Custom Computing for Video Codecs

Project Group WS 2012/13
Computer Engineering Group
Tobias Kenter, Jun. Prof. Christian Plessl
• video requires very high bandwidth
  – HD video uncompressed: 1920px * 1088px/frame * 3bytes/px * 30 frames/s = 1434 Mbit/s

• video can be efficiently compressed
  – exploit spatial correlation within & temporal correlation between frames
  – massive reduction in bandwidth (HD resolution)
    ▪ bandwidth of MPEG2: 80 Mbit/s (1:17)
    ▪ bandwidth of H.264: 20Mbit/s (1:71)

• challenges
  – compression is computationally very expensive
  – progress to higher image quality: ultra HD, 3D, larger color spaces, ...
  – broadcast: real-time streaming, multi-channel, low latency
  – special purpose equipment for certain domains
Video Compression: Basic Principle

encode only differences between images

frame i

frame i+1

absolute difference between two consecutive frames
Video Compression: Basic Principle (2)

improvement: estimate movement of image blocks

estimated motion vectors for blocks in the image

absolute difference between two consecutive frames without motion compensation

absolute difference between two consecutive frames with motion compensation

images: Yao Wang, Brooklyn Polytechnic
Example: Simplified MPEG2 Encoder

lossy image compression (like JPEG)

simulate reconstruction of image at receiver

predict current frame based on previous frames

current frame $\rightarrow$ prediction error $\rightarrow$ prediction error $\rightarrow$ DCT $\rightarrow$ Q $\rightarrow$ Q$^{-1}$ $\rightarrow$ DCT$^{-1}$ $\rightarrow$ RLC $\rightarrow$ reconstructed image

frame memory $\leftarrow$ previous frame $\leftarrow$ motion vector $\leftarrow$ motion estimation $\leftarrow$ motion compensation $\leftarrow$ loop filter $\leftarrow$ predicted frame $\leftarrow$ predicted frame

a $\rightarrow$ b $\rightarrow$ c $\rightarrow$ d $\rightarrow$ e $\rightarrow$ f
What is Custom Computing?

• custom computing is “computing without CPUs”
  – translate algorithms to application-specific “processors”
  – implementation with programmable hardware (FPGAs)
  – massively parallel computation
  – speedups for suitable applications: 10-1000x

• new generation of custom computers
  – specifically targeted at HPC applications
  – programming at a higher abstraction level
  – no circuit design skills required
  – reformulate algorithms as streaming data flow computation
public class Mav_kernel extends Kernel {
    public Mav_kernel (KernelParameters parameters) {
        super(parameters);

        HWVar x = io.input("A", hwFloat(8, 24));
        HWVar prev = stream.offset(x, -1);
        HWVar next = stream.offset(x, +1);
        HWVar sum = prev + x + next;
        HWVar result = sum / 3;
        io.output("B", result, hwFloat(8, 24));
    }
}
Using a Maxeler Custom Computer

software (C)

device = max_open_device
(maxfile, "/dev/max0");
float A[SIZE]
...
stream_data(device, A);

manager (Java)

Manager m = new Manager
("Loop", MAX3);
m.kernel(Mav_kernel,
link("A", PCIE);
link("B", DRAM(LINEAR));
m.build();

public class Mav_kernel extends Kernel{
    public Mav_kernel(KernelParameters
parameters) {
        super(parameters);
        HWVar x = io.input("A", hwFloat(8,24));
        HWVar prev = stream.offset(x, -1);
        HWVar next = stream.offset(x, 1);
        HWVar sum = prev + x + next;
        HWVar result = sum / 3;
        io.output("B", result, hwFloat(8,24));
    }
}
Project Group: Custom Computing for Video Codecs

• create the world's fastest custom computing-accelerated implementation of a state-of-the-art video codec

• tasks
  – understand video compression methods
  – analyze building blocks and bottlenecks of codecs
  – reformulate building blocks in streaming data flow representation
  – build performance estimation models for accelerated versions
  – implement accelerators on Maxeler data flow computer
  – show results with cool demo

• infrastructure and tools
  – access to custom computers (Maxeler MaxStation and MaxNode)
  – high-level design tools, no circuit design knowledge required
Organization and further information

- the project group is held in English
- website
  - http://homepages.uni-paderborn.de/plessl/teaching/2012-PG-Custom-Computing
- lecture “Reconfigurable Computing” by Prof. Marco Platzner
  - runs concurrently with the project group
  - background knowledge on custom computing technology and tools
- advisors
  - Tobias Kenter kenter@uni-paderborn.de
  - Jun.-Prof. Christian Plessl christian.plessl@uni-paderborn.de
- registration
  - please register via the PAUL