

CS184a: Computer Architecture (Structures and Organization)

Day7: October 16, 2000
Instruction Space
(computing landscape)

Last Time

- Instruction Taxonomy
- VLSI Scaling

Today

- Model Instruction Architecture
 - implied costs
 - gross application characteristics
- Broader comparisons of computing devices
 - custom
 - MPGA (Metal or Mask Programmable)
 - FPGA
 - processors

Quotes

- *If it can't be expressed in figures, it is not science; it is opinion.* -- Lazarus Long

Modeling

- Why do we model?

Motivation

- Need to understand
 - How costly (big) is a solution
 - How compare to alternatives
 - Cost and benefit of flexibility

What we really want:

- Complete implementation of our application
- For each architectural alternatives
 - In same implementation technology
 - w/ multiple area-time points

Reality

- Seldom get it packaged that nicely
 - much work to do so
 - technology keeps moving
- Deal with
 - estimation from components
 - technology differences
 - few area-time points

Modeling Instruction Effects

- Restrictions from “ideal” save area
- Restriction from “ideal” limits usability (yield) of PE

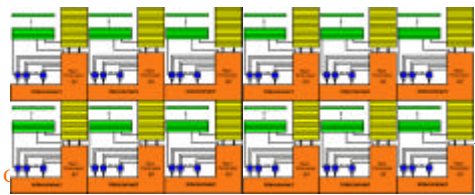
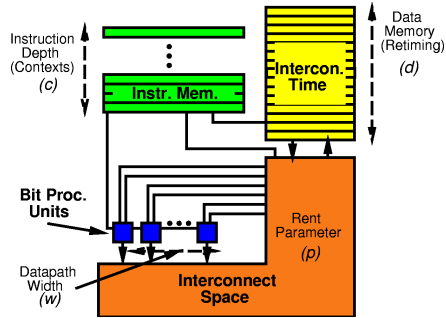
- Want to understand effects
 - area model
 - utilization/yield model

Efficiency/Yield Intuition

- What happens when
 - Datapath is too wide?
 - Datapath is too narrow?
 - Instruction memory is too deep?
 - Instruction memory is too shallow?

Computing Device

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

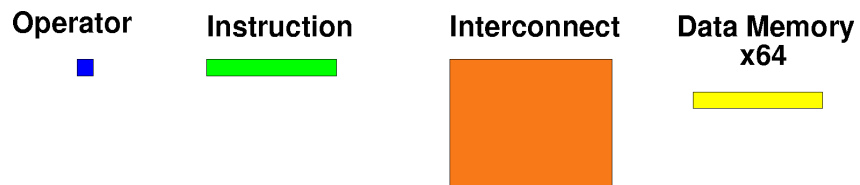


Tile together to build device

11

Relative Sizes

- Bit Operator $10-20K\lambda^2$
- Bit Operator Interconnect $500K-1M\lambda^2$
- Instruction (w/ interconnect) $80K\lambda^2$
- Memory bit (SRAM) $1-2K\lambda^2$



Model Area

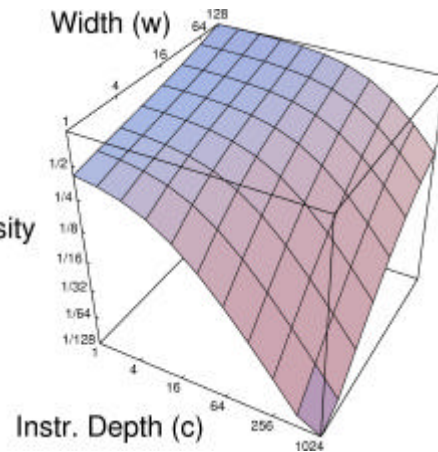
$$\begin{aligned}
 A_{bit_elm} = & A_{fixed} + \underbrace{N_{SW}(N_p, w, p) \cdot A_{SW}}_{\text{interconnect}} \\
 & + \underbrace{\left(\frac{c}{w}\right) \cdot n_{ibits} \cdot A_{mem_cell}}_{\text{instruction memory}} \\
 & + \underbrace{d \cdot A_{mem_cell}}_{\text{retiming memory}}
 \end{aligned}$$

Calibrate Model

FPGA	model $w = 1, d = c = 1, k = 4$	880K λ^2
	Xilinx 4K	630K λ^2
	Altera 8K	930K λ^2
SIMD	model $w = 1000, c = 0, d = 64, k = 3$	170K λ^2
	Abacus	190K λ^2
Processor	model $w = 32, d = 32, c = 1024, k = 2$	2.6M λ^2
	MIPS-X	2.1M λ^2

Peak Densities from Model

- Only 2 of 4 parameters
 - small slice of space
 - 100× density across
- Large difference in peak densities
 - large design space!



