CS250B: Modern Computer Systems

The Past, Present, and Future of Specialized Accelerators

Sang-Woo Jun

Contents

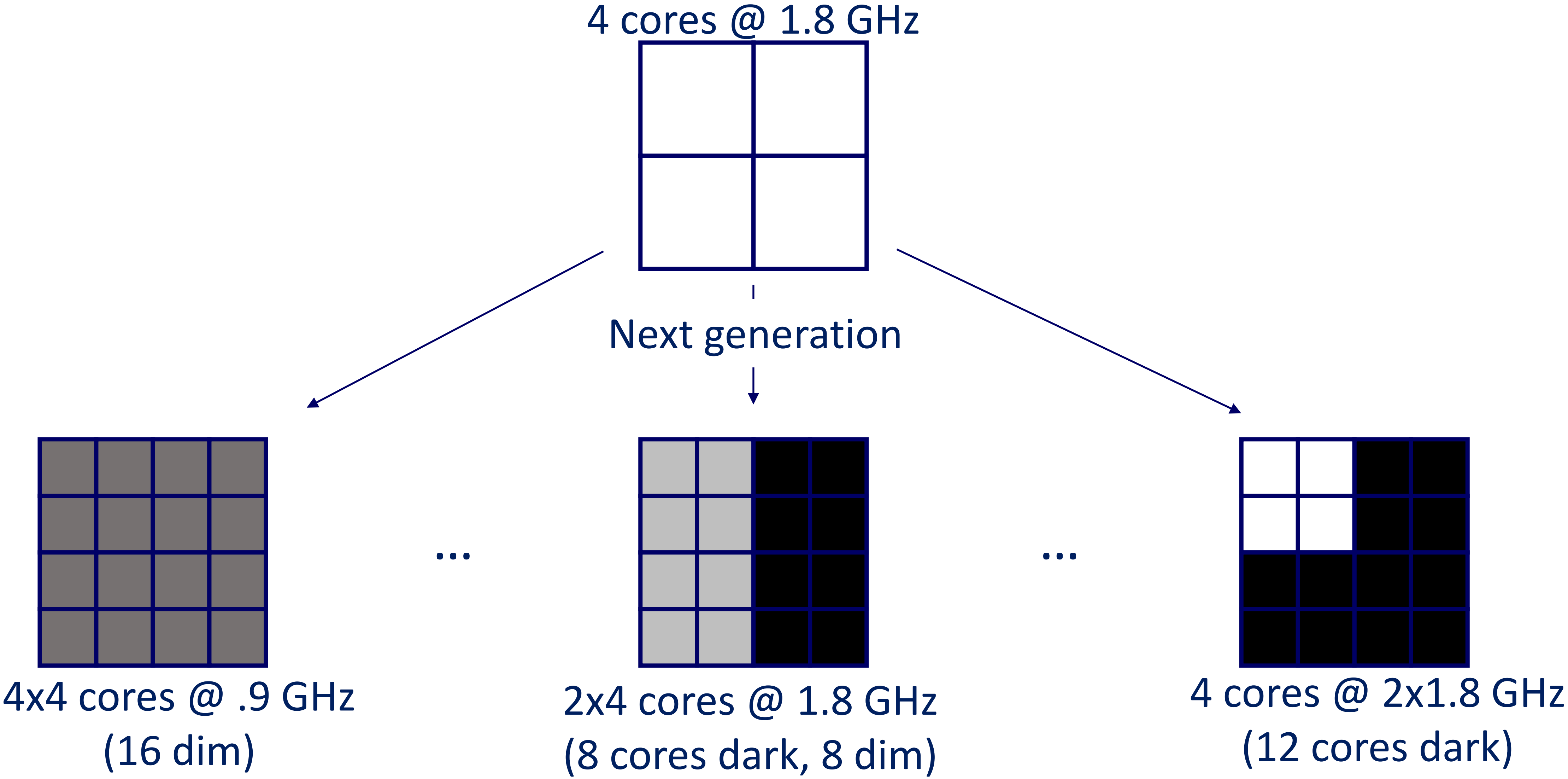
- ❑ We will briefly go through three papers
- ❑ The Past: “Why specialized accelerators?”
  - ○ Taylor, Michael B. "Is dark silicon useful? Harnessing the four horsemen of the coming dark silicon apocalypse." DAC Design Automation Conference 2012. IEEE, 2012.
- ❑ The Present: “Where does improvements come from?”
  - ○ Hameed, Rehan, et al. "Understanding sources of inefficiency in general-purpose chips." Proceedings of the 37th annual international symposium on Computer architecture. 2010.
- ❑ The Future: “How long can this last?”
  - ○ Fuchs, Adi, and David Wentzlaff. "The accelerator wall: Limits of chip specialization." 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2019.

The Past:
Why Specialized Accelerators?

- ❑ Despite continued transistor scaling, not all of them can be useful
  - o Power consumption no longer scales with transistor size
  - o “Utilization wall”: “With each successive process generation, the percentage of a chip that can switch at full frequency drops exponentially due to power constraints.” -- Venkatesh, ASPLOS ‘10

- ❑ The following slides adapted from Michael Taylor’s 2012 talk
  “Is Dark Silicon Useful? Harnessing the Four Horsemen of the Coming Dark Silicon Apocalypse”

Tradeoffs Between Cores And Frequency



4x4 cores @ .9 GHz
(16 dim)

2x4 cores @ 1.8 GHz
(8 cores dark, 8 dim)

4 cores @ 2x1.8 GHz
(12 cores dark)

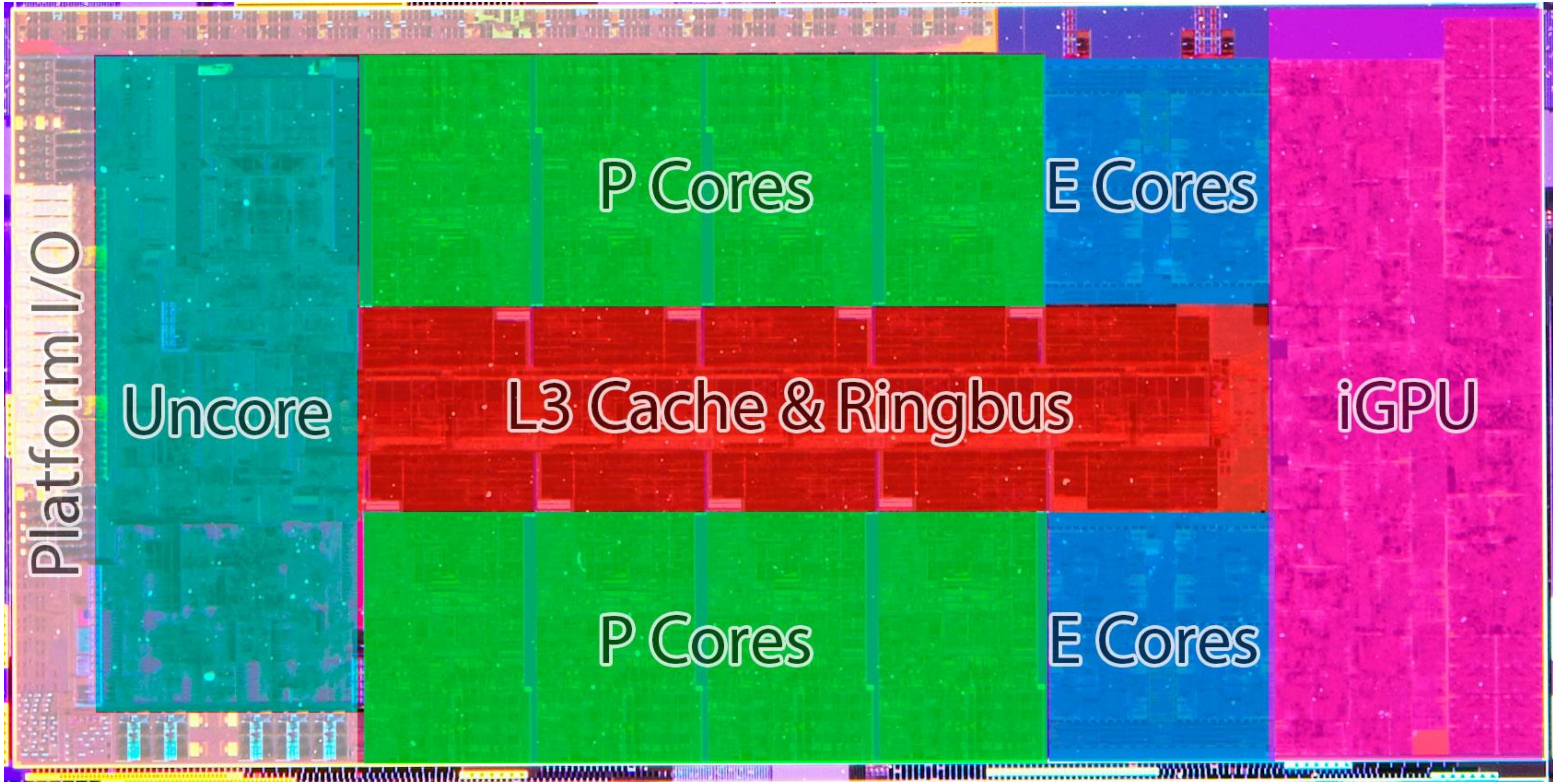
The Four Horsemen

- ❑ What do we do with this dark silicon?
- ❑ The paper/talk presents four potential directions
  - None are ideal solutions, but each has its benefits
  - Optimal solution probably incorporates all four of them

The Shrinking Horseman (#1)

- “Area is expensive. Chip designers will just build smaller chips instead of having dark silicon in their designs!”
- First, dark silicon doesn’t mean useless silicon, it just means it’s under-clocked or not used all of the time.
- There’s lots of dark silicon in current chips:
  - On-chip GPU on recent x86 chips, when running GCC or web server
  - L3 cache is very dark for applications with small working sets
  - SIMD units for integer apps
  - …

Example: Intel Alder Lake (7 nm)

