inst.eecs.berkeley.edu/~eecs251b

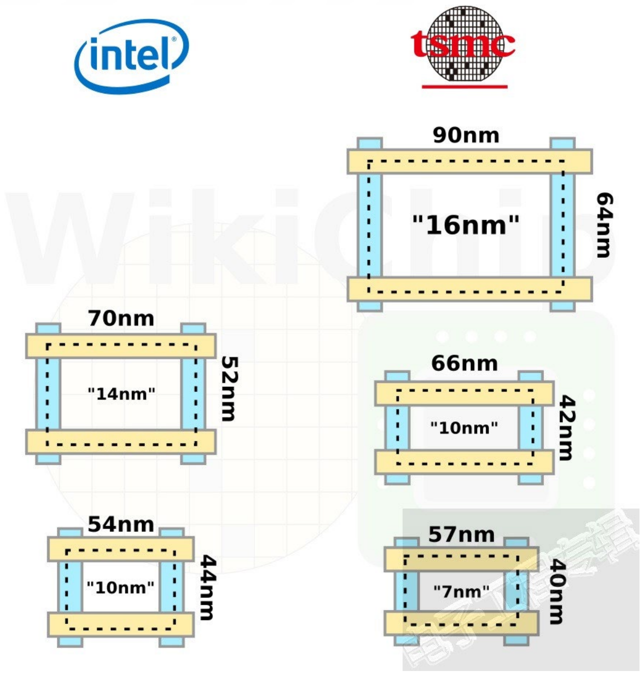
EECS251B : Advanced Digital Circuits and Systems

Lecture 8 – Features of Modern Technologies

Borivoje Nikolić

Apple, Huawei Use TSMC, But Their 7nm SoCs Are Different. When talking about the most advanced semiconductor manufacturing processes, it seems that most of the SoCs in 2019 can be collectively classified as 7nm. But not all 7nm is equal.

EE Times, January 22, 2020.



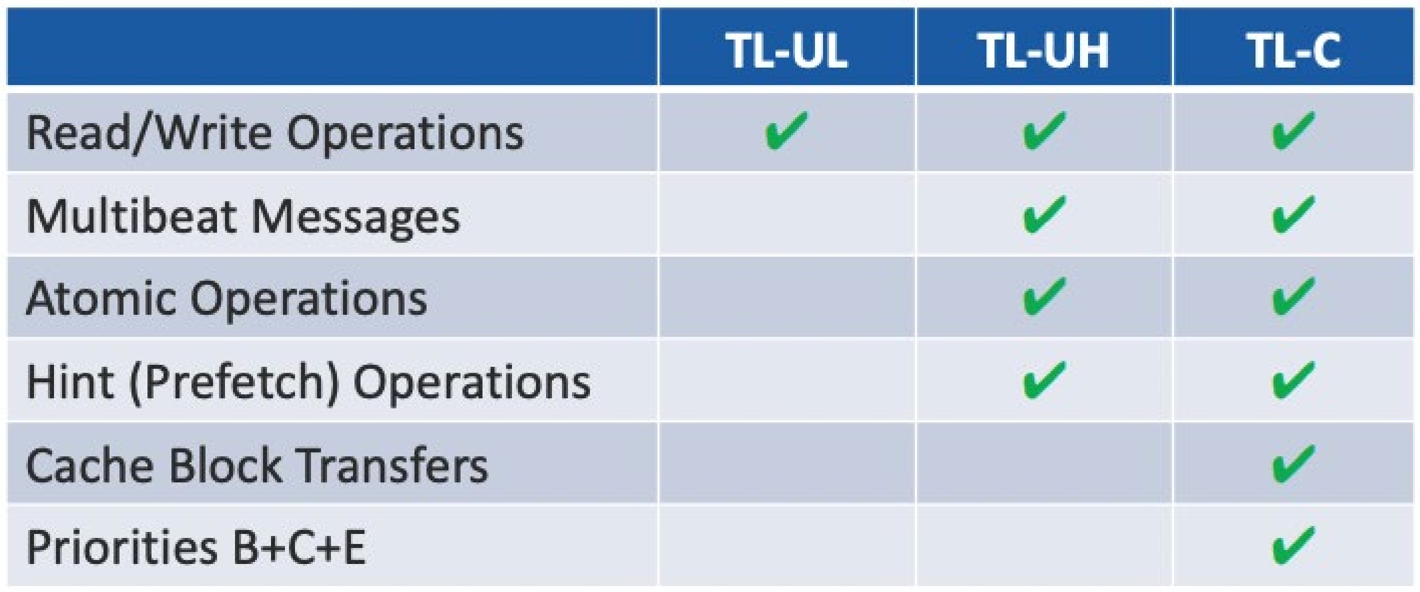
Announcements

- Lab 3 due this week
- Lab 4 released this week
- Homework 2 will be posted ~~at the end of the~~ **next** week
- Project teaming this week

- Accelerator Integration
- Tightly-coupled Acc. w/ RoCC
- **MMIO Acc. w/ TileLink**
- Examples

TileLink Protocol Levels

- TileLink Uncached Lightweight (TL-UL)
  - Only simple memory read/write (Get/Put) operations of single words (similar to AXILite)
- TileLink Uncached Heavyweight (TL-UH)
  - Adds various hints, atomic, and burst accesses but w/o coherence (similar to AXI4)
- TileLink Cached (TL-C)
  - Complete protocol, which supports use of coherent caches (similar to ACE)

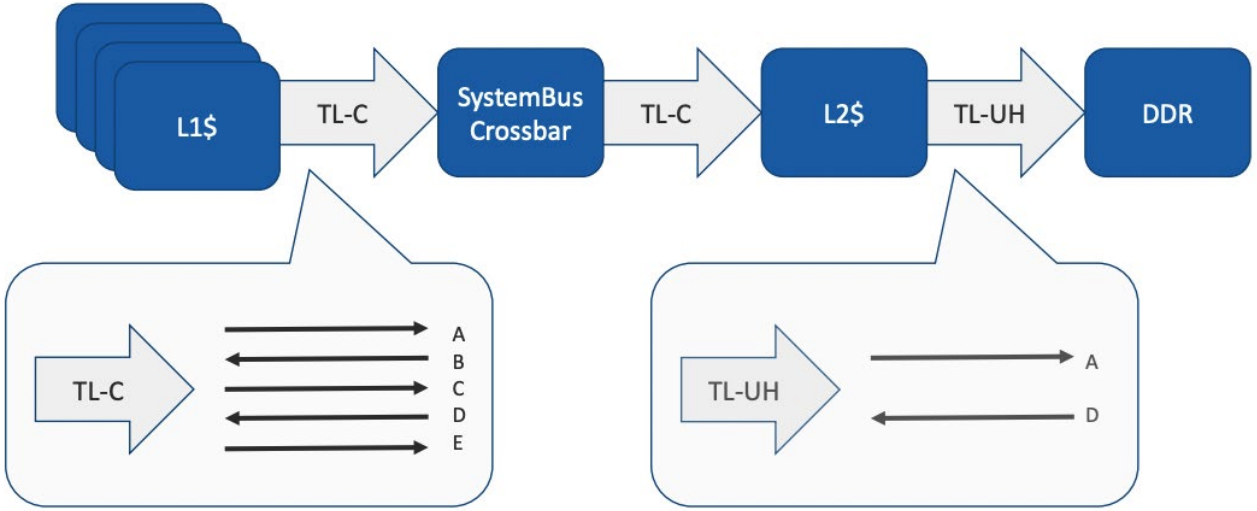


	TL-UL	TL-UH	TL-C
Read/Write Operations	✓	✓	✓
Multibeat Messages		✓	✓
Atomic Operations		✓	✓
Hint (Prefetch) Operations		✓	✓
Cache Block Transfers			✓
Priorities B+C+E			✓

https://riscv.org/wp-content/uploads/2017/12/Wed-1154-TileLink-Wesley-Terpstra.pdf

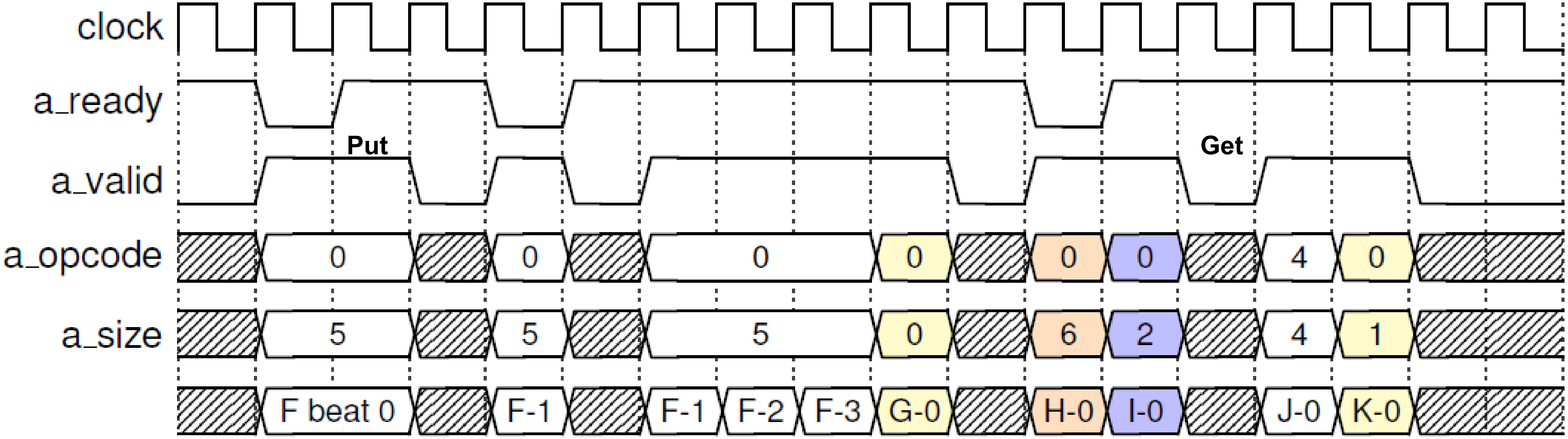
TileLink Channels

Channel	Direction	Purpose	
A	Manager to Subordinate	Request messages sent to an address	
B	Subordinate to Manager	Request messages sent to a cached block	(TL-C only)
C	Manager to Subordinate	Response messages from a cached block	(TL-C only)
D	Subordinate to Manager	Response messages from an address	
E	Manager to Subordinate	Final handshake for cache block transfer	(TL-C only)



