Computation: Digital and Analog

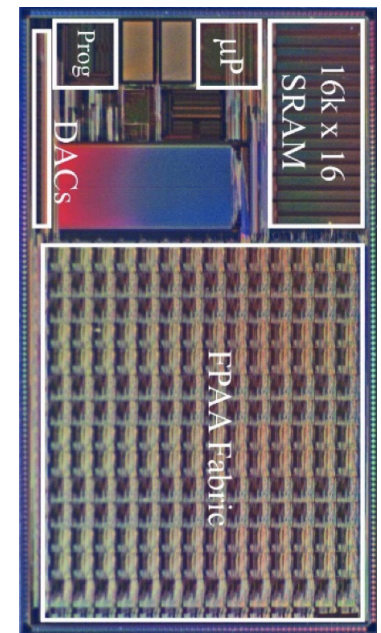## Digital
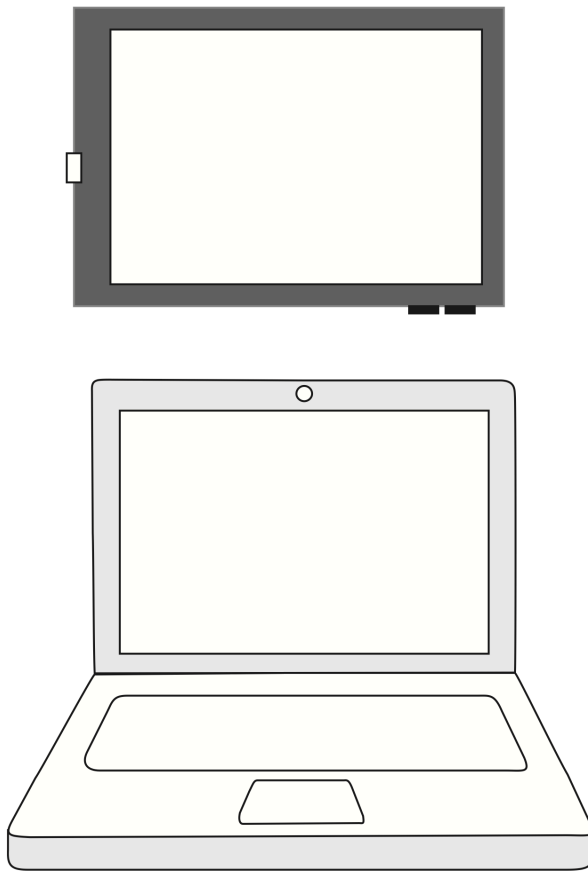
- Programmable
- Configurable

Analog

Soldering Iron
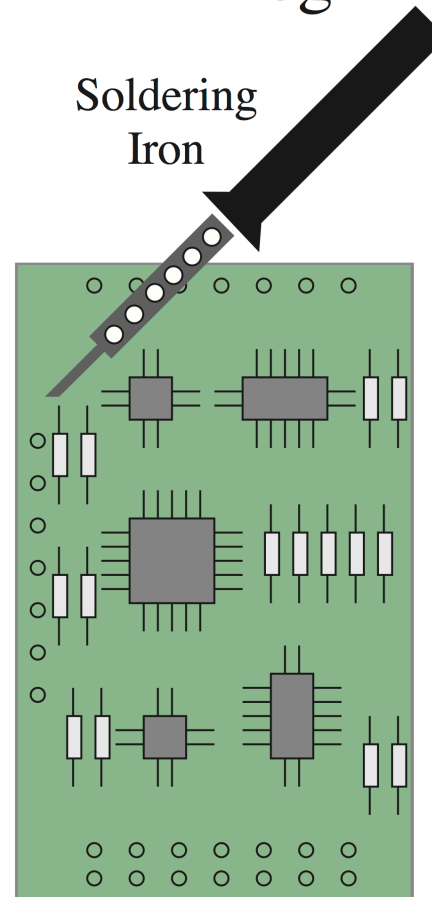
- Fixed Function
- Custom Design

Large-Scale FPAA

Prog
μP
DACs
16k x 16 SRAM
FPAA Fabric

SoC FPAA

- Analog & Digital
- Programmable
- Configurable
- Design tools

Why Physical / Analog Computing?

Digital

Multiplication (digital: 16bit)

20 transistors

Analog

I_{out}

	Digital	Analog
Energy/operation	x1000	x1
Size	x100	x1
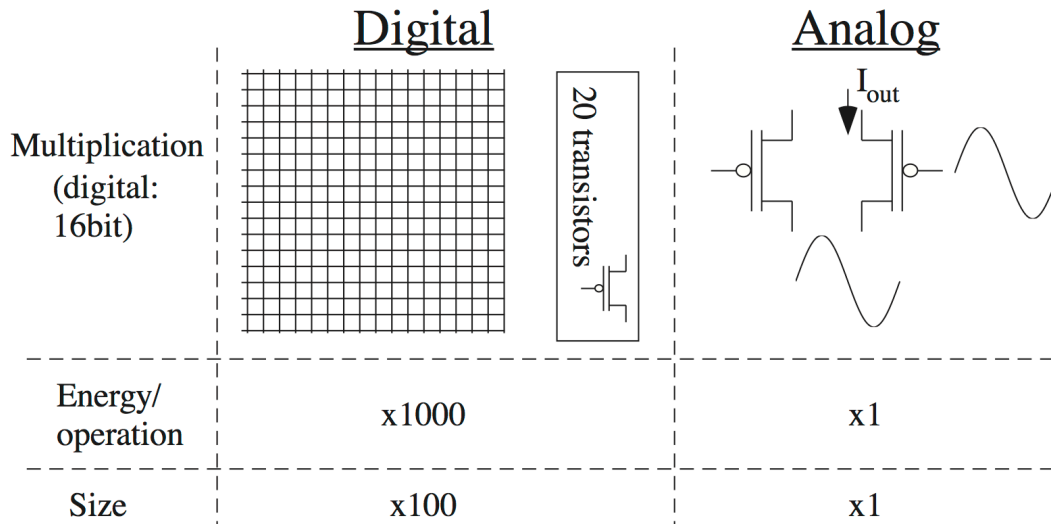
Expansion of C. Mead's original hypothesis, 1990

