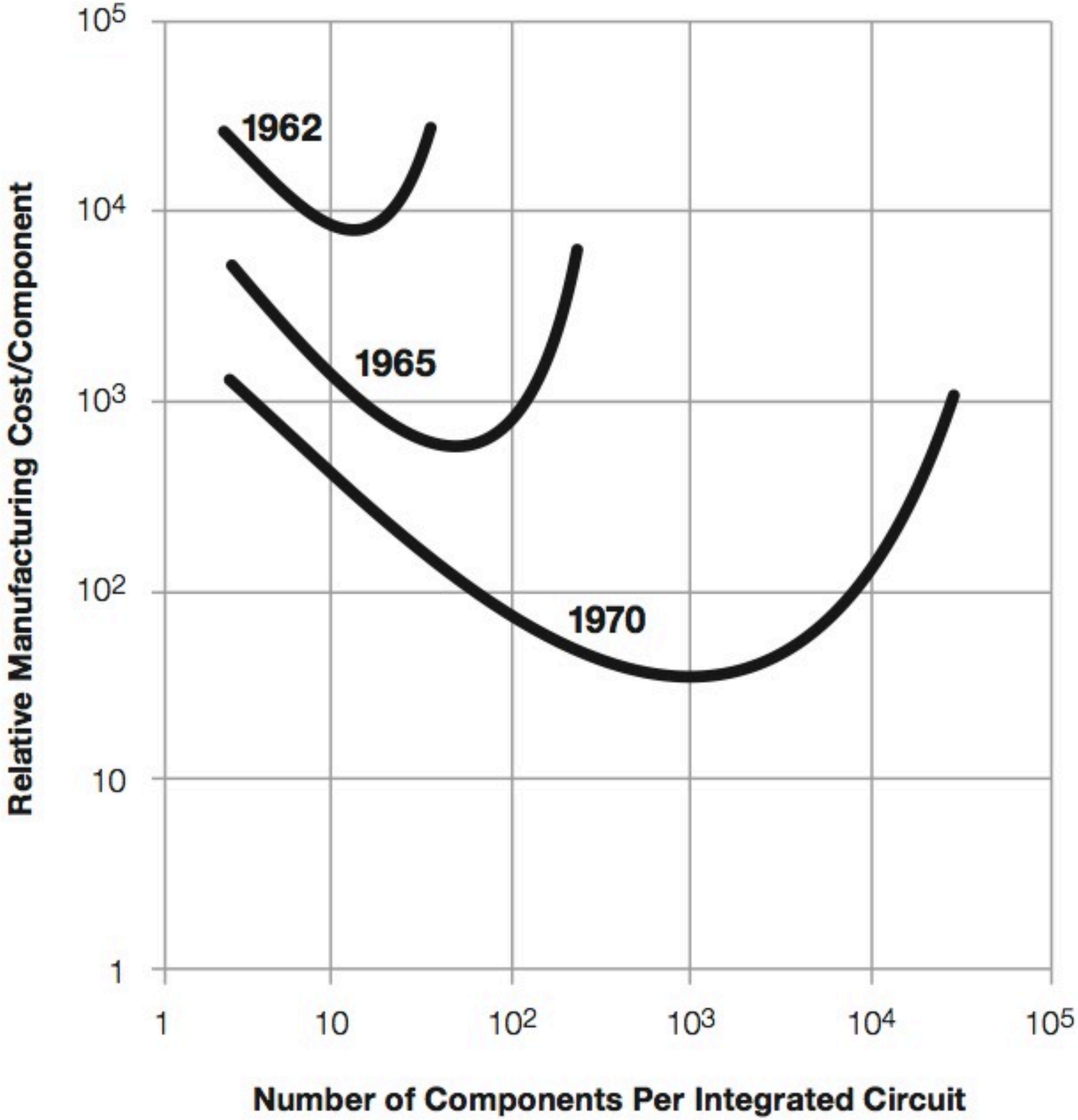
548

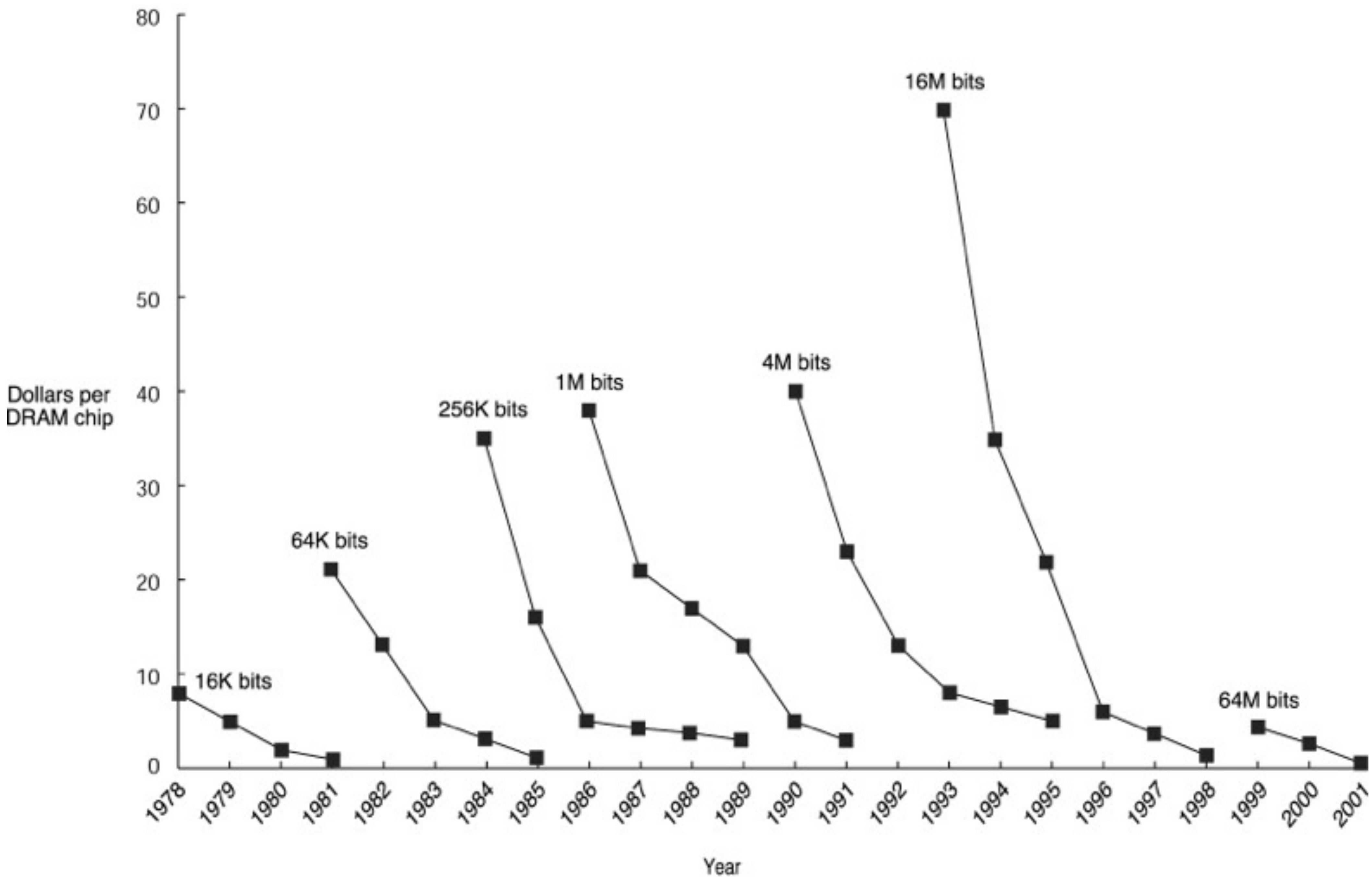
Lecture 2 - Technology

What is Moore's Law?

