“Dark silicon” is a growing problem: there isn’t enough power for increasing logic. Tightening power budgets call for more efficient, less power. Hardware accelerators achieve better efficiency and have shown 100-500x performance improvements compared to general-purpose cores.

- **Research challenges in the many-accelerator architecture**
  - Need many more accelerator designs
  - Few working accelerators are available to the academic community and rarely support common design patterns
  - Need many-accelerator architectures
    - Current architectures do not scale to many-accelerator systems
  - Need more interconnect designs
    - Accelerator based SoCs currently rely on shared bus schemes. We need more scalable interconnects, like NoC designs used by homogeneous multi-core architectures
  - Need accelerator benchmarks and applications
    - Benchmarks and compilers should be designed for and mapped to future accelerated capabilities

---

### Acceleration architecture issues

Accelerators can only be used for a specific operation (video, compression, security)

The accelerators supporting a wide set of workloads will be turned on; most will be off

Accelerators use a lot of memory, which is a waste if the accelerators aren’t in use:

<table>
<thead>
<tr>
<th>Accelerator</th>
<th>Function</th>
<th>Memory use</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>Data encryption</td>
<td>28%</td>
</tr>
<tr>
<td>JPEG</td>
<td>Image compression</td>
<td>35%</td>
</tr>
<tr>
<td>FFT</td>
<td>Signal processing</td>
<td>68%</td>
</tr>
<tr>
<td>Virtex</td>
<td>Convolutional coding</td>
<td>78%</td>
</tr>
<tr>
<td>Ethernet</td>
<td>Networking</td>
<td>79%</td>
</tr>
<tr>
<td>DDR-1</td>
<td>Decapilation bus</td>
<td>67%</td>
</tr>
<tr>
<td>L2 cache</td>
<td>Graphics</td>
<td>65%</td>
</tr>
<tr>
<td>Real subunit</td>
<td>Block coding</td>
<td>84%</td>
</tr>
<tr>
<td>DCT/SVD decoder</td>
<td>Sub-coding</td>
<td>89%</td>
</tr>
<tr>
<td>CAN</td>
<td>Automotive</td>
<td>70%</td>
</tr>
<tr>
<td>DVB FEC encoder</td>
<td>Video error correction</td>
<td>81%</td>
</tr>
</tbody>
</table>

DMA requires complex pre-programmed schedules to get data in and out of accelerators

### The accelerator store

The accelerator store (AS) gives accelerators a new way to store and exchange data

The AS contains memory to store accelerator state, which other accelerators can access

A general purpose (GP) CPU is only needed to map memory to accelerators

Data transfers are controlled by accelerators rather than by pre-programmed schedules

---

### Accelerator store features

- **Reduced memory buffers**
  - Accelerators directly access data in the AS without large DMA buffers

- **Memory reuse**
  - Any accelerator can use memory in the AS, not just one

- **Simple accelerator I/O**
  - The AS provides the FIFO interface to accelerators; exchanging data is just push and get

- **AS handles**
  - Each handle represents a block of memory in the AS
  - Each handle has a matching HID number
  - Accelerators specify a HID when accessing AS memory
  - Handles support random access (read/write) and FIFO (get/put)

### AS memory savings

- **Reduced GP-CPU energy**
  - Accelerators specify data transfers, so GP-CPUs can sleep more

- **Reduced memory energy**
  - The AS aggressively detects and turns off unused memory

---

### Optical flow accelerator

Optical flow used by micro air vehicle to estimate direction and distance

“Hardware in the loop” OpenGL-based virtual reality to evaluate effect of real-world optics

Goals for next year:

- Design space exploration of optics & optical flow implementations
- Develop MAV PID controller

---

[2] R. Hummel et al., "Understanding Sources of Inefficiency in General-Purpose Chips" in ISCA 2010

---

**Fine-grained Accelerator Architectures for Power, Performance, and Portability (Task 5.2.2)**

**StreamIt: complex accelerator workloads**

Filter nodes (accelerators)
- Each filter inputs some data, does an operation, and outputs data

Links (handles)
- Represents data moving between filters. Can be implemented as memory (DMA) or FIFOs (AS).

**Algorithmic design space projection**

Yakun Sophia Shao & Simone Campanoni

Algorithmic-level projection of accelerator cost/benefits

ILDJIT (run-time compilation framework) facilities code analysis and parallelism extraction

Goals for next year:

- Early-stage estimates of accelerator design space
- Integration of accelerators and parallel general purpose cores
- Come see ILDJIT tutorial at MICRO in Atlanta this December

---

**Optical flow accelerator**

Judson L. Porter III

- Optical flow used by micro air vehicle to estimate direction and distance
- "Hardware in the loop" OpenGL-based virtual reality to evaluate effect of real-world optics

Goals for next year:

- Design space exploration of optics & optical flow implementations
- Develop MAV PID controller

---

**Fine-grained Accelerator Architectures for Power, Performance, and Portability (Task 5.2.2)**

**StreamIt: complex accelerator workloads**

Filter nodes (accelerators)
- Each filter inputs some data, does an operation, and outputs data

Links (handles)
- Represents data moving between filters. Can be implemented as memory (DMA) or FIFOs (AS).

**Algorithmic design space projection**

Yakun Sophia Shao & Simone Campanoni

Algorithmic-level projection of accelerator cost/benefits

ILDJIT (run-time compilation framework) facilities code analysis and parallelism extraction

Goals for next year:

- Early-stage estimates of accelerator design space
- Integration of accelerators and parallel general purpose cores
- Come see ILDJIT tutorial at MICRO in Atlanta this December

---

**Optical flow accelerator**

Judson L. Porter III

- Optical flow used by micro air vehicle to estimate direction and distance
- "Hardware in the loop" OpenGL-based virtual reality to evaluate effect of real-world optics

Goals for next year:

- Design space exploration of optics & optical flow implementations
- Develop MAV PID controller

---

**Fine-grained Accelerator Architectures for Power, Performance, and Portability (Task 5.2.2)**

**StreamIt: complex accelerator workloads**

Filter nodes (accelerators)
- Each filter inputs some data, does an operation, and outputs data

Links (handles)
- Represents data moving between filters. Can be implemented as memory (DMA) or FIFOs (AS).

**Algorithmic design space projection**

Yakun Sophia Shao & Simone Campanoni

Algorithmic-level projection of accelerator cost/benefits

ILDJIT (run-time compilation framework) facilities code analysis and parallelism extraction

Goals for next year:

- Early-stage estimates of accelerator design space
- Integration of accelerators and parallel general purpose cores
- Come see ILDJIT tutorial at MICRO in Atlanta this December

---

**Optical flow accelerator**

Judson L. Porter III

- Optical flow used by micro air vehicle to estimate direction and distance
- "Hardware in the loop" OpenGL-based virtual reality to evaluate effect of real-world optics

Goals for next year:

- Design space exploration of optics & optical flow implementations
- Develop MAV PID controller