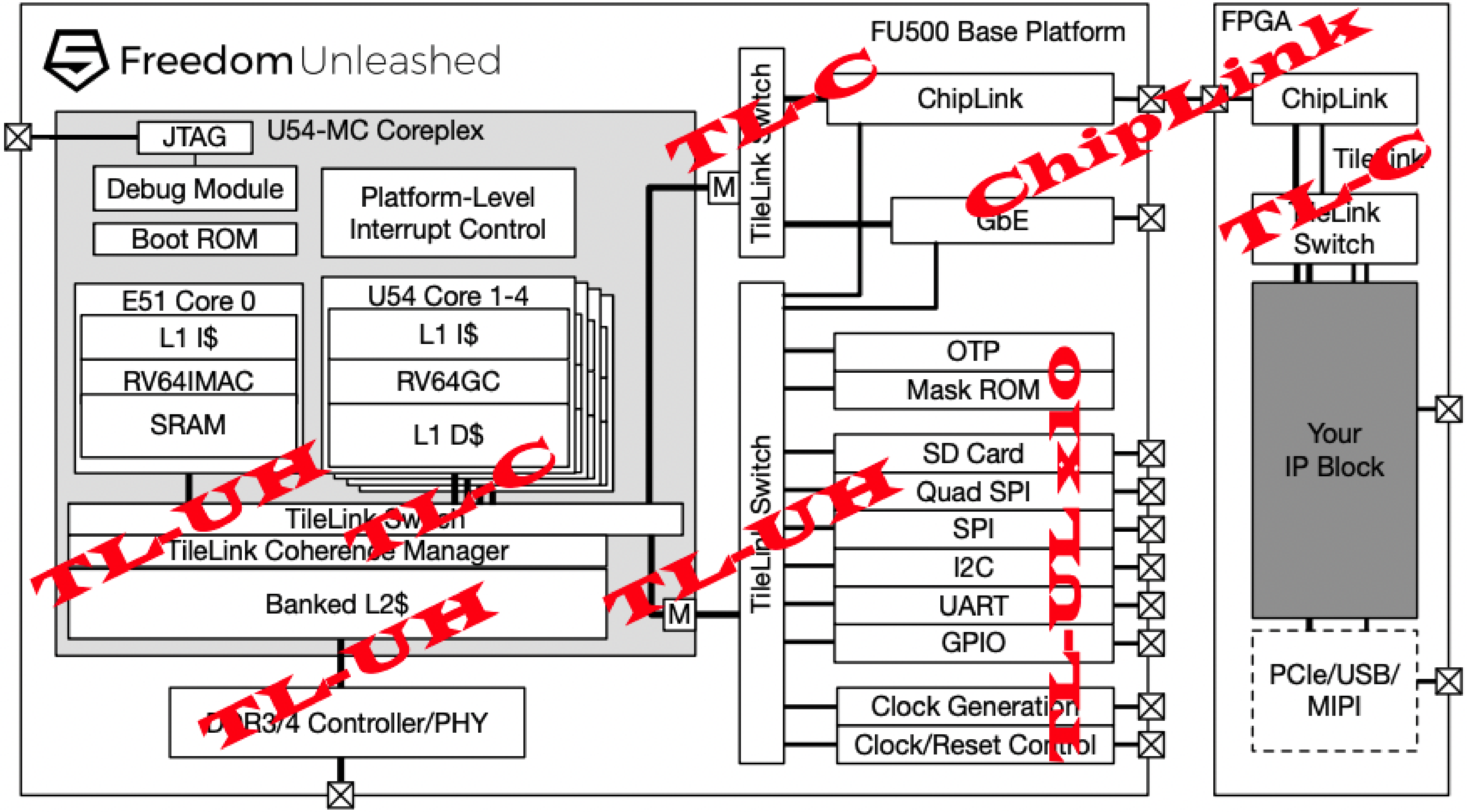
* Using AXI agent names here
* TileLink Client -> AXI Manager
* TileLink Manager -> AXI Subordinate

TileLink Basics: Flow-Control



A beat is exchanged only when both ready and valid are HIGH

TileLink: The Foundation of SiFive's FU500

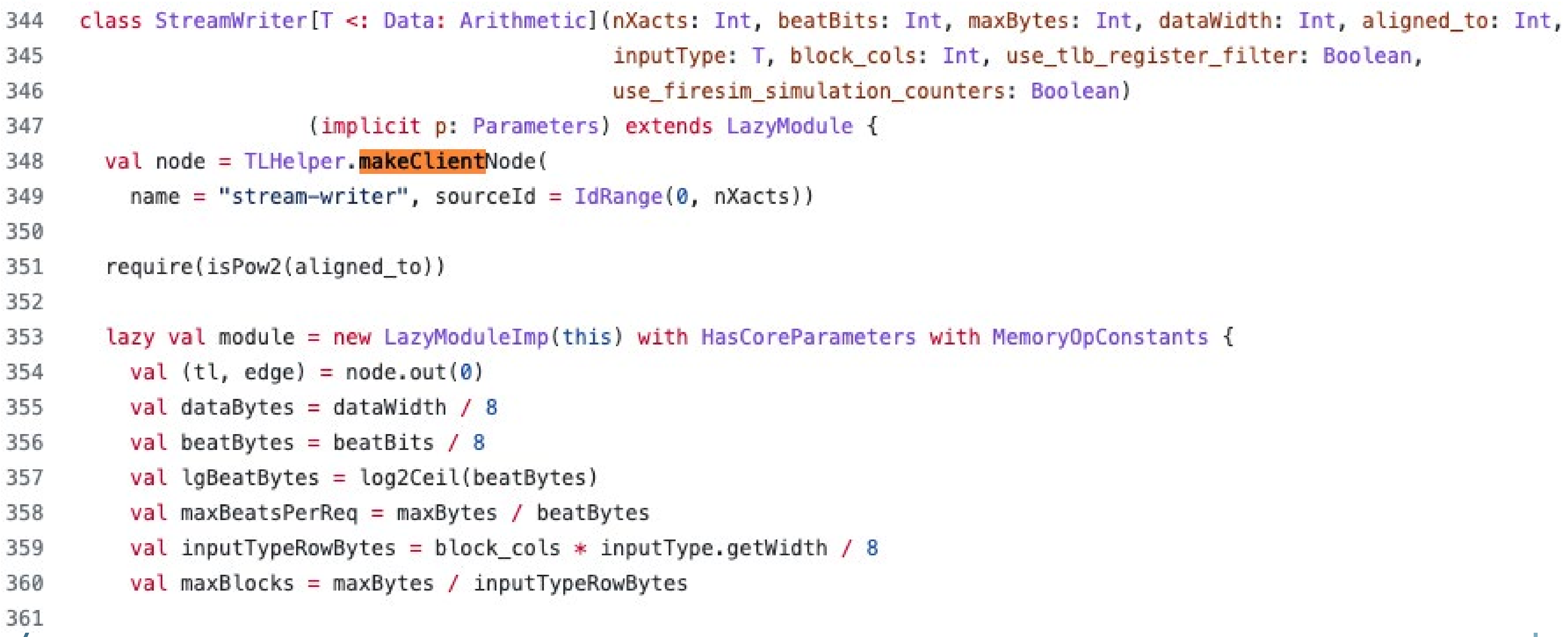


https://riscv.org/wp-content/uploads/2017/12/Wed-1154-TileLink-Wesley-Terpstra.pdf

TileLink Examples

- RoCC accelerators: SHA3
  - https://github.com/ucb-bar/sha3
- RoCC + TL-UL: protobuf accelerator
  - https://github.com/ucb-bar/protoacc
- RoCC + TL-UH: Gemmini accelerator
  - https://github.com/ucb-bar/gemmini
- RoCC + TL-UH: Hwacha vector accelerator
  - https://github.com/ucb-bar/hwacha
- TL-UH: IceNIC network interface controller for FireSim
  - https://github.com/firesim/icenet

Instantiate a TileLink node for your module



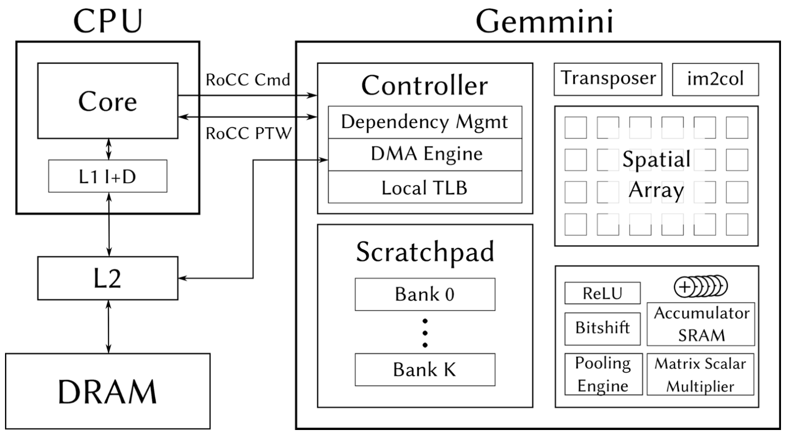
```scala
class StreamWriter[T <: Data: Arithmetic](nXacts: Int, beatBits: Int, maxBytes: Int, dataWidth: Int, aligned_to: Int,
                                          inputType: T, block_cols: Int, use_tlb_register_filter: Boolean,
                                          use_firesim_simulation_counters: Boolean)
                  (implicit p: Parameters) extends LazyModule {
  val node = TLHelper.makeClientNode(
    name = "stream-writer", sourceId = IdRange(0, nXacts))

  require(isPow2(aligned_to))

  lazy val module = new LazyModuleImp(this) with HasCoreParameters with MemoryOpConstants {
    val (tl, edge) = node.out(0)
    val dataBytes = dataWidth / 8
    val beatBytes = beatBits / 8
    val lgBeatBytes = log2Ceil(beatBytes)
    val maxBeatsPerReq = maxBytes / beatBytes
    val inputTypeRowBytes = block_cols * inputType.getWidth / 8
    val maxBlocks = maxBytes / inputTypeRowBytes

```

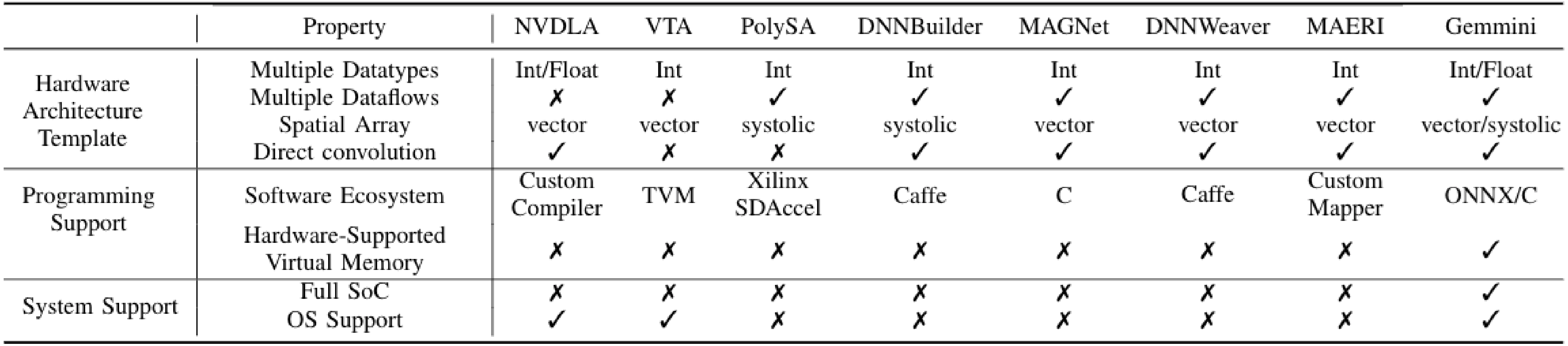
https://github.com/ucb-bar/gemmini/blob/master/src/main/scala/gemmini/DMA.scala#L348

- Accelerator Integration
- Tightly-coupled Acc. w/ RoCC
- MMIO Acc. w/ TileLink
- **Examples**

Gemmini: Full-System Co-Design of Hardware Accelerators

- **Full-stack**
  - Includes OS
  - End-to-end workloads
  - “Multi-level” API
- **Full-SoC**
  - Host CPUs
  - Shared memory hierarchies
  - Virtual address translation





	Property	NVDLA	VTA	PolySA	DNNBuilder	MAGNet	DNNWeaver	MAERI	Gemmini
Hardware Architecture Template	Multiple Datatypes	Int/Float	Int	Int	Int	Int	Int	Int	Int/Float
	Multiple Dataflows	✗	✗	✓	✓	✓	✓	✓	✓
	Spatial Array	vector	vector	systolic	systolic	vector	vector	vector	vector/systolic
	Direct convolution	✓	✗	✗	✓	✓	✓	✓	✓
Programming Support	Software Ecosystem	Custom Compiler	TVM	Xilinx SDAccel	Caffe	C	Caffe	Custom Mapper	ONNX/C
	Hardware-Supported Virtual Memory	✗	✗	✗	✗	✗	✗	✗	✓
System Support	Full SoC	✗	✗	✗	✗	✗	✗	✗	✓
	OS Support	✓	✓	✗	✗	✗	✗	✗	✓

https://github.com/ucb -bar/gemmini

[DAC’2021 Best Paper Award]

Using RoCC + TileLink w/ Gemmini

- How **Gemmini**, a DNN accelerator, uses RoCC and TileLink
  - How does Gemmini **read** data from main memory into Gemmini’s scratchpad?