Efficiency

- What do we want to maximize?
 - Useful work per unit silicon
 - (not potential/peak work)
- Yield Fraction / Area
- (or minimize (Area/Yield))

Efficiency

- For comparison, look at relative efficiency to ideal.
- Ideal = architecture exactly matched to application requirements
- Efficiency = $A_{\text{ideal}}/A_{\text{arch}}$
- $A_{\text{arch}} = \text{Area Op}/\text{Yield}$

Efficiency Calculation

$$\text{Efficiency} = \frac{A_{\text{matched_arch}}}{A_{\text{arch}}}$$

E.g.

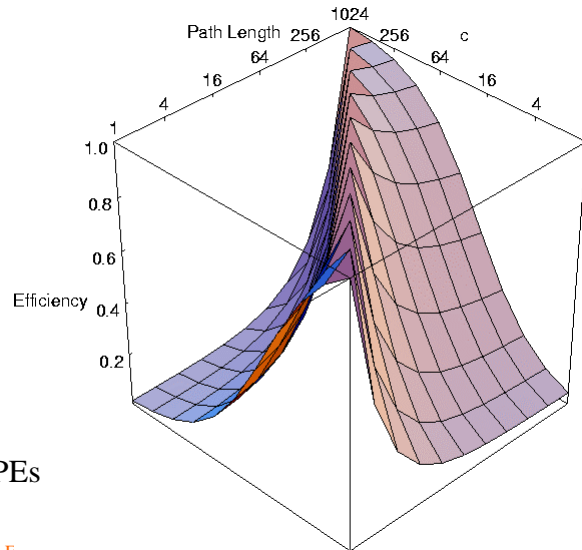
If $w_{\text{task}} > w_{\text{arch}}$:

$$\text{Efficiency} = \frac{w_{\text{task}} \times A_{\text{bit_elm}}|w=w_{\text{task}}}{\left\lceil \frac{w_{\text{task}}}{w_{\text{arch}}} \right\rceil \times w_{\text{arch}} \times A_{\text{bit_elm}}|w=w_{\text{arch}}}$$

If $w_{\text{task}} < w_{\text{arch}}$:

$$\text{Efficiency} = \frac{w_{\text{task}} \times A_{\text{bit_elm}}|w=w_{\text{task}}}{w_{\text{arch}} \times A_{\text{bit_elm}}|w=w_{\text{arch}}}$$

