Interconnect and 3D Systems

NSF Workshop on Emerging Technologies for Interconnects—WETI

Trevor Mudge
Bredt Family Professor of CSE
Director ARM Research Center
The University of Michigan, Ann Arbor

February 2, 2012, Washington, DC
What are we putting on a chip these days?

- Intel’s “right turn” in 2005 accelerated the trend to multi-core
- 100’s of cores on a chip will soon be economically sensible
- Will they be heterogeneous?
- Depends on the market
  - Embedded will continue to produce Frankenchips
  - Convergence?
  - Server and cloud computing will continue to employ large numbers of identical cores in many-core architectures with ever increasing memory requirements
- How are we going to interconnect cores and memories?
  - Wires?
  - Optical?
  - What will be the characteristics?
NoC—Mesh connected

- One approach has been to employ local connections
- Create a tessellation of components through nearest neighbor interconnects—scalable
- NUMA/NUCA organization
- Highly non-deterministic memory accesses—QoS
- Shared memory model requires significant support for routing/coherence/consistency
- Energy inefficient—support for buffering etc.
- Unnecessarily restrictive for on-chip interconnect
- Today’s processes have up to ten layers of metal—32nm TSMC
- Higher radix switches will be possible in the future—this is the emerging trend for on-chip interconnect
Ideal Answer—Connect “everything to everything”

- Simplifies system level considerations—non-blocking
  - Relatively predictable behavior
  - Coherence and consistency are straightforward
- Doesn’t have to be a planar graph to make layout easy
- Constructing a bi-partite graph—quadratic complexity
- MUX-based solutions are challenging to layout

![Diagram of a network with multiplexers (mux) and demultiplexers (demux)]
Crossbar Interconnects On-Chip

- Connecting N inputs to N outputs
- Memories have been doing this for year
- Result is a crossbar-like
  - PRO—simple single stage routing and simple control
  - CON—quadratic growth
- Our designs† are actually swizzle switches networks—SSNs
  - permutations + multicast
  - 64 x 64 x 128 bit buses
  - Least recently granted priority—LRG
  - sub-Watt power consumption
  - 2mm x 2mm in 45nm—600 MHz
  - one clock set-up
  - one clock transfer
  - 32nm much smaller and faster—more metal layers

† A 4.5Tb/s 3.4Tb/s/W 64×64 switch fabric with self-updating least recently granted priority and quality of service arbitration in 45nm CMOS. Sudhir Satpathy, Korey Sewell, Thomas Manville, Yen-Po Chen, Ronald Dreslinski, Dennis Sylvester, Trevor Mudge, David Blaauw. Int Solid-State Circuits Conference 2012
Swizzle Functionality

- **Permutations** are 1-to-1 mappings of input to output ports.
- **Swizzles** are the generic form of duplication and permutation combined where the output does not need to contain all the inputs.
- **Broadcast** is a special form of swizzle where one of the values is broadcasted to all the outputs.
- **Multicasting** is another special form of swizzle where multiple input ports are duplicated in a regular pattern to the outputs.
On-Chip Many-Core Architectures

Total Area = 204 mm²
Scalability

- Ultimately this style of interconnect reaches a limit—$O(n^2)$
- Works for single chip—enough for a wide range of applications
- Provides a very high radix NoC
- Yet another one-shot improvement
- 3D stacking
- Large systems are possible—100’s of cores
Stacked Many-Core System

- Reduced interconnect—3GHz SSN now possible
- Opportunity to increase L2—layers are ¼ area
- Power per layer on the edge of manageable
- Naïve layout solution
Thank You
Future Opportunities—Many Cores

- 64 core single chip system
  - A5 based with 32kB icache + 32kB dcache in 32nm
  - 3 Swizzle switches—core/L2 L2/core core/core
  - 64x32 by 128 @ 1.5 GHz
  - Delay 1-2 clocks to SSN + 2 across +1-2 to L2
  - ~30 W less L2
Stacking SSNs

- Bit lines can run vertically