

# Computations on graphics processors

Ingemar Ragnemalm
Information Coding, ISY



# This lecture:

Plan for this part of the course

GPU evolution

GPU architecture

A first intro to general computing solutions with GPUs

# Course plan:

5 lectures

1 lesson

3 labs

Information Coding / Computer Graphics, ISY, LiTH

# Lectures:

1. GPU evolution and architecture

2. Intro to CUDA

3. CUDA memory, threads, synchonization

4. More CUDA, computing with shaders

5. Intro to OpenCL

# Labs:

1. CUDA

2. More CUDA

3. OpenCL and shaders

No lab reports,
demonstrations in the lab

Information Coding / Computer Graphics, ISY, LiTH

# Literature for this part

Primary source:
CUDA on-line manual

Recommended extra:
CUDA by example (Sanders & Kandrot)

Hand-outs

Lecture material

# Questions

1. How can a GPU be much faster than a CPU?

2. Why is the G80 so much faster than the previous GPUs (e.g. 7000 series)?

3. A texturing unit provides access to texture memory. What more is it than just another memory?

4. Suggest two major differences in the Fermi architecture that will make a difference from the G80/G92/GT200


Information Coding / Computer Graphics, ISY, LiTH

# The decline of CPU evolution

Three ”walls”:

Power wall

Memory wall

ILP wall

# The decline of CPU evolution

Three ”walls”:

Power wall

Memory wall

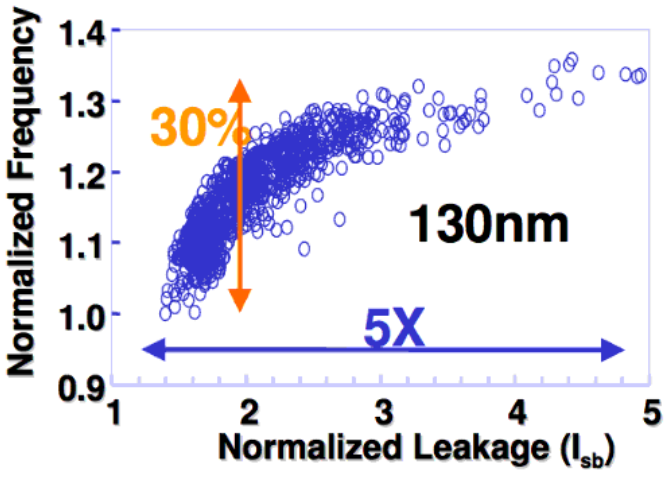
ILP wall

- Clock frequency can no longer go up
- The memory architecture is insufficient
- Attempts to parallelize have failed

Information Coding / Computer Graphics, ISY, LiTH

# Power wall

13% higher frequency = 73% more (almost double)
double power consumption!



# Power wall

Reverse reasoning: Lower frequency a little, win much power.

Replace one high-frequency CPU with two slightly slower - for the same cost!

Works nicely for two CPUs.

Intel promises 80 cores in a few years

BUT

this will run into the ”memory wall”

Information Coding / Computer Graphics, ISY, LiTH

# Memory wall

Already, the memory is slower than the CPU.

With more and more CPUs fighting for accessing the same RAM and caches, efficiency will degrade!

Memory bandwidth helps - if we can get it.

# ILP wall

Instruction level parallelism

Many parallel systems have been made in the past. They have all suffered from one problem: It takes an effort to program them. Programs must be rewritten to fit. The programs must be parallelized.

But one problem here has been *availability*. The machines were there, but you couldn't just sit down and try it out at any time! Awkward tools, few machines, unavailable experts, nobody could help.

Another problem in the past was that the standard CPUs caught up too quickly with parallel machines.