High Performance and Low Power Design Techniques for ASIC and Custom in Nanometer Technologies

David Chinnery

Mentor Graphics®

Outline

- Introduction
- Microarchitecture
- Clock distribution, clock gating, and registers
- Logic style
- Cell design, cell sizing, and wire sizing
- Voltage scaling, and process technology
- Summary

Outline

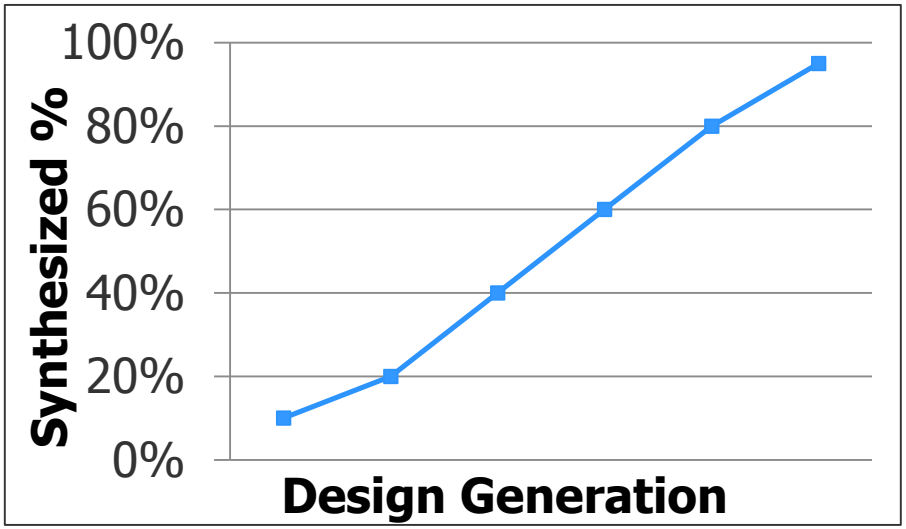
- Introduction
- Microarchitecture
- Clock distribution, clock gating, and registers
- Logic style
- Cell design, cell sizing, and wire sizing
- Voltage scaling, and process technology
- Summary

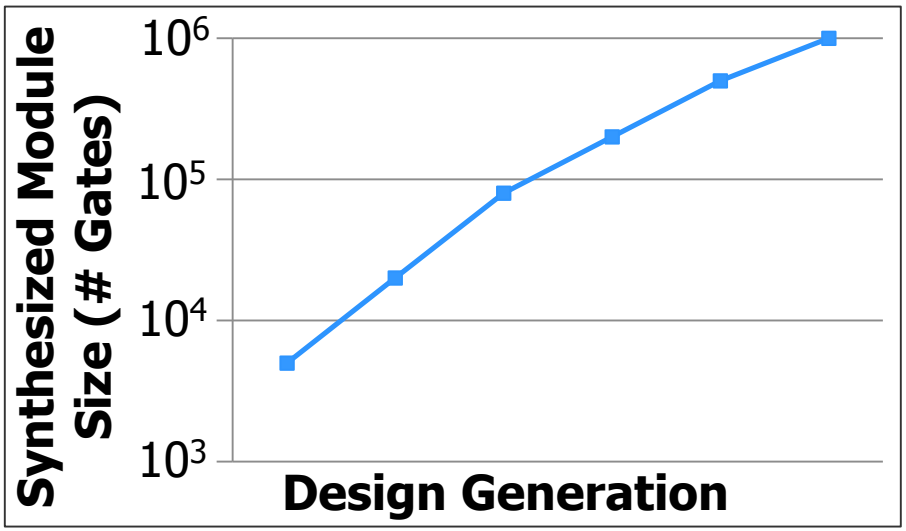
Digital circuit design styles

- Synthesized
  - Standard cell library of logic
  - **Automated** synthesis, placement, and route
- Semi-custom
  - Standard cell library with customized cells for the project
  - **Manual** schematic entry, **cell-level** layout, pre-routes
  - Can be mixed with synthesized logic by using **size only** or **don't modify** attributes and **relative placement constraints**
- Full custom
  - Additional cells specific for the design
  - **Manual** schematic entry, **transistor-level** layout and wiring
  - Must be encapsulated in a macro that is characterized
- We'll compare a high productivity Application-Specific Integrated Circuit (ASIC) methodology versus custom

