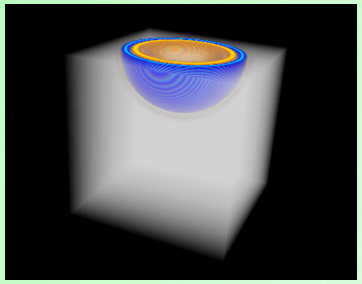

Building an SOC
How to do it?
What will it cost?

Marty Deneroff

Emu Solutions, Inc.
marty@deneroff.us

On face, it looks extremely attractive to build an HPC computer using SOC techniques and IP from consumer product space.

- Is this a good idea?
- What will it cost?
- What are the issues?
- How do we address them?

Good Idea?

The Consumer Market* is much larger (~10X or more) than the PC market.

- Requires very fast product cycles -> simplified design processes
- Enormous competitive pressures to keep pushing cost down and capability up
- Since volumes are huge, can amortize huge investments across volume
- Huge investments in off-the-shelf resalable IP

If we can use the consumer infrastructure, we can take advantage of these investments!

*smart phones, tablets, set-top boxes, game consoles, etc.

Cost?
NRE: $20,000,000 +/-

Assumptions:

- Current established (*Not Bleeding Edge!*) process
- Large (near reticle limit) die size
- Vendors understand what you are doing, trust your competence

- $5M NRE to Silicon Integrator (eSilicon, GUC, etc.)
  - Physical design
  - Package design
  - Test design
  - Mask & proto charges
- $5M for IP
- $2M for CAD tools
- $8M for engineering salaries and expenses
  - 20% architecture / logic design
  - 20% system software development
  - 30% Design Verification
  - 30% Floorplanning / placement / vendor engagement

Issues

Building an SOC for the HPC space has many issues to consider, and is (in some ways) different from an SOC for a consumer product

- Engineering costs more important than piece price.
- All desired IP may not be available.
- Vendors may not be motivated because of low volumes.

Can we now use IP from the Consumer space to do HPC, with better economics?

IP Requirements

- ✓ CPU Core – license ~ $500k
- ✓ Memory Controller – license ~500k
- ❑ (NOC – *yes, but may choose to design – license ~$800k)*
- ✓ External Interconnect (SERDES) – *yes, but availability can be problematic, especially if need multiple orientations – license ~ $2.5M*
- ✓ On-chip SRAM (compiler) – license ~ 500k
- ✓ Other IO Interfaces – license ~ $200k
- ✓ PCIexpress controller (not incl. SERDES) – license ~$800k

Designing an SOC

1. Co-develop Logical and Physical Architecture, incorporating available information about available process capabilities; try to approach (with margin for growth) reticle limit
2. Develop notional floor plan
3. Negotiate with and select vendors – important to lock down pricing and responsibilities early – clear specs are important!
4. Design top level floor plan and global wiring early, and force block ports to conform; define and freeze pinout
5. Select target clock rate conservatively, and be flexible about adjusting it as design progresses; Set target DV coverage and test coverage, planning to be strict about these.
6. Design and verify primary architectural blocks, including local timing closure
7. In parallel, perform full chip verification with behavioral models for blocks

Designing an SOC (2)

8. Work out detailed schedule with Silicon Integrator – make sure package design, test development, etc. are in parallel and fall within shadow of chip design
9. You probably want to develop and adopt a partial-good (spare block) strategy to enhance yield and lower costs. Work closely with Silicon Integrator on this.
10. Do physical design of major blocks as soon as each is designed – relies on early global wiring lock-down
11. Don't get hung up on hitting a particular clock rate – easy to lose many months in timing closure if you are unwilling to adjust
12. *DV coverage, signal integrity, and test coverage goals must be achieved – These are the things that cause most re-spins*

Example: Green Wave Chip Block Diagram

- 12 x 12 2D on-chip torus network
- 676 Compute cores (500 in compute clusters, 176 in peripheral clusters)
- 33 Supervisory cores
- 1 PCIexpress interface (16 bit Gen 3)
- 8 Hybrid Memory Cube (HMC) interfaces
- 1 Flash controller
- 1 1000BaseT ethernet controller

It is not anticipated that all cores will be utilized – some are spares for yield enhancement.