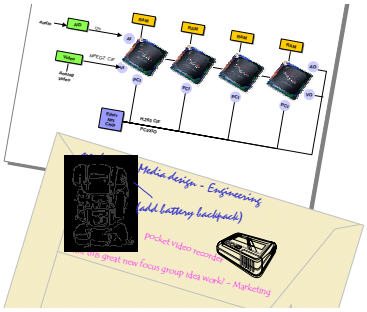
TriMedia Optimization: 4-day 'hands-on' class

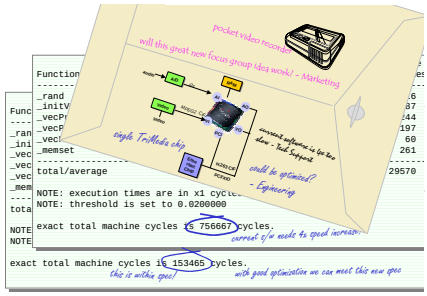
TriMedia Foundation series

Benefits

Optimizing software reduces its size and increases its speed: both result in lower hardware usage and so lower production cost. Optimizing can be the essential step in making product functionality fit into the available hardware resource.

Optimization is sometimes thought of as a 'black art' but insight and understanding make it straightforward. This advanced 'hands on' workshop explains optimization techniques and tips that make a real difference and that can be easily applied.





Optimizing: the alternative to bigger hardware

Contents

The class is a workshop with lots of 'hands-on' exercises. Delegates need to bring their own laptops but software can be provided if needed (most delegates will already have TriMedia software installed). The class progressively introduces optimization strategies and techniques, illustrating through the 'hands-on' exercises.

CPU and cache architecture

Relating the CPU and cache architectures to program behavior and optimization.

- CPU and cache architectures
- parallel scheduling

Profiling

How to investigate program behaviour and use this knowledge to guide choice of optimizations. Targetting different optimizations: registers, cache, parallelism.

- How to read and graph DTREE code
- DTREE code and inherent parallelism
- How to read and use schedule reports
- How to use profile reports
- Using the profiling API
- Cache, paralleism and custom ops

Optimization strategies

Why and how to choose an optimization strategy. Understanding levels of optimization.

- Increasing parallelism
- Reducing memory traffic
- Using the cache efficiently
- Using registers efficientl;y

Compiler optimizations

How and when to use compiler optimizations. Understading what these optimizations aim to do, and what are their effects. Using pragmas to tune optimization.

- Effect of optimization levels
- Unrolling, grafting and inlining
- Profile-driven optimization
- Tuning by independent linking

Increasing parallelism

How to increase parallelism..

- Investigating paralleism in your code
- Barriers to paralleism:
- Pointer(and array) aliasing
- Dirty floating point
- Memory latency

Reducing memory traffic

How to eliminate memory accesses.

- Avoiding implied memory loads
- Using packed memory data

Using cache efficiently

How to use the cache optimally.

- Keeping data in the cache
- Increaising temporal and spatial locality
- Organising data for the cache

Using registers efficiently

How to use the register file efficiently.

- Letting the compiler use registers
- Register-to-register operations
- Monitoring register usage

Custom operations

Why, when and how to use costom operations. How to progressively uncover possibilities for using custo operations.

- SIMD - FIR example
- Pack and merge - matrix transposition
- Median filter example
- Motion estimation example

Optimizing for other than speed

How to monitor and control code size when optimizing. Understnding the effect of optimizations on code size and power consumption.

- Optimizations that reduce code size
- Optimizations that increase code size
- When not to optimize
- Optimizing for power consumption

Time and arrangements

We will present this 4-day workshop 'on-site' by arrangement - the material can be adapted if you have specific needs (at extra cost). We recommend that delegates should already have attended our 3-day 'Introduction to TriMedia' class which gives a thorough grounding in TriMedia core, cache, optimization, peripherals and software architecture.

Class schedules are posted on the Internet from time to time:

- http://www.bores.com/schedule.htm

TriMedia optimization seminar

- 4-day 'hands on' workshop
- on-site by arrangement

Introduction to TriMedia

- 3-day class
- on-site by arrangement

To book or find out more

Call us by 'phone or send email to book or to ask questions.

- contact: Dr Chris Bore
- 'phone: +44 (0)1483 740138
- mobile: +44 (0)7785 268905
- email: chris@bores.com

TriMedia foundation series

The 'TriMedia foundation' is a series of classes designed to give a thorough understanding of all aspects of the TriMedia. TriMedia Foundation series currently includes the following classes:

- TriMedia Software Architecture
- TriMedia peripheral architecture
- TriMedia CPU architecture
- TriMedia cache architecture
- TriMedia optimization

BORES Signal Processing, 'Fordwater', Pond Road, Woking, Surrey GU22 0JZ, England:
'Phone: +44 (0)1483 740138 Web site: www.bores.com email: chris@bores.com