SAND2015-1333PE

Scaling Beyond Moore's Law with Processor-In-Memory-and-Storage (PIMS)

Erik P. DeBenedictis

ITRS ERD/IEEE Rebooting Computing Meeting

February 26, 2015

PENDING RELEASE

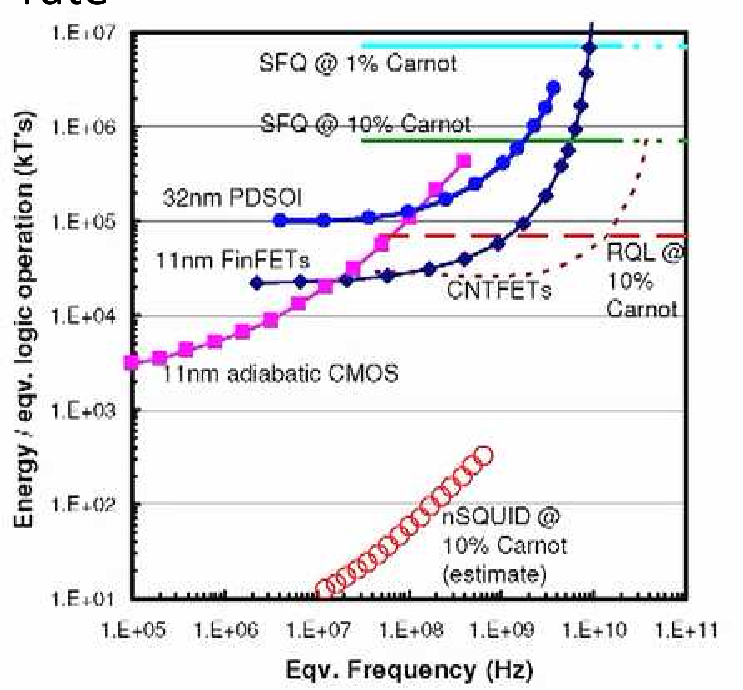
SAND2014-19930 PE and
SAND2014-20016 PE

Outline

- Formulate 3D scaling rule
- Architecture options
  - ☒ von Neumann
  - ☑ Logic-memory integration
- Programming
- Performance
- Device implications

Energy efficiency can depend on clock rate

- David Frank (IBM) discussed adiabatic and reversible computing at RCS 2, where energy efficiency varies by clock rate



From David Frank's presentation at RCS 2; viewgraph 23. "Yes, I'm ok with the viewgraphs being public, so it's ok for you to use the figure. Dave" (10/31/14)

- Adiabatic circuits have behavior close to
  - Energy/op $\propto f$ (clock rate)
  - Power $\propto f^2$
- This would be equivalent to slope 1 on chart at left
- This effect depends on
  - Adiabatic circuitry
  - Devices – 11 nm adiabatic CMOS and nSQUID on David Frank's chart, but many other options
- Let's work with this

A plot will reveal what we will call "optimal adiabatic scaling"

- Impact of manufacturing cost
  - At RCS 2, David Frank put forth the idea that a computer costs should include both purchase cost and energy cost.
  - However, let's adapt this idea to a situation where manufacturing cost drops with time, as in Moore's Law
- Let's plot economic quality of a chip:

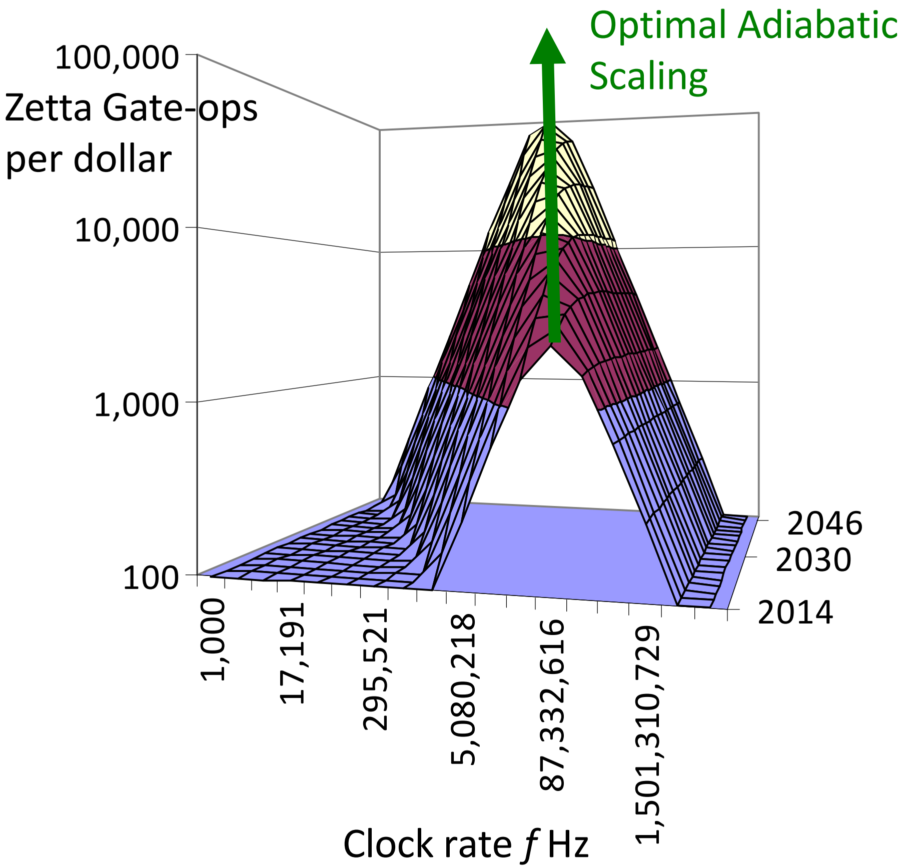
$$Q_{chip} = \frac{Ops_{lifetime}(f)}{\$_{purchase} + \$_{energy}(f^2)}$$

Where $\$_{purchase} = A\, 2^{-t_{year}/3}$

$Ops_{lifetime} = Bf$, *and*

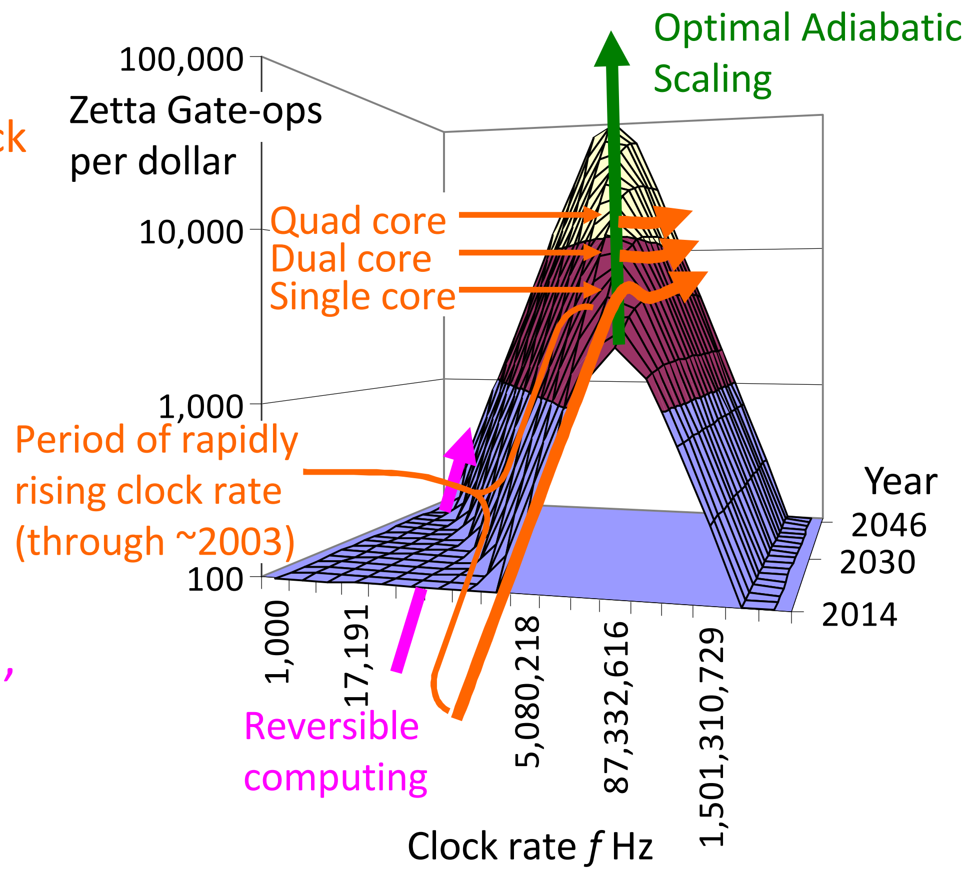
$\$_{energy} = Cf^2$ (A, B, and C constants)

