システムアーキテクチャ論：概要

99/10/7　木曜4限

琉球大学　工学部　情報工学科
和田　知久

wada@ie.u-ryukyu.ac.jp
http://bw-www.ie.u-ryukyu.ac.jp/~wada/

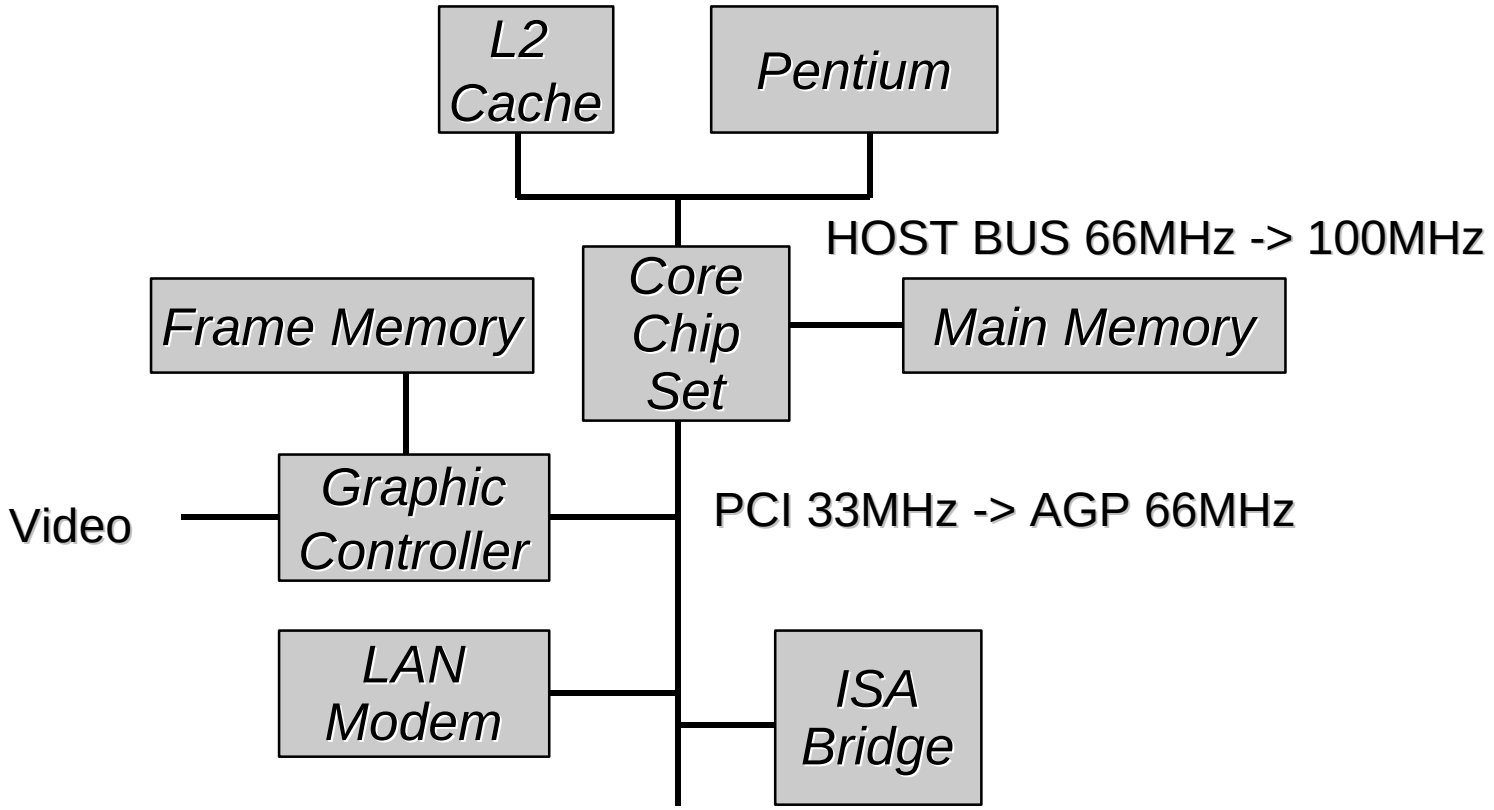
アウトライン

- 本講義では最近のデジタル機器のアーキテクチャをレビューしてゆく。色んな情報を提供する。
- 具体的には、
  1) PCアーキテクチャ　2) MPEG2
  3) 3次元グラフィックス　4) 携帯電話
- 可能であれば専門家を招待して、講演を行ってもらう。
- 評価は出席と最終レポートで行う。
  最終レポート：自分の好きな製品/システム等を調査しレポートする。

[1] PCアーキテクチャ

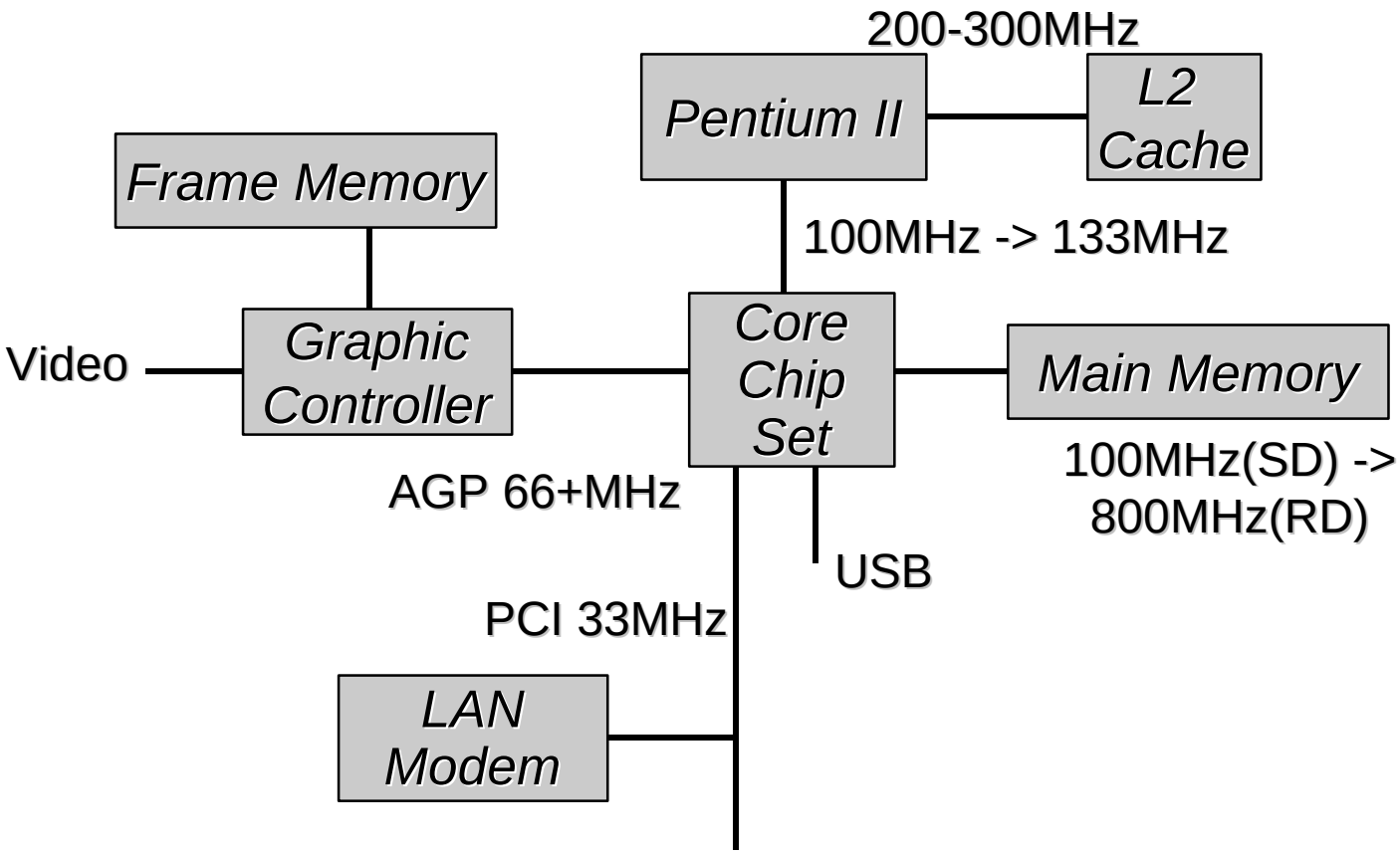
- LSIの進歩により、コストパフォーマンスが劇的に進化。
  - ●5万円PCの時代
- コスト／パフォーマンスでワークステーションに勝利。
  - ●LSIの集積度向上が直接低コスト化を実現