Efficiency with fixed Width

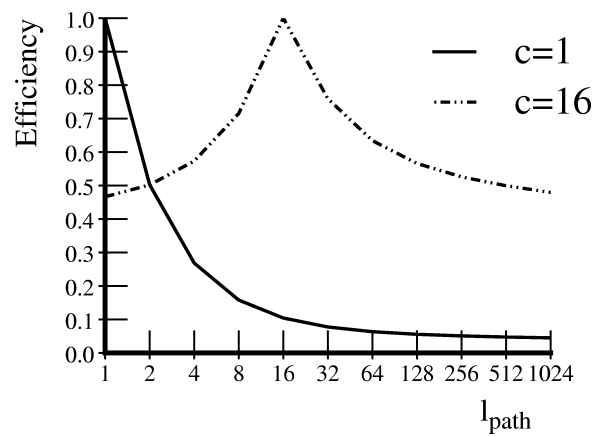


$w=1$,
16K PEs

Caltech CS184a Fall2000 -- DeHon

19

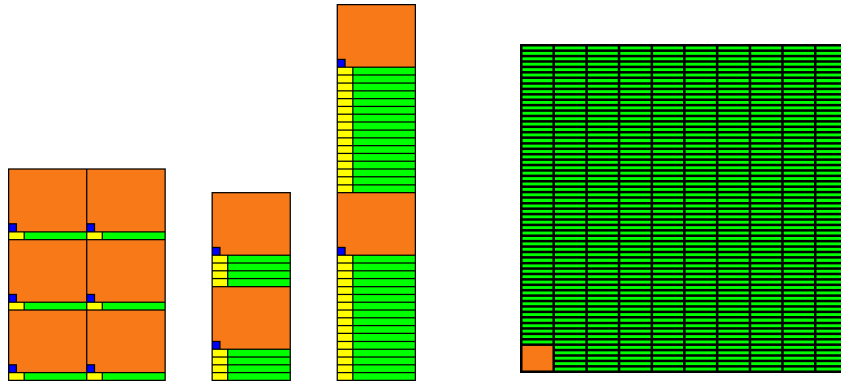
FPGA vs. DPGA Compare



Caltech CS184a Fall2000 -- DeHon

20

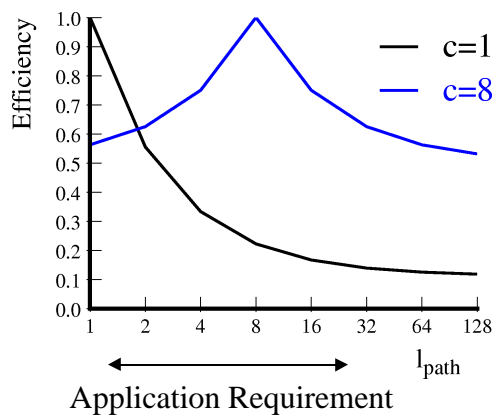
FPGA vs. DPGA



Caltech CS184a Fall2000 -- DeHon

21

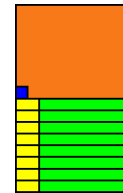
Ideal Efficiency (different model)



Two resources here:

- active processing elements
- operation description/state

Applications need in different proportions.

