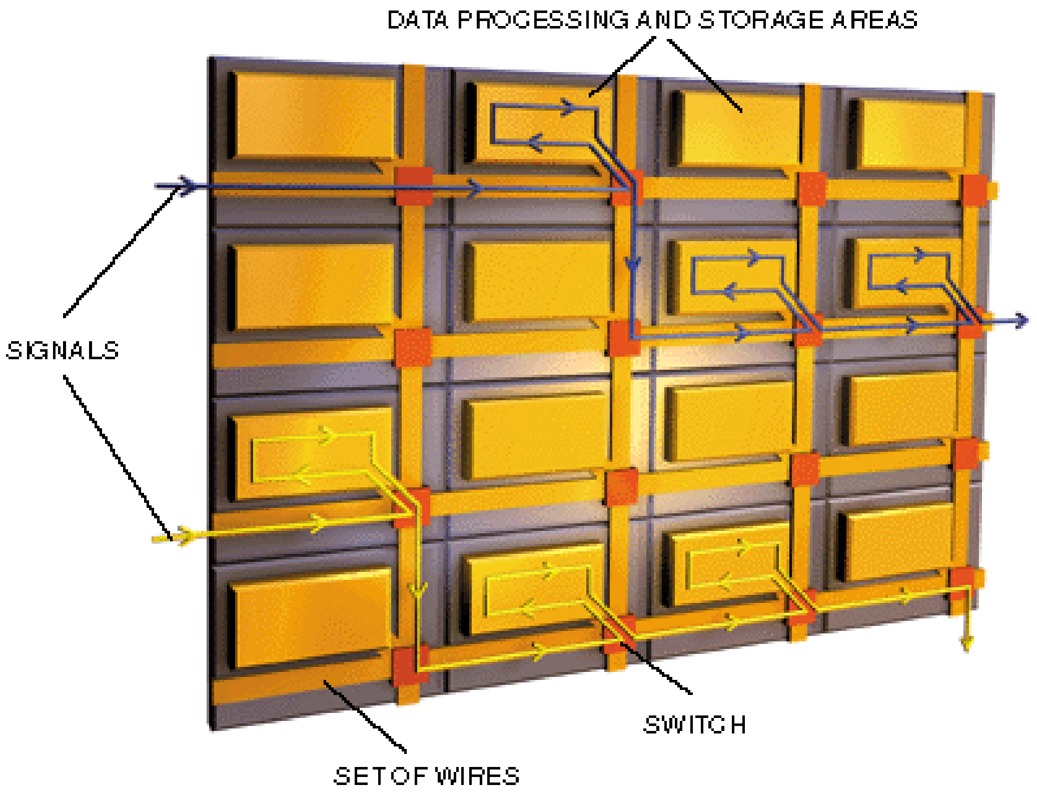
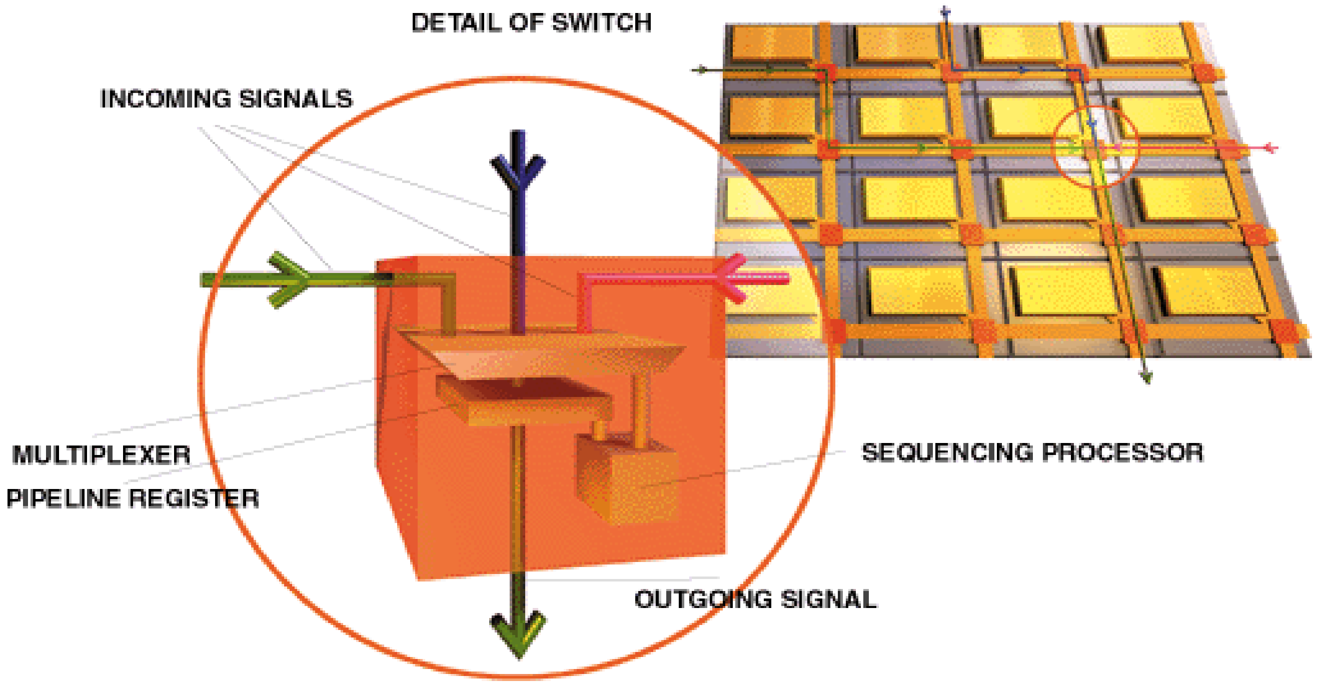
The Raw Architecture

