Digital Computing Beyond Moore's Law

IPAM

UCLA

November 28, 2018

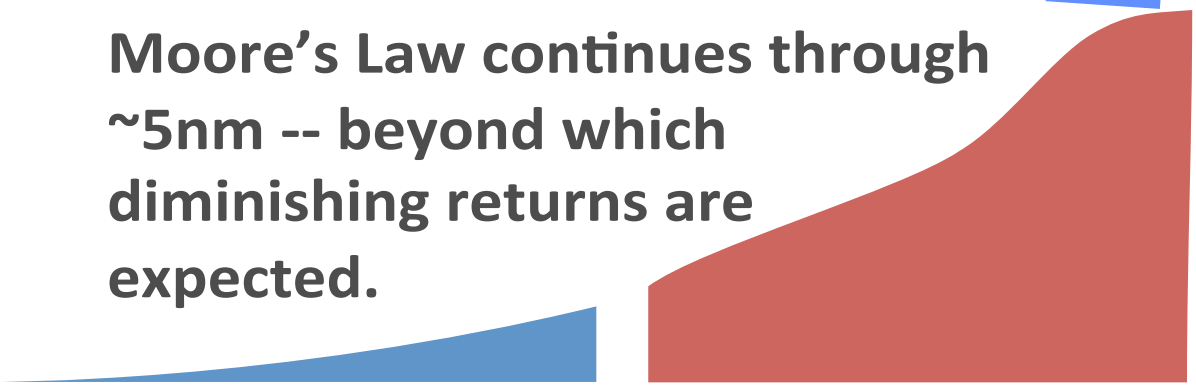
John Shalf

Lawrence Berkeley National Laboratory

Post Exsascale Landscape
MIND THE GAP!

Moore's Law
Lithography Scaling
2x increased density
2x lower power
Every 2 years!

Now – 2025
Moore's Law continues through ~5nm -- beyond which diminishing returns are expected.



2016

2016-2025

End of Moore's Law
2025-2030?

Post Moore Scaling
New materials and devices introduced to enable continued scaling of digital electronics performance and efficiency.

2025+

50 years of exponential technology scaling is an amazing thing *that is often taken for granted*

Kenichi Miura

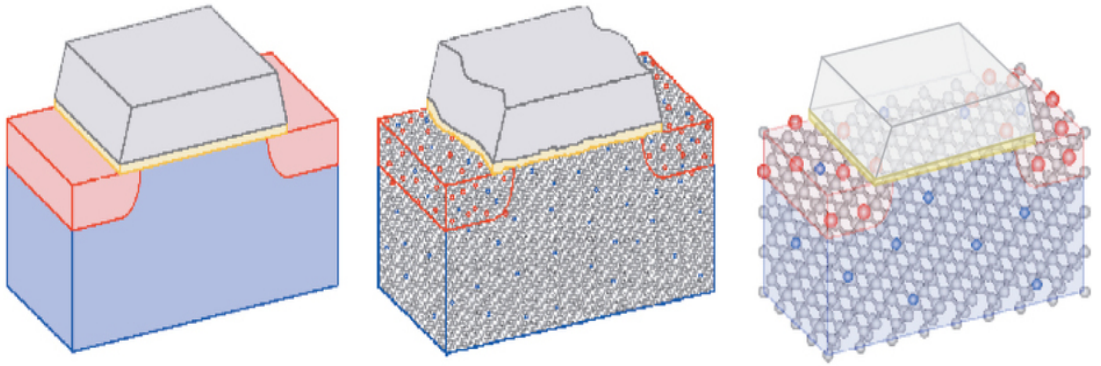
Is it the end?

"I predict Moore's Law will never end.

That way I will only be wrong once!"

Alan Kay: Communications of the ACM 1989

50 years of Sem



Atomic scale limit case for 2D Lithography Scaling

Traditional 22 nm Sub-10 nm

10 um
1 um
100 nm
10 nm
1 nm

Modern CMOS

Beginni
submicron CMOS

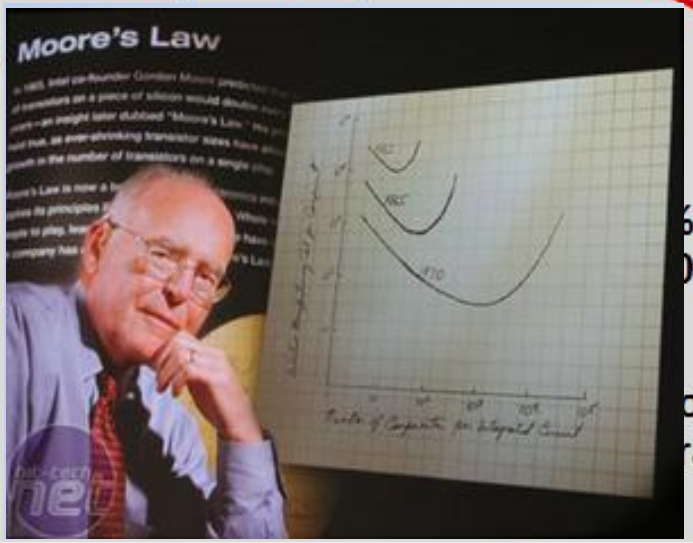
Deep UV litho

65 nm in 2006

28 nm in 2012

Presumed limit
to scaling

40 years of
scaling history



0%

over the past 40 years
rate is slowing

1970 1980 1990 2000 2010 2020 2027?

5nm

Technology Scaling Trends

Exascale in 2021… and then what?

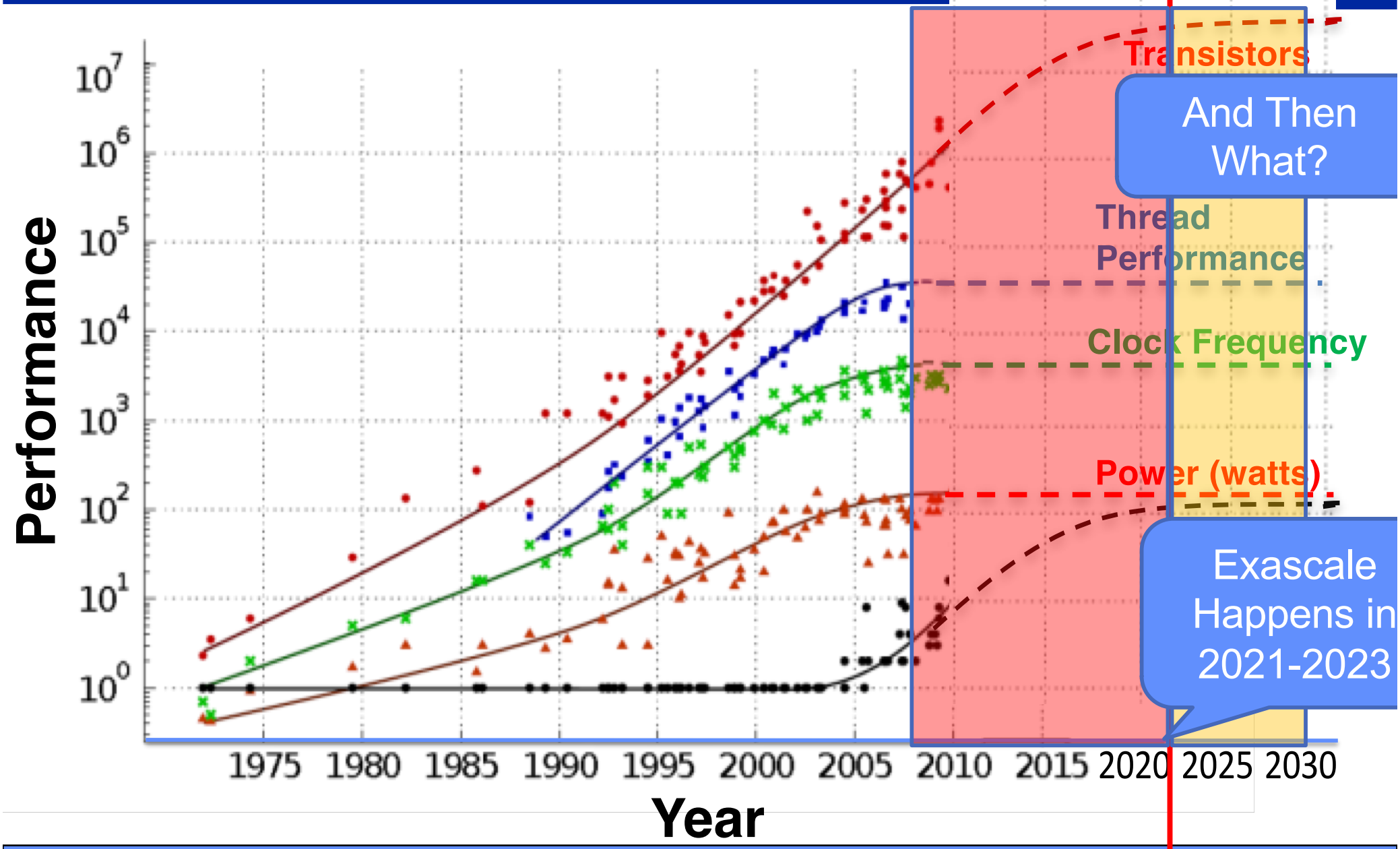


Figure courtesy of Kunle Olukotun, Lance Hammond, Herb Sutter, and Burton Smith

Innovation is the Answer!

Moore's Law is an economic theory.

There are ways to continue scaling of digital technology after the end of classical lithographic scaling

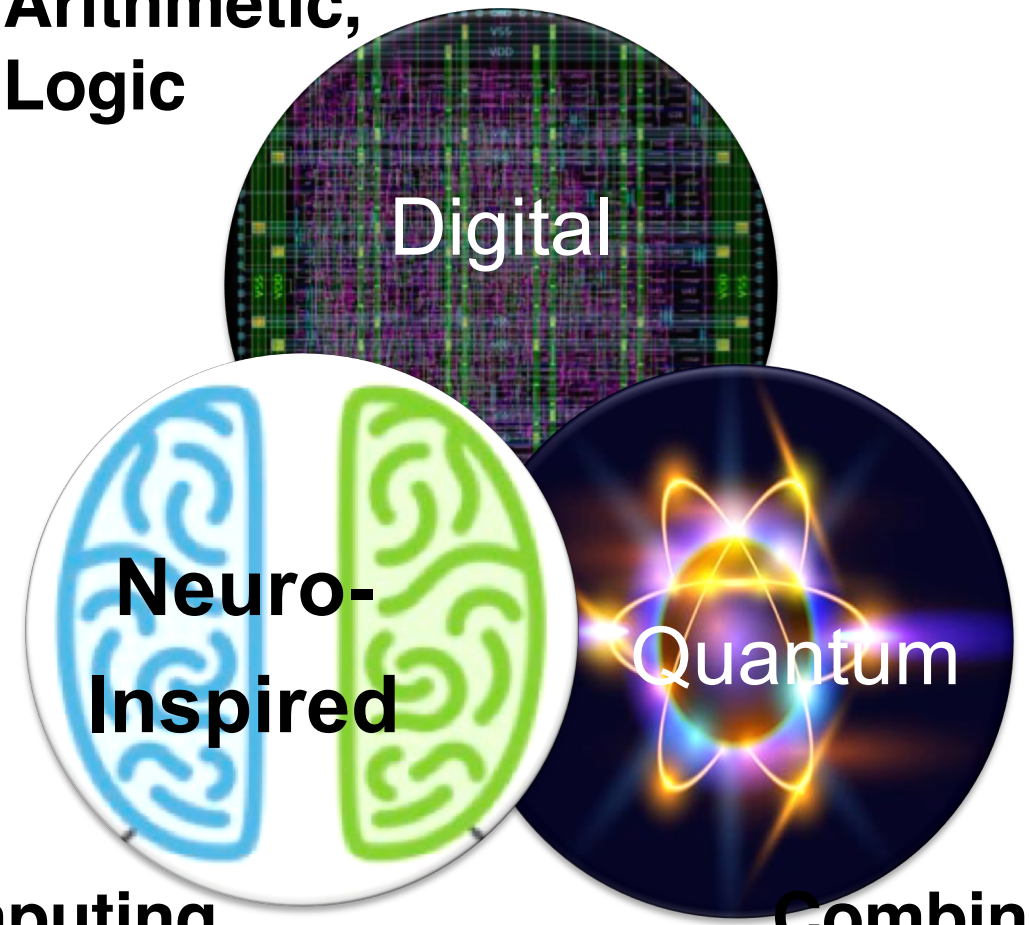
(e.g. end of Dennard Scaling in ~2004

No more exponential clock frequency scaling

Move to exponentially increasing parallelism)

Beyond Moore Computing Taxonomy

Symbolic Computation, Arithmetic, Logic

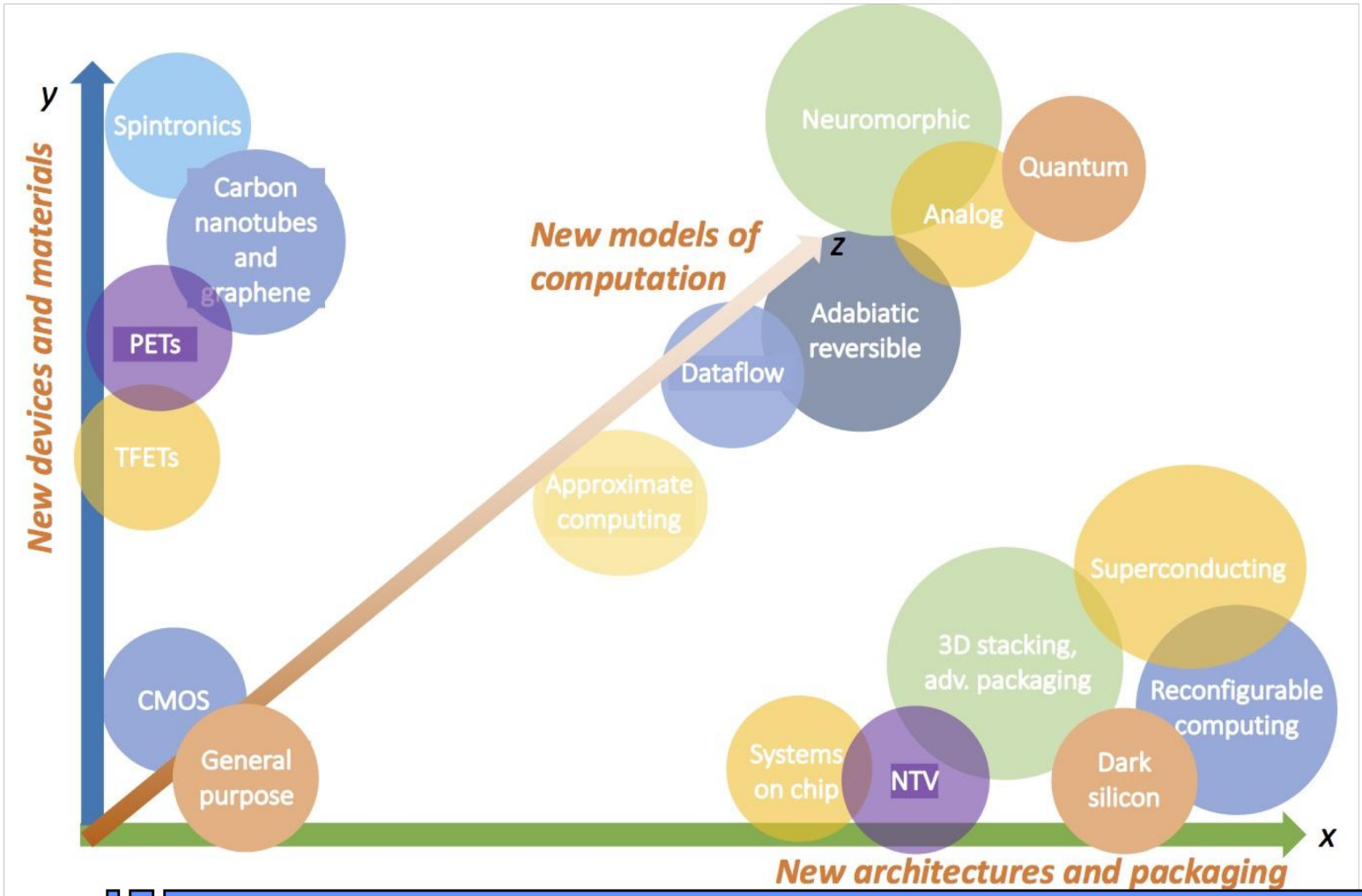


Cognitive Computing, Pattern Recognition

Combinatorial/NP, Annealing/Optimization, Simulated Atoms

Future of Computing

New Materials and Devices
20+ years (10 year lead time)



More Efficient Architectures and Packaging
The next 10 years after exascale

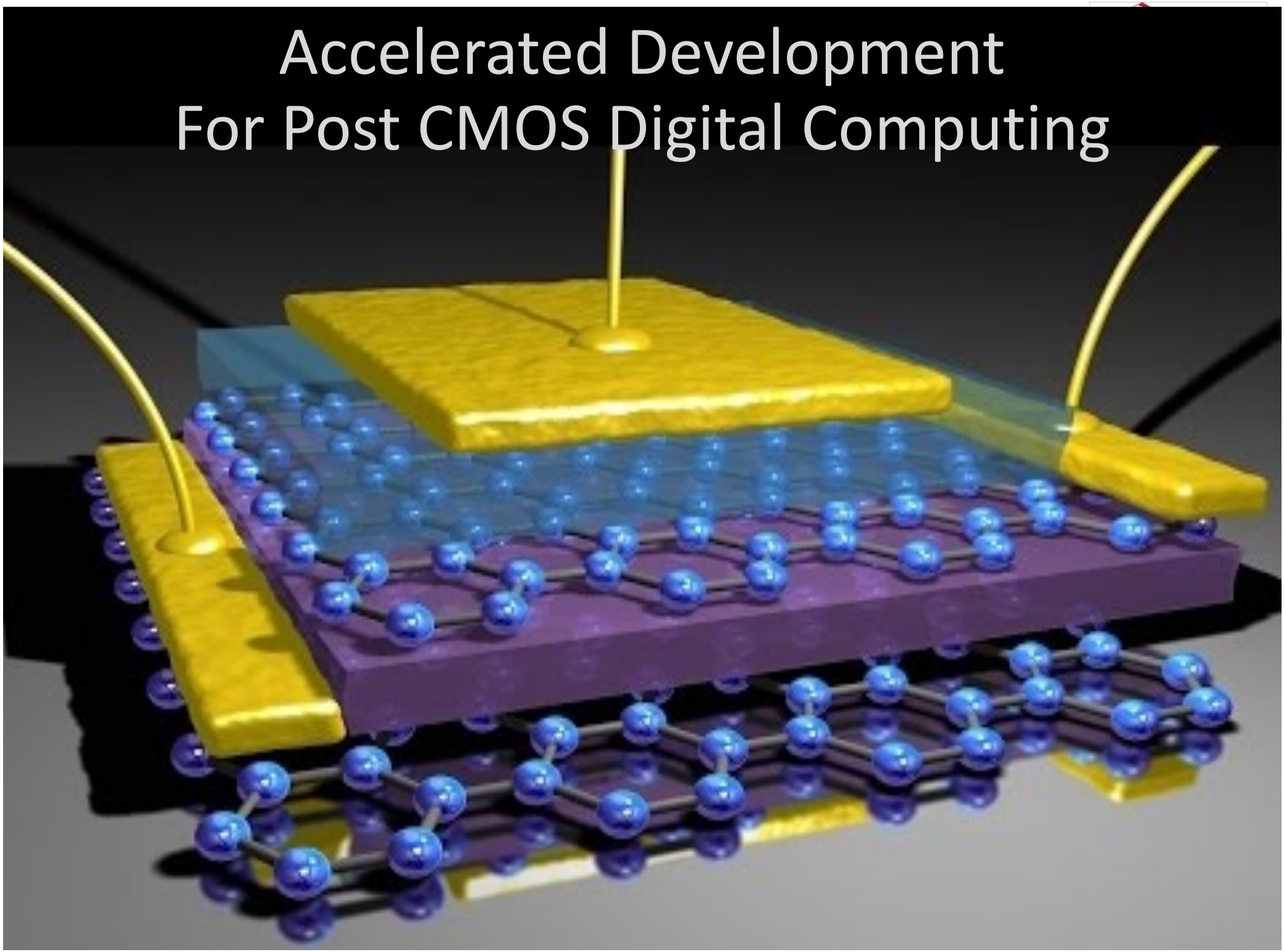
What about a new transistor?

How is that going?

TABLE 1. Summary of techology options for extending digital electronics.

Improvement Class	Technology	Timescale	Complexity	Risk	Opportunity
Architecture and software advances	Advanced energy management	Near-Term	Medium	Low	Low
	Advanced circuit design	Near-Term	High	Low	Medium
	System-on-chip specialization	Near-Term	Low	Low	Medium
	Logic specialization/dark silicon	Mid-Term	High	High	High
	Near threshold voltage (NTV) operation	Near-Term	Medium	High	High
3D integration and packaging	Chip stacking in 3D using thru-silicon vias (TSVs)	Near-Term	Medium	Low	Medium
	Metal layers	Mid-Term	Medium	Medium	Medium
	Active layers (epitaxial or other)	Mid-Term	High	Medium	High
Resistance reduction	Superconductors	Far-Term	High	Medium	High
	Crystaline metals	Far-Term	Unknown	Low	Medium
Millivolt switches (a better transistor)	Tunnel field-effect transistors (TFETs)	Mid-Term	Medium	Medium	High
	Heterogeneous semiconductors/strained silicon	Mid-Term	Medium	Medium	Medium
	Carbon nanotubes and graphene	Far-Term	High	High	High
	Piezo-electric transistors (PFETs)	Far-Term	High	High	High
Beyond transistors (new logic paradigms)	Spintronics	Far-Term	Medium	High	High
	Topological insulators	Far-Term	Medium	High	High
	Nanophotonics	Near/Far-Term	Medium	Medium	High
	Biological and chemical computing	Far-Term	High	High	High

Accelerated Development For Post CMOS Digital Computing



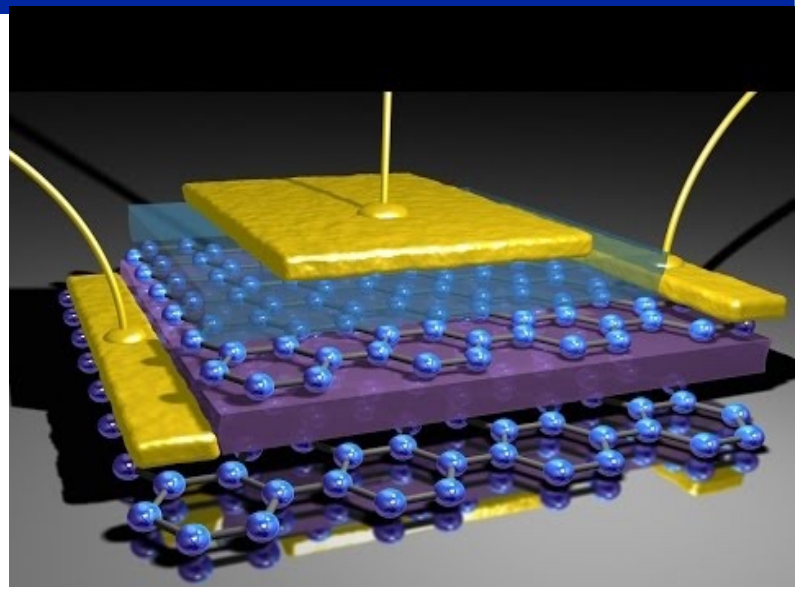
We might already be too late

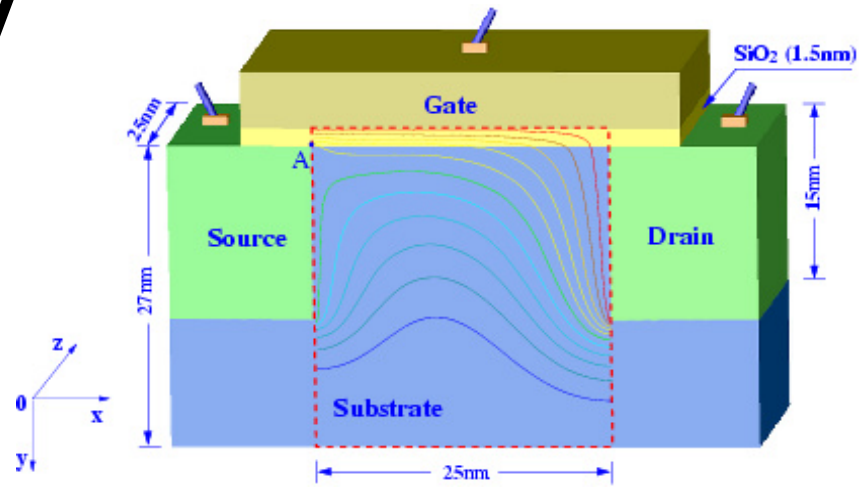
Historically it is 10 years from lab to Fab…

But lets talk about it anyways.