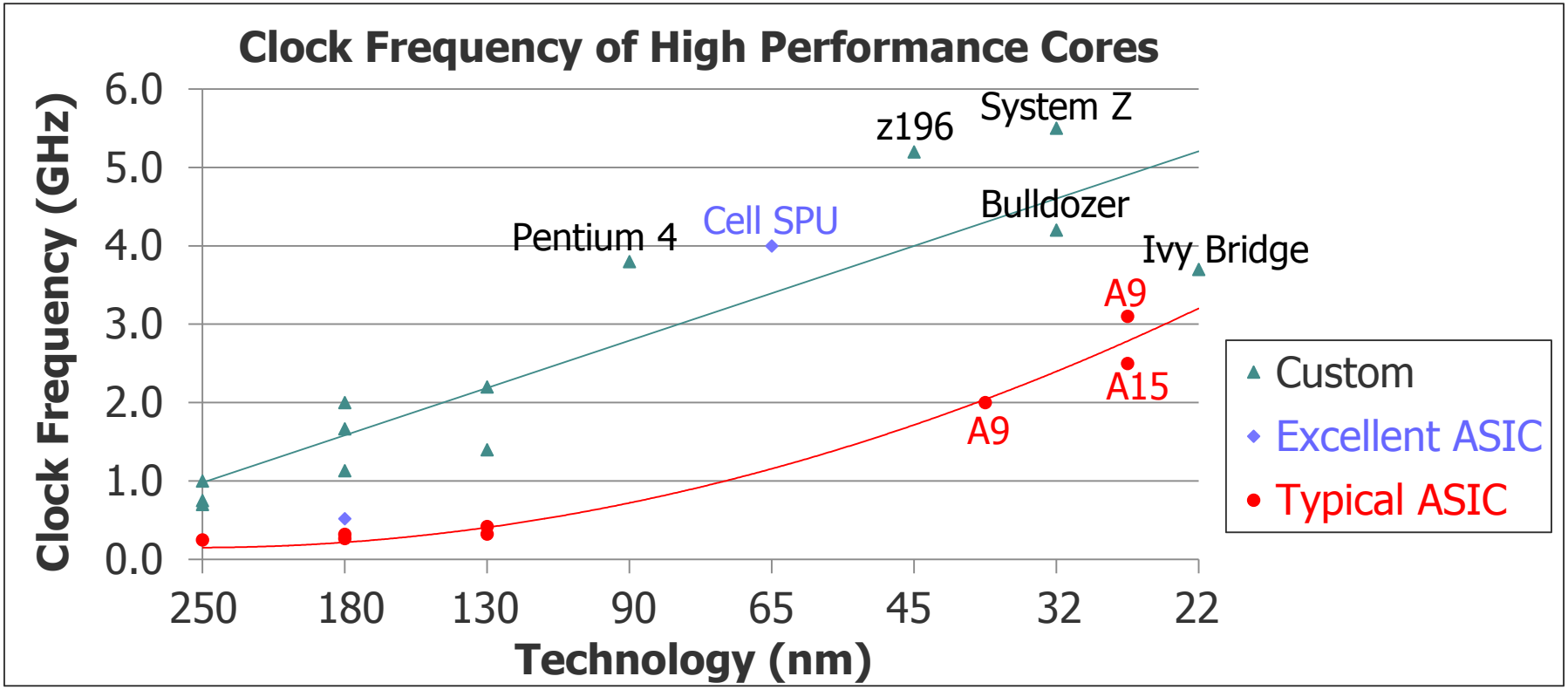
Custom design trends

- ASIC flow productivity is roughly 4× semi-custom, 16× full custom
- Larger designs and time-to-market motivate greater use of synthesis
- Moving from small synthesized sub-blocks to fewer timing critical custom datapath sub-blocks, e.g. IBM's 32nm 5.5GHz System z
- AMD's Bobcat and Jaguar cores have 1.1 and 1.25 million instances & were synthesized flat with multiple instances of a few custom memory macros



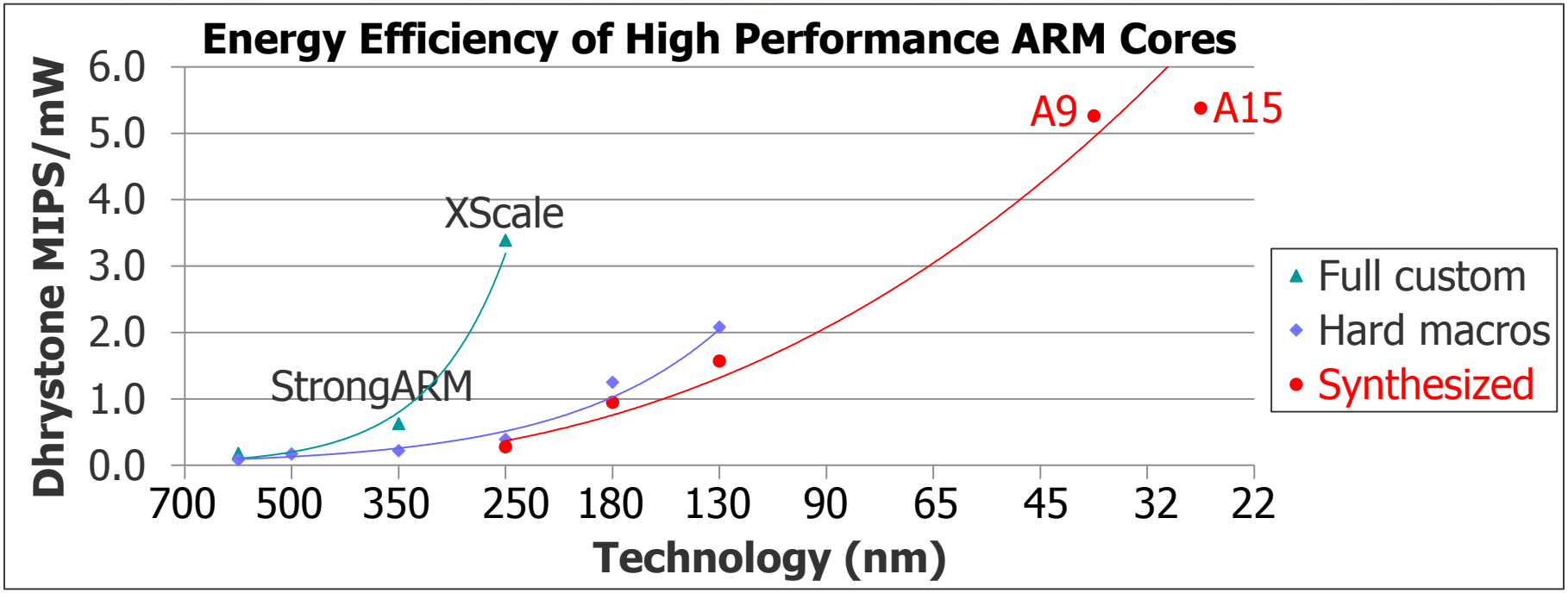


What was the performance gap?



- Custom designs were 3 to 8× faster than ASICs
- Performance gap is below 2× today, custom limited by long design time
  - Toshiba synthesized 4GHz Cell streaming processor unit (SPU) in 2007

What was the power gap?