- 4 x 4 mesh of tiles
- Each tile contains:
  - 8-stage MIPS-like 32 bit processor
  - Pipelined FPU
  - 32 KB Data and Instruction Caches
  - Interconnect routers
- 250 MHz Clock
- 16 FLOPs/cycle
- 4 GFLOPS Peak Computation Rate



Raw Static Interconnects



- Separate instruction stream - unlimited virtual channels
- Pipelined, register-mapped communication

Raw Static Interconnects cont'd

- Networks fully integrated into FU pipelines/bypasses

Comparison of Communication Paradigms

Typical Memory Mapped	Raw Register Mapped

`lw $1, a`
`lw $2, b`
`add $1, $1, $2`
`sw c, $1`
	`add $csto, $csti, $csti2`

- Networks extensible to very large fabrics
- FUs can do useful work every cycle - peak FLOP rate can be realized
- Communication faster than memory

Programming Raw Using Registers

- Map systolic array onto Raw tiles
- 90-100% utilization possible

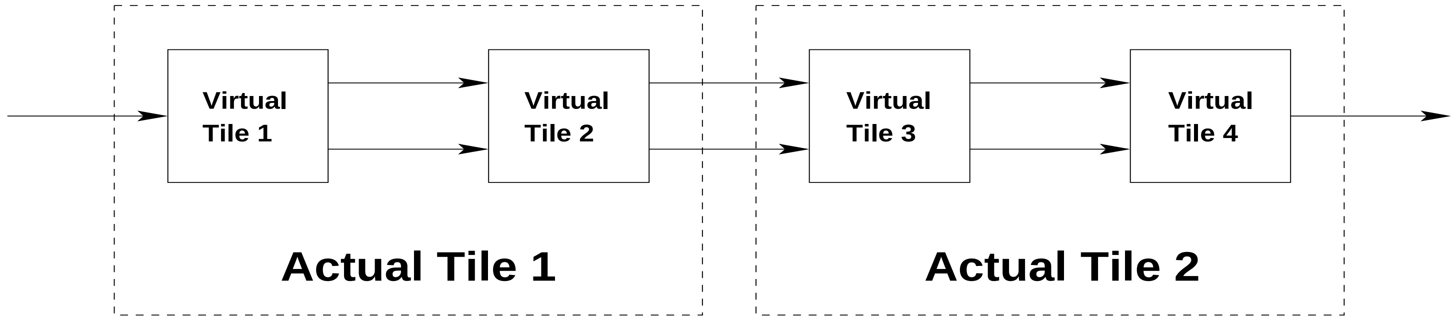
Register-bound Performance of DSP Kernels

Kernel	Performance: % of Peak	Performance: GFLOPS
16 tap FIR Filter	98	3.94
4x4 * 4xN Matrix Multiply	94	3.76
2 tap FIR Filter and 4 channel, 2 beam beamformer with corner turn	91	3.30

