Trends toward Spatial Computing Architectures

Dr. André DeHon
BRASS Project
University of California at Berkeley

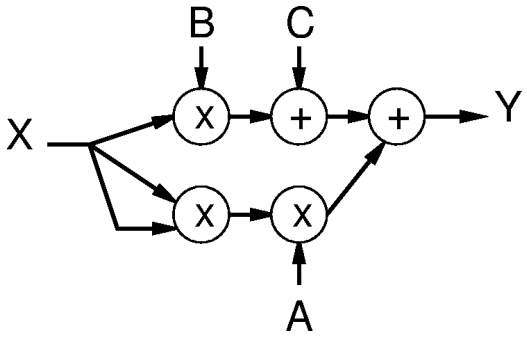
➨ How do we build *programmable* VLSI computing devices in the era of $G\lambda^2 \rightarrow T\lambda^2$ silicon die capacity? (billion transistors)

- Capacity available 1000→100,000×
- Opens up architectural space
- Spatial architectures become viable and beneficial

Back to Basics

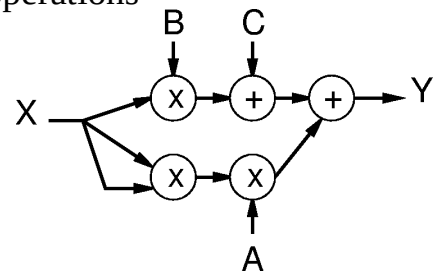
- What is a computation?

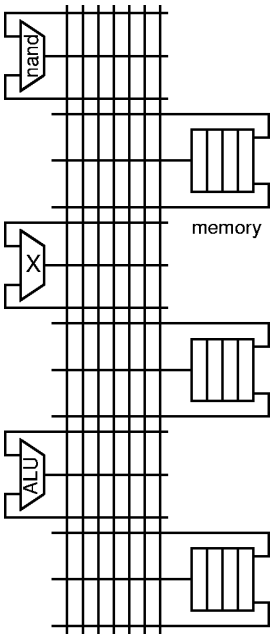
$Y=Ax^2+bx+c$



Basics

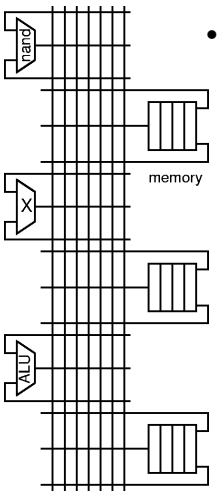
- How do we implement a computation?
  - Perform operations
  - Communicate among operations





Implement Computation

- Perform operations
  - universal computational modules
    - nand, ALU, Lookup-Table
  - specialized operators
    - multiple, add, FP-divide



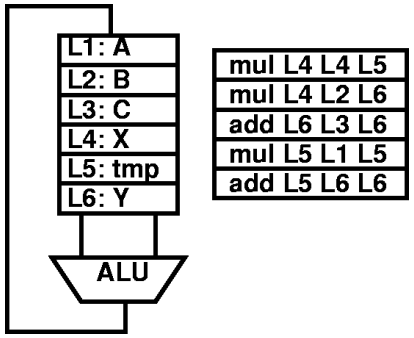
- Communicate among operations
  - spatially
    - network
  - temporally
    - memory

Implementation

- Choice in implementation:
  - How many compute elements?
  - How much sequentialization?

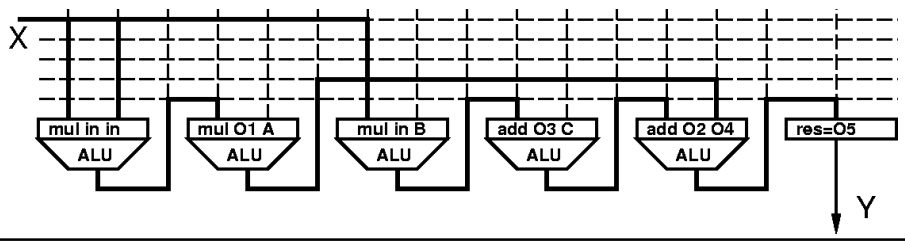
Serial Implementation

- Single Operator
- Reuse in time
- Store instructions
- Store intermediates
- Communication across time
- One cycle per operation



Spatial Implementation

- One operator for every operation
- Instruction per operator
- Communication in space
- Computation in single cycle



Some Numbers

- Binary Operator w/ Interconnect 500K→1Mλ^2
  - (*e.g.* ALU bit, LUT (gate), …)
- Instruction (include interconnect) 80Kλ^2
- Memory bit (SRAM) 1→2Kλ^2

◊ Fully Sequential: N×80Kλ^2 + S×1Kλ^2 +1Mλ^2

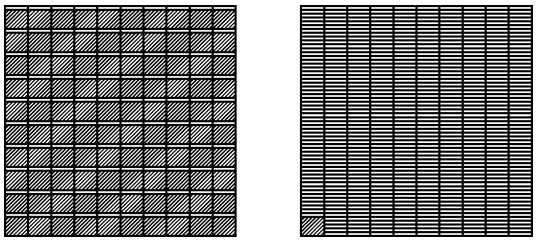
◊ Fully Spatial: N×1Mλ^2

➜ Temporal N slower, 12× smaller

Programmable Device: 50Mλ^2

Sample die: 7mm×7mm, 2.0μm

- Spatial: 50-100 bit operators
  - 2 32b addrs?, small bit-serial datapath?
- Sequential: 600+ instructions (data)
  - kernel on chip?