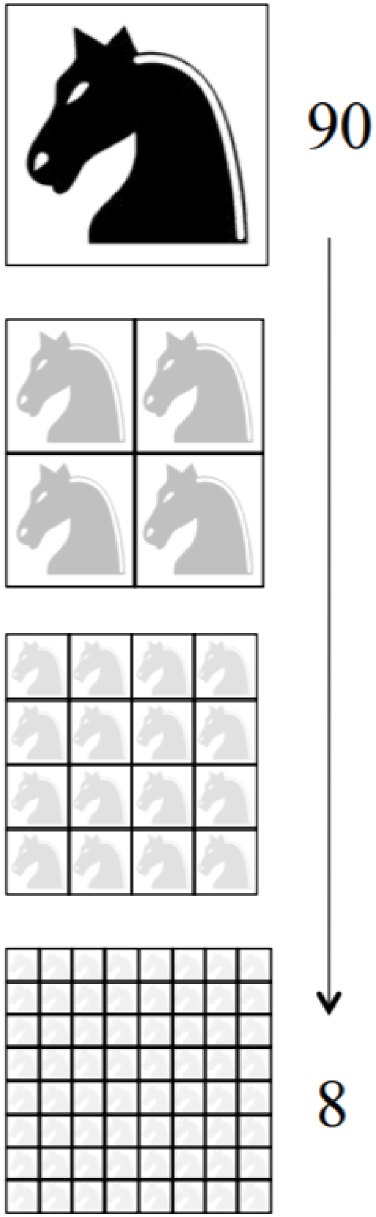


https://www.techpowerup.com/review/intel-core-i9-12900k-alder-lake-12th-gen/2.html

The Shrinking Horseman (#1)

- ❑ Competition and Margins
  - o If there is an advantage to be had from using dark silicon, you have to use it too, to keep up with competitors
- ❑ Diminished Returns
  - o Savings Exponentially Diminishing with smaller chips
  - o Overheads: packaging, test, marketing, etc.
  - o Chip structures like I/O Pad Area do not scale
- ❑ Exponential increase in Power Density -> Exponential Rise in Temperature
- ❑ But, some chips will shrink
  - o Low margin, high competition chips; …

The Dim Horseman (#2)



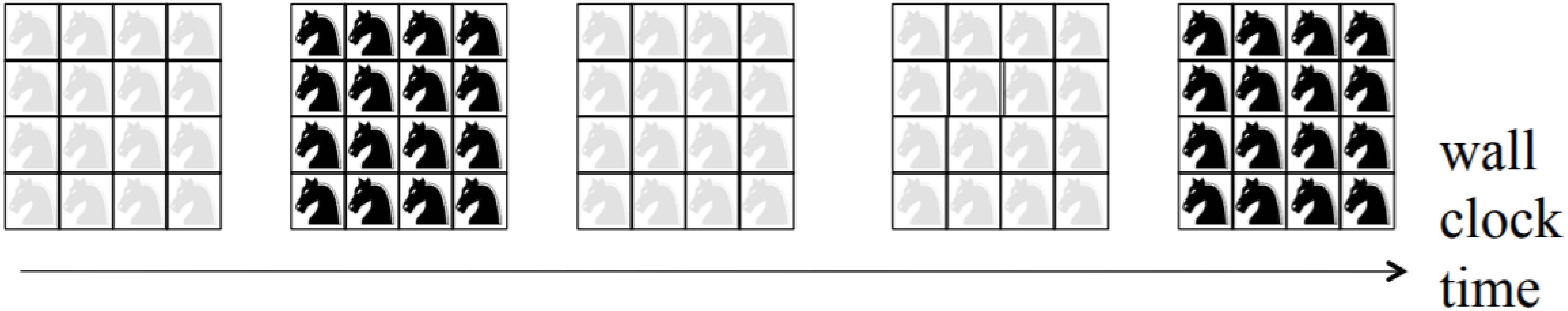
❑ Spatial dimming: Have enough cores to exceed power budget, but underclock them

❑ Gen 1 & 2 Multicores (higher core count, lower freqs)

❑ Near Threshold Voltage (NTV) Operation
  - Lower voltage -> Slower clock -> Performance loss
  - But, make it up with lots of dim cores
  - Watch for Non-Ideal Speedups / Amdahl’s Law

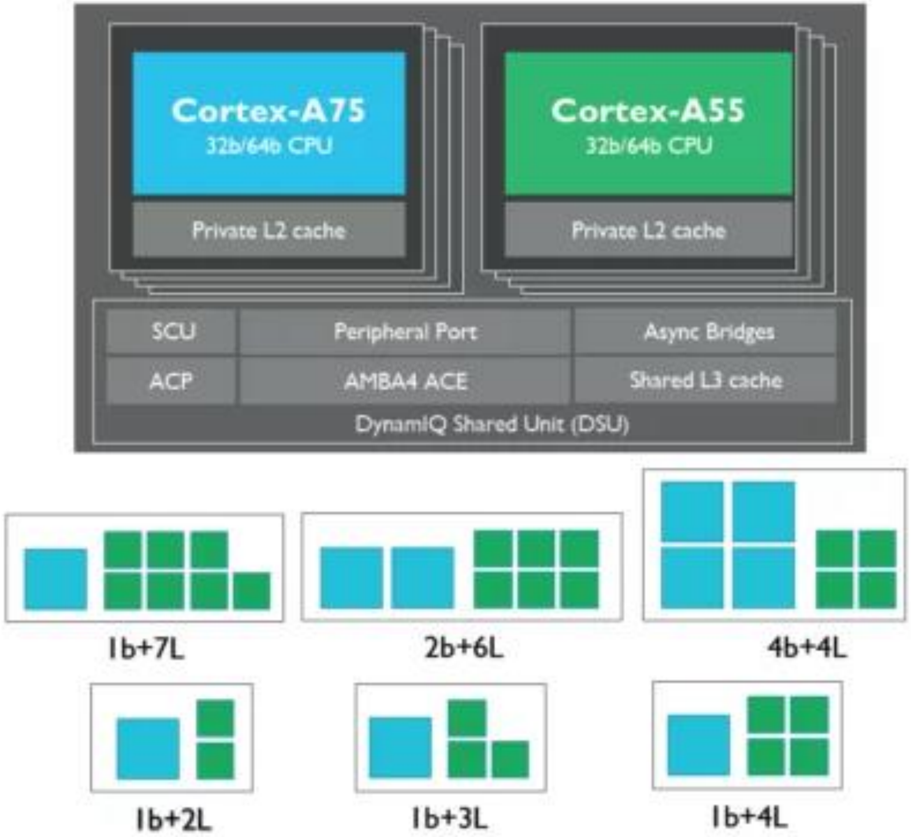
The Dim Horseman (#2)

- ❑ Temporal Dimming : Have enough cores to exceed power budget, but use them only in bursts
  - o Dim cores, but overclock if cold – e.g., Intel TurboBoost
  - o E.g., ARM Cortex-A75 for mobile phones
    - A75 power usage not sustainable for phone. (Battery, heat!)
    - 10 second bursts at most (big.LITTLE with DynamIQ, Intel E- and P-cores)



Aside: ARM big.LITTLE

- SoC has multiple compatible cores
  - Same ISA
  - Different performance, power efficiency
- OS transparently migrates threads
  - More speed?
  - Less power?
- Multiple pairs of core designs
  - Cortex A7 vs. A12/A15/A17
  - Cortex A55 vs. A75



Example DynamIQ big.LITTLE configurations

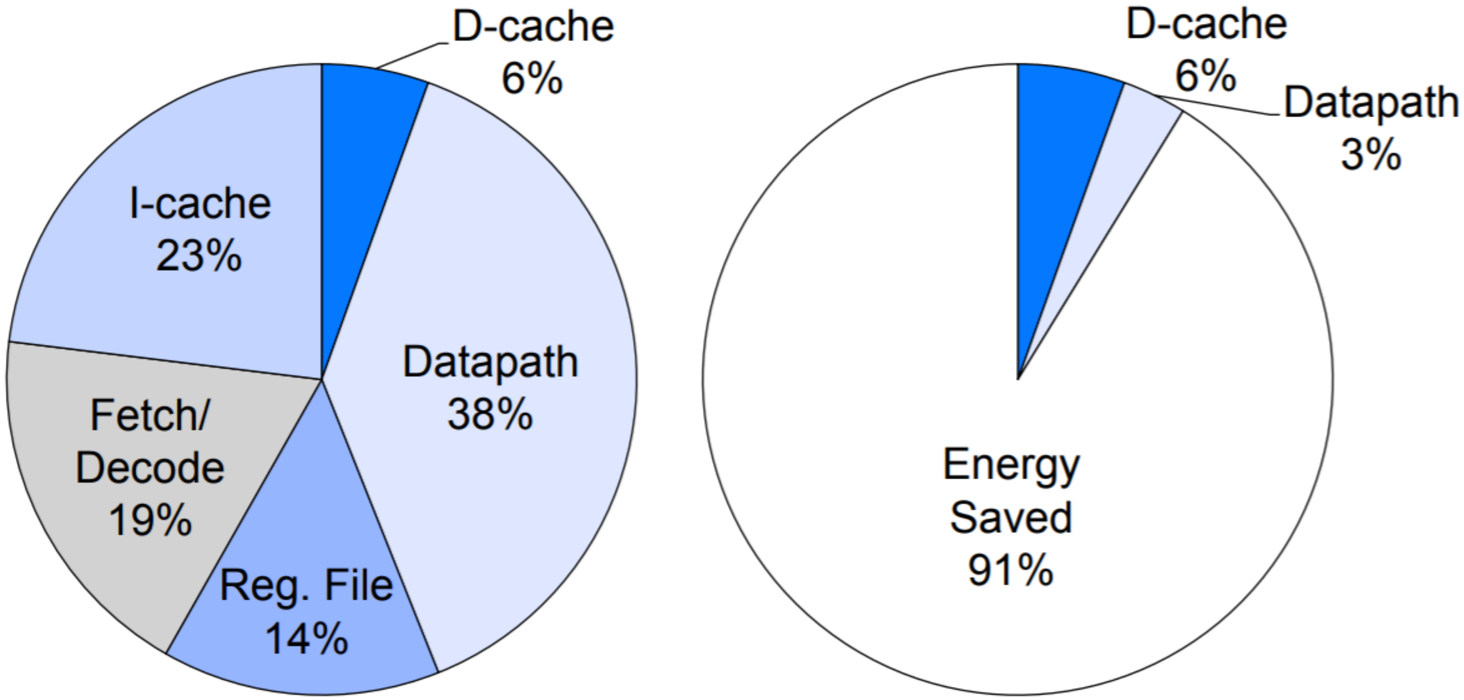
The Specialized Horseman (#3)

- ❑ “We will use all of that dark silicon area to build specialized cores, each of them tuned for the task at hand (10-100x more energy efficient), and only turn on the ones we need…”
- ❑ Insights:
  - Power is now more expensive than area
  - Specialized logic can improve energy efficiency by 10-1000x

The Specialized Horseman (#3)

- C-cores Approach:
  - Fill dark silicon with Conservation Cores, or c-cores, which are automatically-generated, specialized energy-saving coprocessors that save energy on common apps
- Execution jumps among c-cores (hot code) and a host CPU (cold code)
  - Power-gate HW that is not currently in use
    - As if they're not there!
  - Coherent Memory & Patching Support for C-cores

Typical Energy Savings



RISC baseline
91 pJ/instr.