PentiumPCのアーキテクチャ



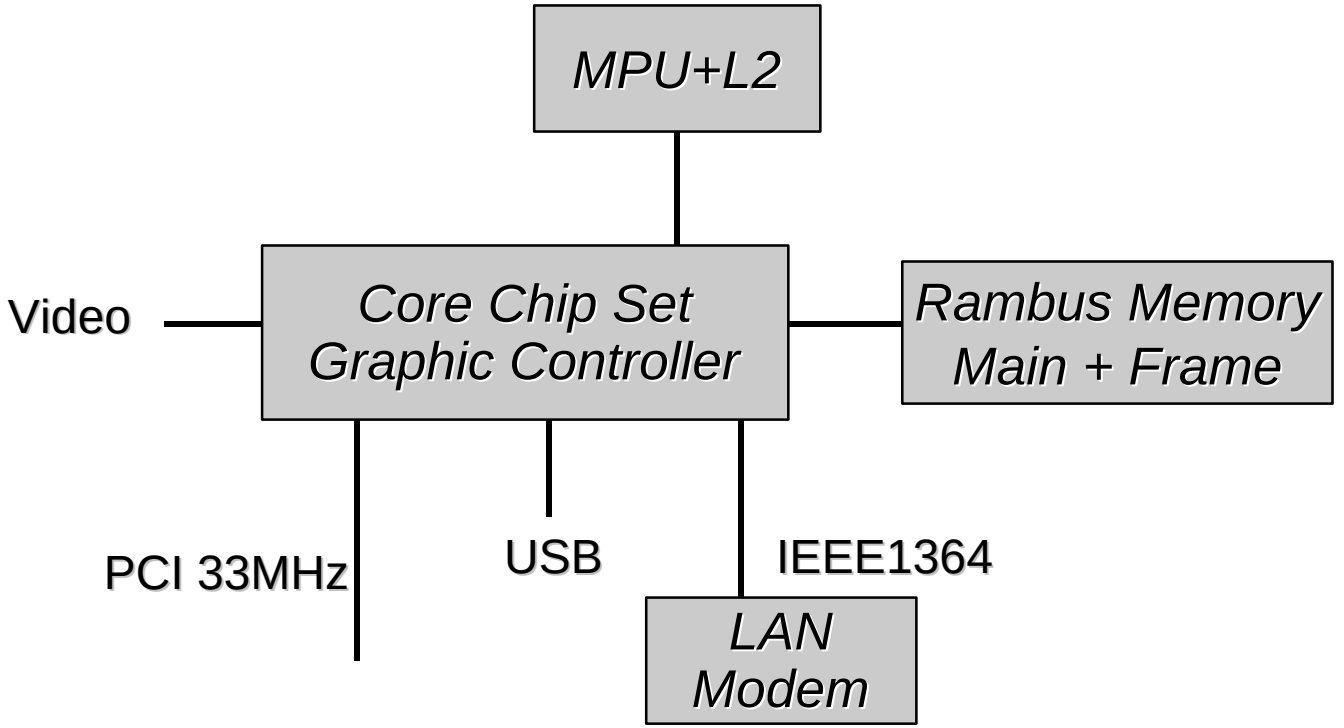
● HOST BUSの分離で、MPUと主記憶　ｸﾞﾗﾌｨｯｸの周波数が分離

PentiumII PCのアーキテクチャ



● L2 BUSの分離し、MPU周波数の1／2に設定

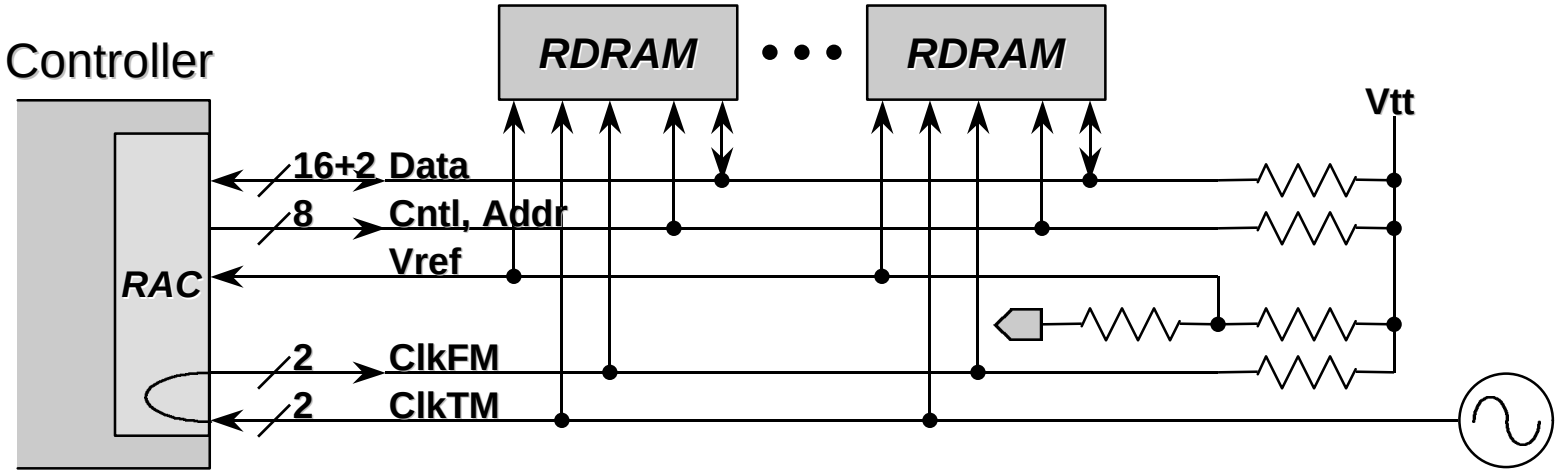
近未来PCのアーキテクチャ



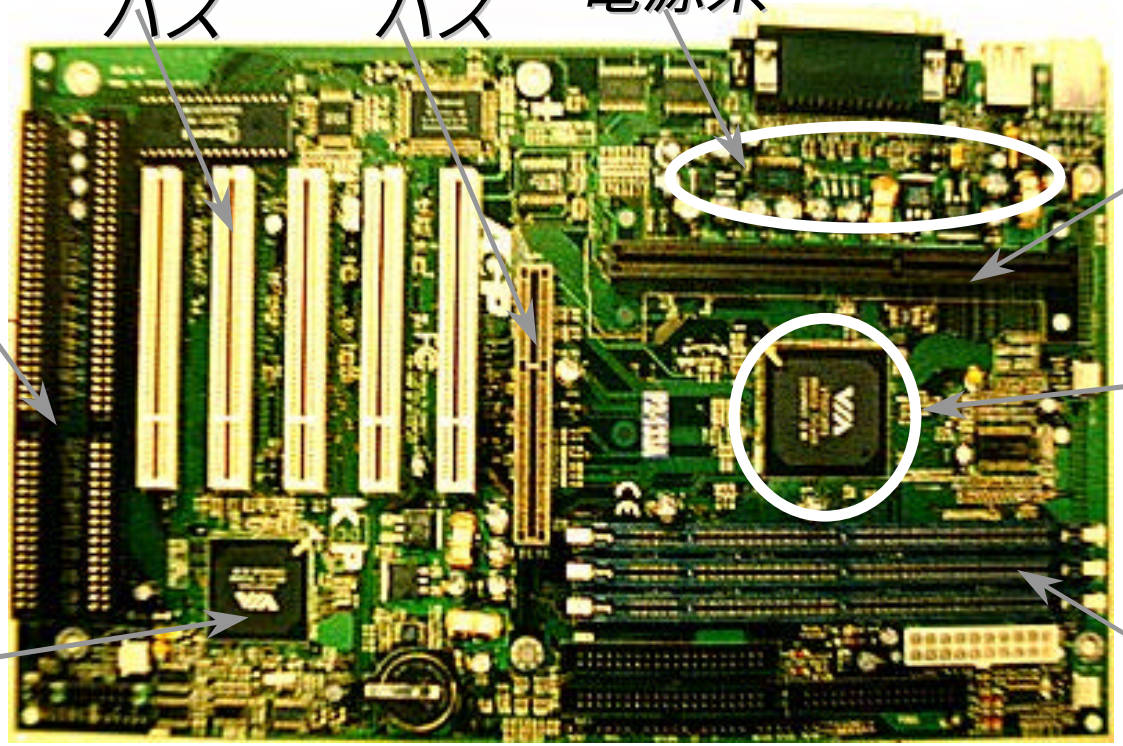
- 高速シリアル／低ピンカウントBUSで低ピン化

Direct RAMBUS

- 400MHz clockの両エッジデータ転送
- 18bitデータバス、アドレスバスも8ビットと少ない
- 最大2Byte x 800MHz = 1.6GB/s
- 現状の100MHz SDRAM, 64bitバス（800MB/s）の2倍
- 但し、Latencyが遅めで、既存アプリで性能上がらず。



