

集積回路の歴史(プロセッサ)

□ マイクロプロセッサ (MPU)

- i4004(1970, Intel)
- i8008(1972, Intel)
- i8086(1978, Intel)
- 68000(1979, Motorola)
- Pentium(1993, Intel)



f=714kHz, 2300Tr.



f=66MHz, 3,100,000Tr.

集積回路の歴史(メモリ)

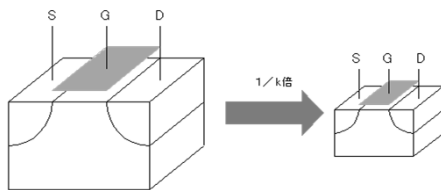
□ メモリIC (DRAM)

- 1kb (1972, Intel)
- 4kb (1975, Texas Inst.)
- 16kb (1977, Mostek)
- 64kb (1980, Hitachi)
- 256kb (1983, Fujitsu)
- 1Mb (1986, Toshiba)
- 4Mb (1989, Hitachi)
- 16Mb (1991, Hitachi)
- 64Mb (1994, NEC, Samsung)
- 256Mb (1997, Samsung)
- 512Mb (2003, Samsung)
- 1Gb (2004, Samsung)

メモリ容量:
3年で4倍
32年で1,000,000倍

集積回路の進化の源: スケーリング則

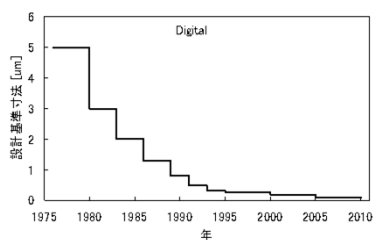
□ 集積回路を構成するMOSTランジスタを小さくすると...



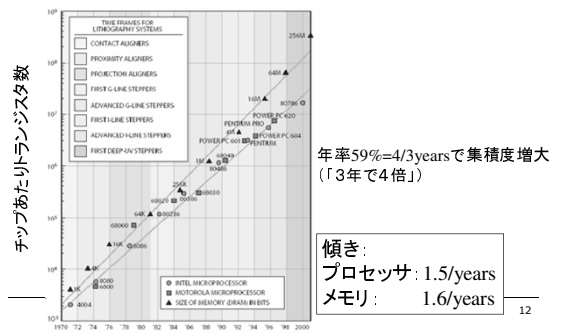
スケーリング則の効果

- 信号遅延 = $1/k$ (高速化)
- 消費電力 = $1/k^2$ (低消費電力化)
- 集積度 = k^2 (高性能化 or 低価格化)

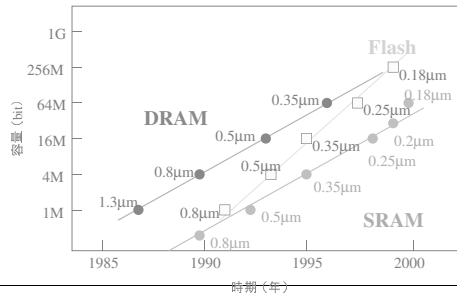
MOSTランジスタ微細化の歴史



Gordon Moore's Law



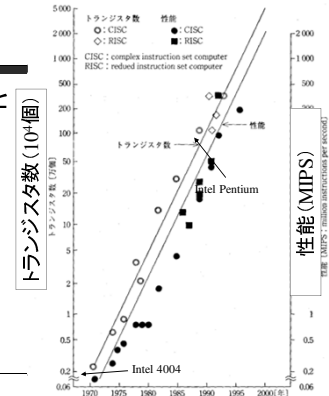
メモリアICの開発動向



13

高性能化の歴史

プロセッサの規模と性能の推移

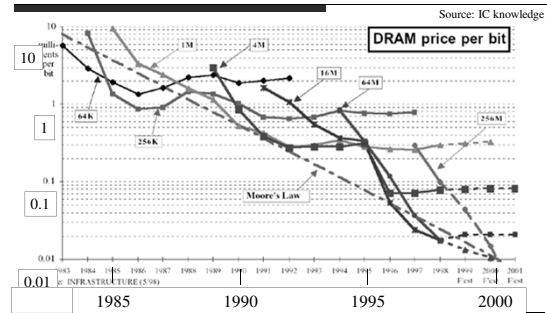


スケーリング則のもたらすもの(その2)

