ESE680-002 (ESE534): Computer Organization

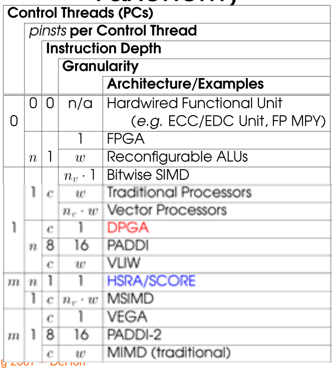
Day 9: February 7, 2007

Instruction Space Modeling

Last Time

- Instruction Requirements
- Instruction Space

Architecture Instruction Taxonomy



Control Threads (PCs)	*pinsts* per Control Thread	Instruction Depth	Granularity	Architecture/Examples
0	0	0	n/a	Hardwired Functional Unit (*e.g.* ECC/EDC Unit, FP MPY)
	n	1	1	FPGA
			w	Reconfigurable ALUs
1	1	c	$n_v \cdot 1$	Bitwise SIMD
			w	Traditional Processors
			$n_v \cdot w$	Vector Processors
	n	c	1	DPGA
		8	16	PADDI
		c	w	VLIW
m	n	1	1	HSRA/SCORE
	1	c	$n_v \cdot w$	MSIMD
m	1	c	1	VEGA
		8	16	PADDI-2
		c	w	MIMD (traditional)

Today

- Instructions
  - Model Architecture
    - implied costs
    - gross application characteristics

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

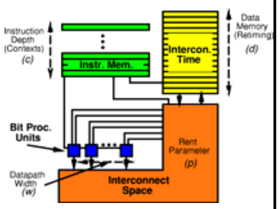
- Want to understand effects
  - area model
  - utilization/yield model

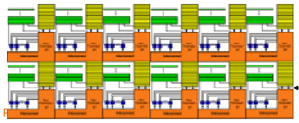
Efficiency/Yield Intuition

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

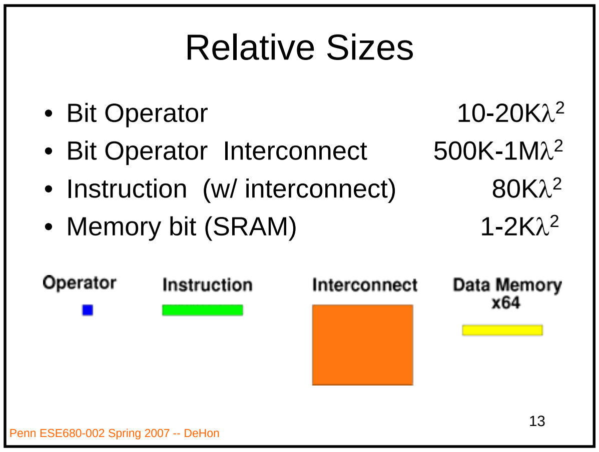
Computing Device

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





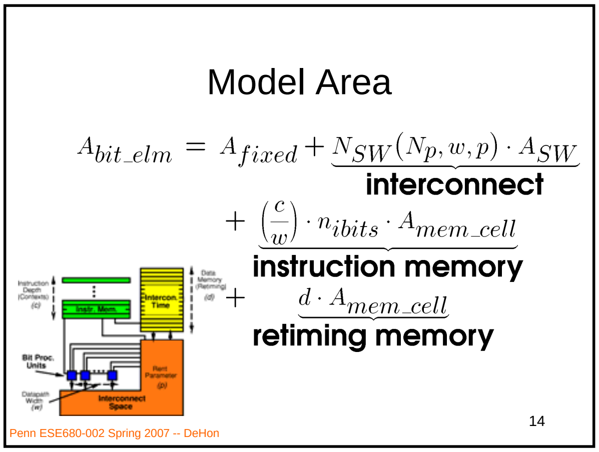
Tile together to build device



Relative Sizes

• Bit Operator	10-20Kλ^2
• Bit Operator Interconnect	500K-1Mλ^2
• Instruction (w/ interconnect)	80Kλ^2
• Memory bit (SRAM)	1-2Kλ^2