あるPentiumII PCマザーボード



PCを構成するLSI数は減った。ソケットが面積を決める。

PCアーキテクチャのトレンド

- 集積化され、LSI数が減る。
- USB, IEEE1394の高速シリアルインターフェイスで低ピン化、ソケットも減る。
- Rambus採用での64→16ビットバス化
- 高性能＋小型化
- コスト的、技術的に難しいものは集積化されにくい。（低コストDRAM、高速通信）

PentiumII PCでのLSI

MPU (1)	INTEL / AMD	独壇場
Cache SRAM (2)	NEC / Toshiba....	集積される
コアチップセット(1-3)	INTEL / VIA	ｸﾞﾗﾌｨｯｸｽを取りこむ
DRAM (4-8)	Samsung / Micron..	コストで日本苦しい
BIOS (1)	AMI / Phenix	ソフトウエア
I/Oコントローラ(1)	???	低コスト、集積される
グラフィックCNTL(1)	Matrox / ATI / S3	独壇場、Intel参入
フレイムバッファ(2)	Micron / NEC..	集積化されそう
LAN/MODEM (1)	3Com etc.	まだ高性能化続く
SOUND (1)	Creative etc.	集積される
電源系	Maxim etc.	安いが必要

[2] MPEG2

- DVDやPCでの動画再生で、画像・音声の圧縮・解凍の成功した標準
- その他
  静止画用： JPEG
  VHSクオリティ：MPEG１
  ビデオ会議： H.26x
  デジタルビデオテープ： DV

MPEG2画像系のアーキテクチャ

- 画像圧縮の簡単な原理

1) 以前の良く似た画像を見つけて、違いのみ転送する。差のデータ量が小さい。

2) 画像の高周波成分をカットする。

3) 符号化を工夫してデータを減らす。

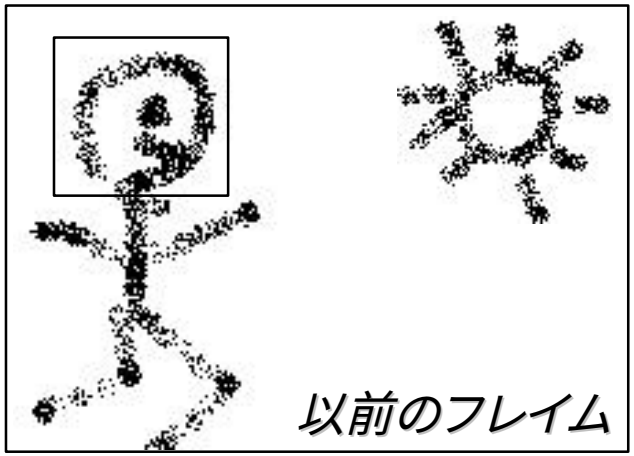
MPEG画像に関する基礎知識

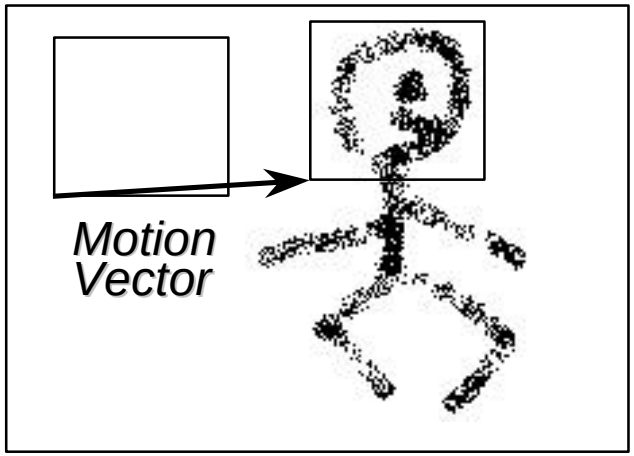
- コンピュータではRed,Green,Blueで画素を表す。
- MPEG（Video）ではY(luminance),Cb,Cr(color-difference)

  Y = 0.299R’ + 0.587G’ + 0.114B’
  Cb=- 0.169R’ - 0.331G’ + 0.500B’
  Cr= 0.500R’ - 0.419G’ - 0.081B’
- 人間はcolor成分に鈍感なので、Cb,CrはYの1／4サンプル
- 画像の処理単位＝マクロブロック、8x8画素

1) 差の小さいデータを探す。

- Motion Estimation
  ○あるサーチ領域でエラー最小のMotion Vectorを探す。
  ○画像の代りにMotion VectorとErrorを転送。





Motion Estimationの基本計算式

平均絶対値誤差

$$MAD(x,y) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} |S(m+x,n+y)-B(m,n)|$$

$-p <= x, y <= p$

$(2p+1)x(2p+1)$ Search Window in previous picture

Input NxN Block

- 多数のアダーとコンパレータで実現できるが、
- 現実にはアルゴリズムの改良で、サーチウインドウを広げ、H/Wを小さくしたものが主流。

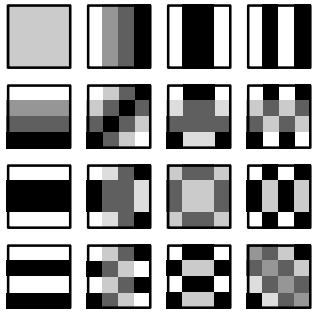
2）画像の高周波成分をカット

- Transform Cordingとも言うが、原理は自然の画像は空間的に高周波の成分を落としても質が低下しない。
- 低周波成分は荒く量子化できる。
- Discrete Cosine Transformをし、DCT要素を表現するビット数を減らす。

DCTの基本要素の例

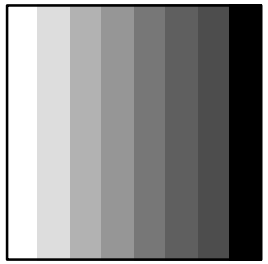
- 実際は8x8であるが、ここでは4x4の例を示す。
- 任意の画像を下記要素に分解する。

DC



高周波

8x8ブロックの変換例



```
87.5 75 62.5 50 37.5 25 12.5 0
87.5 75 62.5 50 37.5 25 12.5 0
87.5 75 62.5 50 37.5 25 12.5 0
87.5 75 62.5 50 37.5 25 12.5 0
87.5 75 62.5 50 37.5 25 12.5 0
87.5 75 62.5 50 37.5 25 12.5 0
87.5 75 62.5 50 37.5 25 12.5 0
87.5 75 62.5 50 37.5 25 12.5 0
```

DCT

```
350 228 0 24 0 7 0 2
  0   0 0  0 0 0 0 0
  0   0 0  0 0 0 0 0
  0   0 0  0 0 0 0 0
  0   0 0  0 0 0 0 0
  0   0 0  0 0 0 0 0
  0   0 0  0 0 0 0 0
  0   0 0  0 0 0 0 0
```

Scalar Quantization

```
44 14 0 1 0 0 0 0
 0  0 0 0 0 0 0 0
 0  0 0 0 0 0 0 0
 0  0 0 0 0 0 0 0
 0  0 0 0 0 0 0 0
 0  0 0 0 0 0 0 0
 0  0 0 0 0 0 0 0
 0  0 0 0 0 0 0 0
```

1D-DCTの基本計算

$$y(u) = 0.5\, C(u) \sum_{n=0}^{7} \{x(n) \cos[(2n+1)u\pi/16]\}$$

- cosとの掛け算の結果をROMに入れてると、ROMとアキュミュレータで実現できる。
- 2次元は繰り返しで求まる。
- Distributed Arithmeticと呼ばれる方法。