- Assume manufacturing costs drops to ½ every three years
- Top of ridge rises with time



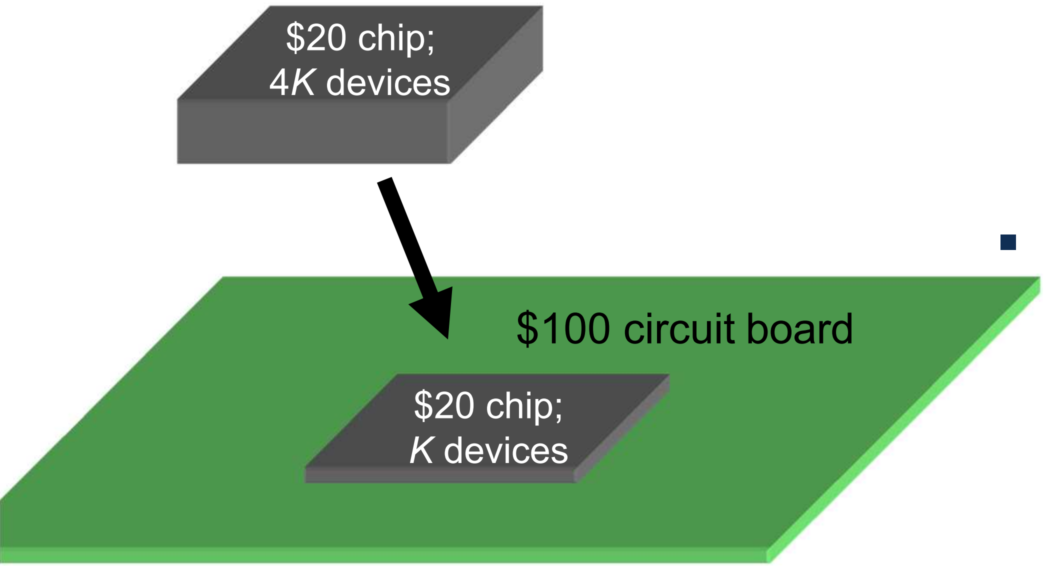
Backup: historical context and reversible computing

- Prior to around 2003, purchase costs dominated energy
  - The economically enlightened approach would be to raise clock rate, which happened
- Around 2003, technology went over the optimal point
  - Multi-core was the technical remedy to the economic problem – had lower clock rate
- Reversible computing would be an advance in the right direction, but too extreme for now



How to derive a scaling rule

- Chip vendor says: "How would you like a chip with 4× as many devices for the same price?"



- Optimal adiabatic scaling says:
  - Cut clock rate to $1/\sqrt{4}\times$ (halve)
  - Power per device drops to 1/4×
  - Power per chip stays same
  - Throughput doubles: 4× as many devices runn at $1/\sqrt{4}\times$ the speed, for a net throughput increase of $\sqrt{4}\times$
- "Throughput" is in accordance with the way throughput is measured for semiconductors, which does not include effects of architecture and algorithms (which we discuss later)
- To make a scaling rule, replace "4" with α^2 (line width scaling)

Resulting scaling scenario (standard chart with additional column)

If C and V stop scaling, throughput ($f\ N_{tran}\ N_{core}$) stops scaling.

Under optimal adiabatic scaling, throughput continues to scale even with fixed V and C

	Const field	Constant V				Optimal Adiabatic Scaling
		Max f	Const f	Const f, N_{tran}	Multi core	
L_{gate}	$1/\alpha$	$1/\alpha$	$1/\alpha$	$1/\alpha$	$1/\alpha$	1 *
W, L_{wire}	$1/\alpha$	$1/\alpha$	$1/\alpha$	1	$1/\alpha$	$N=\alpha^2$ †
V	$1/\alpha$	1	1	1	1	1
C	$1/\alpha$	$1/\alpha$	$1/\alpha$	1	$1/\alpha$	1
$U_{stor} = ½\ CV^2$	$1/\alpha^3$	$1/\alpha$	$1/\alpha$	1	$1/\alpha$	$1/\sqrt{N}=1/\alpha$ ‡
f	α	α	1	1	1	$1/\sqrt{N}=1/\alpha$
N_{tran}/core	α^2	α^2	α^2	1	1	1
N_{core}/A	1	1	1	1	α	$\sqrt{N}=\alpha$
P_{ckt}	$1/\alpha^2$	1	$1/\alpha$	1	$1/\alpha$	$1/\sqrt{N}=1/\alpha$
P/A	1	α^2	α	1	1	1 §
$f\ N_{tran}\ N_{core}$	α^3	α^3	α^2	1	α	$\sqrt{N}=\alpha$

← Theis and Solomon → New

* Term redefined to be line width scaling; 1 means no line width scaling
† Term redefined to be the increase in number of layers; previously was 1 for no scaling
‡ Term redefined to be heat produced per step. Adiabatic technologies do not reduce signal energy, but "recycle" signal energy so the amount turned into heat scales down
§ Term clarified to be power per unit area including all devices stacked in 3D

Ref: T. Theis, In Quest of the "Next Switch": Prospects for Greatly Reduced Power Dissipation in a Successor to the Silicon Field-Effect Transistor, Proceedings of the IEEE, Volume 98, Issue 12, 2010

Outline

Sandia National Laboratories

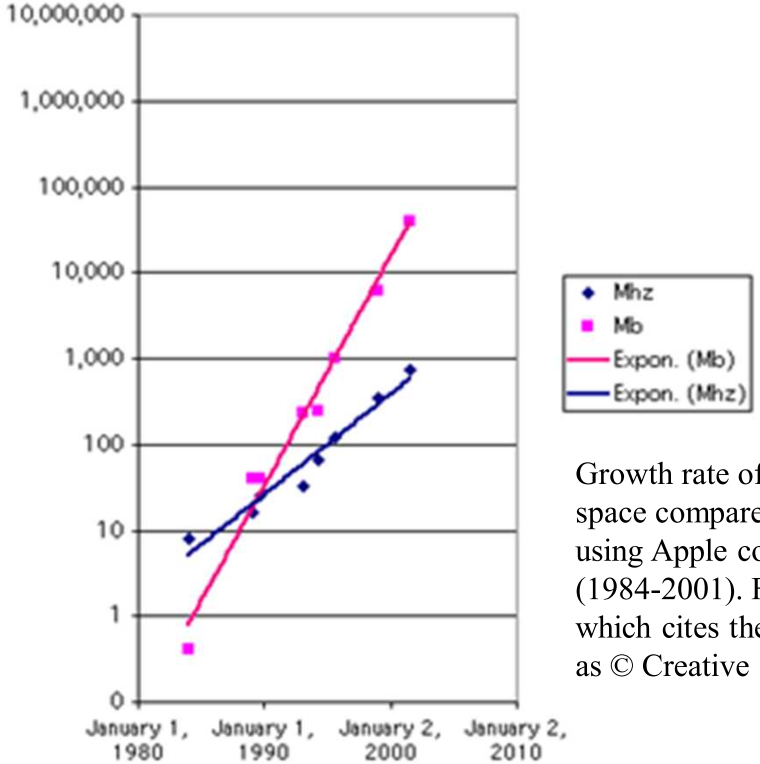
- Formulate 3D scaling rule
- Architecture options
  - ☒ von Neumann
  - ☑ Logic-memory integration
- Programming
- Performance
- Device implications

Need a new architecture; von Neumann architecture won't do

- Optimal adiabatic scaling proportions
  - Device count scales up by N ($N = \alpha^2$)
  - Clock rate scales down by $1/\sqrt{N}$
  - Throughput scales up by $N \times 1/\sqrt{N} = \sqrt{N}$
- The von Neumann architecture cannot exploit this throughput
  - Processor and memory contribute independently to performance
  - Slower computer with more memory – not viable
- We need an architecture whose performance is the product of memory size and clock rate
  - Processor-in-memory?
    - Easily said, but we need a specific architecture that scales properly and has good generality

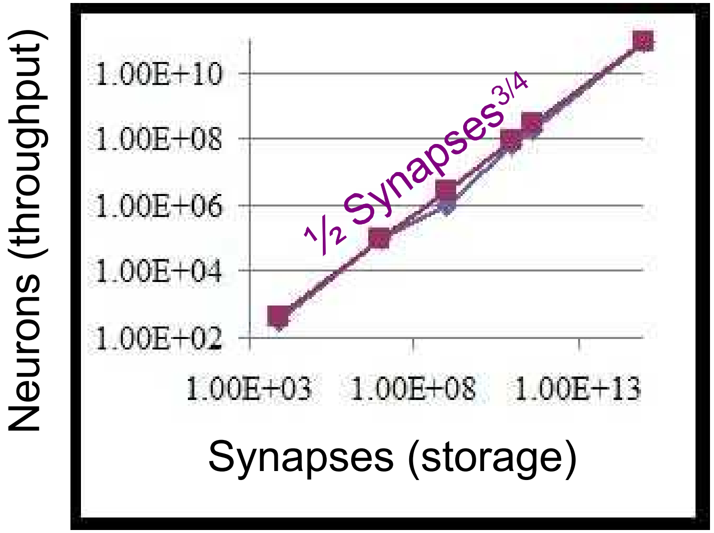
What applications scale like PIMS?