- Not all problems fit into registers

Programming Raw Using Data Cache

- Virtualization - map N virtual tiles to 1 actual tile



- Use local memory to buffer data

The Cost of Virtualization

- Requires local memory/control code
- Consider multiply and accumulate

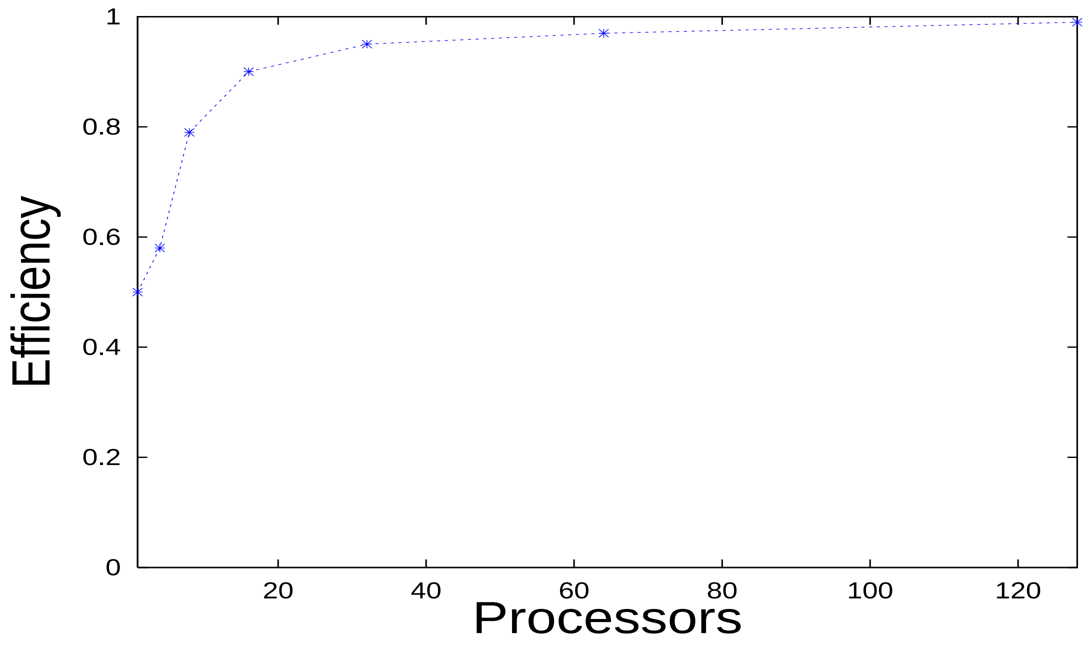
Virtualized Multiply and Accumulate

Data Type	FLOPs	Memory Ops	Total Ops	Max Efficiency
Real	2	4	6	33%
Complex	8	8	16	50%

- Virtualization too expensive
- How to get efficiency of registers on large problem sizes?

Stream Algorithms

- Decouple problem into memory and computation
- Computation proceeds in systolic manner
- Use more tiles for computation than memory



- More tiles, more efficiency

Steam Algorithms cont'd

- Designed for Virtual Architecture
  - Tiled, 2D Mesh
  - Each tile has memory, processing, and communication
- Decouple problem into memory and processing tasks
  - Assume problem size N (could be a function)
  - Allocate P tiles to computation (could be a function)
  - Allocate M tiles to memory/control (could be a function)
  - Efficiency Condition: $M = o(P)$, M is strictly less than P
  - Predicted Efficiency: $\frac{P}{P + M}$

