Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

Comparing FPGAs, GPUs and the PS2 using a unified source description

L. W. Howes, P. Price, O. Mencer, O. Beckmann, O. Pell

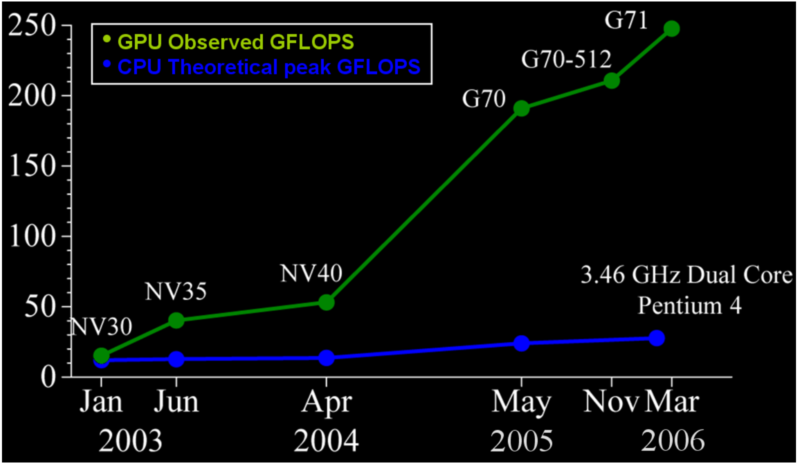
Department of Computing, Imperial College London

August 28, 2006

Motivation: Graphics Processing Units - the future?

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions



Thanks to Mark Harris of NVIDIA for this graph

Motivation: Comparing Accelerators

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

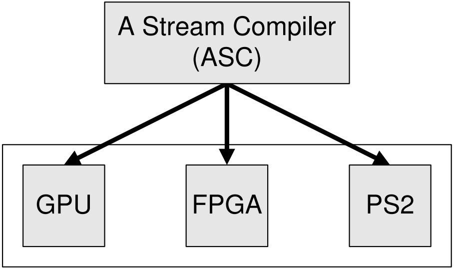
- Different characteristics
  - Applications
  - Accelerators
- As a result, accelerators match some applications better than others
- Wish to learn which accelerator is best
  - Experiment fairly
  - A single representation

Motivation: Development

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

- Heterogeneous architectures
- A variety of programming methodologies
- Even high level languages require low level knowledge
- Development becomes slow and expensive
- Use a single source description

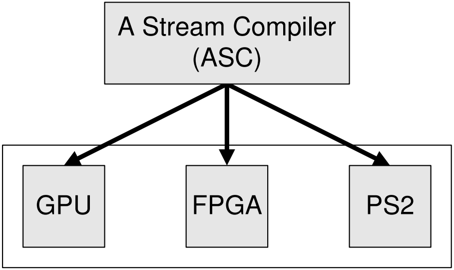


Motivation: Single Source Benefits

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

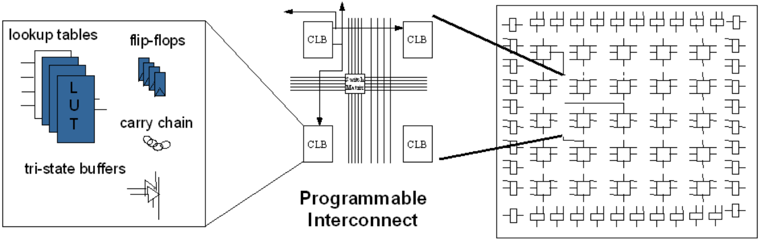
Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

- Fair comparison of performance on different architectures
  - May need architecture specific optimisations
- Easier development for multiple architectures
  - Could use architecture specific optimisations
  - Allow integration of multiple accelerators into a project - sharing the performance gain



Target: FPGAs

- Flexible
- Highly parallel
- Generally considered to be very difficult to program