← ~11x →

C-cores
8 pJ/instr.

The Deus Ex Machina Horseman (#4)

❑ Deus Ex Machina: “A plot device whereby a seemingly unsolvable problem is suddenly and abruptly solved with the unexpected intervention of some new event, character, ability or object.”

❑ “MOSFETs are the fundamental problem”

❑ “FinFets, Trigate, High-K, nanotubes, 3D, for one-time improvements, but none are sustainable solutions across process generations.”

The Deus Ex Machina Horseman (#4)

- ❑ Possible “Beyond CMOS” Device Directions
  - o Nano-electrical Mechanical Relays?
  - o Tunnel Field Effect Transistors (TFETS)?
  - o Spin-Transfer Torque MRAM (STT-MRAM)?
  - o Graphene?
  - o Quantum computing?
  - o Human brain?
  - o DNA Computing?

The Present: Where Does Improvements Come From?

❑ How “specialized” must specialized accelerators be, to achieve high performance and power efficiency?

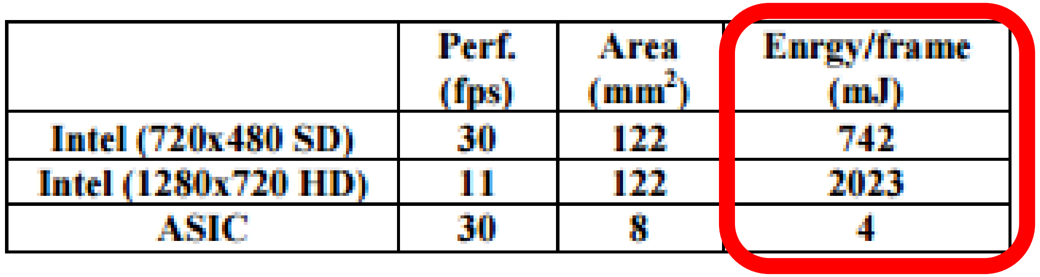
- There is a trade-off between general-purpose and application-specific
- Is there a sweet spot? Still software-programmable, but high performance/efficiency?

❑ The following slides adapted from Hameed Rehan et. al., “Understanding sources of inefficiency in general-purpose chips,” ISCA 2010

Exploring
Chip Multiprocessors (CMP) vs ASIC gap

- Example application: H.264 encoding (MPEG-4 advanced video coding)
    - Large CMP vs. ASIC gap to explore

- Authors compare ASIC implementation against software
    - General purpose processor modified in steps until it becomes ASIC
    - What are the improvements at each stage?



	Perf. (fps)	Area (mm^2)	Enrgy/frame (mJ)
Intel (720x480 SD)	30	122	742
Intel (1280x720 HD)	11	122	2023
ASIC	30	8	4

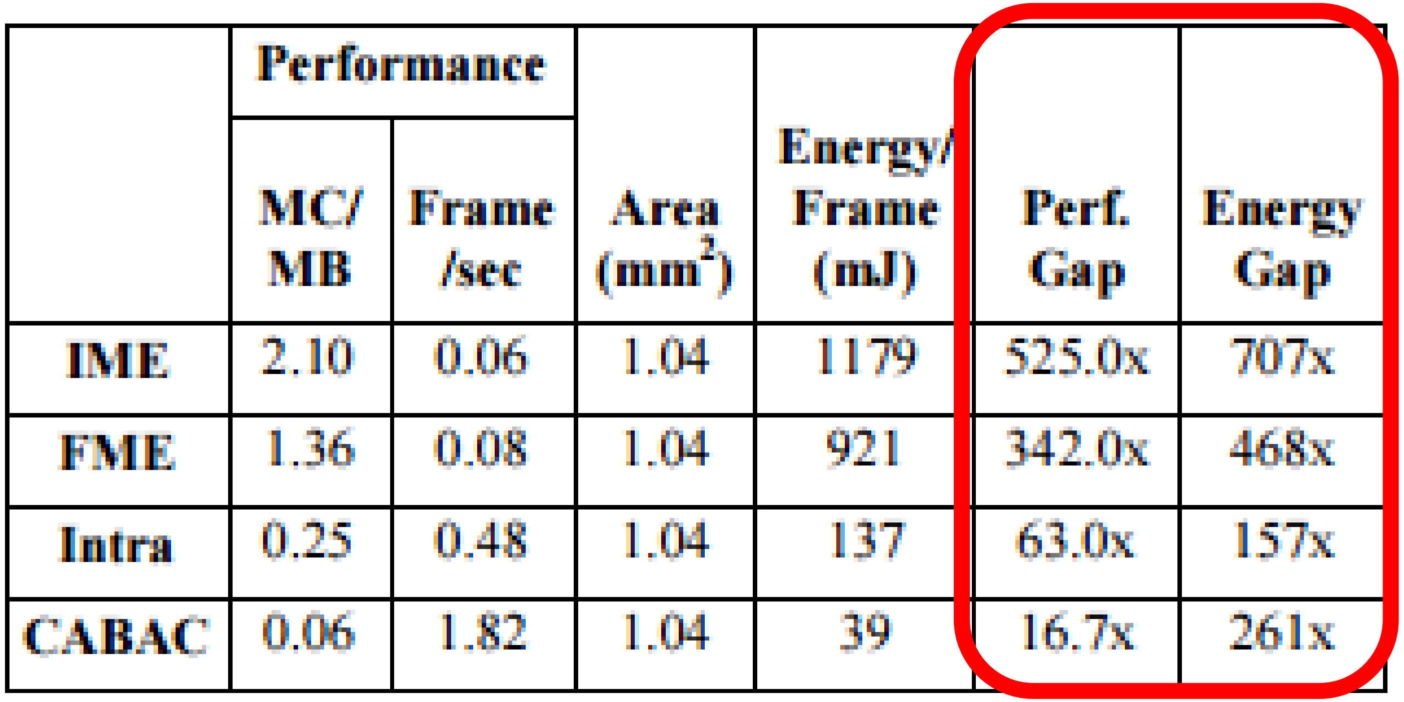
150-500x power gap

Some H.264 Internals

- ❑ Most computation divided into four steps
  - **IME:** Integer Motion Estimation
    - Computes vector of image-block motion
  - **FME:** Fractional Motion Estimation
    - Refines initial match to quarter-pixel resolution
  - **Intra:** Intra Prediction + Transform and Quantization
    - Based on surrounding image-blocks, makes prediction
  - **CABAC:** Context Adaptive Binary Arithmetic Coding
    - Encodes bits
- ❑ Individual steps not important for us right now

General-Purpose Processor Power Breakdown

- ❑ Large performance gap, but even larger energy gap
  - ○ From higher efficiency of ASICs

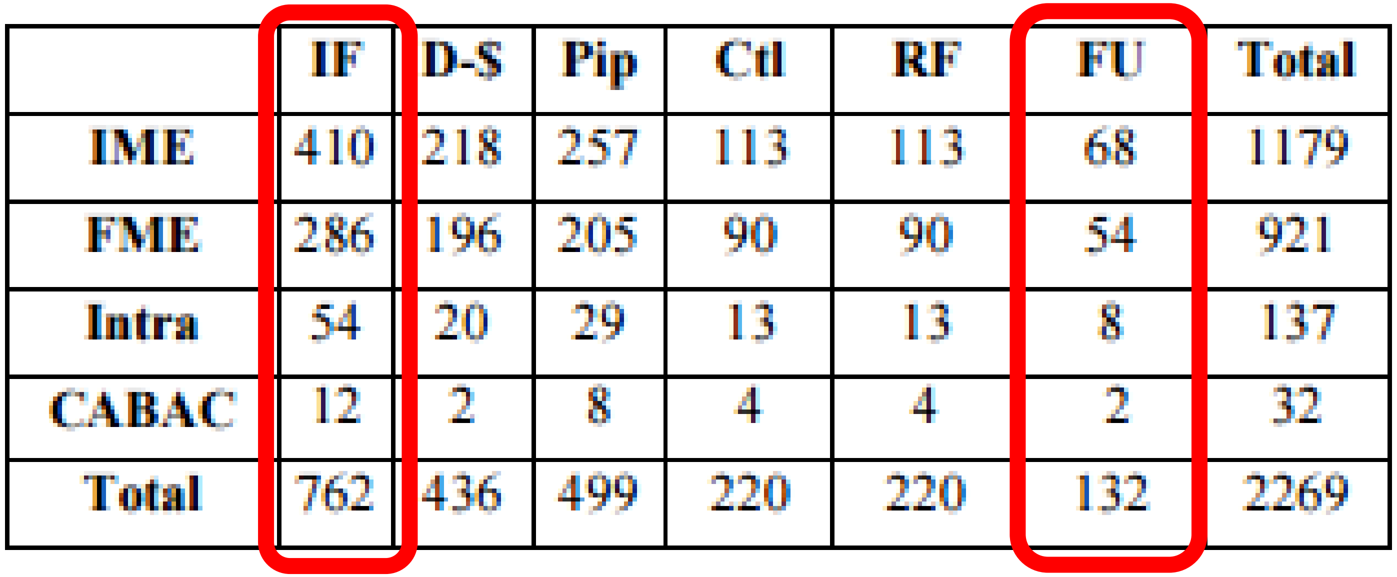


	Performance					
	MC/ MB	Frame /sec	Area (mm^2)	Energy/ Frame (mJ)	Perf. Gap	Energy Gap
IME	2.10	0.06	1.04	1179	525.0x	707x
FME	1.36	0.08	1.04	921	342.0x	468x
Intra	0.25	0.48	1.04	137	63.0x	157x
CABAC	0.06	1.82	1.04	39	16.7x	261x

Can we close this gap?