Efficiency Wall

GMAC(/s)/W (32-bit MAC)

- IBM Cell
- ARM Cortex-A
- core i7 980 ee
- Curve Fit
- Monarch
- Tile64 Virtex6
- nVidia Fermi GTX480
- tms320c6412-500
- Atom N270
- Pentium 4 ee HT 3.2
- Core 2 QX670o
- tms320c6424-700
- Koomey's Law
- Athlon X2 BE2300
- Pentium Dual-Core E2140

Power Efficiency wall (Production ICs) 1 MAC in 100pJ (10MMAC/mW)

130 nm → 90 nm → 65 nm → 45 nm → 32 nm → ?? nm   [Marr, et. al, 2013]

2000 2002 2004 2006 2008 2010 2012 2014 2016

Year

Power Efficiency Scaling

10MMAC(/s)/W	1st DSPs (1978 - 1981)
100MMAC(/s)/W	[Hasler and Marr 2013]
1MMAC(/s)/mW	
10MMAC(/s)/mW	Energy Efficiency Wall (32bit inputs)
100MMAC(/s)/mW	
1MMAC(/s)/uW	
10MMAC(/s)/uW	Typical Analog VMM
100MMAC(/s)/uW	FPAA compiled Dendritic Classifier (350nm IC)
1MMAC(/s)/nW	
10MMAC(/s)/nW	Dendritic Classifier (350nm IC)
100MMAC(/s)/nW	Shrink by scaling L, V_{dd}
1MMAC(/s)/pW	(<) Biological Neuron
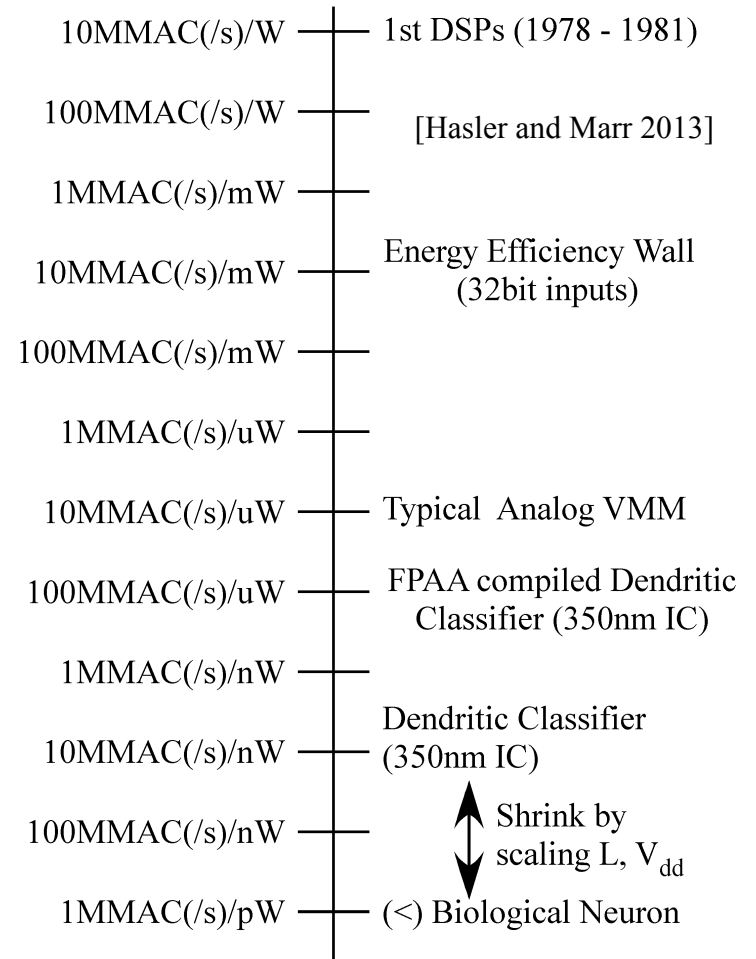
Framework for Physical (Analog) Computing

Digital Computation
Digital Turing Machine

Computer $\rightarrow$ $\mathbf{Z}$ $(\aleph_0)$

Algorithm Order
sort: $O(n \log_2 n)$
NP: Traveling Salesman

Sync Digital Blocks

μP $\rightarrow$ ROM / RAM

Numerics
$\overrightarrow{\mathbf{A}\vec{x}} = \vec{b}$
$(\overline{\mathbf{LU}})\,\vec{x} = \vec{b}$
$\vec{x} = \overline{\mathbf{A}}^{-1}\,\vec{b}$
$x^5 - 5x + 1 = 0$

Classical Analog Computation
Analog Computing Model

? ? ? ?

A miracle occurs

Config Analog Computation
Analog Turing Machine

Computer $\rightarrow$ $\mathbf{R}$ $(\aleph_1)$

CT Analog Blocks
Sub band
VMM+ WTA
HMM

Algorithm
Analog Algorithm Theory

Numerics
$\dfrac{d\vec{V}}{dt} = f(\overline{\mathbf{W}}, \overline{\mathbf{M}}, \vec{V})$

Digital Computation + Tools

SoC FPAA IC