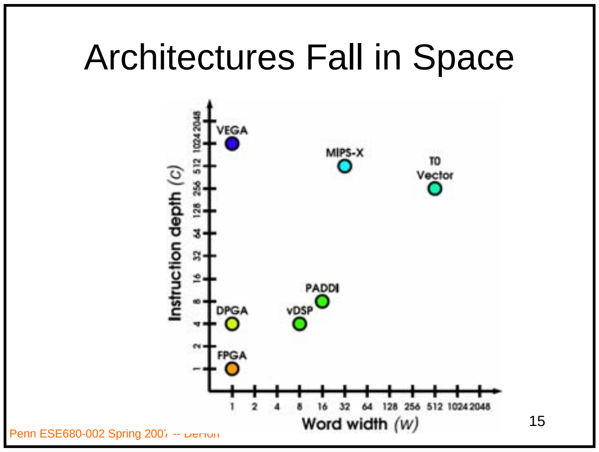
Operator
Instruction
Interconnect
Data Memory x64



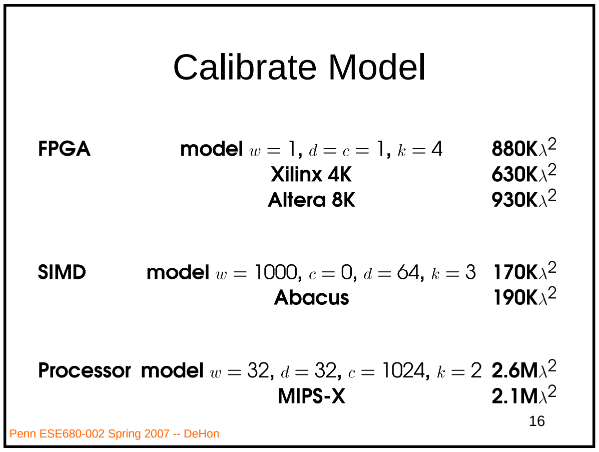
Model Area

$$A_{bit\_elm} = A_{fixed} + \underbrace{N_{SW}(N_p, w, p) \cdot A_{SW}}_{\textbf{interconnect}} + \underbrace{\left(\frac{c}{w}\right) \cdot n_{ibits} \cdot A_{mem\_cell}}_{\textbf{instruction memory}} + \underbrace{d \cdot A_{mem\_cell}}_{\textbf{retiming memory}}$$

Instruction Depth (Contexts) (c)
Instr. Mem.
Intercon. Time
Data Memory (Retiming) (d)
Bit Proc. Units
Rent Parameter (p)
Datapath Width (w)
Interconnect Space