General-Purpose Processor Energy Breakdown

- ❑ Energy breakdown in mJ/frame
  - ○ Functional units (FU) responsible for only ~6%!
  - ○ IF (Instruction fetch + decode + Instruction cache) responsible for ~30%



	IF	D-$	Pip	Ctl	RF	FU	Total
IME	410	218	257	113	113	68	1179
FME	286	196	205	90	90	54	921
Intra	54	20	29	13	13	8	137
CABAC	12	2	8	4	4	2	32
Total	762	436	499	220	220	132	2269

Three Steps of Customization

- ❑ SIMD + VLIW
  - o Improves ratio of computation to instruction fetch/decoding
  - o Relatively general solution
- ❑ Specialized instructions
  - o New instructions, still following the ISA operand structure
  - o Two source operands, one destination operand
- ❑ Unrestricted ISA modification
  - o Instructions no longer restricted by ISA operand structure
  - o New register files, complex computation units
  - o But still invoked by “instructions”, generated by compiler

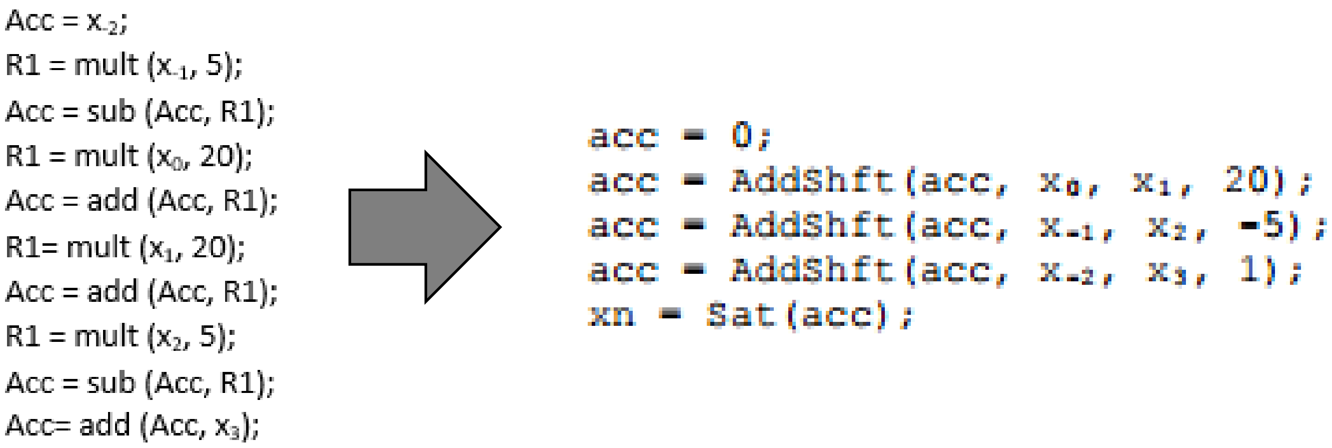
Customization #1: SIMD+VLIW

- SIMD: Reduce the ratio of instruction fetch + decode energy
  - Very wide, 16 and 18-way SIMD datapaths
- VLIW: Execute many instructions in parallel
  - 2 and 3-slot VLIW instructions

- Improves performance and power efficiency
  - 10x performance, 1/10 energy
  - While energy share of functional units increased, it is still very small
  - IF still consumes ~30%

Customization #2: Operation Fusion

- ❑ Application specific instructions, still following ISA structure
  - ○ New instructions for common operations in application
    - Fusing many basic instructions into one
  - ○ More functional units if each fused function uses many basic units
  - ○ Reduces register file access by creating separate registers between pipeline stages
- ❑ Further benefit: Compilers can take advantage automatically

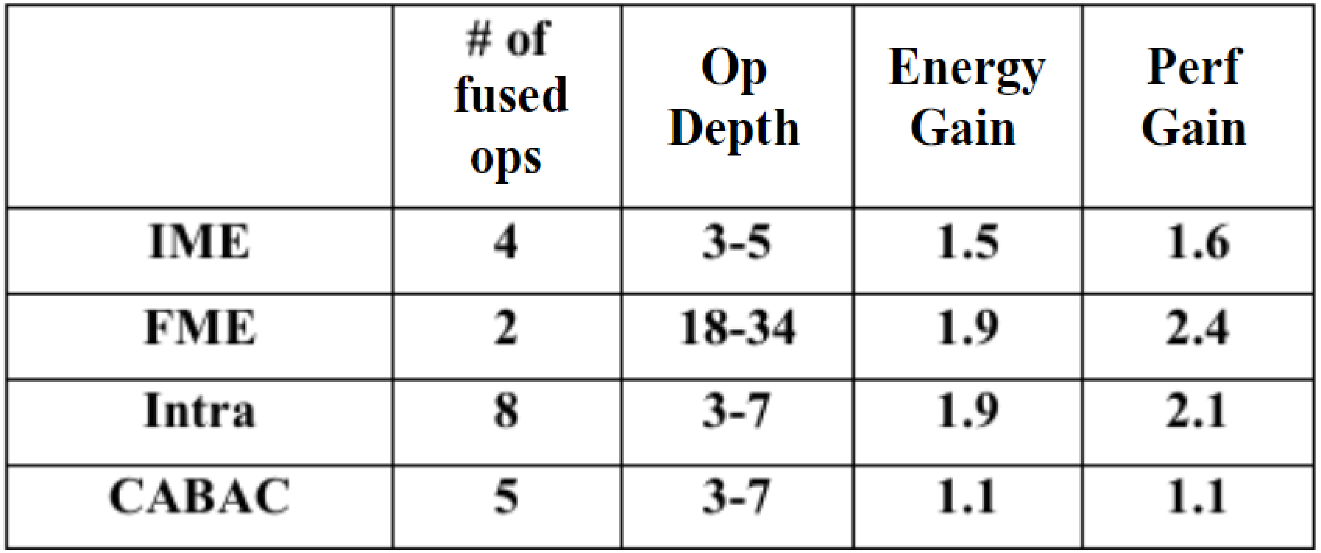


```
Acc = x_{-2};
R1 = mult (x_{-1}, 5);
Acc = sub (Acc, R1);
R1 = mult (x_{0}, 20);
Acc = add (Acc, R1);
R1= mult (x_{1}, 20);
Acc = add (Acc, R1);
R1 = mult (x_{2}, 5);
Acc = sub (Acc, R1);
Acc= add (Acc, x_{3});
```

```
acc = 0;
acc = AddShft(acc, x_{0}, x_{1}, 20);
acc = AddShft(acc, x_{-1}, x_{2}, -5);
acc = AddShft(acc, x_{-2}, x_{3}, 1);
xn = Sat(acc);
```

Customization #2: Operation Fusion

- ❑ Around 2x performance/energy gains at best
  - Despite high number of fused operations
  - Why? Basic operations are still simple



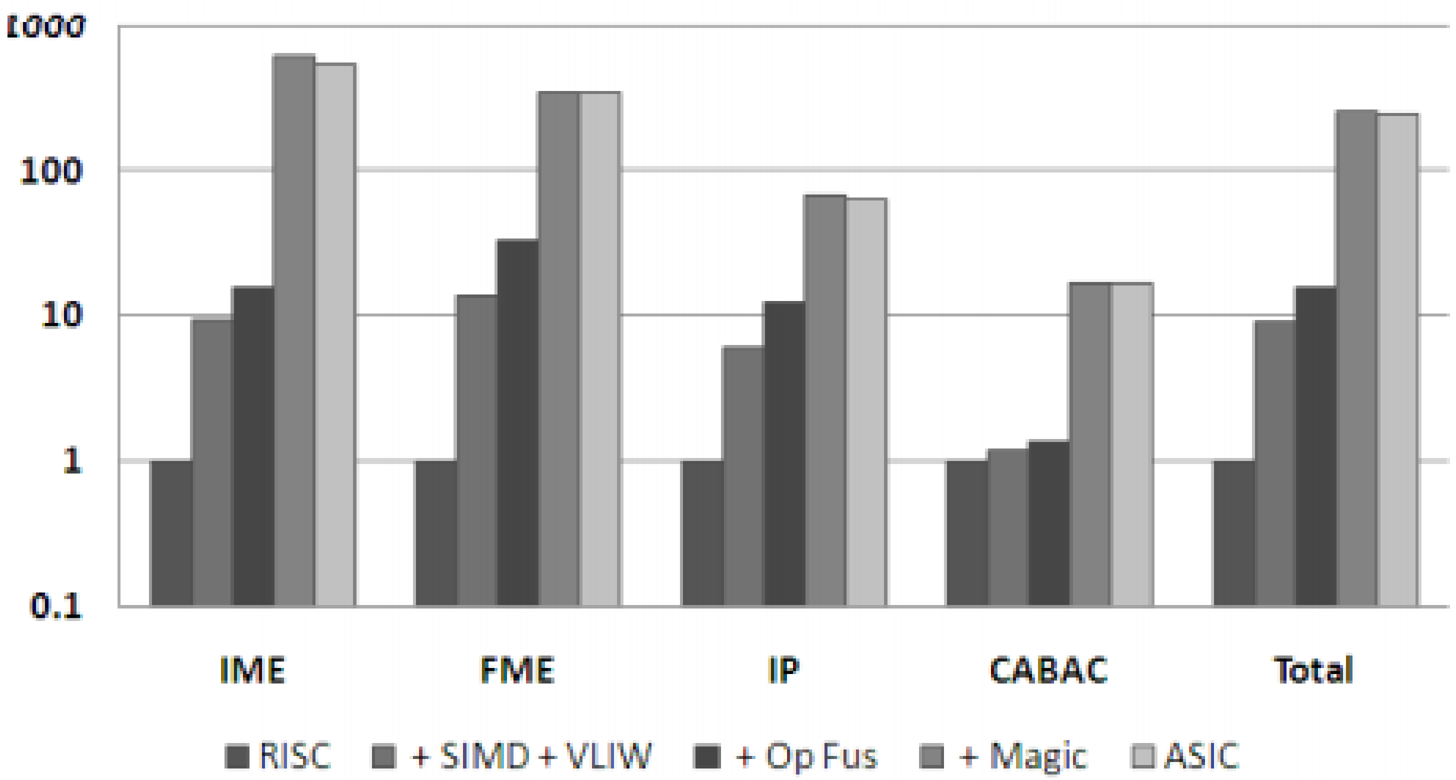
	# of fused ops	Op Depth	Energy Gain	Perf Gain
IME	4	3-5	1.5	1.6
FME	2	18-34	1.9	2.4
Intra	8	3-7	1.9	2.1
CABAC	5	3-7	1.1	1.1