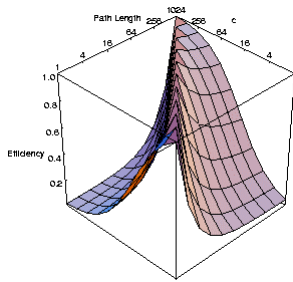


Robust point: $c \cdot A_{ctx} = A_{base}$

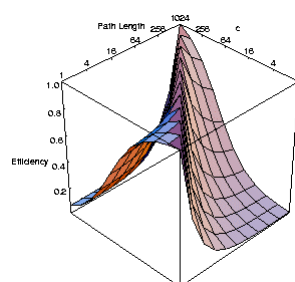
Caltech CS184a Fall2000 -- DeHon

22

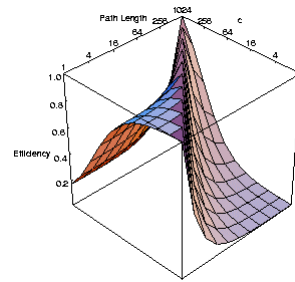
Robust Point depend on Width



w=1



w=8

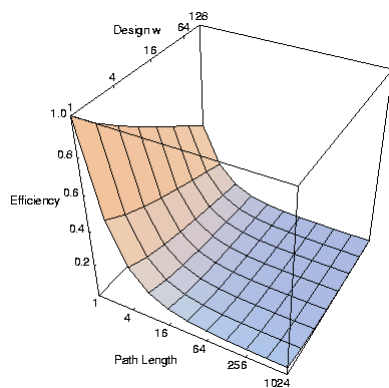


w=64

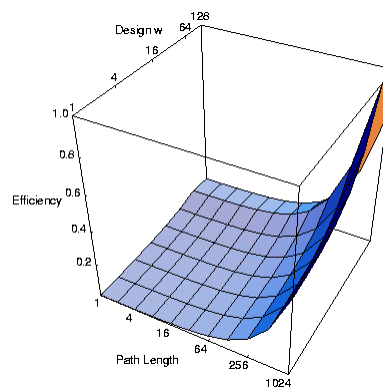
Caltech CS184a Fall2000 -- DeHon

23

Processors and FPGAs



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



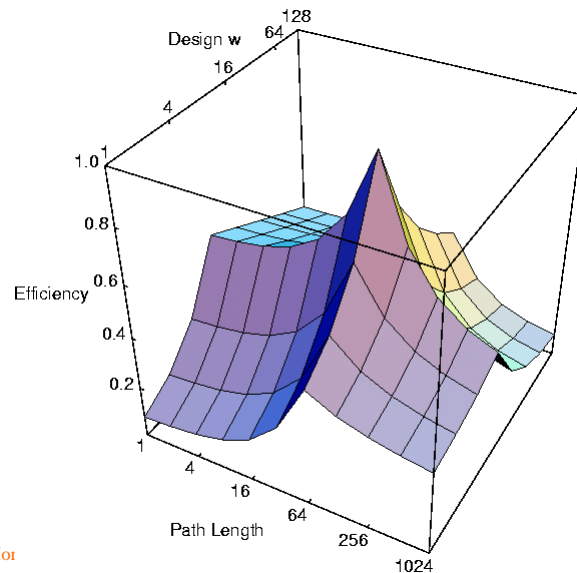
“Processor”
c=d=1024, w=64, k=2

Caltech CS184a Fall2000 -- DeHon

24

Intermediate Architecture

w=8
c=64
16K PEs



Caltech CS184a Fall2000 -- DeHon

25

Caveats

- Model abstracts away many details which are important
 - interconnect (day 11--13)
 - control (day 18)
 - specialized functional units (last half today)
- Applications are a heterogeneous mix of characteristics

Caltech CS184a Fall2000 -- DeHon

26

Modeling Message

- Architecture space is **huge**
- Easy to be very inefficient
- Hard to pick one point robust across entire space

- Why we have so many architectures?

General Message

- Parameterize architectures
- Look at continuum
 - costs
 - benefits
- Often have competing effects
 - leads to maxima/minima

Empirical Comparisons

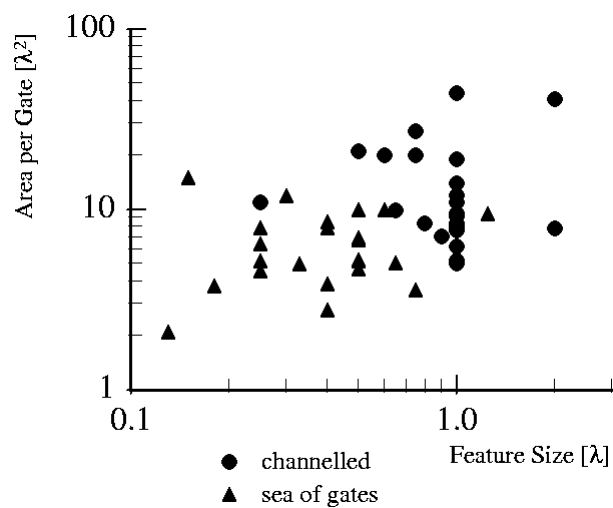
Empirical

- Ground modeling in some concretes
- Start sorting out
 - custom vs. configurable
 - spatial configurable vs. temporal

MPGA vs. Custom?

- AMI CICC'83
 - MPGA 1.0
 - Std-Cell 0.7
 - Custom 0.5
- Toshiba DSP
 - Custom 0.3
- Mosaid RAM
 - Custom 0.2
- GE CICC'86
 - MPGA 1.0
 - Std-Cell 0.4--0.7
 - FF/counter 0.7
 - FullAdder 0.4
 - RAM 0.2

Metal Programmable Gate Arrays



MPGAs

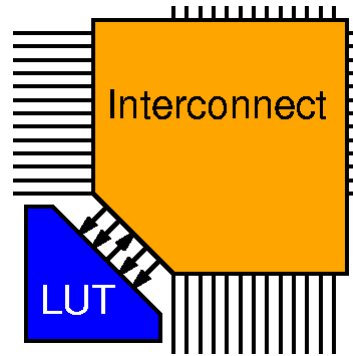
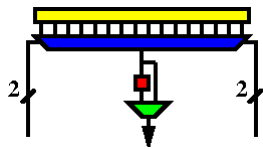
- Modern -- “Sea of Gates”
- yield 35--70%
- maybe $5k\lambda^2/\text{gate}$? (quite a bit of variance)