- Custom had 2.6 to 7× energy efficiency of high performance ASICs
  - Custom ARMs had 3 to 4× energy efficiency versus synthesized
- Apple's 32nm Swift ARM core has custom layout and similar performance vs. energy efficiency trade-off to ASIC ARM cores
- Today, synthesizable ARMs dominate x86 in embedded, strong rivals in tablets, and entering the server market

Factors contributing to the gap today, calculated at a tight performance constraint

	ASIC Slower vs. Custom		ASIC Power vs. Custom	
Contributing Factor	**Typical**	**Excellent**	**Typical**	**Excellent**
microarchitecture	2.1×	1.0×	3.7×	2.0×
clock distribution & gating, registers	1.6×	1.2×	1.8×	1.1×
logic style	1.2×	1.2×	1.5×	1.5×
logic design	1.3×	1.0×	1.2×	1.0×
technology mapping	1.0×	1.0×	1.4×	1.0×
floorplanning & placement	1.4×	1.0×	1.5×	1.1×
cell design, cell sizing, wire sizing	1.5×	1.1×	1.6×	1.1×
voltage scaling	1.1×	1.0×	2.0×	1.0×
process technology & variation	2.0×	1.2×	2.6×	1.3×

- There are typically insufficient design resources for custom integrated circuits to fully exploit all of these
- These factors are not multiplicative
  - Analyze with model of pipelining, gate sizing, and voltage scaling

What isn't covered in this presentation?

These also have large impact on performance and power:

- Parallelism, as impact varies significantly with application
- Heterogeneous architectures, e.g. CPU + GPU
- On-chip communication architecture and off-chip I/O
- Memory hierarchy
- Higher system-level and software factors
- Power-gating to reduce leakage power in standby
  - Entering/restoring from a power-gated state takes 10,000 to 200,000+ clock cycles, thus system and software considerations
  - Our focus is on total power when circuit is active or clock-gated

See the paper and books for discussion of logic design, tech mapping, floorplanning & placement, and process variation.

Outline

- Introduction
- Microarchitecture
- Clock distribution, clock gating, and registers
- Logic style
- Cell design, cell sizing, and wire sizing
- Voltage scaling, and process technology
- Summary

Microarchitecture comparison

Processor	Process (nm)	Issue Width	Instruction Ordering	Integer Width (bits)	Integer Pipeline Stages	# of Cores	Clock (GHz)	Power (W)
Intel Nehalem	45	4-way	out-of-order	64	16	4	3.33	130.0
Intel Atom	32	2-way	in-order	64	16	2	2.26	+GPU 10.0
AMD Bobcat	40	2-way	out-of-order	64	13	2	1.70	+GPU 18.0
AMD Jaguar	28	2-way	out-of-order	64	14	4	1.85	2.0
ARM A9 (TSMC)	40	2-way	out-of-order	32	8	2	2.00	1.9
ARM A9 (TSMC)	28	2-way	out-of-order	32	8	4	3.10	unknown
ARM A7 (Samsung)	28	2-way	in-order	32	8	4 A7 and	1.00	0.4
ARM A15 (Samsung)	28	3-way	out-of-order	32	15	4 A15	2.00	5.2

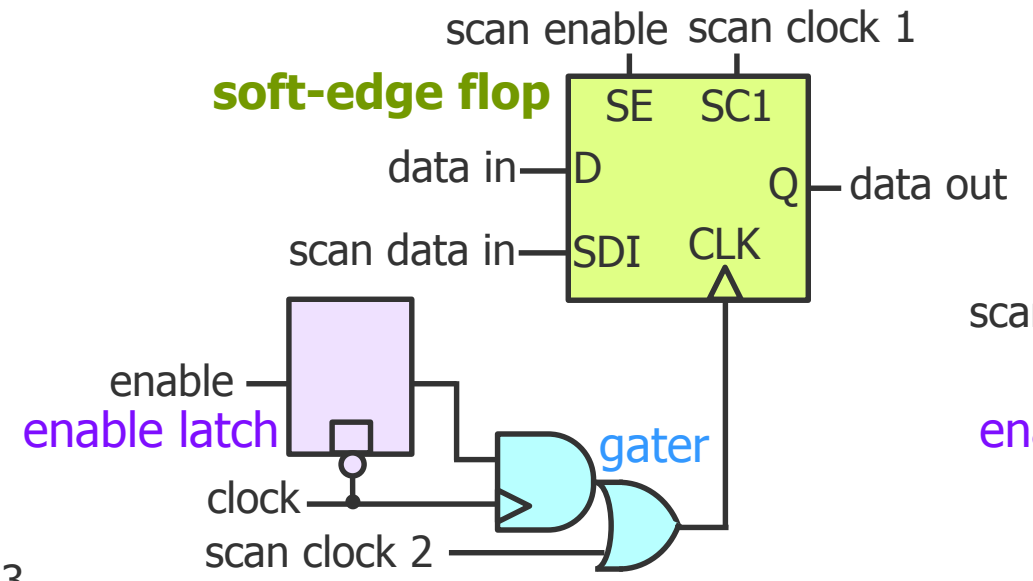
- ASIC microarchitectures have improved greatly in recent years
  - 64-bit ARMs will appear in the next couple of years
- ARM big.LITTLE architecture swaps from high performance to low power cores with dynamic voltage frequency scaling (DVFS)
  - Energy efficiency can improve 18% for 5% performance penalty
- Intel's low power Haswell parts will also target 10W power envelope

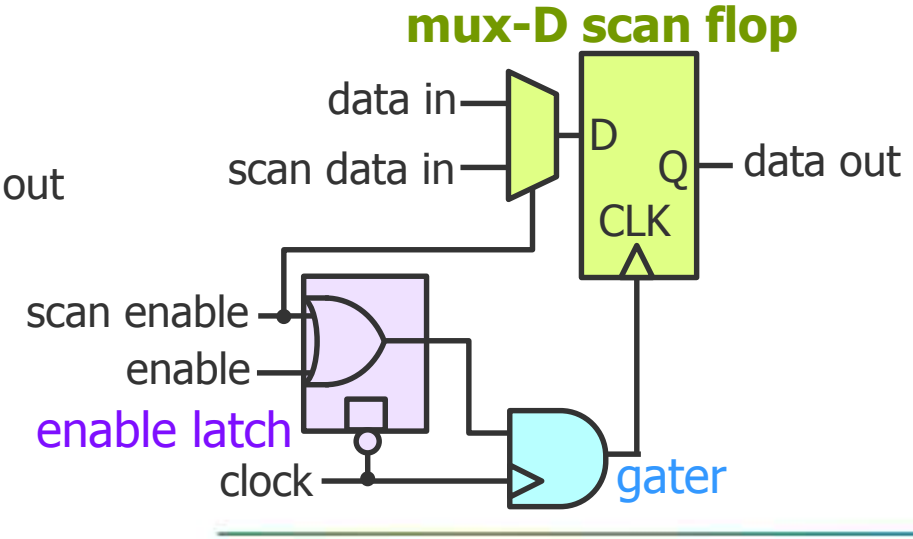
Outline

- Introduction
- Microarchitecture
- Clock distribution, clock gating, and registers
- Logic style
- Cell design, cell sizing, and wire sizing
- Voltage scaling, and process technology
- Summary