- 機能単価: 「価格/機能」
 - スケーリングにより継続的な機能向上が可能
 - 他の産業では見られない特異性

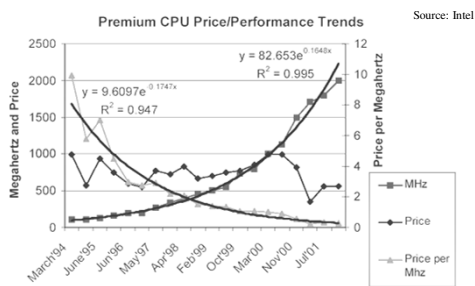
15

低コスト化(その1: DRAMのビット単価)



16

低コスト化(その2: クロック周波数あたりのプロセッサ価格)



17

機能単価の低減のみでよいか?

- 否。
- 「機能飢餓」が必要
 - 「より高性能なものが求められている」状態
 - 電子産業は長年この状態にある
- シーズとニーズの両立
 - ユーザ側: 機能飢餓
 - メーカー側: スケーリング則によるメリット

18

集積回路の世界の今後は・・・？

□ ポイント:

- 「機能飢餓」は続くのか？
- スケーリングなどの技術的実現性は続くのか？

19

Moore's Lawの終わり: 技術面

□ MOSTランジスタの微細化の限界

- MOSTランジスタは原子よりは小さくならない
- $L: \sim 0.1\mu\text{m} (=100\text{nm})$
 \leftrightarrow Si原子 $\sim 1\text{nm}$
- その他の制限要因
 - 消費電力の増加(もれ電流)
 - 回路規模の増大と設計技術の乖離
- $L: \sim 0.01\mu\text{m} (=10\text{nm})$ (めどはたっている)

20

Moore's Lawの終わり: ニーズ面

□ 果たして、Moore's Lawによる高性能な集積回路は必要なのか？

- 例) PCの性能？
- 例) 携帯電話の機能？
- 技術的要因だけでは決まらない
- ……これ以上の技術の進歩は必要なのか・・・？

21

集積回路の今後に対する別の見方

□ 集積回路の製造・設計技術の成熟

- 高性能な集積回路(MPU, メモリ, ...)
- 低価格な集積回路(性能はそこそこ)

□ 高性能な集積回路:

- = 最先端の製造技術(工場)
 \rightarrow (~1000億円以上)

□ 低価格な集積回路

- 性能はそこそこ。しかし・・・

22

これからの情報技術と集積回路

□ 「道具」としての集積回路

- 設計技術・製造技術の成熟
- 敷居が下がってきている(高専の講義でも扱う!)

□ 情報技術の実現方法としての集積回路

- パソコンを使ってプログラム:
 できることは、たかが知っている
 (パソコンの枠の中だけの世界)
- 面白いもの・作りたいものを実現したいときに、
 道具として活用する(実世界とのつながり)

23

集積回路を「道具」にするためには・・・

- 「作りたいもの」を設計しなければいけない
- 1億個のトランジスタが「使える」(available)
- ……どうやって設計するのか？

□ 「作りたいもの」を「言語」で記述する

- Hardware Description Language: HDL
- 言語で書いたものを、半自動で集積回路に変換
 (トップダウン設計)

24

講義のスケジュール(前半)

- 第1週(4/14) 現代社会と集積回路
- 第2週(4/21) ハードウェア記述言語と論理回路
- 第3週(4/28) シミュレーションと論理合成
- 第4週(5/12) 実習(1): FPGAへの論理回路の実装(1)
- 第5週(5/19) 組み合わせ論理回路のHDL記述(1): デコーダ等
- 第6週(5/26) 組み合わせ論理回路のHDL記述(2): 演算器等
- 第7週(6/2) 演算器の高速化
- (6/9頃) 前期中間試験

25

講義のスケジュール(後半)

- 第8週(6/16) 順序回路のHDL記述(1): フリップフロップ・カウンタ
- 第9週(6/23) 実習(2): FPGAへの論理回路の実装(2)
- 第10週(6/30) 順序回路のHDL記述(2): ステートマシン
- 第11週(7/7) 実習(3): FPGAへの論理回路の実装(3)
- 第12週(7/14) 論理合成と制約条件
- 第13週(7/21) HDL設計の関連技術
- (9/8: 予備日)
- 第14週(9/15) IPコアと集積回路設計
- (9/22頃) 前期末試験
- 第16週(9/29) 前期復習

26

教科書

- 深山正幸ほか
「HDLによるVLSI設計」(共立出版)

27