MPEGでのLSI技術

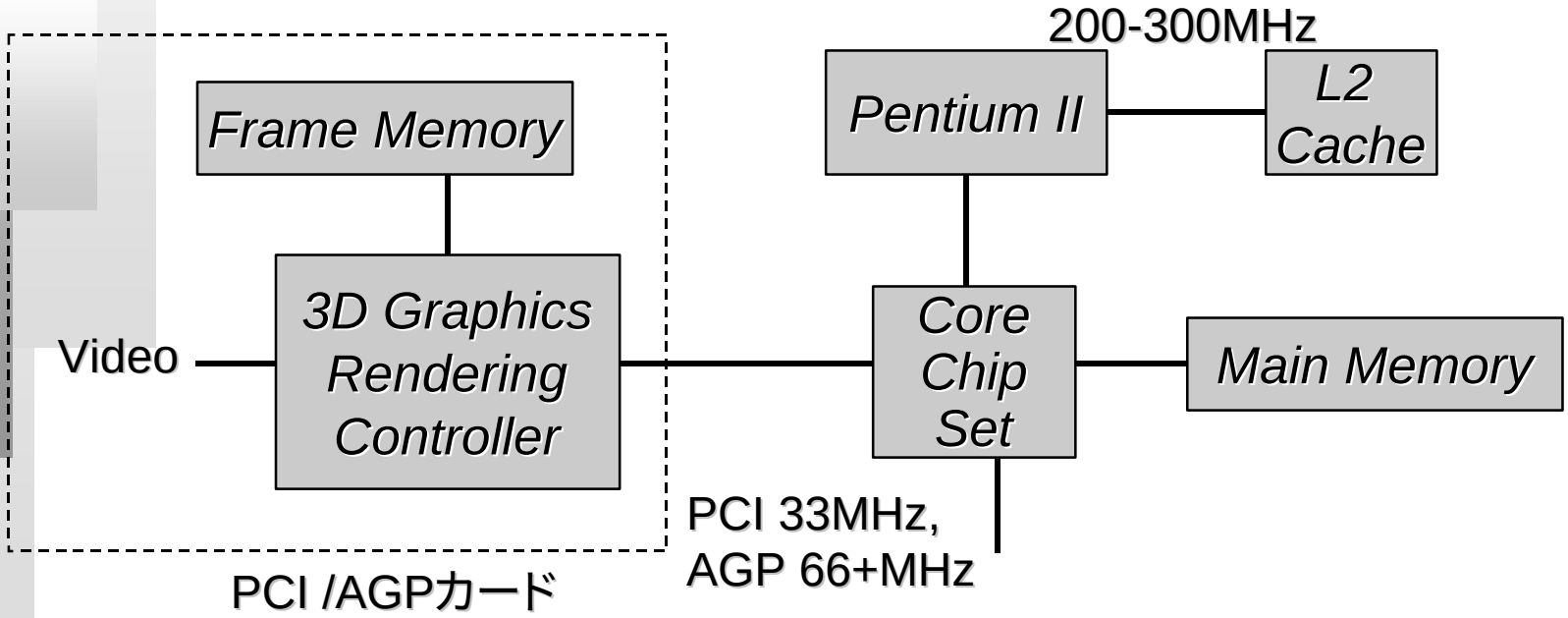
- 多量の加算演算や特殊演算
- DCTではROM演算
- 符号化も特殊

- 汎用MPUでは一見やりにくく見える。
- 信号処理の知識といかにH/WにマッピングするかがKEY。
- H/W設計の工夫よりアルゴリズムの改善が効く。

[3] ３次元グラフィックス

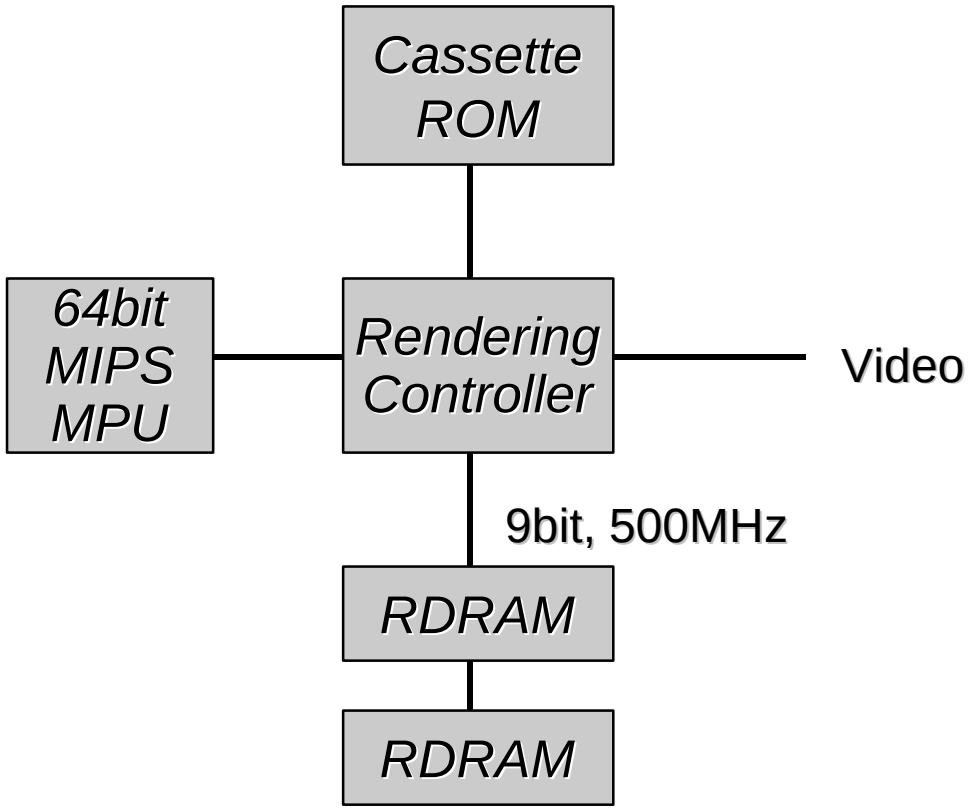
- Nintendo64、PlayStationそしてPCで3次元グラフィックスゲームが今や主流
- 映画やTV放送でも3次元グラフィックスは当たり前