- Computer system clock rate grew at about the square root the rate of storage capacity



Growth rate of HDD storage space compared to clock rate using Apple consumer products (1984-2001). From Wikipedia, which cites the diagram to left as © Creative Commons.

- Brain CPU throughput grows at ¾ power of storage capacity
  - Which is consistent because brains get bigger too



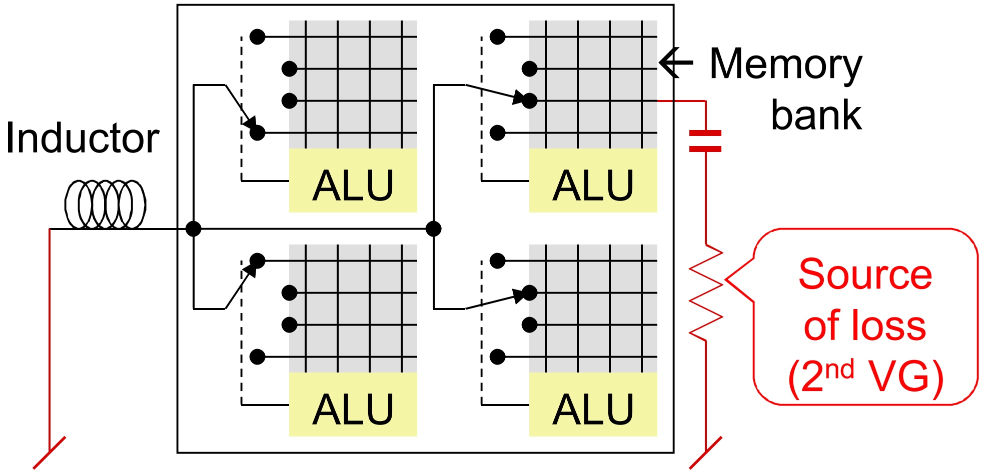
	Synapses	Neurons
Roundworm	7.50E+03	3.02E+02
Fruit fly	1.00E+07	1.00E+05
Honeybee	1.00E+09	9.60E+05
Mouse	1.00E+11	7.10E+07
Rat	4.48E+11	2.00E+08
Human	1.00E+15	8.60E+10

Source: Wikipedia

Design for energy management

- Design around fixing competitor's weakest features:
  - Von Neumann bus/bottleneck
  - CV^2 losses
- Make principal energy pathway into a resonant circuit
  - Recycle the energy that the competitor's system turns into heat

- Chip



- Size expectations for 128 Gb
  - 1024×1024 bits/memory bank
  - 128×128 banks/chip

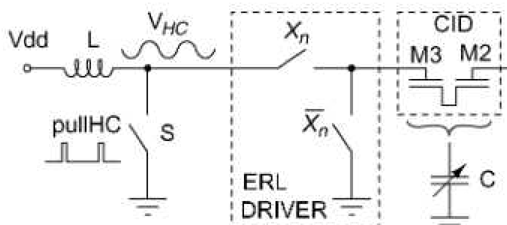
Backup: adiabatic memory (low) maturity level

- Source

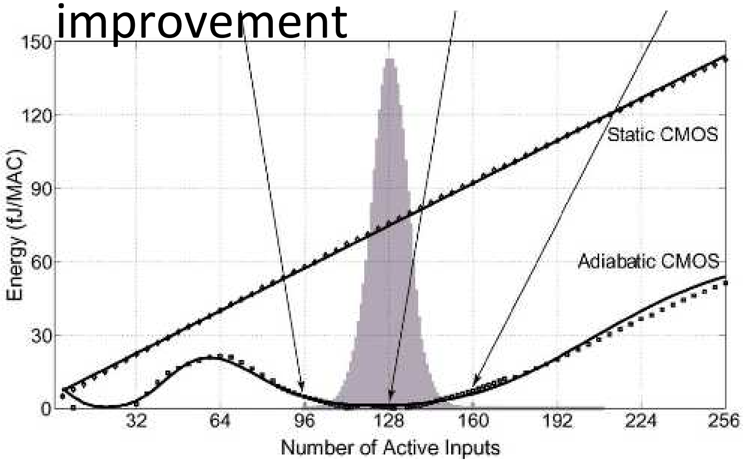
1.1 TMACS/mW Fine-Grained Stochastic Resonant Charge-Recycling Array Processor

Rafal Karakiewicz, *Senior Member, IEEE*, Roman Genov, *Member, IEEE*, and Gert Cauwenberghs, *Fellow, IEEE*

- Energy-recycling row drive



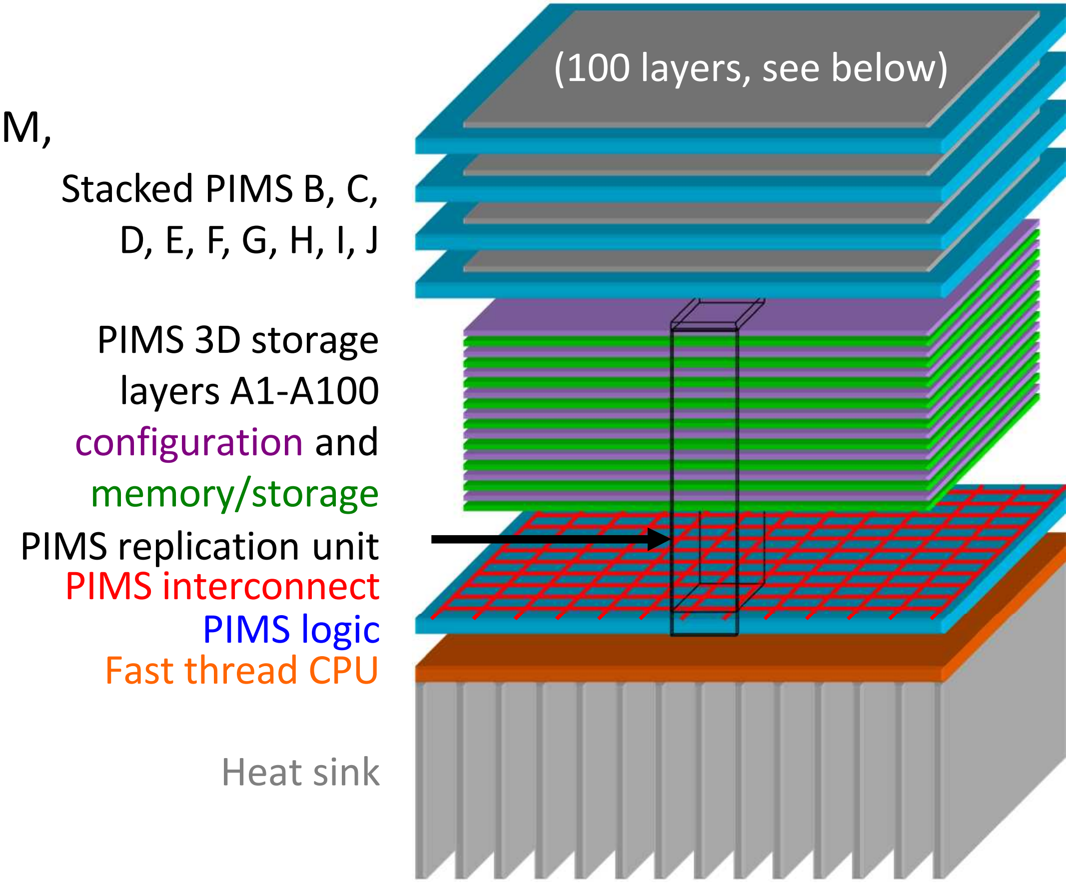
- Result 85× energy efficiency improvement



- TRL 3 or 4 for Charge Injection Devices (CID). TRL definitions:
  - 3. Analytical and experimental critical function and/or characteristic proof of concept
  - 4. Component and/or breadboard validation in laboratory environment
- Above research is for charge injection devices. Author does not see a theoretical reason why it could not work for memristors and flash
- Resonators and inductors ought to be OK

Nominal physical implementation

- Storage/Memory
  - Flash, ReRAM (memristor), STM, DRAM
- Base layer
  - PIMS logic
- 3D
  - Whole structure is layered
- SOME ADDITIONAL DETAIL IN BACKUP