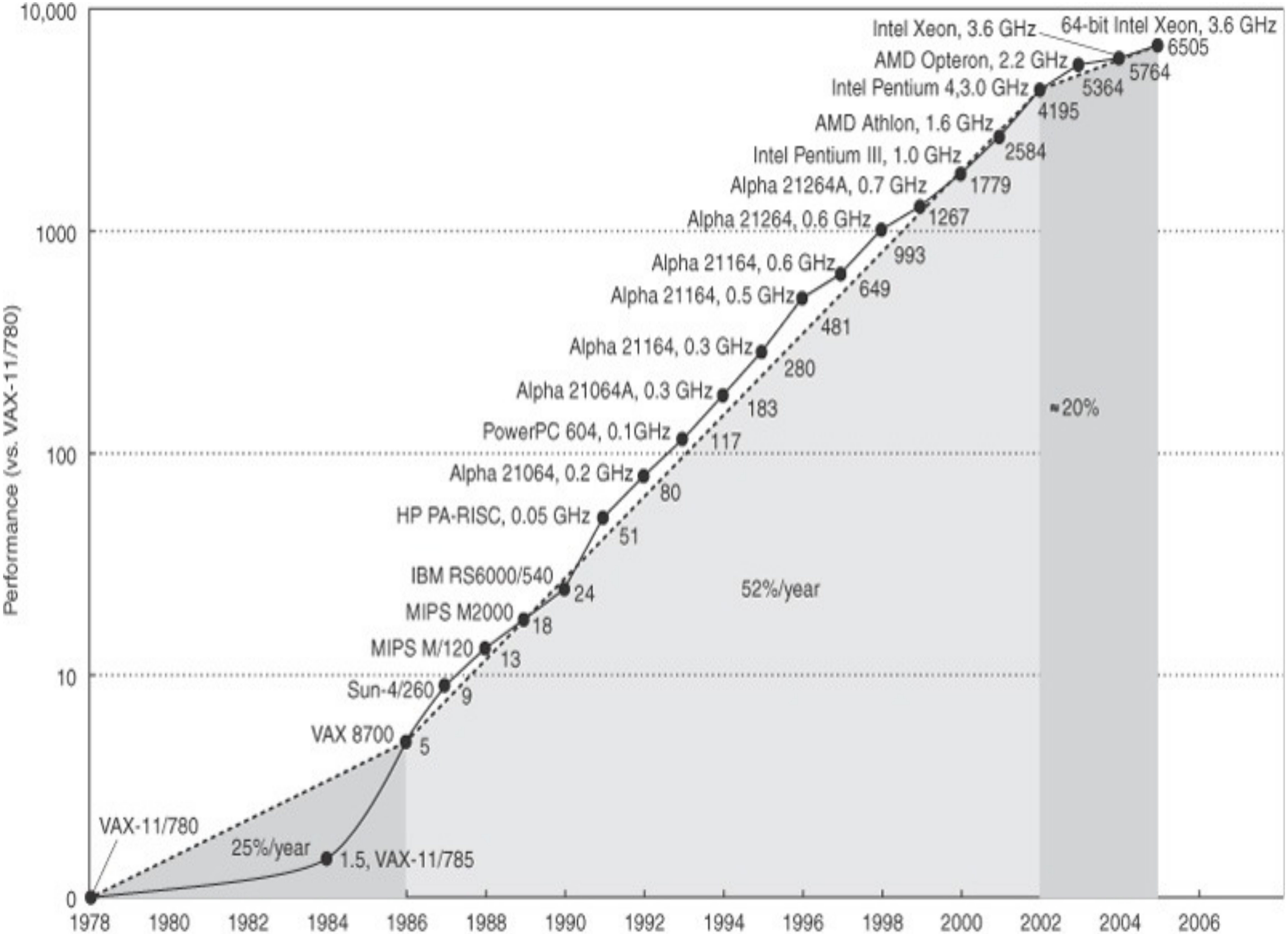
- The number of components that achieves minimum cost on a die doubles about every 18 months.



