0x05 - Design Evaluation

ENGR 3410: Computer Architecture

Jon Tse

Fall 2020

Housekeeping

- Next week's lecture is optional
- Please use the Makefile
- Ask for help early!
-

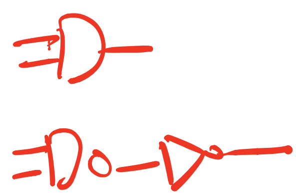
Review - Delay

Delay Unit is dependent on CMOS technology

Table of Timing Arcs from Input -> Output pins

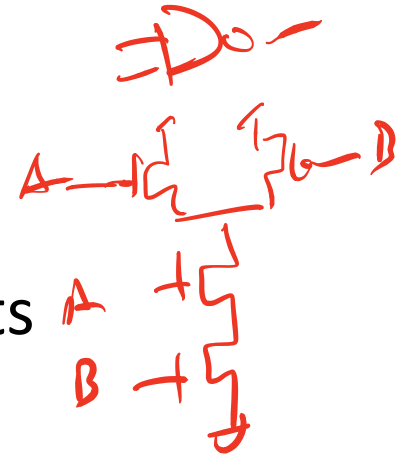
- **Naive** - Gate Delay = 1 Delay Unit
- **Improved** - Gate Delay ∝ # of gate inputs
- **Actual** - Simulation + Experimentation -> GIANT table to cover statistical variation

Remember, CMOS Implementation!



- NAND, NOR, NOT = 1 delay
- AND, OR = 2 delay

Review - Area



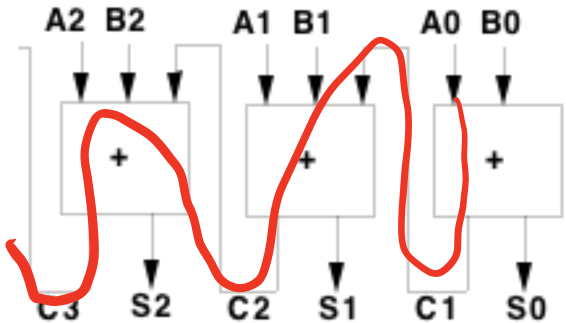
1 Area Unit ≈ 1 Transistor

Area Table - Gates to Transistor Counts

- **Naive** - 2 transistors/input
- **Better** - Ideal implementation count (below)
- **Actual** - Measure CMOS implementation

Gate	Transistors	Gate	Transistors
NAND2	4	AND2	6
NOR2	4	OR2	6
NOT	2	DLatch	16
XOR2	12	DFF	34

Review - Delay Scaling



NAND Full Adder
Max Carry Delay = 5 + 2N

LUT Full Adder
Max Carry Delay = 3N

	A/B0	A/B1	A/B2
S0	6	-	-
C1	5	-	-
S1	5+3	6	-
C2	5+2	5	-
S2	5+2+3	5+3	6
C3	5+2+2	5+2	5

	A/B0	A/B1	A/B2
S0	3	-	-
C1	3	-	-
S1	3+3	3	-
C2	3+3	3	-
S2	3+3+3	3+3	3
C3	3+3+3	3+3	3

Review - Delay Scaling

Which is better?

NAND FA?