Outline

- Formulate 3D scaling rule
- Architecture options
  - ☒ von Neumann
  - ☑ Logic-memory integration
- Programming
- Performance
- Device implications

Tile programming

x

1	2	3	4

A

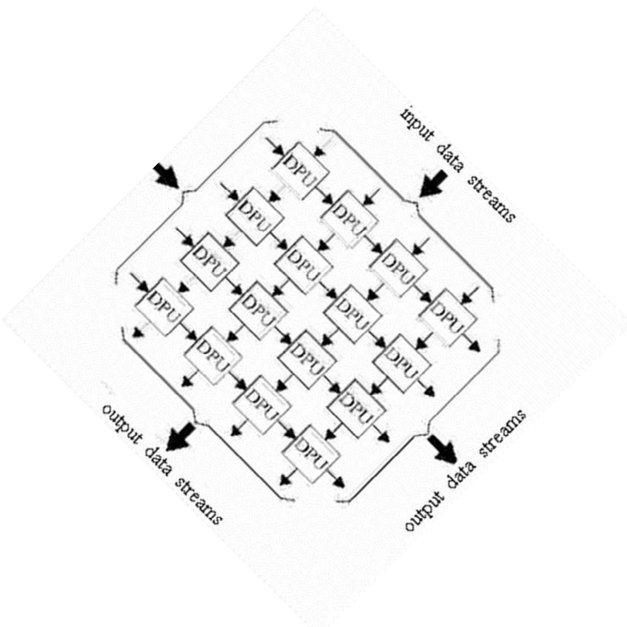
1	0	0	2
0	0	3	0
0	4	0	5
6	0	0	0

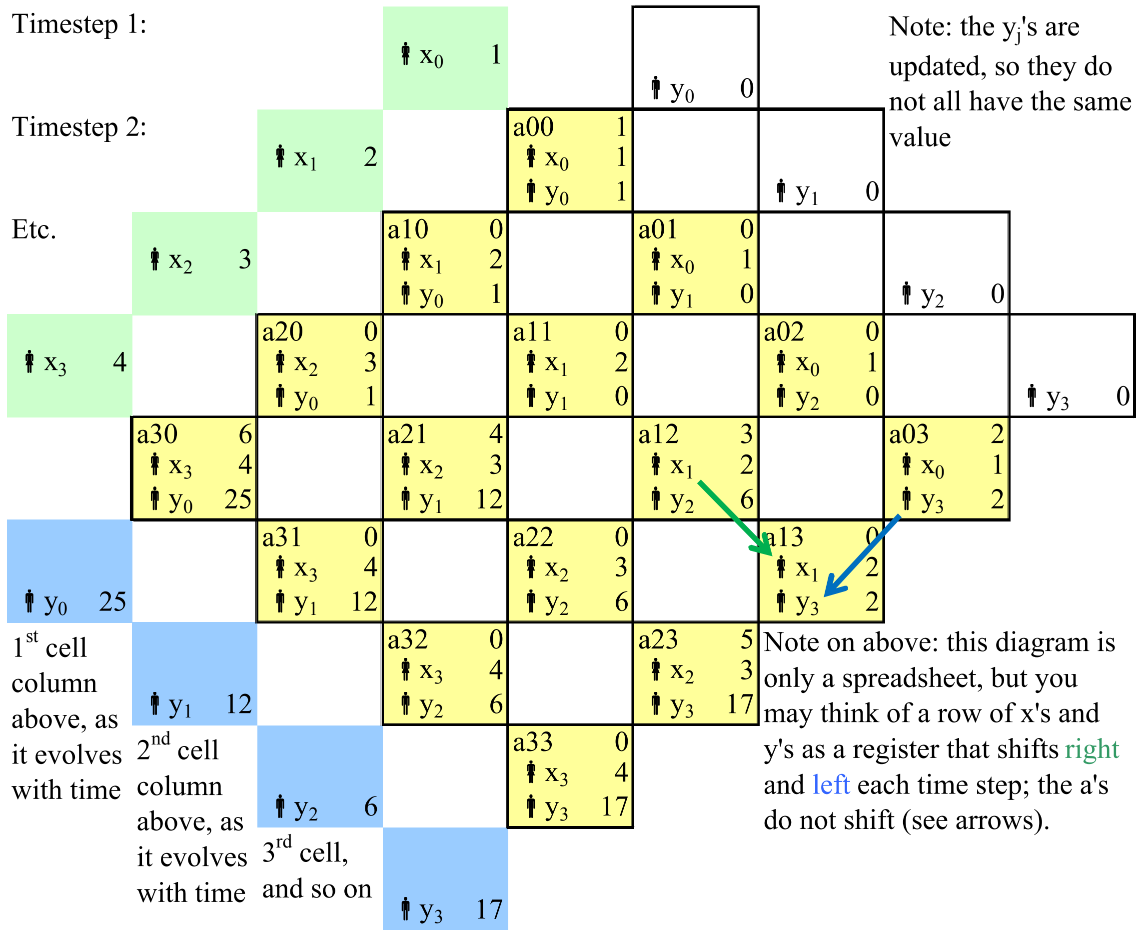
=

y

25	12	6	17

Vector-matrix multiply on left implemented by dataflow-like spreadsheet below.





Note: the y_j's are updated, so they do not all have the same value

Note on above: this diagram is only a spreadsheet, but you may think of a row of x's and y's as a register that shifts right and left each time step; the a's do not shift (see arrows).