PCにおける3次元グラフィックス



- 今までの2次元グラフィックスカードがそのまま3次元グラフィックスサポートに置き換わった。
- ただ処理内容は大きく異なる。

Nintendo64

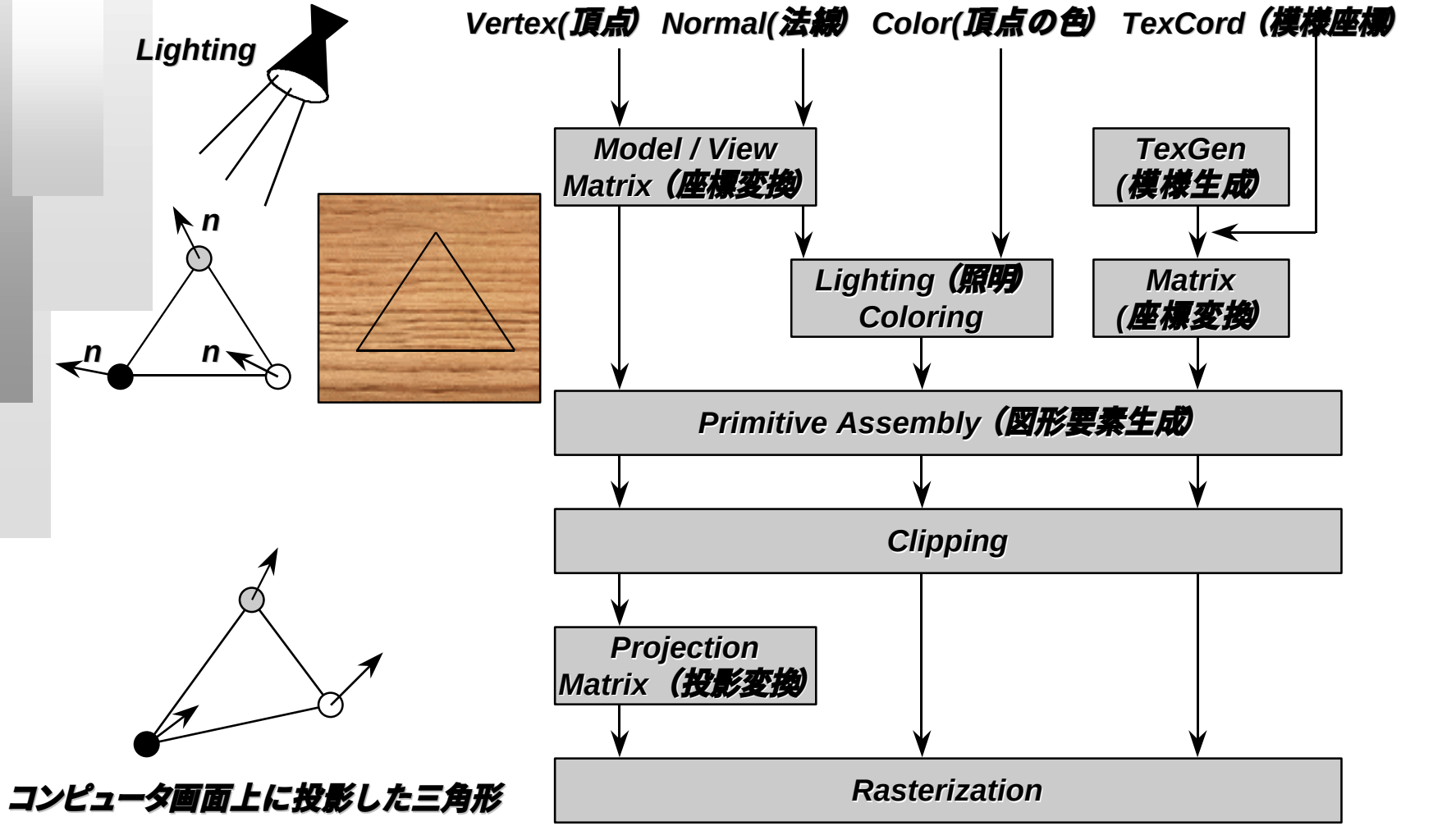


- 3Dグラフィックスではメモリアクセスが多く、
  RambusDRAMでボトルネックを解消している。
- TVゲームではPCに比べて画素数が少なく、その点でメモリB/Wで楽。

3Dグラフィックスに関する基礎知識

- 3次元モデルを三角形等の図形要素でモデル化
- 陰線消去必要であり、2次元ディスプレイ座標（x、y）に加えてDepthの zが加わる。
- ガラス等の半透明サポートのために、(R,G,B)に加えてA(アルファ）＝透過度が加わる。

OpenGL Machine(前半)



ジオメトリ処理

- 頂点（Vx,Vy,Xz,Vw）の座標変換
- 法線(Nx,Ny,Nz)の変換
  ⇒　浮動小数点加算・乗算のマトリックス演算：SIMD向き

- 色データ(R,G,B,A)に対する同じ処理
  ⇒　浮動小数点加算・乗算：SIMD向き

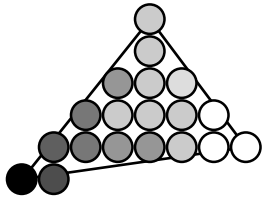
- ベクトル正規化、光源との距離、鏡面反射
  ⇒逆数、平方根、累乗：特殊計算

前半はヘビーな計算処理が主体、高性能MPU向き。
MMX Pentium in PC, 64bit RISC in Nintendo64.

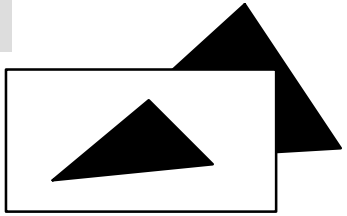
128bit MPU for PlayStation2??

- ISSCC99, TP15.1, SCE & Toshiba
  - 2way 64bit superscalar MPU
  - 10 IEEE single precision FMAC
  - 4 IEEE single FDIV
  - MPEG support--VLD, DCT/IDCT, IQ, CSC, VQ