2N + 5

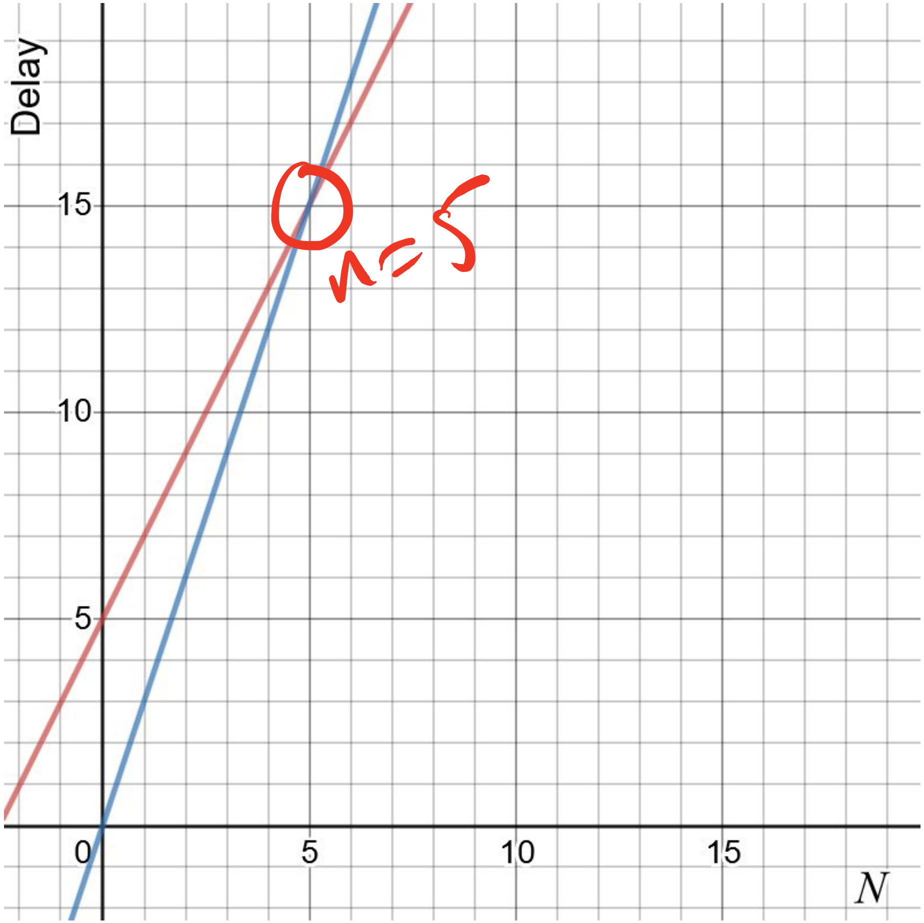
LUT FA?

3N

Depends on N!

