Improve Memory Access for Achieving Both Performance and Energy Efficiencies on Heterogeneous Systems

HONGYUAN DING, MIAOQING HUANG
Department of Computer Science and Computer Engineering
University of Arkansas

{hyding,mqhuang}@uark.edu

FPT'14, SHANGHAI
December 10^{th}-12^{th}, 2014

Outline

1. **Introduction**
2. **Accelerating the SIFT Algorithm**
   - Dedicated Hardware Accelerators
   - Distributed Multiprocessor System
3. **Experiments and Results**
4. **Conclusions**

Outline

1. **Introduction**
2. **Accelerating the SIFT Algorithm**
   - Dedicated Hardware Accelerators
   - Distributed Multiprocessor System
3. **Experiments and Results**
4. **Conclusions**

Introduction

- Limited power budgets for embedded system.
- Increasing demands for high-performance computing.
    - Increase memory bandwidth.
    - Optimize memory hierarchy.
    - Parallel computing with multiple processors.
    - Dedicated hardware accelerators.

Introduction

- Limited power budgets for embedded system.
- Increasing demands for high-performance computing.
  - Increase memory bandwidth.
  - Optimize memory hierarchy.
  - Parallel computing with multiple processors.
  - Dedicated hardware accelerators.
- In this work:
  - We examine how heterogeneous computation units affects both performance and energy efficiency.
  - We examine how memory access methods affect both performance and energy efficiency.

Outline

1. Introduction
2. Accelerating the SIFT Algorithm
   - Dedicated Hardware Accelerators
   - Distributed Multiprocessor System
3. Experiments and Results
4. Conclusions

Analysis of SIFT Algorithms

- Scale-invariant feature transform (SIFT) as a case study.
  - An algorithm in computer vision to detect and describe local features in images.

Analysis of SIFT Algorithms

- Scale-invariant feature transform (SIFT) as a case study.
  - An algorithm in computer vision to detect and describe local features in images.

Stage	Execution Time	# of Function Calls
Down sample	0.61%	–
Up sample	0.18%	–
Convolution	**37.49%**	**72**
DoG	0.44%	–
Find & refine key	0.38%	–
Octave gradient	**19.34%**	**29,944**
Key description generation	**41.56%**	**34,873**
Total	100%	–

*The size of the test image: 4,288×2,848.

Analysis of SIFT Algorithms (Cont.)

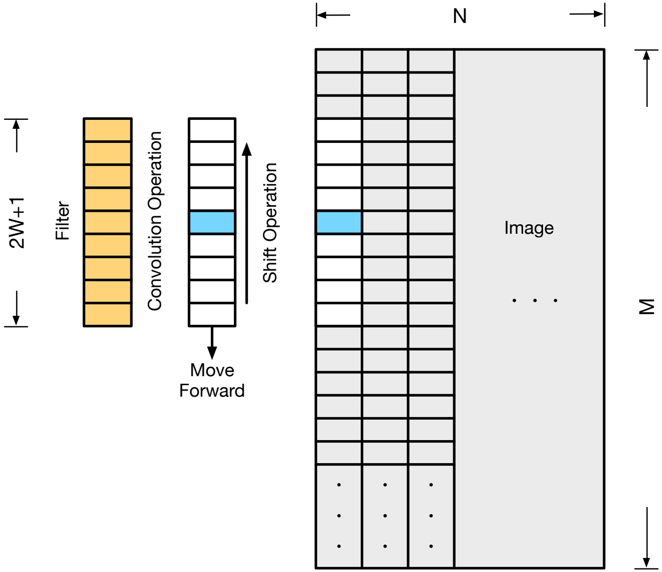
- Convolution:
  - Large amount of data processing in one iteration.
  - Consecutive memory addresses for input data.
- Octave Gradient & Key Description Generation:
  - Multiple function calls with relatively small data processing inside each one.
  - Inconsecutive memory access inside each iteration.

Analysis of SIFT Algorithms (Cont.)