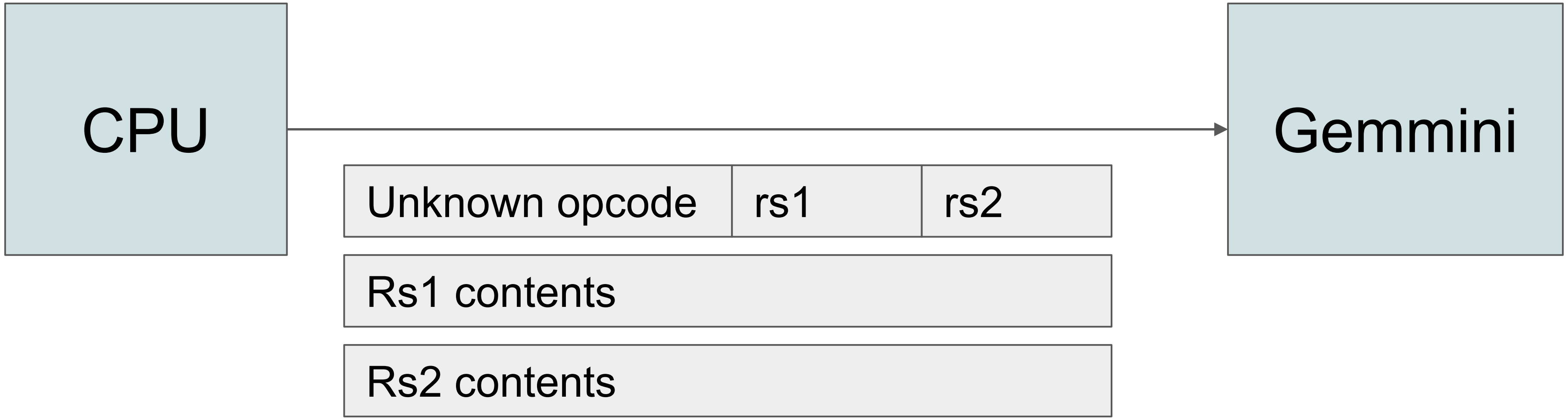
1. Host CPU encounters **unknown** RISC-V instruction

Unknown opcode	rs1	rs2

Using RoCC + TileLink w/ Gemmini

- How **Gemmini**, a DNN accelerator, uses RoCC and TileLink
  - How does Gemmini **read** data from main memory into Gemmini’s scratchpad?

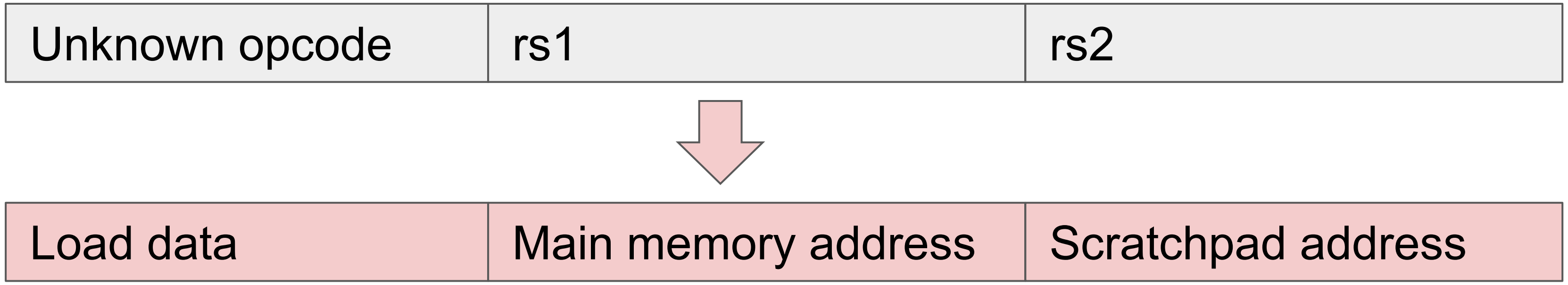
1. Host CPU encounters **unknown** RISC-V instruction
2. Host CPU **dispatches** unknown instruction to RoCC accelerator
   a. As well as Rs1 and Rs2 contents (128 bits extra bits)



Using RoCC + TileLink w/ Gemmini

- How **Gemmini**, a DNN accelerator, uses RoCC and TileLink
  - How does Gemmini **read** data from main memory into Gemmini's scratchpad?

1. Host CPU encounters **unknown** RISC-V instruction
2. Host CPU **dispatches** unknown instruction to RoCC accelerator
   a. As well as Rs1 and Rs2 contents (128 bits extra bits)
3. Gemmini **decodes** instruction
   a. It's a load instruction!



Unknown opcode	rs1	rs2
Load data	Main memory address	Scratchpad address

Using RoCC + TileLink w/ Gemmini

- How **Gemmini**, a DNN accelerator, uses RoCC and TileLink
    - How does Gemmini **read** data from main memory into Gemmini’s scratchpad?

1. Host CPU encounters **unknown** RISC-V instruction
2. Host CPU **dispatches** unknown instruction to RoCC accelerator
    a. As well as Rs1 and Rs2 contents (128 bits extra bits)
3. Gemmini **decodes** instruction
    a. It’s a load instruction!
4. Gemmini asks CPU’s page table walker to **translate** addresses in Rs1, Rs2
    a. PTW is only available through RoCC interface

Using RoCC + TileLink w/ Gemmini

- How **Gemmini**, a DNN accelerator, uses RoCC and TileLink
  - How does Gemmini **read** data from main memory into Gemmini's scratchpad?

1. Host CPU encounters **unknown** RISC-V instruction
2. Host CPU **dispatches** unknown instruction to RoCC accelerator
   a. As well as Rs1 and Rs2 contents (128 bits extra bits)
3. Gemmini **decodes** instruction
   a. It's a load instruction!
4. Gemmini asks CPU's page table walker to **translate** addresses in Rs1, Rs2
   a. PTW is only available through RoCC interface
5. Gemmini sends **TileLink requests** to read data from main memory
   a. Often, multiple TileLink requests must be sent, due to TileLink's alignment and length limitations

Review

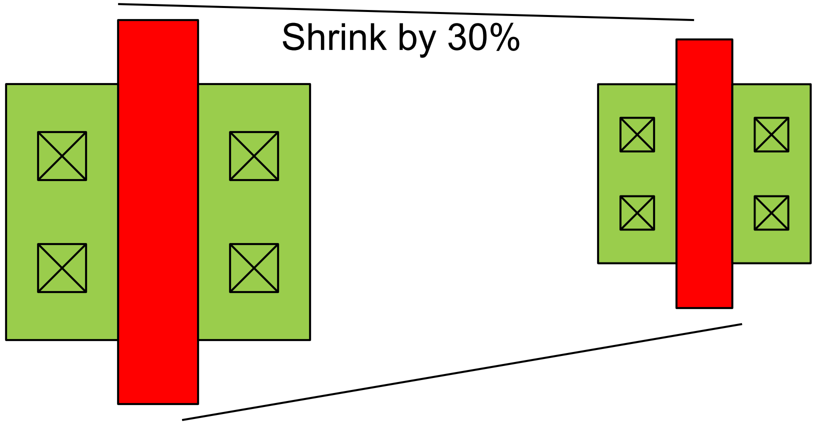
- Accelerators don't exist in isolation.
- RoCC for tightly-coupled accelerators
- TileLink for loosely-coupled, MMIO accelerators
- Examples:
  - RoCC accelerators: SHA3
    - https://github.com/ucb-bar/sha3
  - RoCC + TL-UL: protobuf accelerator
    - https://github.com/ucb-bar/protoacc
  - RoCC + TL-UH: Gemmini accelerator
    - https://github.com/ucb-bar/gemmini
  - RoCC + TL-UH: Hwacha vector accelerator
    - https://github.com/ucb-bar/hwacha
  - TL-UH: IceNIC network interface controller for FireSim
    - https://github.com/firesim/icenet

A Perspective on Scaling

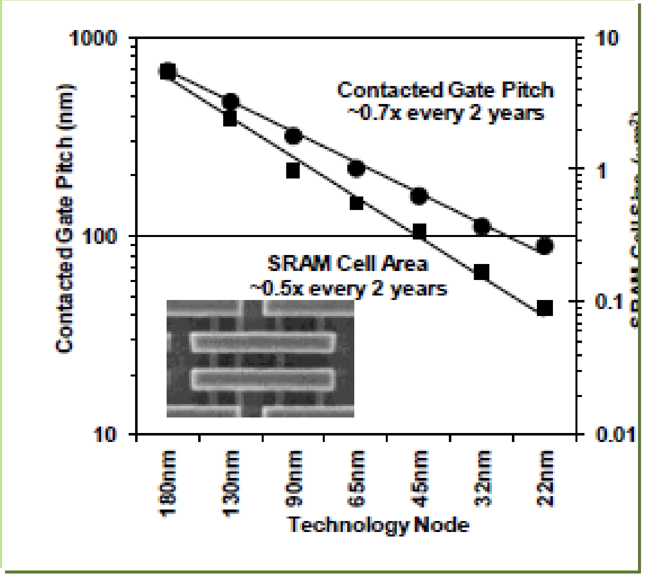
Key Points

- Technology scaling (Moore's law) is slowing down
  - But logic and memory are continuing to scale, possibly at different pace
  - We anyway can't power up all transistors we can put on a chip
- Dennard scaling has ended >10 years ago
  - We cover it for understanding of current issues
- There are many technology flavors available at the moment
  - We need to know what each one brings to us, so we can choose the right one for our project
    (may have to wait until the end of the class to figure out all options)
- Physical (velocity-saturated) models scale across technologies

Transistor Scaling

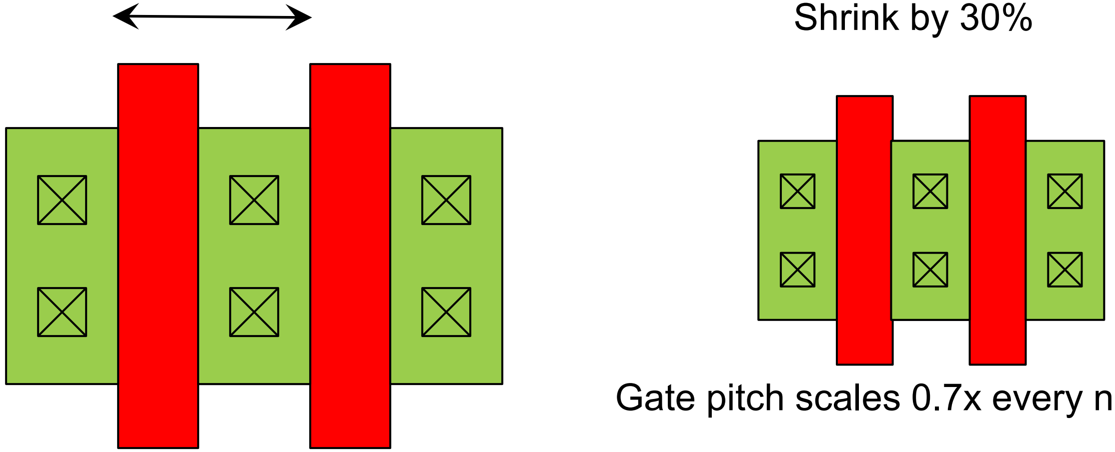


Shrink by 30%



C. Auth, VLSI’12

“Contacted gate pitch”