Time programming

x

1	2	3	4

A

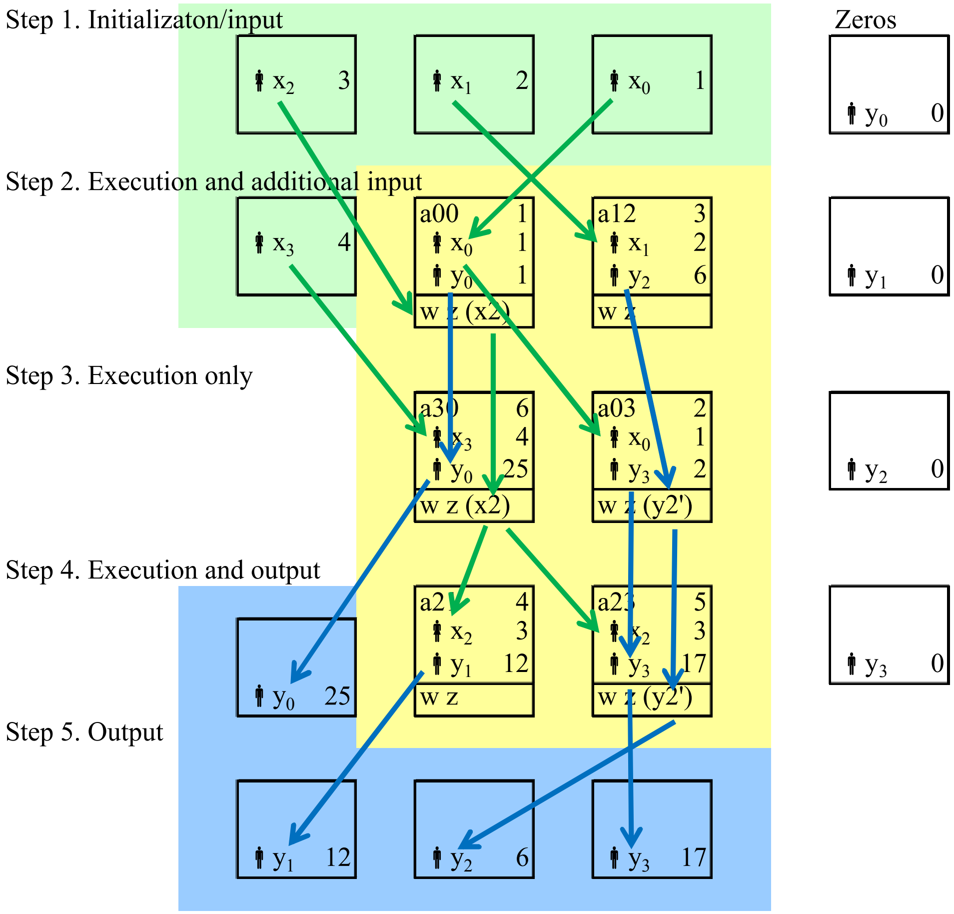
1			2
		3	
	4		5
6			

=

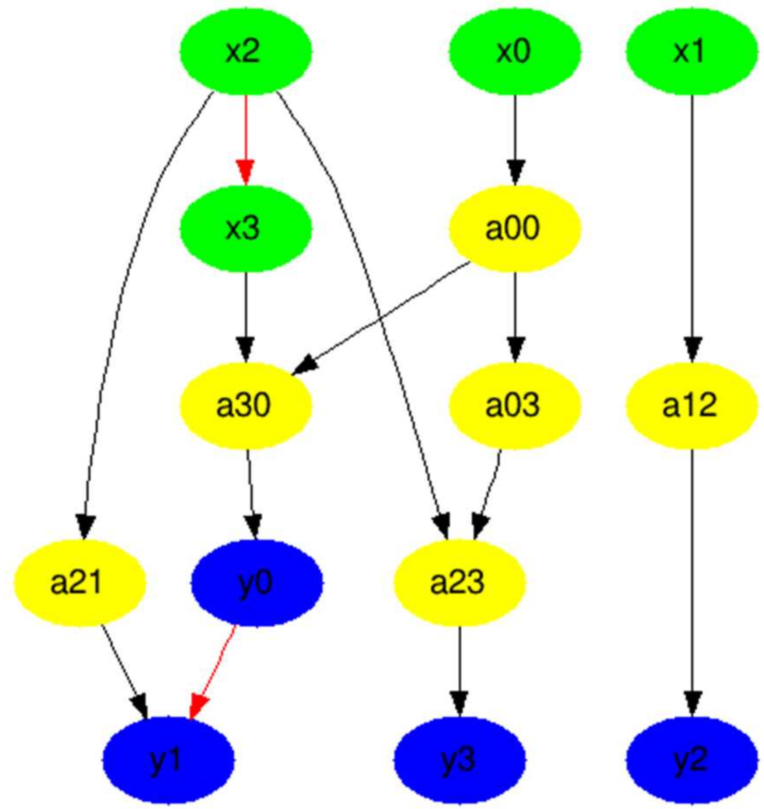
y

25	12	6	17

Arrows indicate data flow; wth no data flow faster than nearest neighbor per step. Sometimes dance steps for ladies and gents.



GraphViz:



Outline

- Formulate 3D scaling rule
- Architecture options
  - ☒ von Neumann
  - ☑ Logic-memory integration
- Programming
- Performance
- Device implications

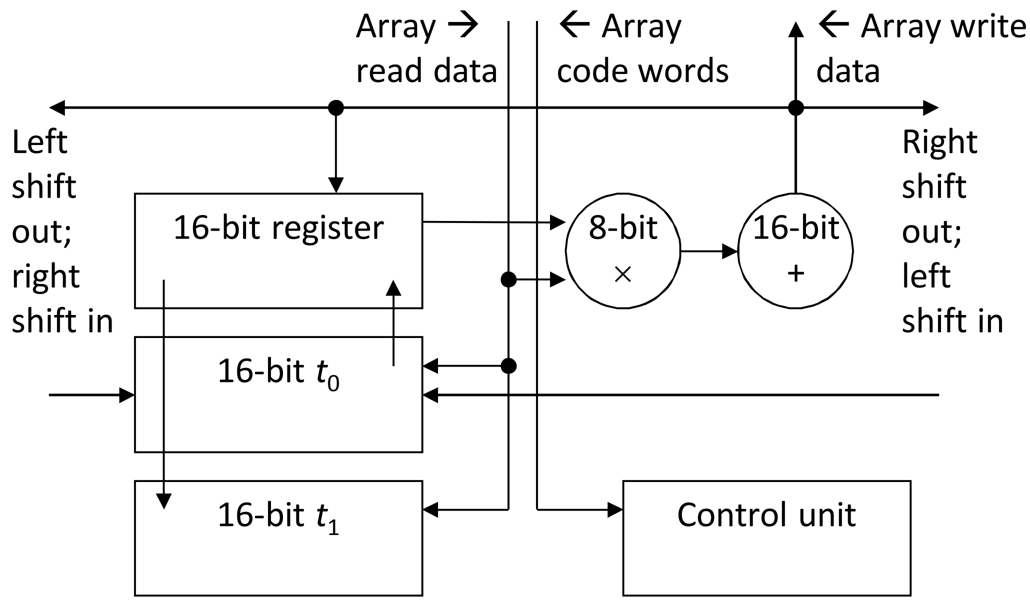
Exemplary ALU

- Note that this is neither a microprocessor nor a GPU

Storage array format:

Synapse value: 8 bits as signed integer, but often interpreted at a higher level as a fixed point number	Green pointer code word	Red pointer code word
12 bits total: 8 bits +	2 bits +	2 bits

ALU (one for each 12 storage bits):



Performance on Deep Learning example

Memory / Logic type	GTX 750 Ti 0.1 nj/bit	DRAM 46.0 fj/bit	Adiabatic Mem 0.9 fj/bit
TFET 1.3 fj/synapse 12 bits needed	1.0 nj 0.0 j 1.0 nj 20.8 mw	552.0 fj 1.3 fj 553.3 fj 11.1 kw	10.9 fj 1.3 fj 12.2 fj 244.3 w
CMOS HP 21.8 fj/synapse 12 bits needed	1.0 nj 0.0 j 1.0 nj 20.8 mw	552.0 fj 21.8 fj 573.7 fj 11.5 kw	10.9 fj 21.8 fj 32.7 fj 653.2 w
TFET 21 bits 7.7 fj/synapse 25 bits needed	2.2 nj 0.0 j 2.2 nj 43.4 mw	1150.0 fj 7.7 fj 1157.6 fj 23.2 kw	22.7 fj 7.7 fj 30.4 fj 607.9 w
CMOS HP 21 bits 127.8 fj/synapse 25 bits needed	2.2 nj 0.0 j 2.2 nj 43.4 mw	1150.0 fj 127.8 fj 1277.7 fj 25.6 kw	22.7 fj 127.8 fj 150.5 fj 3010.2 w

Line 1: Femto joules to access memory for one synapse
Line 2: Femto joules logic energy to act on one synapse
Line 3: Sum of previous two lines
Line 4: System energy (watts, kilowatts, megawatts)

Note: NVIDIA GTX 750 Ti is memory bandwidth limited so the logic energy is ignored.

CMOS HP and TFET per Nikonov and Young's study

First two rows are 8-bit synapse; last two rows are 16-bit synapse

Outline

- Formulate 3D scaling rule
- Architecture options
  - ☒ von Neumann
  - ☑ Logic-memory integration
- Programming
- Performance
- Device implications

Device implications; conclusions

Device implications

- There is nothing wrong with transistor function
- We need to drive down manufacturing cost, which probably requires a new device
  - could be a more manufacturable transistor
  - could be something different, but the difference is not essential
- Logic-memory integration is essential

Conclusions

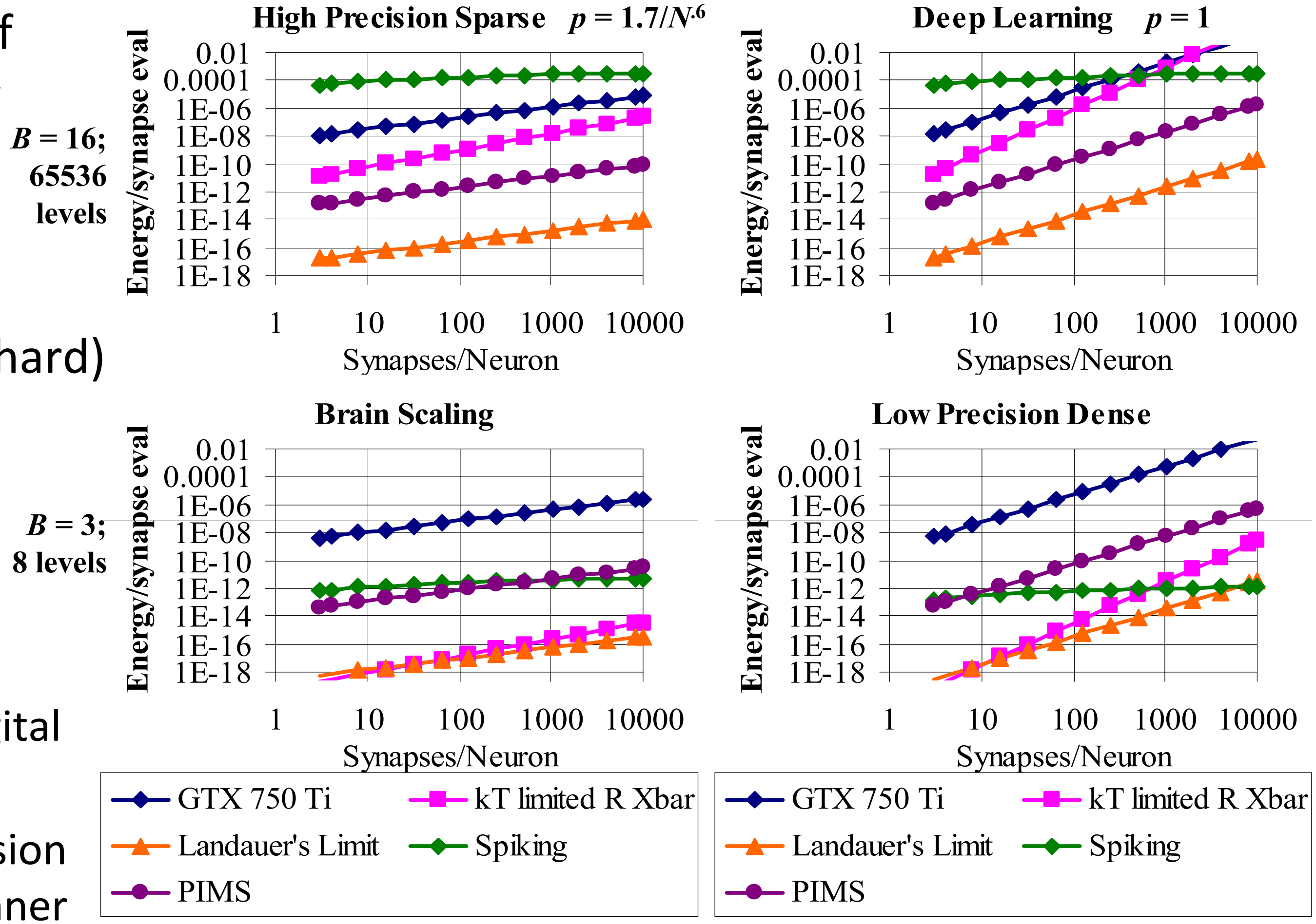
- With logic-memory integration, we could possibly have an exponential improvement path until we end up with a structure with the parameters of a brain (throughput/storage)
  - We don't claim to know how to program a brain