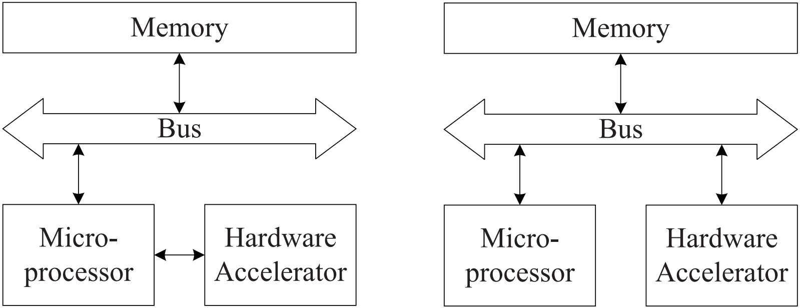
- Convolution:
  - Large amount of data processing in one iteration.
  - Consecutive memory addresses for input data.
- Octave Gradient & Key Description Generation:
  - Multiple function calls with relatively small data processing inside each one.
  - Inconsecutive memory access inside each iteration.
- Proposed Solutions:
  - Dedicated hardware accelerators.
  - Distributed multiprocessor system.

Outline

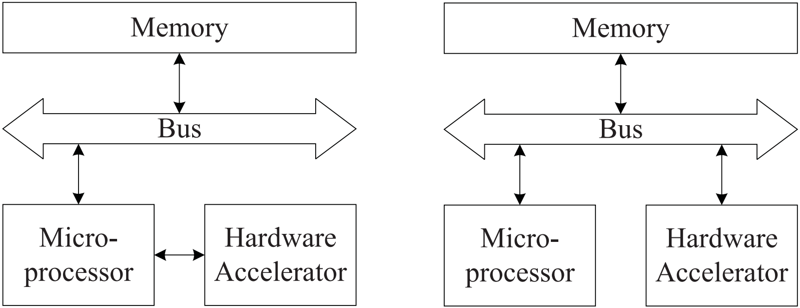
Principles of 1D Convolution



Data Flows

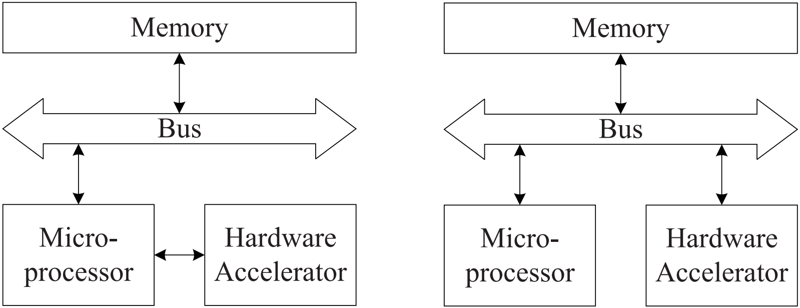

Memory
Bus
Micro-processor
Hardware Accelerator
Memory
Bus
Micro-processor
Hardware Accelerator

Data Flows



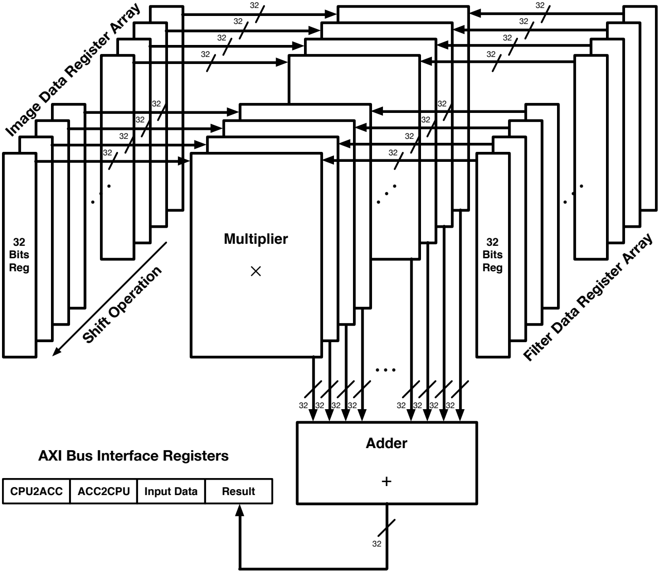
- Hardware accelerators access memory indirectly.