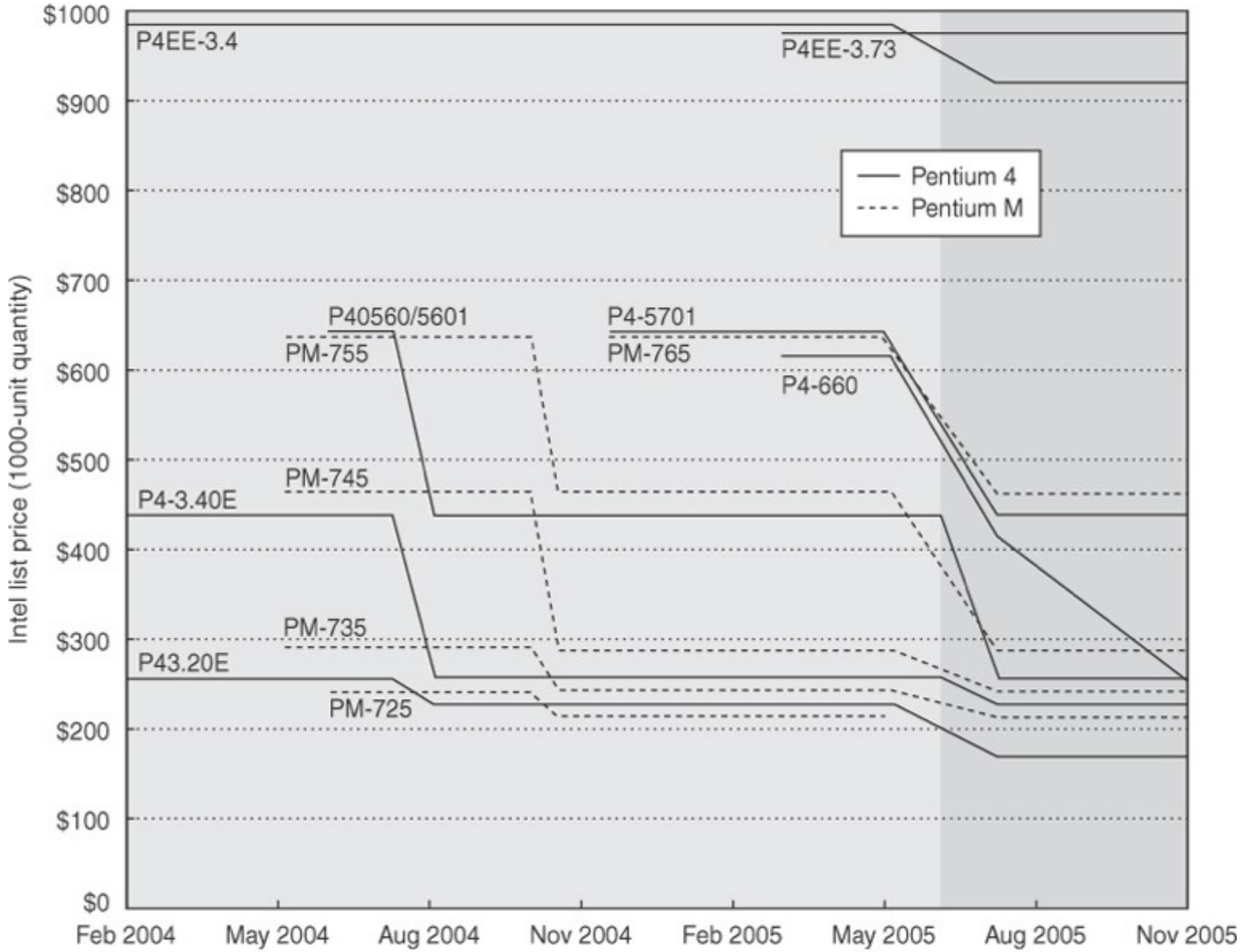
1962
1965
1970
Relative Manufacturing Cost/Component
Number of Components Per Integrated Circuit
10^5
10^4
10^3
10^2
10
1
1
10
10^2
10^3
10^4
10^5