Three neuromorphic options?

- Crossbar with a boost from level-based analog (memristor)
- Spiking with a boost from time-based analog
- Digital emulation of neurons with a boost from adiabatic digital tricks and 3D integration

Expected comparison result

- We did a study of energy efficiency of neuromorphic approaches
- Not ready for publication (too hard)
- Conclusions
  - Physical limits of computation apply to both analog and digital
  - Scale, coding, sparsity, precision determine winner

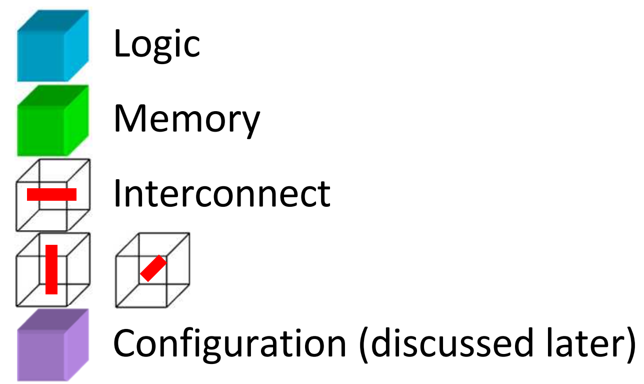


Backup

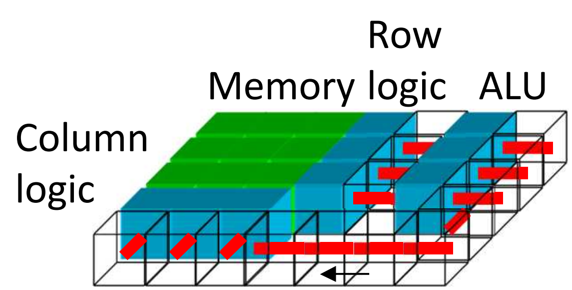
Architecture versus design rules

- Answerable to whom?
  - Architecture is a human choice
  - Design rules are answerable to nature
- Example: rotate instruction
  - A chip designer cannot just wire anything to anything because a customer wants him/her to do so
  - Nature will not approve of long-distance communications at constant time and energy
  - Chip designer has to follow design rules; can't change them without approval from nature

- PIMS tiles (building blocks)

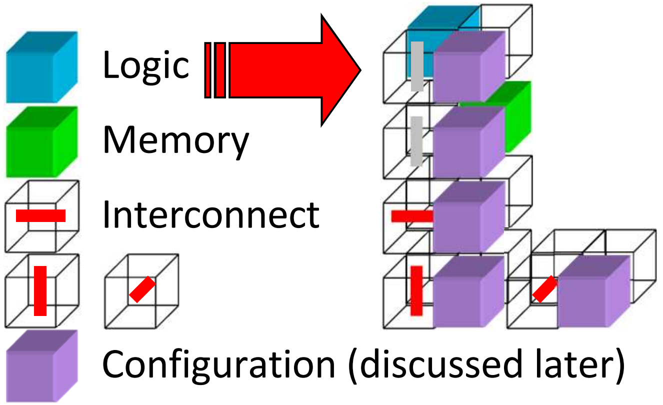


- PIMS program

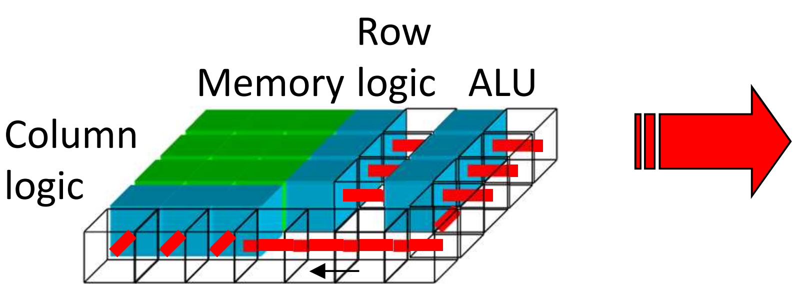


Programmable

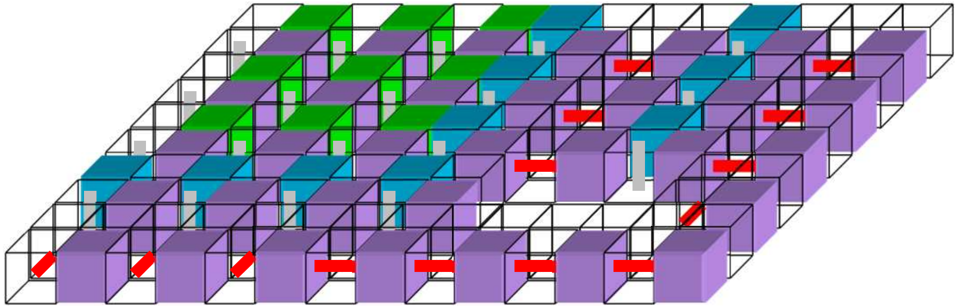
- Let's make a machine that can emulate ANY arrangement of PIMS tiles
  - Use tile clusters that can be configured to create any of the three tiles
  - Load the desired tile configuration as though it were software
- Programmable cluster blocks