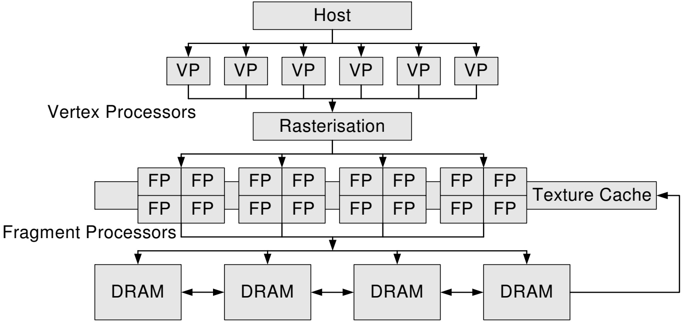


Target: GPUs

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

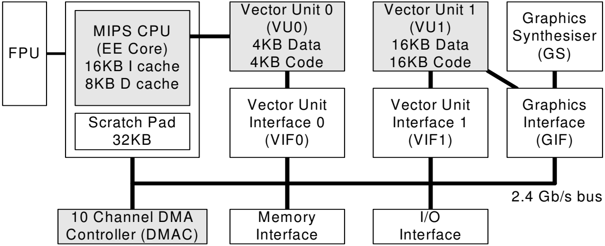
Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

- Highly parallel
- Widespread and used to accelerate graphics processing, largely for games
- Relatively low cost
- Recently being investigated for general purpose computation



Target: PS2

- Core MIPS processor
- Programmable vector units with local memory
- Large install base
- The real benefit: A step towards Cell



Related Work

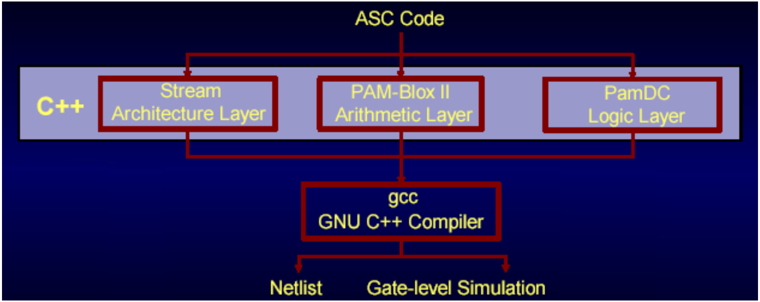
L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

- McCool et. al.;SIGGRAPH 2002
  Shader Metaprogramming
- Cope et. al.; FPT 2005
  Have GPUs made FPGAs redudant in the field of Video Processing?
- Cornwall et. al.; IPDPS 2006
  Automatically Translating a General Purpose C++ Image Processing Library for GPUs
- Trancoso et. al.; DSD 2005
  Exploring Graphics Processor Performance for General Purpose Applications
- Pavan Tumati; Undergraduate Thesis, Univ. Illinois
  Sony Playstation-2 VPU: A Study on the Feasibility of Utilizing Gaming Vector Hardware for Scientific Computing

A Stream Compiler - ASC

- Generates stream architectures for FPGAs
- C++ object oriented approach to development
- Combines algorithm, architecture and arithmetic levels into a single tool

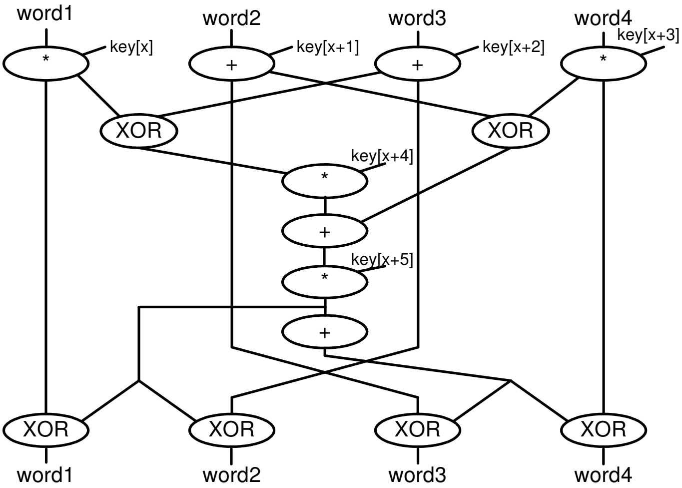


ASC Compilation

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

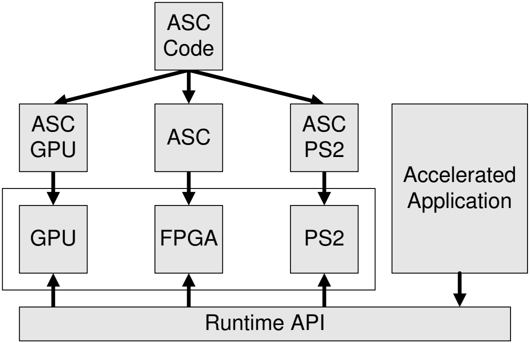
Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

