Custom Computing for Video Codecs

Project Group WS 2012/13
Computer Engineering Group
Tobias Kenter, Jun. Prof. Christian Plessl
Introduce yourselves

- who are you?
- why did you join this project group?
- what is your technical background?
  - programming in Java, C, C++
    - debugging
    - profiling
  - digital hardware design or FPGA knowledge
  - video codecs
  - project management, developing methods
  - version control, git
  - seminar experience
1. Overview of Topic

2. Organization of the project group

3. Discussion, open questions
Video Codecs

- video requires very high bandwidth
  - HD video uncompressed: $1920 \times 1088 \times 3 \times 30$ 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: 20 Mbit/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

Elgato turbo.264 HD USB video encoding accelerator
Video Compression: Basic Principle

code only differences between images

encode only differences between images

frame i

frame i+1

absolute difference between two consecutive frames

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

lossy image compression (like JPEG)

predict current frame based on previous frames
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));
    }
}

Maxeler Data Flow Computer

- integrated custom computing solution
  - high-level accelerator specification with Java library
  - FPGA internals and tools hidden from developer
  - suitable for streaming applications

ex: smoothing data with moving average
- b[i] = (a[i+1]+a[i]+a[i-1]) / 3
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();

kernel (Java)

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
1. Overview of Topic

2. Organization of the project group

3. Discussion, open questions
• the project group is held in English
• website
  – http://homepages.uni-paderborn.de/plessl/teaching/2012-PG-Custom-Computing
  – github repository to be set up
• 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
Your Project Group

• your **PROJECT** group
  – a big thing: 2 semesters, 30 credit points á 30 hours
    ▪ e.g. 3 days a week for 9 months
  – entire development process
    ▪ specification, implementation, testing, presentation, documentation
  – nice demo in the end

• **YOUR** project group
  – not a lecture or exercise: we support you, but
  – you need to organize yourself
  – you are responsible for defining and reaching goals
  – you need to acquire the needed knowledge

• your project **GROUP**
  – teamwork and organization
  – regular meetings and documentation of process
  – delegation and subdivision + collaboration and synchronization
  – get to know each other
• **first week: warm-up mini seminar (NOW)**
  – foundations of working techniques and video codecs
  – 10 minute presentations next week

• **five weeks: Maxeler Tutorials (to end November)**
  – provided Maxeler material
  – four Video Codec related examples
  – hardware performance modeling
  – test working techniques

• **six weeks: main seminar (to mid January)**
  – scientific papers
  – presentations and reports

• **in parallel: prototype project**
  – e.g. Motion JPEG

• **2 weeks: specification of the main project (to end January)**
  – specify the further goals and schedule of the project
Mini seminar

• topics part 1: working techniques
  – debugging with *gdb*, core dumps
  – profiling, *gprof* or others
  – agile development, extreme programming
  – version control with *git* and github

• topics part 2: video codec basics
  – spectral transformations, DCT
  – arithmetic coding
  – runlength coding, Huffman coding
  – color spaces, color subsampling

• prepare up to 10 minutes talk
  – slides + examples and demo where possible
  – textbook and online material (Wikipedia etc.) is fine this time
  – references to further material welcome
  – no written report required
• we need a one time time slot next week (up to 3 hours)
  – mini seminar talks
  – Tuesday 10am – 1pm or 3pm
  – Wednesday 11am – open end
• we need a weekly time slot (half day)
  – tutorial common sessions
  – Monday all day, Tuesday to Friday before 1pm
• you will work remotely on two Maxeler machines
  – Tutorial in the pool
  – what else do you need?
Some rules

• controlling
  – find out if things don’t work out -> risk plan in specification
  – if someone does not work: three steps
    ▪ talk with supervisor
    ▪ talk with the whole group
    ▪ exclusion from project group

• grading
  – results of whole project group
  – results of subgroup for specific task
  – individual commitment and performance

create the world's fastest custom computing-accelerated implementation of a state-of-the-art video codec
1. Overview of Topic

2. Organization of the project group

3. Discussion, open questions