- Previous system

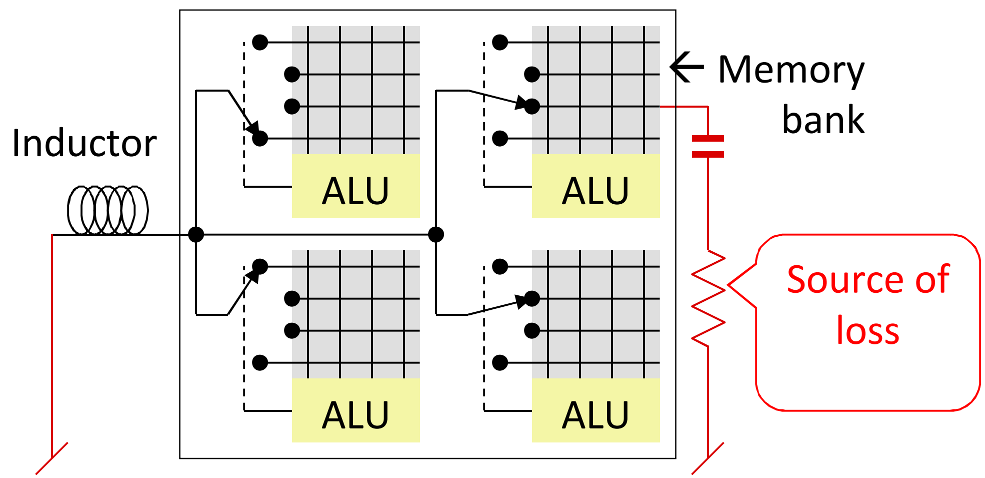


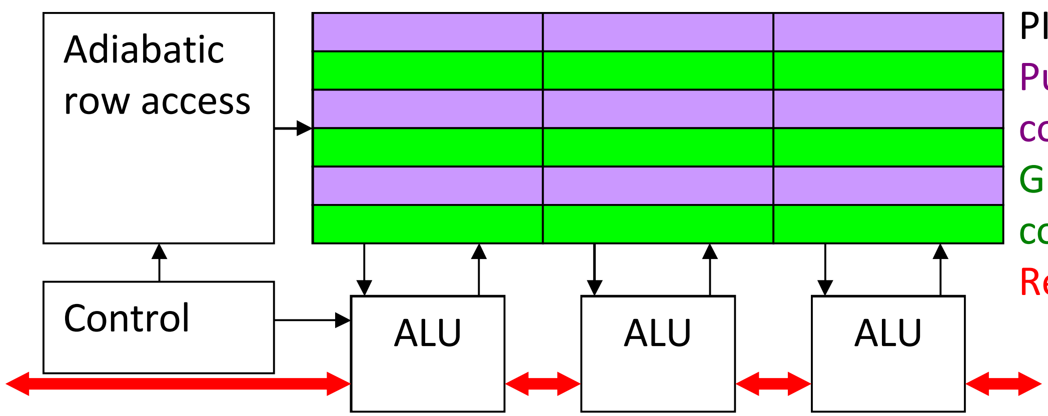
- Programmed equivalent



PIMS engine

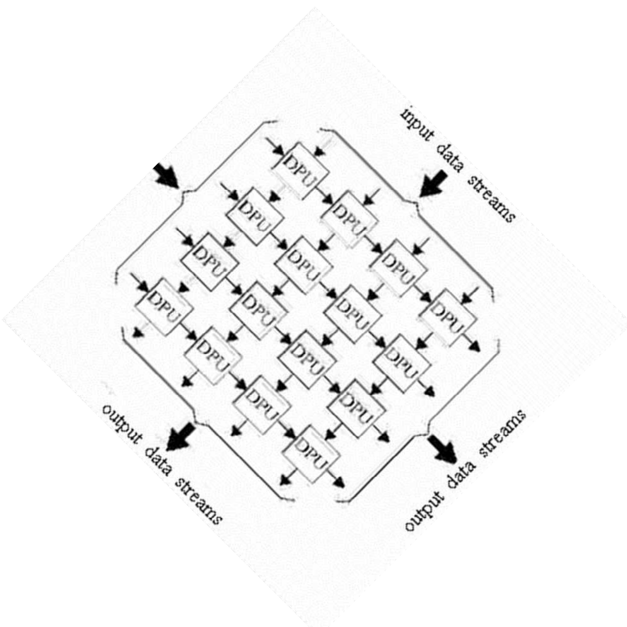
- PIMS is a hardware device that can
  - Execute the tile structures
  - Emulate the nanotechnology
- Adiabatic memory structure →
- Adaptation of tiles for efficient execution with time multiplexing to allow bigger machines ↓





PIMS memory/storage:
Purple is the configuration "opcode"
Green is memory contents
Red is communications between ALUs

Backup (embedded spreadsheet)



x

1	2	3	4

A

1	0	0	2
0	0	3	0
0	4	0	5
6	0	0	0

=

y

25	12	6	17

Vector-matrix multiply on left implemented by dataflow-like spreadsheet below.

Timestep 1:			x_0 1		y_0 0			
Timestep 2:		x_1 2		a00 1 x_0 1 y_0 1		y_1 0		
Etc.	x_2 3		a10 0 x_1 2 y_0 1		a01 0 x_0 1 y_1 0		y_2 0	
x_3 4		a20 0 x_2 3 y_0 1		a11 0 x_1 2 y_1 0		a02 0 x_0 1 y_2 0		y_3 0
	a30 6 x_3 4 y_0 25		a21 4 x_2 3 y_1 12		a12 3 x_1 2 y_2 6		a03 2 x_0 1 y_3 2	
y_0 25		a31 0 x_3 4 y_1 12		a22 0 x_2 3 y_2 6		a13 0 x_1 2 y_3 2		
1st cell column above, as it evolves with time	y_1 12		a32 0 x_3 4 y_2 6		a23 5 x_2 3 y_3 17			
	2nd cell column above, as it evolves with time	y_2 6		a33 0 x_3 4 y_3 17				
		3rd cell, and so on	y_3 17					

Note: the y_j's are updated, so they do not all have the same value

Note on above: this diagram is only a spreadsheet, but you may think of a row of x's and y's as a register that shifts right and left each time step; the a's do not shift (see arrows).

Backup (embedded spreadsheet)

x

1	2	3	4

A

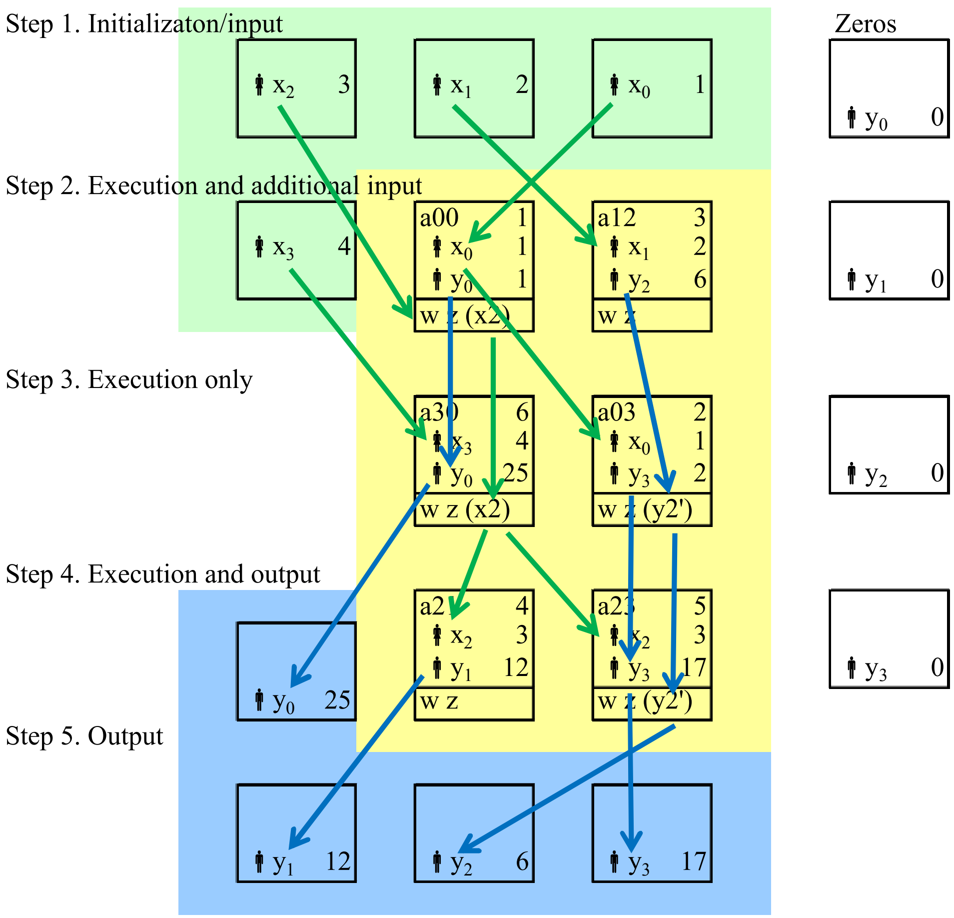
1			2
		3	
	4		5
6			

=

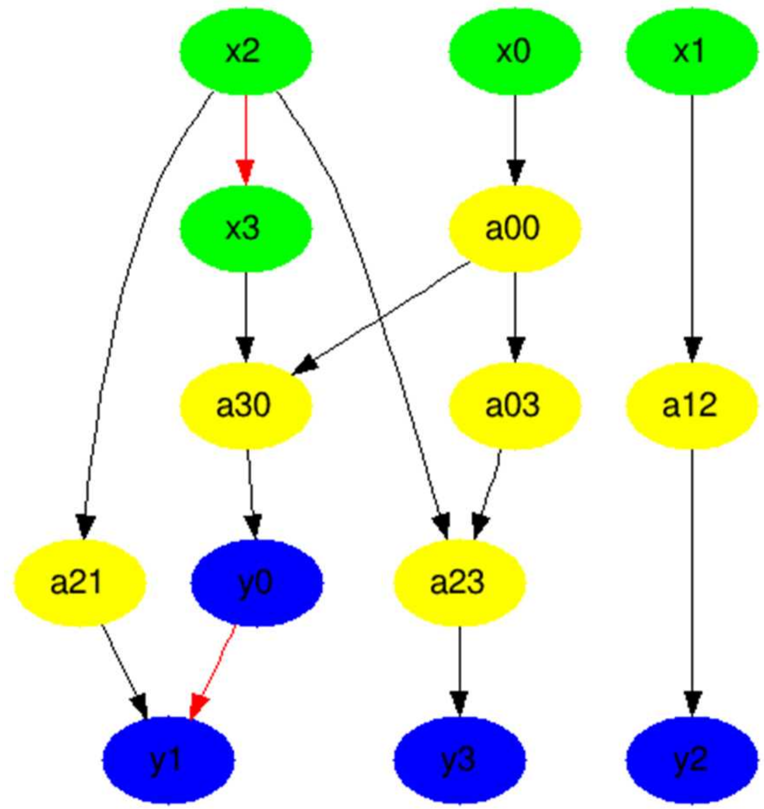
y

25	12	6	17

Arrows indicate data flow; wth no data flow faster than nearest neighbor per step. Sometimes dance steps for ladies and gents.