Customization #3: Unrestricted ISA Modification

- “Magic” instruction
  - Single instruction to perform 100s of operations
  - Custom memory resources, which magic instruction can access without additional instructions
- How is this different from ASICs?
  - Not much! But…
  - Processor is still in charge of execution control
  - Magic instruction performs a single, (albeit complex) deterministic operation

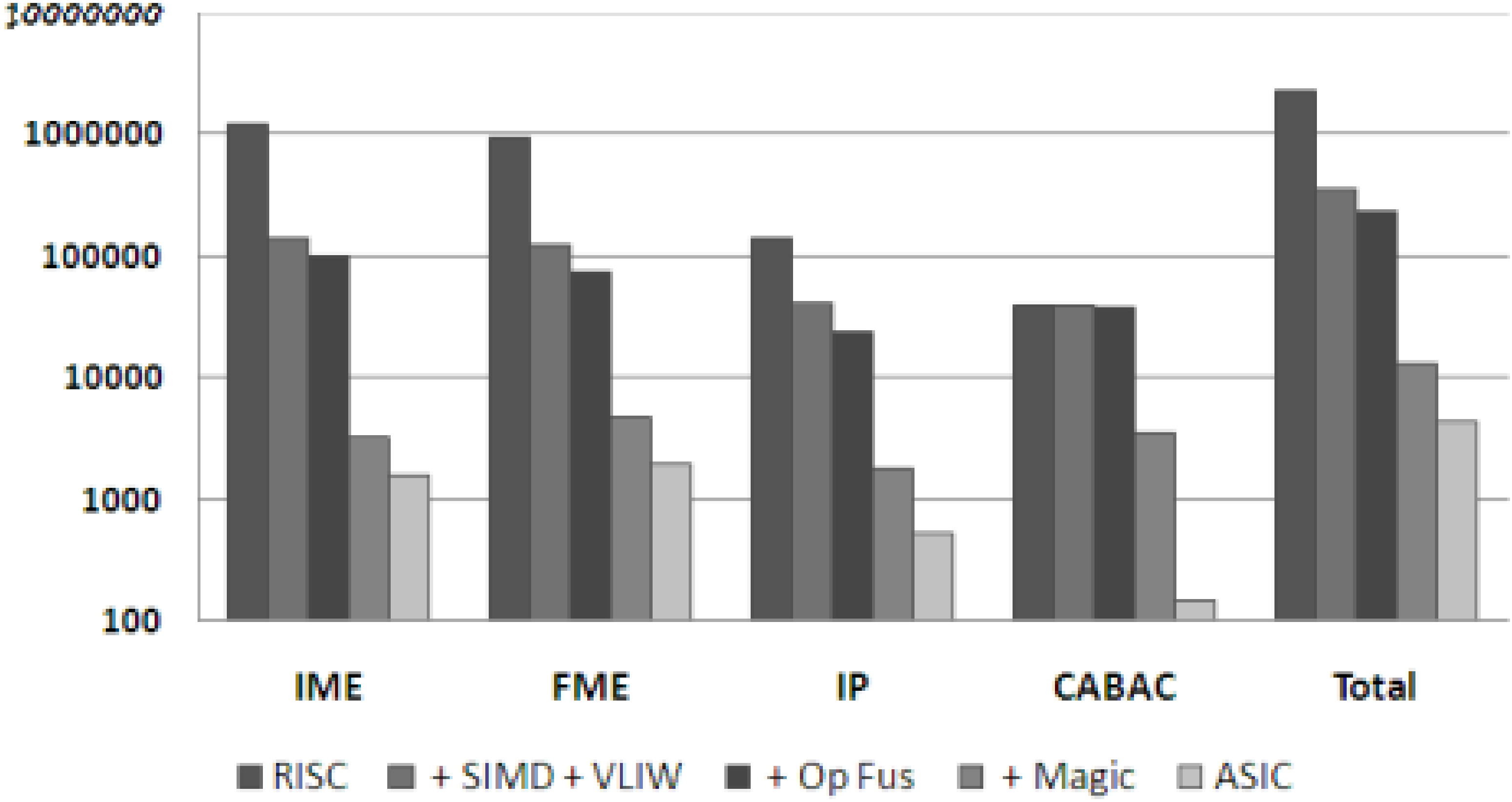
Performance Improvement Breakdown

- Reaches ASIC-level performance only after Magic instructions



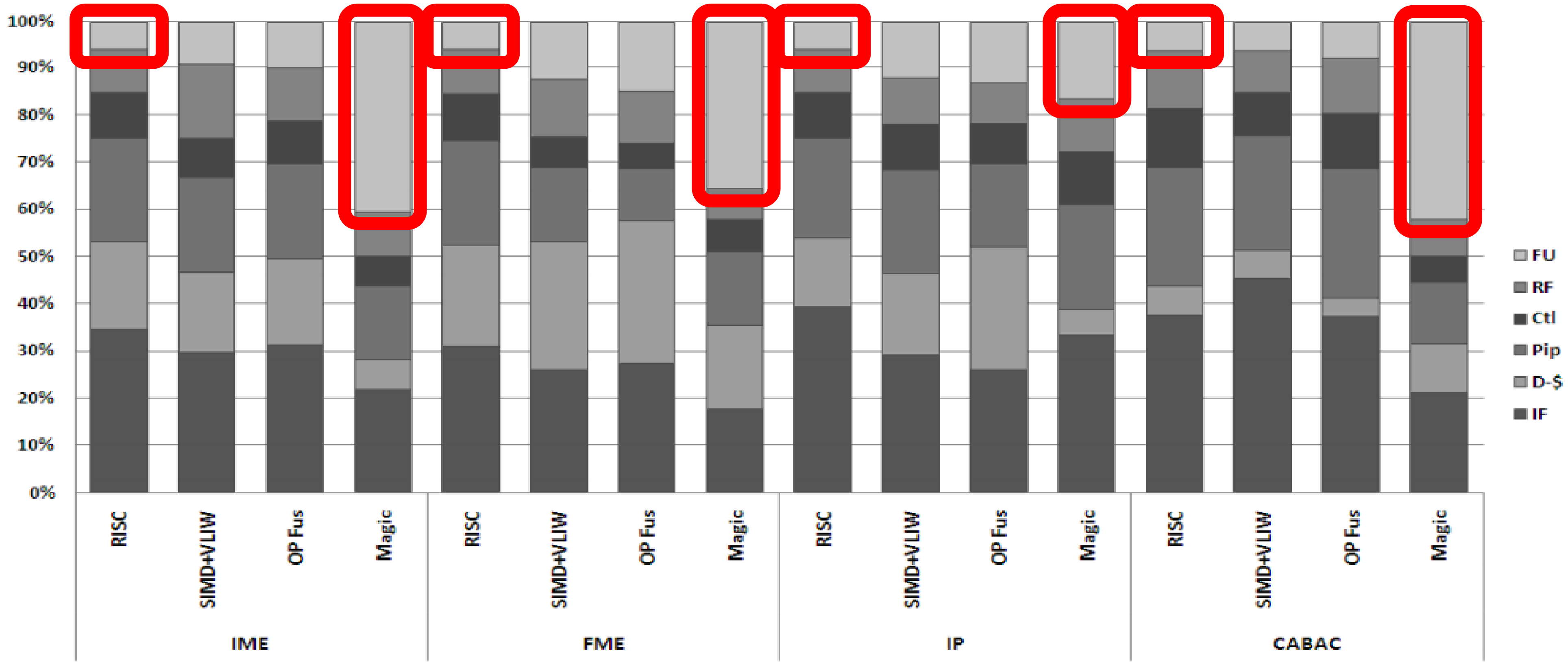
Energy Improvement Breakdown

- ❑ Still significant energy efficiency gap against ASIC!



Energy Improvement Breakdown

- ❑ Functional unit ratio improved drastically, but still not dominant
- ❑ However, energy of FU already exceed total ASIC energy



The Answer:
Where Do Improvements Come From?

- ❑ Performance-wise, application-specific datapath is enough
- ❑ Energy-wise, even control must be optimized to reach ASIC-levels
  - o Instruction fetch/decode is expensive

- ❑ For energy efficiency, even extensible processors are not enough!

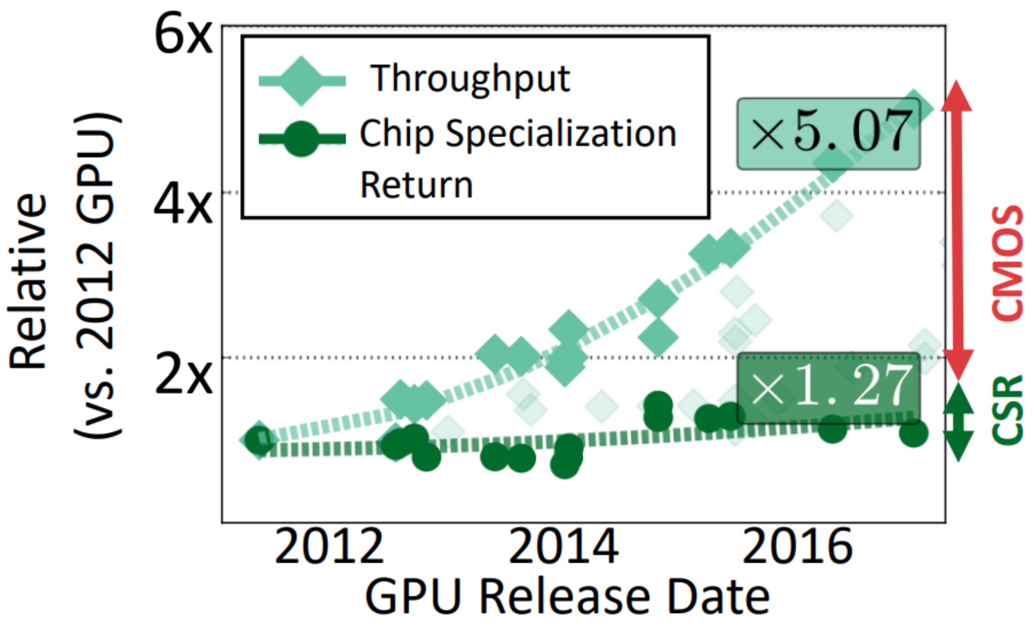
The Future: How Long Can This Last?