- Map a data-flow graph directly to hardware
- High throughput, low clock frequency



ASC for other architectures

- ASC code represents the data flow of a program
- The ASC data flow can be implemented for various architectures

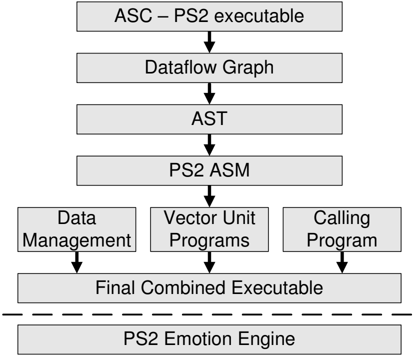


Targeting the PS2

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

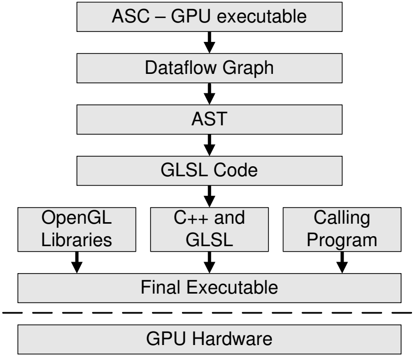
Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

- PS2 vector units take entire data flow
- Input data is split into blocks
- Data is fed to vector units to process each block in turn
- Makes use of operations on vector registers
- Can use both vector units to improve parallelism



Targeting the GPU

- Split data flow at various points and divide into computation kernels separated by intermediate arrays
  - Split at points of data reuse
  - Split where kernel complexity would be high
- Uses the OpenGL Shader Language to program the GPU



Example: ASC code targeting the GPU

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

Example

```
STREAM_START;

HWfloat input(IN);
HWfloat temporary(TMP);
HWfloat intermediate(TMP);
HWfloat output(OUT);

STREAM_LOOP(40);

temporary = input + prev(input,2);
intermediate = temporary + prev(temporary,2);
output = input + prev(intermediate,3)
        + prev(temporary,4);

STREAM_END_GLSL;
```

Example: GLSL output for previous example

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

Example

```
string ks__temporary10 =
  "void main(uniform samplerRect in__inputvar1) \n"
  "{\n"
  "  vec4 _temporary10;\n"
  "  vec4 in__inputvar1_var_P0;\n"
  "  in__inputvar1_var_P0 =
textureRect( in__inputvar1, vec2(gl_TexCoord[0].s, 0)).rgba;\n"
  "  vec4 in__inputvar1_var_P2;\n"
  "  in__inputvar1_var_P2.r = in__inputvar1_var_P0.b;\n"
  "  in__inputvar1_var_P2.g = in__inputvar1_var_P0.a;\n"
  "  in__inputvar1_var_P2.b =
    textureRect( in__inputvar1, vec2(gl_TexCoord[0].s + 1, 0)).r;\n"
  "  in__inputvar1_var_P2.a =
    textureRect( in__inputvar1, vec2(gl_TexCoord[0].s + 1, 0)).g;\n"
  "  _temporary10.rgba = (
    in__inputvar1_var_P2.rgba + in__inputvar1_var_P0.rgba );\n"
  "  gl_FragColor.rgba = _temporary10;\n"
  "} \n"; /* End kernel k__temporary10*/

...
    glslProgram.setProgram(ks_out__C5);
    glslProgram.setInputArray("in__A1", in__A1, TEXTURESIZEX, TEXTURESIZEY, 4);
    glslProgram.setInputArray("in__B3", in__B3, TEXTURESIZEX, TEXTURESIZEY, 4);
    glslProgram.setIteratorDimensions(TEXTURESIZEX, TEXTURESIZEY);
    float *outputsout__C5[1] = {out__C5};
    glslProgram.setOutputs(1, outputsout__C5, TEXTURESIZEX, TEXTURESIZEY, 4);
    glslProgram.run();
...
```