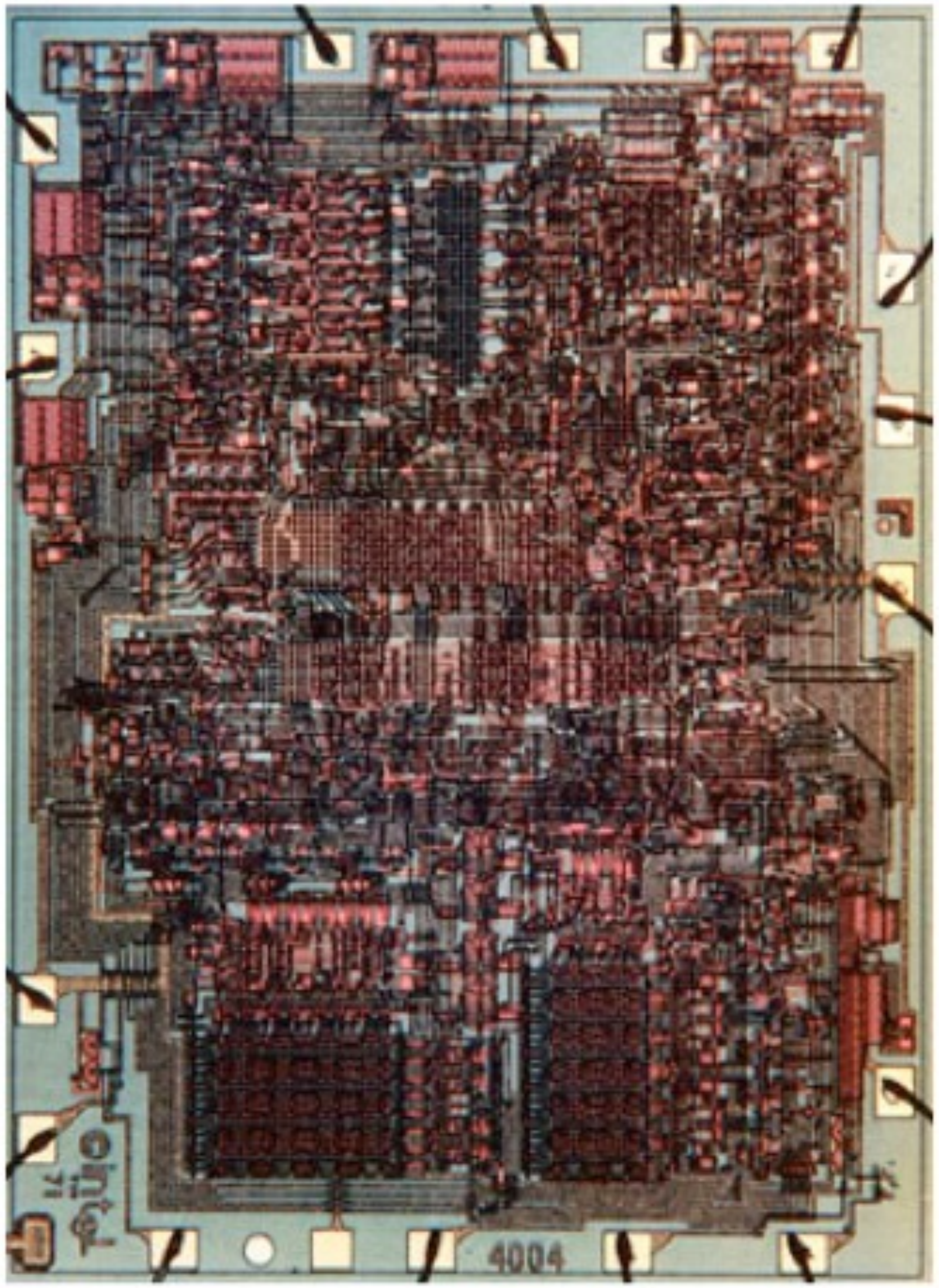
80
70
60
50
40
30
20
10
0
Dollars per
DRAM chip
16K bits
64K bits
256K bits
1M bits
4M bits
16M bits
64M bits
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
Year