Borkar-Shalf Criteria for New Device Technology

1. Gain
2. Signal to Noise
3. Scalability
4. Manufacturability

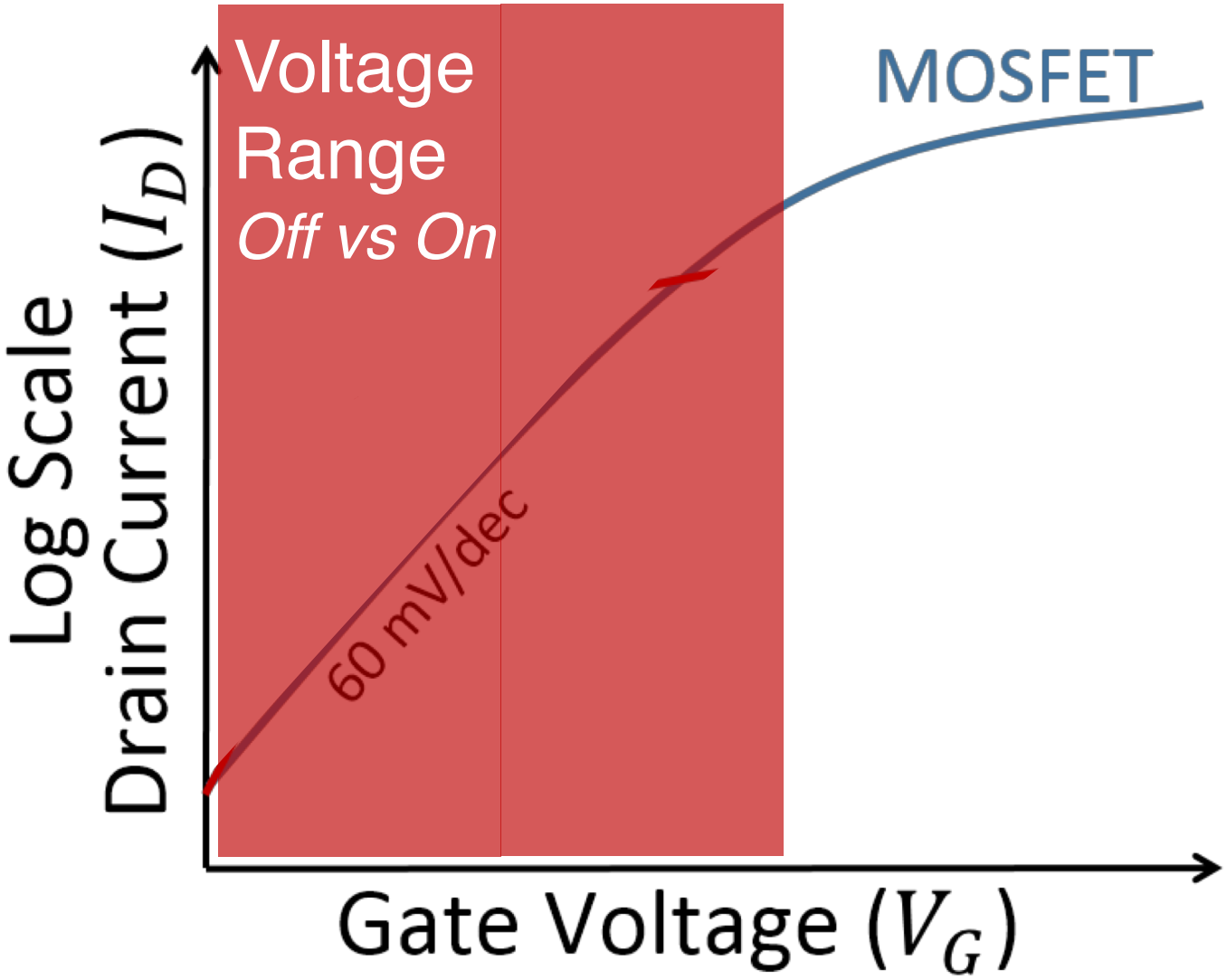




New Breakthroughs in Transistor Technology Require Fundamentally New Principles of Operation

TFET (Tunneling FET): A More sensitive switch

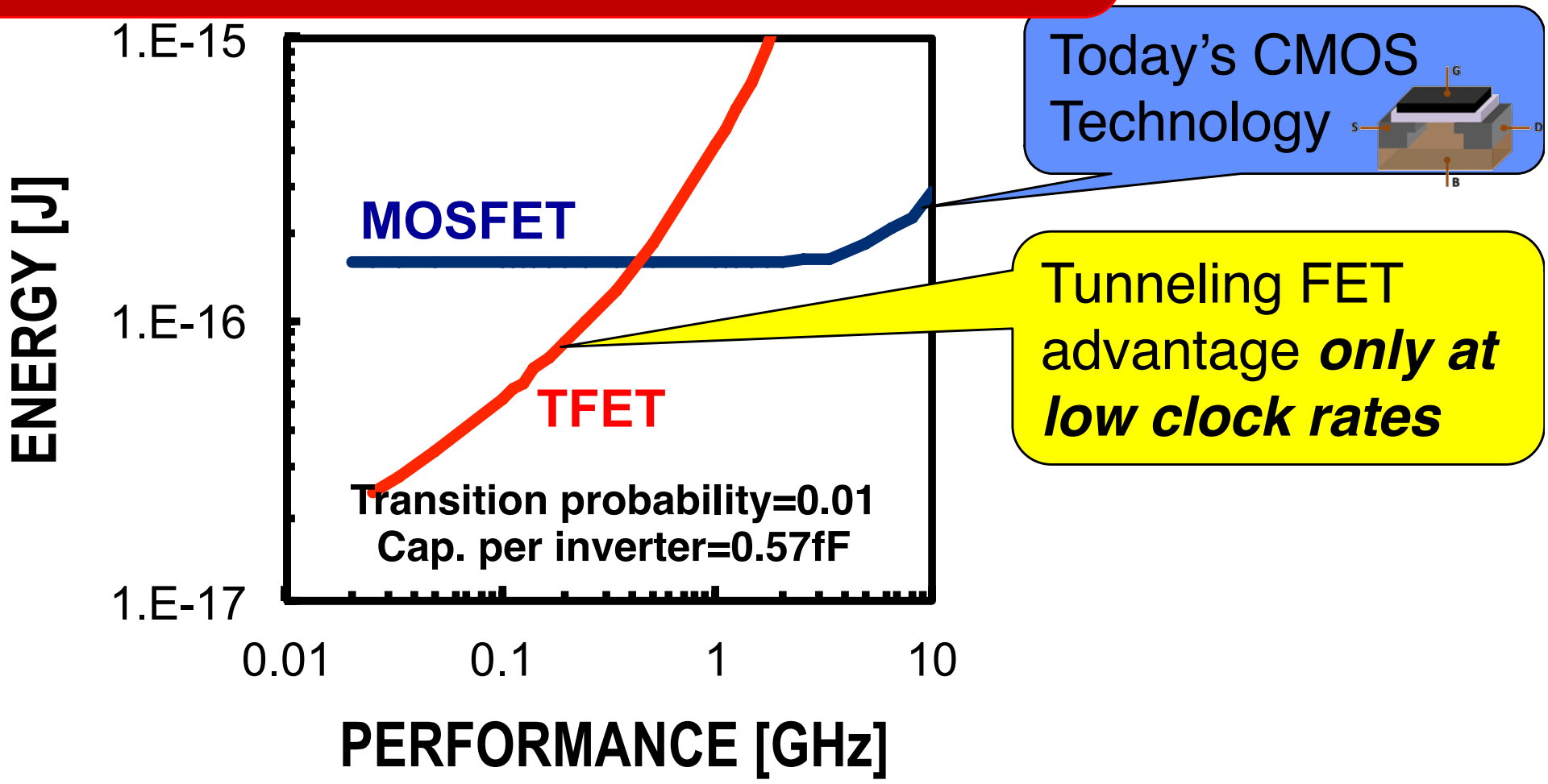
Modulated by quantum tunneling instead of thermionic emission



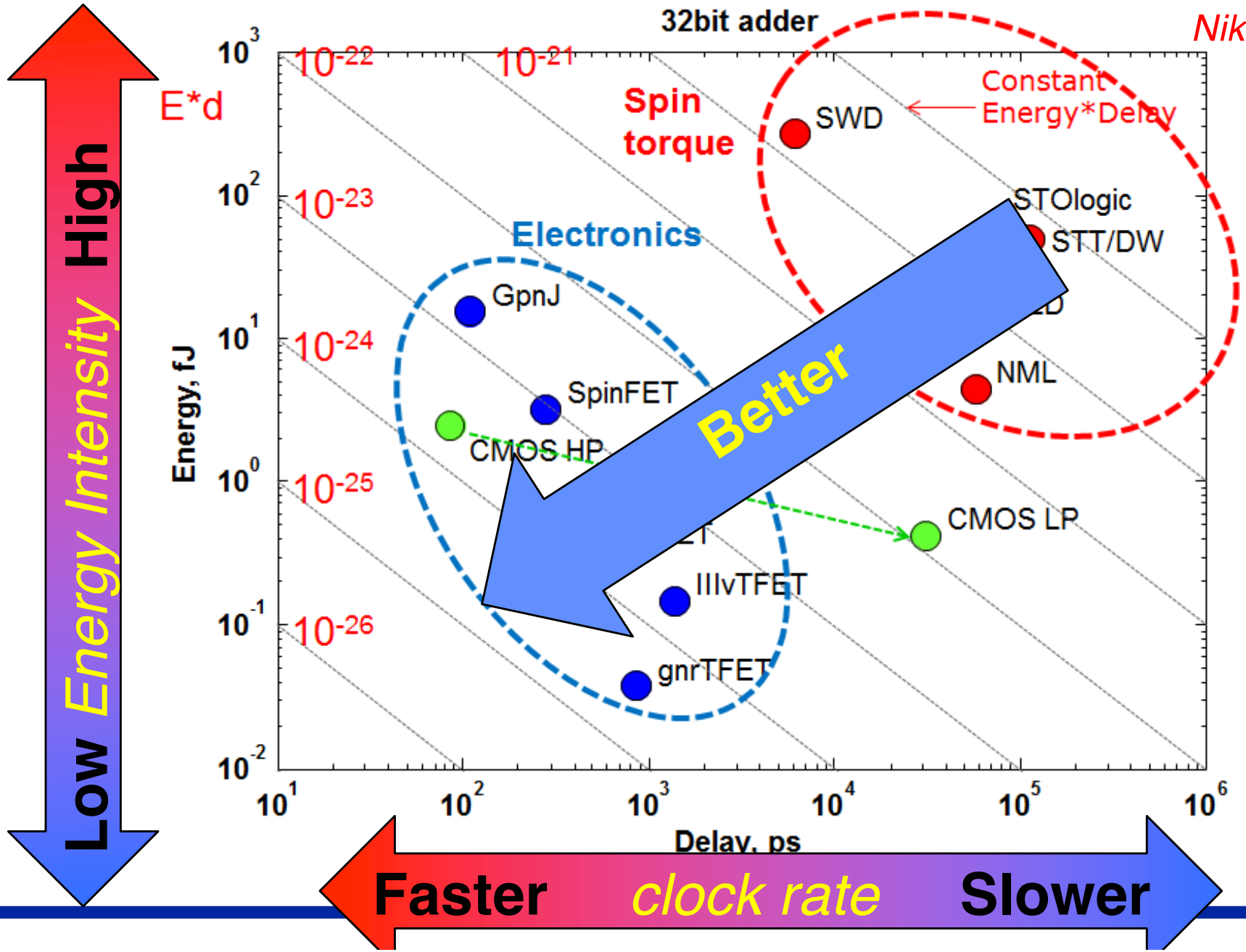
Alternatives to Conventional MOS Switches

(all require lower clock rate, and much more parallelism)

Need another 100x parallelism just to maintain status quo!

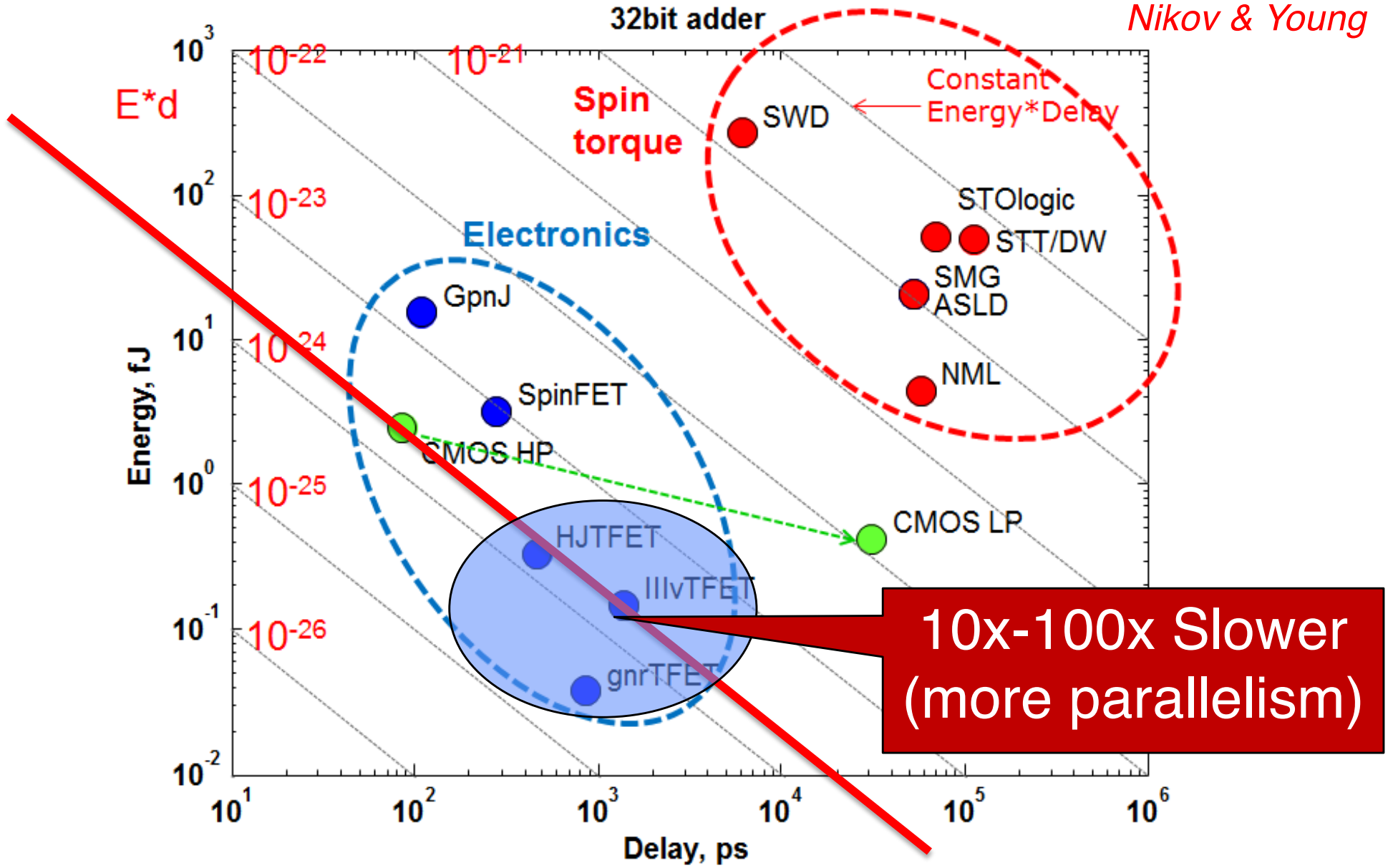


Comparing CMOS Technology Alternatives

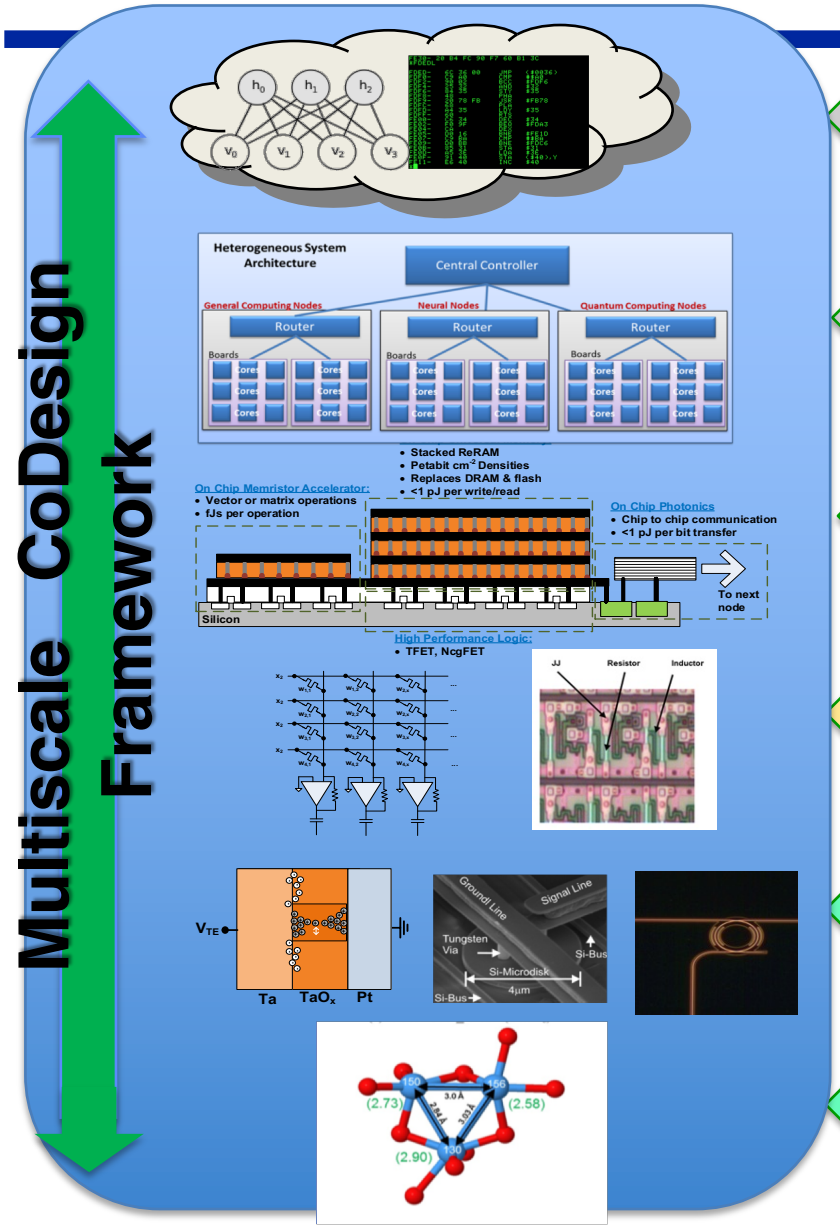


Comparing CMOS Technology Alternatives

(32 bit adder "benchmark" revolutionized field by focusing effort on device feature improvements that directly benefit the adder)



Multiscale CoDesign Framework



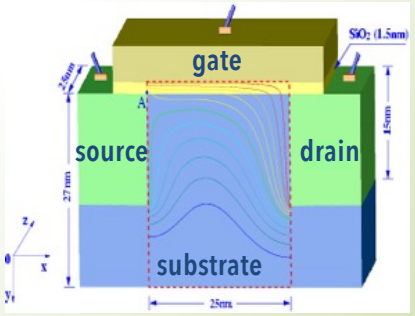
- Programming paradigms
- System architecture modeling
- Component hetero-integration
- Device/Circuit integration
- Device physics
- Fundamental material science

Berkeley Lab · Argonne National Laboratory · Oak Ridge National Laboratory · Los Alamos National Laboratory · Lawrence Livermore National Laboratory · Pacific Northwest National Laboratory · Sandia National Laboratories

FROM DEVICES TO SYSTEMS

The impact of advances in materials and structures must be understood at the systems level.

Clock-Rates, Power, Area ← Current Drive, switching energy, transients ← Junction Physics, I-V curves ← Material Physics, Carrier Mobility



Systems ← Circuits → Device Physics → Junction Physics → Materials Physics

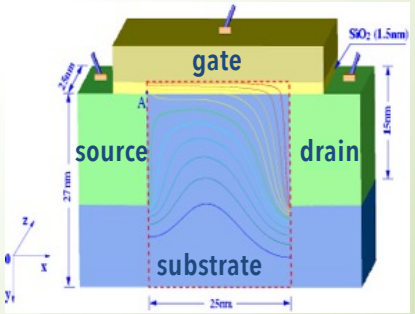
Processor/System:	Circuit/Std. Cell:	1 Device:	1 Junction:	Bulk Material:
>10k Circuits	**10-100 Devices**	**~1M Atoms**	**~100k Atoms**	**~100 Atoms**

A major theme at this workshop is the need to develop a co-design approach.

CONNECTIONS AND SCALING

Accelerated, bidirectional feedback paths are needed from scale to scale.

Application performance, System power → Switch speed, Power, Area, Fan-out, Stability → Interface-level, Losses/Per-formance → Materials metrics



Systems | Circuits | Device Physics | Junction Physics | Materials Physics

Processor/System:	Circuit/Std. Cell:	1 Device:	1 Junction:	Bulk Material:
>10k Circuits	**10-100 Devices**	**~1M Atoms**	**~100k Atoms**	**~100 Atoms**

"As the focus of innovation in architecture shifts from the general-purpose CPU to domain-specific and heterogeneous processors, we will need to achieve major breakthroughs in design time and cost. " (Hennessy-Patterson)

High-throughput screening of Novel Materials

Using first-principles calculations and Big Data Analytics on HPC

The Materials Project

Kristin Persson

Today's CMOS

Voltage limit

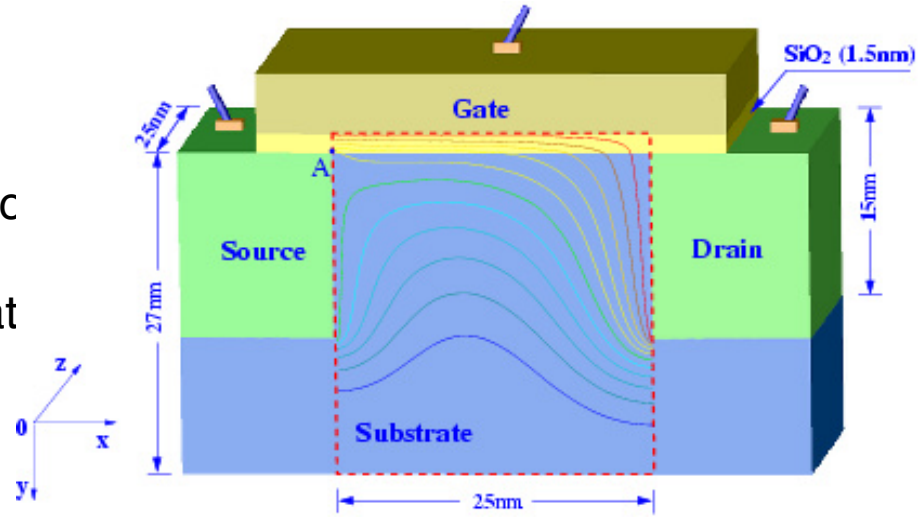
Interesting materials!!!

Accelerate discovery of new materials by factor of 1000x

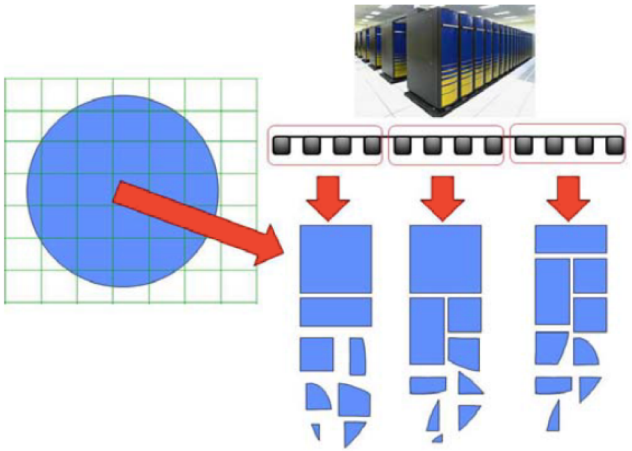
Ab-Initio Full Electronic Device Simulations

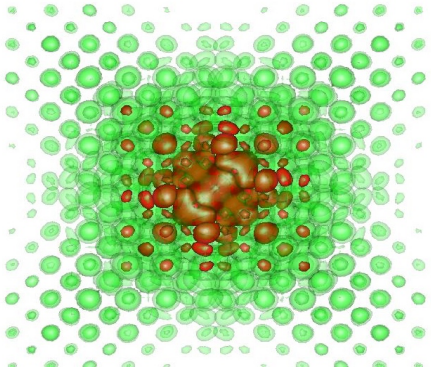
LS3DF → O(N) DFT Methods

Lin Wang Wang, Andrew Canning

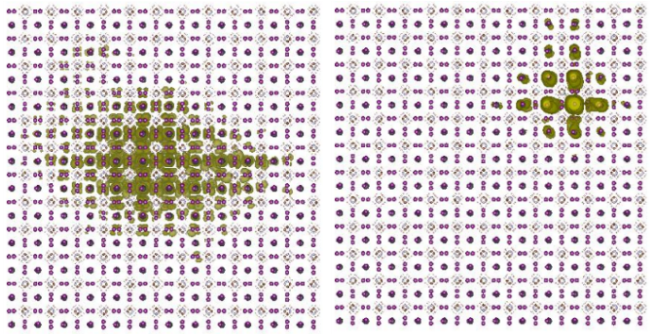


- Combined several techniques for a holistic, ab-initio, atomistic (beyond TCAT) device simulatic
- LS3DF Device-size selfconsistent ab initio calculat get atomistic potential profile, band alignment, based boundary conditioned Poisson solver
- Based on the potential profile, and scattering state calculations to simulate the device transport, and leakage current etc.
- Using electron-phonon coupling to calculate the heat generation and dissipation at atomic scale





A shallow defect state in Si.



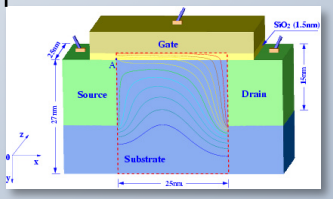
the electron (left) and hole (right) localizations in a bulk $CH_3NH_3PbI_3$ material. The small dots are atoms.

Gaps: Connections and Scaling

Metrics: Fan-out, area, switching speed, power

Compact Device Model to Verilog-A for analog circuit sim

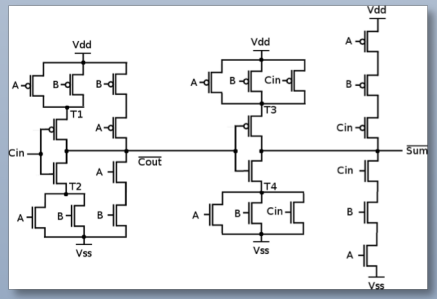
Metrics: I-V curves, current drive, switching energy, transients



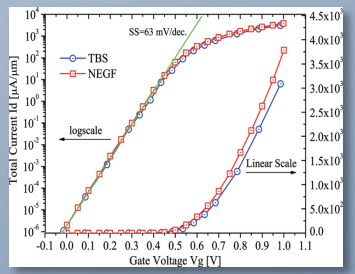
Systems | **Circuits** | **Device Physics**

Architectural Simulation

Analog Simulation



Compact Models



BERKELEY LAB

Gaps: Connections and Scaling

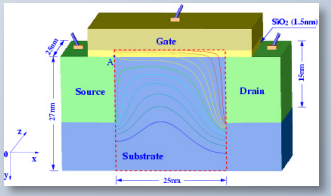
Metrics: Application performance, system power

Models of digital logic components scaled to systems that run applications

Metrics: clock rate, area, power

OpenSoC System Architect

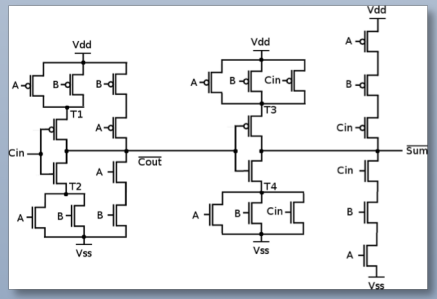
PARADISE



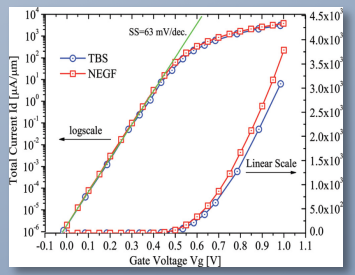
Systems | **Circuits** | **Device Physics**

Architectural Simulation

Analog Simulation



Compact Models

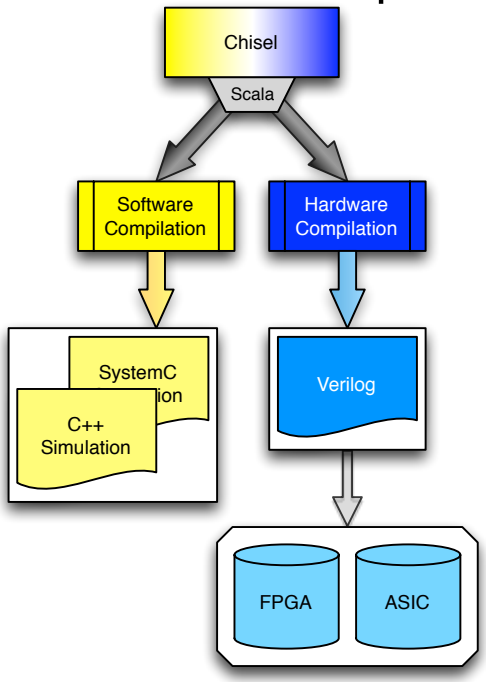


BERKELEY LAB

Rapidly Prototype/Synthesize SoCs
(Synthesis & Simulation)

Chisel

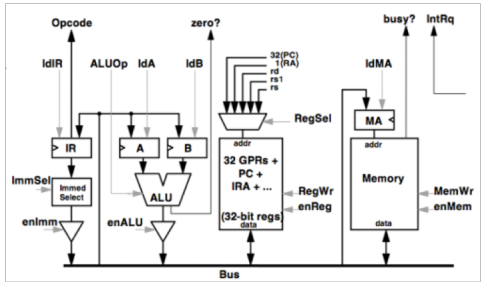
DSL for rapid prototyping of circuits, systems, and arch simulator components

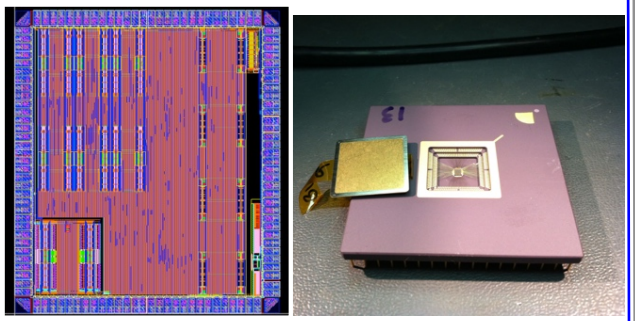


Back-end to synthesize HW with different devices Or new logic families

RISC-V

Open Source Extensible ISA/Cores

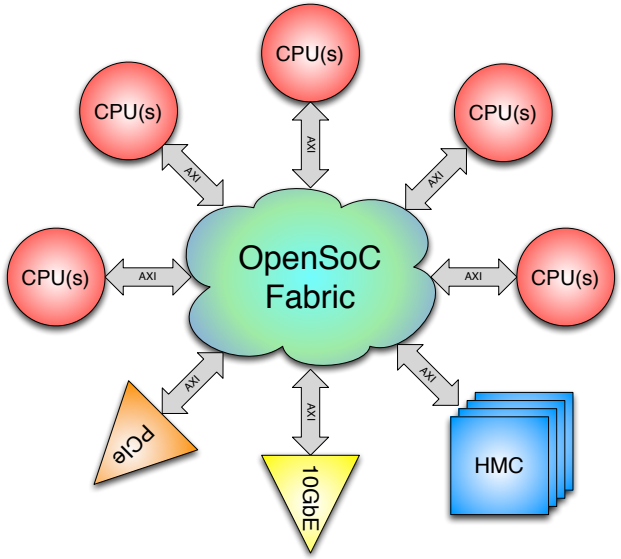




Re-implement processor With different devices or Extend w/accelerators

OpenSOC

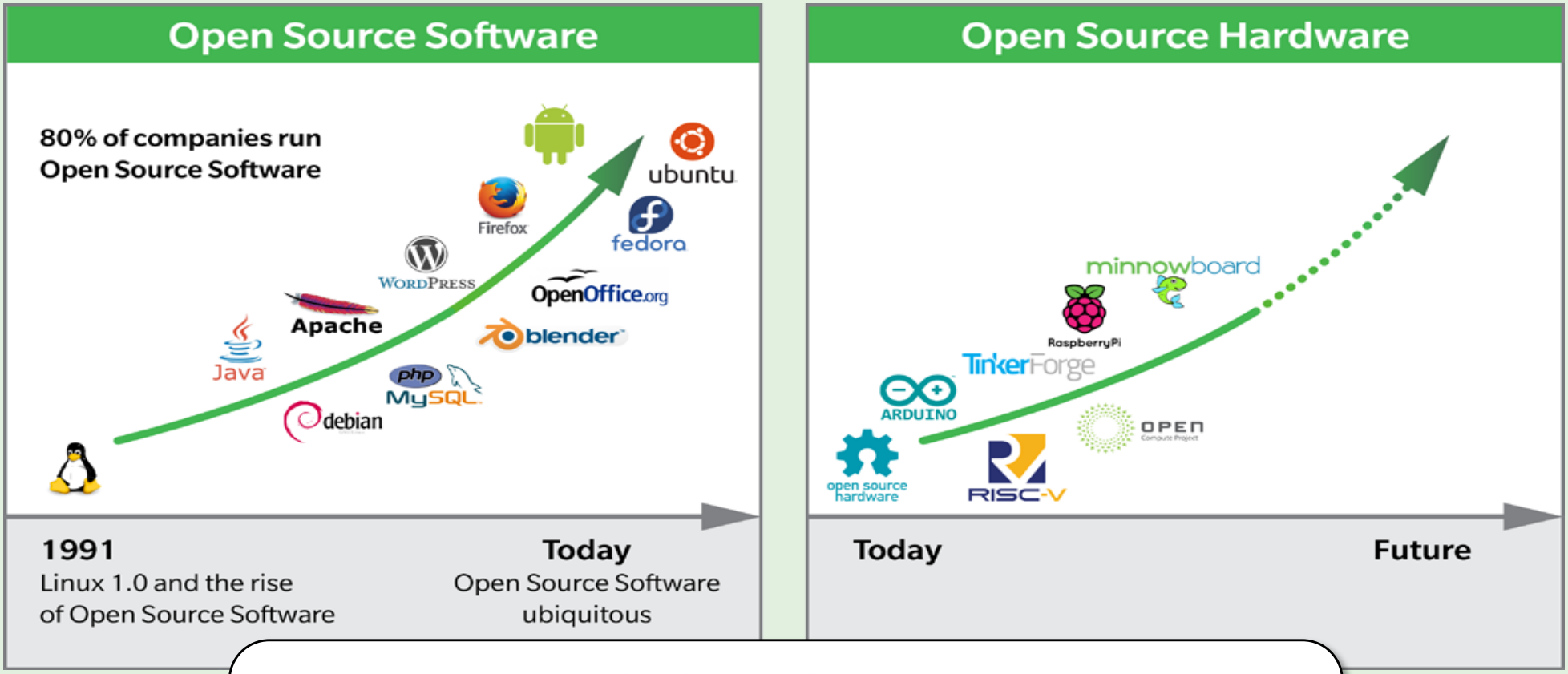
Open Source fabric To integrate accelerators And logic into SOC



Platform for experimentation with specialization to extend Moore's Law

Open Source Hardware:

The Rise of Open Source Software: Will Hardware Follow Suit?