Types of registers

- **Latches** are faster, and reduce clock load, but clocking by pulse generators has process variation in pulse width
- **Mux-d scan flops** have a multiplexer in the data path
  - Functional clock used for scan, can have scan path hold issues
- **Level-sensitive scan design (LSSD) flops** are faster
  - Two separate clocks prevent scan path races
  - AMD's single-clock soft edge flops (SSEFs) are fast LSSD flops





Scan flip-flop characteristics

Comparison of 28nm flops	Percentage of Clock Period			
	Mux-D Flip-Flops		SSEFs	
	Fast	Low Power	Fast	Low Power
Relative Area	1.18	1.00	1.97	1.97
Hold Time	-4.3%	-6.6%	15.0%	6.8%
Clock-to-Q Delay	13.2%	19.0%	14.6%	15.7%
Setup Time	8.5%	10.0%	1.3%	10.7%
Clock-to-Q Delay + Setup Time	**21.7%**	**29.0%**	**15.9%**	**26.4%**

- LSSD flops are faster as no multiplexer in data path
  - The fast SSEFs are transparent for 10% of the clock period
  - Reduces setup time but increases hold time for data path
  - Allows time borrowing, giving some immunity to clock skew & jitter
- Mux-d scan flops are lower power, smaller area, but slower
  - In high speed designs, area is comparable to LSSD accounting for delay cells to fix mux-D scan path hold violations
- Jaguar uses faster mux-D flops with a dynamic front-end latch

Clock distribution methods

Distribution Methodology	Design Style	Design Effort	Typical Skew in 32nm	Number of Clock Tree Levels
Clock tree synthesis (CTS)	ASIC	Low	70 - 100ps	Deep, variable, e.g. 15 to 17
Hybrid: shallow CTS driving fixed # of levels	custom	Low - Medium	50 - 70ps	Shallow CTS (e.g. 3 to 4), then fixed # levels to flops (1 or 2)
Multi-source CTS (MSCTS)	ASIC	Medium	30 - 50ps	Fewer: e.g. 6 to 8
Clock mesh	custom	High	10 - 30ps	Fixed # levels: 1, 2, or 3

- Clock skew is worse if clock trees are deep or if depth varies, and process variation exacerbates this further
- Multi-source clock tree synthesis (MSCTS) has a grid of clock sources driven by a top level clock mesh, H-tree, or similar approach
- A fixed clock tree depth requires RTL and TCL specification of clock gaters and buffers to be cloned, and requires MSCTS or clock mesh
  - Hybrid approach is only possible with in-house custom tool support
- Tool support has improved for clock mesh placement restrictions, vendor support for clock mesh should be more widely available soon

Timing overhead per pipeline stage

	FO4 Delays for Different Design Styles		
	Typical ASIC	**Excellent ASIC**	**Custom**
Flop Type	low power mux-D	fast mux-D	fast LSSD
Clock Distribution Type	CTS	MSCTS	clock mesh
Clock-to-Q Delay	2.0	1.4	1.6
Setup Time	1.1	0.9	0.1
Clock Skew	4.3	1.3	0.5
Clock Jitter	2.6	1.3	0.3
Total	**10.0**	**4.9**	**2.5**

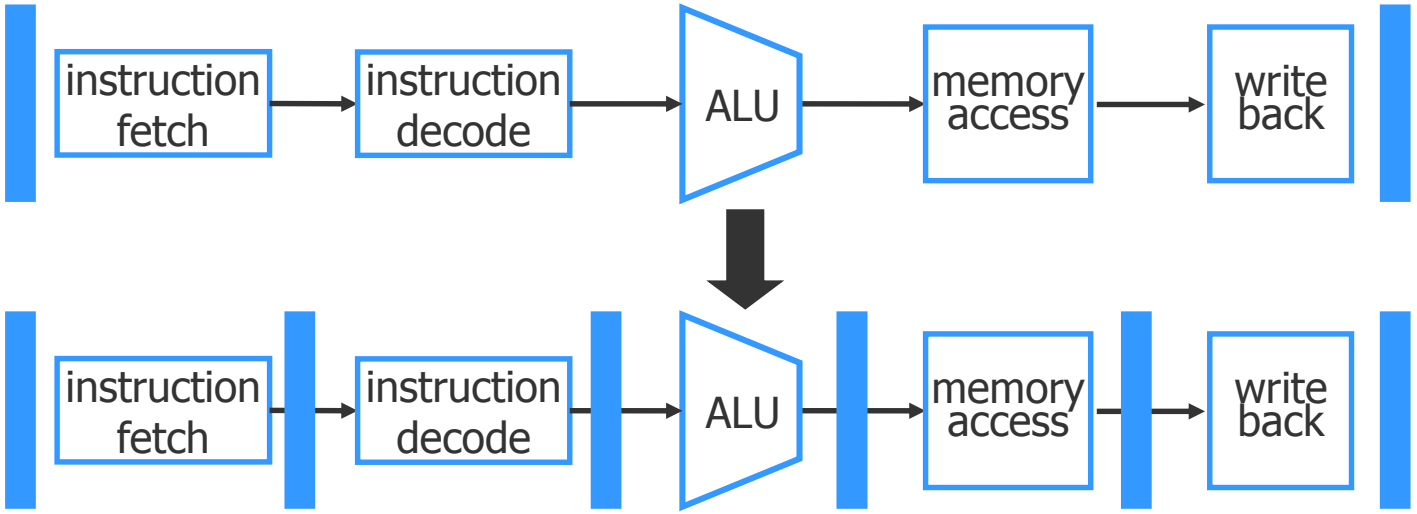
- Delay of inverter driving a fanout-of-4 (FO4) load is the delay metric
- Typical ASIC can have 10% extra timing overhead for pipeline stages not balanced by register retiming, useful clock skew, or RTL changes
- High performance design with 12 FO4 combinational delay per stage is slower by 1.6× for typical ASIC, 1.15× for excellent ASIC overhead