Tipping Point

N = 5



Wrap Up - Area Scaling

NAND 2 Full Adder

9x NAND2 = 36 transistors

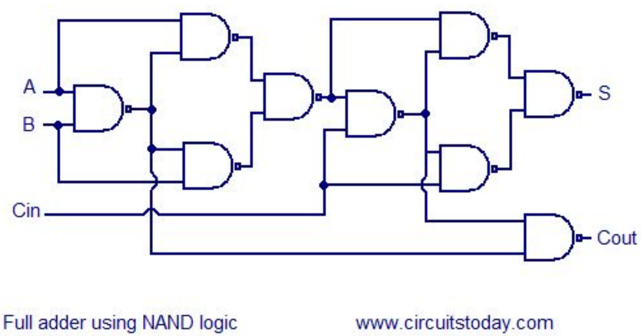
LUT Full Adder

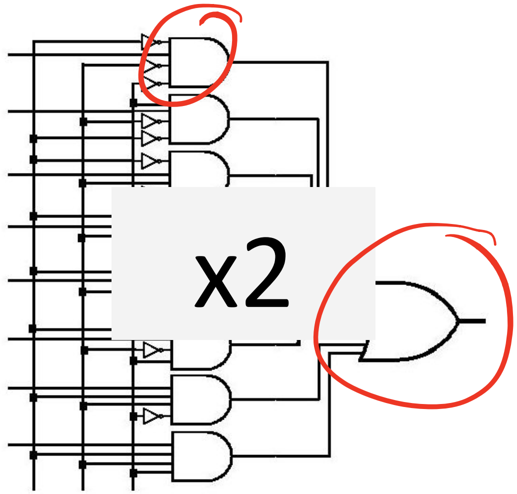
Decoder = 2^3 AND3 + 3 NOT

OR8 = 16 transistors

1 LUT = $2^3(6+2) + 6 + 16 = 86$

2 LUT = 172 !



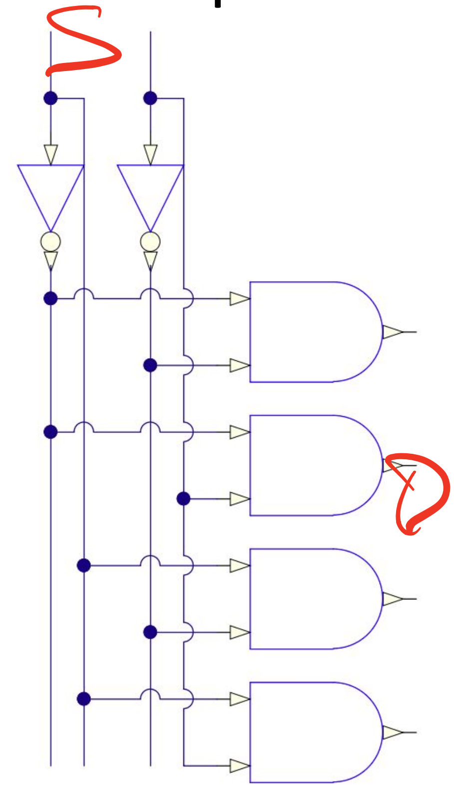


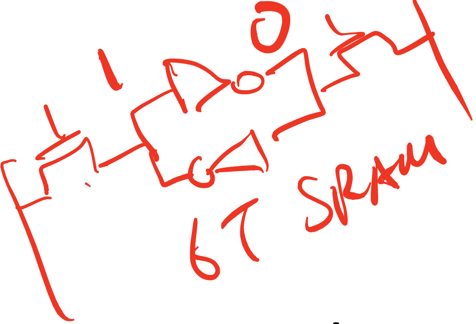
Wrap Up - Area Scaling

- For a decoder with S select bits and D outputs
  - Num Outputs: $D = 2^S$
- Need
  - S Inverters
  - D AND Gates with S inputs
- Area

  AND

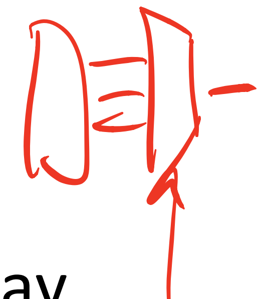
  - $S + 2^S (2S) = S(1+2^{S+1})$
  - If we include D-input OR, add 2D+2





LUTs Seem Bad? $f(x) \to g(x)$

Used to represent arbitrary gates, but less efficient than dedicated circuit.



Very flexible, used in FPGAs

FPGA = Field Programmable Gate Array

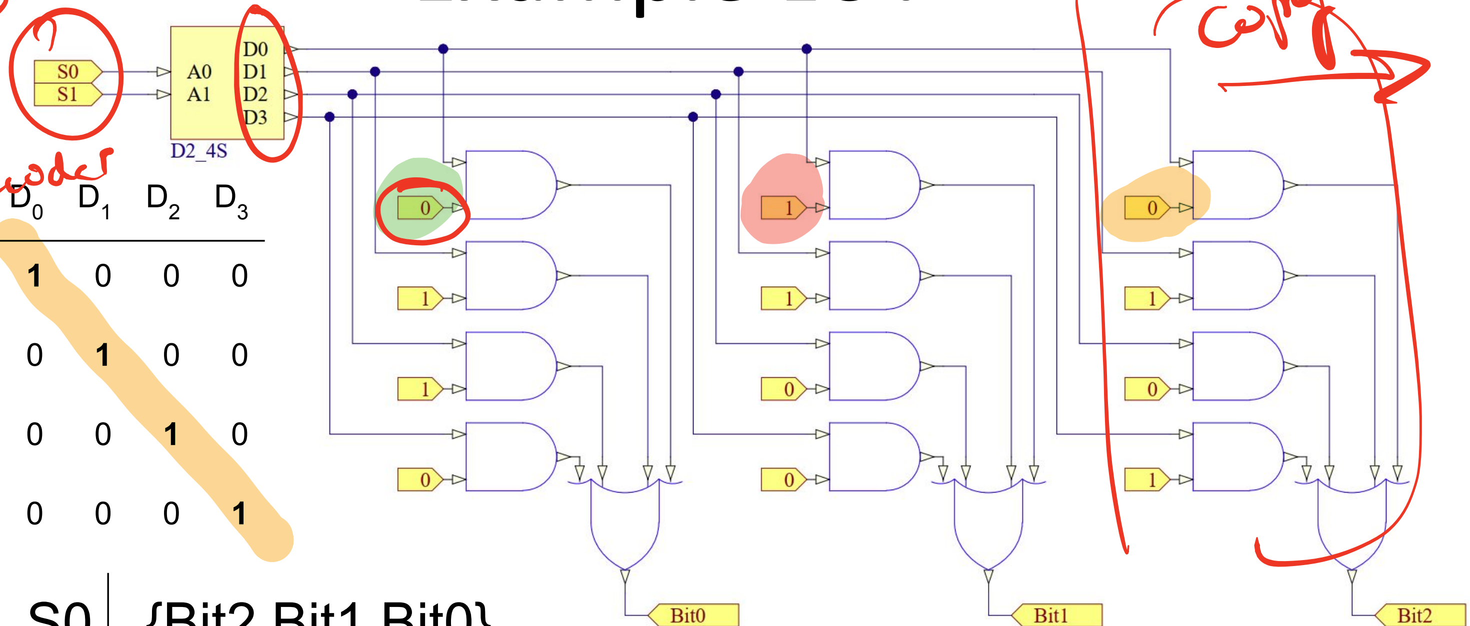
Key Concept: Specialization -> Efficiency