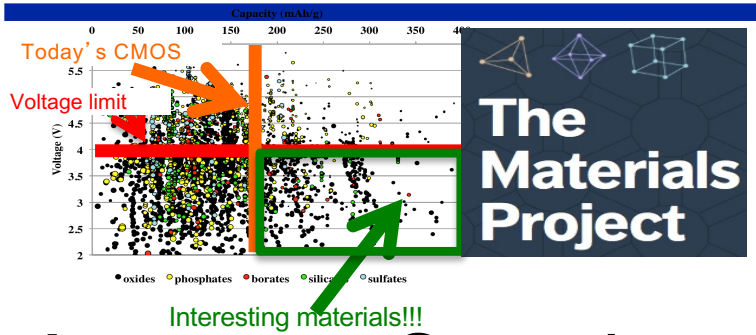


Mathematician in-the-loop ...
. . . Lawyer OUT!

- Rapid gro
- More than
  of smartphones run Android variants
- Will open source hardware ignite the semiconductor industry?
  Is RISC-V the hardware industry's Linux?

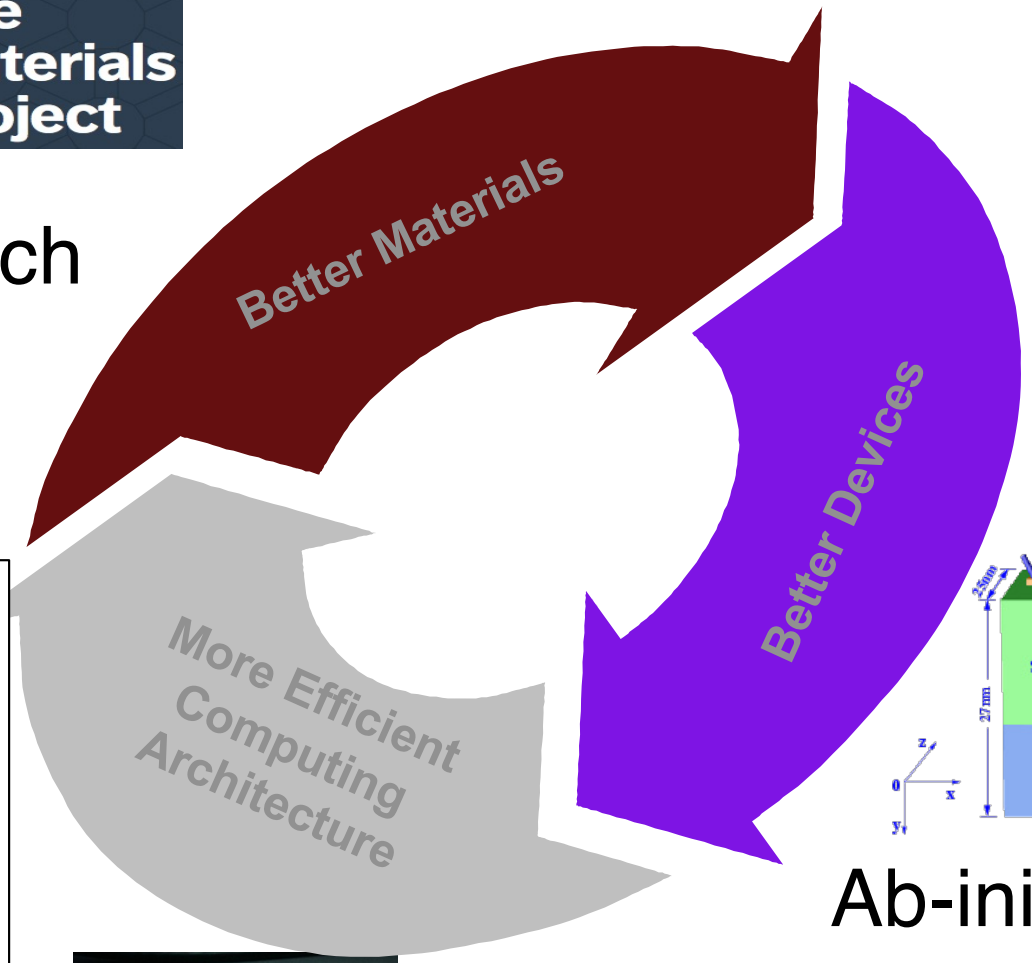
Industrializing the Discovery Process with CoDesign

(Accelerate Discovery of new Materials/Devices/Architectures by 1000x)



The Materials Project

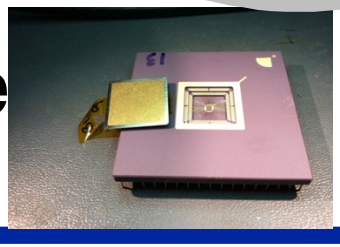
Automate Search for Better Materials

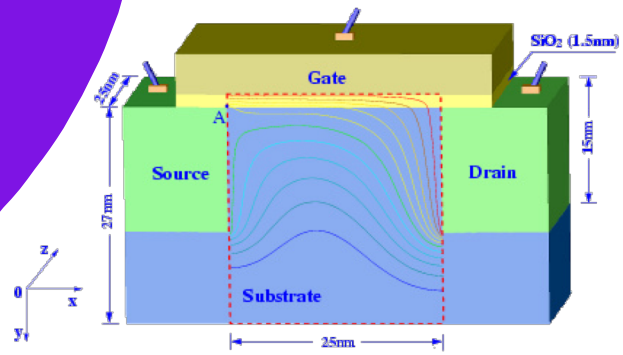


OpenHPC

SystemArchitect

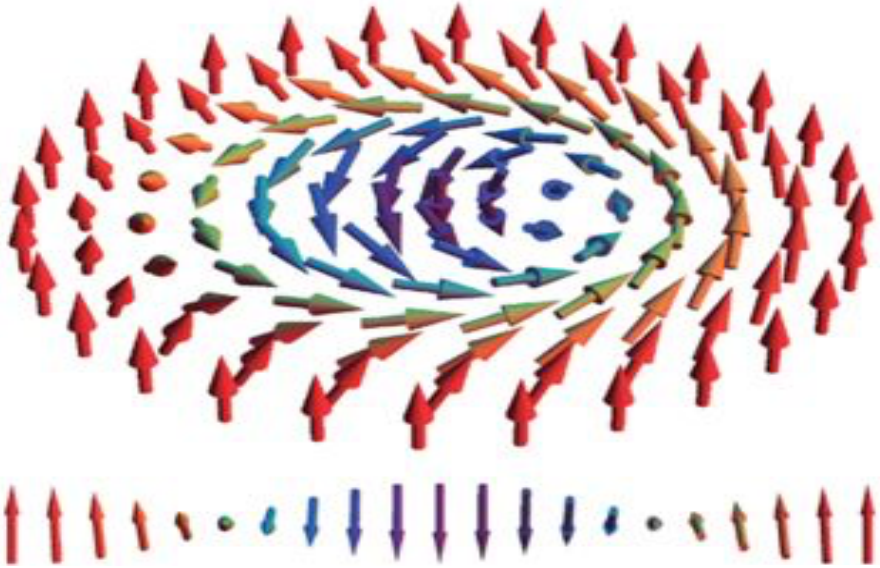
Rapid Prototype Hardware

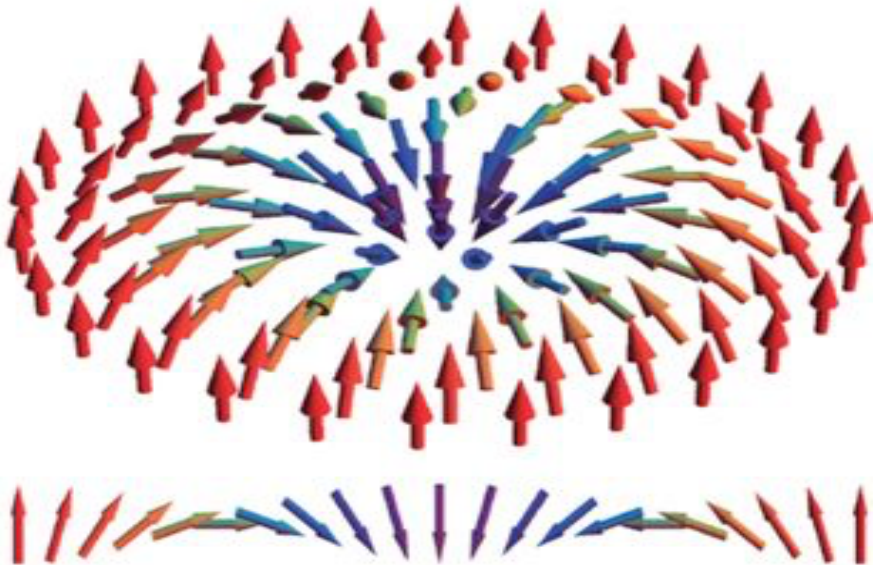


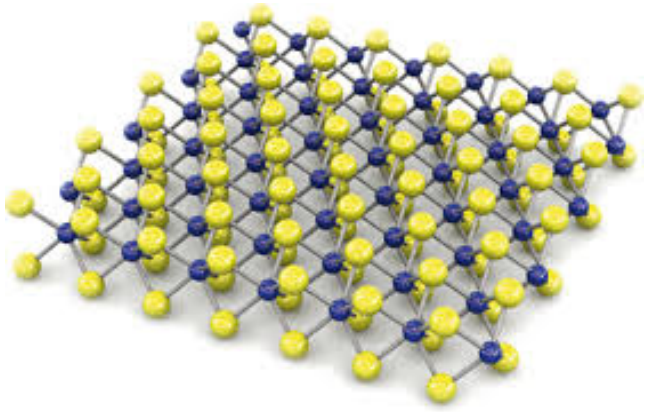


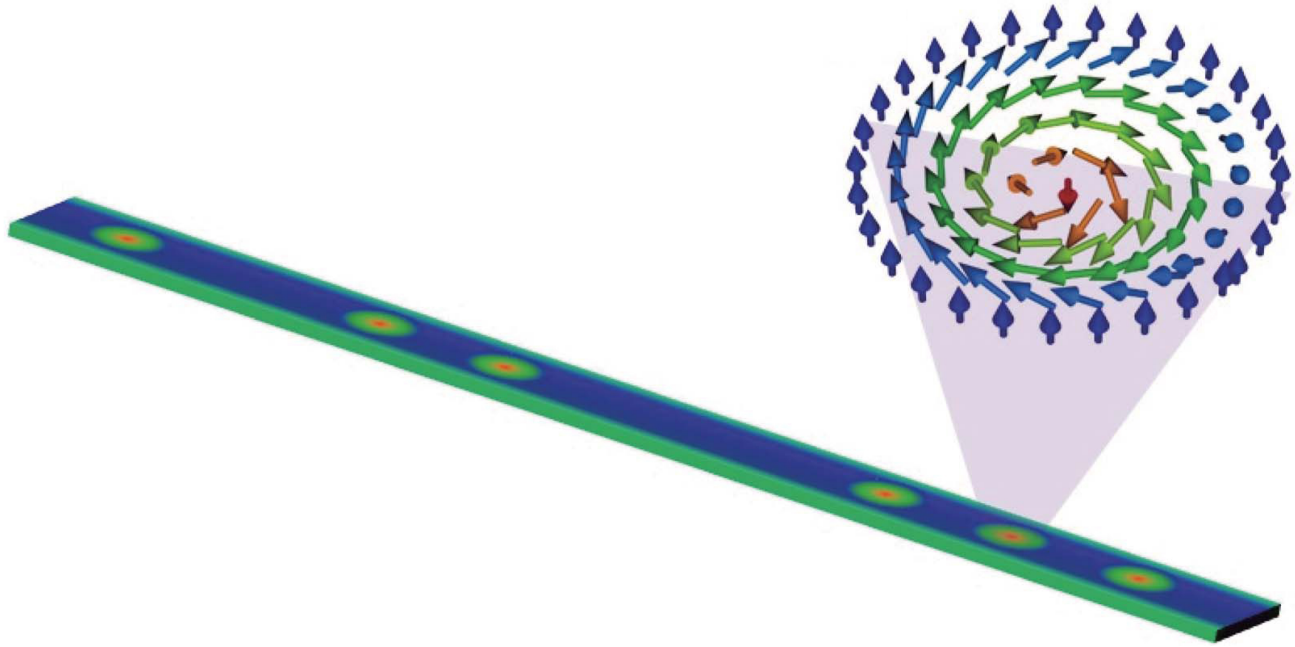
Ab-initio Full Device Simulation

Skyrmions



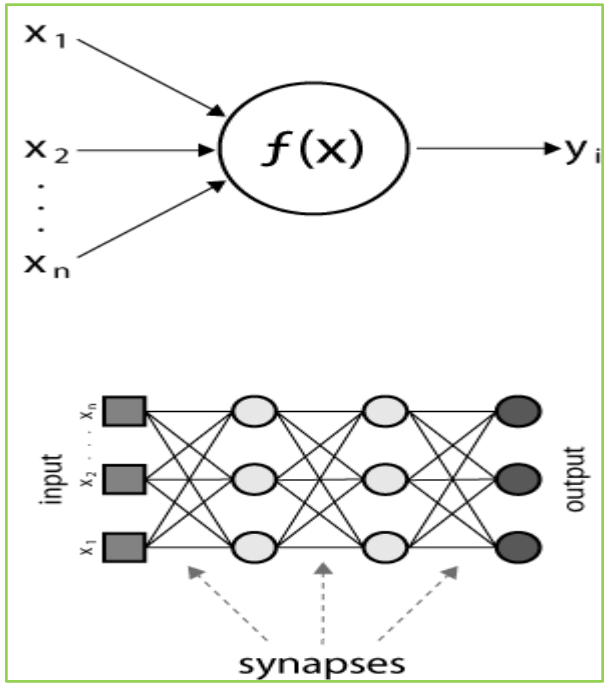


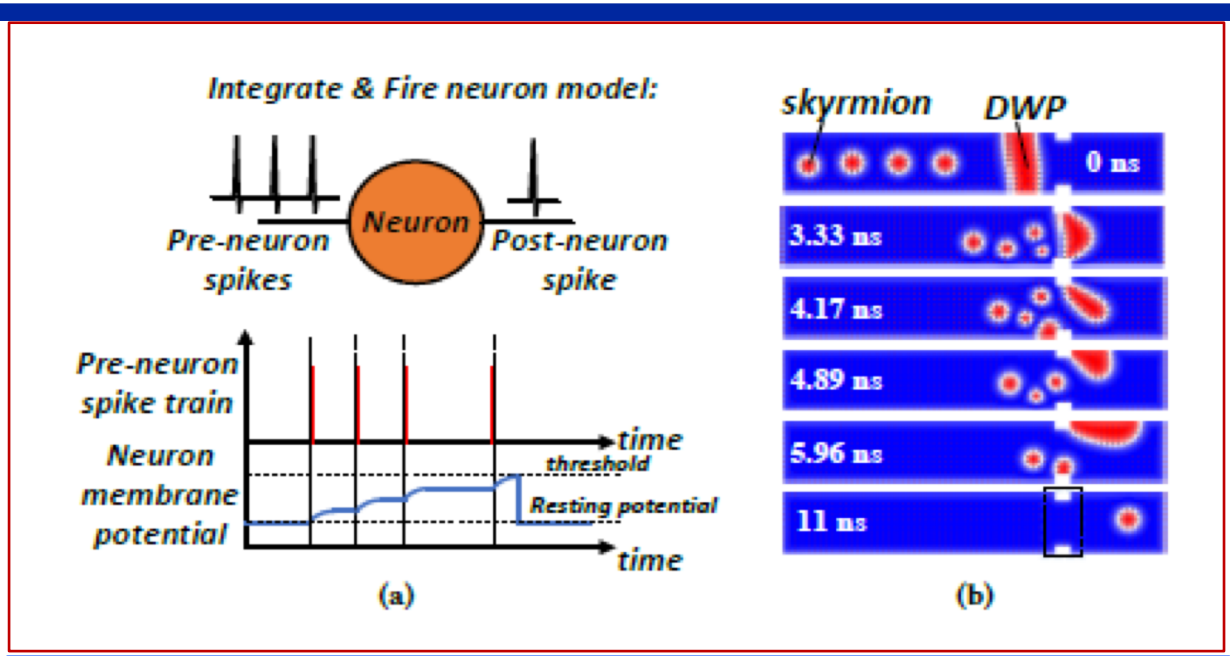


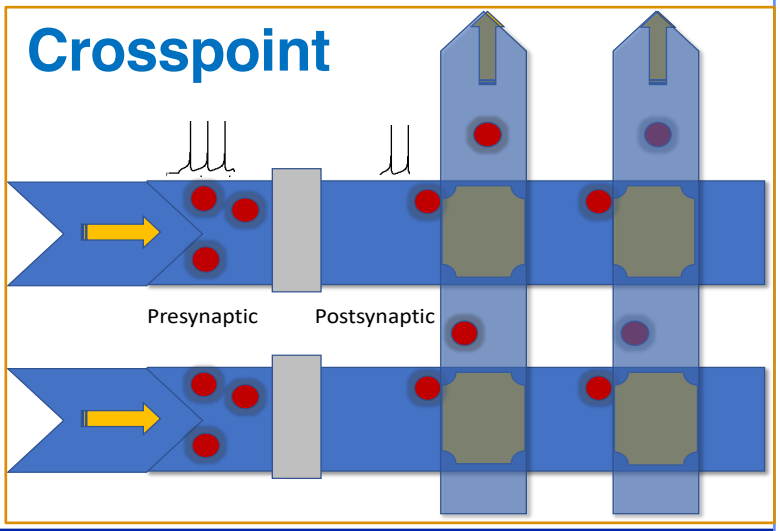


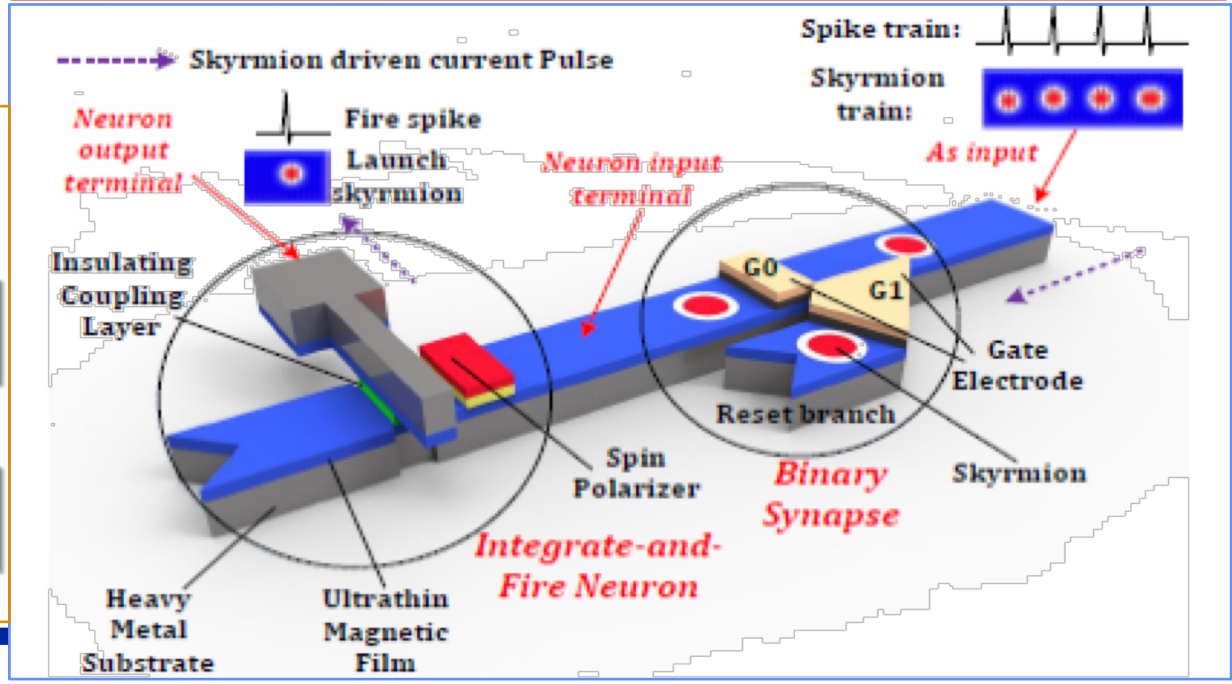
Single-track Skyrmion-Based Spiking Neural Network

Z. He et al., 1705.02995v1 (2017)









Impact of End of Moore's Law on Parallel Software

Impact on Programming of Parallel Systems

Results of DOE 2018 Extreme Heterogeneity Workshop *Video*

- **Exponentially Increasing Parallelism (central challenge for Exascale)**
  - **Trend**: *End of exponential clock frequency scaling (end of Dennard scaling)*
  - **Consequence:** *Exponentially increasing parallelism*

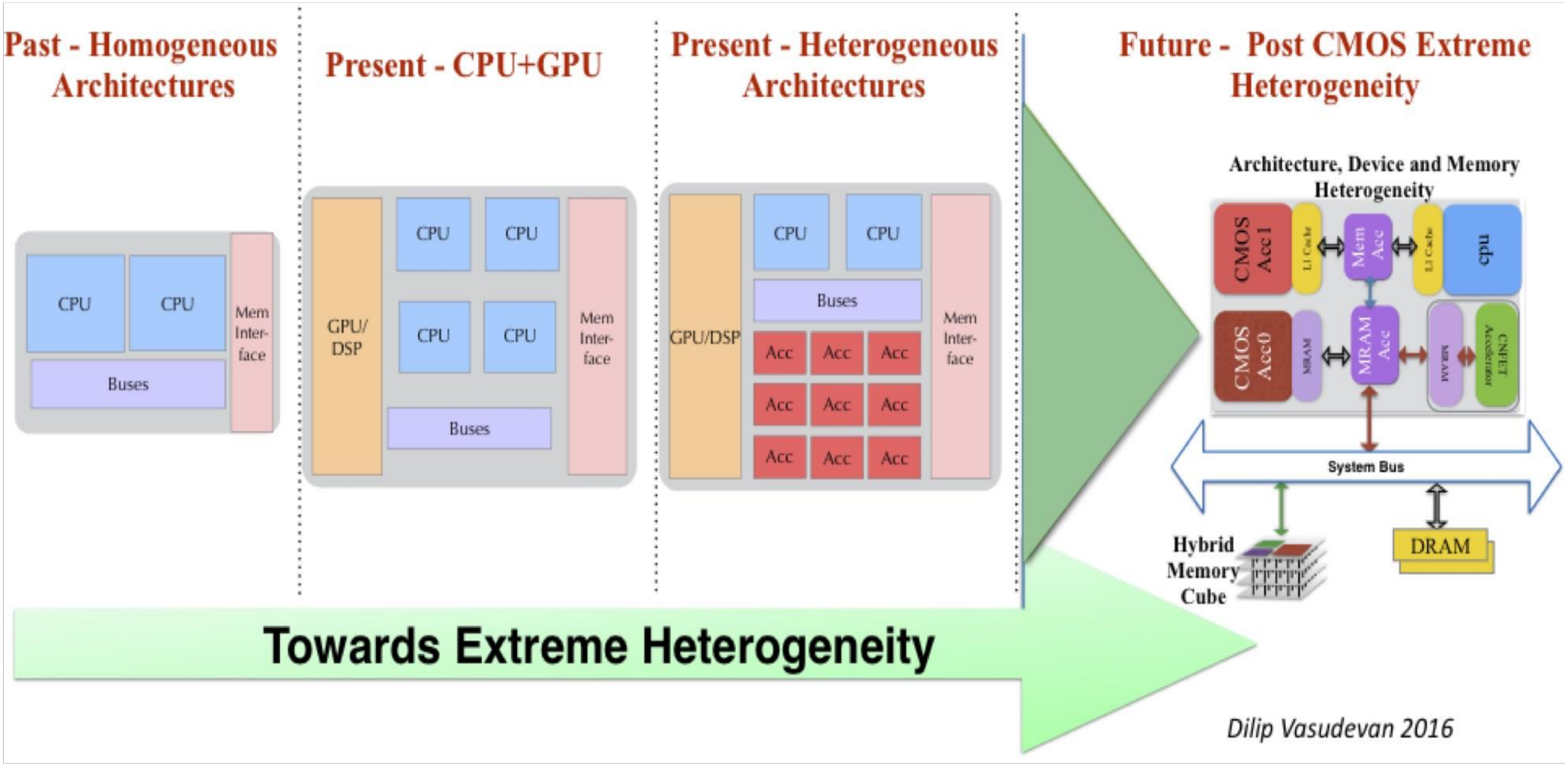
- **End of Lithography as Primary Driver for Technology Improvements**
  - **Trend:** *Tapering of lithography Scaling*
  - **Consequence:** *Many forms of heterogeneous acceleration (not just GPGPUs)*

- **Data Movement Heterogeneity and Increasingly Hierarchical Machine**
  - **Trend**: *Moving data operands costs more than computation performed on them*
  - **Consequence**: *More heterogeneity in data movement performance and energy*

- **Performance Heterogeneity**
  - **Trend**: *Heterogeneous execution rates from contention and power management*
  - **Consequence**: *Extreme variability and heterogeneity in execution rates*

- **Diversity of Emerging Memory and Storage Technologies**
  - **Trend**: *Emerging memory technologies and stall in performance improvements*
  - **Consequence**: *Disruptive changes to our storage environment* ***(POSIX just won't cut it anymore!!!)***

