *AIRP 2009 Workshop*, 2009.

[LK81] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, 1981.

[RB] Robobee project website. http://robobees.seas.harvard.edu/.

[Woo07] R. Wood. Liftoff of a 60mg flapping-wing mav. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.