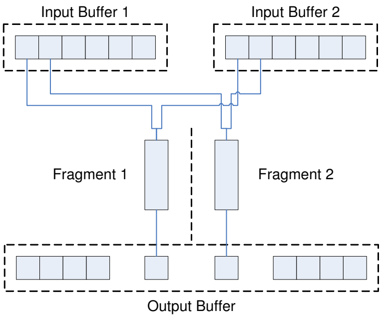
Limitations of the GPU

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

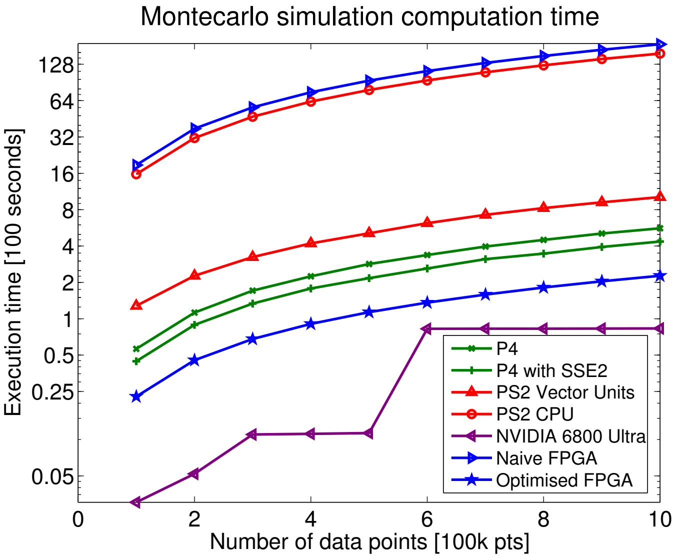
Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

- Each output value requires a separate kernel execution
- Fragment executions cannot communicate
- Feedback loops limited by the lack of communication
- Executions occur automatically in hardware
- The order of execution is left undefined

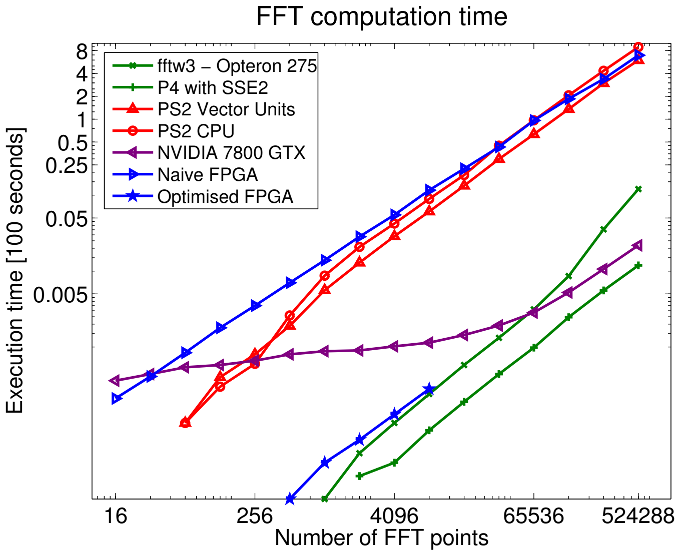
GPU Fragment Processing



Results: Montecarlo Simulation



Results: FFT


FFT computation time
fftw3 – Opteron 275
P4 with SSE2
PS2 Vector Units
PS2 CPU
NVIDIA 7800 GTX
Naive FPGA
Optimised FPGA
Execution time [100 seconds]
8
4
2
1
0.5
0.25
0.05
0.005
16
256
4096
65536
524288
Number of FFT points

Summary and conclusions

L. W. Howes
P. Price
O. Mencer
O. Beckmann
O. Pell

Motivation
The Future?
Accelerators
Benefits
Technology
Related Work
Implementation
ASC targets
Targeting the architectures
Example
Limitations
Results
Conclusions

- Multiple heterogenous acceleration architectures
- Experimenting can be difficult
- Use a single representation and try multiple targets
  - Compare the performance characteristics of the target architectures
  - Utilise multiple target architectures in achieving acceleration goals
  - Make best use of individual characteristics
- Early results, more optimisation effort needed

Questions?

Any questions?