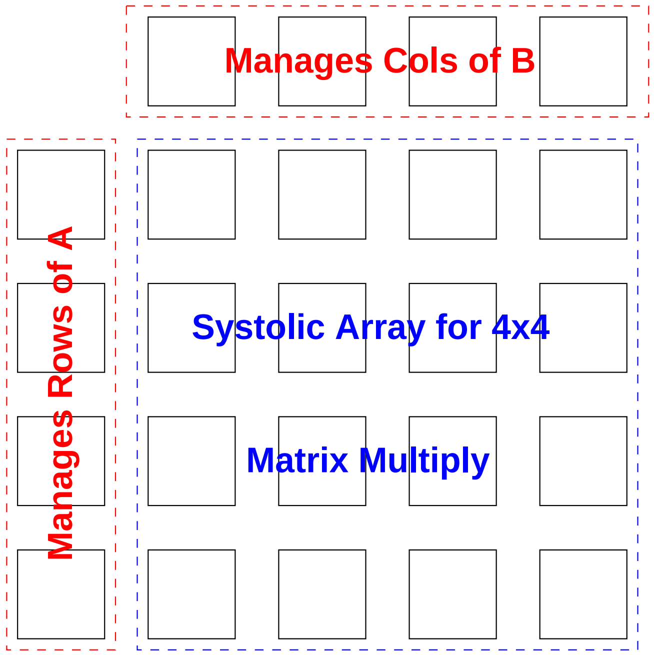
Stream Design Methodology

- Systolic Partitioning
  - Divide into subproblems
  - Solve subproblems with systolic array
- Decoupling
  - Use memory tiles to buffer intermediate results
  - Memory tiles can perform reductions
- Check Efficiency Condition

Matrix Multiply Example: C = AB

- Assume A, B are N x N, N^3 work
- Recursive formulation: series of inner products
- Use $P = T^2$ tiles for systolic array (compute tiles)
  - Each tile performs N^3/T^2 inner products
  - Each tile is fully utilized
- Use M = 2T tiles for memory/control
  - T tiles manage N/T rows of A
  - T tiles manage N/T columns of B
- Efficiency: $T = o(T^2)$

Matrix Multiply Example cont'd

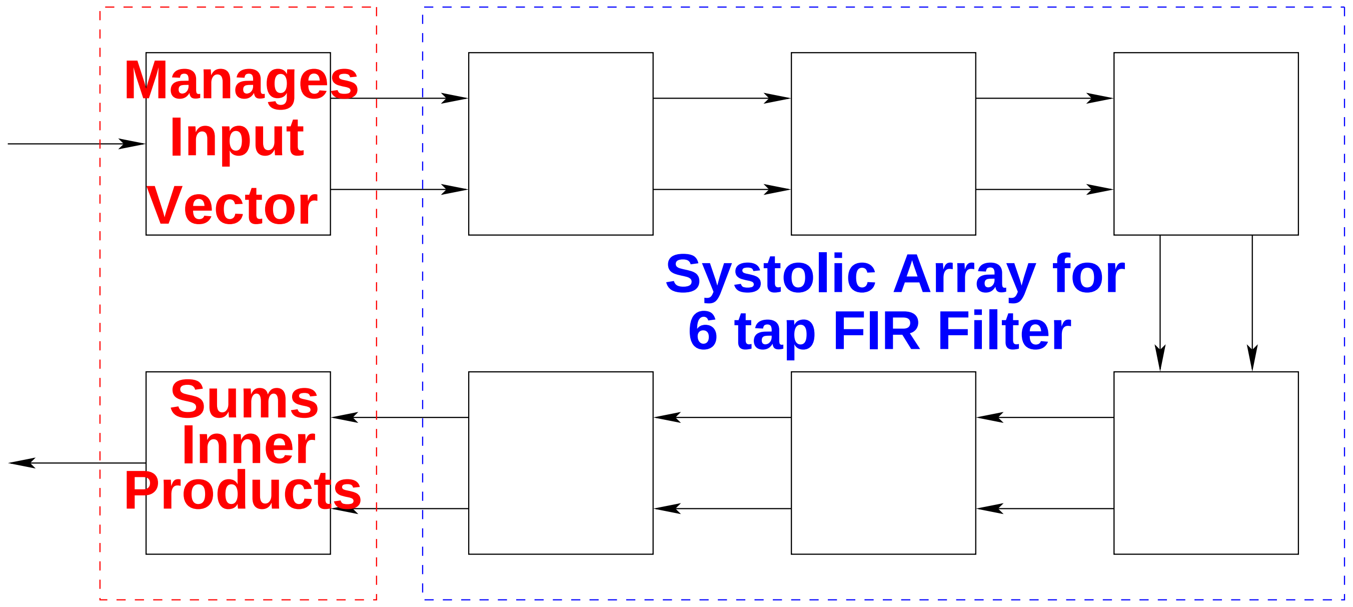


Implementation	Efficiency for Real	Efficiency for Complex
Virtualized Systolic Array	33%	50%
Stream Algorithm, T = 4	66%	66%
Stream Algorithm, T = 18	90%	90%

