Comparing Computing Machines

Dr. André DeHon
UC Berkeley

November 3, 1998

Talk

- Confusion (difficulties)
- Comparisons
- Caveats
- Characteristic Caricatures

Confusion

- **Proponents**:
  - $10\times \rightarrow 100\times$ benefit
- **Opponents**:
  - $10\times$ slower
  - $10\times$ larger

- …and $\exists$ examples where both are right.

Difficulty

- When we hear raw claims:
  - X is faster 10× faster than Y

- We know to be careful
  - How old is X compared to Y?
    - Know technology advances steadily
    - Even in same architecture family
      - 5 years can be 10×
  - How big/expensive is X compared to Y?
    - X have 10× resources of Y?

Clearing up Confusion

- How do we sort it all out?
  - Step 1: implement computation each way
  - Step 2: assess the results
  - Step 3: generalize lessons
- This talk about step 2:
  - much difficulty lies here

Common Fallacies

- Comparing across technology generations without normalizing for technology differences
- Comparing widely different capacities
  - single chip versus board full of components
- Comparing
  - clock rate
  - or clock cycles
  - but not the total execution time (product)

Common Commodity

- Convert costs to a common, technology independent commodity
  - total normalized silicon area
- As an IC/system-on-a-chip architect
  - die area is the primary commodity

Technology (Area)