Data Flows

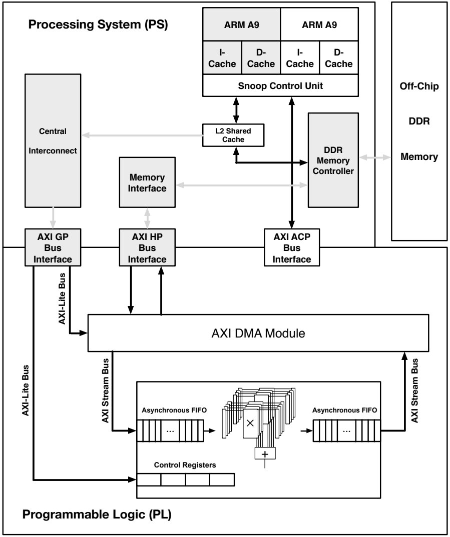


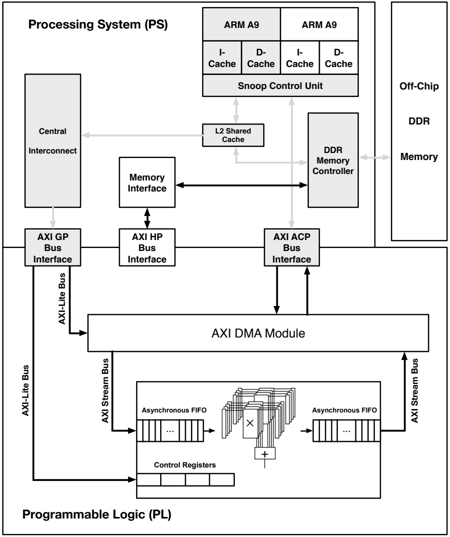
- Hardware accelerators access memory indirectly.
- Hardware Accelerators access memory directly.
  - Through AXI high-performance (HP) bus.
  - Through AXI accelerator coherency port (ACP) interface.

Convolution Accelerator: Indirectly Access



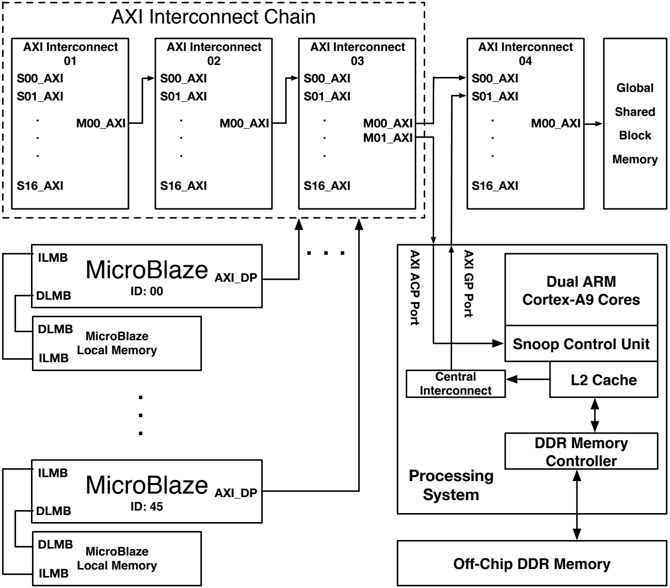
System Architectures: Directly Access





Outline

System Architecture



Pseudocodes of Host

```
//Host:
int main () {
  const int numSlaves = TotalMicroBlazes;
  writeGlobalData();
  //WRITE global data to shared block memory
  DCacheFlush();
  for (i = 0; i < numSlaves; i++) {
    globalStatus[i] = SignalStart;
  }
  DCacheFlush();
  for (i = 0; i < numSlaves; i++) {
    while (globalStatus[i]) != SignalFinish)
      wait();
  }
  return 0;
}
```

- Pass global shared data to slaves.
- Trigger slaves.
- Wait for them to stop.