FIR Example

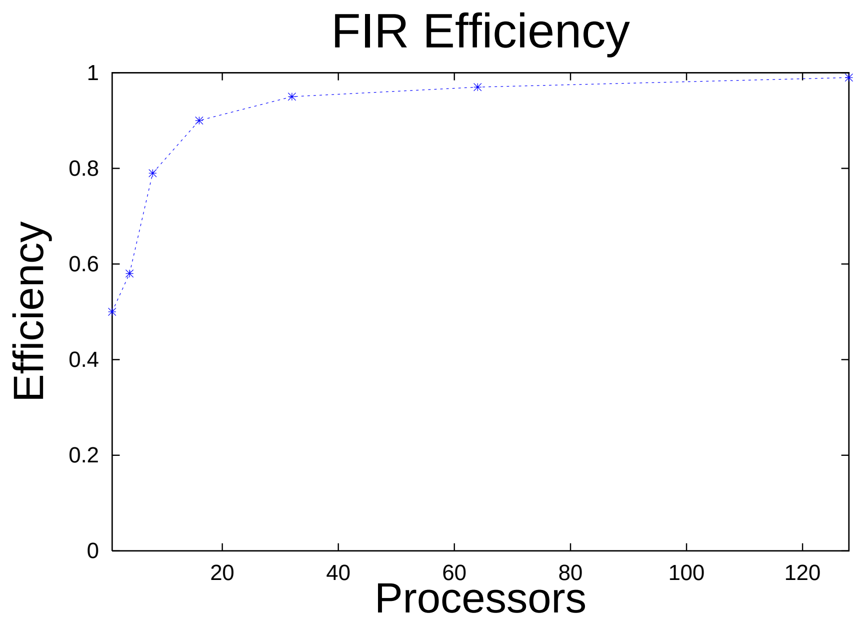
- Assume K taps and N elements, KN work
- Recursive formulation: sum of inner products
- Use P = T tiles for systolic array (compute tiles)
  - Each tile performs KN/T multiply and accumulates
  - Each tile is fully utilized
- Use M = 2 tiles for memory/control
  - 1 tile manages the input vector
  - 1 tile computes the sums of the inner products (33% utilized)
- Efficiency: $2 = o(T)$

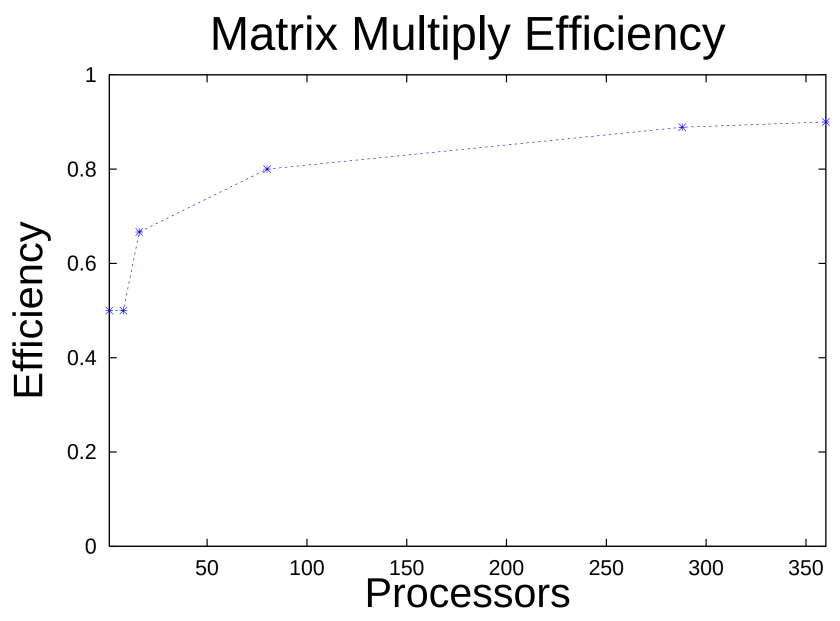
FIR Example cont'd



Implementation	Efficiency for Real	Efficiency for Complex
Virtualized Systolic Array	33%	50%
Stream Algorithm, T = 8	66%	79%
Stream Algorithm, T = 16	83%	90%

Efficiency Curves for Examples





Problems We Have Tackled:

- Inner Product
- Matrix Multiply
- FIR Filter
- DFT
- Triagonal Solver
- LU Decomposition

Conclusion

- Efficiency increases with increasing resources
- Raw + Stream Algorithms well suited for DSP
- Future Work:
    - Extension to other architectures
    - Apply to QR, SVD, Shortest-Pat