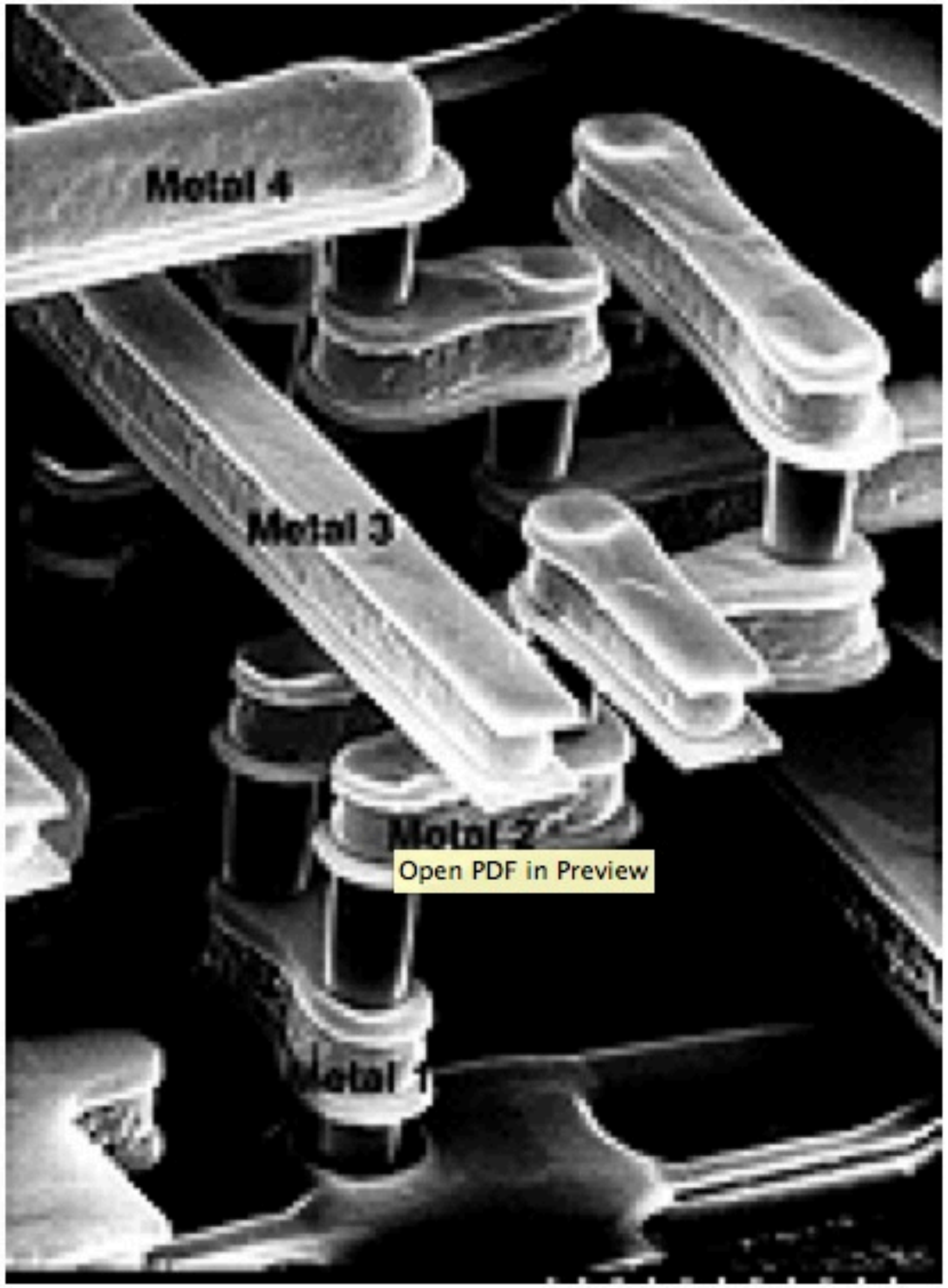


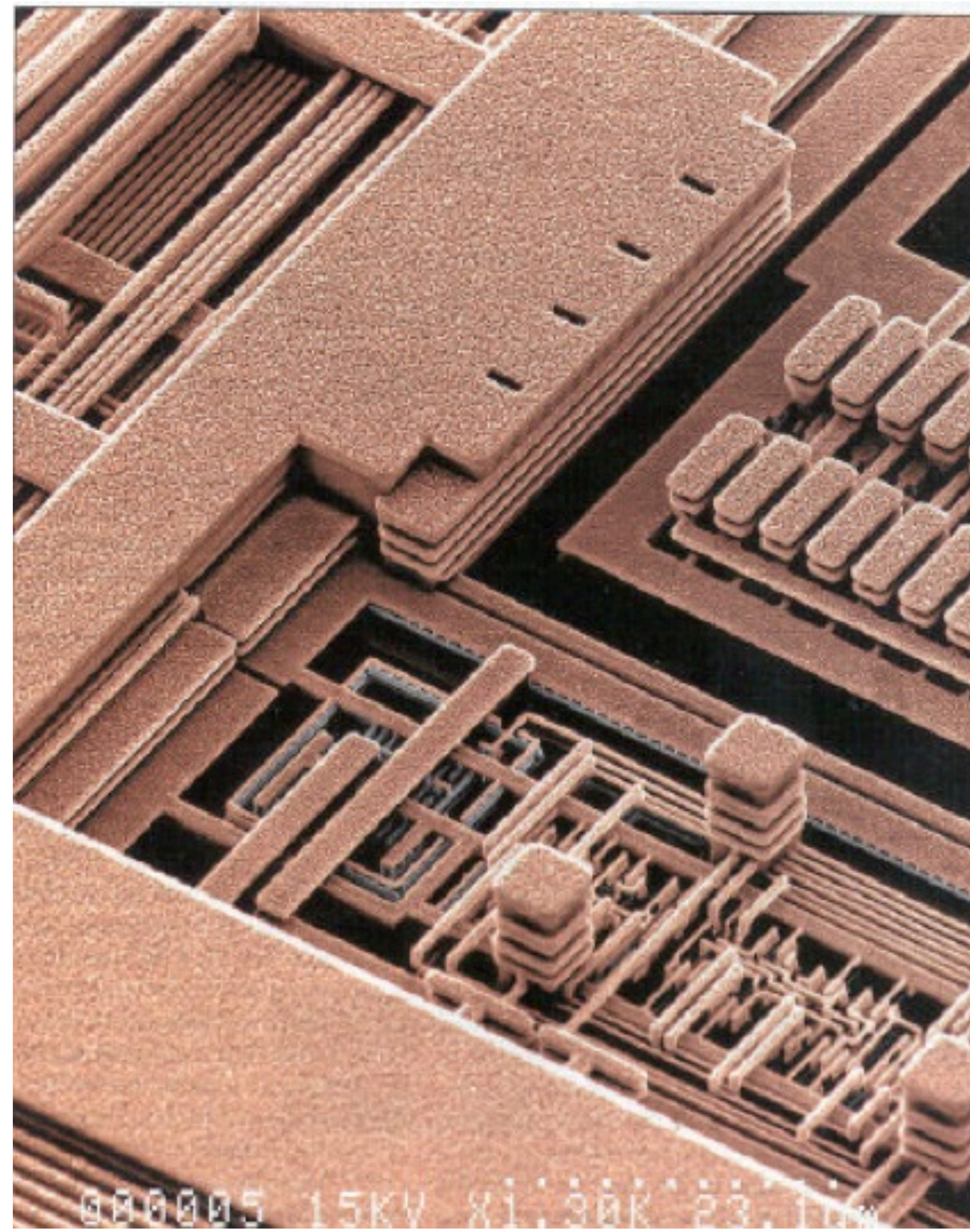
Performance (vs. VAX-11/780)
10,000
1000
100
10
0
VAX-11/780
25%/year
1.5, VAX-11/785
VAX 8700
5
Sun-4/260
9
MIPS M/120
13
MIPS M2000
18
IBM RS6000/540
24
HP PA-RISC, 0.05 GHz
51
Alpha 21064, 0.2 GHz
80
PowerPC 604, 0.1GHz
117
Alpha 21064A, 0.3 GHz
183
Alpha 21164, 0.3 GHz
280
Alpha 21164, 0.5 GHz
481
Alpha 21164, 0.6 GHz
649
Alpha 21264, 0.6 GHz
993
Alpha 21264A, 0.7 GHz
1267
Intel Pentium III, 1.0 GHz
1779
AMD Athlon, 1.6 GHz
2584
Intel Pentium 4,3.0 GHz
4195
AMD Opteron, 2.2 GHz
5364
Intel Xeon, 3.6 GHz
5764
64-bit Intel Xeon, 3.6 GHz
6505
52%/year
≈20%
1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006