- ❑ Accelerators have shown x100+ performance/efficiency
  - o Can accelerators be a solution forever? Is there an end in sight?
- ❑ More specifically, how will the end of Moore’s Law impact accelerators?
  - o General purpose scaling is stopping despite (yet) continuing Moore’s law
  - o So far, accelerators make good use of available silicon
  - o The final CMOS node is predicted to be 5nm. How will accelerators fare?

- ❑ The following slides adapted from Adi Fuchs et. al., “The accelerator wall: Limits of chip specialization,” HPCA 2019

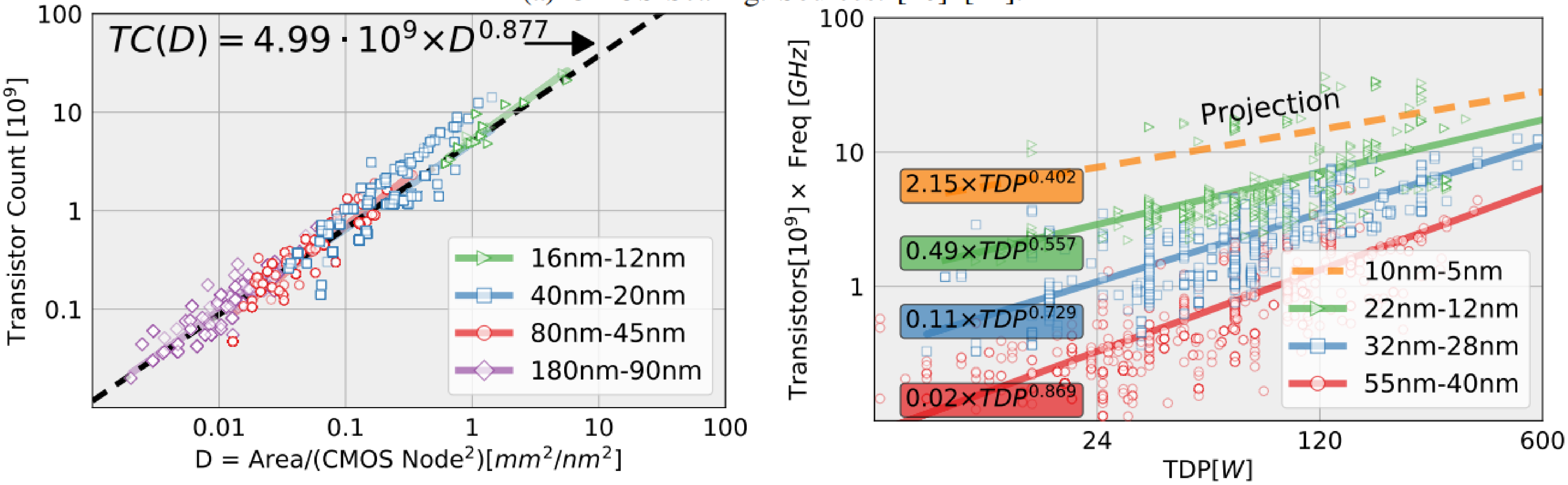
The Big Question

- What part of accelerator benefits come from
  - CMOS technology scaling
  - Accelerator design
- Example: Gaming on GPUs
  - Throughput improvement: 5x
  - CMOS scaling contribution: 4x
  - Improvement via architecture: Only 1.27x
    - "Chip Specialization Return"
- Is this a general trend?



Evaluating The Sources of Accelerator Performance Improvements

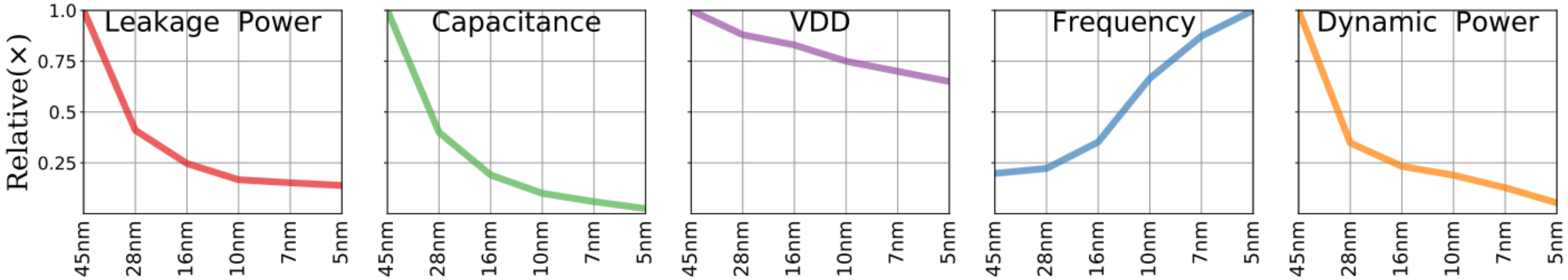
❑ Authors analyzed thousands of existing chips to discover a trend of transistor budget per CMOS node and power envelope



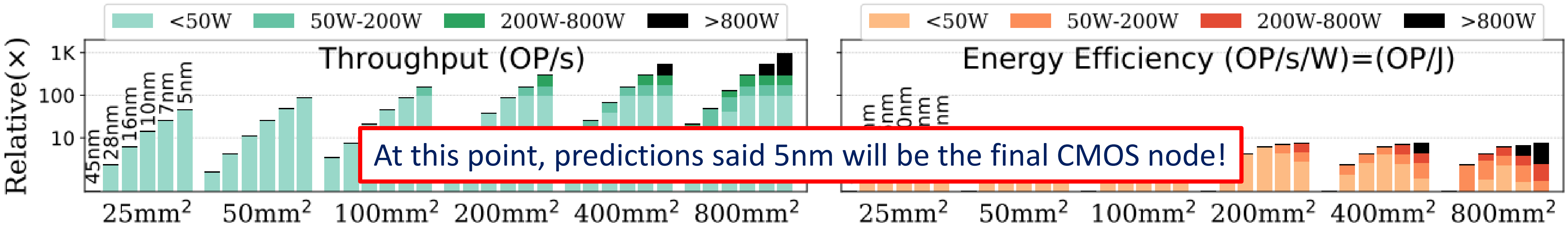
(a) CMOS Scaling. Sources: [20]–[22].

Evaluating The Sources of Accelerator Performance Improvements

- ❑ Then applied it to projected CMOS scaling
  - ○ Sources including International Roadmap for Devices and Systems (IRDS)



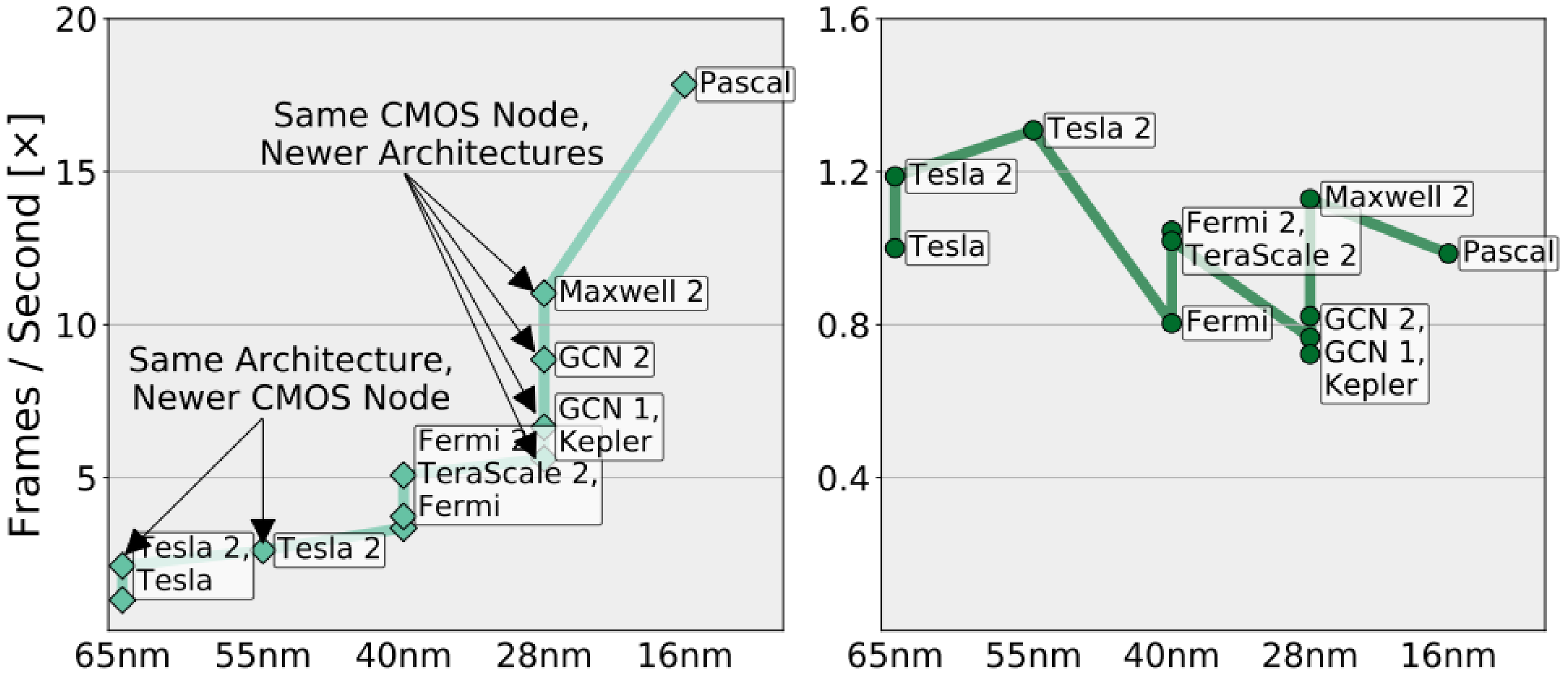
- ❑ To predict upper limit on future performance and energy scaling



At this point, predictions said 5nm will be the final CMOS node!