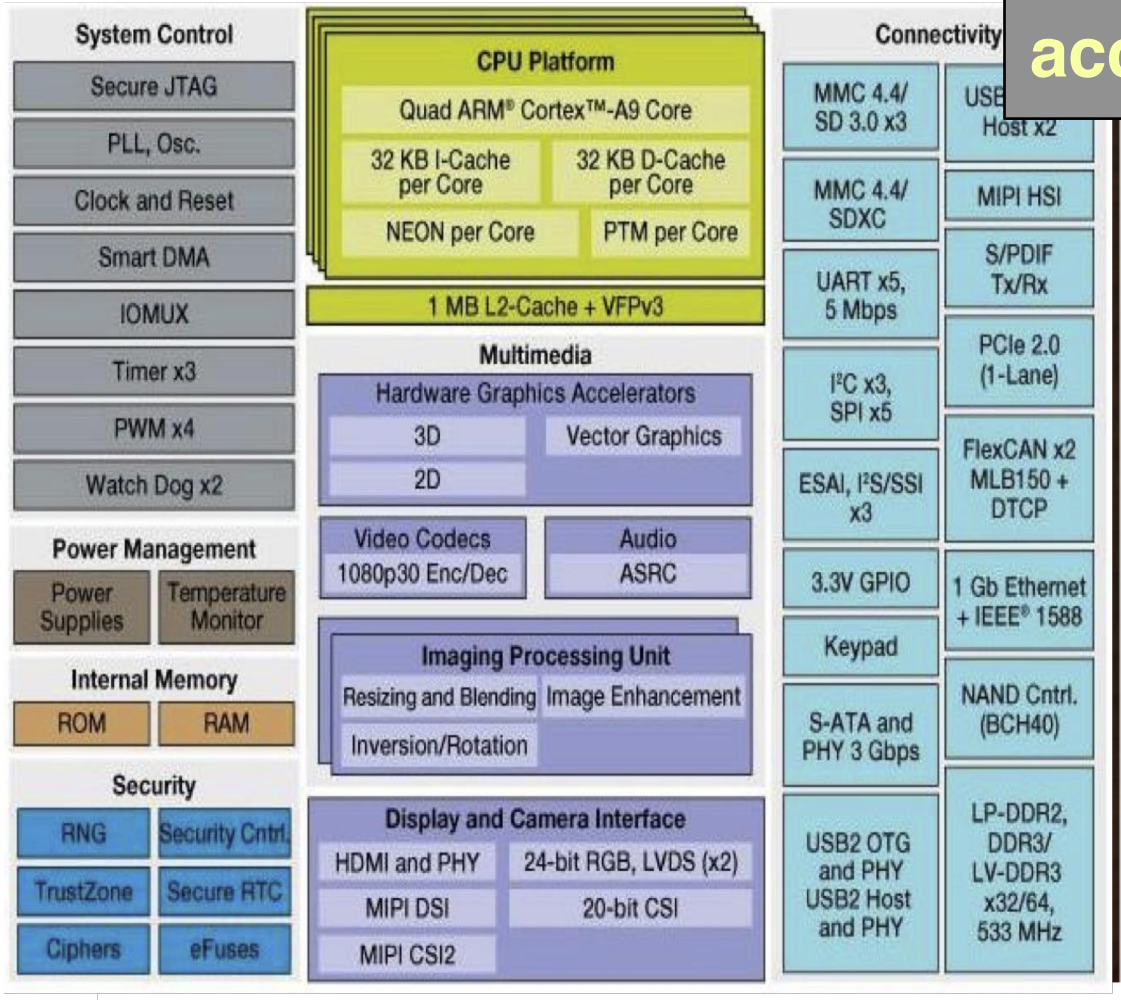
Towards Diverse Integrated Accelerators



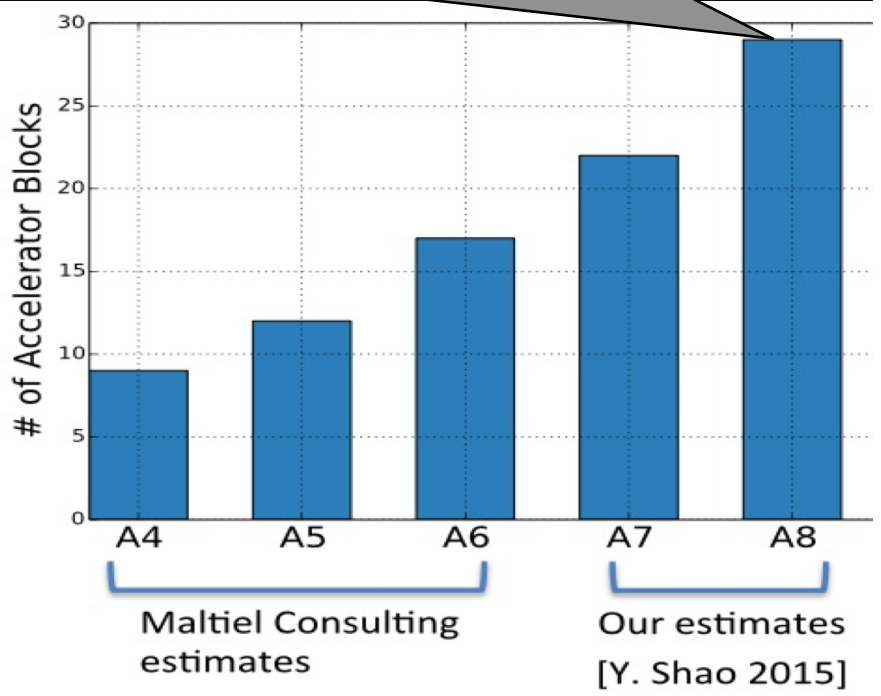
Extreme Hardware Specialization is Happening Now

This trend is already well underway in broader electronics industry
Cell phones and even megadatacenters (Google TPU, Microsoft FPGAs…)

(and it will happen to HPC too… will we be ready?)



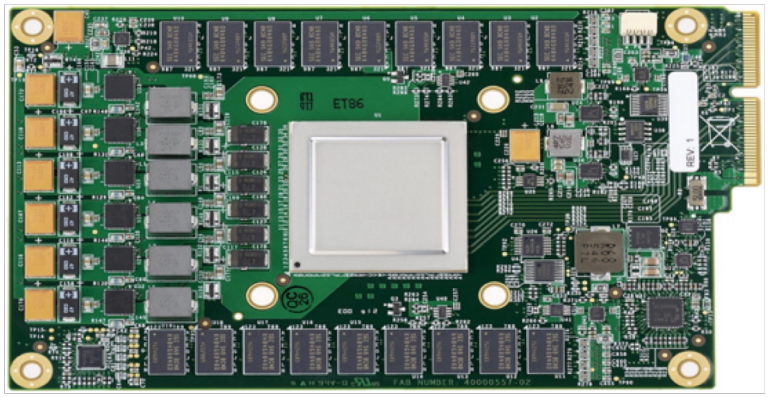
29 different heterogeneous accelerators in Apple A8 (2016)



[www.anandtech.com/show/8562/chipworks-a8]

Google Tensor Processing Unit (TPU)

- **Deployed in datacenters since 2015**
- 10-30x Faster than NVIDIA G80 or Intel Haswell for ML workloads *(64k arithmetic ops per cycle)*
- Could be faster with better memory subsystem.
- 8bit integer arithmetic (all that is needed for ML)



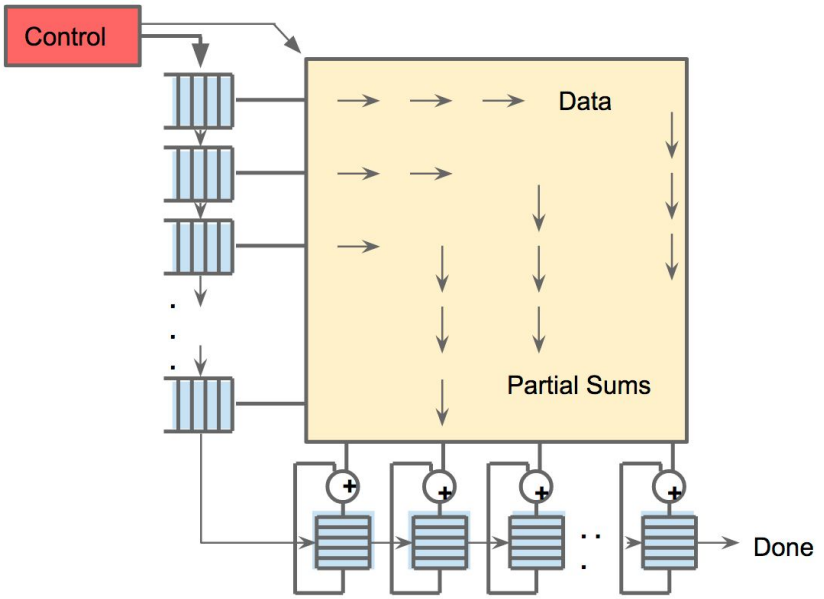


Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Model	Die										Benchmarked Servers				
	mm²	nm	MHz	TDP	Measured		TOPS/s		GB/s	On-Chip Memory	Dies	DRAM Size	TDP	Measured	
					Idle	Busy	8b	FP						Idle	Busy
Haswell E5-2699 v3	662	22	2300	145W	41W	145W	2.6	1.3	51	51 MiB	2	256 GiB	504W	159W	455W
NVIDIA K80 (2 dies/card)	561	28	560	150W	25W	98W	--	2.8	160	8 MiB	8	256 GiB (host) + 12 GiB x 8	1838W	357W	991W
TPU	NA*	28	700	75W	28W	40W	92	--	34	28 MiB	4	256 GiB (host) + 8 GiB x 4	861W	290W	384W

Why us and why now?

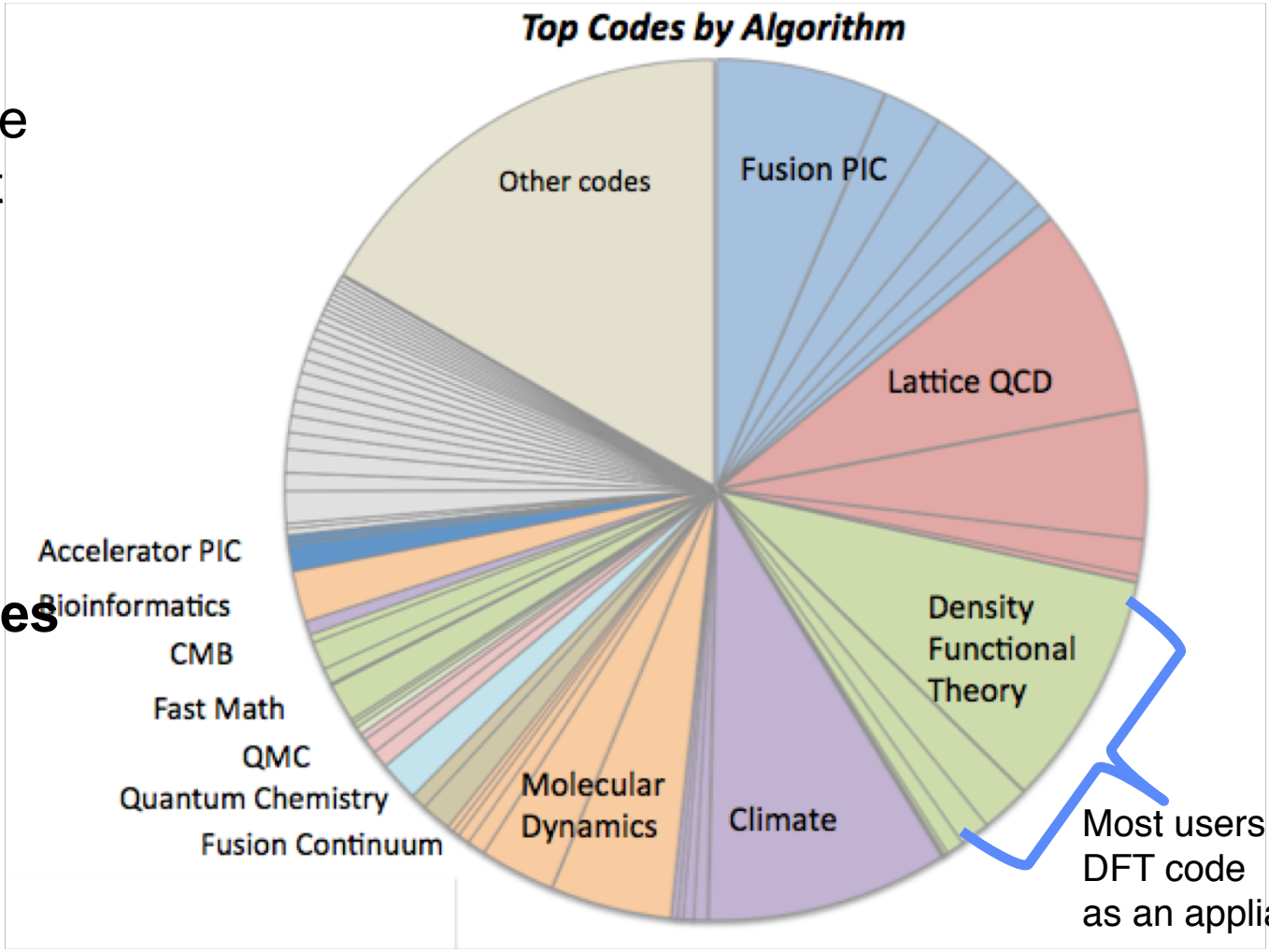
- **The rest of industry is already moving forward with specialization**
  - Gen2 Google TPU 2,300x faster than CPU for AI workloads
  - Microsoft FPGA accelerators for search acceleration
  - Numerous other specialized architectures in AI space

- **Little incentive for industry to provide scientifically relevant accelerators**
  - We need to learn how to do this ourselves!
  - When should we begin? ***NOW!***

- **Why will this work now *(not very successful in past)***
  - Already demonstrated successes in cell phones, datacenters and AI
  - Won't be eclipsed by Moore's Law performance growth **this time!**

Why DFT?

- **Mature code that has large user base**
  - Hard to specialize for moving target
  - Offers high scientific impact
- **O(N) (LS3DF) algorithmic alternative maximizes spatial locality**
- ***Most users regard DFT codes as an "appliance"!***
  (good candidate)

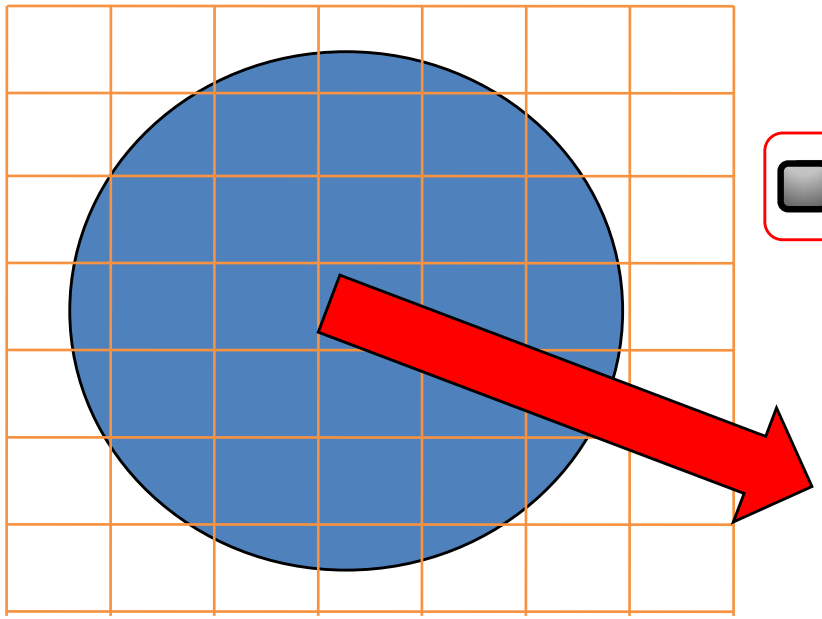


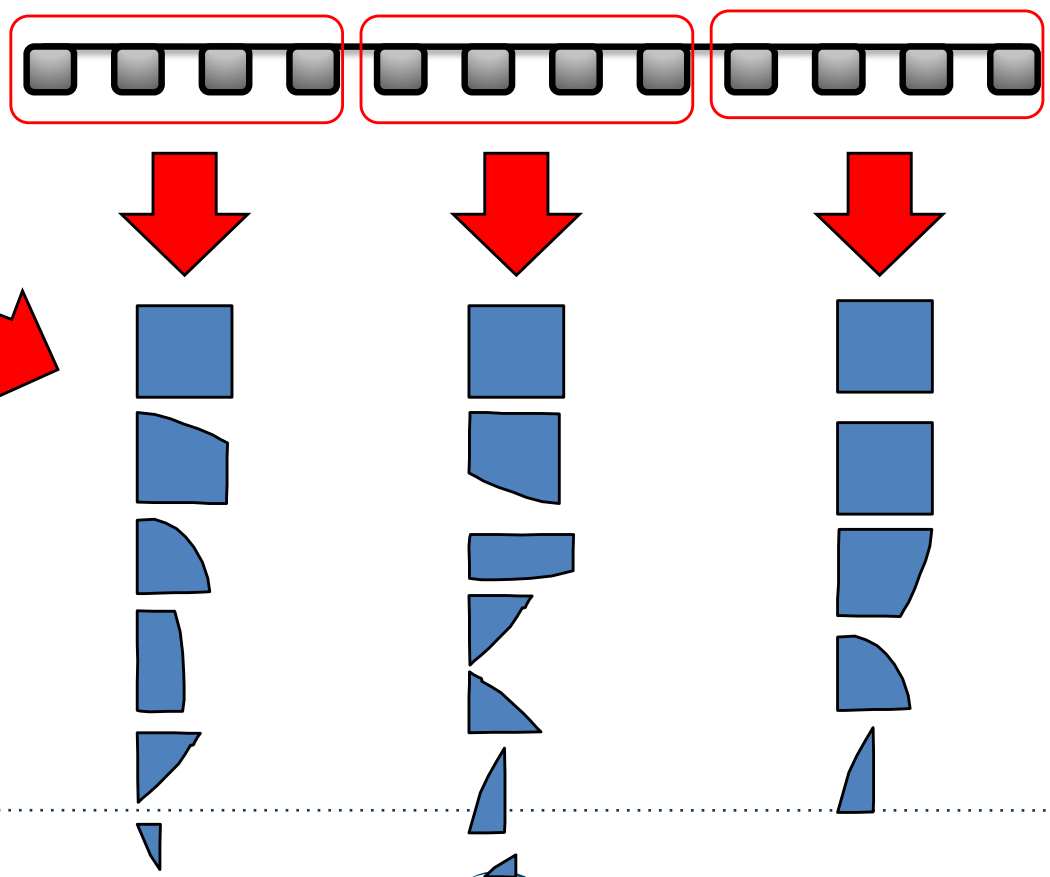
Why O(N) divide & conquer method (e.g., LS3DF)?

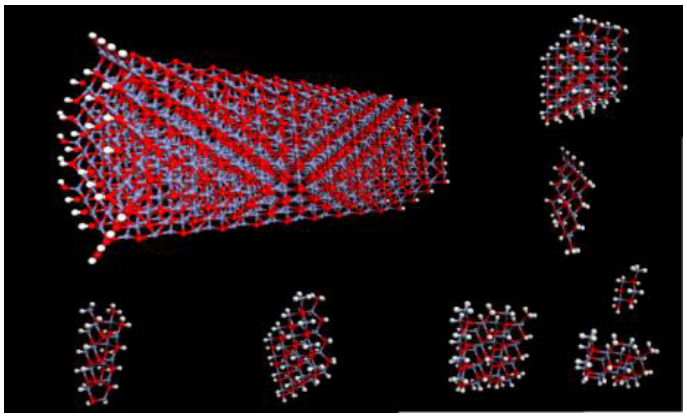
Each Patch executes entirely within FPGA

eliminate memory bottleneck

Scale problem by adding FPGAs

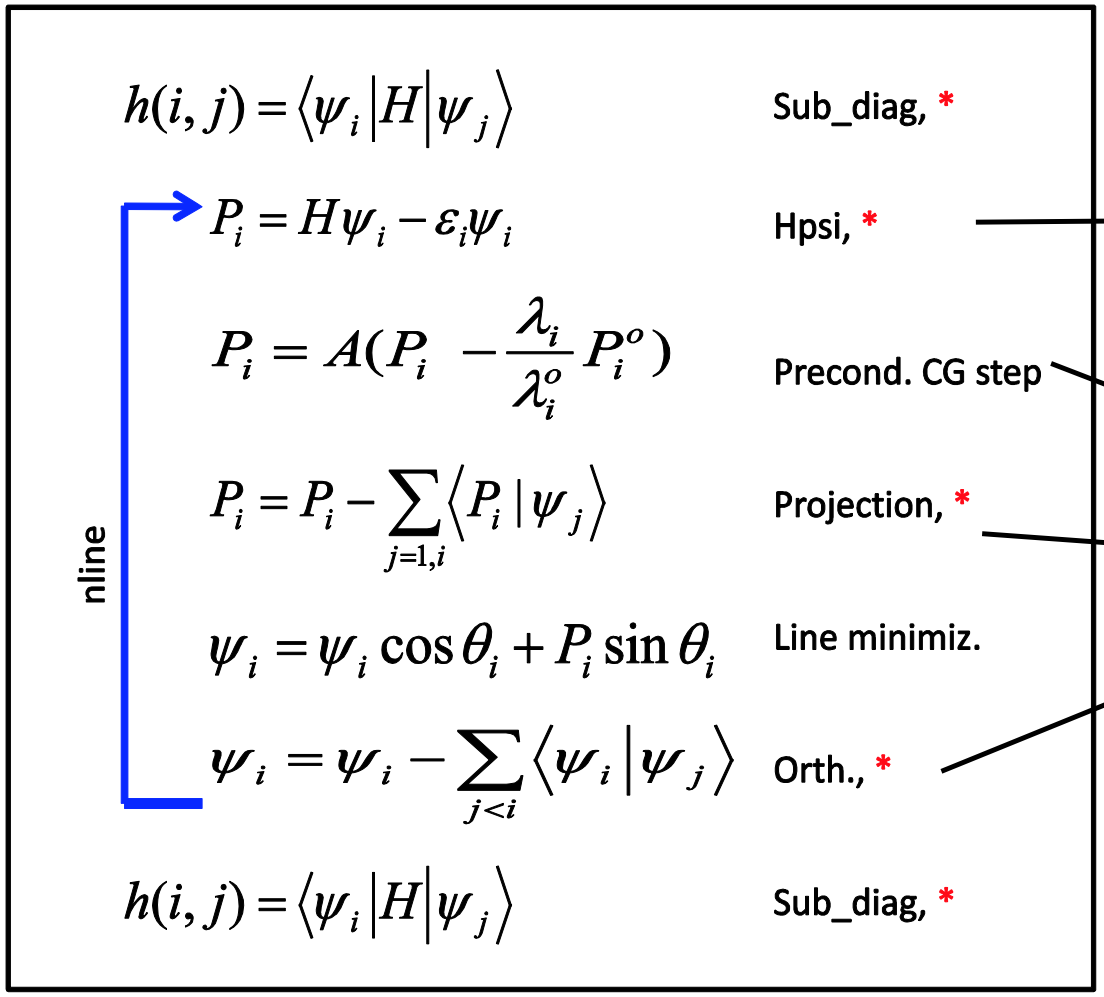






The DFT kernel for each fragment
(to be performed within each FPGA)

Scale to larger problems by adding FPGAs



$O(N^2 \log(N))$

Comm bound if non-local

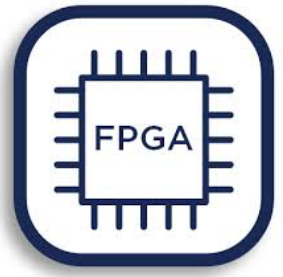
3D parallel FFT

TSQR & Choelesky

ZGEMM

$O(N^3)$

Compute-bound



One patch per FPGA

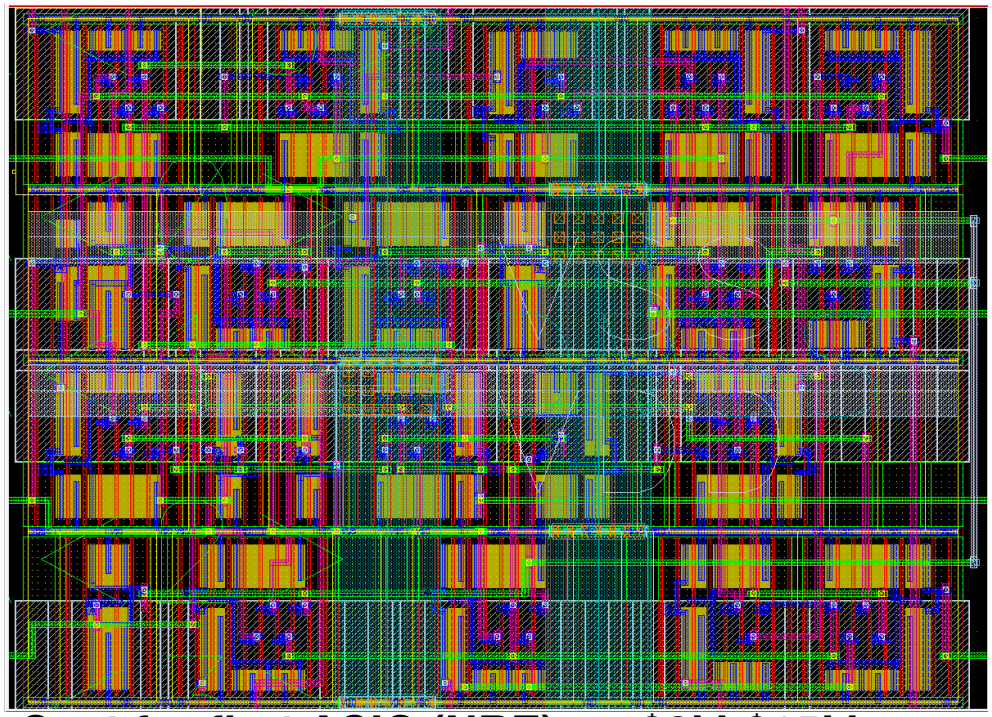
400 atoms/patch

Field Programmable Gate Arrays (FPGA)

FPGA

Cost for first FPGA (NRE):	$1,200-$3,000
Cost for 20,000th :	$1,200-$3,000
Clock Rate:	0.1-0.3Ghz

ASIC



Cost for first ASIC (NRE):	$2M-$15M
Cost for 20,000th :	$100
Clock Rate:	1-2 Ghz (10x)
Area Efficiency:	10x FPGA
Energy Efficiency :	10x-100x FPGA

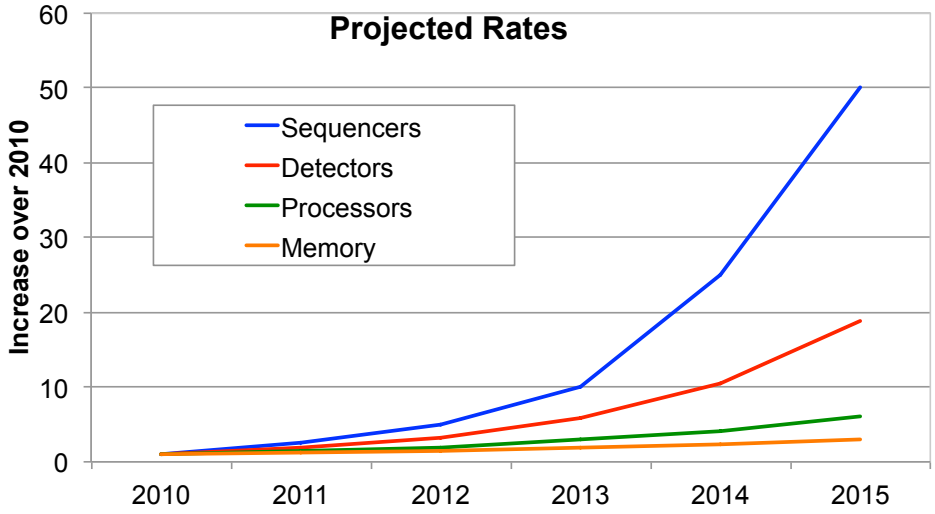
Use FPGA as a testbed for ASIC

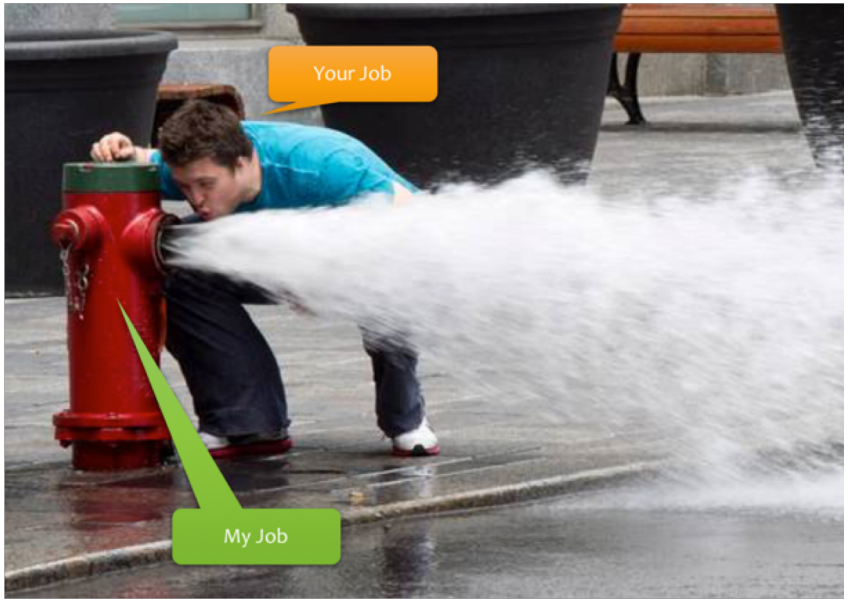
If successful, can project ASIC performance