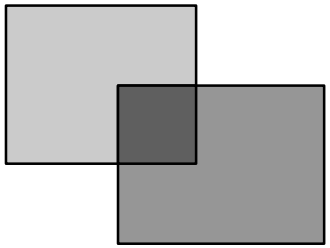
OpenGL Machine（後半）



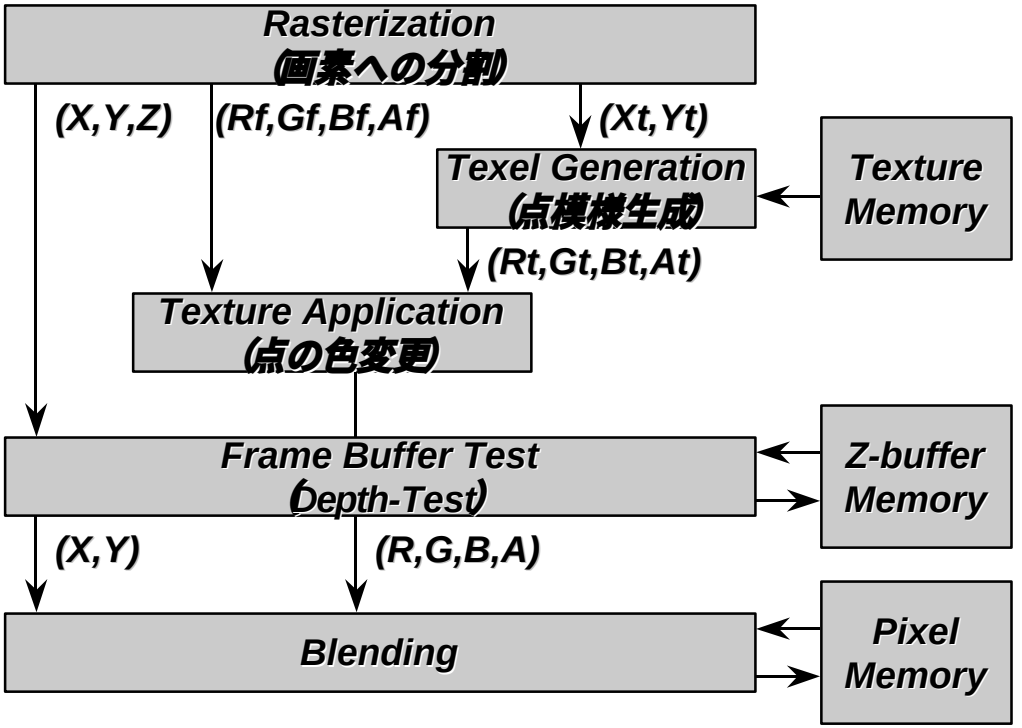
色を線形補間して、画素に分解
グーロー・シェーデイング



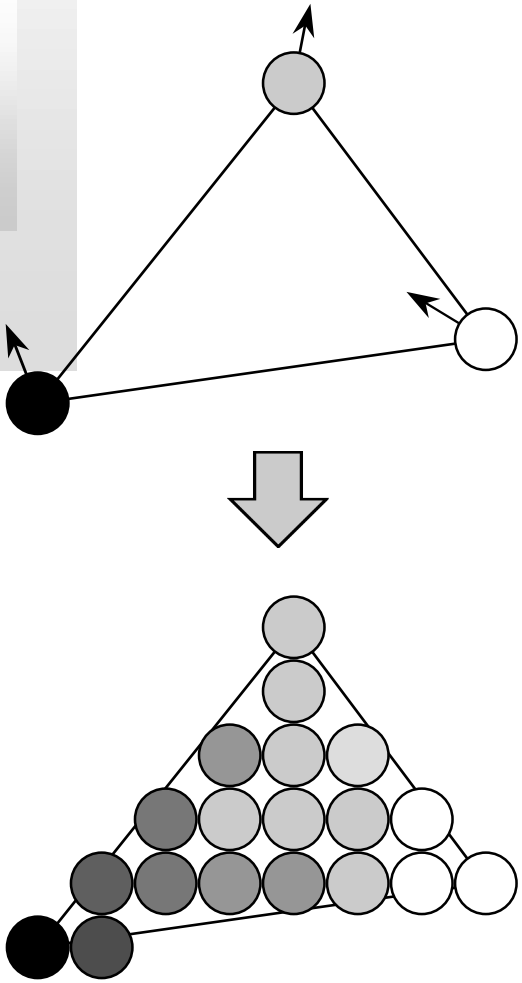
Z-bufferにより見えない
ピクセルは書きこまない



半透明ではブレンド

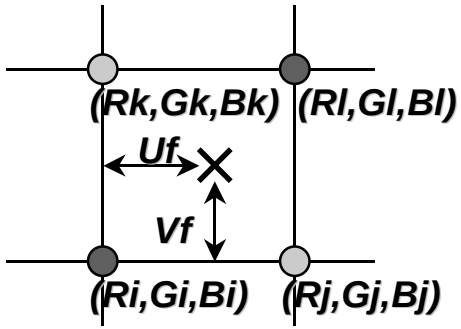


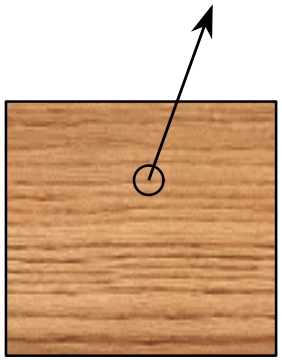
画素への分割



- 3頂点から画素へ分割
  データ増加
- グーロー・シェーディング
  色（Rf,Gf,Bf,Af）を線形補間でつくる。
  整数演算
- フォン・シェーディング
  法線ベクトルを各画素に対して計算し、
  各画素の照明を計算する。
  計算量大

テキスチャマッピング

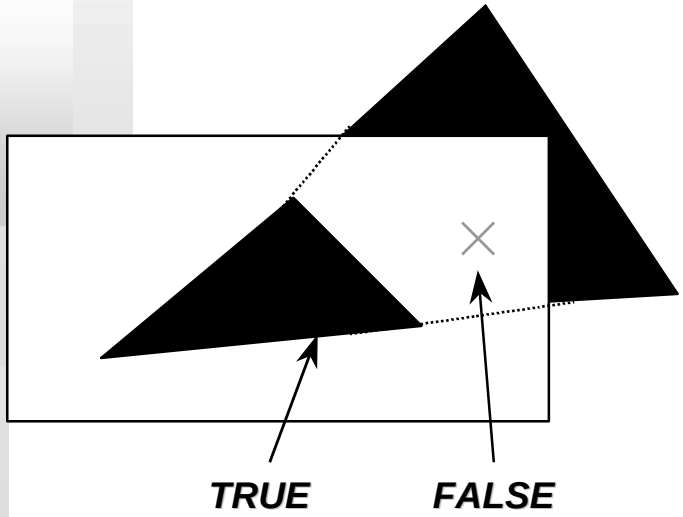




- テキスチャデータは各グリッドポイントにおける (RGB)データ。
- ニアレスト(1点サンプル)
  $Rt = Rl$
- リニア (4点サンプル)
  $Rt = (1-Vf)\{Uf\ Rj + (1-Uf)Ri\} + Vf\ \{Uf\ Rl + (1-Uf)Rk\}$
- ミップマップ (8点サンプル)
  精度の異なる2つのテキスチャデータ間で補間

1つのピクセルに対して、多くのテキスチャデータが必要。
テキスチャメモリアクセスがボトルネック

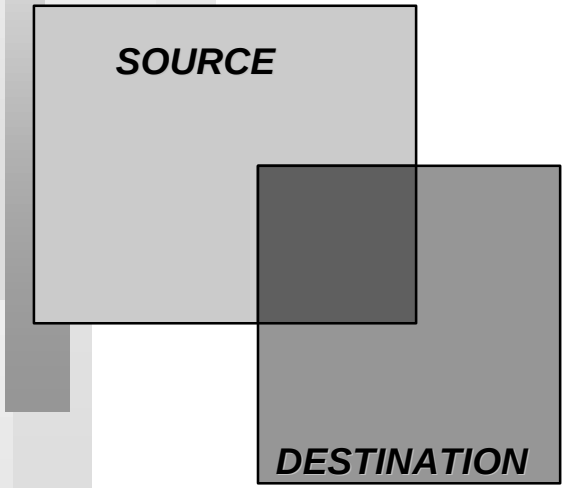
Depth Test



- Depth Testをするには
  1) Z-bufferよりZdestを読み出す
  2) ZsourceとZdestを比較
  3) Zsource < Zdestならば
     ZdestとPixelメモリを更新
- Z-bufferに対してRead Modify Writeが必要。
- Z-bufferは各ピクセルに対して32bit程度。
  これはRGBAの３２ビットと同じ。

Z-bufferとPixelメモリは同一のフレイムメモリを使う場合が多く、
フレイムメモリのI/O BWがボトルネックになる。

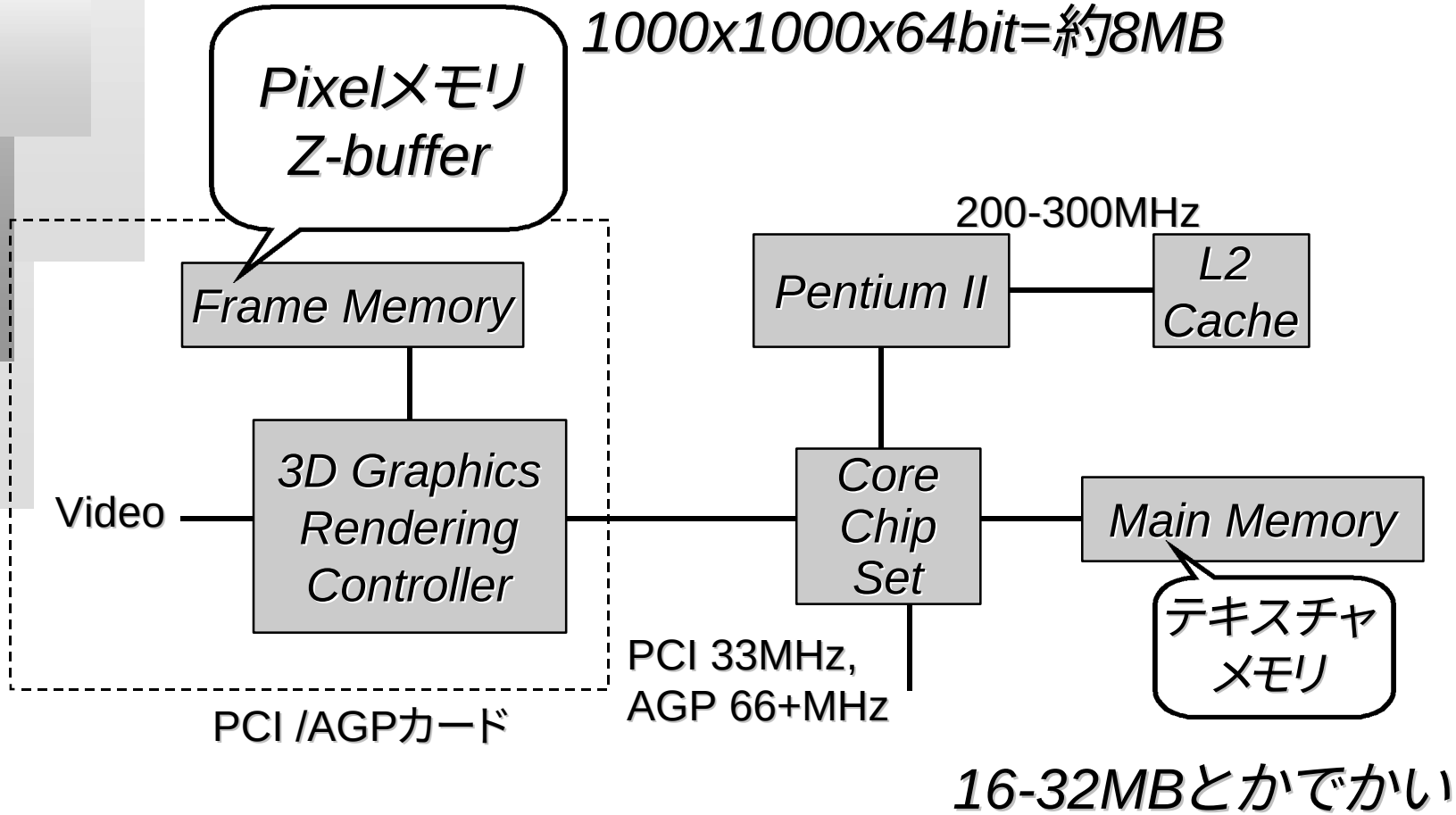
ブレンド



- SOURCE図形上に半透明なDESTINATION図形を描画する場合、
  1) ピクセルメモリより(RGBA)を読み出す
  2) 以下のような混合計算実行
     $Rrst = As \cdot Rsrc + (1-As)\ Rdest$
  3) 結果をピクセルバッファへ書き戻す
- これも、Pixelメモリに対してRead Modify Writeが必要。