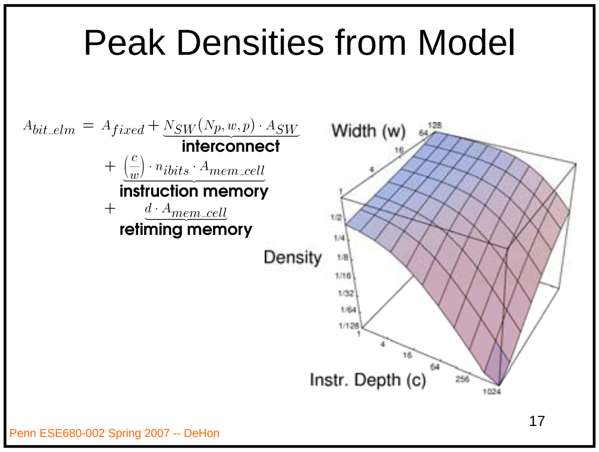


Architectures Fall in Space



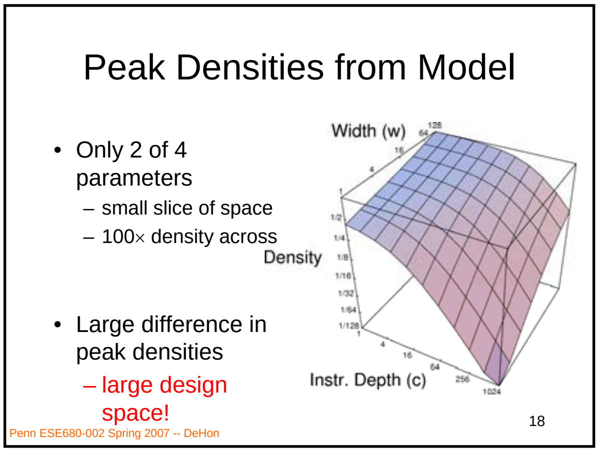
Calibrate Model

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

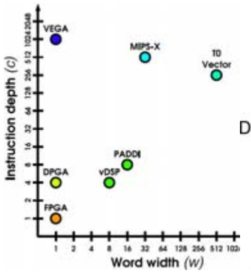
$$A_{bit\_elm} = A_{fixed} + \underbrace{N_{SW}(N_p, w, p) \cdot A_{SW}}_{\textbf{interconnect}} + \underbrace{\left(\frac{c}{w}\right) \cdot n_{ibits} \cdot A_{mem\_cell}}_{\textbf{instruction memory}} + \underbrace{d \cdot A_{mem\_cell}}_{\textbf{retiming memory}}$$

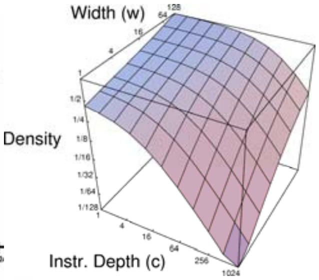


Peak Densities from Model

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

Peak Densities from Model





Efficiency

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

Efficiency

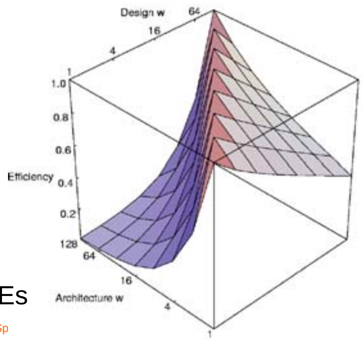
- For comparison, look at relative efficiency to ideal.
- Ideal = architecture exactly matched to application requirements
- Efficiency = A_{ideal}/A_{arch}
- A_{arch} = Area Op/Yield

Width Mismatch Efficiency Calculation

$$E = \frac{Area(Task - on - matched - Architecture)}{Area(Task - on - this - architecture)}$$

$$E = \frac{W_{task} \times A_{bitelm|w=w_{task}}}{W_{arch} \times \left\lceil \frac{W_{task}}{W_{arch}} \right\rceil \times A_{bitelm|w=w_{arch}}}$$

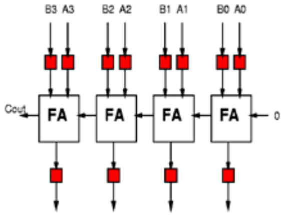
Efficiency: Width Mismatch

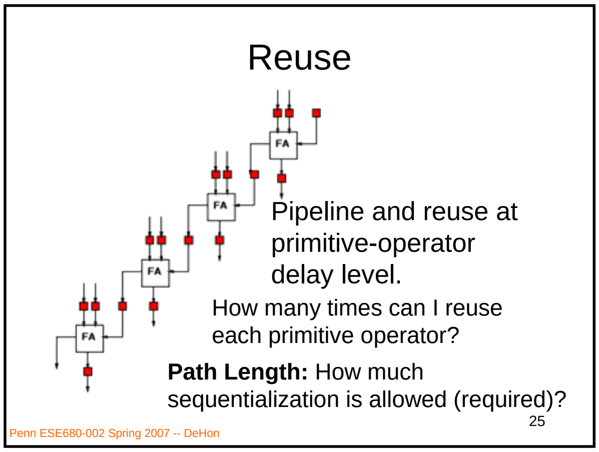


c=1,
16K PEs

Path Length

- How many primitive-operator delays before can perform next operation?
  - Reuse the resource