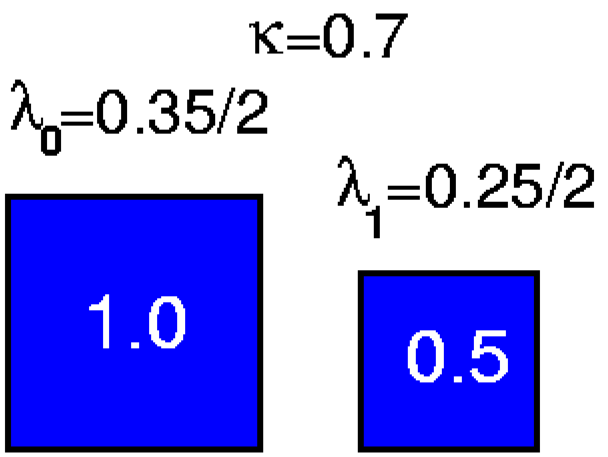
- Feature size (λ) shrinks

  $\lambda_1 = \kappa \lambda_0$

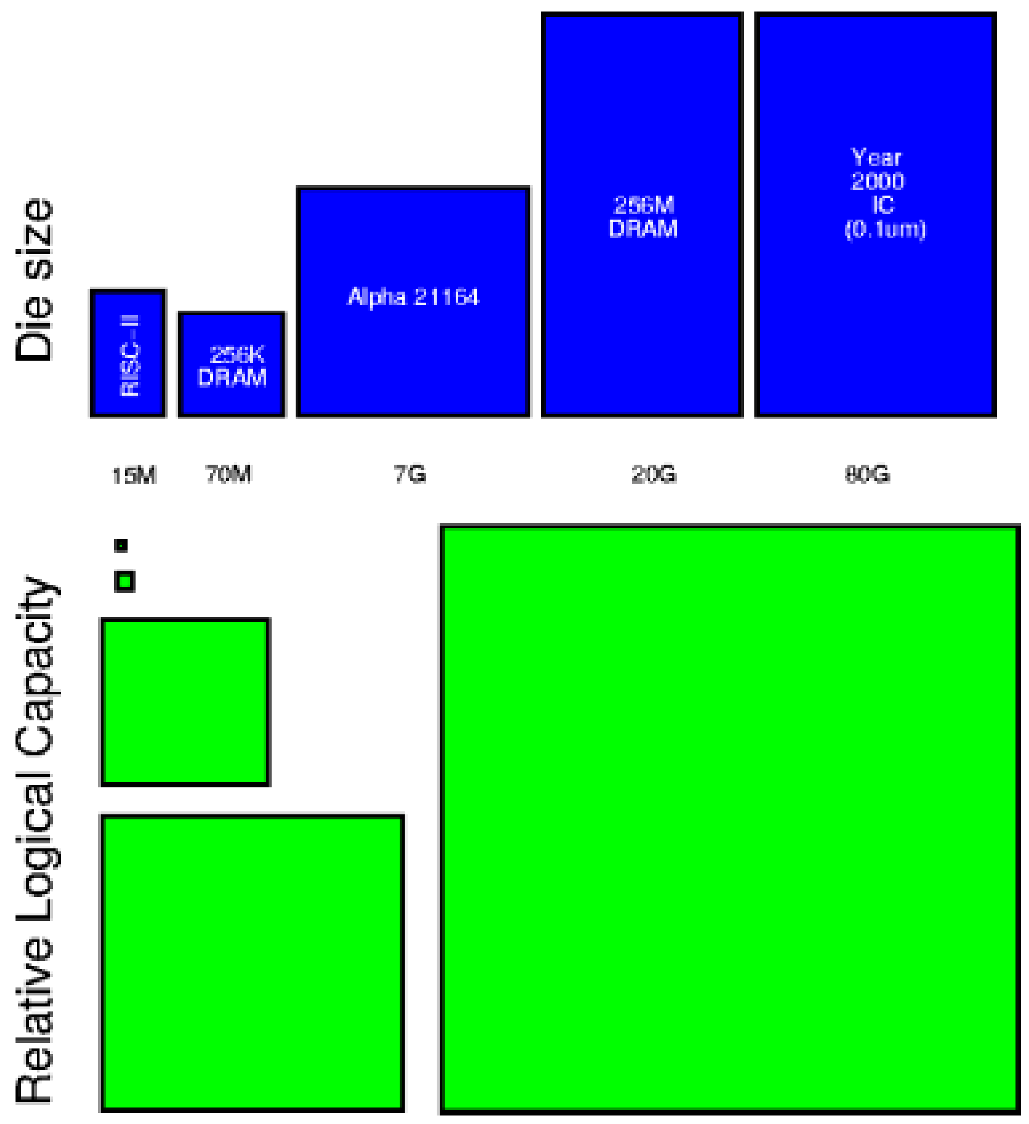
  – devices shrink (κ^2)

  – device capacity grows

    - $1/\kappa^2$ keep same die size
    - greater, if grow die size



Area Perspective



Technology (Speed)

- Raw speed:
  - logic delays decrease (κ, assuming $V_1 = \kappa V_0$)
    - but voltage often not scaled
  - interconnect delays
    - break even in normalized units
    - process advances (Cu, thicker lines) improve
    - larger chips have longer wires

Capacity

- For highly parallel problems
  - more silicon
  - more computation
  - faster execution
- A board full of FPGAs gives a 10× speedup
  - would a board full of Processors also provide this speedup?
  - density or scalability advantage?

Most Economical Solution

- As an Engineer, want most computational power for my $ (silicon area)
  - normalize silicon area to feature size
    - results mostly portable across technologies
  - normalize performance to capacity
    - least area for fixed performance
    - most performance in fixed area
  - look at throughput (compute time) in absolute time, possibly normalized to technology

Example: Multiply

Architecture	Feature	Time
Custom 16$\times$16	0.63μm	40 ns
FPGA	0.60μm	26 ns
16b DSP	0.65μm	50 ns
RISC	0.75μm	2900 ns

Example: Multiply Area

Architecture	Feature	Area	Time
Custom 16$\times$16	0.63μm	2.6Mλ^2	40 ns
FPGA	0.60μm	1.25Mλ^2/CLB$\times$316 CLBs 395Mλ^2	26 ns
16b DSP	0.65μm	350Mλ^2	50 ns
RISC	0.75μm	125Mλ^2, 66 ns/cycle$\times$44 cycles	2900 ns

Example: Multiply Normalized

Architecture	**Feature**	**Area**	**Time**	$\frac{\textbf{mpy}}{\lambda^2\textbf{s}}$
Custom 16×16	0.63μm	2.6Mλ^2	40 ns	9.6
FPGA	0.60μm	1.25Mλ^2/CLB×316 CLBs 395Mλ^2	26 ns	0.097
16b DSP	0.65μm	350Mλ^2	50 ns	0.057
RISC	0.75μm	125Mλ^2, 66 ns/cycle×44 cycles	2900 ns	0.0028