Application #1: GPU Gaming

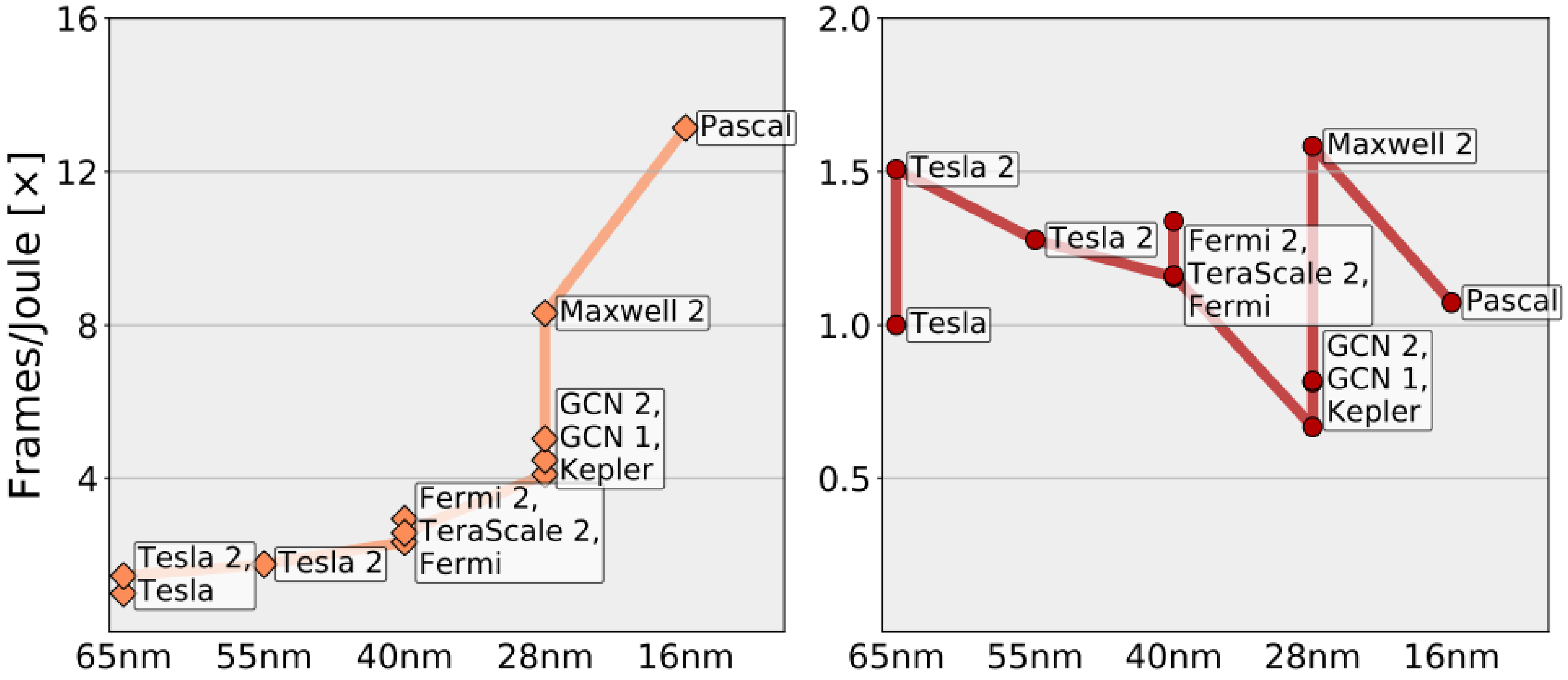
- ❑ Absolute performance has always increased, but chip specialized return is stagnating



(a) Absolute (b) Chip Specialization Return

Application #1: GPU Gaming

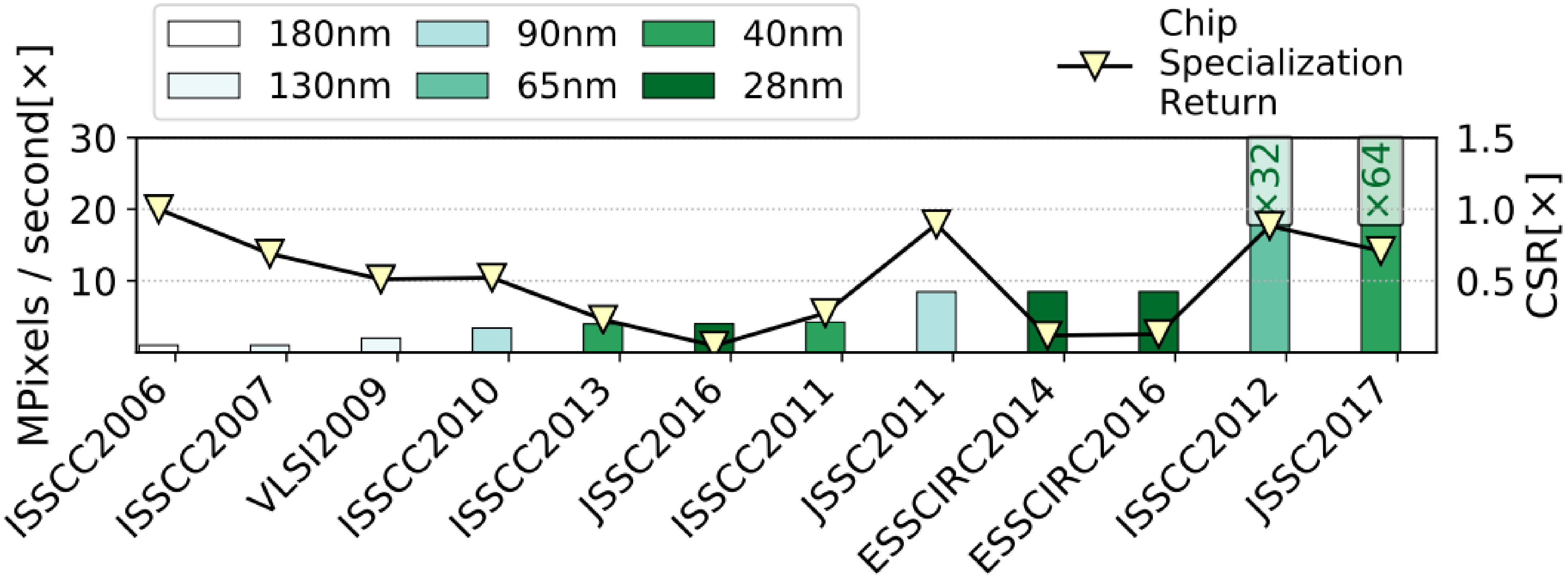
- Same story with power efficiency



(a) Absolute (b) Chip Specialization Return

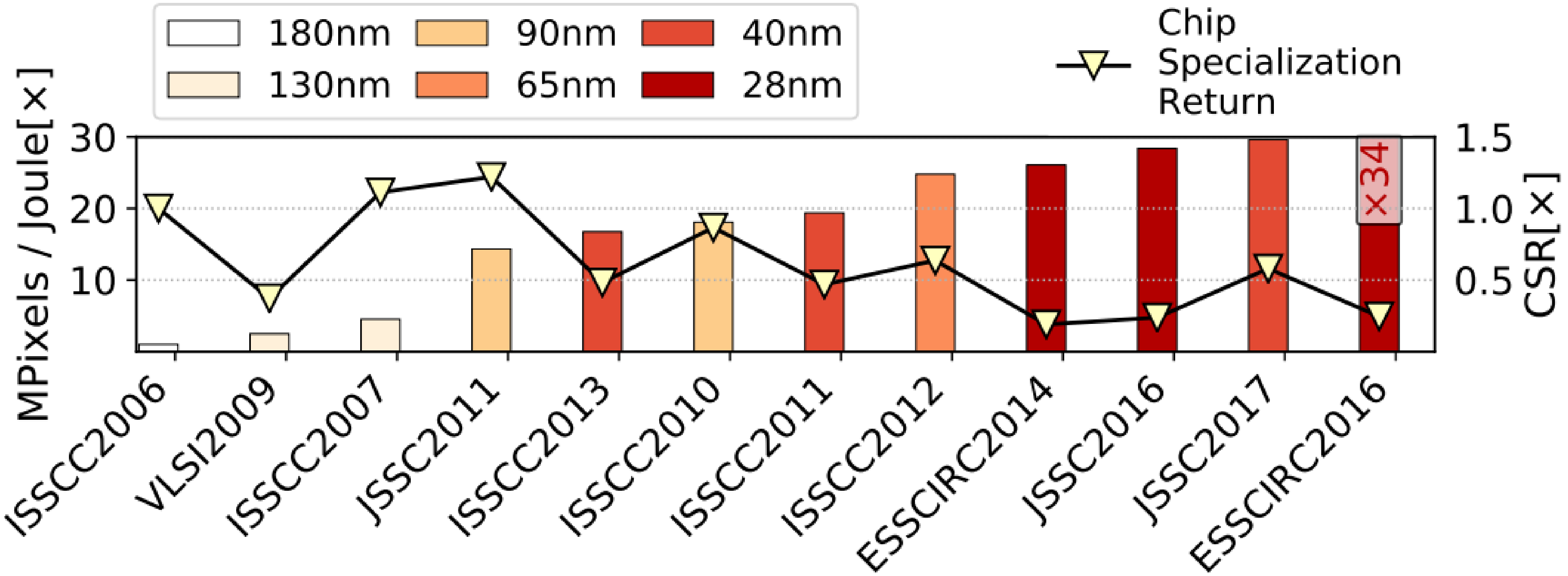
Application #2: Video Decoding

❑ Absolute performance has always increased, but chip specialized return is stagnating