フレイムバッファ（DRAM）を内蔵することでBW問題を解決。

PCでのメモリの用途


1000x1000x64bit=約8MB
Pixelメモリ
Z-buffer
Frame Memory
200-300MHz
Pentium II
L2
Cache
3D Graphics
Rendering
Controller
Video
Core
Chip
Set
Main Memory
テキスチャ
メモリ
PCI 33MHz,
AGP 66+MHz
PCI /AGPカード
16-32MBとかでかい

3Dグラフィックスでの LSI技術

- ジオメトリ処理
  - Floating演算のSIMDタイプの並列処理
  - 逆数、平方根、累乗：特殊計算
- 描画処理
  - 整数演算
  - メモリBWがボトルネック

    ⇒フレイムバッファとコントローラの１チップ化
  - PCIがテキスチャ転送のボトルネック

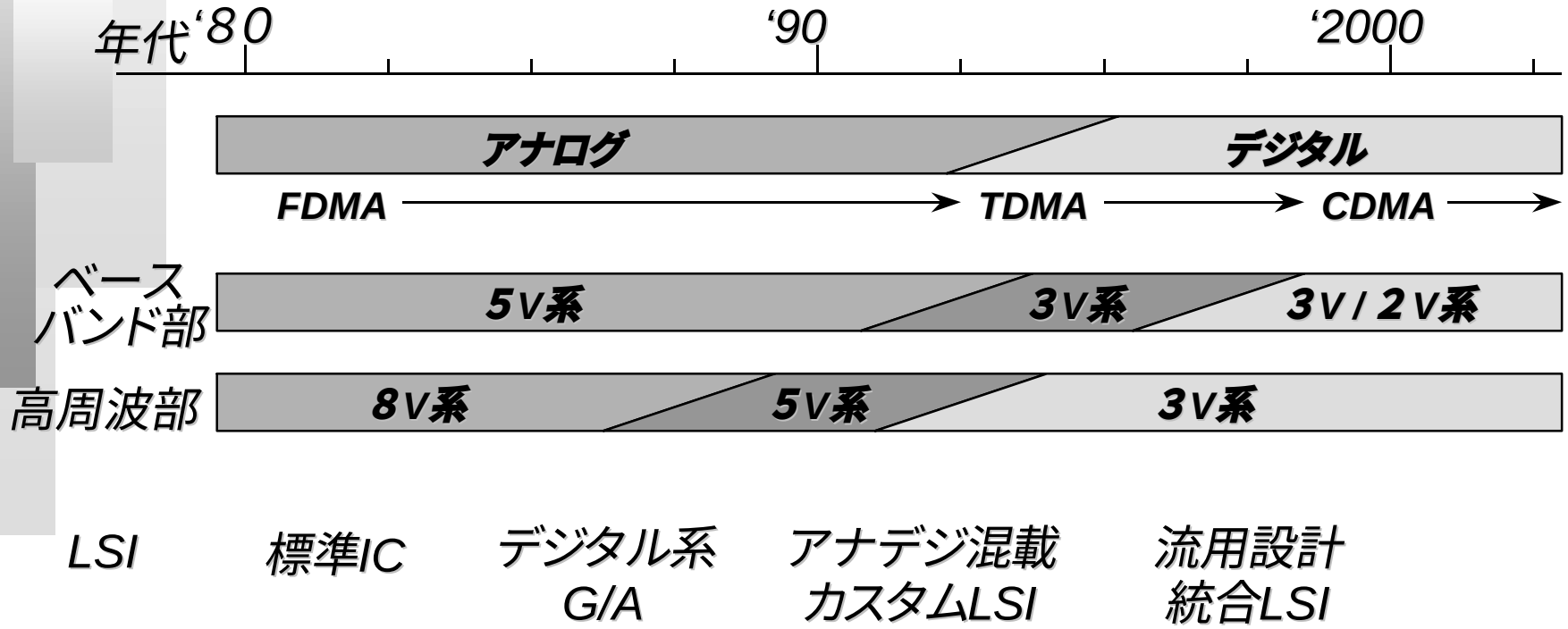
    ⇒AGP倍速化

    ⇒ｸﾞﾗﾌｨｯｸｺﾝﾄﾛｰﾗをコアチップと集積

[4] 携帯電話、PHS

- 日本での加入者数は約5000万加入で伸びは鈍化しているが、さらに増加。
- 世界統一企画の次世代通信端末IMT-2000開発中（CDMA方式）。
- 小型化、低消費電力化（低電圧化）が強く要求される。

移動無線端末の推移

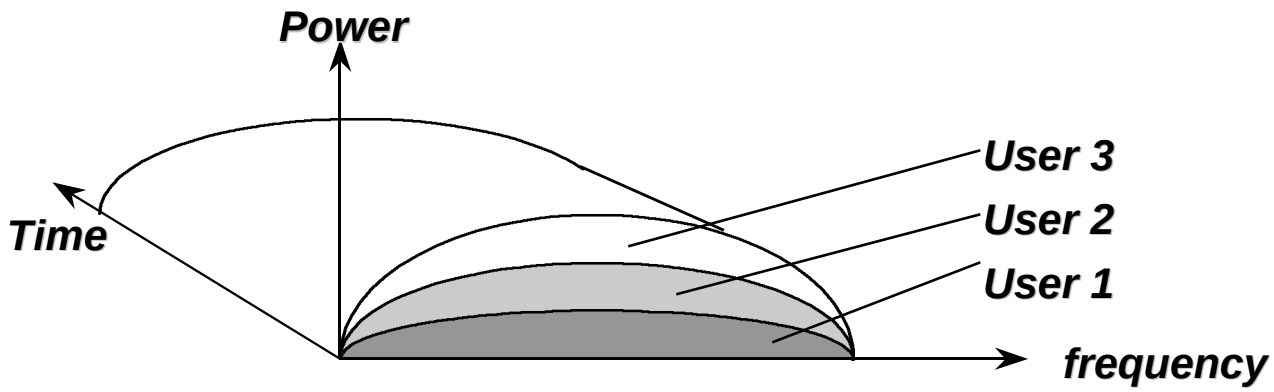

年代
'80
'90
'2000
アナログ
デジタル
FDMA
TDMA
CDMA
ベース
バンド部
5V系
3V系
3V/2V系
高周波部
8V系
5V系
3V系
LSI
標準IC
デジタル系
G/A
アナデジ混載
カスタムLSI
流用設計
統合LSI

次世代携帯無線端末のLSI技術

- CDMA方式で
  - デジタル信号処理が複雑かつ大規模化（4倍）、高速化（6倍
- 動画転送を行うので
  - MPEG4などの画像のCODEC必要
- Rake受信
- 誤り訂正技術
  - Interleave, Convolution Code, Viterbi Decode
- 外部インターフェース
  - USB(Universal Serial Bus)、BlueToothなど

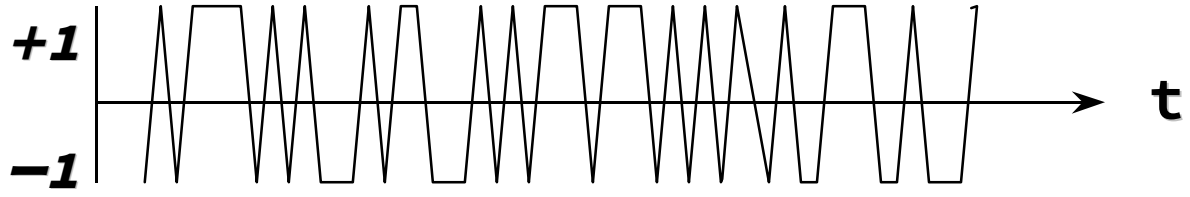
CDMA(符号分割多元接続)方式

- FDMA, TDMAとは異なり時間や周波数をわりあてない。
- 各ユーザをPseudo-Noise Random Sequenceで変調。
  (Direct-Sequence Spread-Spectrum)