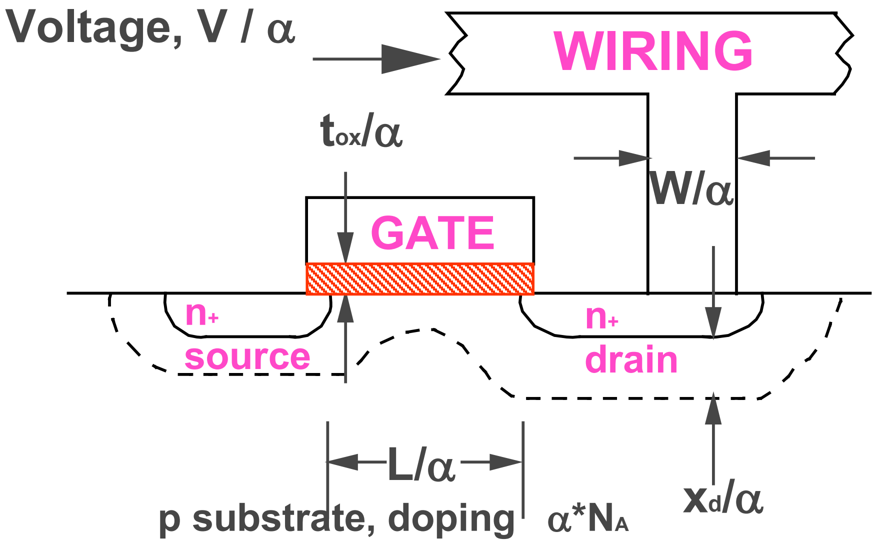
Shrink by 30%

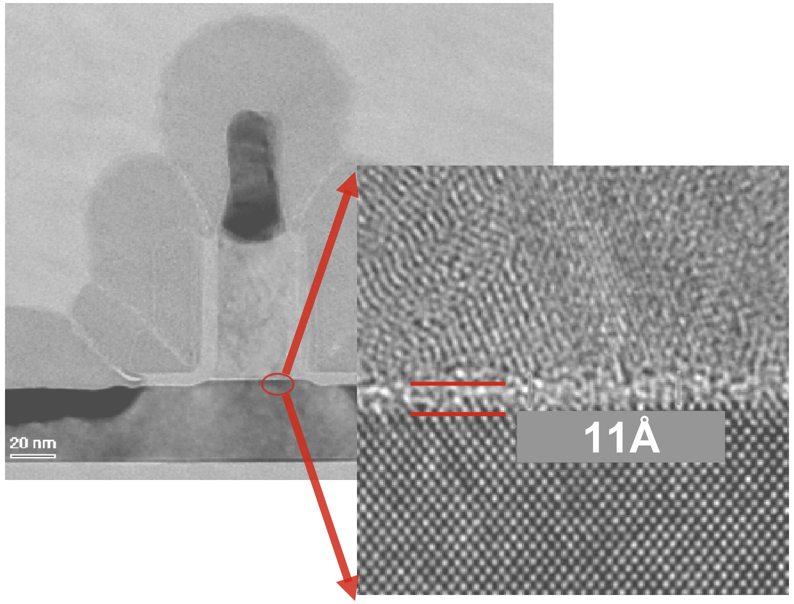
Gate pitch scales 0.7x every node

If x and y scale by 0.7 node to node
=> 2x more transistors in same area!

Intel	45nm	32nm	22nm	14nm	10nm
Contacted gate pitch	160nm	112.5nm	90nm	70nm	54nm

CMOS Scaling Rules (Dennard)





SCALING:

Voltage:	V/α
Oxide:	t_{ox}/α
Wire width:	W/α
Gate length:	L/α
Diffusion:	x_d/α
Substrate:	$\alpha * N_A$

RESULTS:

Higher Density:	$\sim\alpha^2$
Higher Speed:	$\sim\alpha$
Power/ckt:	$\sim 1/\alpha^2$
Power Density:	~~~Constant~~

e.g.

$1/\alpha = 0.7$
$\alpha = 1.43$

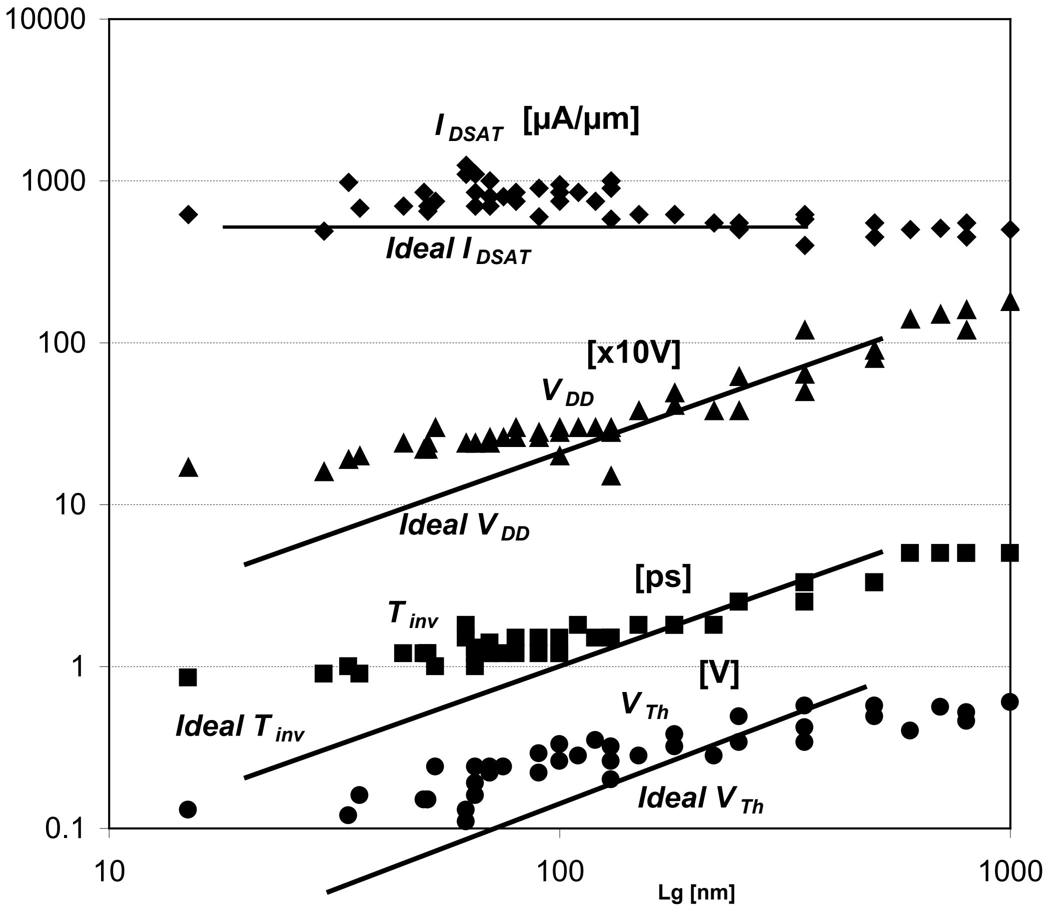
Cg ~ ?   Id ~?

R. H. Dennard *et al.*,
IEEE J. Solid State Circuits, (1974).

Ideal vs. Real Scaling

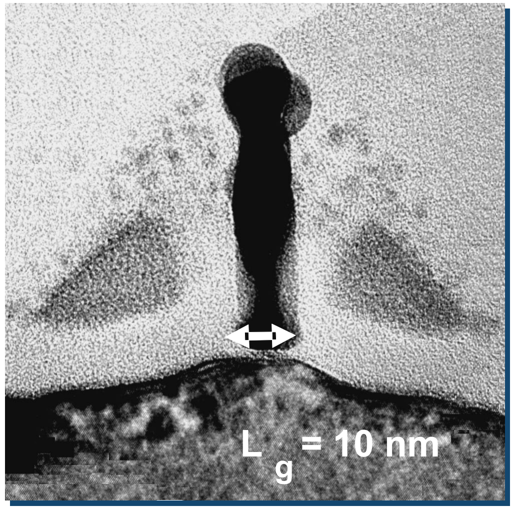
- Leakage slows down V_{Th}, V_{DD} scaling

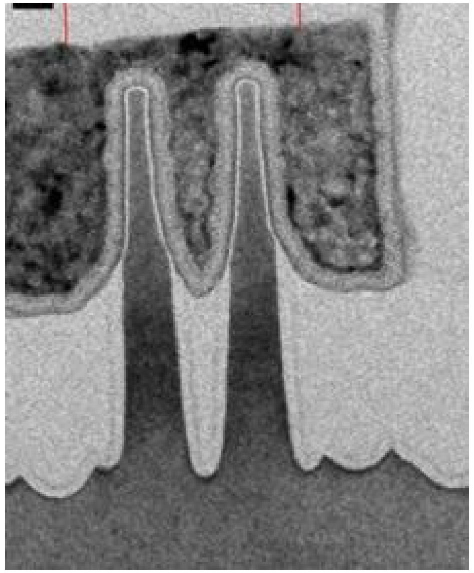
(Dennard’s scaling ended somewhere on this picture)