Programmable Device: 100Gλ^2

16mm×16mm, 0.1μm

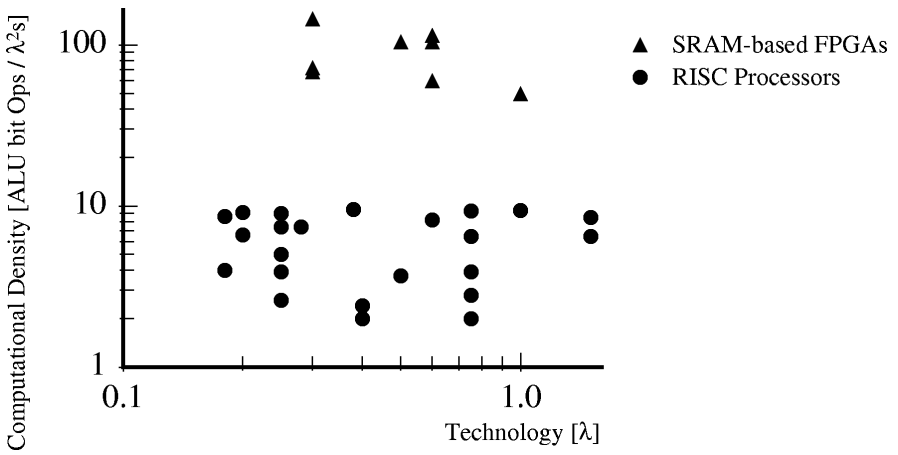
- Spatial: 100,000 bit operators
  - even bit parallel, can support kernels with 1000s of operators
- Sequential: 1.2M instructions (data)
  - entire applications (and data?) fit on chip

Density Advantage

Why implement spatially?

- For these extremes, spatial has:
  - 50-100× operators/cycle 50Mλ^2
  - 100,000× operators/cycle 100Gλ^2
- Conventional word architectures
  - 32b →2-3× 50Mλ^2
  - 4×64b →400× 100Gλ^2

Empirical
Raw Density Comparison



Spatial Advantages

- 10× raw density advantage over processors
- potential for fine-grained (bit-level) control — can offer another order of magnitude benefit versus SIMD/word architectures.
- Demonstrated on select applications
- With 1000's of operators per chip today:
  - substantial problems fit spatially on die.

Spatial Drawbacks

- Lower instruction density
  - 12× bit controlled extremes
  - 12×32≈400× where SIMD-word ops apply
- Unused (infrequently used) operators waste space when not in use

Example: FIR Filtering

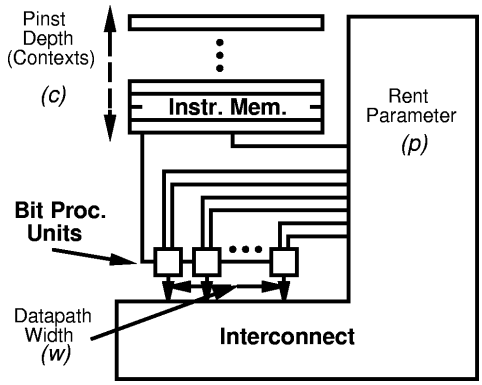
Architecture	Reference	Feature Size (λ)	16b TAPs λ^2s	8b TAPs λ^2s
16b DSP	ISSCC87	0.65μm	0.057	0.057
32b RISC/DSP	VLSI SP95	0.25μm	0.021	0.021
64b RISC	ISSCC96	0.18μm	0.064	0.064
Pentium MMX	ISSCC97	0.18μm	0.019	0.019
FPGA	XC4K	0.60μm	0.57	1.9
	Altera 8K	0.30μm		3.6
Reconfigurable	PADDI2	0.50μm	0.39	0.93
	MATRIX	0.25μm	0.22	0.86
Full Custom	JSSC89	0.75μm		3.6
	JSSC90	0.60μm		3.5
	CICC90	0.75μm	0.68	0.68
(fixed coefficient)	CICC92	0.60μm	56	56

Architecture Space

- Broad space between sequential and spatial extremes
  - 1⟶ to ⟶ 100,000 operators
  - Microprocessors: 4×64=256
- Navigate space to design most efficient architectures

Computing Device

- Composition
  - Bit Processing elements
  - Interconnect
    - space
    - time
  - Instruction Memory