Pseudocodes of Slaves

```
//Slaves:
int main () {
  const int ID = MicroBlazeID;
  const int numSlaves = TotalMicroBlazes;
  while (true) {
    while (globalStatus[ID] != SingalStart)
      wait();
    readGlobalData();
    //READ global data from shared block memory
    for (j = ID; j < numThreads; j += numSlaves) {
      threadExecution(j);
    }
    globalStatus[ID] = SignalFinish;
  } //Continue to next program
  return 0;
}
```

- Wait for host to trigger.
- Retrieve data from global shared memory.
- Traverse every threads.
- Notify host to stop.

Outline

Experiment Setup

- Hardware Configuration.
  - Zynq: ZC706.
  - ARM: 667 MHz.
  - Off-chip memory: 533 MHz.
  - Dedicated hardware accelerator: 50 MHz.
  - Multiple PEs: 200 MHz.

Experiment Setup

- Hardware Configuration.
  - Zynq: ZC706.
  - ARM: 667 MHz.
  - Off-chip memory: 533 MHz.
  - Dedicated hardware accelerator: 50 MHz.
  - Multiple PEs: 200 MHz.
- Acceleration Options.
  - Acc-1: Acc is connected through GP AXI ports.
  - Acc-2.1: Acc is connected through HP AXI ports.
  - Acc-2.2: Acc is connected through ACP AXI ports.
  - Acc-3: Acc is performance as multiple PEs.

Resource Utilization

Resource Types	D-Cache Disabled	D-Cache Enabled			
	w/ Acc-1	w/ Acc-1	w/ Acc-2.1	w/ Acc-2.2	w/ Acc-3
Registers	7,466 (1.7%)	7,466 (1.7%)	11,058 (2.5%)		137,783 (31.3%)
LUTs	11,648 (5.3%)	11,648 (5.3%)	14,635 (6.7%)		148,044 (67.7%)
DSPs	137 (15.2%)	137 (15.2%)	137 (15.2%)		413 (45.9%)
BRAMs	0 (0%)	0 (0%)	5 (0.9%)		432 (79.3%)

Resource Utilization

Resource Types	D-Cache Disabled	D-Cache Enabled			
	w/ Acc-1	w/ Acc-1	w/ Acc-2.1	w/ Acc-2.2	w/ Acc-3
Registers	7,466 (1.7%)	7,466 (1.7%)	11,058 (2.5%)		137,783 (31.3%)
LUTs	11,648 (5.3%)	11,648 (5.3%)	14,635 (6.7%)		148,044 (67.7%)
DSPs	137 (15.2%)	137 (15.2%)	137 (15.2%)		413 (45.9%)
BRAMs	0 (0%)	0 (0%)	5 (0.9%)		432 (79.3%)

- Overheads of DMA modules and FIFOs between Acc-1 and Acc-2.
- Up to 46 high-performance MicroBlazes with corresponding AXI interconnections.

Performance of SIFT implementation (unit: *s*)

Stage	D-Cache Disabled		D-Cache Enabled				
	w/o Acc	w/ Acc-1	w/o Acc	w/ Acc-1	w/ Acc-2.1	w/ Acc-2.2	w/ Acc-3
1	36.56	35.96	1.66	1.66	1.67	1.66	1.66
2	10.98	10.78	0.15	0.17	0.15	0.16	0.15
3	**2,248.98**	**436.22**	**121.42**	**217.34**	**19.52**	**12.57**	**12.61**
4	26.34	26.56	3.49	3.51	3.52	3.51	3.52
5	22.80	22.83	5.43	5.39	5.42	5.44	5.43
6	**1,148.30**	**1,150.21**	**70.92**	**70.87**	**70.90**	**70.88**	**58.42**
7	**2,492.48**	**2,490.65**	**175.51**	**175.50**	**175.47**	**175.48**	**142.65**
Total	5,997.30	4,180.26	380.36	476.01	280.78	272.94	227.58