Research vs. Production Devices

10nm device (Intel), circa 2003

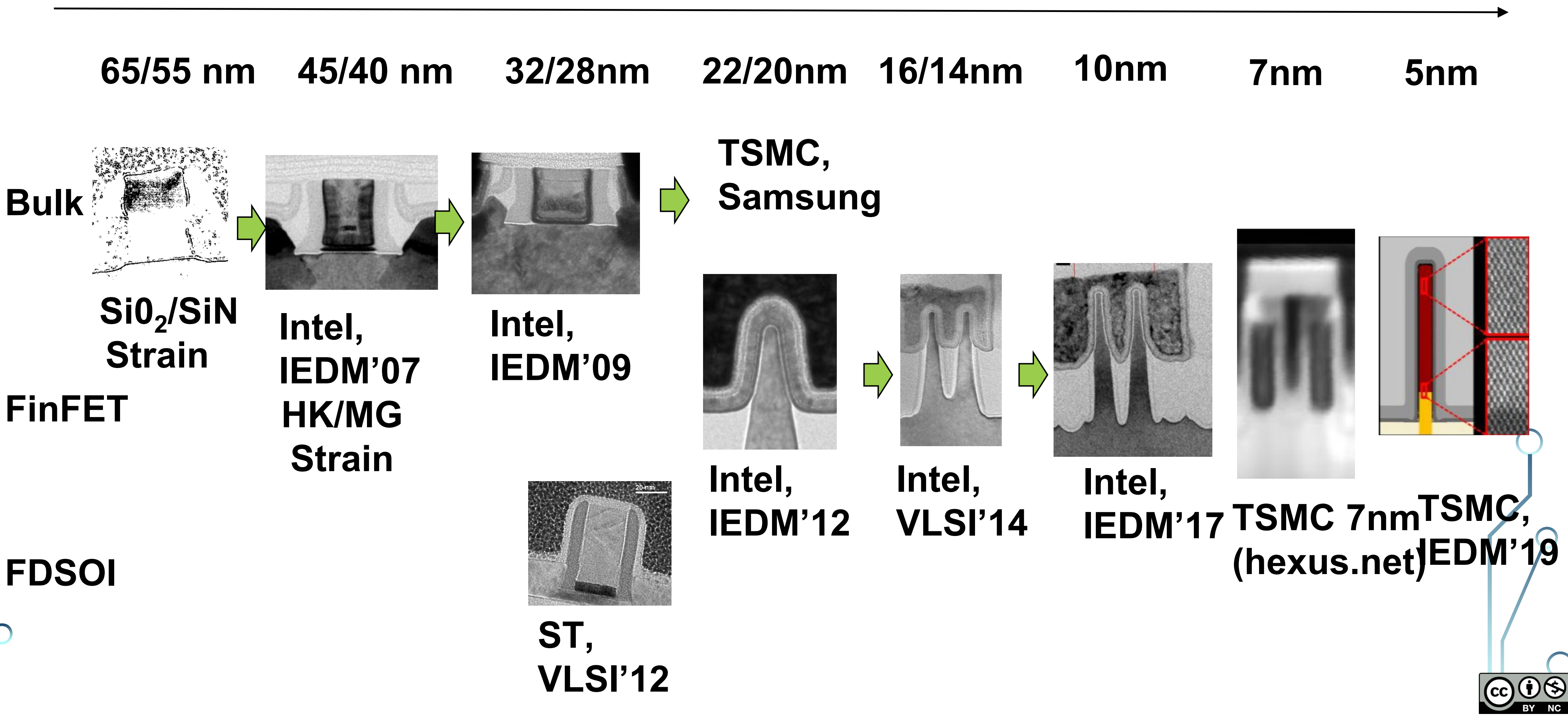




10nm node (Intel), IEDM’2017

Transistors are Changing

- From bulk to finFET and FDSOI



65/55 nm

45/40 nm

32/28nm

22/20nm

16/14nm

10nm

7nm

5nm

Bulk

SiO_2/SiN Strain

Intel, IEDM'07 HK/MG Strain

Intel, IEDM'09

TSMC, Samsung

FinFET

Intel, IEDM'12

Intel, VLSI'14

Intel, IEDM'17

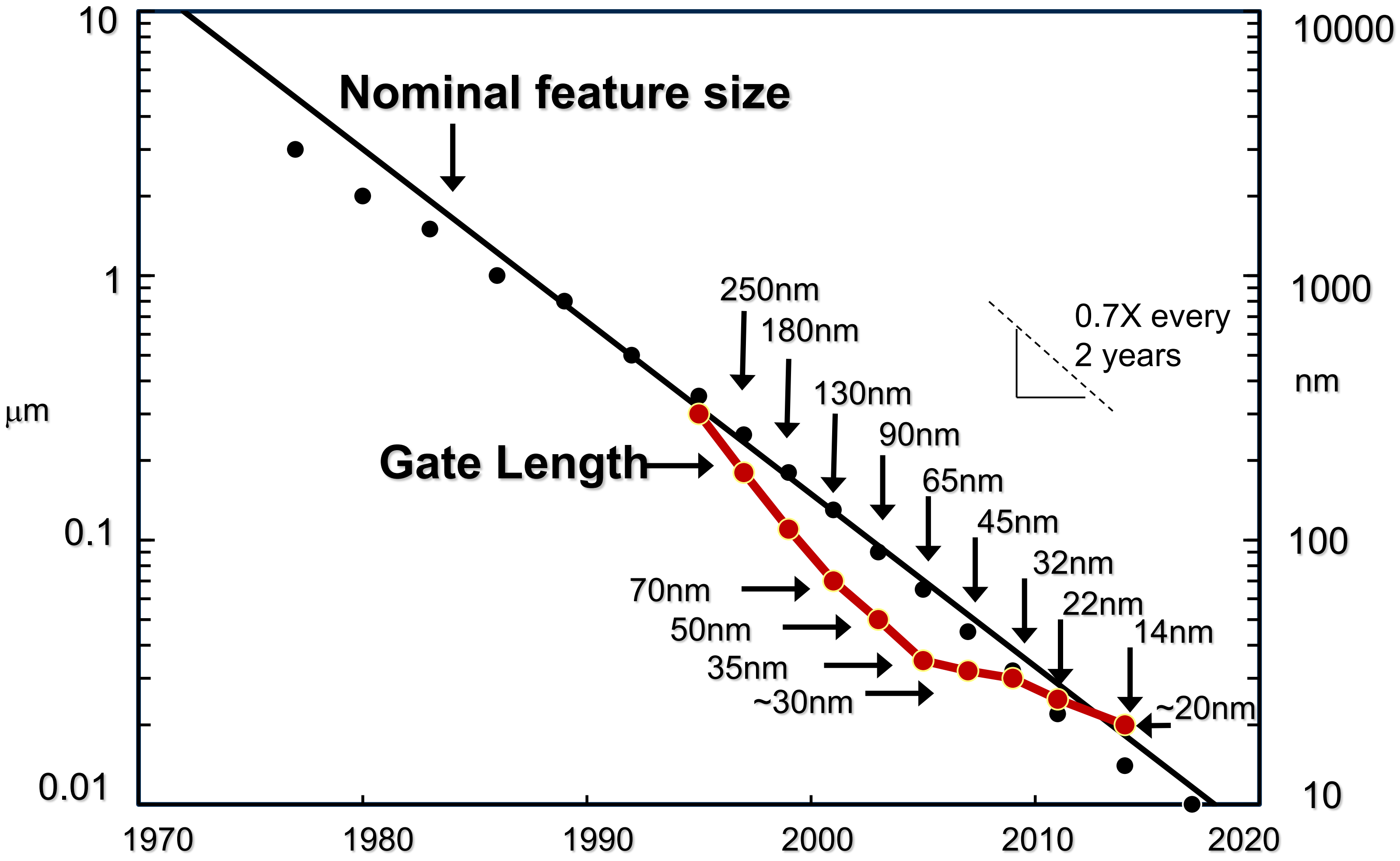
TSMC 7nm (hexus.net)

