Small is beautiful


2. *Design of ion-implanted MOSFET's with very small physical dimensions*, R.H. Dennard et al, 1974 – *The right way to get there.*


- S.N. Hemanth Meenakshisundaram
Historical Perspective

1959
- EDVAC, ENIAC
- DEC PDP, CDC 6000
- IBM System/360

1964
- IBM /370
- 4004
- IC invention
  - Jack Kilby (TI)
  - Robert Noyce

Moore’s Law
Costs & Curves

Too few components - No economies of scale
Too many – Bad Yield (Unreliable)
Costs & Curves

Too few components - No economies of scale
Too many – Bad Yield (Unreliable)

Optimal

Curve moves right & down

Cost

Relative Manufacturing Cost per Component

Number of Components per Integrated Circuit

No. of components
And thus we have:

Joining the optimal points from earlier.
Justification

Yield/Reliability problems

- Purely engineering problem, no new science required.

Heat/Power Problems

- 2D surface allows dissipation.

- Power mainly for lines, capacitances which shrink (\( C = \epsilon A/d \) )

- Speed also increases (for same power per unit area)
Justification

Yield/Reliability problems
- Purely engineering problem, no new science required.

Heat/Power Problems
- 2D surface allows dissipation.

- Power mainly for lines, capacitances which shrink (C = \(\varepsilon A/d\))

- Speed also increases (for same power per unit area)

Enter Dennard:
1. There’s only 1 right way to shrink this. How?
Bunch of other stuff Moore said…

1. Package small functions in separate chips, system builders will put stuff together – Didn’t really happen this way (4004, SoCs)
2. Linear systems will still use discrete components – True
3. Good idea to make Op-amp IC – True
4. Microwave integrated circuits - True
Problem with MOS scaling

Scaling causes $V_{th}$ to decrease. So noise can trigger switch.
The right way to scale MOS
The right way to scale MOS

1. Reduce widths too
The right way to scale MOS

1. Reduce widths too
2. Reduce voltage across channel
The right way to scale MOS

1. Reduce widths too
2. Reduce voltage across channel (and Vt)
3. Increase doping levels
Table 1
Scaling Results for Circuit Performance

<table>
<thead>
<tr>
<th>Device or Circuit Parameter</th>
<th>Scaling Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device dimension $t_{ox}$, $L$, $W$</td>
<td>$1/\kappa$</td>
</tr>
<tr>
<td>Doping concentration $N_a$</td>
<td>$\kappa$</td>
</tr>
<tr>
<td>Voltage $V$</td>
<td>$1/\kappa$</td>
</tr>
<tr>
<td>Current $I$</td>
<td>$1/\kappa$</td>
</tr>
<tr>
<td>Capacitance $\epsilon A/l$</td>
<td>$1/\kappa$</td>
</tr>
<tr>
<td>Delay time/circuit $VC/I$</td>
<td>$1/\kappa$</td>
</tr>
<tr>
<td>Power dissipation/circuit $VI$</td>
<td>$1/\kappa^2$</td>
</tr>
<tr>
<td>Power density $VI/A$</td>
<td>1</td>
</tr>
</tbody>
</table>

$P = CV^2f$

Delay depends on C
The 18 month adjusted Moore’s Law
Problems Today

1. Sub-threshold leakage no longer insignificant.
2. The width is down to 4 layers of Si atoms. We are running out of atoms.
3. High doping causes another type of leakage.
Dennard said wire delays were constant. Not an issue.

**Reality:**

- The constant delay can no longer be ignored today.
- Actually ‘constant’ was an approximation no longer true.
What do we do?

About Wires
- Thicker Wires
- Use Copper not Al
- Low k dielectrics

About Transistors
- New devices.
- New materials
The Microcomputer Revolution – Historical Context

The computer world was divided into 2 camps in the 1960s:

Mainframes: IBM (System/360) and the seven dwarfs (later BUNCH)

• Big Iron. Large Businesses, Universities, Transaction Processing etc.