Performances of SIFT implementation (unit: *s*)

Stage	D-Cache Disabled		D-Cache Enabled				
	w/o Acc	w/ Acc-1	w/o Acc	w/ Acc-1	w/ Acc-2.1	w/ Acc-2.2	w/ Acc-3
3	**2,248.98**	**436.22**	**121.42**	**217.34**	**19.52**	**12.57**	**12.61**
6	**1,148.30**	**1,150.21**	**70.92**	**70.87**	**70.90**	**70.88**	**58.42**
7	**2,492.48**	**2,490.65**	**175.51**	**175.50**	**175.47**	**175.48**	**142.65**
Total	5,997.30	4,180.26	380.36	476.01	280.78	272.94	227.58

Performances of SIFT implementation (unit: *s*)

Stage	D-Cache Disabled		D-Cache Enabled				
	w/o Acc	w/ Acc-1	w/o Acc	w/ Acc-1	w/ Acc-2.1	w/ Acc-2.2	w/ Acc-3
3	**2,248.98**	**436.22**	**121.42**	**217.34**	**19.52**	**12.57**	**12.61**
6	**1,148.30**	**1,150.21**	**70.92**	**70.87**	**70.90**	**70.88**	**58.42**
7	**2,492.48**	**2,490.65**	**175.51**	**175.50**	**175.47**	**175.48**	**142.65**
Total	5,997.30	4,180.26	380.36	476.01	280.78	272.94	227.58

- Enabling D-Cache brings benefits for both software and hardware implementation.
    - 2 times improvement for Acc-1.

Performances of SIFT implementation (unit: *s*)

Stage	D-Cache Disabled		D-Cache Enabled				
	w/o Acc	w/ Acc-1	w/o Acc	w/ Acc-1	w/ Acc-2.1	w/ Acc-2.2	w/ Acc-3
3	**2,248.98**	**436.22**	**121.42**	**217.34**	**19.52**	**12.57**	**12.61**
6	**1,148.30**	**1,150.21**	**70.92**	**70.87**	**70.90**	**70.88**	**58.42**
7	**2,492.48**	**2,490.65**	**175.51**	**175.50**	**175.47**	**175.48**	**142.65**
Total	5,997.30	4,180.26	380.36	476.01	280.78	272.94	227.58

- Enabling D-Cache brings benefits for both software and hardware implementation.
  - 2 times improvement for Acc-1.
- Accelerator coherence port (ACP) connected to L2 cache controller. 10$\times$ speedup.

Performances of SIFT implementation (unit: *s*)

Stage	D-Cache Disabled		D-Cache Enabled				
	w/o Acc	w/ Acc-1	w/o Acc	w/ Acc-1	w/ Acc-2.1	w/ Acc-2.2	w/ Acc-3
3	**2,248.98**	**436.22**	**121.42**	**217.34**	**19.52**	**12.57**	**12.61**
6	**1,148.30**	**1,150.21**	**70.92**	**70.87**	**70.90**	**70.88**	**58.42**
7	**2,492.48**	**2,490.65**	**175.51**	**175.50**	**175.47**	**175.48**	**142.65**
Total	5,997.30	4,180.26	380.36	476.01	280.78	272.94	227.58

- Enabling D-Cache brings benefits for both software and hardware implementation.
  - 2 times improvement for Acc-1.
- Accelerator coherence port (ACP) connected to L2 cache controller. $10\times$ speedup.
- Multiple PEs performs better than ARM core.
  - Advanced micro-architecture of ARM.
  - Low frequency for power concerns.
  - Limited memory bandwidth for multiprocessor system.