TSMC, IEDM'19

FDSOI

ST, VLSI'12

Physical Gate Scaling

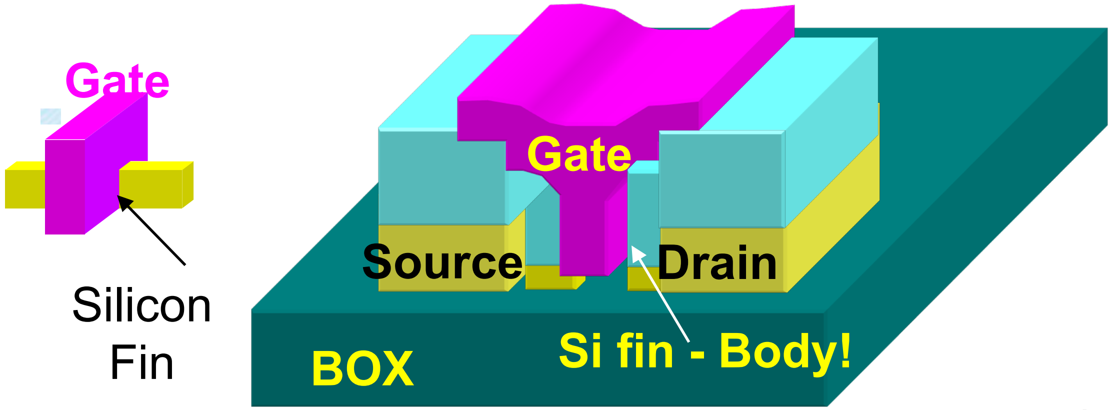


Changes in slope at 250nm, 45nm

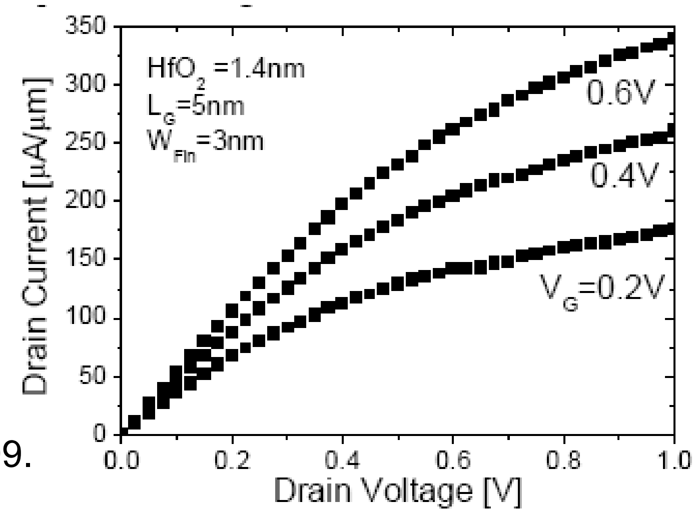
Minimum Leff for 5nm finFET is ~15nm

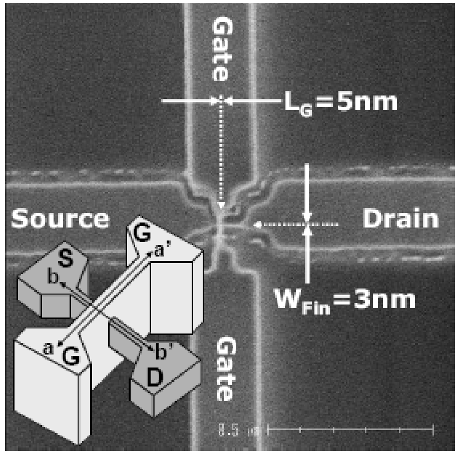
Source: Intel, IEDM presentations

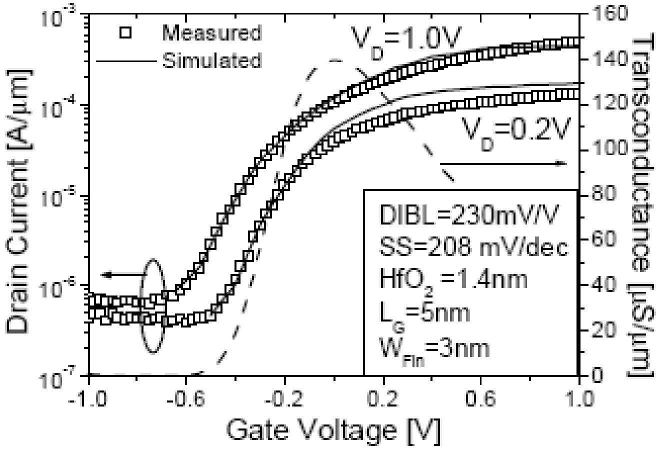
Sub-5nm FinFET



X. Huang, et al, IEDM’1999.



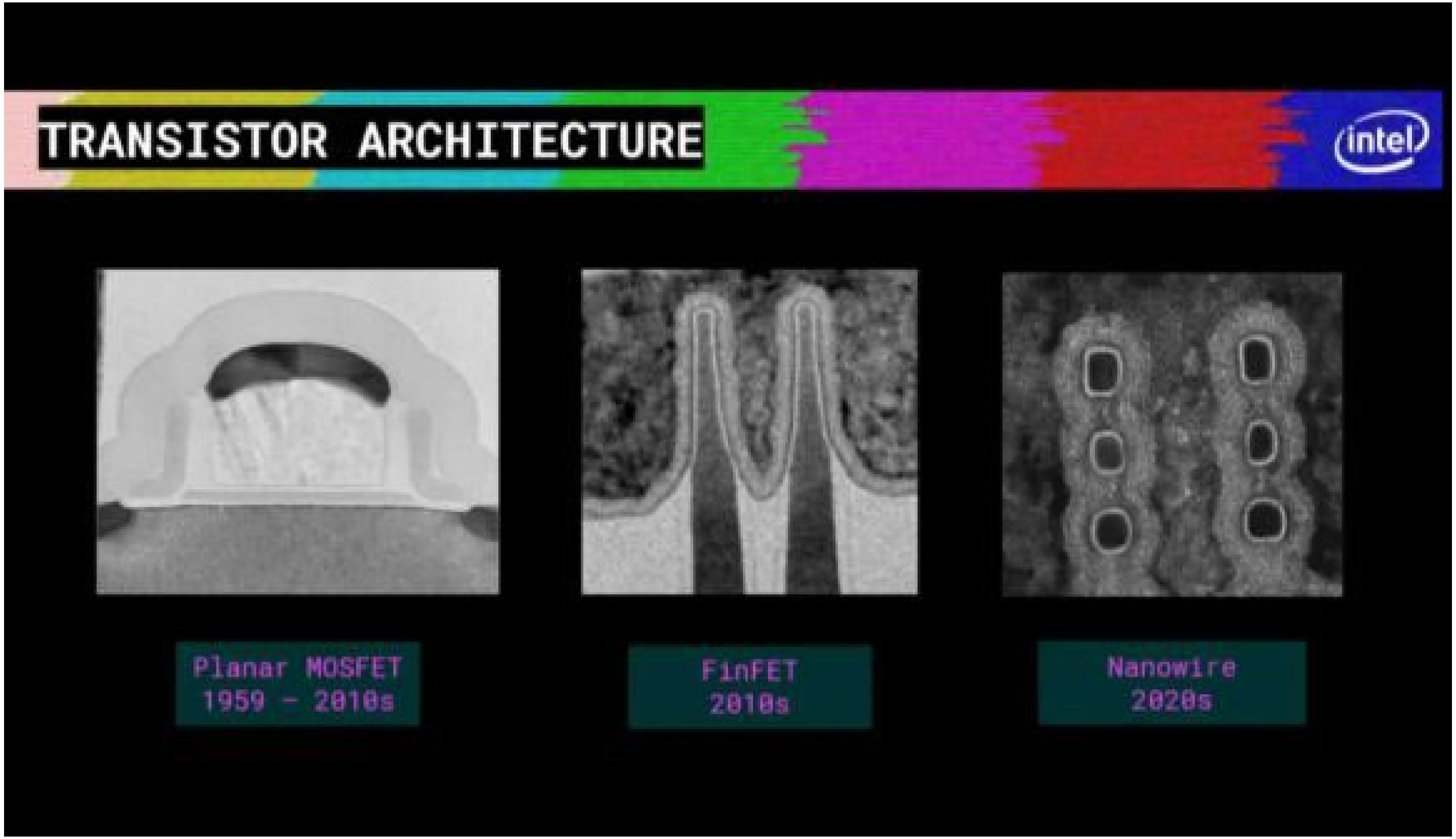




Lee, VLSI Technology, 2006

Beyond 5nm

- Gate-all-around transistors/nanowires



TRANSISTOR ARCHITECTURE

intel

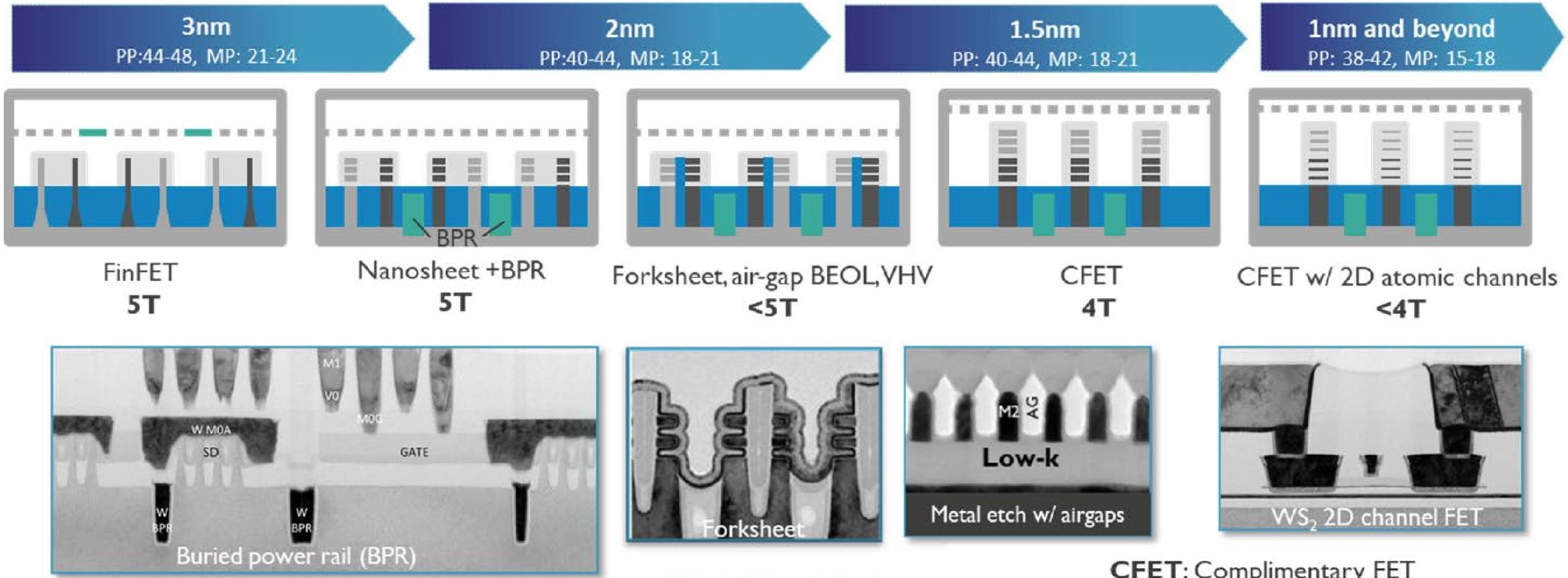
Planar MOSFET
1959 - 2010s

FinFET
2010s

Nanowire
2020s

J. Keller, EECS Colloquium, UC Berkeley, Sept 2019.

Current Perspective for <5nm



PP: poly pitch (nm)
MP: densest metal pitch (nm)

CFET: Complimentary FET
BPR: Buried power rail
VHV:Vertical-Horizontal-Vertical std cell architecture

- Samavedam, et al, IEDM’20

Pitch Scaling

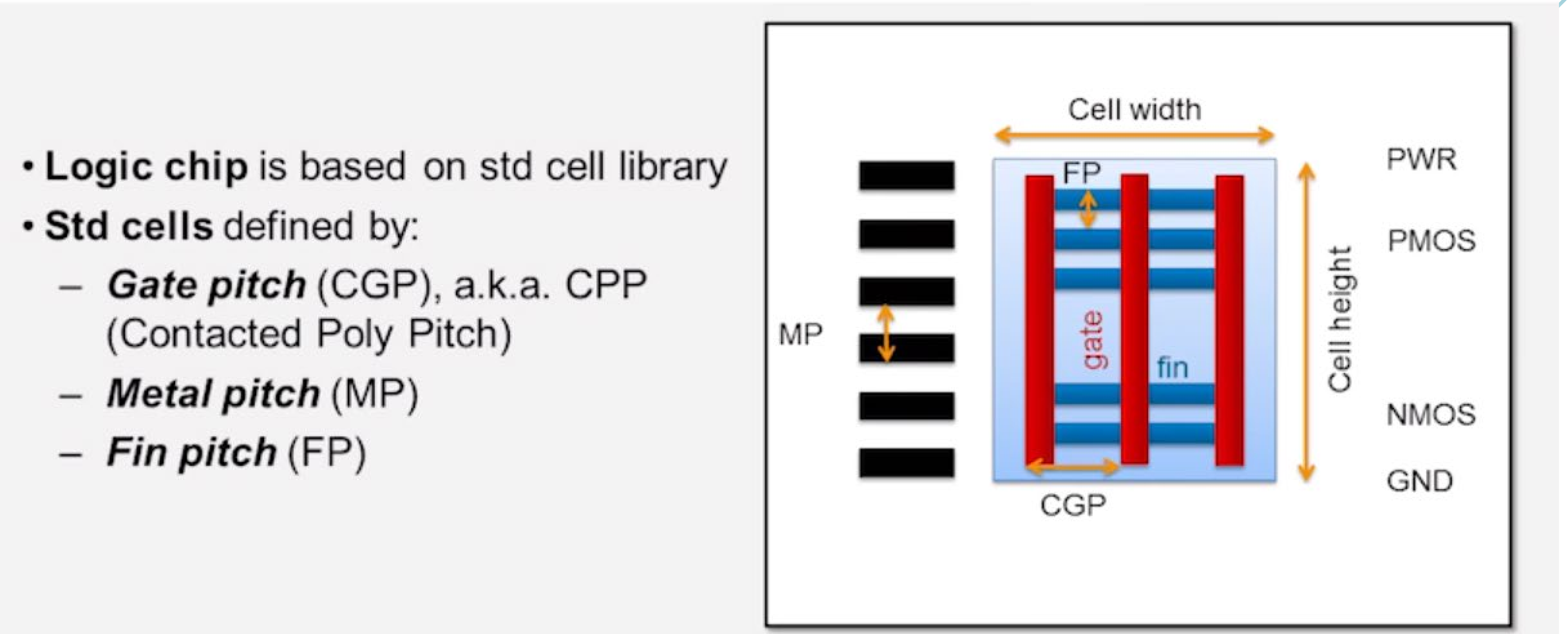
Intel	45nm	32nm	22nm	14nm	10nm
Contacted gate pitch	160nm	112.5nm	90nm	70nm	54nm
Shrink		0.7	0.8	0.78	0.77

- **Clearly not 0.7 anymore…**
- **But (Intel 14nm, Natarajan, IEDM'14)**
  - Fin pitch: 42nm (0.7x shrink)
  - Metal 0: 56nm (-)
  - Metal 1: 70nm (0.78x)
  - Metal 2: 52nm (0.65x)

- **Intel's metric:**
  - CPP x MXP
  - 0.78 x 0.65 = 0.5!
- **CPP & FP matter more**

Understanding the scaling terminology

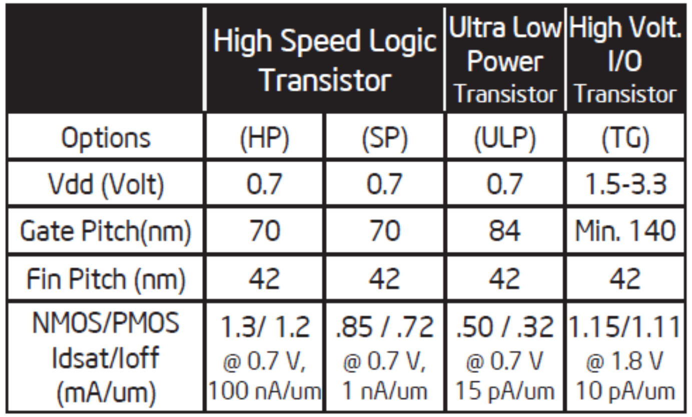
Source: Synopsys



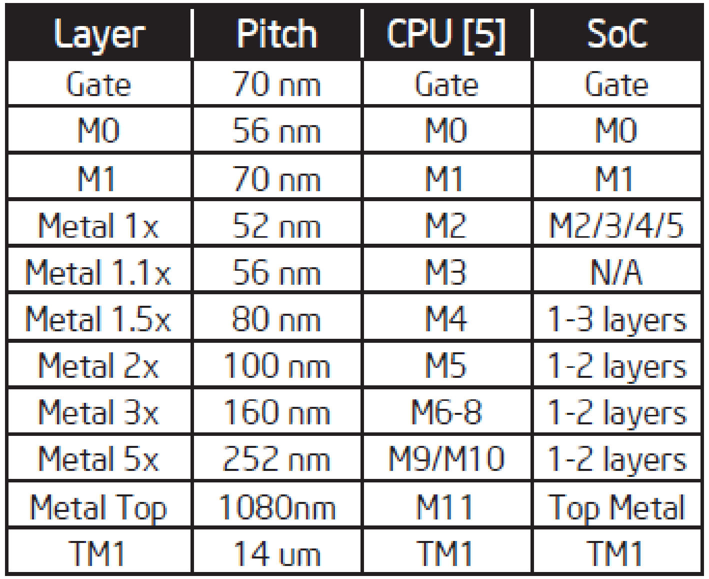
- **Logic chip** is based on std cell library
- **Std cells** defined by:
  - ***Gate pitch*** (CGP), a.k.a. CPP (Contacted Poly Pitch)
  - ***Metal pitch*** (MP)
  - ***Fin pitch*** (FP)

Various Technology flavors

- Intel 14nm



	High Speed Logic Transistor		Ultra Low Power Transistor	High Volt. I/O Transistor
Options	(HP)	(SP)	(ULP)	(TG)
Vdd (Volt)	0.7	0.7	0.7	1.5-3.3
Gate Pitch(nm)	70	70	84	Min. 140
Fin Pitch (nm)	42	42	42	42
NMOS/PMOS Idsat/Ioff (mA/um)	1.3/ 1.2 @ 0.7 V, 100 nA/um	.85 / .72 @ 0.7 V, 1 nA/um	.50 / .32 @ 0.7 V 15 pA/um	1.15/1.11 @ 1.8 V 10 pA/um



Layer	Pitch	CPU [5]	SoC
Gate	70 nm	Gate	Gate
M0	56 nm	M0	M0
M1	70 nm	M1	M1
Metal 1x	52 nm	M2	M2/3/4/5
Metal 1.1x	56 nm	M3	N/A
Metal 1.5x	80 nm	M4	1-3 layers
Metal 2x	100 nm	M5	1-2 layers
Metal 3x	160 nm	M6-8	1-2 layers
Metal 5x	252 nm	M9/M10	1-2 layers
Metal Top	1080nm	M11	Top Metal
TM1	14 um	TM1	TM1

- Different foundries

Feature	Samsung 14 nm	Intel 14 nm	TSMC 16 nm
Fin pitch (nm)	48	42	48
1/3 fin pitch	16	14	16
Gate length (nm)	~30	~24	~33
Contacted gate pitch (nm)	78	70	90
Minimum metal pitch (nm)	64	52	64
6T SRAM cell area (μm^2)	0.08	0.059	0.074

Source:
Tech Insights
EETimes

Not All Technologies are Equal

Intel

Node	CPP	MxP	FP
65nm	230	230	
45nm	160	160	
32nm	112.5	112.5	
22nm	90	80	60
14nm	70	52	42
10nm	54	36	34
7nm	37	32	

Samsung

Node	CPP	MxP	FP
45nm	180	140	
32nm	130	100	
28nm	115	90	
20LPE	90	80	60
14LPE	78	64	48
10LPE	68	48	42
7LPP	54	36	27