FPGA Table

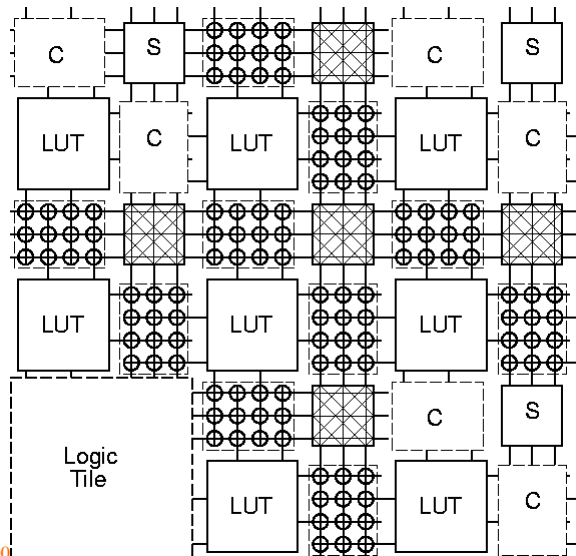
Year	Design	Organization	Max	λ	λ^2 area	cycle
1986	Xilinx 2K	CLB (4-LUT)	100	1μ	500K	20 ns
1988	Xilinx 3K	CLB (2×4-LUT)	320	0.6μ	1.3M	13 ns
1992	Xilinx 4K	CLB (2×4-LUT +)	1024	0.6μ	1.25M	7 ns
1995	Xilinx 5K	CLB (4×4-LUTS)	484	0.3μ	2.25M	6 ns
1995	Altera 8K	LE (4-LUT)	1296	0.3μ	920K	7.5 ns
1995	ORCA 2C	PLC (4×4-LUT)	900	0.3μ	4.3M	7 ns
1998	HSRA	BLB (5-LUT/2×4-LUT ?)	–	0.2μ	2M	4 ns
	Model	4-LUT	2K	–	800K	–
	Model	4-LUT	16K	–	1M	–

Conventional FPGA Tile

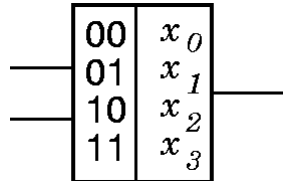
K-LUT (typical $k=4$)
w/ optional
output Flip-Flop



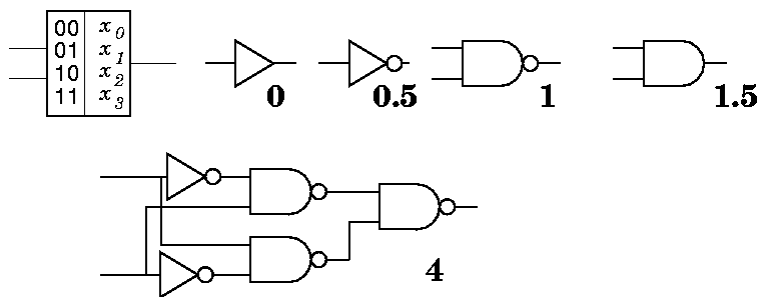
Toronto FPGA Model



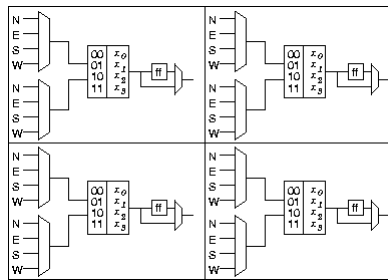
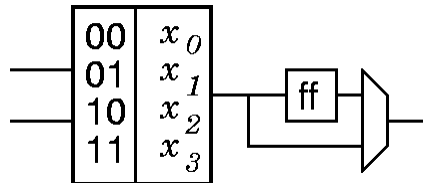
How many gates?



“gates” in 2-LUT

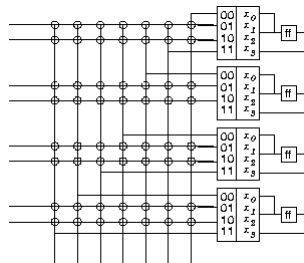


Now how many?