$$T = \frac{t_{combinational}}{n} + t_{timing\ overhead}$$

Pipelining timing overhead impact on power

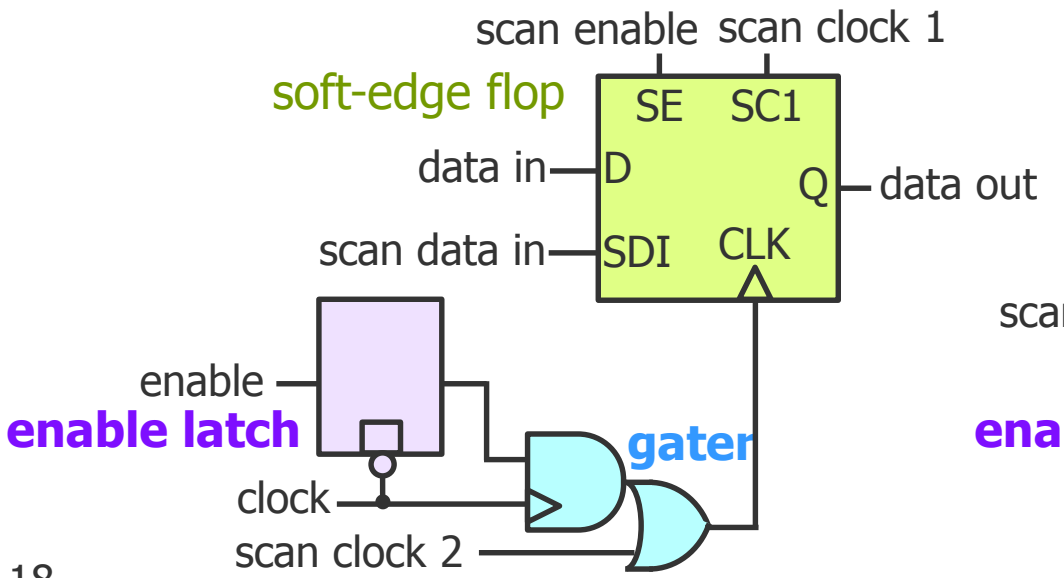
Reduced timing slack for gate sizing and voltage scaling:

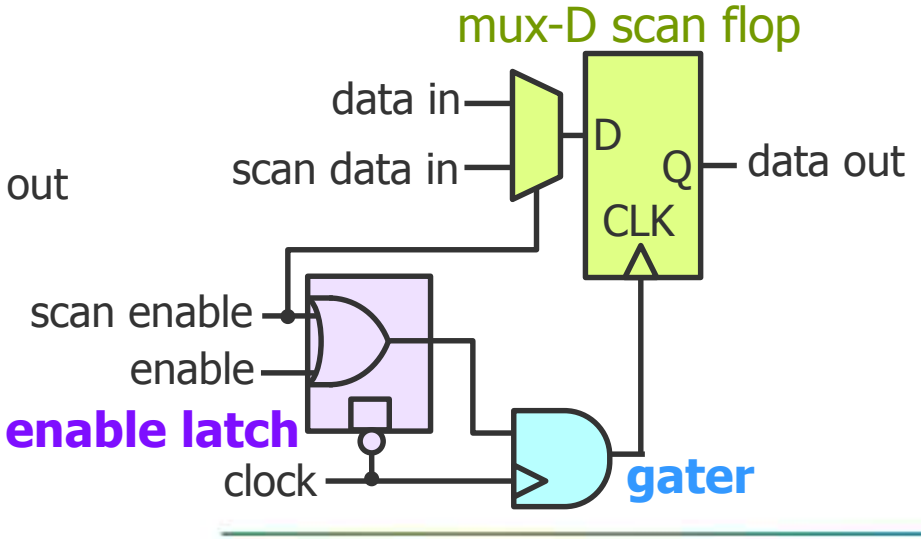
- 42 FO4/instruction is a tight constraint for a **typical ASIC**, where it has 3.7× higher energy/operation than custom
- 27 FO4/instruction is a tight constraint for an **excellent ASIC**, where it has 2.0× higher energy/operation than custom



Tool limitations impacting clock power

- No smarts to trade-off deeper enable buffering to reduce clock load versus enable latch cloning where enable path timing is critical
  - Poor support for split enable latch and clock gater
- Cluster or align flops to reduce clock wire load – user can specify relative placement constraints; a TCL script reduced it by 30%
- No support for mapping registers on same enable to multi-bit flops to reduce clock load by sharing clock circuitry



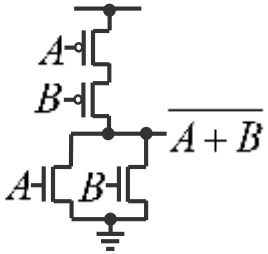


Outline

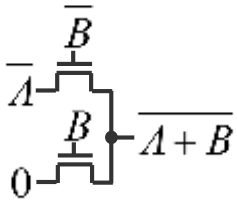
- Introduction
- Microarchitecture
- Clock distribution, clock gating, and registers
- Logic style
- Cell design, cell sizing, and wire sizing
- Voltage scaling, and process technology
- Summary