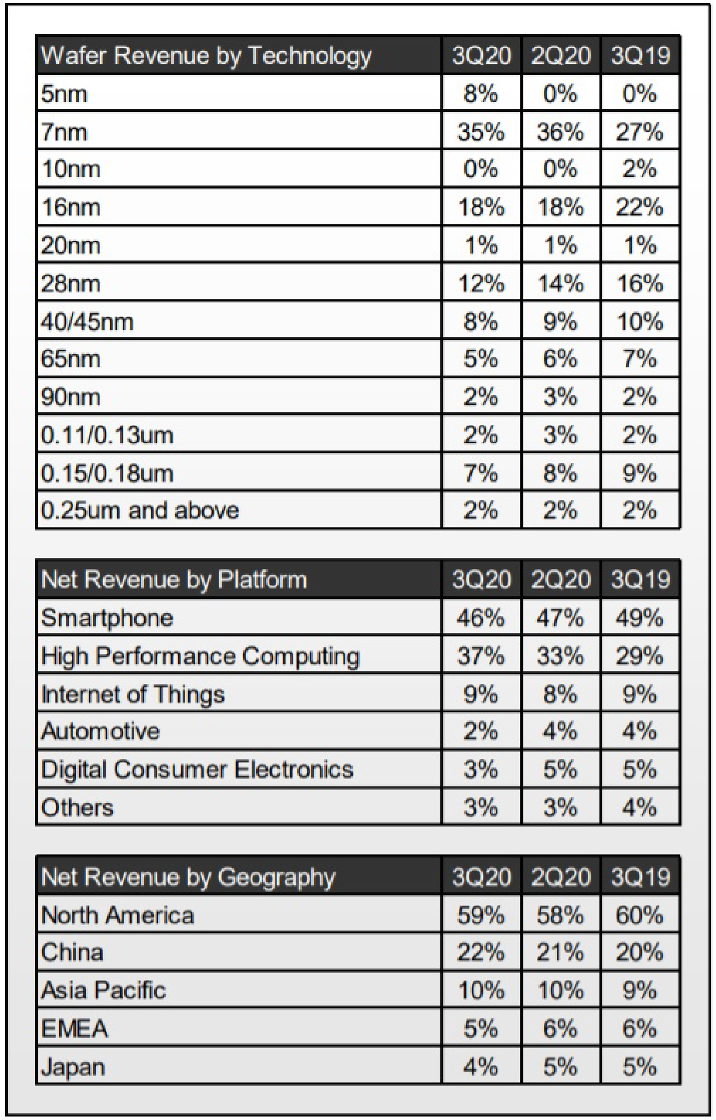
TSMC

Node	CPP	MxP	FP
45nm	190	140	
40nm	170	130	
28nm	120	90	
20SoC	90	64	
16FF	90	64	48
16FFC	96	64	48
10FF	66	44	36
7FF	57	40	30
5FF	50	28	

- **CPP = Contacted poly pitch**
- **MxP = Minimum metal pitch**
- **FP = Fin pitch**

Source:
A. Wei, TechInsights
IEDM'17, IEDM'19, WikiChip,
SemiWiki'20

Many Nodes Co-Exist



Wafer Revenue by Technology	3Q20	2Q20	3Q19
5nm	8%	0%	0%
7nm	35%	36%	27%
10nm	0%	0%	2%
16nm	18%	18%	22%
20nm	1%	1%	1%
28nm	12%	14%	16%
40/45nm	8%	9%	10%
65nm	5%	6%	7%
90nm	2%	3%	2%
0.11/0.13um	2%	3%	2%
0.15/0.18um	7%	8%	9%
0.25um and above	2%	2%	2%

Net Revenue by Platform	3Q20	2Q20	3Q19
Smartphone	46%	47%	49%
High Performance Computing	37%	33%	29%
Internet of Things	9%	8%	9%
Automotive	2%	4%	4%
Digital Consumer Electronics	3%	5%	5%
Others	3%	3%	4%

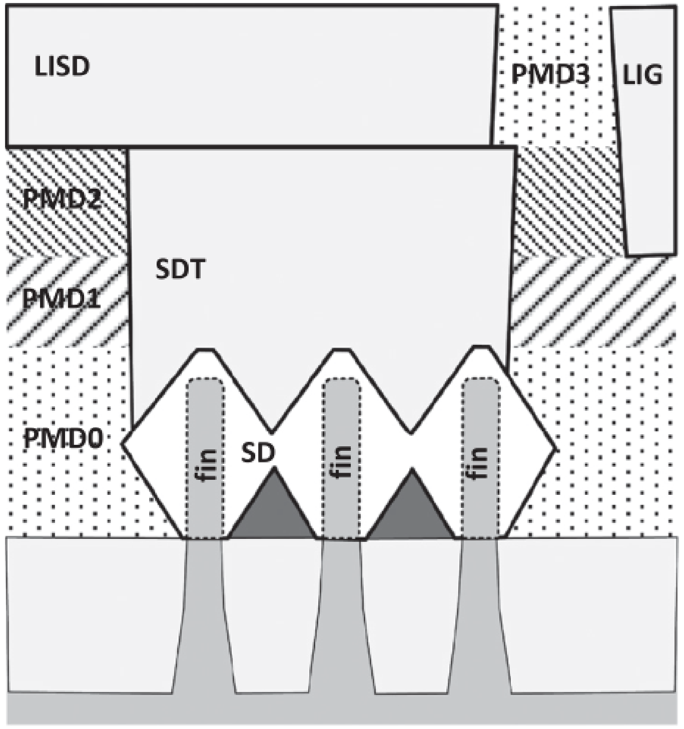
Net Revenue by Geography	3Q20	2Q20	3Q19
North America	59%	58%	60%
China	22%	21%	20%
Asia Pacific	10%	10%	9%
EMEA	5%	6%	6%
Japan	4%	5%	5%

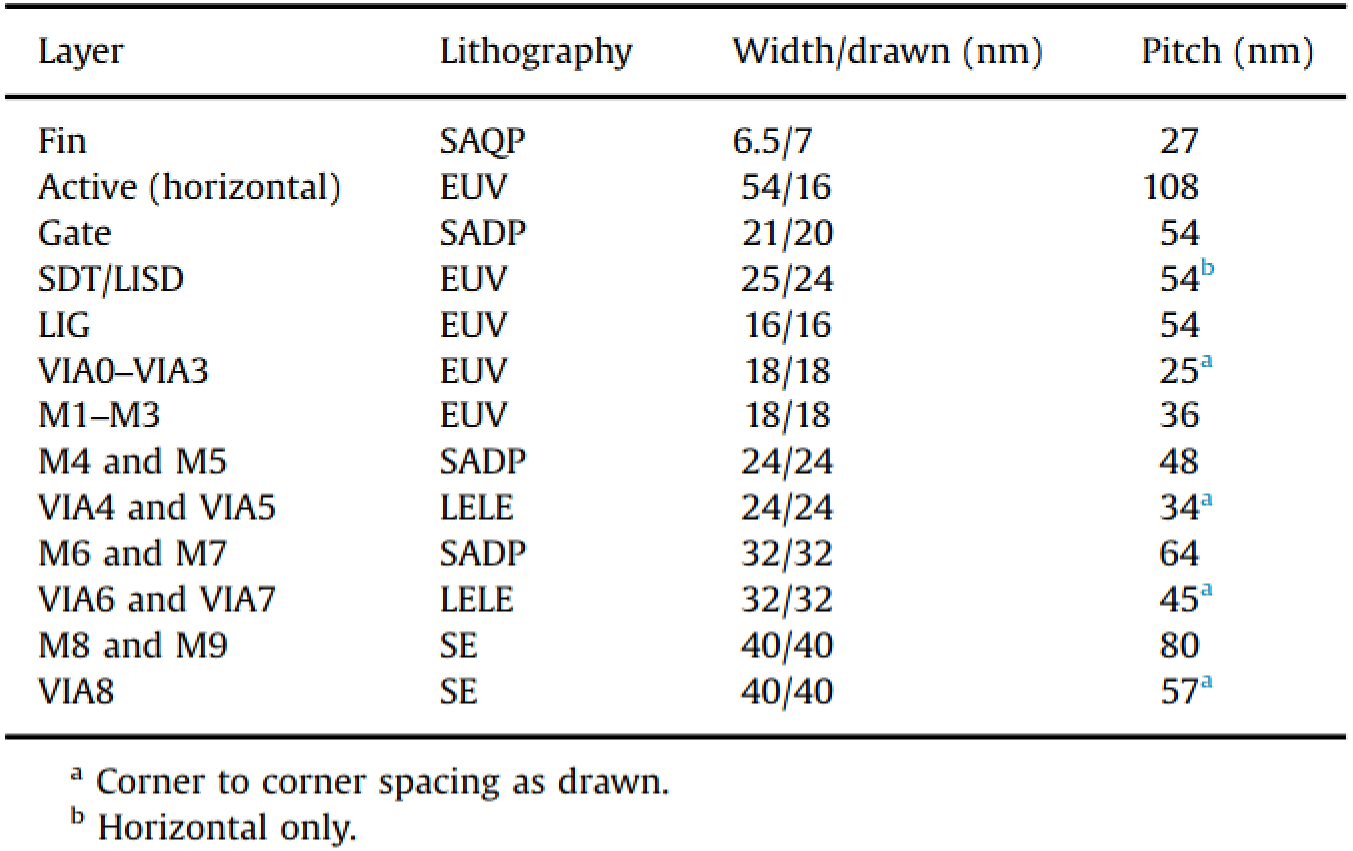
- TSMC revenue per node

  https://semiwiki.com/semiconductor-manufacturers/292174-tsmc-sets-the-stage-for-a-great-2021/

ASAP7

- Predictive technology kit used in this class
  - None of the above processes, but close