Example: Multiply Summary

Architecture	**Feature Size** (λ)	**Area and Time**	**16×16**		**8×8**	
			mpy / λ^2**s**	**scale** / λ^2**s**	**mpy** / λ^2**s**	**scale** / λ^2**s**
Custom 16×16	0.63μm	2.6Mλ^2, 40 ns	9.6	9.6	9.6	9.6
Custom 8×8	0.80μm	3.3Mλ^2, 4.3 ns			70	70
Gate-Array 16×16	0.75μm	26Mλ^2, 30ns	1.3	1.3	1.3	1.3
FPGA	0.60μm	1.25Mλ^2/CLB				
(XC4K)		316 CLBs, 26 ns	0.097			
		84 CLBs, 40 ns		0.24		
		220 CLBs, 12.1 ns			0.30	
		22 CLBs, 25 ns				1.5
16b DSP	0.65μm	350Mλ^2, 50 ns	0.057	0.057	0.057	0.057
RISC	0.75μm	125Mλ^2, 66 ns/cycle				
(no multiplier)		two 16b operands – 44 cycles	0.0028			
		16b constant – 7 cycles		0.017		
		one 8b operand – 24 cycles			0.0051	
		8b constant – 4 cycles				0.030

Example: FIR

Architecture	Feature Size (λ)	$\frac{TAPs}{\lambda^2 \mathbf{s}}$
32b RISC	0.75μm	0.020
16b DSP	0.65μm	0.057
32b RISC/DSP	0.25μm	0.021
64b RISC	0.18μm	0.064
FPGA (XC4K)	0.60μm	1.9
(Altera 8K)	0.30μm	3.6
Full Custom	0.75μm	3.6
	0.60μm	3.5
	0.75μm	2.4
(fixed coefficient)	0.60μm	56
(*n.b.* 16b samples)		

Example: FIR

Architecture	**Feature Size (λ)**	**Area and Time**	$\frac{TAPs}{\lambda^2\mathbf{s}}$
32b RISC	0.75μm	125Mλ^2, 66 ns/cycle$\times$6+cycles/TAP	0.020
16b DSP	0.65μm	350Mλ^2, 50 ns/TAP	0.057
32b RISC/DSP	0.25μm	1.2Gλ^2, 40 ns/TAP	0.021
64b RISC	0.18μm	6.8Gλ^2, 2.3 ns/TAP	0.064
FPGA (XC4K)	0.60μm	240 CLBs, 14.3 ns/8-TAPs	1.9
(Altera 8K)	0.30μm	30 LEs$\times$0.92Mλ^2/LE, 10 ns/TAP	3.6
Full Custom	0.75μm	400Mλ^2, 45 ns/64 TAPs	3.6
	0.60μm	140Mλ^2, 33 ns/16 TAPs	3.5
	0.75μm	82Mλ^2, 50 ns/10 TAPs	2.4
(fixed coefficient)	0.60μm	114Mλ^2, 6.7 ns/43 TAPs	56
		(*n.b.* 16b samples)	