On-detector processing

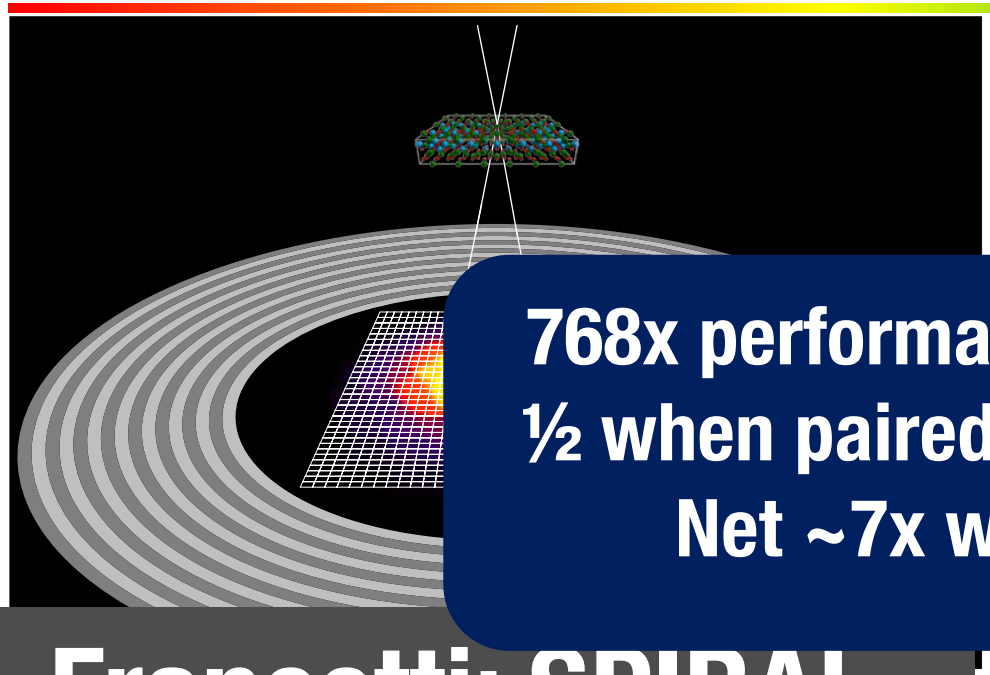
Putting our hardware design tool suite to work to augment existing HPC resources

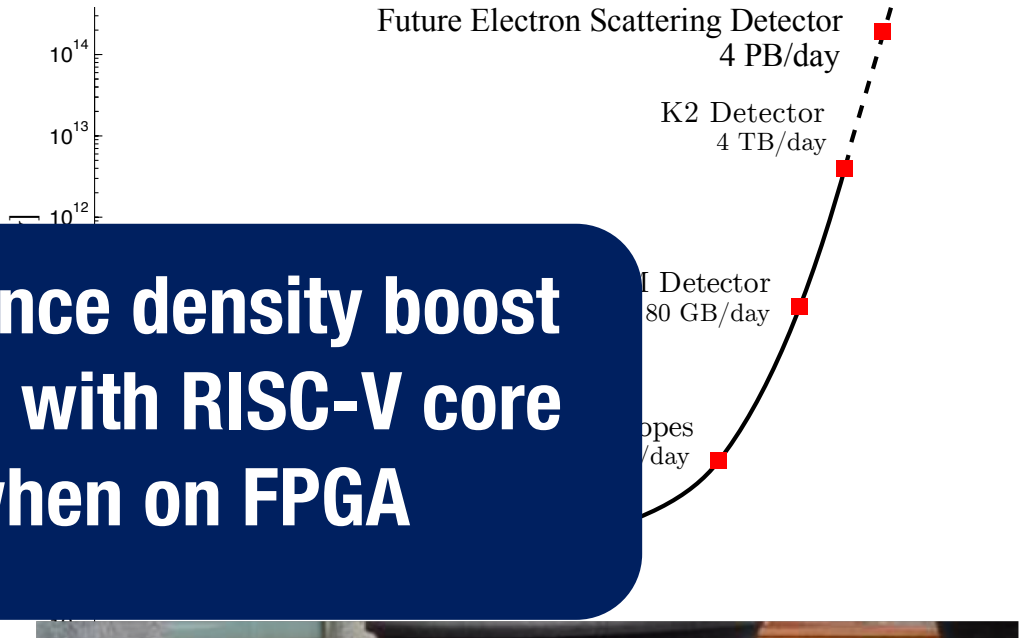
- **The Problem:**
  - Future detectors threaten to overwhelm data transfer and computing capabilities w/ data rates exceeding 1 Tb/s
  - Data processing experiment driven
- **Proposed solution:**
  - Process the data *before it leaves the sensor*
  - Application-tailored, programmable processing allows data reduction to occur on-sensor
  - Programmability allows data reduction techniques to be tailored to the experiment – even *after* the sensor is built!





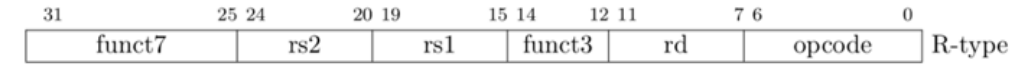
On-detector processing: CryoEM



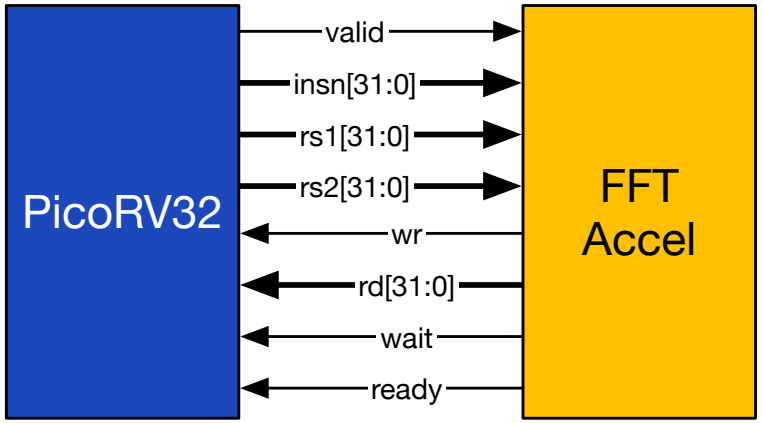


768x performance density boost
½ when paired with RISC-V core
Net ~7x when on FPGA

Francetti: SPIRAL



PCPI



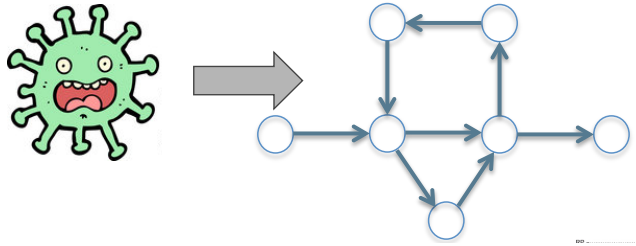
- All FFT instructions are contained within:
  `opcode[1:0] = 00b`
  - All standard RV32 instructions are contained in:
    `opcode[1:0] = 11b`

Instruction	opcode[3:2]	Description
`fft_config`	`10b`	Configures FFT parameters
`fft_status`	`01b`	Reads FFTAccel status registers
`fft_start`	`11b`	Starts FFT processing
`fft_stop`	`00b`	Stops FFT processing

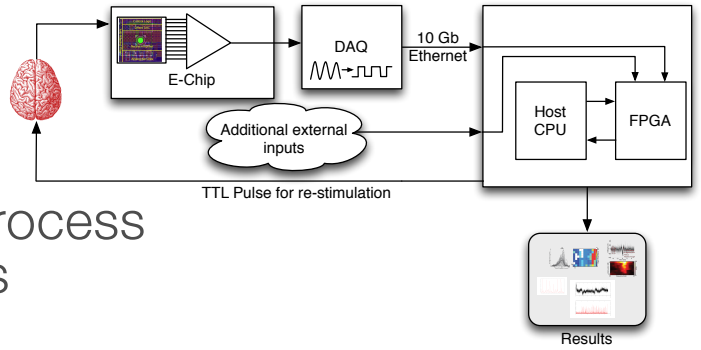
- This leaves `opcode[6:4]`, `func3`, and `func7` unused
  - Potentially used to address multiple FFTAccel

More opportunities for specialization

Custom computing could be applied and incorporated in many areas across DOE



- **JGI – Dedicated appliance for BLAST**
  - Majority of compute time at JGI spent on BLAST
  - Design, build and deploy custom BLAST computing appliance



- **BRAIN – Real-time signal processing**
  - To perform closed-loop experiments need to process and respond to incoming neural data in < 10ms
- **LSST – Transient detection**
  - Power constrained on-site, custom, low-power computing could reduce need for lossy compression or high-bandwidth, long-haul networks

- **All these applications *augment* rather than *replace* future HPC resources**

There are Multiple Ways to Specialize

1. **Extended ISA:** *extend ISA to replace common motifs with Fixed function within each core.*
   - *Requires a more flexible/extensible compiler*
   - *Tensilica: ISA extensions exposed as intrinsics that could easily be executed as subroutine or compiler generates code*
2. **Diverse function accelerators within each chip:** *SmartPhone, DE Shaw Anton and Amazon Cloud examples*
   - *Expose as software interfaces at first, and then invoke hardware*
   - *Currently labor-intensive to build underlying interfaces (need standardization of low-level interfaces (an accelerator ABI?)*
3. **Diverse Discrete Accelerators In System:** *Our current approach with GPU, but extended to diverse accelerators*
   - *We already see how this works out (need to better manage data movement)*

Data Movement Could Undercut any Gains from Accelerators and Specialization

Under what circumstances?
How bad is it?

Lets just do a simple thought experiment

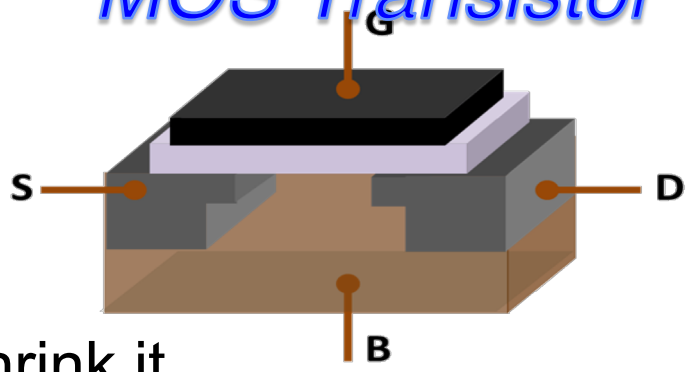
The problem with Wires

Energy to move data proportional to distance

- **Energy Efficiency of copper wire:**
  - **Power = frequency* Length / cross-section-area**

  - Wire efficiency *does not improve* as feature size shrinks

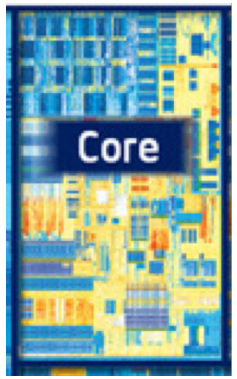
MOS Transistor



- **Energy Efficiency of a Transistor:**
  - Power = V^2 * frequency * Capacitance
  - Capacitance ~= Area of Transistor
  - Transistor efficiency improves as you shrink it

- ***Net result is that moving data on wires is starting to cost more energy than computing on said data***

Basic Stats



Core Energy/Area est.

Area: 12.25 mm^2
Power: 2.5W
Clock: 2.4 GHz
E/op: 651 pj

Area: 0.6 mm^2
Power: 0.3W (<0.2W)
Clock: 1.3 GHz
E/op: 150 (75) pj

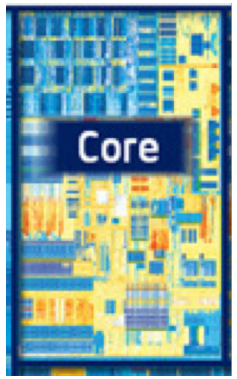
Area: 0.046 mm^2
Power: 0.025W
Clock: 1.0 GHz
E/op: 22 pj

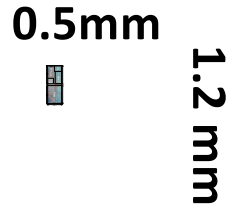
Wire Energy

Assumptions for 22nm
100 fj/bit per mm
64bit operand

Energy:
1mm=~6pj
20mm=~120pj

When does data movement dominate?





Core Energy/Area est.	Data Movement Cost
Area: 12.25 mm^2 Power: 2.5W Clock: 2.4 GHz E/op: 651 pj	*Compute Op == data movement Energy @* **108mm** *Energy Ratio for 20mm* **0.2x**
Area: 0.6 mm^2 Power: 0.3W (<0.2W) Clock: 1.3 GHz E/op: 150 (75) pj	*Compute Op == data movement Energy @* **12mm** *Energy Ratio for 20mm* **1.6x**
Area: 0.046 mm^2 Power: 0.025W Clock: 1.0 GHz E/op: 22 pj	*Compute Op == data movement Energy @* **3.6mm** *Energy Ratio for 20mm* **5.5x**

Way Back When

(Use Parallel DO and life was good)

```
DO I=2,N
     B(I) = (A(I) + A(I-1)) / 2.0
ENDDO
```

Data Movement Directives

(directives in red) *John Levesque*

CRAY THE SUPERCOMPUTER COMPANY

Keep data on the accelerator with acc_data region

```
!$acc data copyin(cix,ci1,ci2,ci3,ci4,ci5,ci6,ci7,ci8,ci9,ci10,ci11,&
!$acc& ci12,ci13,ci14,r,b,uxyz,cell,rho,grad,index_max,index,&
!$acc& ciy,ciz,wet,np,streaming_sbuf1, &
!$acc&    streaming_sbuf1,streaming_sbuf2,streaming_sbuf4,streaming_sbuf5,&
!$acc&    streaming_sbuf7s,streaming_sbuf8s,streaming_sbuf9n,streaming_sbuf10s,&
!$acc&    streaming_sbuf11n,streaming_sbuf12n,streaming_sbuf13s,streaming_sbuf14n,&
!$acc&    streaming_sbuf7e,streaming_sbuf8w,streaming_sbuf9e,streaming_sbuf10e,&
!$acc&    streaming_sbuf11w,streaming_sbuf12e,streaming_sbuf13w,streaming_sbuf14w, &
!$acc&    streaming_rbuf1,streaming_rbuf2,streaming_rbuf4,streaming_rbuf5,&
!$acc&    streaming_rbuf7n,streaming_rbuf8n,streaming_rbuf9s,streaming_rbuf10n,&
!$acc&    streaming_rbuf11s,streaming_rbuf12s,streaming_rbuf13n,streaming_rbuf14s,&
!$acc&    streaming_rbuf7w,streaming_rbuf8e,streaming_rbuf9w,streaming_rbuf10w,&
!$acc&    streaming_rbuf11e,streaming_rbuf12w,streaming_rbuf13e,streaming_rbuf14e, &
!$acc&    send_e,send_w,send_n,send_s,recv_e,recv_w,recv_n,recv_s)
  do ii=1,ntimes
        o o o
     call set_boundary_macro_press2
     call set_boundary_micro_press
     call collisiona
     call collisionb
     call recolor
```

Where is the Abstraction?

OMP4 taking on Data-movement/Locality Issues

And you have to do this for EVERY SINGLE LOOP!


```
#pragma omp target data if(N>THRESHOLD) map(from: p[0:N])
{
    #pragma omp target if (N>THRESHOLD) map(to: v1[:N], v2[:N])
    #pragma omp parallel for
    for (i=0; i<N; i++)
      p[i] = v1[i] * v2[i];
    init_again(v1, v2, N);
    #pragma omp target if (N>THRESHOLD) map(to: v1[:N], v2[:N])
    #pragma omp parallel for
    for (i=0; i<N; i++)
      p[i] = p[i] + (v1[i] * v2[i]);
}
```

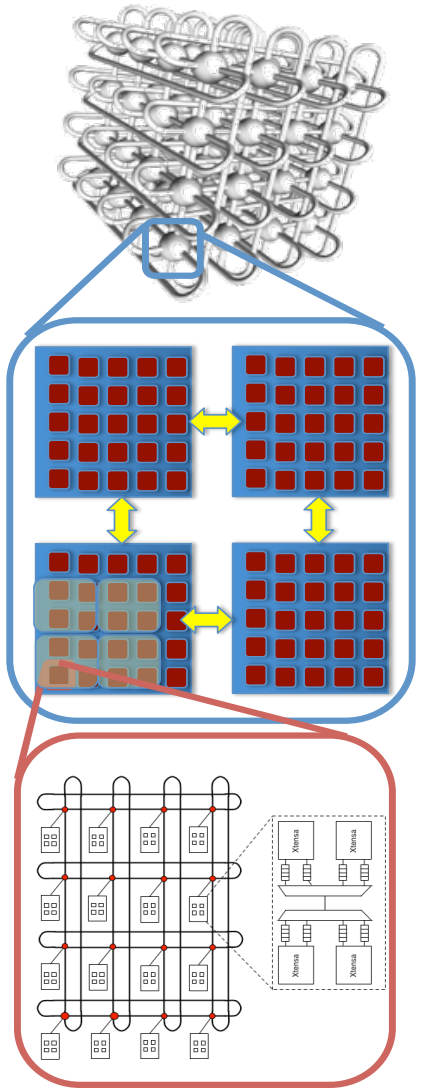
Where is the Abstraction?

What do you mean by "Data Centric Programming Model?"

- **Old Model (Compute-Centric)**
  - Describe how to parallelize loop iterations
  - Parallel "DO" divides loop iterations evenly among processors
  - . . . but where is the data located?

- **New Model (Data-Centric)** ***also in big data***
  - Describe how data is laid out in memory
  - Loop statements operate on data where it is located
  - Similar to MapReduce, but need more sophisticated descriptions of data layout for scientific codes

```
forall_local_data(i=0;i<NX;i++;A)
   C[j]+=A[j]*B[i][j]);
```



Its "owner computes" with runtime info for RTS to map

Loops Should Bind to Data Layout and not the other way around!

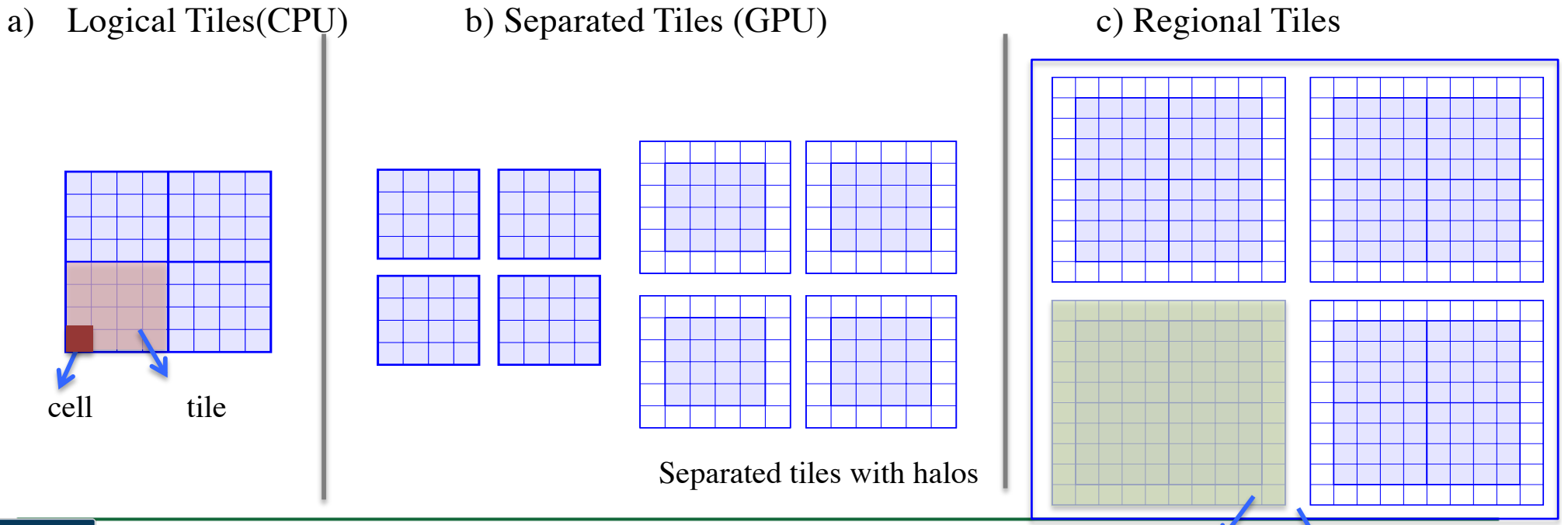
Building up a hierarchical layout

- Layout block coreblk {blockx,blocky};
- Layout block nodeblk {nnx,nny,nnz};
- Layout hierarchy myheirarchy {coreblk,nodeblk};
- Shared myhierarchy double a[nx][ny][nz];

Then use data-localized parallel loop

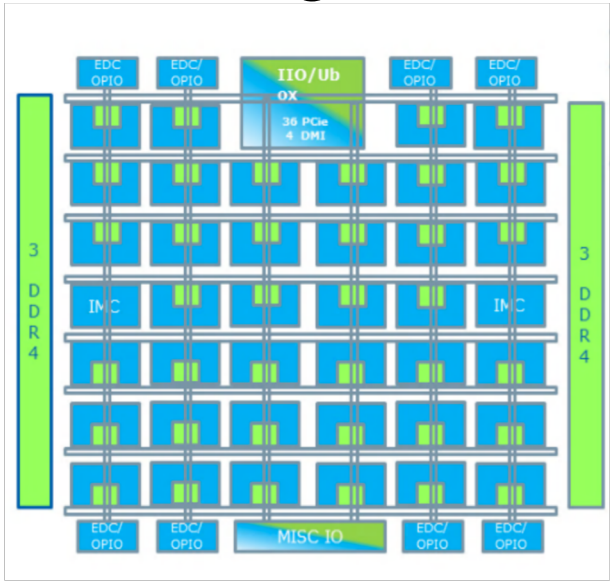
```
doall_at(i=0;i<nx;i++;a){
  doall_at(j=0;j<ny;j++;a){
    doall_at(k=0;k<nz;k++;a){
      a[i][j][k]=C*a[i+1]…>
```

And if layout changes, loop remains the same

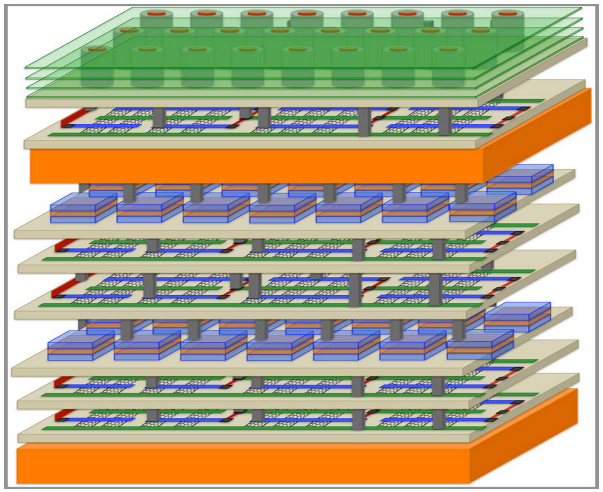


Need Data-Centric Programming Models

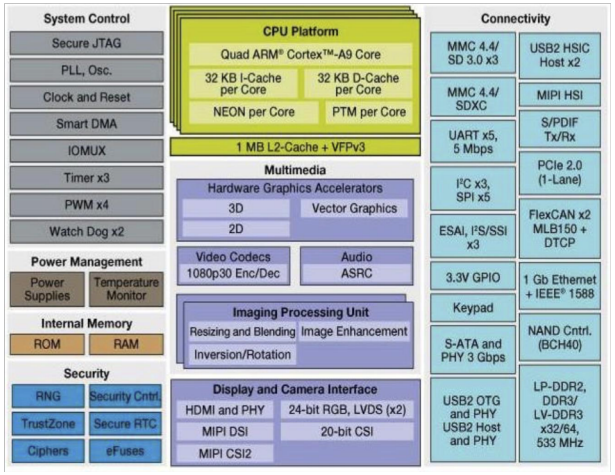
Whether Future Looks Homogeneous



Or Stacked



Or Heterogeneous



A Programming Model that Efficiently Expresses Data Locality so that Runtime Can Optimize Is Essential ***(see http://www.padalworkshop.org)***

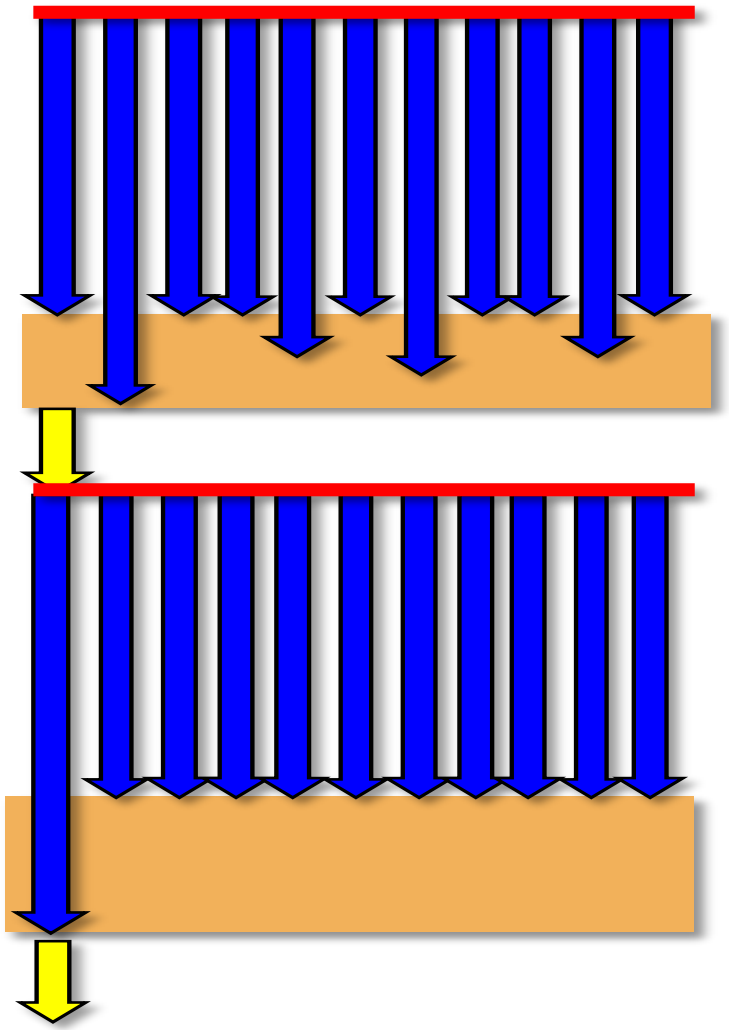
Beyond Bulk Synchronous

How will we handle performance diversity and also hardware diversity?