Combinational logic style

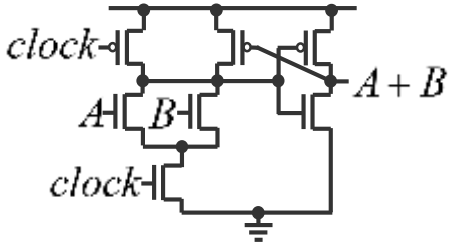
- **Dynamic domino logic** was faster but is less used now
  - Cannot use faster, leaky low threshold voltage transistors
  - Only usable in **full custom** macros, usually memory blocks today
- **Pulsed static CMOS logic** has similar speed to domino
  - Fast rise/fall path through logic, with slow return to initial state
  - Must use glitch free cells, so must be manually constructed
  - Used in **semi-custom** designs alongside synthesized logic
  - 1.25× faster than **static CMOS**, giving timing slack to reduce power
- **Pass transistor logic** is still used in custom as part of larger cells
- Little vendor support for classifying and verifying custom logic styles
- **Static CMOS logic** is robust to noise, lower power if timing not tight



static CMOS logic



pass transistor logic



domino logic

Outline

- Introduction
- Microarchitecture
- Clock distribution, clock gating, and registers
- Logic style
- Cell design, cell sizing, and wire sizing
- Voltage scaling, and process technology
- Summary

Cell sizing

- Global optimization of cell size can on average reduce total power by 16% and leakage power by 29% versus iterative greedy sizing in today's vendor tools
  - State-of-the-art research has shown run times fast enough to run on large designs, e.g. 13 hours on 361,000 gates
- Cells that are oversized post-route can be downsized with minimum perturbation if there are smaller size footprint compatible cells with pins on the same track
  - 60% of clock gaters in a 32nm CPU were downsized post-route due to poor pre-route in-house tool Steiner wire load estimates

Wire sizing

- Wire sizing is more important now, e.g. clock gater-to-flop wire loads increased from 40% in 45nm to 50% in 32nm
- Wire RC delay is also now a larger fraction of total delay
- Vendor tools support only a single non-default rule (NDR) for wire width and spacing during optimization
  - Can be sub-optimal by 10% for delay
- TCL scripts can assign NDRs versus load capacitance to limit electromigration, or reduce RC delay & resistive heating

Cell design

- Libraries with taller cells are faster for datapaths, but shorter heights increase cell density & reduce wire length so are a good choice for lower power designs
  - For Toshiba's SPU, track-height of 12 was 15% faster than height of 9, but track-height of 16 was higher power than height of 12
- Some custom cells are not safe for use in synthesis, e.g. bare pass gates on cell input or output connecting nearby, so some gap remains for ASICs

Outline

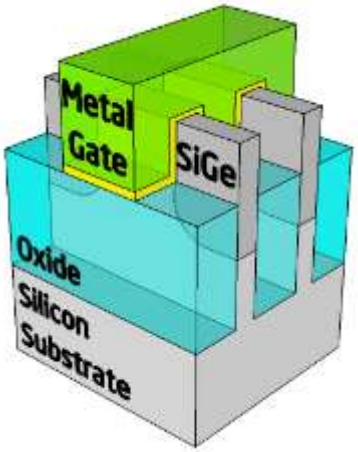
- Introduction
- Microarchitecture
- Clock distribution, clock gating, and registers
- Logic style
- Cell design, cell sizing, and wire sizing
- Voltage scaling, and process technology
- Summary

Voltage scaling support has improved

- Improved tool support for different voltage regions
- Improved access to libraries with a variety of supply and threshold voltages to trade-off delay versus power
- On-chip temperature sensors to manage temperature in DVFS turbo-mode to temporarily boost performance
  - Turbo-mode with throttling is in use in some recent ASICs
- DVFS adjustment of supply voltage or threshold voltage body biasing can compensate for slow or leaky chips
- Near-threshold voltage operation offers further power savings soon but may require on-chip delay sensors
  - Must disallow low drive cells, transistor stacks of at most three
  - Critical path distribution changes as slower at low supply voltage
  - In 32nm, energy efficiency is 2× at 0.5V versus 0.85V supply

Process technology

- 22nm FinFETs are triple-gated greatly reducing leakage
  - Intel's low power process is 50% faster than 32nm
  - Fins limit fine granularity in transistor size
- Intel's process is a year ahead of other foundries
  - With the decline in the PC market, Intel is now providing foundry capacity to some other companies
- Global Foundries, TSMC, and others are racing to catch up, promising transistor shrinks to 20nm then 14nm
  - But wire widths are not reducing as much, narrow wires need expensive double-patterning
  - Double-patterning and other yield issues complicate custom layout – generally needs to be cell-based



[Jan IEDM 2012]

Outline

- Introduction
- Microarchitecture
- Clock distribution, clock gating, and registers
- Logic style
- Cell design, cell sizing, and wire sizing
- Voltage scaling, and process technology
- Summary

A great ASIC example: Toshiba's synthesized Streaming Processor Unit (SPU) for the Cell

- Original 90nm design was mostly full custom with blocks of several thousand transistors, 19% was dynamic logic
- Toshiba achieved 4GHz at 1.4V in 65nm technology
- Significant improvements versus a 65nm custom version:
  - 12.5% faster than the custom design
  - 30% lower area from square floorplan instead of a tall one, reducing wire length by 30% and improving timing by 10%
  - 4× productivity with 10 to 100× larger synthesized blocks
  - Estimate 10% to 20% lower power with voltage scaling as faster
- Advanced techniques used by Toshiba:
  - Clock skew was limited to 20ps by using a clock mesh
  - 15% faster with 12-track height standard cell library vs. 9-track
  - 10% faster by using double-width wires to reduce the worst path delays by halving the RC delay

Conclusions

- Synthesis, automated place and route is an order of magnitude higher productivity than custom design
  - Tool capacity and quality for gate-sizing, placement, and routing continues to improve
- Most custom designs now use an automated methodology except for memory and timing critical datapaths
- Synthesized ASICs can achieve similar speed and power to custom with improved tools and advanced techniques:
  - Datapath and flop placement restrictions, customized library cells, and NDR wire sizing
- Hope is on the horizon for automated datapath placement
- Run time concerns designers seeking high productivity
  - Sparse database loading reduces runtime by an order of magnitude, but not available in vendor tools

