I do not have mystical clairvoyance, but I do have a VG set from an influential meeting that hasn't occurred yet…

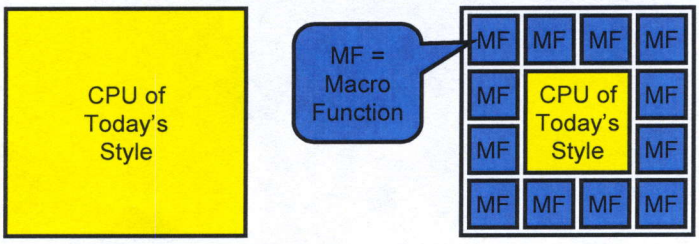
Emerging Research Logic Technologies

Traditional Goal

Logic technology that is scaleable beyond CMOS, high-speed, and low-power.

Macro Function Direction

- **Current CPU style**
- **New direction proposed to industry will be to keep CPU but augment it with "macro functions."**
- **Macro functions may include non-CMOS logic devices specialized to nontraditional functions, such as speech recognition, etc.**



Outline

- Degree of Innovation
- Non-Architecture Projections
- Architecture Projections
- Programming
- Architecture Summary
- Current Activities to Watch and Why
- Conclusions

Programmability Considerations

- Has code changed since the "late nCUBE" era?
  - MPI replaced proprietary message passing
  - We have a huge code base of math code (at Sandia)
  - We have frameworks (at Sandia)
- Conclusion
  - A lot of code written and put into reusable form, but little change in underlying programming method
- Implication
  - Further migration towards putting code into libraries, but the code will have the same basis

Programming

- Industry will integrate the following macro functions:
  - Graphics processors
  - Speech recognition
  - Visual recognition
- However, the hardware will be sufficiently general purpose to be used for supercomputing
- Still CMOS in this timeframe
- A small number of super-duper programming jocks will write supercomputing code for the macro functions
  - LAPACK
  - FEM meshing
  - Etc.
- Regular programmers will write C++/Fortran code interfacing like DirectX (Microsoft's GPU API)

Programming Example

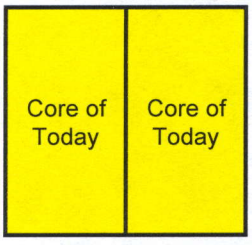
- I went by the PeakStream booth yesterday and see that they have a scientific programming library for graphics processors. I've never used it, but I think the approach might work with hardware up to 2020.

Outline

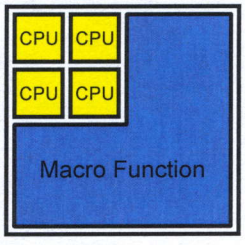
- Degree of Innovation
- Non-Architecture Projections
- Architecture Projections
- Programming
- Architecture Summary
- Current Activities to Watch and Why
- Conclusions

Processor Chip Prediction

- ¼ of chip to be four CPUs each with 10× throughput of today's cores
- ¾ of chip to be a new Macro Function
- Layered nano memory



- Macro Function will be developed by industry and repurposed for supercomputing, originally
  - speech recognition
  - vision for robots

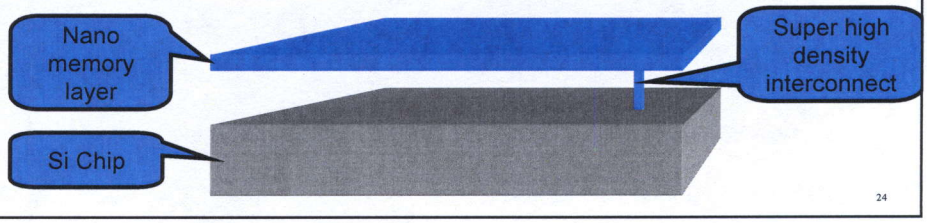


CPU Detail

- Entry
  - Four cores at 50 GF Linpack-peak each, total 200 GF
  - 36 macro functions of 440 GF each, total 15.8 TF total
    - graphics, speech, vision, repurposed to scientific kernels
  - 16 TF per chip
  - Each chip to have 1 GB+ layered nano memory
  - As much external memory as you like (not a limit)
  - 50,000 chips in a 2 MW system → 800 Petaflops

Memory Story – No Memory Wall

- I predict one of the 4+ nano memory options will succeed
- 1 GB+ memory will be integrated onto the CPU
  - I don't care if you call it cache, main memory, etc.
- Memory will be non-volatile
- This will boost CPU performance quite a bit over the 5× predicted by architecture study



Interconnect

- Interconnect is likely to be optics, but not necessarily fiber
  - Free space
  - Waveguides
- Luxtera comes up often in discussions of optical interconnect. The Luxtera approach works with Si by having external lasers.

Outline

- Degree of Innovation
- Non-Architecture Projections
- Architecture Projections
- Programming
- Architecture Summary
- Current Activities to Watch and Why
- Conclusions

Current Activities to Watch and Why

- Cyclops – highly multicore architecture that could (with suitable systems software) blend legacy code compatibility with efficient use of multiple cores
  - Memory hierarchy is where the action is
  - I predict future will hold Cyclops + layered memory
- Layered memory (Nantero?)
- Optical interconnect (Luxtera?)
- Programming (PeakStream?)

Outline

- Degree of Innovation
- Non-Architecture Projections
- Architecture
- Programming
- Architecture Summary
- Current Activities to Watch and Why
- Conclusions

Conclusion I

- Industry is now putting additional resources created by Moore's Law into more cores and is talking about the same for graphics chips and Macro Functions
- Coders are getting further away from programming the bare hardware
- My solution has the following properties:

Conclusion II

- The majority of users will program the conventional cores. They will see a fairly flat parallel Von Neumann computer. Of course, they are accustomed to using libraries for inner loops
- A small number of users will optimize low level code (libraries) for edge of the envelope hardware where the programmers need to be cognizant of data and operation placement
- I believe this is the most likely to happen, even if it does not make for the most exciting computer architecture research