• Supercomputers – elite subset. MFLOPS vs MIPS

• CDC had this market almost all to itself. (Until Cray started Cray)
The Microcomputer Revolution – Historical Context

Minicomputers: Segment invented by DEC

• Programmed Data Processors *(Gordon Bell lecture)*
• Timesharing OS, in businesses, schools, *Spacewar!*
• PDP-7: First UNIX (in assembly)
• PDP-8: Commercial success. (De Castro)
• Nova series (De Castro, now Data General)
• LINC series with MIT – Minicomputer name *(Clark & Molnar)*
The Microcomputer Revolution

- Intel in 1969 – A memory chip maker.
- Busicom (Japan) wanted chips for a calculator.
- 8 chips for arithmetic, display, keyboard etc.
- Intel had only 2 chip designers - Ted Hoff, Frederico Faggin (Stanley Mazor?)

What to do?
And One Chip to do them all

Solution Hoff proposed:
1. Design one chip (General purpose stripped down CPU)
2. Use Intel’s memory chip expertise.
3. Store programs for 8 chips’ functions in memory.
4. Use the single CPU to do all the work.

Result:
The MCS-4 in 1971.
4001 – ROM. 4002 – RAM, 4003 – Shift Register (For Peripherals) & 4004 - CPU.
Chip Design – Hoff, Mazor, Faggin.
MOS Logic Design – Faggin
Firmware – Shima (Busicom)
Announcing a new era of integrated electronics

A micro-programmable computer on a chip!

Intel introduces an integrated CPU complete with a 4-bit parallel adder, sixteen 4-bit registers, an accumulator and a push-down stack on one chip. It's one of a family of four new ICs which comprise the MCS-4 microcomputer system—the first system to bring you the power and flexibility of a dedicated general-purpose computer at low cost in as few as two dual inline packages.

MCS-4 systems provide complete computing and control functions for test systems, data terminals, billing machines, measuring systems, numeric control systems and process control systems.

The heart of any MCS-4 system is a Type 4004 CPU, which includes a powerful set of 45 instructions. Adding one or more Type 4001 ROMs for program storage and data tables gives you a fully functioning micro-programmed computer. To this you may add Type 4002 RRAMs for read-write memory and Type 4003 registers to expand the output ports.

Using no circuitry other than ICs from this family of four, you can create a system with 6096 8-bit bytes of ROM storage and 5120 bits of RAM storage. When you require rapid turn-around or need only a few systems, Intel's erasable and re-programmable ROM, Type 1701, may be substituted for the Type 4001 mask-programmed ROM.

MCS-4 systems interface easily with switches, keyboards, displays, teletypewriters, printers, readers, A-D converters and other popular peripherals.

The MCS-4 family is now in stock at Intel's Santa Clara headquarters and at our marketing headquarters in Europe and Japan. In the U.S., contact your local Intel representative for technical information and literature. In Europe, contact Intel at Avenue Louise 216 B 1050 Brussels, Belgium, Phone 043 300 03. In Japan, contact Intel Japan, Inc., Parkside Flat Bldg. No. 4-2-2 Sendaigaya, Shibuya-Ku, Tokyo 151. Phone 03-403-4747.

Intel Corporation now produces micro computers, memory devices and memory systems at 3066 Bowers Avenue, Santa Clara, Calif. 95051. Phone (408) 248-7001.

intel delivers.
The 4001 – ROM chip (256 bytes)

- 8 bits per instruction.
- 4 bit I/O port.
  Hence 5 cycles to transfer 1 instruction byte.
- Up to 16 can be part of MCS-4. (12 bit PC)
- Unique mask burned in to identify the ROM chip.
4002 – RAM Chip (40 bytes)

• 16 digits (fraction) + 2 digits (exponent) + 2 digits (+/-, control) = 20 BCD digits
• 20 * 4 bits per BCD digit * 4 numbers stored = 40 bytes
• DRAM chip! (The ROM was SRAM technology)
• Same pins as 4001.