Acknowledgements

- My advisor Professor Kurt Keutzer at UC Berkeley guided me in this research area in earlier years
- Thanks to Trevor Meyerowitz, Joseph Shinnerl, and Eleyda Negrón for editing feedback
- Thanks to Kiyoji Ueno for additional details of Toshiba's SPU

Additional references

- De Galas, J. 2013. Calxeda's ARM server tested. http://www.anandtech.com/show/6757/calxedas-arm-server-tested
- Harstein, A., and Puzak, T. 2003. Optimum Power/Performance Pipeline Depth. MICRO-36.
- Johnson, R. 2012. Multiscale processors tackle human interface. EDN. http://dc.ee.ubm-us.com/i/64572/19
- Ramirez, A. 2011. Energy Efficient Computing on Embedded and Mobile Devices. SC11.
- Shimpi, A., Klug, B., and Gowri, V. 2012. The iPhone 5 Review. http://www.anandtech.com/show/6330/the-iphone-5-review
- Shimpi, A. 2013. The ARM vs. x86 Wars Have Begun: In-Depth Power Analysis of Atom, Krait & Cortex A15. http://www.anandtech.com/show/6536/arm-vs-x86-the-real-showdown
- Singh, T., Bell, J., and Southard, S. 2013. Jaguar: A Next-Generation Low-Power x86-64 Core. ISSCC.
- Srinivasan, V., et al. Optimizing Pipelines for Power and Performance. MICRO-35.
- TSMC. 2012. TSMC's 28nm Based ARM Cortex-A9 Test Chip Reaches Beyond 3GHz. http://www.tsmc.com/tsmcdotcom/PRListingNewsAction.do?action=detail&newsid=6781
- Warnock, J., et al. 2013. 5.5GHz System z Microprocessor and Multichip Module. ISSCC.
- Williamson, R. 2012. Apple iPhone 5 – the A6 Application Processor. http://www.chipworks.com/blog/recentteardowns/2012/09/21/apple-iphone-5-the-a6-application-processor
- Yilmaz, M., et al. 2010. The Scan-DFT Features of AMD's Next-Generation Microprocessor Core. ITC.

Extra slides

Microarchitecture model of timing slack from pipelining for gate sizing and voltage scaling

$$P = \left(\frac{1}{T} \alpha_{clock\ gating} E_{dynamic} + P_{leakage} \right) \left(1 + \beta n^{\theta} \right) \qquad \alpha_{clock\ gating} = 1/(1+\gamma n)$$

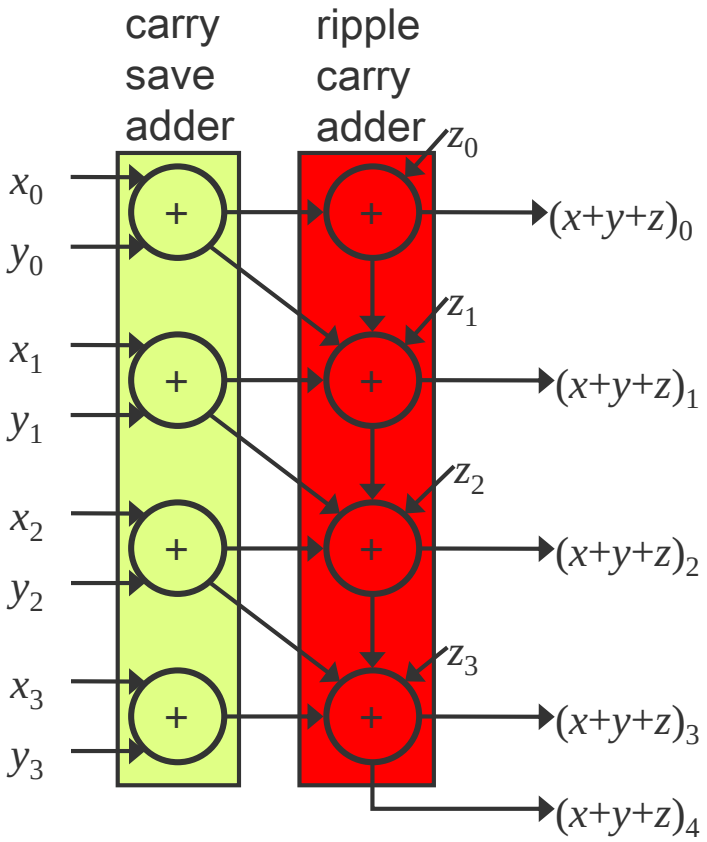
$$T \geq T_{\min} = \frac{t_{combinational}}{n} + t_{timing\ overhead} \qquad Delay\ per\ instruction = T(1+\gamma n)$$

Symbol	Represents	Value
n	number of pipeline stages	optimization variable
T	clock period	optimization variable
β	additional power for pipeline registers	0.05
γ	cycle per instruction penalty per stage	0.05
θ	increase in registers with more stages	1.10
$t_{combinational}$	combinational delay before pipelining	180 FO4 delays
$t_{timing\ overhead}$	timing overhead per stage	varies for ASIC & custom
$\alpha_{clock\ gating}$	reduction in pipeline stall power by clock gating	depends on n
$E_{dynamic}$	dynamic energy when switching	depends on $T/T_{\min}$
$P_{leakage}$	leakage power	depends on $T/T_{\min}$

- Model based on Srinivasan 2002, Harstein and Puzak 2003
- Dynamic and leakage power fits for voltage scaling and gate sizing