When to exercise the tradeoff?

Example LUT

select

output



copy

LSB

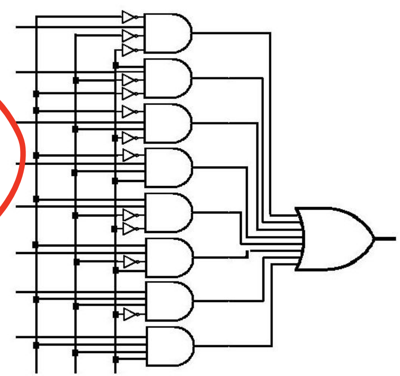
MSB

Decoder

S_1	S_0	D_0	D_1	D_2	D_3
0	0	**1**	0	0	0
0	1	0	**1**	0	0
1	0	0	0	**1**	0
1	1	0	0	0	**1**

S1	S0	{Bit2,Bit1,Bit0}
0	0	010
0	1	111
1	0	001
1	1	100

$$\text{Output} = \begin{cases} 010, & S = 0 \\ 111, & S = 1 \\ 001, & S = 2 \\ 100, & S = 3 \end{cases}$$



Today

VERY Brief Discussion of Energy/Power

This class = 3 Classes

Digital Logic + Computer Org + Computer Arch

Add 1 more class!

Bonus Class: Heuristic Methods for Optimization

https://www.jontse.com/courses/cs5722.html

https://insidesmallbusiness.com.au/planning-management/bonuses-a-good-ideal

Energy Consumption

- Energy (Joules) is dissipated every time a circuit switches from 0-1 and back

- Energy (stored in capacitor): $\frac{1}{2}CV^2$

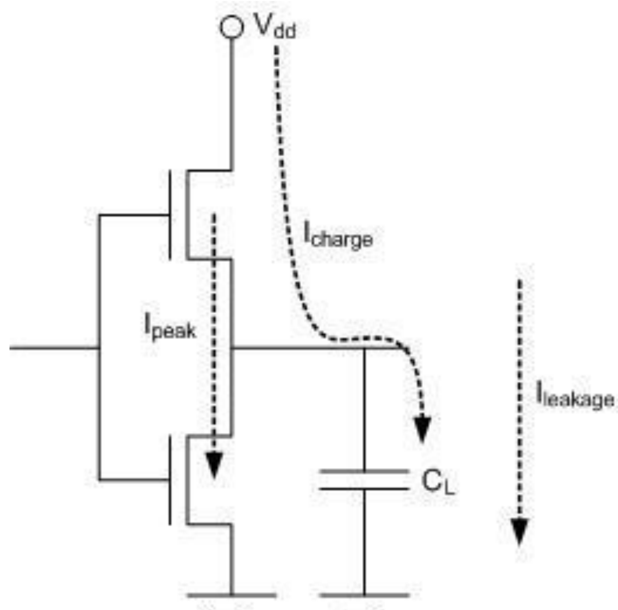
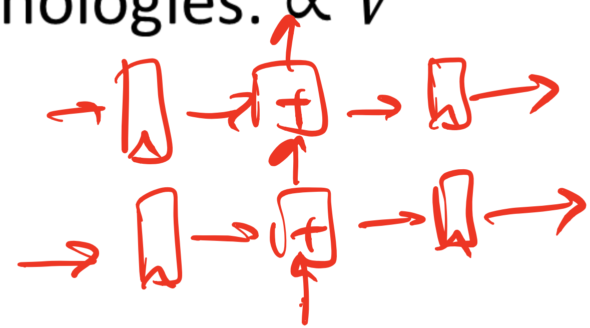