4003 – Shift Register for peripherals

• Receive BCD digits in 4 bit words from CPU.
• Output serially to printer etc.
The 4004
The 4004

Intel 4004 Architecture

- Accumulator
- Temp. Register
- Instruction Register
- Instruction Decoder and Machine Cycle Encoding
- Stack Pointer
- Program Counter
- Address Stack
- Scratch Pad
- Register Multiplexer
- Index Register Select
- Timing and Control
- ROM Control
- RAM Control
- Test
- Sync
- Phases
- CM ROM
- CM RAM 0-3
- Test
- Sync Ph 1
- Ph 2
- Reset

80 µs/digit
The 4004

Intel 4004 Architecture

- Accumulator
- Temp. Register
- Instruction Register
- Instruction Decoder and Machine Cycle Encoding
- ALU
- Flag Flip Flops
- Decimal Adjust

For subroutines

80 µs/digit
The 4004

Octal 80 µs/digit

For subroutines

On Chip DRAM
Key Ideas

1. Program & Data Memory separate – this is not Von Neumann.

2. Precursor of the 8051 µcontroller, not the µprocessors.

3. Equivalent to a 1960s IBM computer, but on a single chip! ~2200 trx

4. All I/O functions through firmware on ROMs, single computing chip.

5. Minimize pins, not logic! Intel had only 16-pin package at the time.

6. Family of chips meant to be used together.
Enter the real microcomputer - 8008

• Initially designed for CTC/Datapoint.

• 8 bit ALU, registers, data bus etc.

• Worked with any memory chip not just Intel 4001, 4002.

• Differing speeds meant sync/ready pins.

• Address Registers, Multiplexors, I/O Latches added 40 chips.

• 18-pin package from 1103 DRAM plant.

• 14-bit PC, use of EEPROMs.

• Single address space for memory & data. Indirect addressing only (HL)

• ‘Unfortunate’ Little Endian format for JUMP.
Logic Design : Hal Feeney
ISA : Pyle, Poor from CTC
(48 instructions)
The great leap forward

Intel 8080 Architecture

D0-D7 bidirectional Data Bus

Data Bus Buffer/Latch

8 Bit internal Data Bus

Instruction Register

Instruction Decoder and Machine Cycle Encoding

Multilexer

Register Select

Stack Pointer

Program Counter

Incrementer/Decrementer Address Latch

Address Buffer

Timing and Control

WRITE

Data Bus Control

Interrupt Control

Hold Control

Wait Control

Sync

Clocks

A0-A15 Address Bus

#WR

DBIN

INT

INTE

Hold Ack

Wait

Ready

Sync Ph1

Ph2

Reset

Accumulator

Temp. Register

Accumulator Latch

Flag Flip Flops

ALU

Decimal Adjust
The great leap forward

- 1972: 8080 project starts.
- NMOS technology – smaller, 2x faster.
- ~4500 trx, redesign gave 10x performance ↑
- Stack offloaded to memory, unlimited stack depth. Stack grows ↓
- Pairs of 8-bit registers for some 16-bit operations.
- Indirect & Direct Memory Addressing.
- XHLD, XTHL – No more (244 out of 256 instructions)
- Many clones, competitors. Z-80 (Faggin and Shima), Motorola.
The Altair gets its own slide

- Altair 8800 – Ed Roberts, 1975, Mail order ads
- CP/M – Gary Kildall
- Altair-Basic from Micro-Soft
- IMSAI with IMDOS, other clones, 8800B.
1977 : 8085
- Obsolete on release
- ~6500 trx,
- 2/12 more instructions.
- Only 5V supply
- Multiplexed data & lower order address
- More interrupts
In the meantime…

- PDP-11: 16-bit architecture, byte addressable, ‘Unibus’
- Later LSI-11, Q-Bus,
  - 4 chip set from Western Digital
- C programming language,
  - UNIX rewritten.
- 32-bit VAX and VAX/VMS OS.
- Influenced all x86, 68000 and many others.
The x86 Family

8086 – 1978

• 30k Transistors
• 16-bit operations
• Multiplication and Division
• 20-bit addresses. Segment address + Offset

8088 – 8086 with 8-bit data bus for compatibility.

• Chosen by IBM for their word processor & microcomputer (along with DOS)
• Almost lost out to Motorola 68000 as the IBM choice.
• Started off the Wintel domination.
The Motorola 68000 (1979 – Present)

• Hybrid 16/32 bit architecture. Twice the 8086 trx
• Fully 32-bit starting 86020.
• Sun, Commodore Amiga, Atari, Sinclair, Apple Lisa, Macintosh
• Branched out to embedded processors.
• Freescale DragonBall, now mostly replaced by ARM.
Conclusion

Observe how the time between processor generations shrank.

And number of transistors grew.

The 8080 (and Intel) also started the trend of chipmakers and computer makers being separate entities.