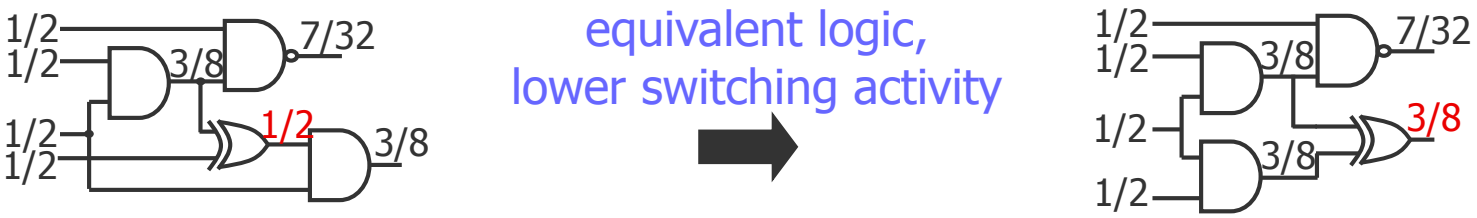
Logic design

- Logic design is the topology and logic structure to implement functional units
- Switching activity of a carry select 32-bit adder was ×1.8 worse than carry lookahead [Callaway VLSI Signal Proc.'92]
- 0.13um 64-bit radix-2 compound domino adder was slower and about ×1.3 energy compared to radix-4 [Zlatanovici ESSC'03]
- We implemented an algorithm to reduce switching activity in multipliers, reduced energy by ×1.1 for 64-bit [Ito ICCD'03]
- Given similar design constraints, ASIC designers can choose the same logic design as custom, ×1.0



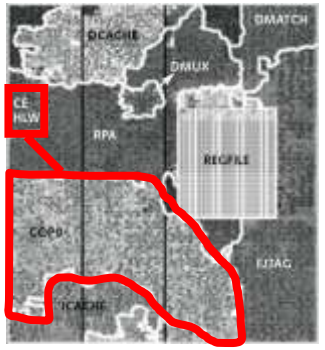
Technology mapping

- Tools don't support minimizing power when technology mapping
  - Targeting minimum area for multipliers results in 1.3× power, minimizing delay is also a poor choice
  - Instead of area, target power with switching activity analysis
- Various tech mapping techniques to reduce active power
  - 1.1 to 1.25×: state encoding assignments [Tsui ICCAD'94]
  - 1.25×: transformations based on controllability, observability, sub-expression elimination, decomposition [Pradhan'96]
  - 1.1×: pin reassignment based on signal activity [Shen ASPDAC'95]
  - Delay balancing to reduce glitching activity
- ASICs can do as well as custom if tools improve, in the meantime designers must carefully craft the RTL



Floorplanning and placement

- Poor floorplanning and cell placement, inaccurate wire loads
  - 1.5× worse power than custom
  - Use derating factors to improve wire load accuracy



automatic place and route

- We compared partitioning a design in 50K vs. 200K gate modules from 0.25um to 0.13um
  - 42% longer wires for 200K partitions
  - Interconnect may contribute 50% of total power
  - 1.2× increase in total power due to wiring, and gates will be upsized to drive the longer wires

block partitioned

[Hauck Micro. Report '01]

Floorplanning and placement

- Bit slices – can reduce wire length by 70% or more vs. automated place-and-route
  - Up to 1.4× energy reduction as faster and lower wiring capacitance [Chang SM Thesis MIT'98]
  - 1.5× energy reduction from bit slicing and some logic optimization [Stok, Puri, Bhattacharya, Cohn]
- Half-perimeter-width-length optimized placement has 2 to 3× wire length for datapaths versus manual placement
- Excellent ASICs have 1.1× higher power than custom due to poorer placement
- Recent research has shown 30% improvement in Steiner wire length by automated datapath placement

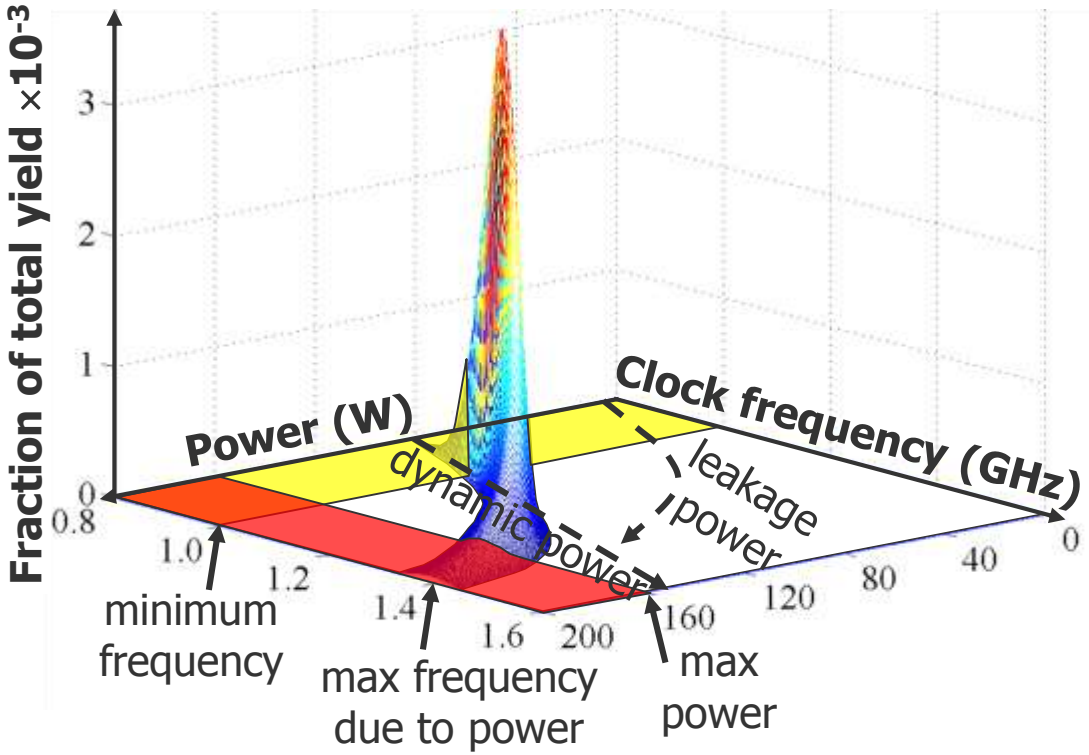
automatic place-and-route

tiled bit-slices

custom

Process variation

- ASICs are usually designed to work at worst case process and operating corners to ensure good yield
- High priced chips can be tested at different speeds to speed-bin or power-bin
- Delay lock loops and adjustable delay buffers can reduce clock skew due to process variation



www.mentor.com