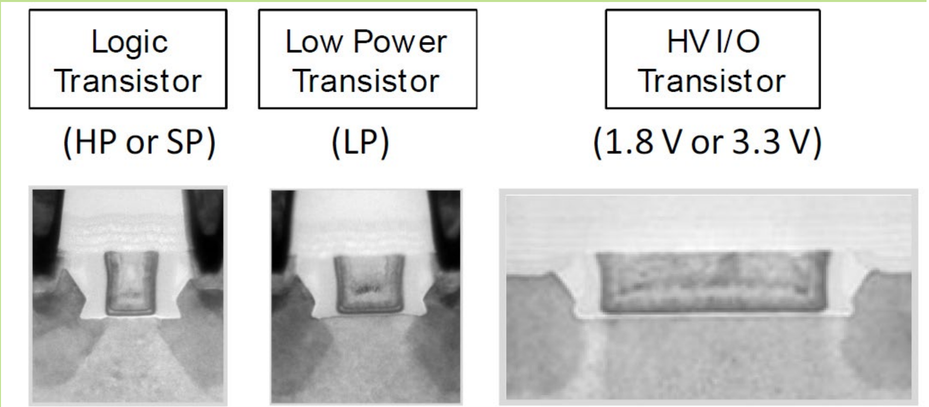


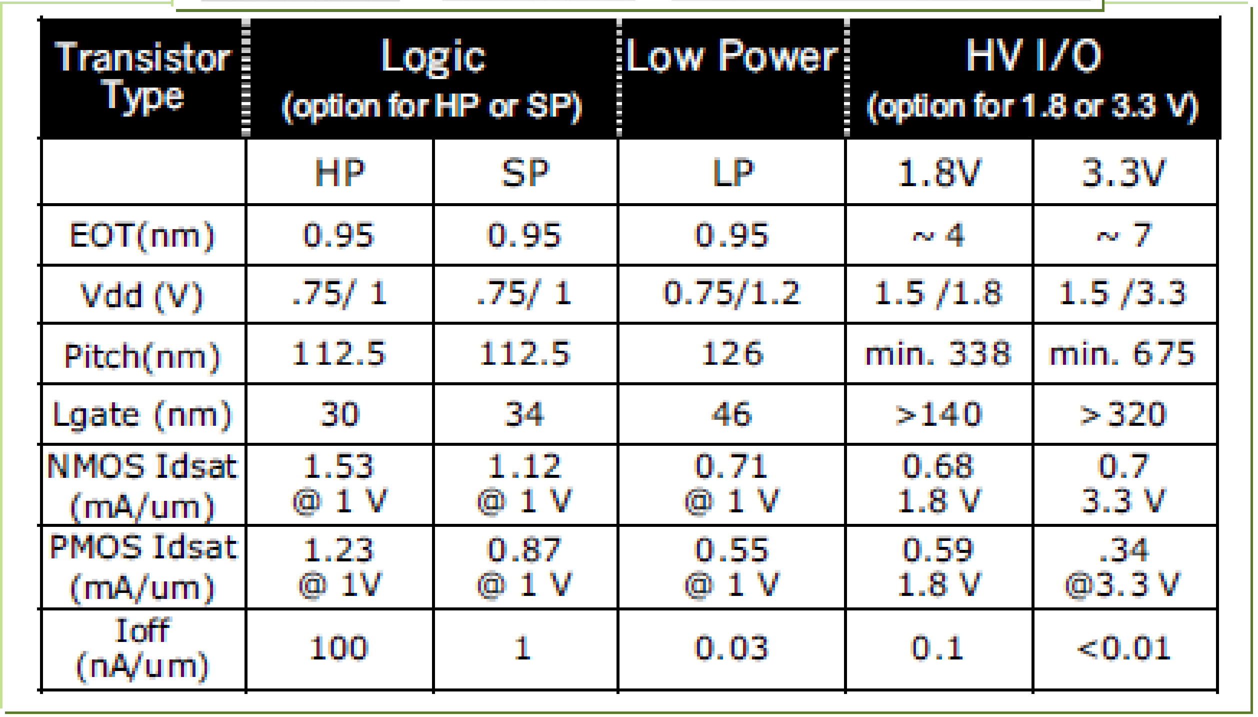


Layer	Lithography	Width/drawn (nm)	Pitch (nm)
Fin	SAQP	6.5/7	27
Active (horizontal)	EUV	54/16	108
Gate	SADP	21/20	54
SDT/LISD	EUV	25/24	54[b]
LIG	EUV	16/16	54
VIA0–VIA3	EUV	18/18	25[a]
M1–M3	EUV	18/18	36
M4 and M5	SADP	24/24	48
VIA4 and VIA5	LELE	24/24	34[a]
M6 and M7	SADP	32/32	64
VIA6 and VIA7	LELE	32/32	45[a]
M8 and M9	SE	40/40	80
VIA8	SE	40/40	57[a]

[a] Corner to corner spacing as drawn.
[b] Horizontal only.

32nm Technology Flavors (Intel)





Transistor Type	Logic (option for HP or SP)		Low Power	HV I/O (option for 1.8 or 3.3 V)	
	HP	SP	LP	1.8V	3.3V
EOT(nm)	0.95	0.95	0.95	~ 4	~ 7
Vdd (V)	.75/ 1	.75/ 1	0.75/1.2	1.5 /1.8	1.5 /3.3
Pitch(nm)	112.5	112.5	126	min. 338	min. 675
Lgate (nm)	30	34	46	>140	> 320
NMOS Idsat (mA/um)	1.53 @ 1 V	1.12 @ 1 V	0.71 @ 1 V	0.68 1.8 V	0.7 3.3 V
PMOS Idsat (mA/um)	1.23 @ 1V	0.87 @ 1 V	0.55 @ 1 V	0.59 1.8 V	.34 @3.3 V
Ioff (nA/um)	100	1	0.03	0.1	<0.01

C.-H. Jan, IEDM'09, P. VanDerVoorn, VLSI Tech'10

5nm Flavors

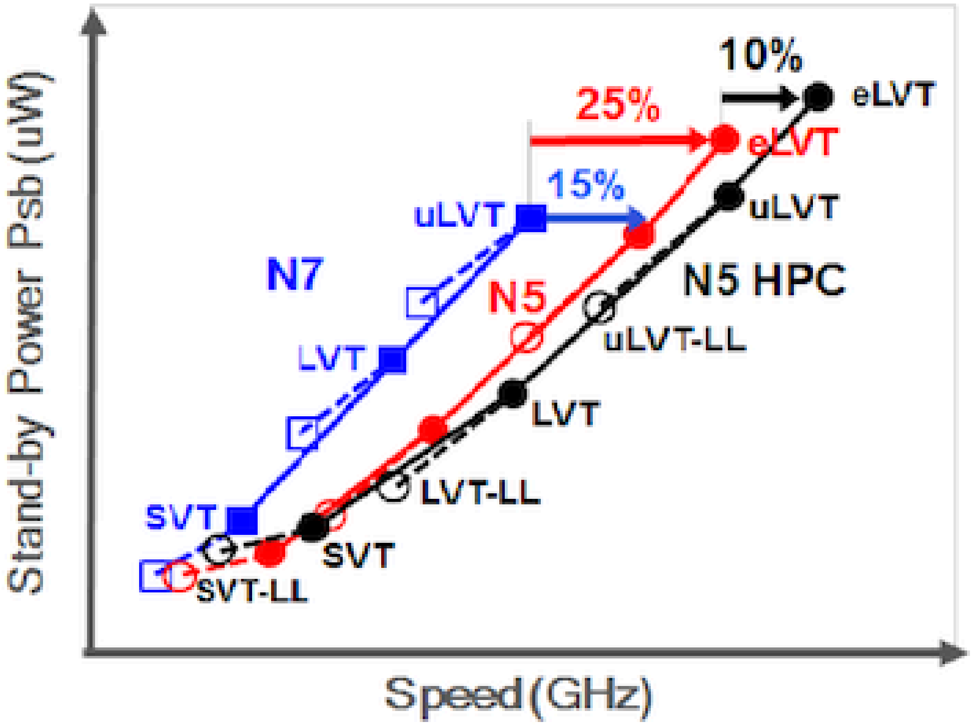
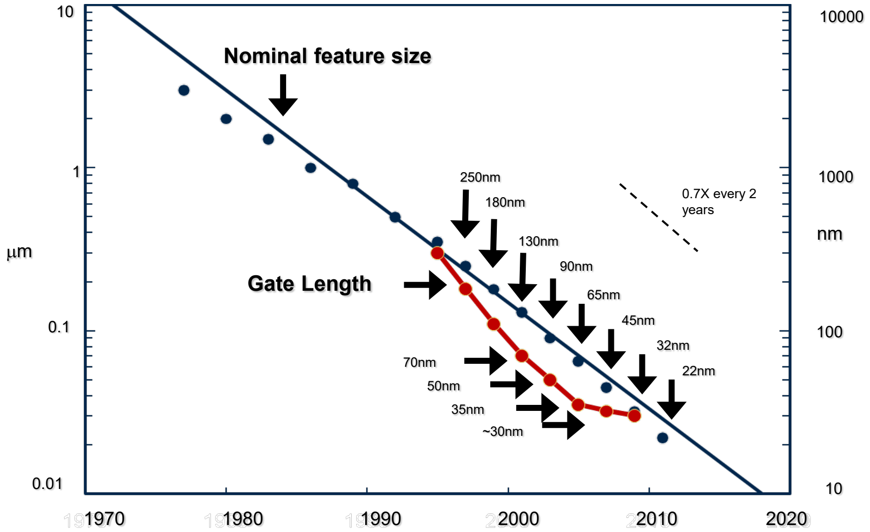
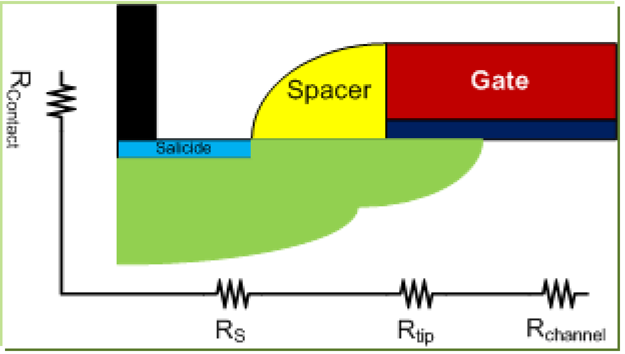


Fig.3 The 5nm also offers a set of critical HPC features. Extremely LVT (eLVT) for 25% faster peak speed over 7nm, and HPC 3-fin standard cell for additional 10% performance.

TSMC, IEDM’19

Lg, R, C scaling

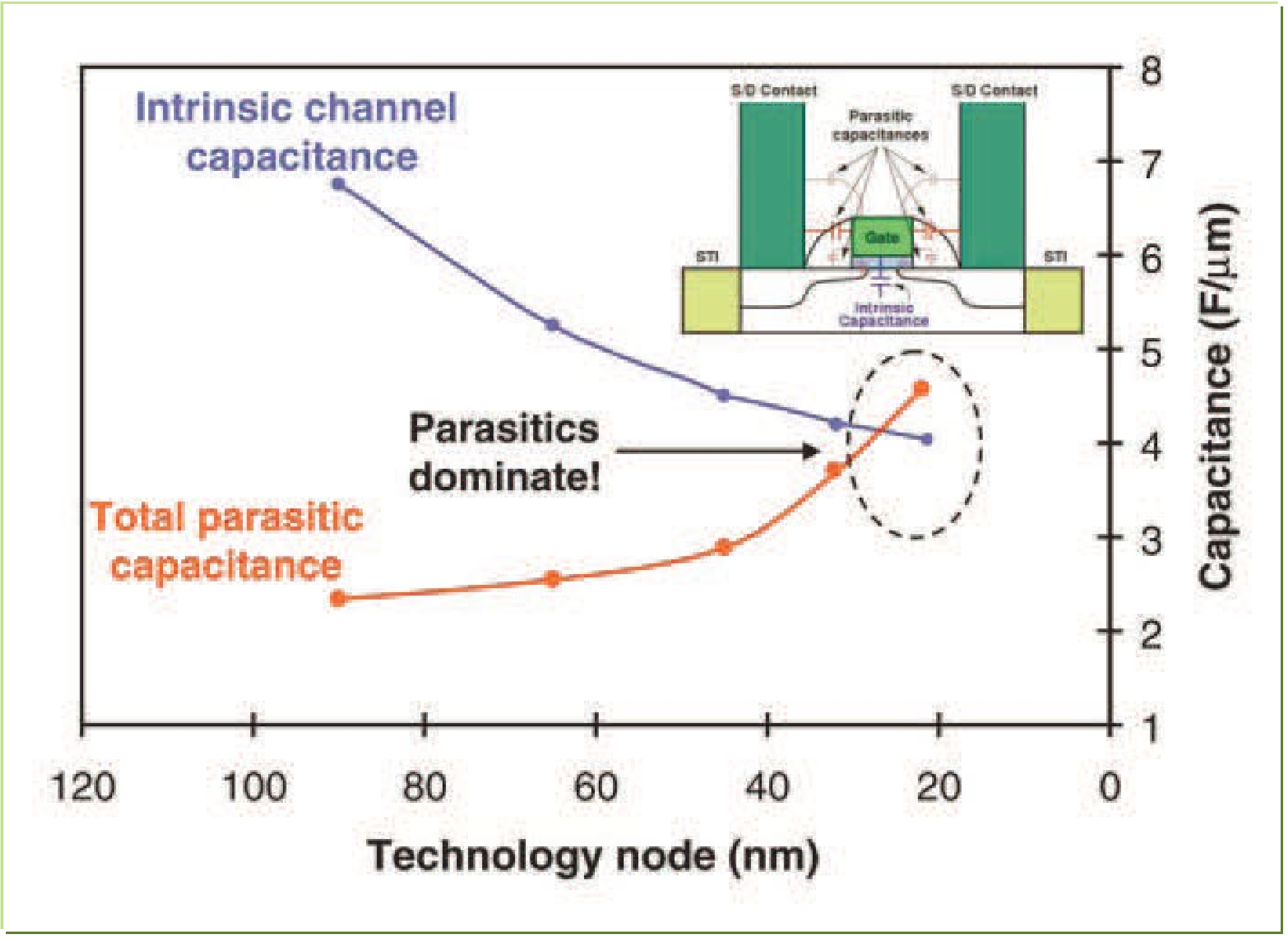




- With scaling L, need to scale up doping - scale junction depth (control leakage) – S/D resistance goes up
- External resistance limits current

$$I_D \approx V_{DS} / \left(R_{channel} + R_{ext}\right)$$

Parasitic Capacitance Scaling



Reality: Overlap + fringe can be 50% of $C_{channel}$ in 32nm

S. Thompson, *Materials Today*, 2006.

Next Lecture

- Lithography implications