Application #2: Video Decoding

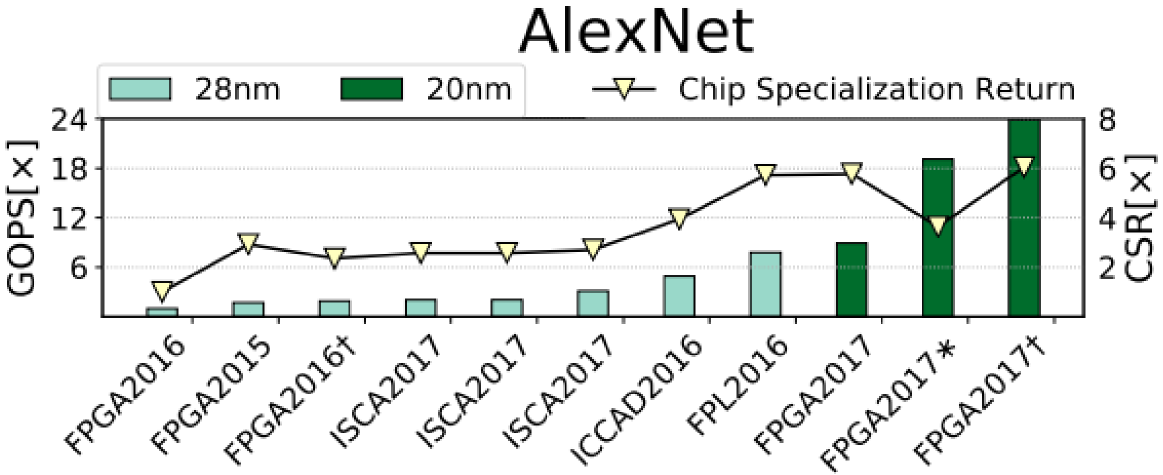
- ❑ Same story with power efficiency

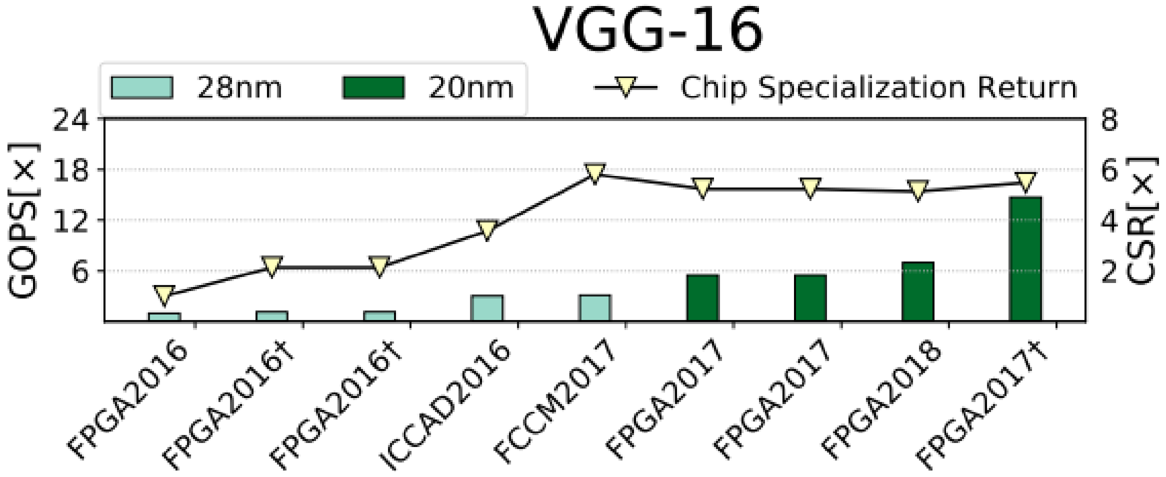


Application #3:
Neural Network Inference on FPGAs

❑ Absolute performance always increasing

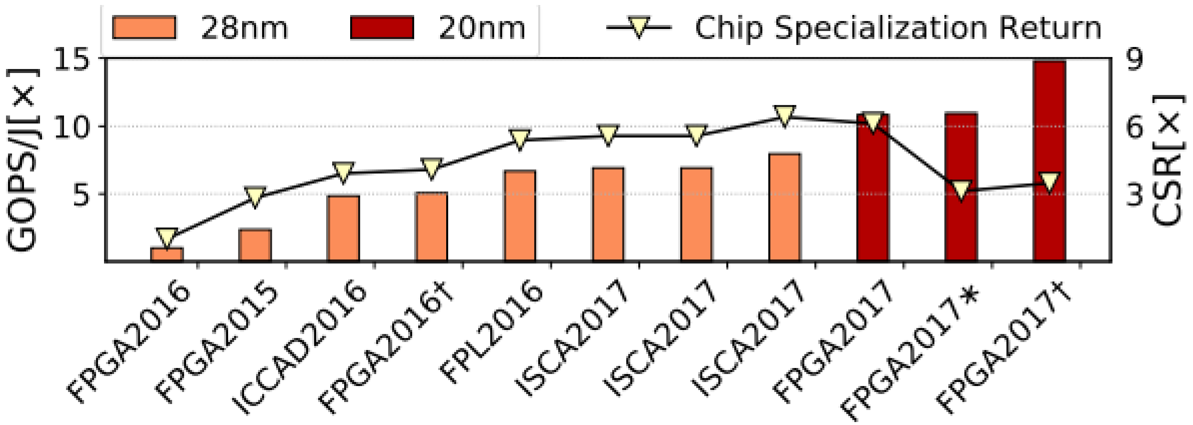
❑ Specialization returns increased to 6x, then stagnating

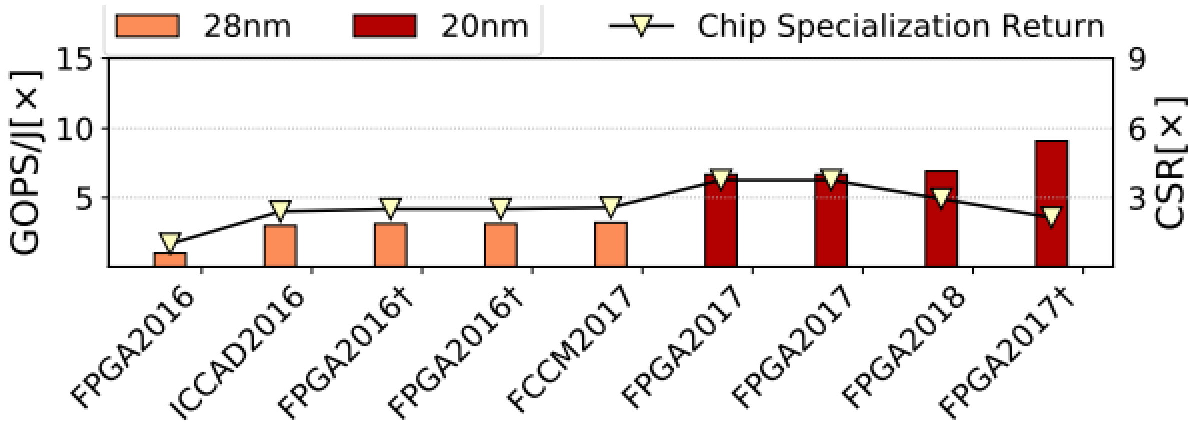




Application #3:
Neural Network Inference on FPGAs

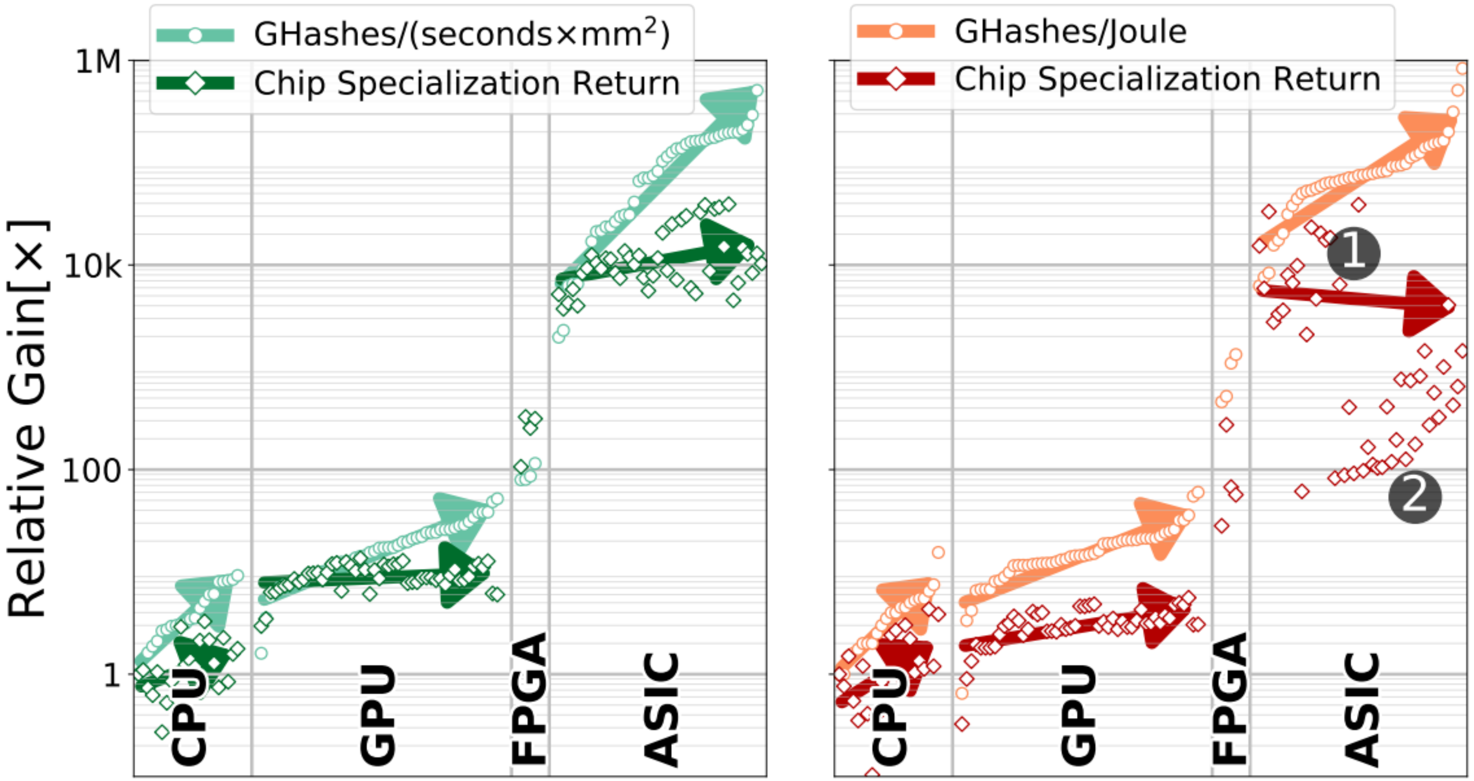
- ❑ Energy efficiency specialization returns also increased before stagnating
- ❑ Relatively new application, new algorithms had driven improvement





Application #4: Bitcoin Mining

- ❑ Same story as before



(a) Performance

(b) Energy Efficiency

Conclusion

❑ Chip specialization is one of the most prominent solutions to dark silicon

- Lots of work/research to be done to explore chip specialization

❑ However, it is not a long-term solution beyond Moore’s law

- Parallelism dies with CMOS scaling: No more transistors = no more cores
- All popular domains will mature. Diminishing optimization returns will follow

❑ Long term:

- We must explore other forms of optimizations that are not CMOS driven