Power and Energy Analysis for Convolution

	D-Cache Disabled		D-Cache Enabled			
	w/o Acc	w/ Acc-1	w/o Acc	w/ Acc-1	w/ Acc-2.1	w/ Acc-2.2
CPU Active (W)	0.50					
Logic Static (W)	0.125					
Logic Active (W)	–	0.157	–	0.157	0.164	0.164
Logic Idle (W)	–				0.139	0.139
Energy (J)	1,405.61	286.60	75.89	142.79	12.55	8.10

Power and Energy Analysis for Convolution

	D-Cache Disabled		D-Cache Enabled			
	w/o Acc	w/ Acc-1	w/o Acc	w/ Acc-1	w/ Acc-2.1	w/ Acc-2.2
CPU Active (W)	0.50					
Logic Static (W)	0.125					
Logic Active (W)	–	0.157	–	0.157	0.164	0.164
Logic Idle (W)	–				0.139	0.139
Energy (J)	1,405.61	286.60	75.89	142.79	12.55	8.10

- Direct access to memory through HP AXI port and ACP AXI port can reduce the energy consumption by $5.97\times$ and $9.23\times$.

Power and Energy Analysis for Convolution

	D-Cache Disabled		D-Cache Enabled			
	w/o Acc	w/ Acc-1	w/o Acc	w/ Acc-1	w/ Acc-2.1	w/ Acc-2.2
CPU Active (W)	0.50					
Logic Static (W)	0.125					
Logic Active (W)	–	0.157	–	0.157	0.164	0.164
Logic Idle (W)	–				0.139	0.139
Energy (J)	1,405.61	286.60	75.89	142.79	12.55	8.10

- Direct access to memory through HP AXI port and ACP AXI port can reduce the energy consumption by 5.97$\times$ and 9.23$\times$.
- D-Cache is more vital for reducing power consumption in software implementation than hardware implementation.

Power and Energy Analysis for SIFT Algorithm

	w/o Acc Disable DC	w/o Acc Enable DC	w/ Acc
CPU Active (W)	0.50		
Logic Static (W)	0.125		
Logic Active (W)	–	–	0.525
Logic Idle (W)	–		0.313
Energy (J)	3,748.31	237.73	224.87

- With Acc means combining both Acc-2 and Acc-3 together to performance the whole SIFT algorithm.

Power and Energy Analysis for SIFT Algorithm

	w/o Acc Disable DC	w/o Acc Enable DC	w/ Acc
CPU Active (W)	0.50		
Logic Static (W)	0.125		
Logic Active (W)	–	–	0.525
Logic Idle (W)	–		0.313
Energy (J)	3,748.31	237.73	224.87

- With Acc means combining both Acc-2 and Acc-3 together to performance the whole SIFT algorithm.
- Accelerators is comparable with ARM enabling D-Cache.

Power and Energy Analysis for SIFT Algorithm

	w/o Acc Disable DC	w/o Acc Enable DC	w/ Acc
CPU Active (W)	0.50		
Logic Static (W)	0.125		
Logic Active (W)	–	–	0.525
Logic Idle (W)	–		0.313
Energy (J)	3,748.31	237.73	224.87

- With Acc means combining both Acc-2 and Acc-3 together to performance the whole SIFT algorithm.
- Accelerators is comparable with ARM enabling D-Cache.
  - ASIC vs. FPGA

Outline

Conclusions

- Accelerating SIFT algorithms:
  - Dedicated hardware acceleration for convolution with various memory access methods.
  - Distributed multiprocessor system to parallelize the last two stages of SIFT algorithms.

Conclusions

- Accelerating SIFT algorithms:
  - Dedicated hardware acceleration for convolution with various memory access methods.
  - Distributed multiprocessor system to parallelize the last two stages of SIFT algorithms.

- Performance and power analysis:
  - Conducting experiments on Zynq devices.
  - Steaming data flows with AXI HP and AXI ACP.
  - D-Cache is more important for software than hardware implementation in terms of performance and power consumption.
  - Multiprocessor system performs better than ARM with cache enabled.

Questions?

Thanks for listenning.