Fig: https://software.intel.com/en-us/blogs/2009/08/25/why-p-scales-as-cv2f-is-so-obvious-pt-2-2

Power Consumption

- Dynamic power (Watts) depends on switching frequency: $\propto C \cdot V^2 \cdot f$
- Not every node switches every cycle, so we can also model an *activity factor*
- Static or leakage power plays a large role in modern technologies: $\propto V$



Nice note: http://www.ti.com/lit/an/scaa035b/scaa035b.pdf

Energy vs Power

- Can trade off power for performance
- Energy represents true cost of a computation
  - Fast or slow, it takes the same number of Joules
- Must keep power within reasonable range
  - e.g. what battery can source or what you can cool
- Static power is a constant "tax" whenever system is operating

Intro to Optimization

Goal:

To Answer the Question: “Which one is best?”

Problem:

Define “Best.”

Example: Buying a Car

https://insidesmallbusiness.com.au/planning-management/bonuses-a-good-ideal

Defining Metrics

What did I care about?

- Cost (in Dollars)
- Fuel Efficiency (in Miles per Gallon)
- Cargo Capacity

My search:

- Look for SUVs
- Looked at Toyota, Subaru, and Hyundai

Raw Data

Make	Model	MPG	Dollars
Toyota	RAV4	35	$25,950.00
Toyota	RAV4	41	$28,350.00
Toyota	4Runner	19	$36,120.00
Toyota	Highlander	29	$34,600.00
Toyota	Highlander	36	$38,200.00
Toyota	Sequoia	17	$49,980.00
Toyota	Sienna	27	$31,640.00
Subaru	Crosstrek	33	$22,245.00
Subaru	Crosstrek	34	$26,495.00
Subaru	Forester	33	$24,795.00
Subaru	Outback	33	$26,795.00
Subaru	Outback	30	$35,145.00
Subaru	Ascent	27	$32,295.00
Subaru	Ascent	26	$39,595.00
Hyundai	Venue	35	$17,350.00
Hyundai	Kona	33	$20,400.00
Hyundai	Tuscon	28	$23,700.00
Hyundai	SantaFe	29	$26,275.00
Hyundai	Palisade	26	$32,525.00