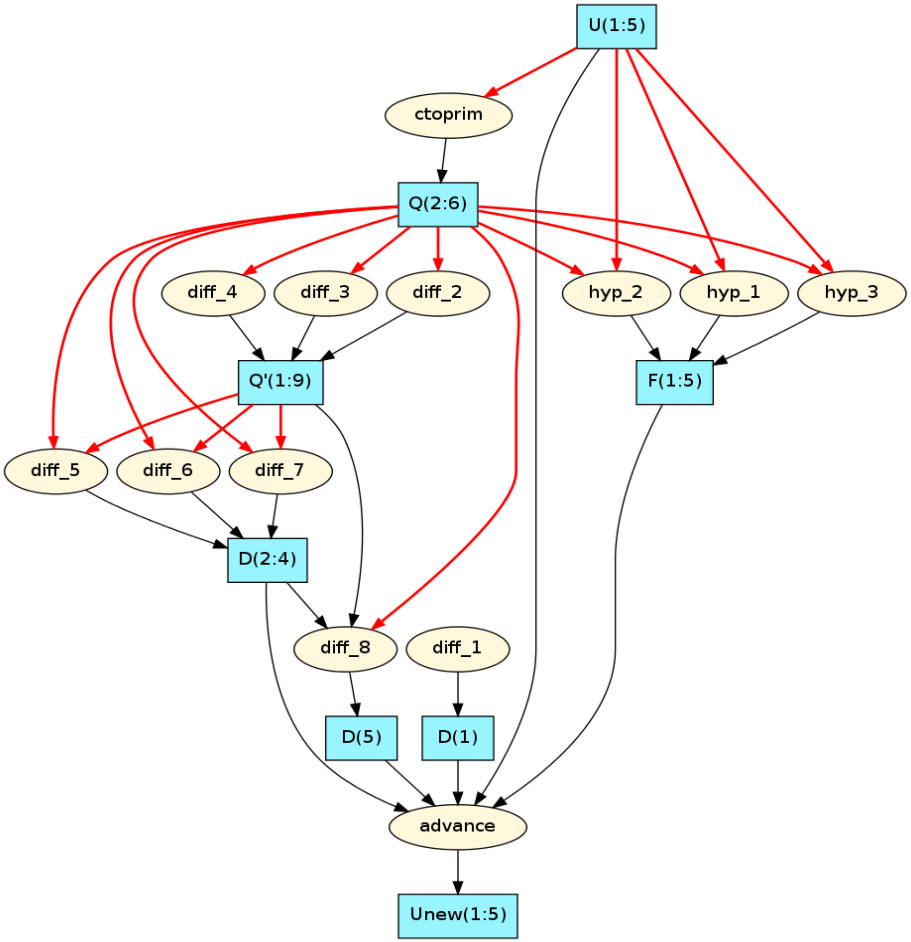
IPAM speakers have made similar observation

Assumptions of Uniformity is Breaking *(many new sources of heterogeneity)*

Bulk Synchronous Parallel Execution Model



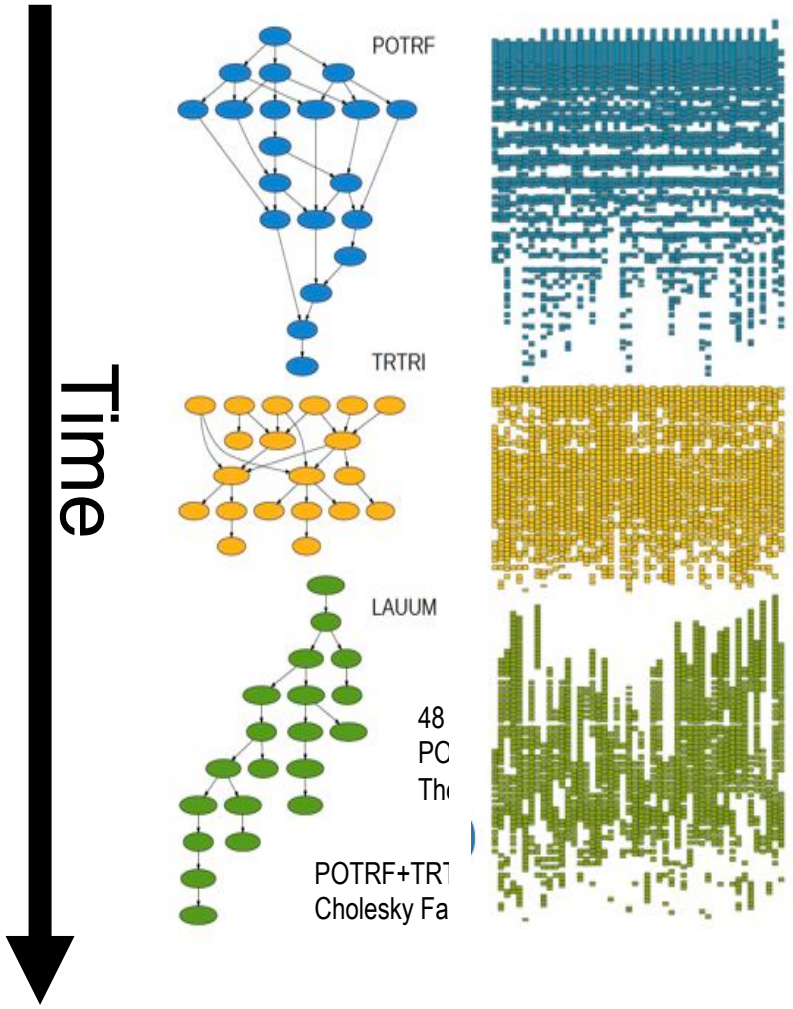
Asynchronous / DAG Execution Model



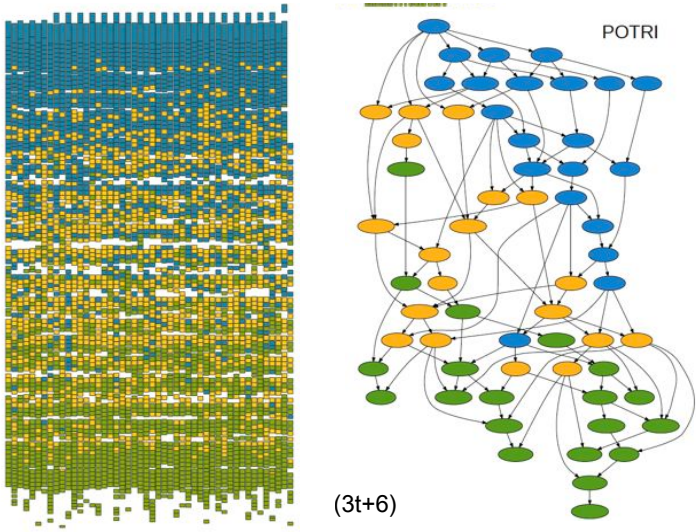
Beyond Bulk-Synchronous Execution

J.Dongarra

Bulk Synchronous
(current practice)

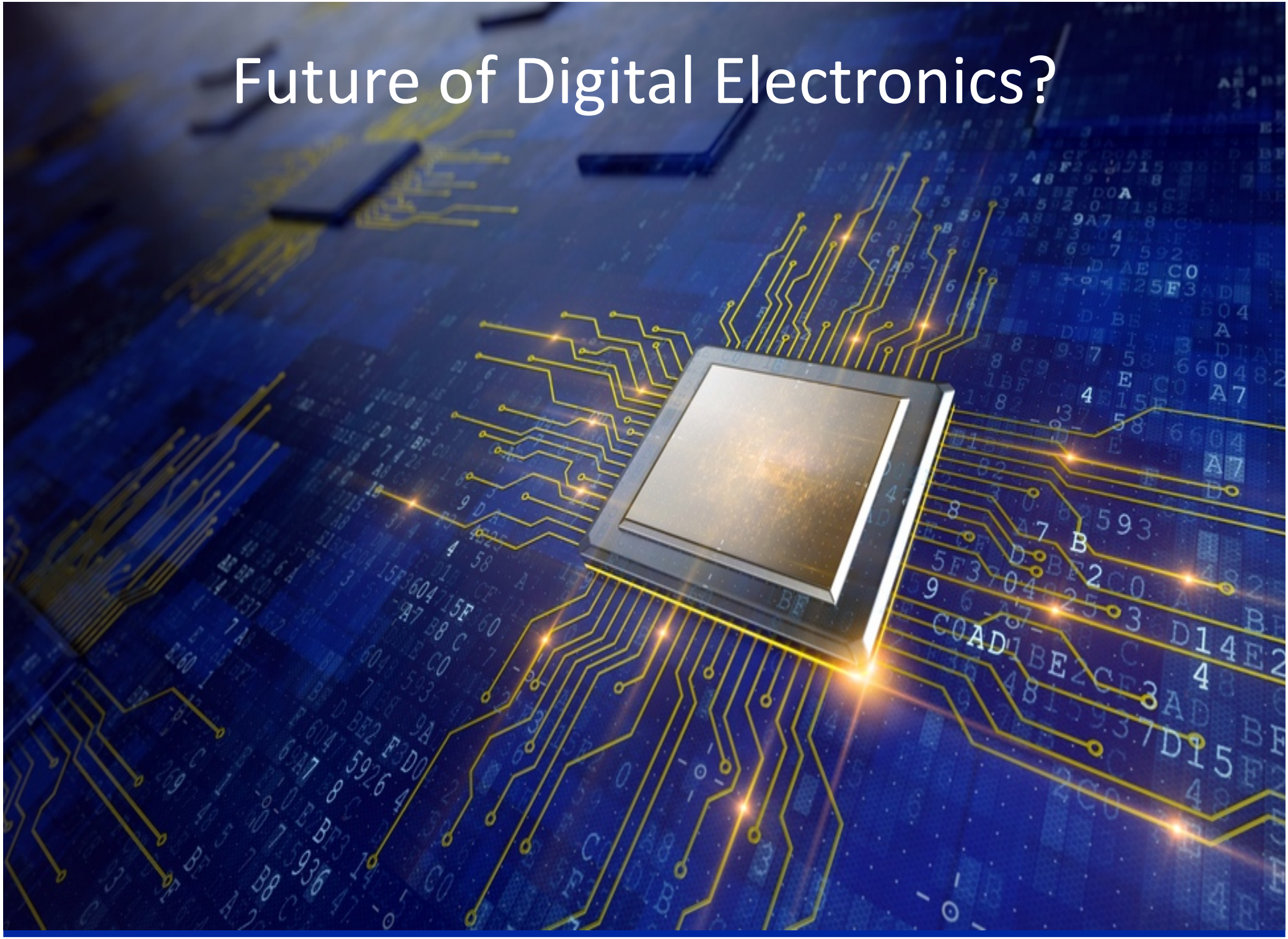


Asynchronous / DAG Model / static schedule
(production interface is still topic of research)



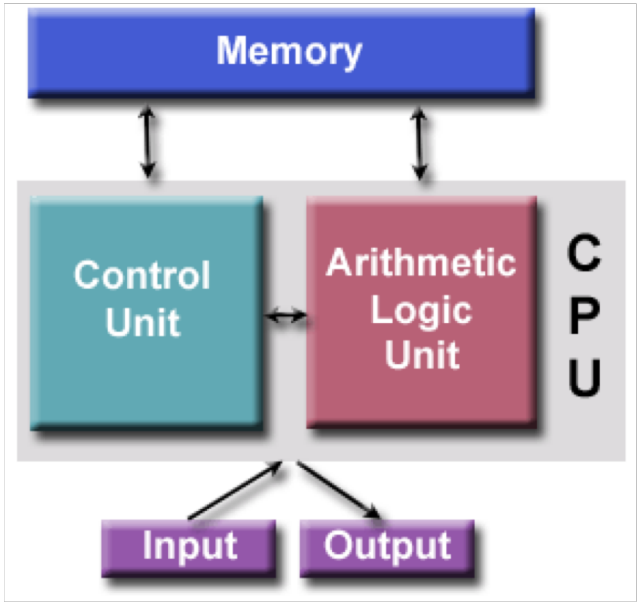
Need to re-examine **Functional semantics for programming systems** or *(taking page from C++)* ***Dysfunctional Languages!***

Future of Digital Electronics?



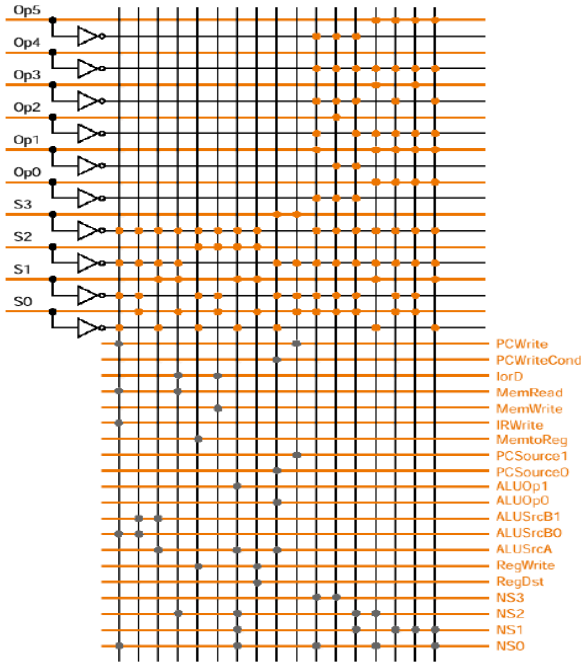
Von Neumann

Von Neumann

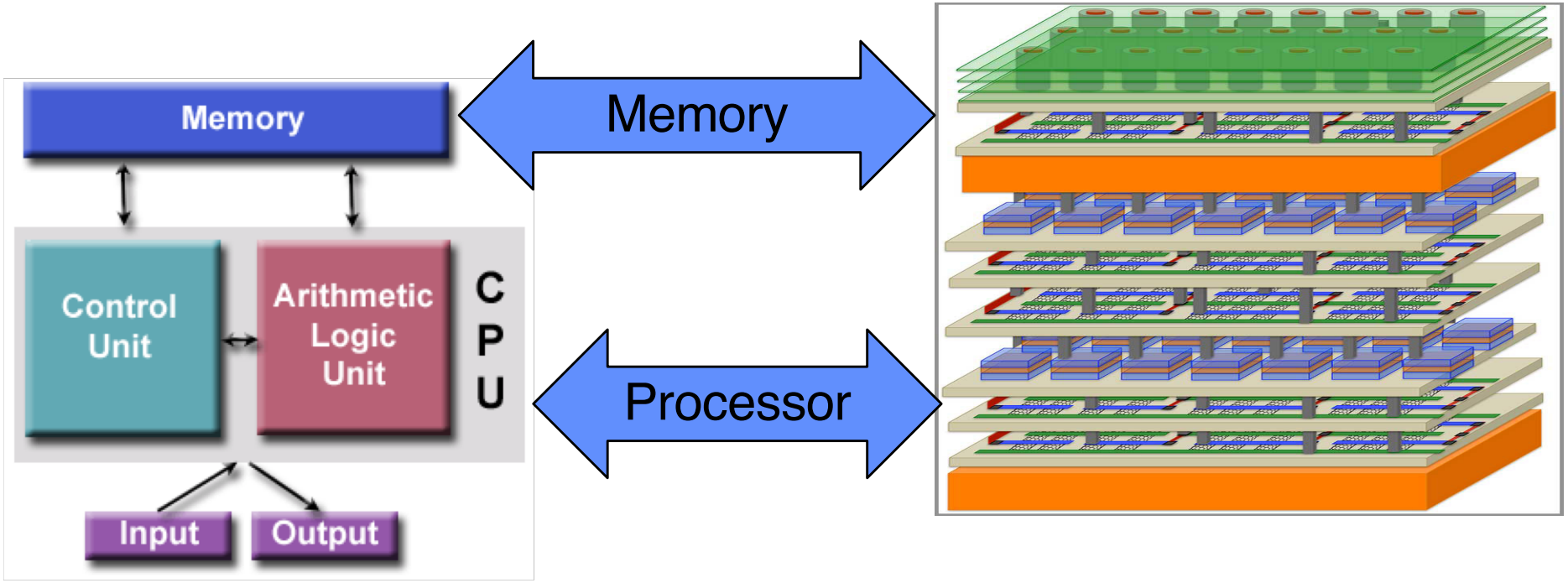


Emulates

Logic Array



PIM is NOT Non-Von Neumann
Its just better packaging

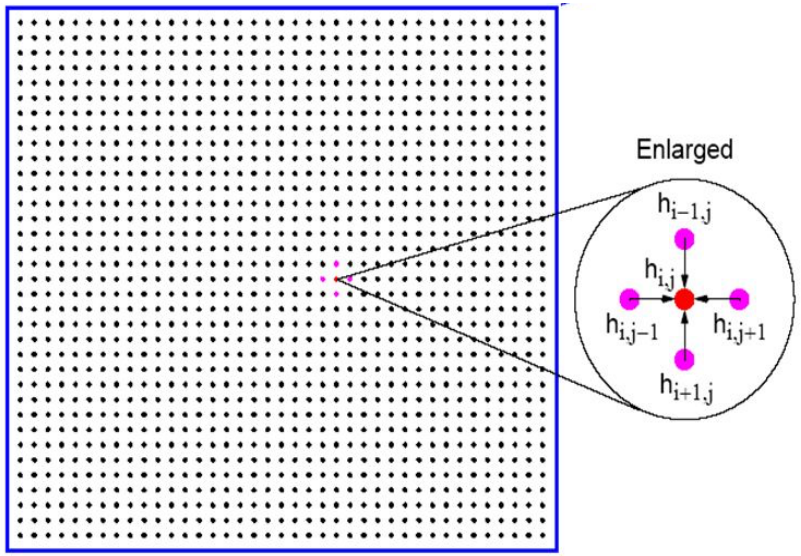


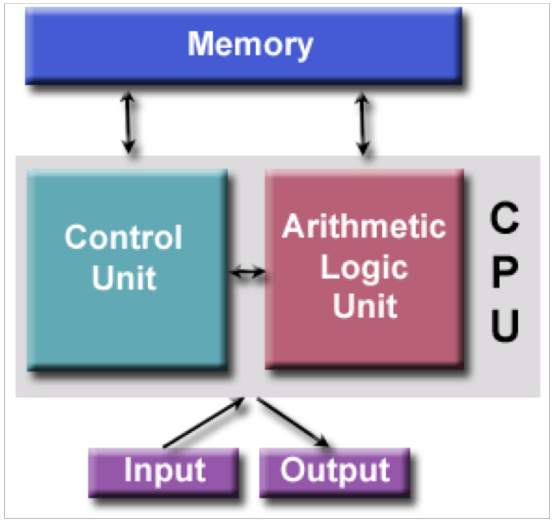
Organizing principles for Non-Von Digital Design

- Data Movement will remain a challenge even with exotic materials, but especially CMOS *(perhaps not with superconducting… ask Burton)*
- Copper is as good of a conductor as you can expect at room temperature
- With even lower power switches, challenges skews even more to data movement *(NEED Spatial Computing approach)*
- Push towards more parallelism (more tesselation of the memory structures)…. Strong Scaling

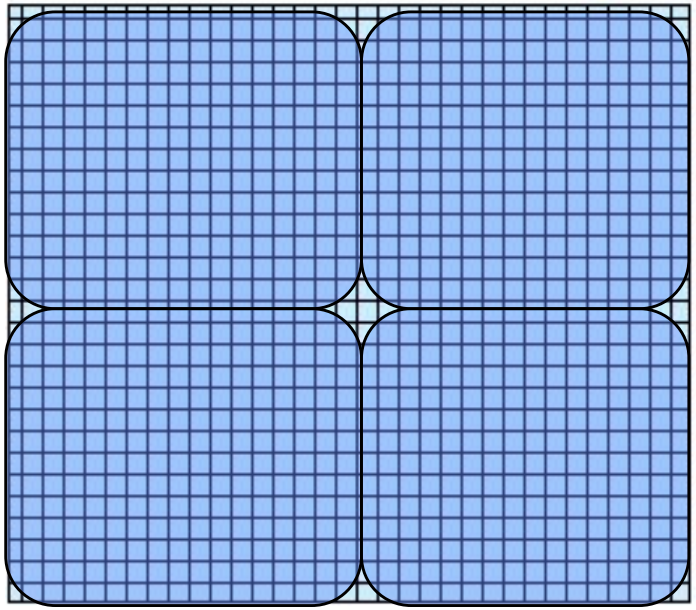
Strong Scaling extrapolates to ***limit case*** with **no separation of memory and compute** *(e.g. one PDE cell per processing element)*

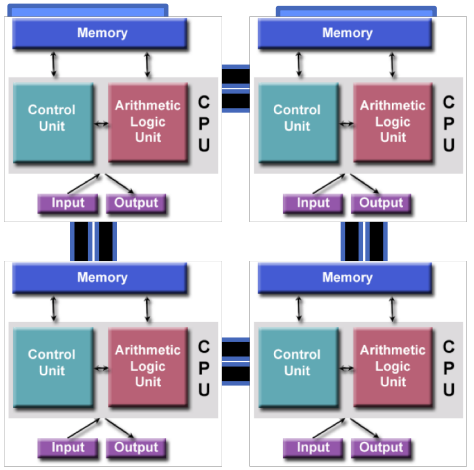
PDE on a Block Structured Grid Extrapolated to Non-Von Neumann



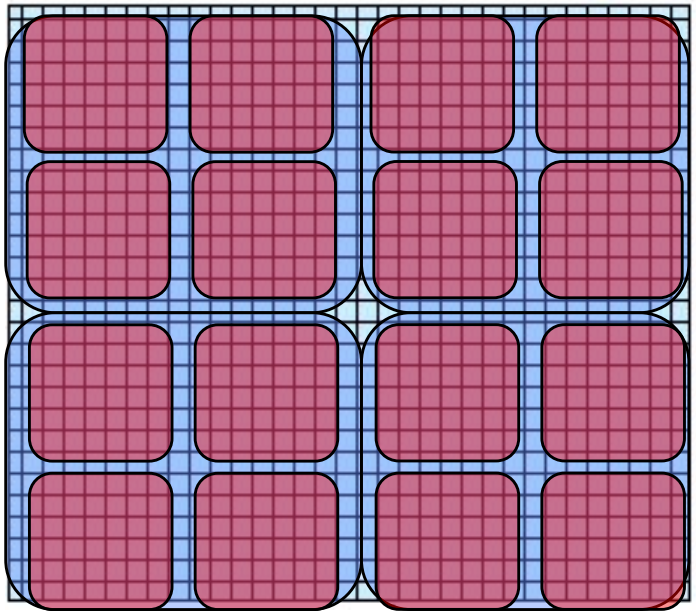


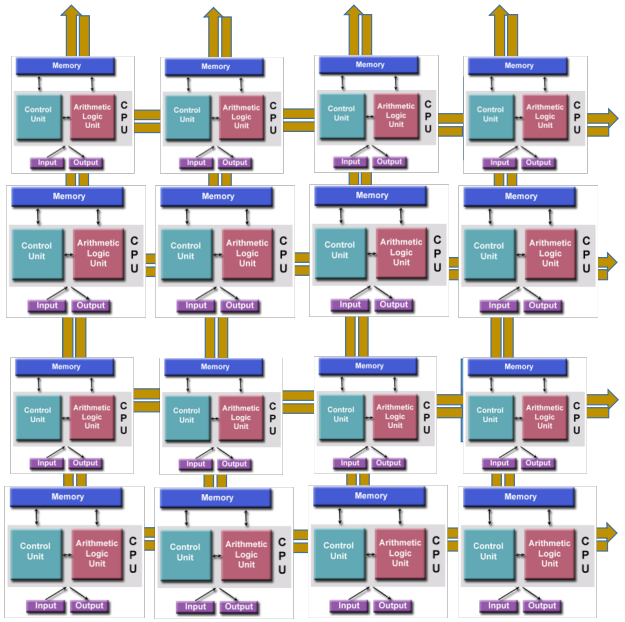
PDE on a Block Structured Grid Extrapolated to Non-Von Neumann



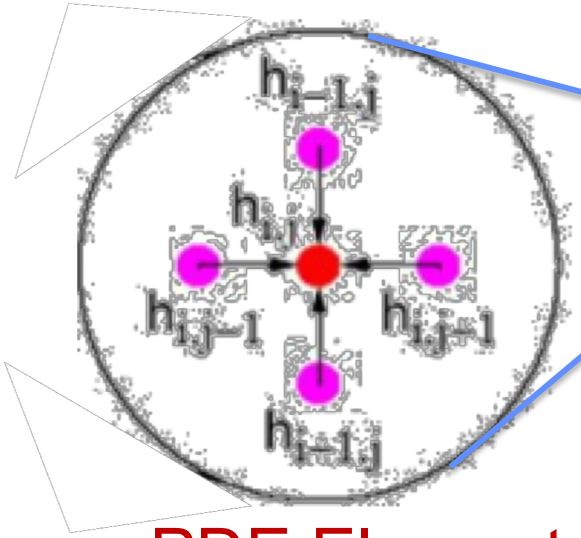


PDE on a Block Structured Grid Extrapolated to Non-Von Neumann

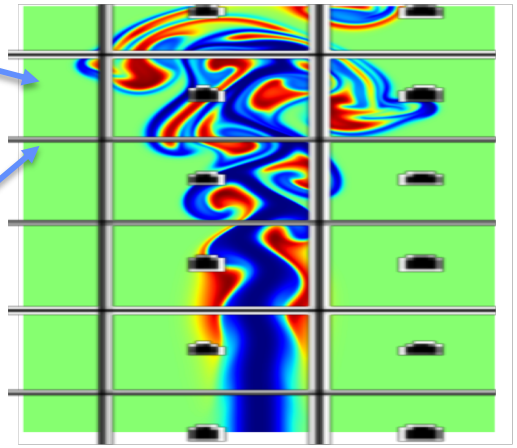




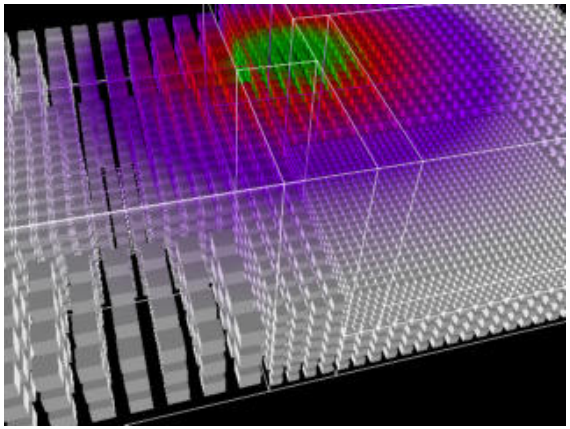
PDE Solvers on Block Structured Grid
”Solid State Digital Fluid”



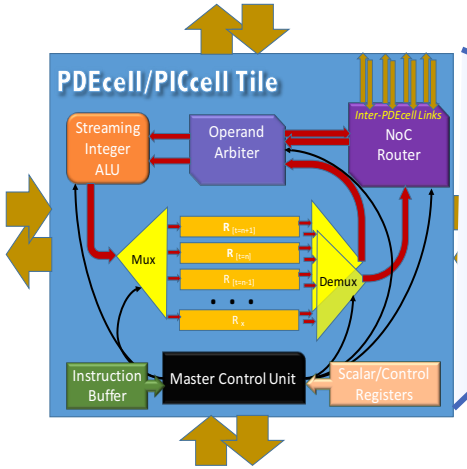
PDE Element

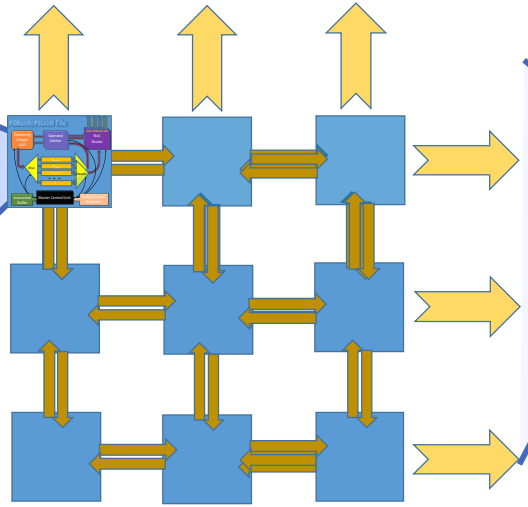


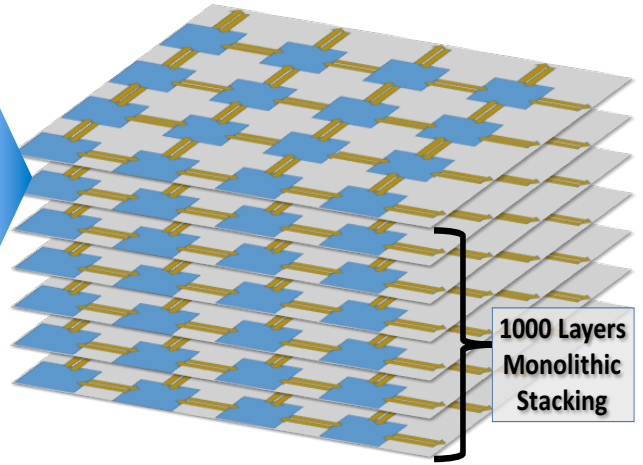
2D Slice



3D problem domain







NOT a CPU

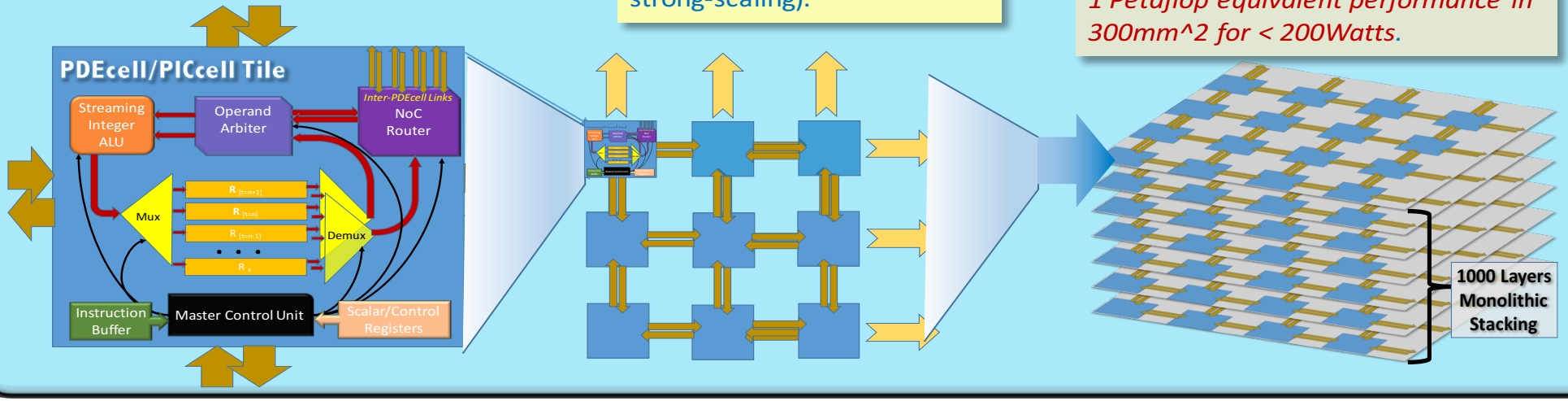
Concept: Solid State Virtual Fluid

Extreme (spatial) Specialization + New Devices + New programming models

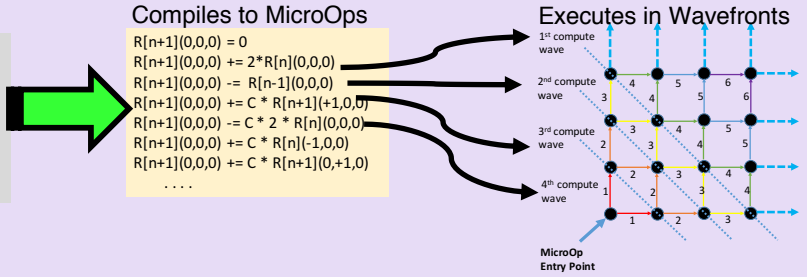
PDEcell / PICcell: Ultra-simple compute engine (50k gates) calculates finite-difference updates, and particle forces from neighbors. Microinstructions specify the PDE equation, stencil, and PIC operators. ***Novel features:*** *variable length streaming integer arithmetic and novel PIC particle virtualization scheme.*

Computational Lattice: PDECells are tiles in a lattice/array on each 2D planar chip layer. Target 120x120 tiles per mm^2 @28nm lithography. Novel Features: each tile represents single cell of computational domain (pushes to limit of strong-scaling).