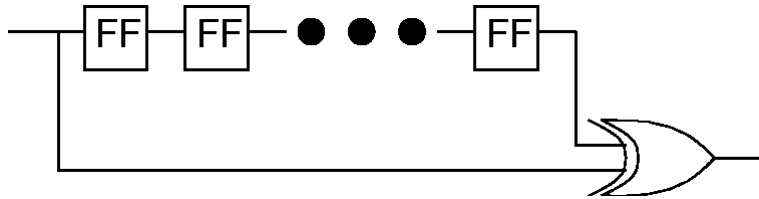


Gates/unit area?

Usable gates?



Gates Required?



Depth=3, Depth=2048?

Gate metric for FPGAs?

- Day5: several components for computations
 - compute element
 - interconnect:
 - space
 - time
 - instructions
- Not all applications need in same balance
- Assigning a single “capacity” number to device is an oversimplification

MPGA vs. FPGA

- MPGA (SOG GA)
 - $5K\lambda^2/\text{gate}$
 - 35-70% usable (50%)
 - $7\text{-}17K\lambda^2/\text{gate net}$
- Xilinx XC4K
 - $1.25M\lambda^2/\text{CLB}$
 - 17--48 gates (26?)
 - $26\text{-}73K\lambda^2/\text{gate net}$
- Ratio: 2--10 (5)

Adding ~2x Custom/MPGA,
Custom/FPGA ~10x

MPGA vs. FPGA

- MPGA (SOG GA)
 - $\lambda=0.6\mu$
 - $\tau_{gd}\sim 1\text{ns}$
- Xilinx XC4K
 - $\lambda=0.6\mu$
 - 1--7 gates in 7ns
 - 2-3 gates typical
- Ratio: 1--7 (2.5)

Note:

Lecture jumped from here to
finishing up
(cover intermediate material later)

Processors and FPGAs

Metric: $\frac{4 \text{ input gate-evaluations}}{\lambda^2 \cdot \mathbf{s}}$

$$\text{Processor: } \frac{2 \times N_{ALU} \times w_{ALU}}{A_{proc} \times t_{cycle}} \quad \text{FPGA: } \frac{N_{4LUT}}{A_{array} \times t_{cycle}}$$

Processors and FPGAs

Year	Design	Organization	λ	λ^2 area	cycle	$\frac{ge's}{\lambda^2 s}$
Microprocessors						
1984	MIPS	1 × 32	1.5 μ t	15M	250ns	17
1987	MIPS-X	1 × 32	1.0 μ t	68M	50ns	19
1994	MIPS	1 × 32	0.28 μ t	1.7G	2ns	19
1992	Alpha	1 × 64	0.38 μ t	1.7G	5ns	15
1995	Alpha	2 × 64	0.25 μ t	4.8G	3.3ns	18
1996	Alpha	2 × 64	0.18 μ t	6.8G	2.3ns	17
Reconfigurable ALUs						
1992	PADDI	8 × 16	0.6 μ t	126M	40ns	50
1995	PADDI-2	48 × 16	0.5 μ t	515M	20ns	150
FPGAs						
1986	Xilinx 2K	1 CLB (4 LUT)	1.0 μ t	500K	20ns	100
1988	Xilinx 3K	64 CLBs (2 4-LUT)	0.6 μ t	83M	13ns	120
1992	Xilinx 4K	49 CLBs (2 4-LUT)	0.6 μ t	61M	7ns	230
1995	Xilinx 5K	49 CLBs (4 4-LUT)	0.3 μ t	110M	6ns	290

Caltech CS18

47

Raw Density Summary

- Area
 - MPGA 2-3x Custom
 - FPGA 5x MPGA
- Area-Time
 - Gate Array 6-10x Custom
 - FPGA 15-20x Gate Array
 - Processor 10x FPGA

Caltech CS184a Fall2000 -- DeHon

48

Raw Density Caveats

- Processor/FPGA may solve more specialized problem
- Problems have different resource balance requirements
 - ...can lead to low yield of raw density