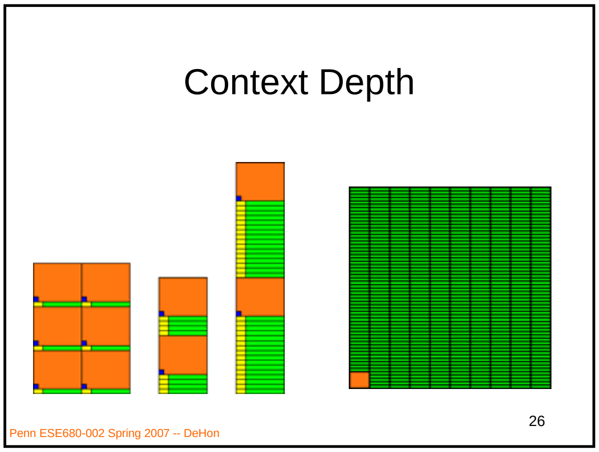


Reuse

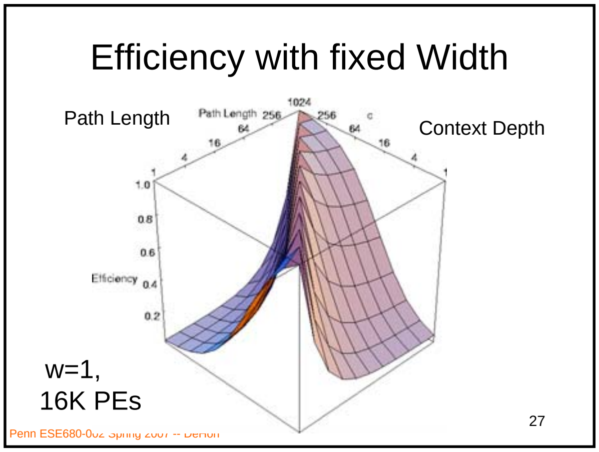
Pipeline and reuse at primitive-operator delay level.

How many times can I reuse each primitive operator?

Path Length: How much sequentialization is allowed (required)?



Context Depth

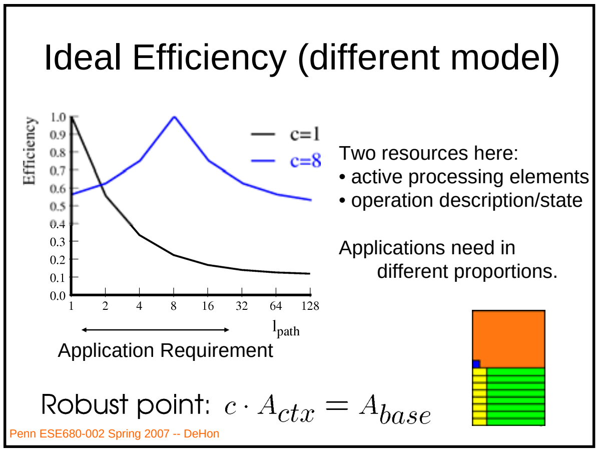


Efficiency with fixed Width

Path Length

Context Depth

w=1,
16K PEs



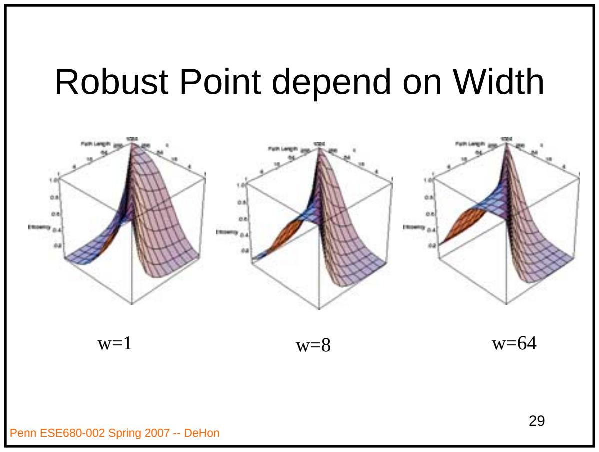
Ideal Efficiency (different model)

Application Requirement

Two resources here:
- active processing elements
- operation description/state

Applications need in different proportions.