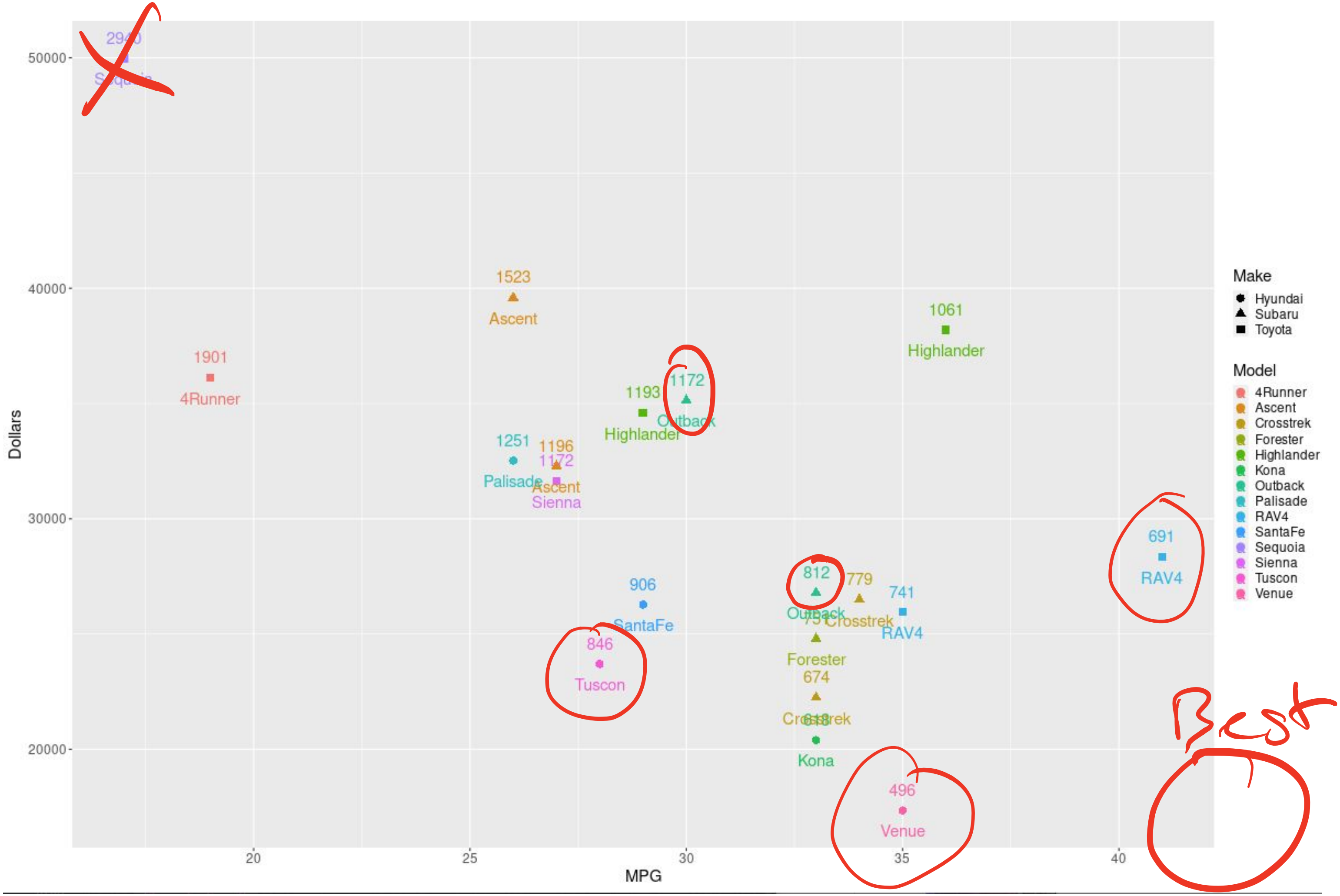
Raw Data

Make	Model	MPG	Dollars	Dollars/MPG
Toyota	RAV4	35	$25,950.00	$741.43
Toyota	RAV4	41	$28,350.00	$691.46
Toyota	4Runner	19	$36,120.00	$1,901.05
Toyota	Highlander	29	$34,600.00	$1,193.10
Toyota	Highlander	36	$38,200.00	$1,061.11
Toyota	Sequoia	17	$49,980.00	$2,940.00
Toyota	Sienna	27	$31,640.00	$1,171.85
Subaru	Crosstrek	33	$22,245.00	$674.09
Subaru	Crosstrek	34	$26,495.00	$779.26
Subaru	Forester	33	$24,795.00	$751.36
Subaru	Outback	33	$26,795.00	$811.97
Subaru	Outback	30	$35,145.00	$1,171.50
Subaru	Ascent	27	$32,295.00	$1,196.11
Subaru	Ascent	26	$39,595.00	$1,522.88
Hyundai	Venue	35	$17,350.00	$495.71
Hyundai	Kona	33	$20,400.00	$618.18
Hyundai	Tuscon	28	$23,700.00	$846.43
Hyundai	SantaFe	29	$26,275.00	$906.03
Hyundai	Palisade	26	$32,525.00	$1,250.96