Compute Model

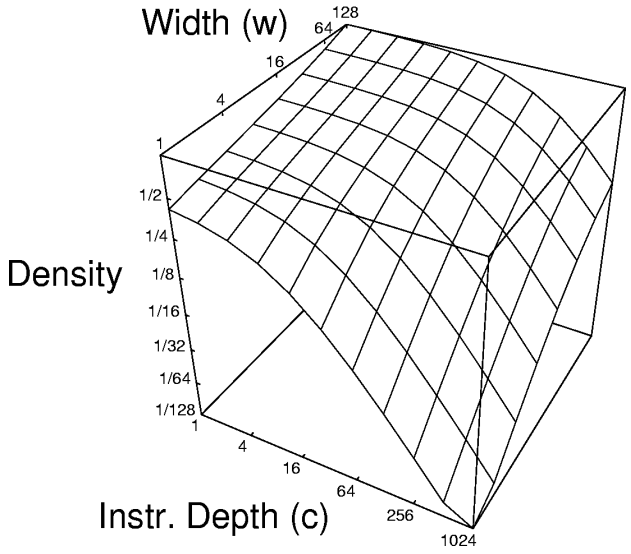
- Use model to estimate area implied by architectural parameters

$$A_{bitop}=A_{op}+A_{instr}(c,w)+ A_{interconnect}(p,w,N)+A_{data}(d)$$

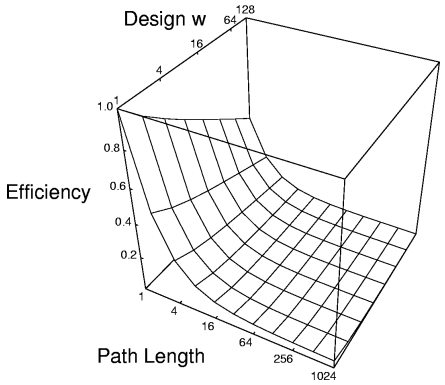
- Use areas to compare density and efficiency

$$\text{Efficiency} = \frac{\text{Area(best matched architecture)}}{\text{Area(evaluation architecture)}}$$

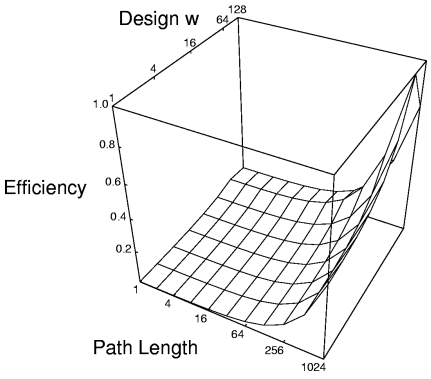
Peak Densities from Model



Processors and FPGAs



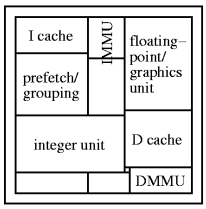
FPGA
c=d=1, w=1, k=4



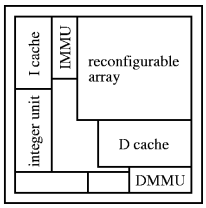
"Processor"
c=d=1024, w=64, k=2

Hybrids: Processor+Array

- Example: UCB GARP
  - MIPS-II Core
  - array⇔memory access
  - on-chip config. cache
  - 1500 4-LUTs



(a) UltraSPARC.



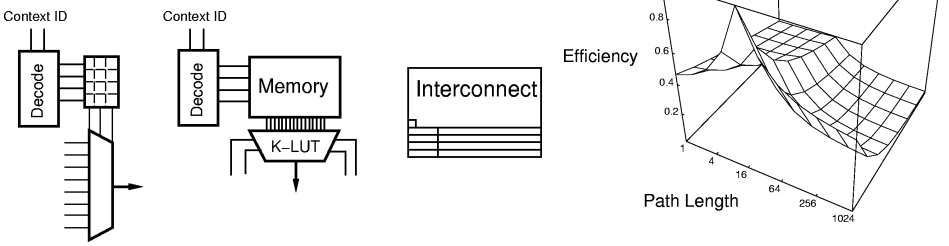
(b) Hypothetical Garp.

Benchmark	167 MHz SPARC	133 MHz Garp	ratio
DES encrypt of 1 MB	3.60 s	0.15 s	24
Dither of 640 × 480 image	160 ms	17 ms	9.4
Sort of 1 million records	1.44 s	0.67 s	2.1

- Also: PRISC, NAPA, OneChip, Chimera, ...

Hybrids: Intermediates

- *E.g.* Multicontext FPGA: MIT DPGA
  - on-chip space for a few instructions
  - single cycle instruction switch



Conclusions

- Growth in silicon capacity makes spatial implementations viable
- Spatial implementations offer density (performance) advantage
- As silicon capacity grows
  - more problems “fit” spatially
- Richer architectural space available today
  - worth rethinking how we build programmable computing systems