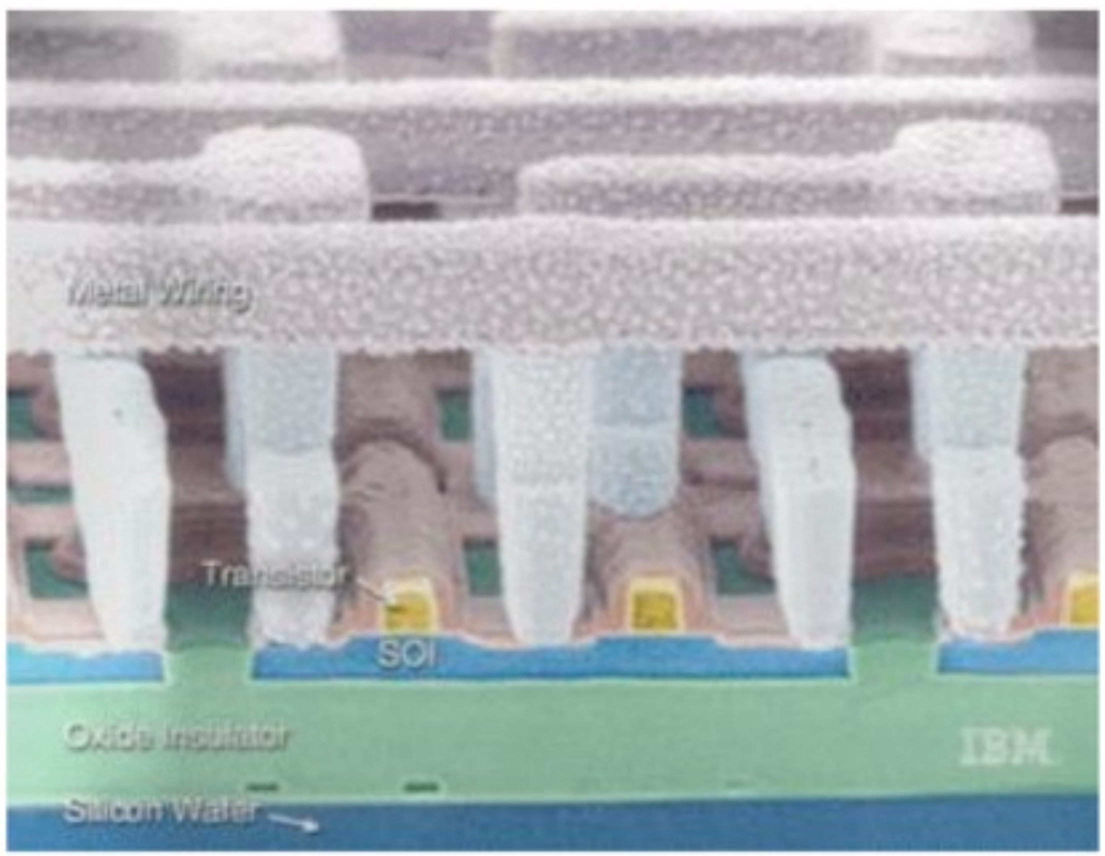


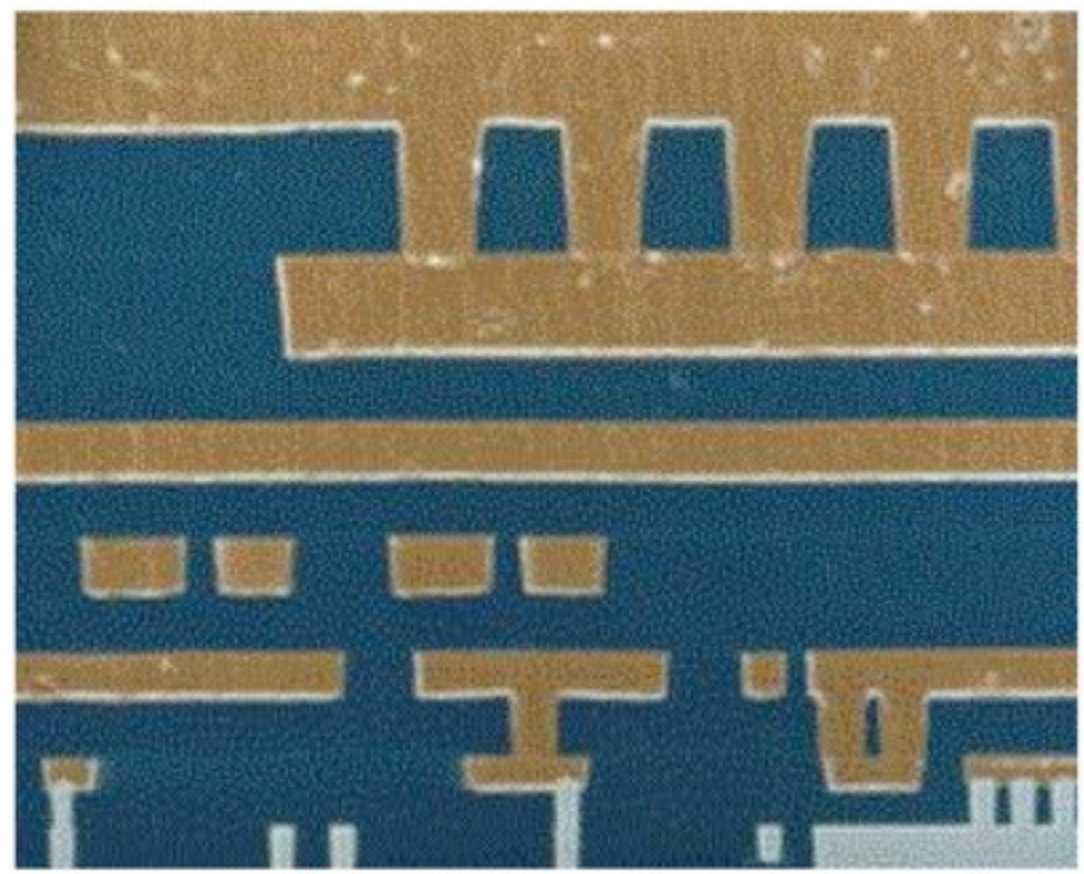
$1000
$900
$800
$700
$600
$500
$400
$300
$200
$100
$0
Intel list price (1000-unit quantity)
Feb 2004
May 2004
Aug 2004
Nov 2004
Feb 2005
May 2005
Aug 2005
Nov 2005
Pentium 4
Pentium M
P4EE-3.4
P4EE-3.73
P40560/5601
PM-755
P4-5701
PM-765
P4-660
PM-745
P4-3.40E
PM-735
P43.20E
PM-725