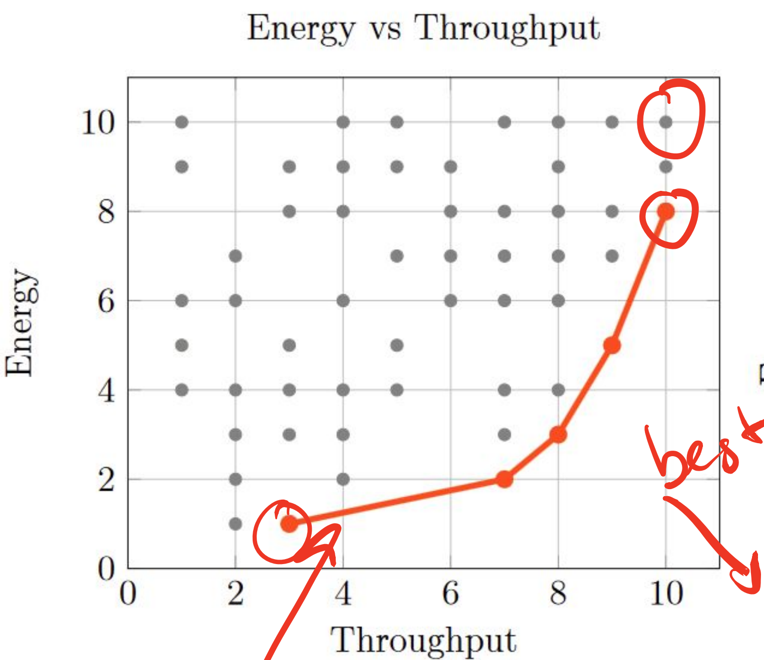
Synthetic Metrics Hide Data

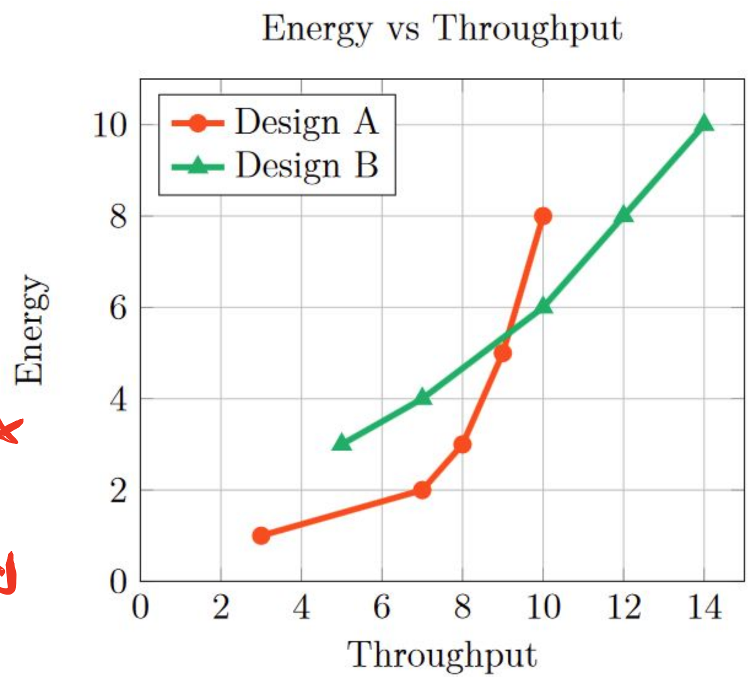
- Dollars per MPG
  - How many MPG does it actually get?
  - What’s the actual COST of the car?!
- Your Grades
  - What did you have trouble with?
  - How can we help you better?