Broadening Picture

- Compare larger computations
- For comparison
 - throughput density metric: results/area-time
 - normalize out area-time point selection
 - high throughput density
 - -> most in fixed area
 - -> least area to satisfy fixed throughput target

Multiply

Architecture	Feature Size (λ)	Area and Time	16x16		8x8	
			mpy $\frac{\lambda^2}{s}$	scale $\frac{\lambda^2}{s}$	mpy $\frac{\lambda^2}{s}$	scale $\frac{\lambda^2}{s}$
Custom 16x16	0.63 μ m	2.6M λ^2 , 40 ns	9.6	9.6	9.6	9.6
Custom 8x8	0.80 μ m	3.3M λ^2 , 4.3 ns			70	70
Gate-Array 16x16	0.75 μ m	26M λ^2 , 30ns	1.3	1.3	1.3	1.3
FPGA (XC4K)	0.60 μ m	1.25M λ^2 /CLB	0.097	0.24	0.30	1.5
		316 CLBs, 26 ns				
		84 CLBs, 40 ns				
		220 CLBs, 12.1 ns				
22 CLBs, 25 ns						
16b DSP	0.65 μ m	350M λ^2 , 50 ns	0.057	0.057	0.057	0.057
RISC (no multiplier)	0.75 μ m	125M λ^2 , 66 ns/cycle	0.0028	0.017	0.0051	0.030
	two 16b operands – 44 cycles					
	16b constant – 7 cycles					
	one 8b operand – 24 cycles					
		8b constant – 4 cycles				

Caltech CS184a Fall2000 -- DeHon

51

Example: FIR Filtering

$$Y_i = w_1 X_i + w_2 X_{i+1} + \dots$$

Application metric:
TAPs = filter taps
multiply accumulate

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

Caltech CS184a Fall2000 -- DeHon

52

IIR/Biquad

Architecture	Feature Size (λ)	Area and Time	16b TAPs $\frac{\text{TAPs}}{\lambda^2 \text{s}}$	10b TAPs $\frac{\text{TAPs}}{\lambda^2 \text{s}}$
16b DSP	$0.60 \mu\text{m}$	$200\text{M}\lambda^2$, 500 ns/biquad	0.010	0.010
FPGA	$0.60 \mu\text{m}$	60 CLBs, 320 ns/biquad	0.044	
(XC4K)		43 CLBs, 200 ns/biquad		0.093
Full Custom	$0.90 \mu\text{m}$	$68\text{M}\lambda^2$, 11.8 ns/4 biquads		5.0

DES Keysearch

Architecture	Feature Size (λ)	Area	Keys/Second	Keys $\frac{\text{Keys}}{\lambda^2 \text{s}}$
DES IC	$1.5 \mu\text{m}$	$11.1\text{M}\lambda^2$	310K	0.028
FPGA (Altera 8K)	$0.30 \mu\text{m}$	$81188 (930\text{M}\lambda^2)$	800K	0.00086
RISC	$0.30 \mu\text{m}$	$1.8\text{G}\lambda^2$	41K	0.000023

<<http://www.cs.berkeley.edu/~iang/isaac/hardware/>>

DNA Sequence Match

- **Problem:** “cost” of transform $S_1 \rightarrow S_2$
- **Given:** cost of insertion, deletion, substitution
- **Relevance:** similarity of DNA sequences
 - evolutionary similarity
 - structure predict function
- **Typically:** new sequence compared to large database

DNA Sequence Match

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

N.B. includes memory area for SPLASH

Floating-Point Add (single prec.)