Example: DNA/Splash Revisited

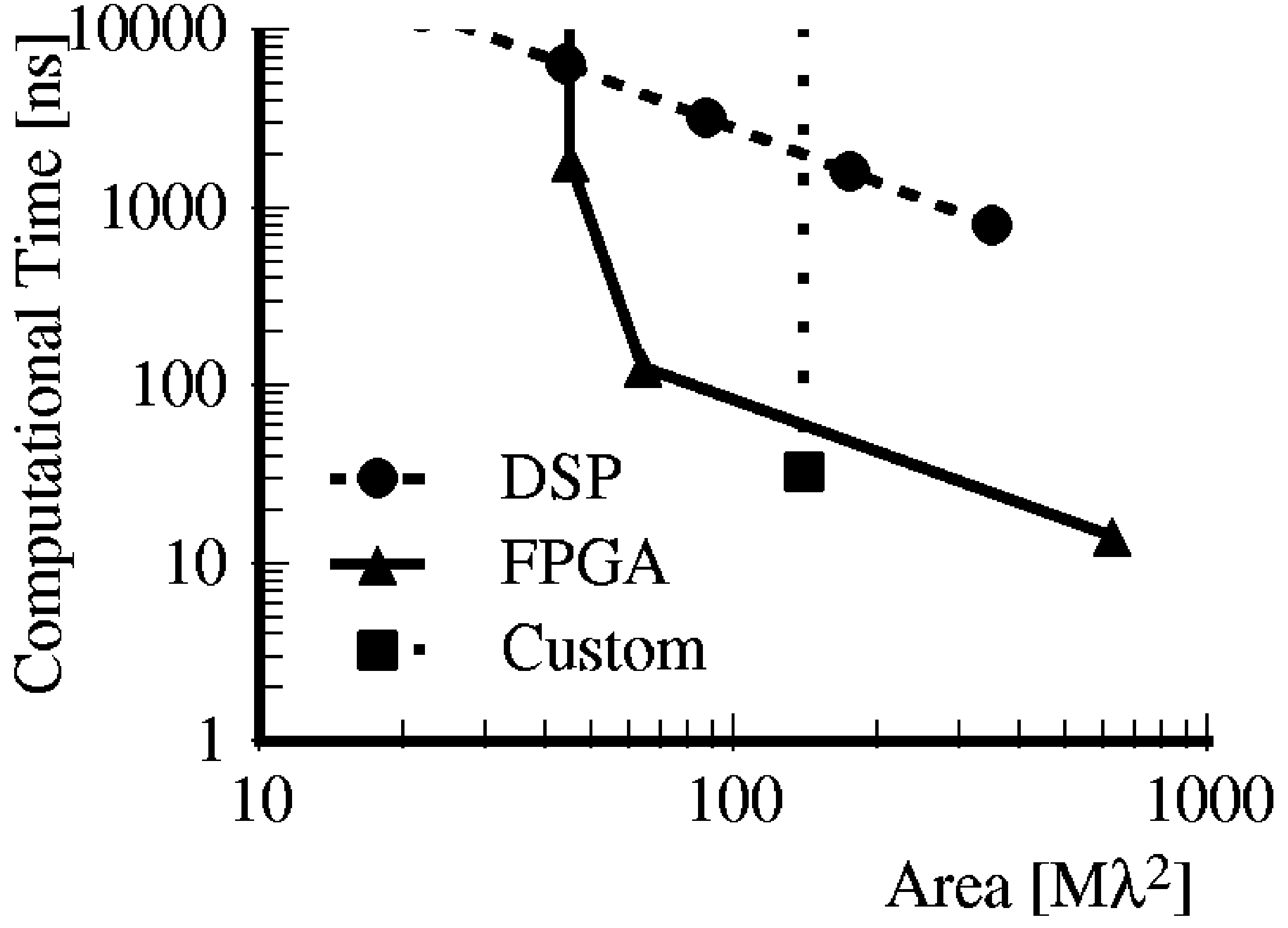
Architecture	**Feature Size (λ)**	**Area**	**Cell Updates per Second**	$\frac{\mathbf{cu}}{\lambda^2\mathbf{s}}$
Custom	2.0μm	270Mλ^2	500M	1.9
FPGA				
(SPLASH 2)	0.60μm	43Gλ^2	3,000M	0.070
(SPLASH)	0.60μm	33Gλ^2	370M	0.012
RISC				
(SparcStation I)	0.75μm	273Mλ^2	0.87M	0.0032
(SparcStation 10)	0.40μm	1.6Gλ^2	1.2M	0.00075

N.B. includes memory area for SPLASH

Area-Time Curves

- Simple performance density picture complicated by:
  - Non-ideal area-time curves
  - Non-scalable designs
  - Limited parallelism
  - Limited throughput requirements

AT Example: FIR



Characterization

- Performance alone doesn’t tell the story
- Need to track:
  - resource requirements
    - *e.g.* CLBs, components
  - absolute compute time
  - energy
  - technology
- Scaling (A-T) curves are beneficial

Summary

- To conquer confusion:
  - compare FPGA-based computations with alternative implementation technologies
  - take care in comparison to normalize
- Many reasons for choosing a technology beyond cost/performance
  - always want to know what you’re paying for what you get