Scatter Plot



Pareto Optimality





calculation ls

Pareto Front

Evaluating Designs

Optimization Space

of dimensions = # of metrics of interest

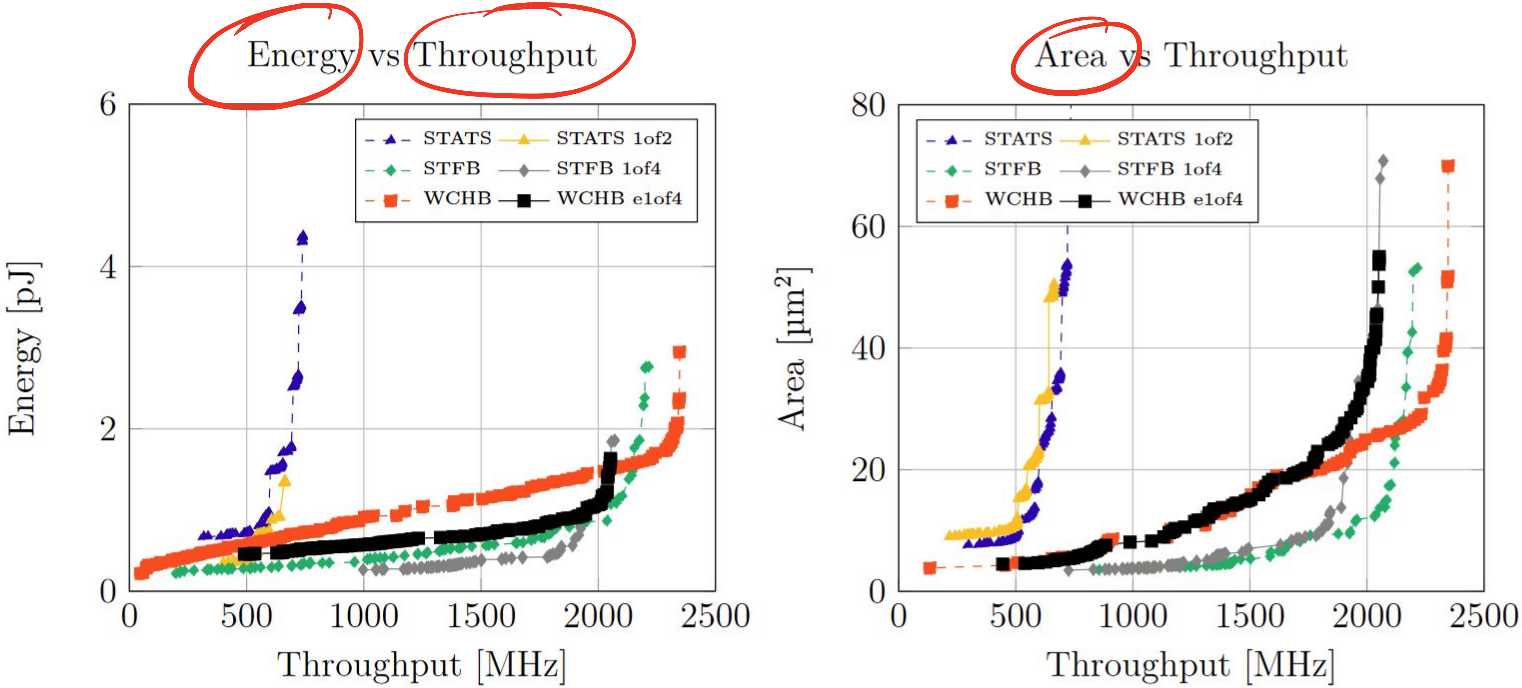


Figure 8.5: 65 nm 600 µm 2-Bit Link

Sweeping Optimization Space

Exhaustive Search Intractable

Therefore, must “sample” space

… and give up on global optimum

… but *where* to sample?

Basic Optimization Flow

1. Define a “Design Instance” or “Sample”

2. Define a “Cost Function” AND a way to evaluate it

3. Try a bunch of samples

4. Do something smart to search the space and try new samples

Heuristic Optimization

The “smart” thing I referred to earlier.

Example Algorithms (There are Many)

Genetic Algorithms

Simulated Annealing

Tabu Search

Genetic Algorithms (Car)

Vector of Design Choices

- Engine
- Gearbox
- Chassis

Cost Function

- Calculate cost to build
- Test MPG

Back to Computer Architecture