Architecture	Reference	λ	area and time	32b FP ADDs $\frac{\lambda^2}{s}$
Custom Macrocells				
32b FP ALU	JSSC92	0.6μ	$32M\lambda^2$, 30 ns	1.0
32b Adder	EUROASIC92	0.4μ	$49M\lambda^2$, 60 ns	0.34
32/64b FPU	CICC92	0.25μ	$980M\lambda^2$, 2 per 12.5 ns	0.16
Coprocessor				
32b FP Processor	ISSCC88	0.75μ	$220M\lambda^2$, 55 ns	0.083
32b FP Processor	SSC89	0.65μ	$400M\lambda^2$, 23 ns	0.11
Processor FP support				
64b RISC w/FPU	ISSCC90	0.4μ	$1.4G\lambda^2$, 2×25 ns cycle	0.014
64b RISC w/FPU	ISSCC92	0.38μ	$1.7G\lambda^2$, 5 ns cycle	0.12
64b RISC w/FPU	ISSCC95	0.25μ	$4.8G\lambda^2$, 3.3ns cycle	0.063
FPGA				
	FCCM96	0.3μ	500 Flex8K LEs ($920K\lambda^2/LE$), 150 ns	0.014
	FCCM98	0.25μ	315 XC4KE CLBs ($1.25M\lambda^2/CLB$), 33 ns	0.077
Processor no FP support				
32b RISC, no FPU	ASPLOS85	1.5μ	$15M\lambda^2$, $16\mu s$	0.0042

Caltech CS184a Fall2000 -- DeHon

57

Floating-Point Mpy (single prec.)

Architecture	Reference	λ	area and time	32b FP MPYs $\frac{\lambda^2}{s}$
Custom Macrocells				
32b mpy	JSSC92	0.6μ	$43M\lambda^2$, 30 ns	0.78
32b mpy	EUROASIC92	0.4μ	$81M\lambda^2$, 42 ns	0.29
32/64b FPU	CICC92	0.25μ	$980M\lambda^2$, 2 per 12.5 ns	0.16
Coprocessor				
dedicated 32b mpy	JSSC84	1μ	$33M\lambda^2$, 79 ns	0.38
dedicated 32b mpy	ISSCC87	0.6μ	$160M\lambda^2$, 67 ns	0.093
32b FP Processor	ISSCC88	0.75μ	$200M\lambda^2$, 55 ns	0.083
32b FP Processor	JSSC89	0.65μ	$400M\lambda^2$, 23 ns	0.11
Processor FP support				
64b RISC w/FPU	ISSCC90	0.4μ	$1.4G\lambda^2$, 2×25 ns cycle	0.014
64b RISC w/FPU	ISSCC92	0.38μ	$1.7G\lambda^2$, 5 ns cycle	0.12
64b RISC w/FPU	ISSCC95	0.25μ	$4.8G\lambda^2$, 3.3ns cycle	0.063
FPGA				
	FCCM96	0.3μ	344 Flex8K LEs ($920K\lambda^2/LE$), 755 ns	0.0042
	FCCM98	0.25μ	265 CLBs ($1.25M\lambda^2/CLB$), 200 ns	0.0151
Processor no FP support				
32b RISC, no FPU	ASPLOS85	1.5μ	$15M\lambda^2$, $7\mu s$	0.0096

Caltech CS184a Fall2000 -- DeHon

58

Finishing Up...

Big Ideas [MSB Ideas]

- Arch. structure and application requirements mismatch \Rightarrow inefficiencies
- Model \Rightarrow visualize efficiency trends
- Architecture space is huge
 - can be very inefficient
 - need to learn to navigate
- Raw densities: custom:ga:fpga:processor
 - 1:5:100:1000
 - close with specialization

Big Ideas

[MSB-1 Ideas]

- Special-purpose functional units in processors/DSPs, much lower net benefit since need to control and interconnect

Degrade Notes

- We'll cover these issues in more detail as we get into them later in the course

Degrade from Peak: Processors

- Ops w/ no gate evaluations (interconnect)
- Ops use limited word width
- Stalls waiting for retimed data

$$E(\text{Functional Density}) = \frac{\text{Gate Evaluations}}{\text{Datapath Bit}} \times \frac{\text{Datapath Bits}}{\text{pinst}} \times \frac{\text{pinsts}}{\text{Issue Slot}} \times \frac{\text{Issue Slots}}{\text{Clock Cycle}} \times \frac{1}{\text{area} \times t_{\text{cycle}}}$$

Degrade from Peak: FPGAs

- Long path length --> not run at cycle
- Limited throughput requirement
 - bottlenecks elsewhere limit throughput req.
- Insufficient interconnect
- Insufficient retiming resources (bandwidth)

Degrade from Peak: Custom/MPGA

- Solve more general problem than required
 - (more gates than really need)
- Long path length
- Limited throughput requirement
- Not needed or applicable to a problem