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

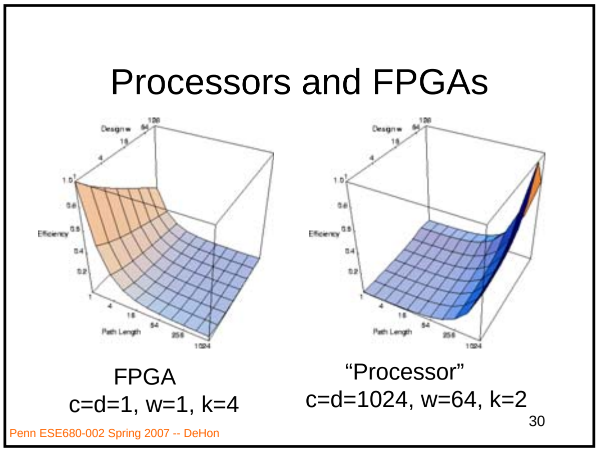


Robust Point depend on Width

w=1

w=8

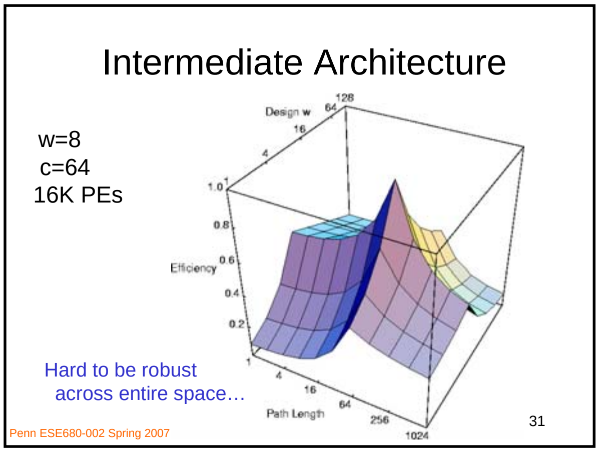
w=64



Processors and FPGAs

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

“Processor”
c=d=1024, w=64, k=2



Intermediate Architecture

w=8
c=64
16K PEs

Hard to be robust
across entire space…

Caveats

- Model abstracts away many details which are important
  - interconnect (day 13--18)
  - control (day 24)
  - specialized functional units (next time)
- Applications are a heterogeneous mix of characteristics

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

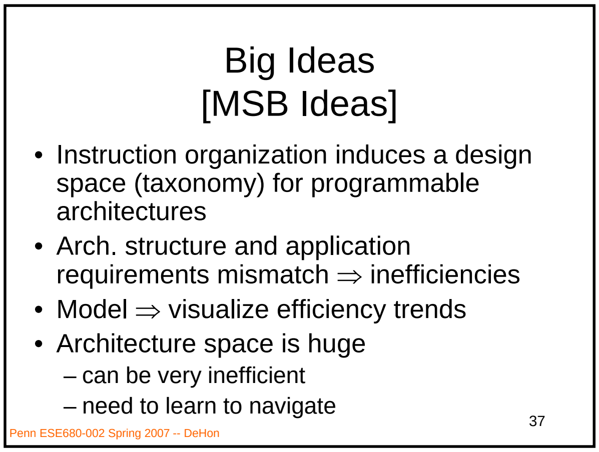
- Parameterize architectures
- Look at continuum
  - costs
  - benefits
- Often have competing effects
  - leads to maxima/minima

Admin

- Assignment 4 out today
  - Did push back due dates for 4 and 5
- Reading for Monday on web
  - Supplemental from this month TRCAD

Big Ideas
[MSB Ideas]

- Applications typically have structure
- Exploit this structure to reduce resource requirements
- Architecture is about understanding and exploiting structure and costs to reduce requirements



Big Ideas [MSB Ideas]

- Instruction organization induces a design space (taxonomy) for programmable architectures
- Arch. structure and application requirements mismatch ⇒ inefficiencies
- Model ⇒ visualize efficiency trends
- Architecture space is huge
  - can be very inefficient
  - need to learn to navigate