Monolithic 3D Integration: Integrate layers of compute elements using emerging monolithic 3D chip stacking.
Novel Features: *1000 layer stacking (20x more than current practice). Area efficient inter-layer connectivity and new energy efficient transistor logic (ncFET).*
1 Petaflop equivalent performance in 300mm^2 for < 200Watts.



Virtualize the program that is applied to the data.



Spatial Computing

locality aware domain-specialized hardware

Challenge

- Data Movement dominates energy losses
- Copper is as good of a conductor as you can expect at room temperature
- *Challenge is exacerbated improving energy efficiency of the devices*

Approach

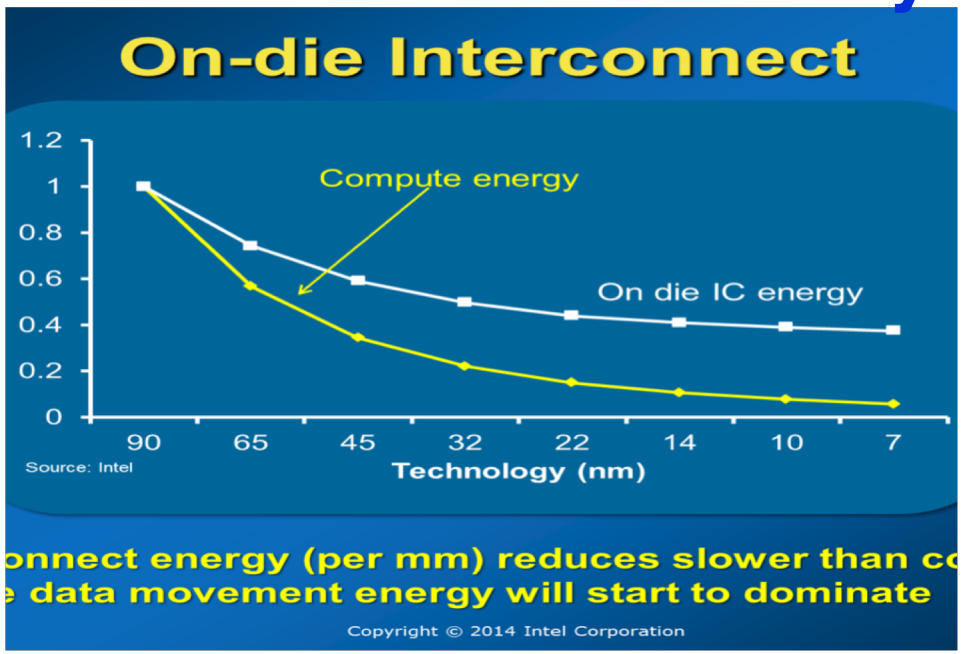
- Identify common computational patterns
- *Design machine that conforms SPATIALLY to a computational pattern in 3D*
- *Design for limit case of parallelism*

Result: Spatial Computing

Extreme Specialization

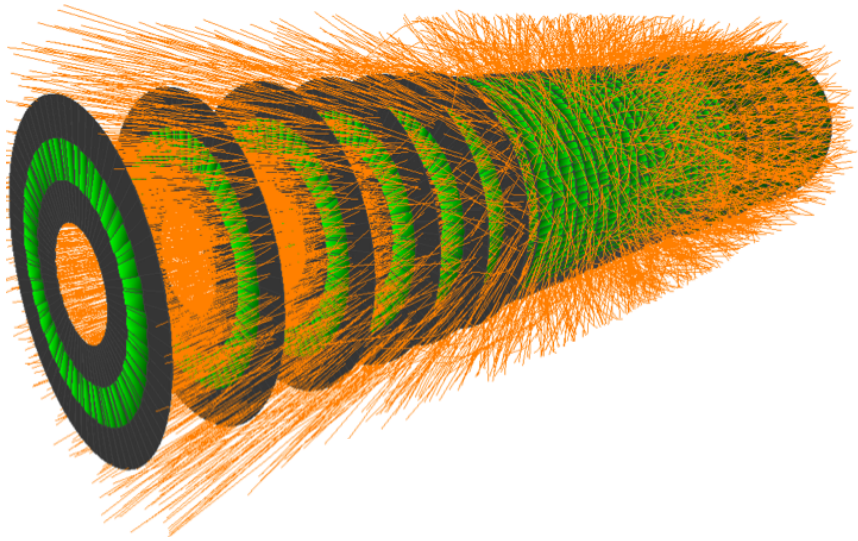
7 Giants of Data (NRC)	7 Motifs of Simulation
Basic statistics	Monte Carlo methods
Generalized N-Body	Particle methods
Graph-theory	Unstructured meshes
Linear algebra	Dense Linear Algebra
Optimizations	Sparse Linear Algebra
Integrations	Spectral methods
Alignment	Structured Meshes

Extreme Data Locality



Spatial Computing Concept for HEP

Following Images from Ted Liu of FNAL
(thliu@fnal.gov)

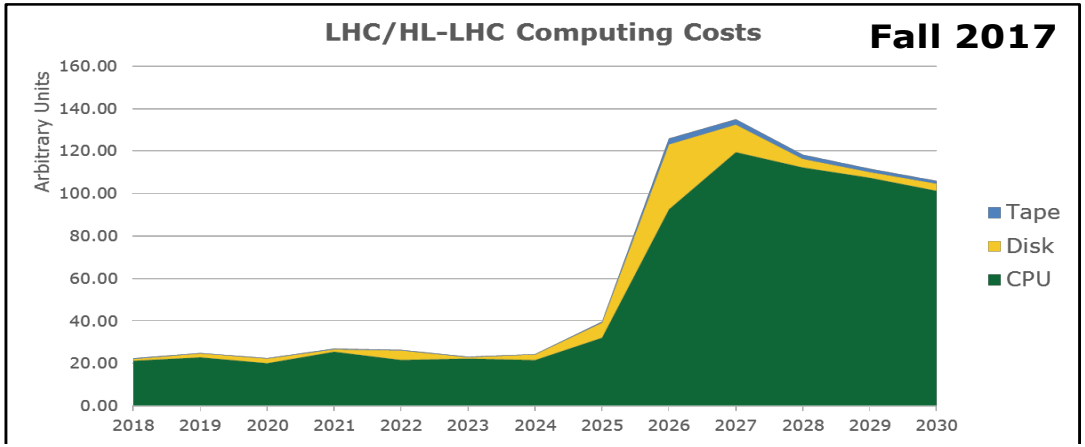


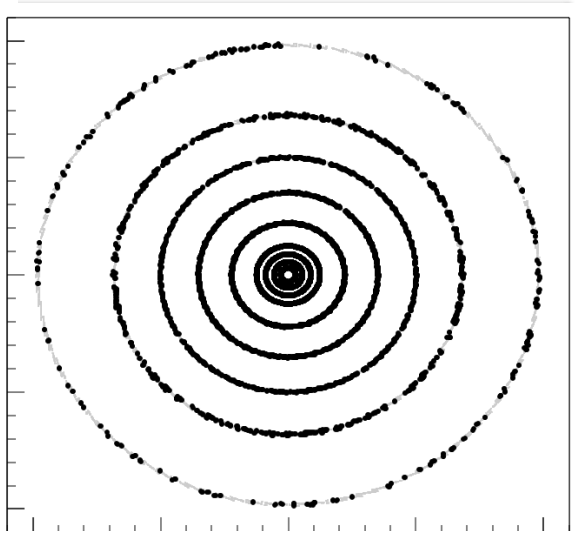
Why do we need more performance?

(Jim Seigrist HEP, May 2018)

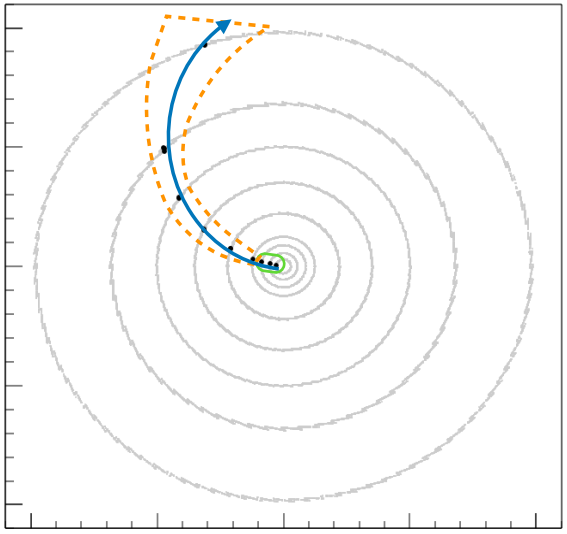
Working group began by conducting an initial survey of the computing needs from each of the three physics Frontiers, and assembled this into a preliminary model

- Energy Frontier portion alone was a large factor beyond the current computing budget
- Large data volumes with the HL-LHC require correspondingly large amounts of computing to analyze it
  - Grid-only solution: **$850M ± 200M**
  - Using the experiments' estimates of future HPC use reduces this to **$650M ± 150M**



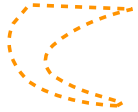


Hits from a fraction of particles in HL-LHC environment



Hits from one particle

seeding
- triplet, quadruplet, n-plet finding



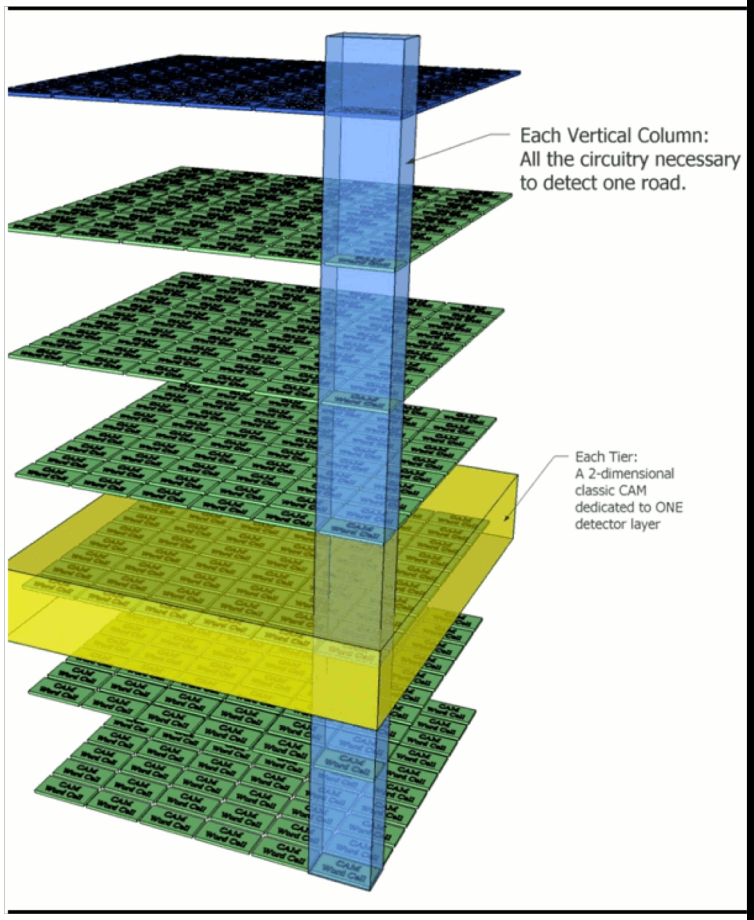
track finding
- combinatorial Kalman filtering

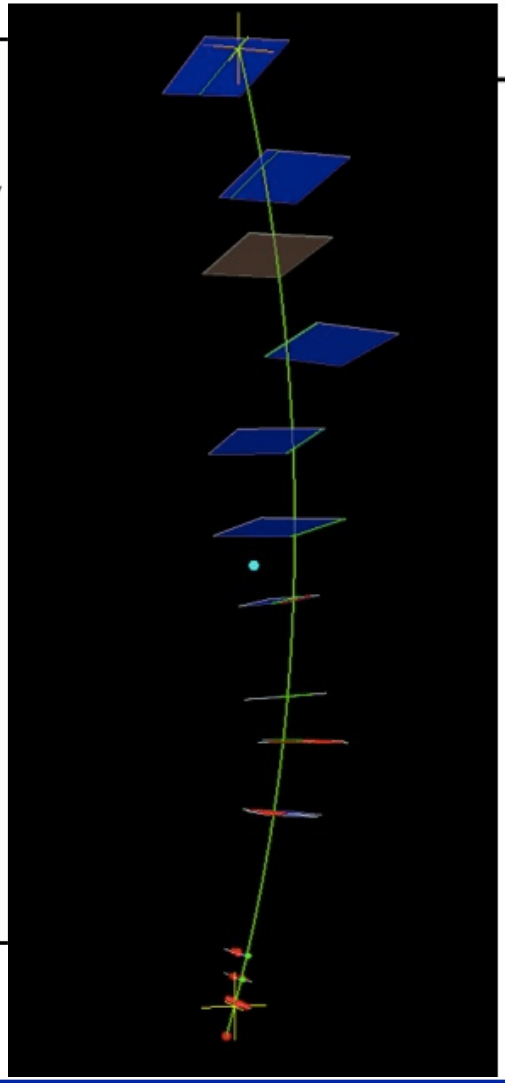
track fitting & classification
- Kalman filtering, smoothing
- χ^2 minimization
- track classification, etc.

"A New Concept of Vertically Integrated Pattern Recognition Associative Memory"

Ted Liu, TIPP 2011 Proceedings
http://www.sciencedirect.com/science/article/pii/S1875389212019165

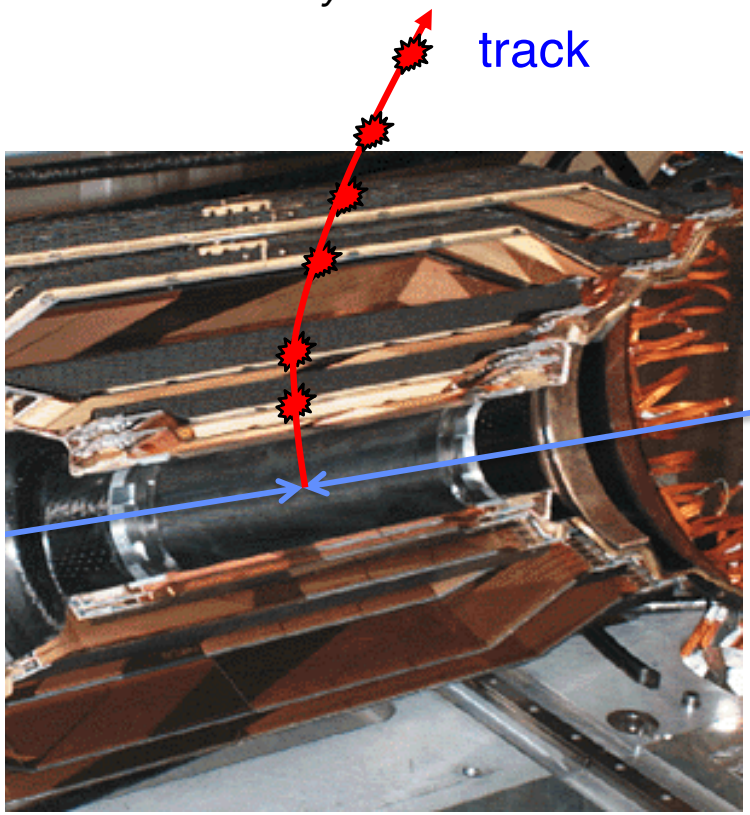
fired road





Pattern recognition for tracking is naturally a task in 3D

track



Conclusion

- **The end of lithography scaling as we know it is coming within a decade** (*close to when Exascale is done*)

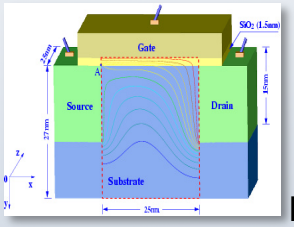
- **Consequence is that parallel processing will be even more central to our future, but has NEW challenges**
  - Diverse Hardware Specialization (*many in single node*)
  - Data Centric Computing (*Spatial Computing*)
  - Non-Bulk Synchronous execution models

- **Cannot ignore this until after exascale**
  - ***Top Science challenges should guide our investments***
  - ***Our next generation of machines should not be judged by "extensive metrics" (scale is a poor metric for success)***
  - ***Judge based on the quality of science enabled.***

Extra

Gaps: Connections and Scaling

Accelerated feedback path
to focus device and material discovery process

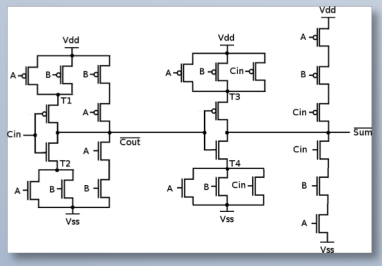


Length Scale

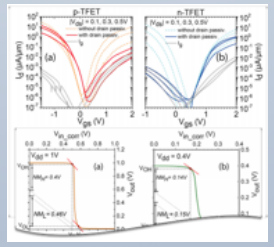
Systems | **Circuits** | **Device Physics** | **Junction Physics** | **Materials Physics**

Architectural Simulation

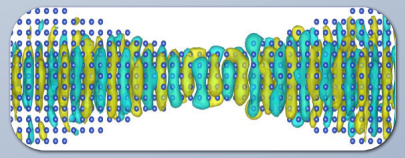
Analog Simulation



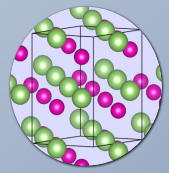
Compact Models



Junctions



Bulk Materials



Processor/System: 10k-100k Circuits

Circuit/Std. Cell: 10-100 Devices

One Device: ~1M Atoms

One Junction: ~100k Atoms

Bulk Material: ~100 Atoms

Length Scales

Device level and materials advances must be understood At the "systems" level! *An end-to-end process is required!*

BERKELEY LAB

Clock-Rates, Power, Area

Current Drive, switching energy, transients

Junction Physics, I-V curves

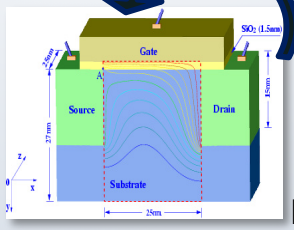
Material Physics Carrier Mobility

OpenSoC System Architect

Gap

PARADISE

Gap



Gap

LS3DF

Gap

Length Scale

Systems

Circuits

Device Physics

Junction Physics

Materials Physics

Application Performance System-Power

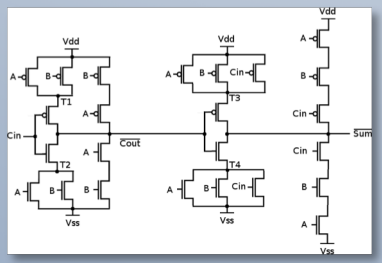
Switch Speed, Power, Area , Fan-out, Stability

Interface-level Losses/Performance

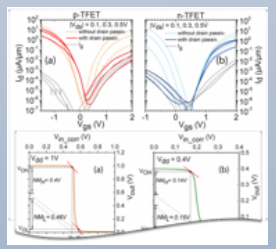
Materials Metrics

Architectural Simulation

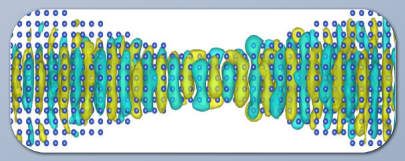
Analog Simulation



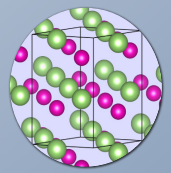
Compact Models



Junctions



Bulk Materials



Processor/System: 10k+ Circuits

Circuit/Std. Cell: 10-100 Devices

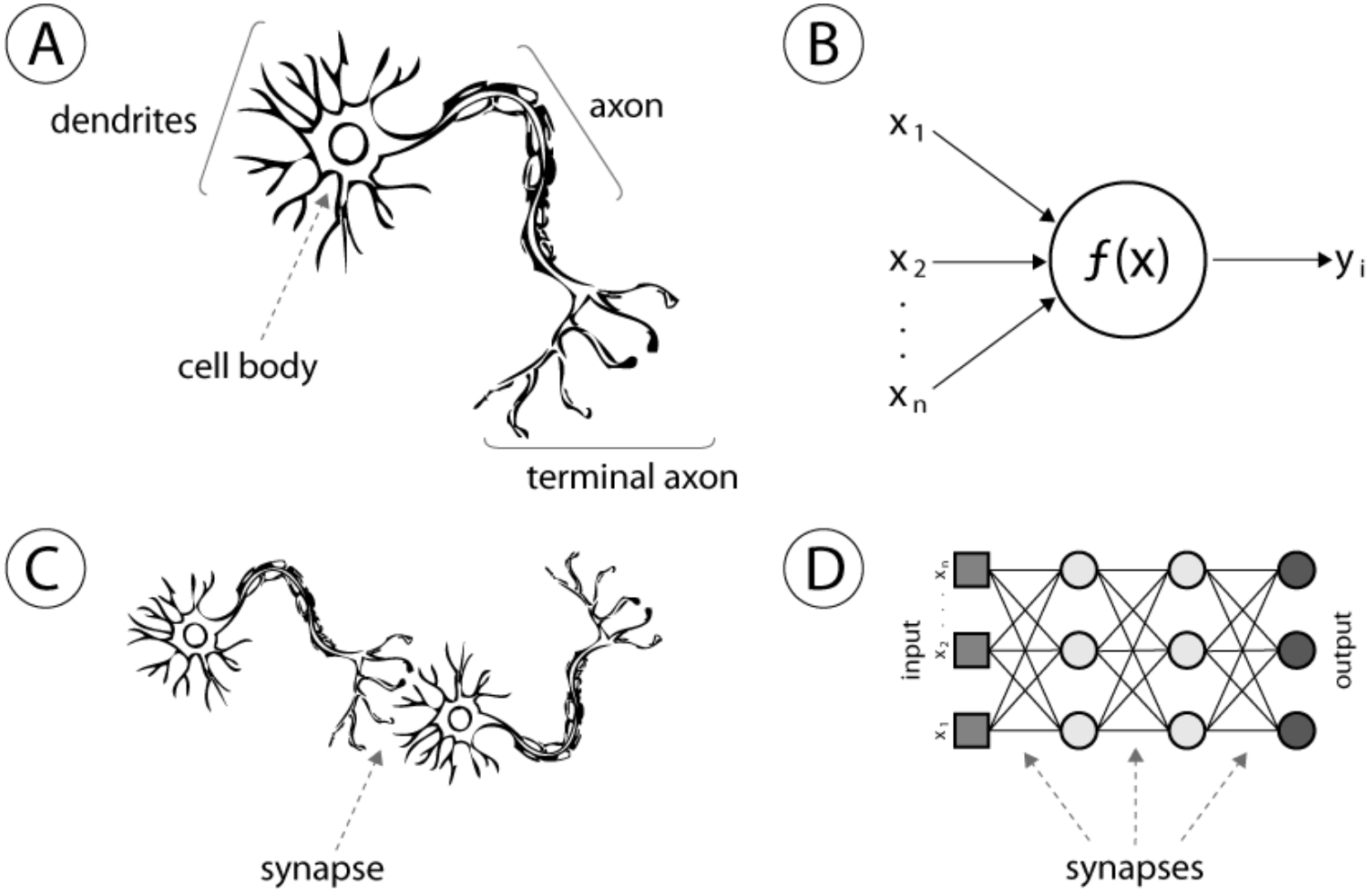
One Device: ~1M Atoms

One Junction: ~100k Atoms

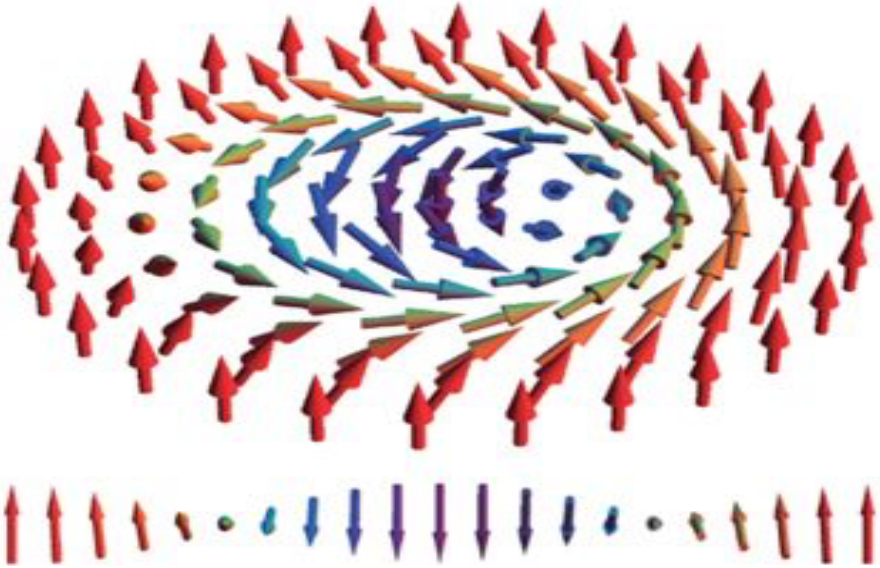
Bulk Material: ~100 Atoms

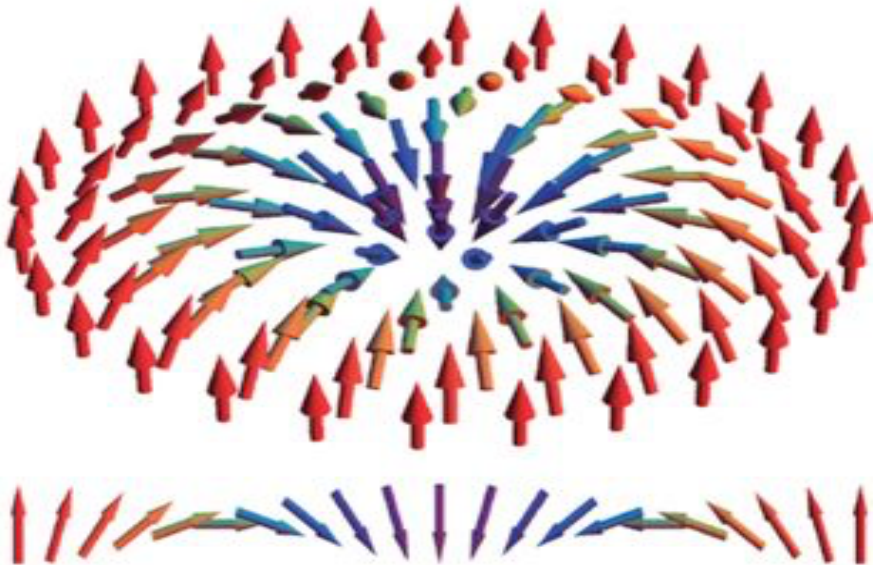
Length Scales

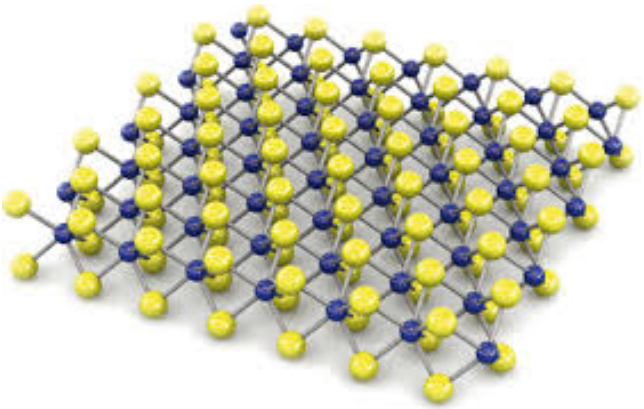
Essence of a Neural Network *(for context)*