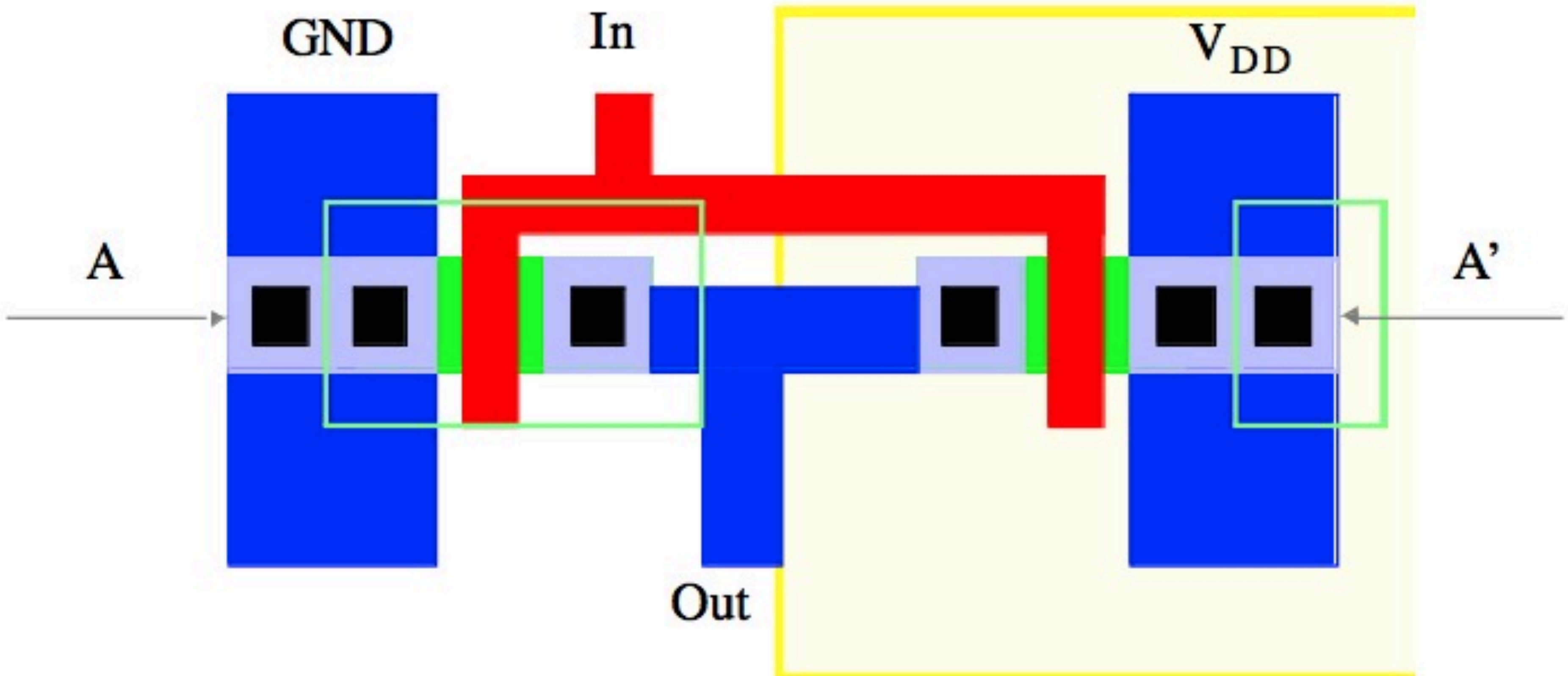

4004

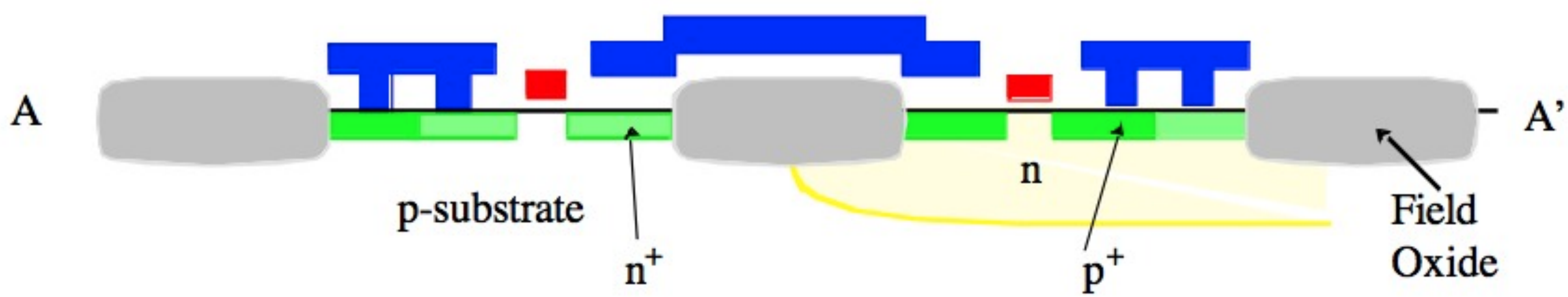

Metal 4
Metal 3
Metal 2
Open PDF in Preview




Metal Wiring
Transistor
SOI
Oxide insulator
IBM
Silicon Wafer




GND
In
V_{DD}
A
A'
Out


A
A'
n
p-substrate
n^+
p^+
Field
Oxide

What is scaling?

- Feature size shrinks
- Change of metal layer material
- 3D stacks
- Dielectrics
- Testing
- used to be voltage and it was great!!
- mask manufacturing

WHY no 3D?

- Heat!
- Design complexity
- Maybe useful for memory
- Silicon through VIAs and complexity
- Flip chip and MCM is "good enough"

What are the current challenges to Moore's Law?

- How can different transmit channel materials such as carbon nanotubes be integrated into traditional silicon chips?
- What would a new fundamental structure for CMOS look like?
- What kinds of advances in system design can help with power consumption?
- How many of these predictions were incorrect?
- Moore claimed that "it is not even necessary to do any fundamental research or to replace present processes. Only the engineering effort is needed". What is this distinction and for how long did it last
- It would seem that Moore's law must eventually reach a physical limit since presumably transistors cannot be made smaller than atoms
- Who is this written for?
- How many of these "Difficult Challenges" are expected to be solved in the near term?
- What effect would solving some but not all of the challenges listed have on semiconductors in the future?
- How accurate are the long term problem predictions likely to be?

- What happens with this advice? (who is motivated by it.)
- This is primarily evolutionary work - how does the energy to solve these problems coexist with more revolutionary redesigns of what a computer is?
- How should computer scientists react to this roadmap? (or - what programs will be be able to create in 2016 that we can't now?)
- Moore mentions that moving to integrated circuits solves some of the heat issues of the time. When does heat start being a major problem again?
- How reliable were the non-integrated circuit computers?
- Was Moore the first to notice this rate of growth / postulate it would continue for the forceable future?
- Where does his data come from?
- Are there opportunities for better collaboration with device and circuit engineers to innovate at the system level?
- what can we do at our level to better use 3D die stacking, lots of Vdd domains, clock optimizations, etc?
- have design tools been keeping up with complexity in last 6 years?
- Does random logic scaling always catch up to regular structures? (ie design/test innovations can be made so that both eventually hit theoretical technology limits and not complexity/cost limits) (e.g. fpga products hit tech nodes earlier)
- At the architecture/systems level, is integration of heterogeneous processing components the way to go?
- design/verification person hours = complexity / sophistication of tools. How fast is hours increasing?
- Do we need to maintain technology scaling today to continue explosive growth of computer industry?

Does it matter?

How do architects do what they do?

What should HW provide?

Why HW?