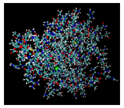
GraphViz:



PIMS algorithm scaling

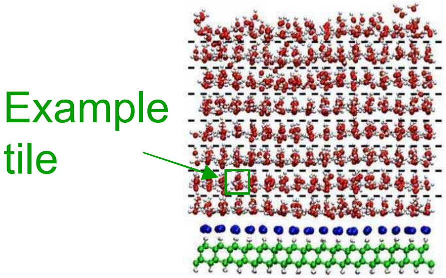
Factor 2-bit composite → Factor 1024-bit composite

N_2 tiles



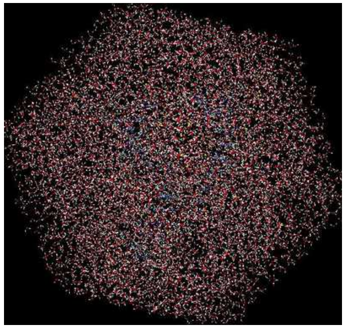
From: http://www.ucd.ie/nanotech/

Speculation: Nature would use a component hierarchy. Unproven, but almost always happens.



http://www.nanowerk.com/spotlight/spotid=2617.php

N_{1024} tiles



From: http://theory.rutgers.edu/Group/Research/Galleries/BiochemicalReaction/index.html

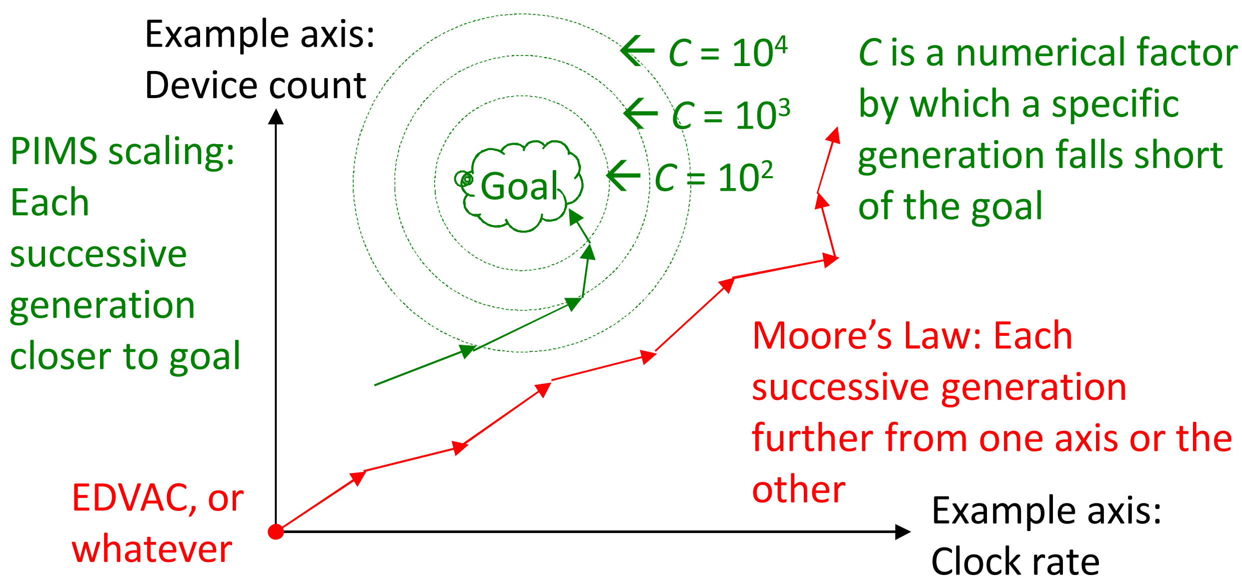
Sequence N_2, N_3, N_4…N_{1024}… becomes the physical/computational complexity for factoring an N-bit number in the physical universe

However, based on assumptions like room temp operation and a certain repertoire of chemical elements

New concept in computing

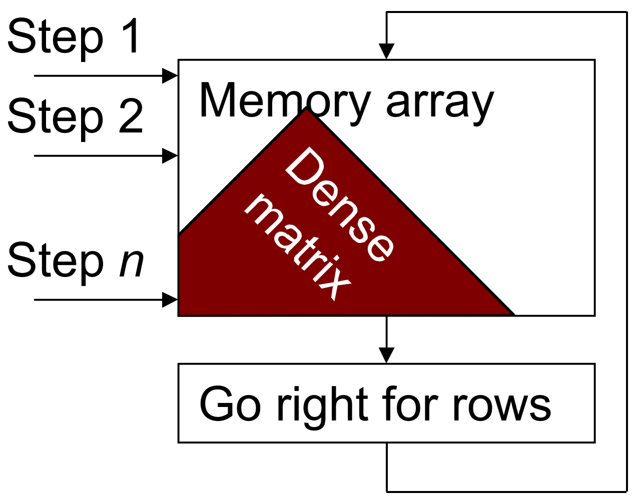
- New scaling concept
  - Head to a goal, instead of
  - measure progress since WW II
- As a computer technology, Moore's Law is based on generations of progress from and implicit starting point that was something like von Neumann's EDVAC computer
- Goal wavers

- PIMS concept is to measure distance from ultimate nanotechnology
  - Let *C* be the factor by which a current implementation is WORSE than the ultimate nanotechnology goal



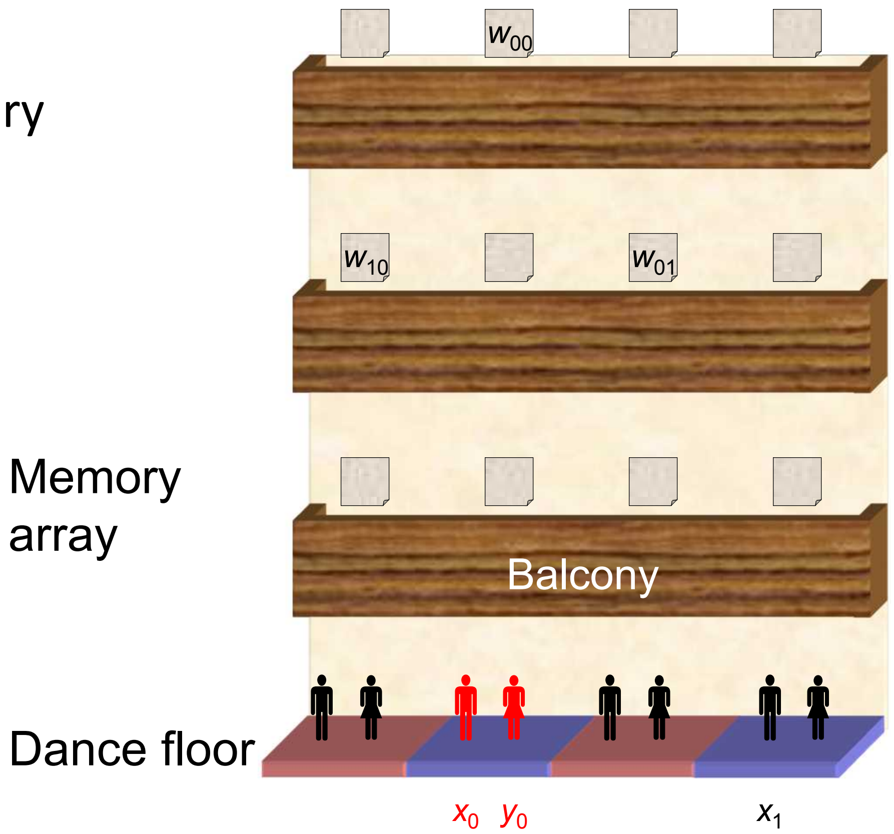
Backup: Programming a dense vector-matrix multiply

- Init: Ladies have vector element; gents have zero accumulation
- Program: Ladies multiply memory output by their vector element, pass to gent; gent adds to accumulating sum; ladies step right; gents step left



Wx = y; gent w_{00} x_0 then w_{10} x_0; lady $y_0 = w_{00}\, x_0 + w_{01}\, x_1$

- Dance hall model



Note: This program only uses half the memory locations; better algorithm would use a hexagonal layout, but is too complex for PowerPoint