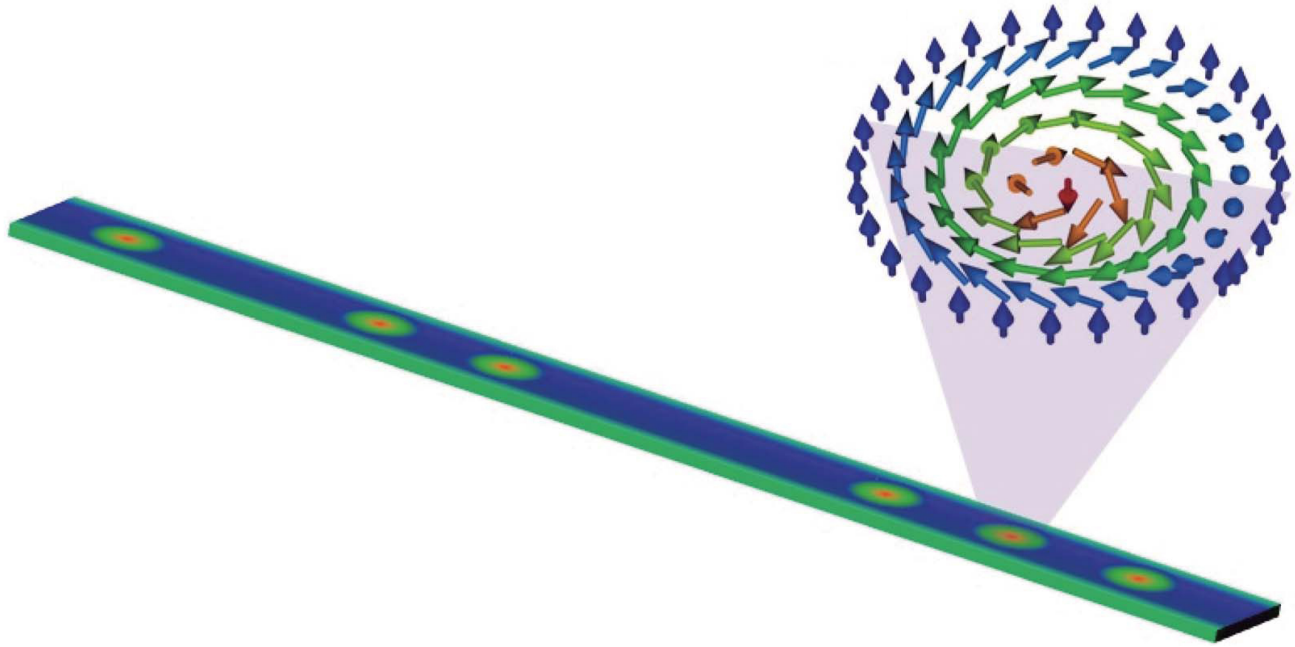


Skyrmions



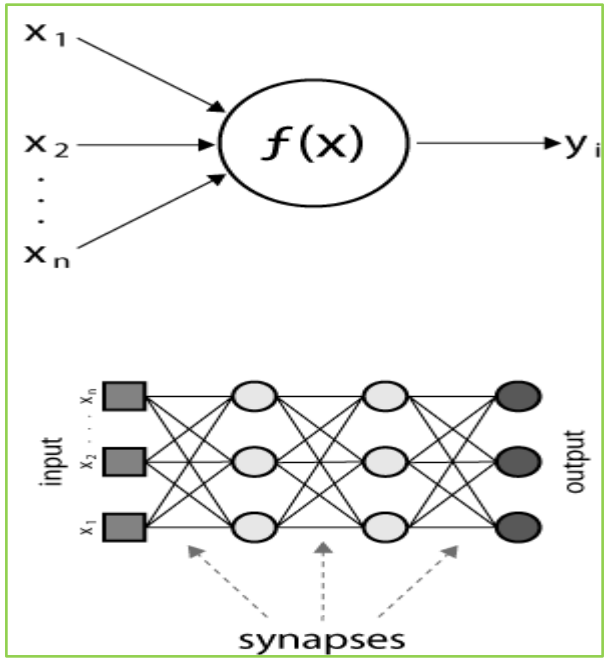


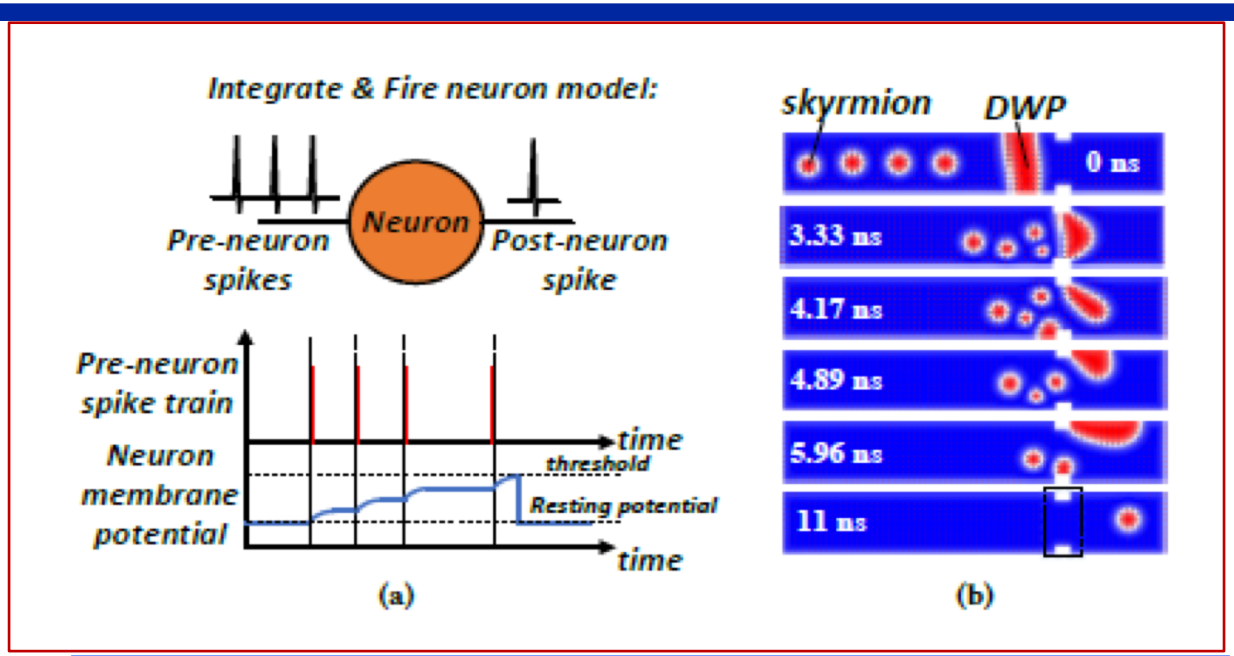


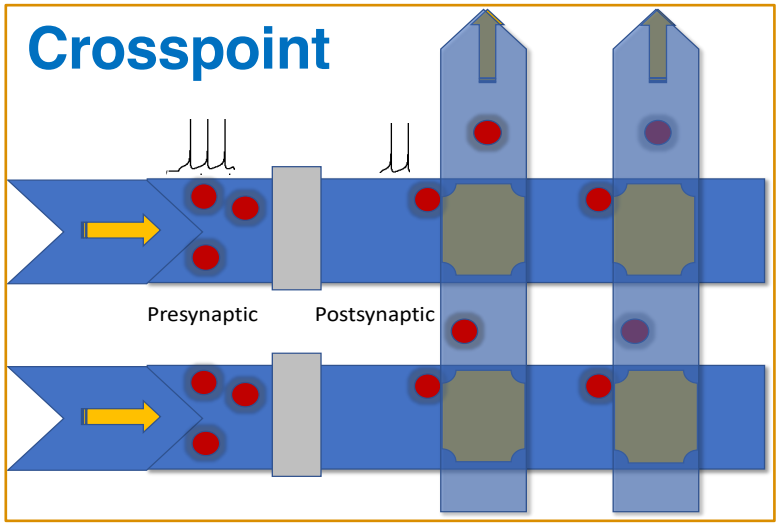


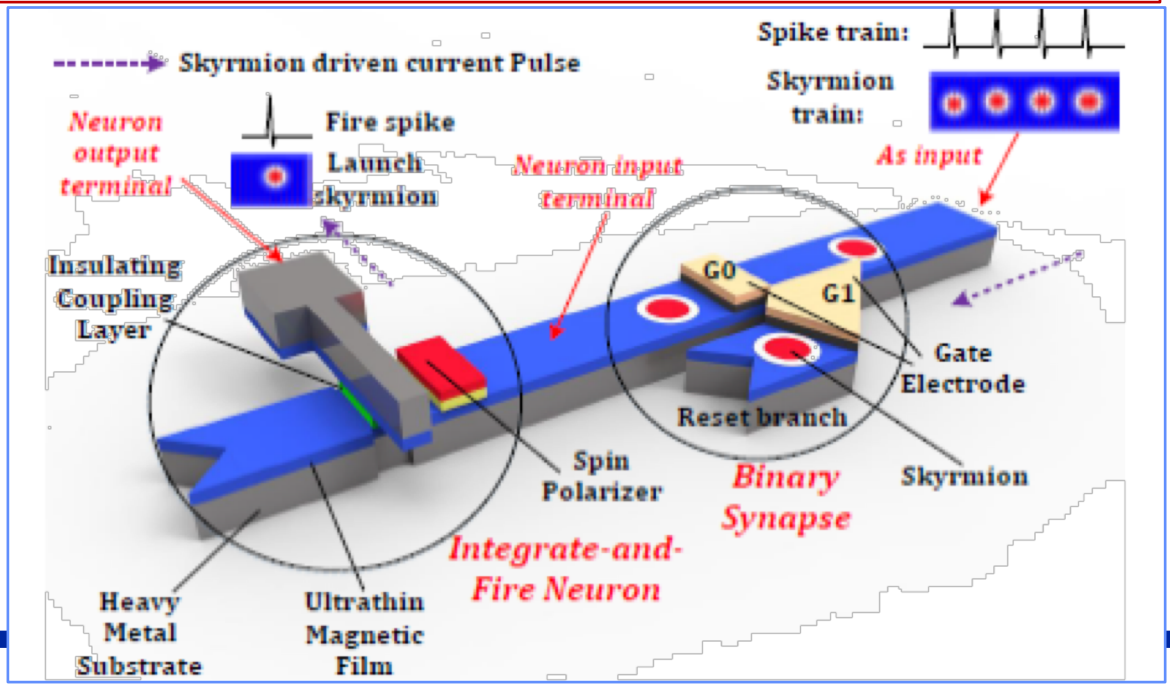
Single-track Skyrmion-Based Spiking Neural Network

Z. He et al., 1705.02995v1 (2017)





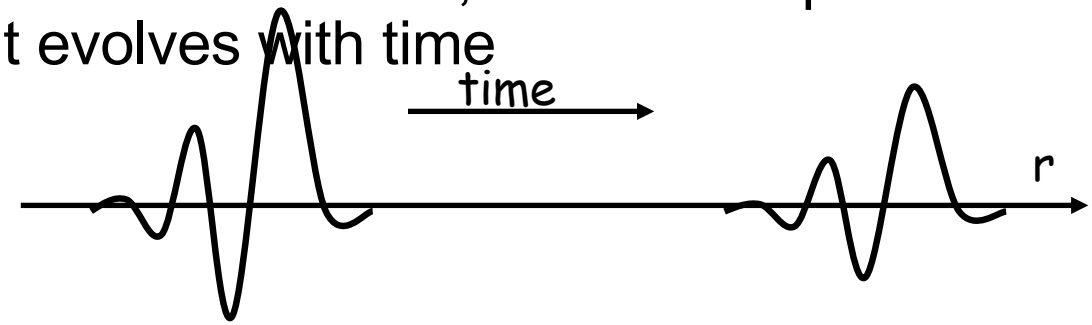




Initial Value Problem (continuous space)

Scalar waves in 3D are solutions of the hyperbolic wave equation: $-\phi_{,tt} + \phi_{,xx} + \phi_{,yy} + \phi_{,zz} = 0$

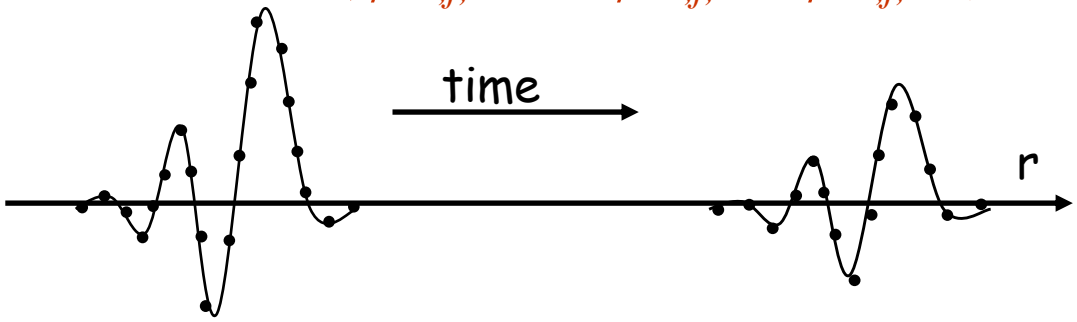
Initial value problem: given data for ϕ and its first time derivative at initial time, the wave equation says how it evolves with time



Discretized Representation

Numerical solve by discretising on a grid, using explicit *finite differencing* (centered, second order)

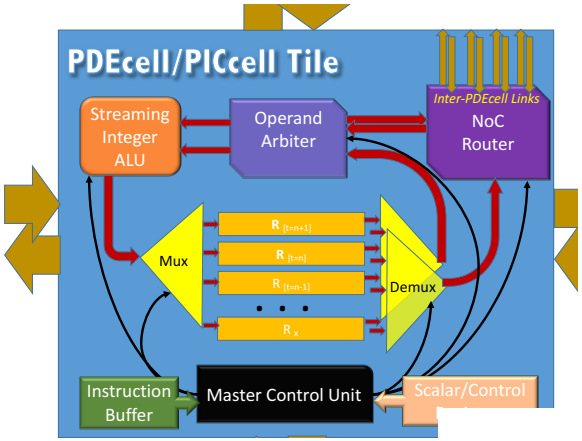
$$\phi^{n+1}_{i,j,k} = 2\phi^{n}_{i,j,k} - \phi^{n-1}_{i,j,k}$$
$$+ \Delta t^2/\Delta x^2(\phi^{n}_{i+1,j,k} - 2\,\phi^{n}_{i,j,k} + \phi^{n}_{i-1,j,k})$$
$$+ \Delta t^2/\Delta y^2(\phi^{n}_{i,j+1,k} - 2\,\phi^{n}_{i,j,k} + \phi^{n}_{i,j-1,k})$$
$$+ \Delta t^2/\Delta z^2(\phi^{n}_{i,j,k+1} - 2\,\phi^{n}_{i,j,k} + \phi^{n}_{i,j,k-1})$$

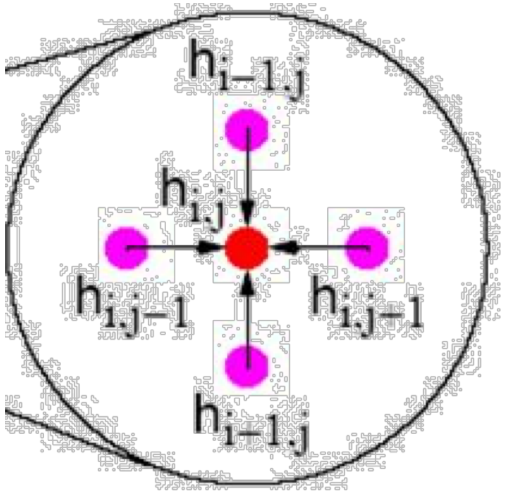


Decomposing Into PDE Weights

$$\phi^{n+1}_{i,j,k} = 2\phi^{n}_{i,j,k} - \phi^{n-1}_{i,j,k}$$
$$+ \Delta t^2/\Delta x^2(\phi^{n}_{i+1,j,k} - 2\,\phi^{n}_{i,j,k} + \phi^{n}_{i-1,j,k})$$
$$+ \Delta t^2/\Delta y^2(\phi^{n}_{i,j+1,k} - 2\,\phi^{n}_{i,j,k} + \phi^{n}_{i,j-1,k})$$
$$+ \Delta t^2/\Delta z^2(\phi^{n}_{i,j,k+1} - 2\,\phi^{n}_{i,j,k} + \phi^{n}_{i,j,k-1})$$

$R_{[t=n+1]}(0,0,0) = 0$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= 2*R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= R_{[t=n-1]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(+1,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(-1,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(0,+1,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(0,-1,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n+1]}(0,0,+1)$
$R_{[t=n+1]}(0,0,0) \mathrel{-}= C * 2 * R_{[t=n]}(0,0,0)$
$R_{[t=n+1]}(0,0,0) \mathrel{+}= C * R_{[t=n]}(0,0,-1)$
Rotate Registers





PDE Domain Specific Representation

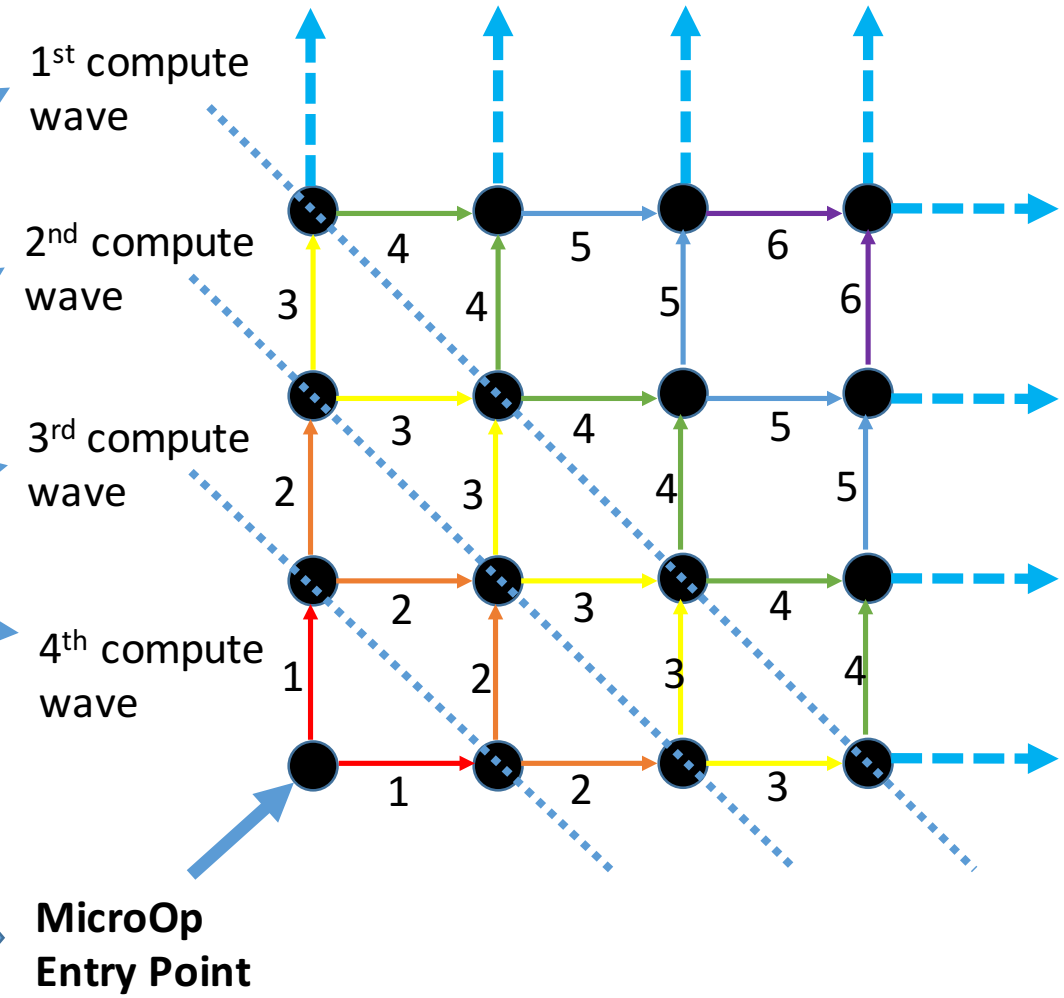
PDE in Domain Specific Representation

$$\phi^{n+1}_{i,j,k} = 2\phi^{n}_{i,j,k} - \phi^{n-1}_{i,j,k}$$
$$+ \Delta t^2/\Delta x^2(\phi^{n}_{i+1,j,k} - 2\,\phi^{n}_{i,j,k} + \phi^{n}_{i-1,j,k})$$
$$+ \Delta t^2/\Delta y^2(\phi^{n}_{i,j+1,k} - 2\,\phi^{n}_{i,j,k} + \phi^{n}_{i,j-1,k})$$
$$+ \Delta t^2/\Delta z^2(\phi^{n}_{i,j,k+1} - 2\,\phi^{n}_{i,j,k} + \phi^{n}_{i,j,k-1})$$

PDE Weights

```
R[n+1](0,0,0) = 0
R[n+1](0,0,0) += 2*R[n](0,0,0)
R[n+1](0,0,0) -=  R[n-1](0,0,0)
R[n+1](0,0,0) += C * R[n+1](+1,0,0)
R[n+1](0,0,0) -= C * 2 * R[n](0,0,0)
R[n+1](0,0,0) += C * R[n](-1,0,0)
R[n+1](0,0,0) += C * R[n+1](0,+1,0)
    . . . .
```

Fed Sequentially to

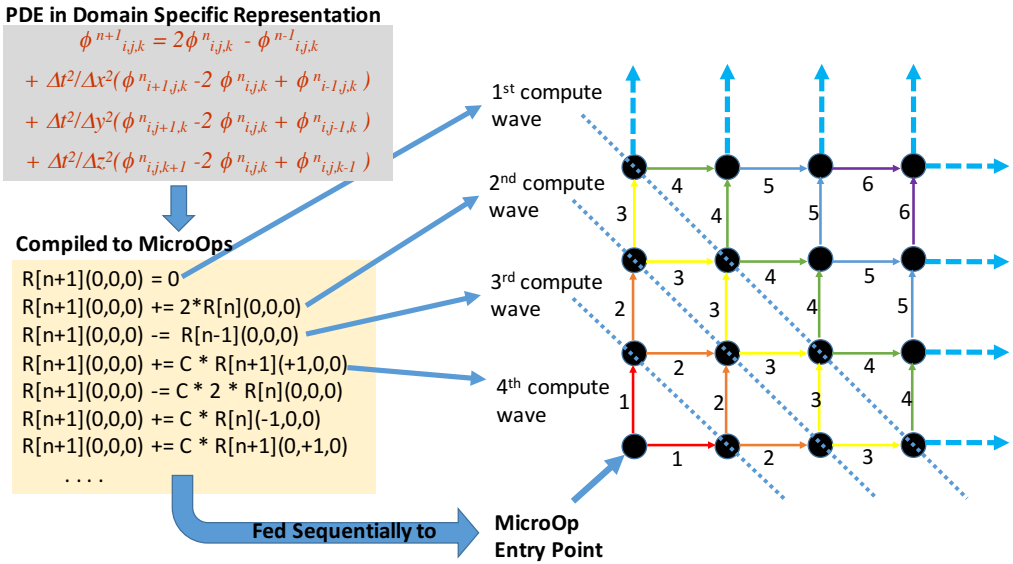


Challenges of Limit-Case Non-Von Neumann Digital

Von Neumann ~= Instruction Processor

```
int main()
{
   int n = 0;
   while(n < 100)
   {
      n = n + 5;
      print("n = %d\n", n);
      pause(200);
      if(n == 50) break;
   }
   print("All done!");
}
```

What the heck is this?



Modern languages (including many classes of DSLs Were designed with instruction processors in mind

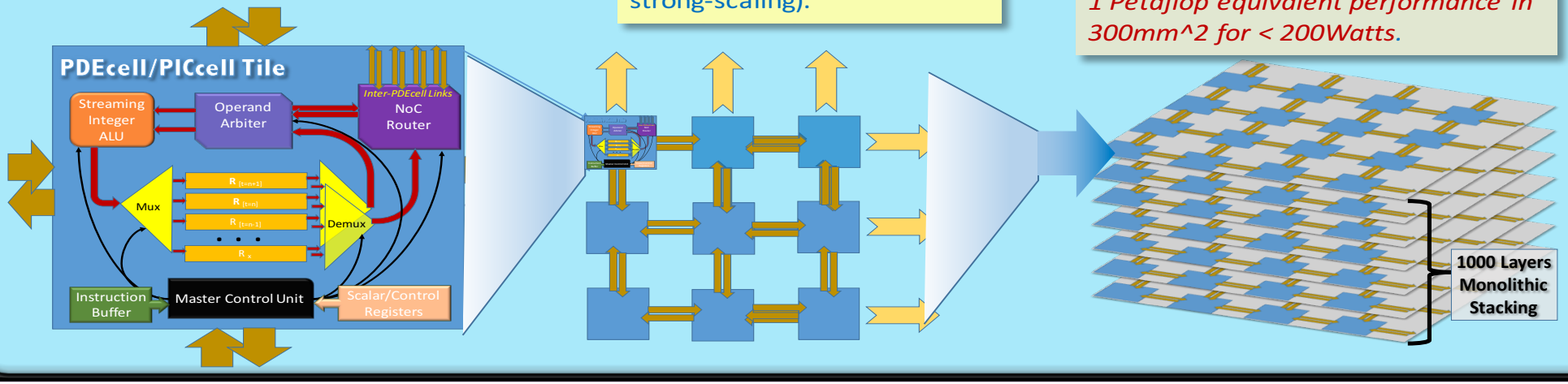
Concept: Solid State Virtual Fluid

Extreme (spatial) Specialization + New Devices + New programming models

PDEcell / PICcell: Ultra-simple compute engine (50k gates) calculates finite-difference updates, and particle forces from neighbors. Microinstructions specify the PDE equation, stencil, and PIC operators. ***Novel features:*** *variable length streaming integer arithmetic and novel PIC particle virtualization scheme.*

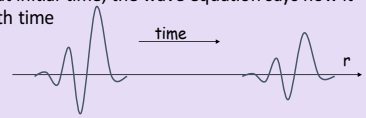
Computational Lattice: PDECells are tiles in a lattice/array on each 2D planar chip layer. Target 120x120 tiles per mm^2 @28nm lithography. Novel Features: each tile represents single cell of computational domain (pushes to limit of strong-scaling).

Monolithic 3D Integration: Integrate layers of compute elements using emerging monolithic 3D chip stacking.
Novel Features: *1000 layer stacking (20x more than current practice). Area efficient inter-layer connectivity and new energy efficient transistor logic (ncFET).*
1 Petaflop equivalent performance in 300mm^2 for < 200Watts.



Scalar waves in 3D are solutions of the hyperbolic wave equation: $-\phi_{,tt} + \phi_{,xx} + \phi_{,yy} + \phi_{,zz} = 0$

Initial value problem: given data for ϕ and its first time derivative at initial time, the wave equation says how it evolves with time



Discretized PDE Representation in DSL

$$\phi^{n+1}_{i,j,k} = 2\phi^{n}_{i,j,k} - \phi^{n-1}_{i,j,k}$$
$$+ \Delta t^2/\Delta x^2(\phi^{n}_{i+1,j,k} - 2\phi^{n}_{i,j,k} + \phi^{n}_{i-1,j,k})$$
$$+ \Delta t^2/\Delta y^2(\phi^{n}_{i,j+1,k} - 2\phi^{n}_{i,j,k} + \phi^{n}_{i,j-1,k})$$
$$+ \Delta t^2/\Delta z^2(\phi^{n}_{i,j,k+1} - 2\phi^{n}_{i,j,k} + \phi^{n}_{i,j,k-1})$$

Compiles to MicroOps

```
R[n+1](0,0,0) = 0
R[n+1](0,0,0) += 2*R[n](0,0,0)
R[n+1](0,0,0) -= R[n-1](0,0,0)
R[n+1](0,0,0) += C * R[n+1](+1,0,0)
R[n+1](0,0,0) -= C * 2 * R[n](0,0,0)
R[n+1](0,0,0) += C * R[n](-1,0,0)
R[n+1](0,0,0) += C * R[n+1](0,+1,0)
    ....
```

Executes in Wavefronts