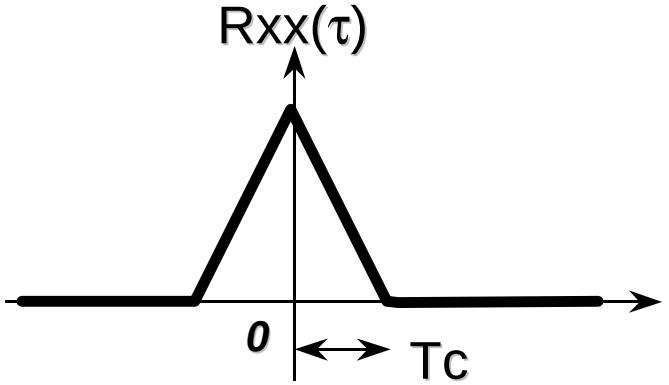
Random Waveformの性質

- バイナリランダム波形 x(t)



- Autocorrelation

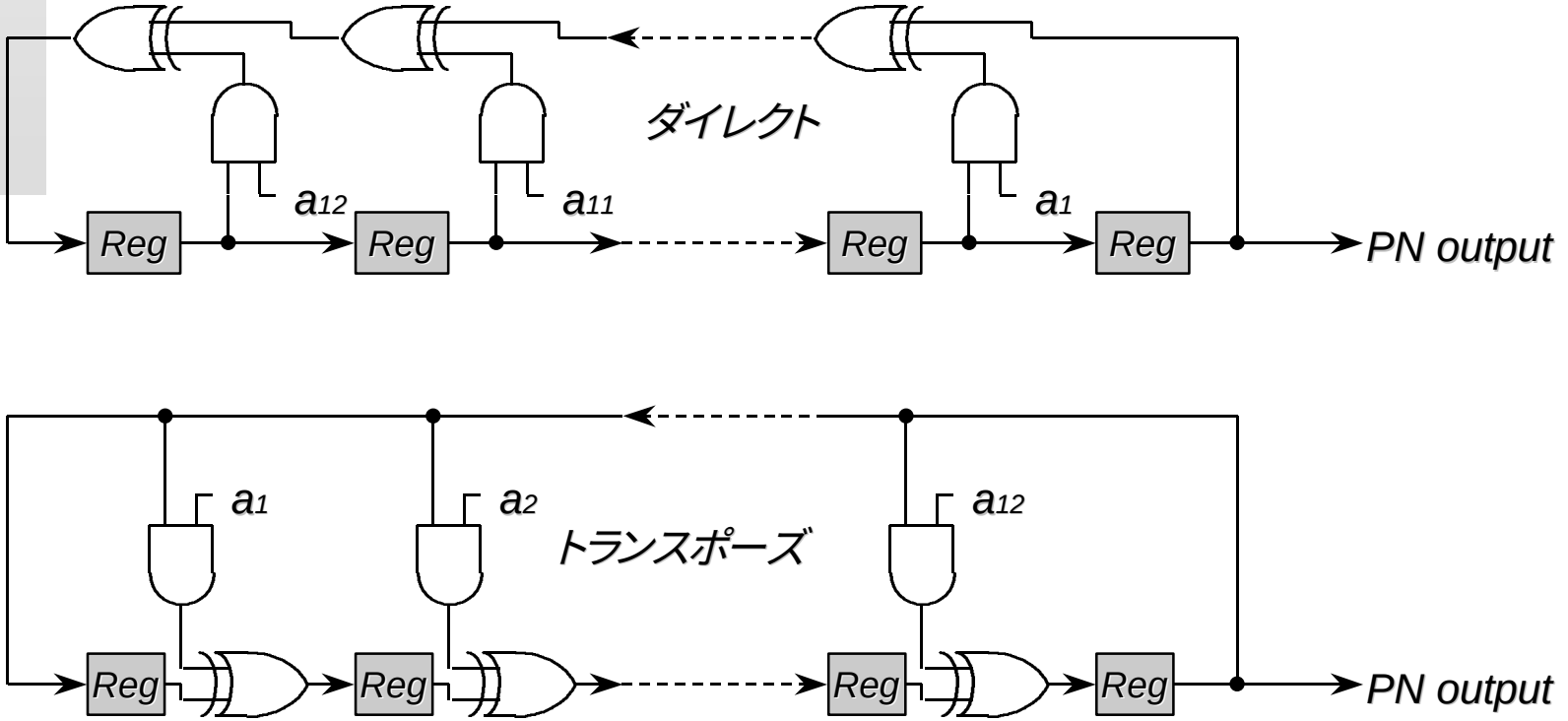
$$Rxx(\tau)=E\{x(t)\ x(t+\tau)\}$$
$$\sim \int x(t)\ x(t+\tau)dt$$



- この性質によりランダムノイズのような信号より所望の信号を取り出せる。

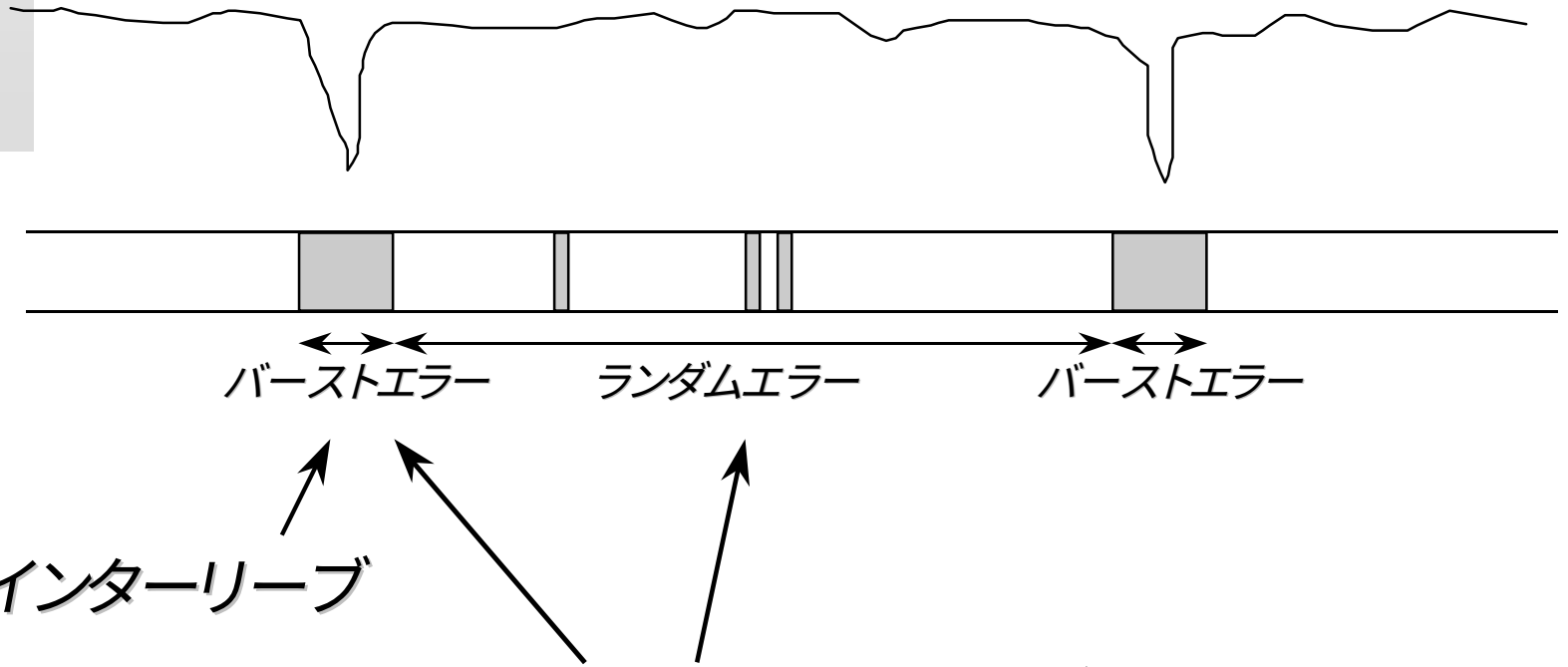
Linear Feedback Shift Register

$$f(x) = 1 + a_1X^1 + a_2X^2 + \ldots + a_{11}X^{11} + a_{12}X^{12} + X^{13}$$



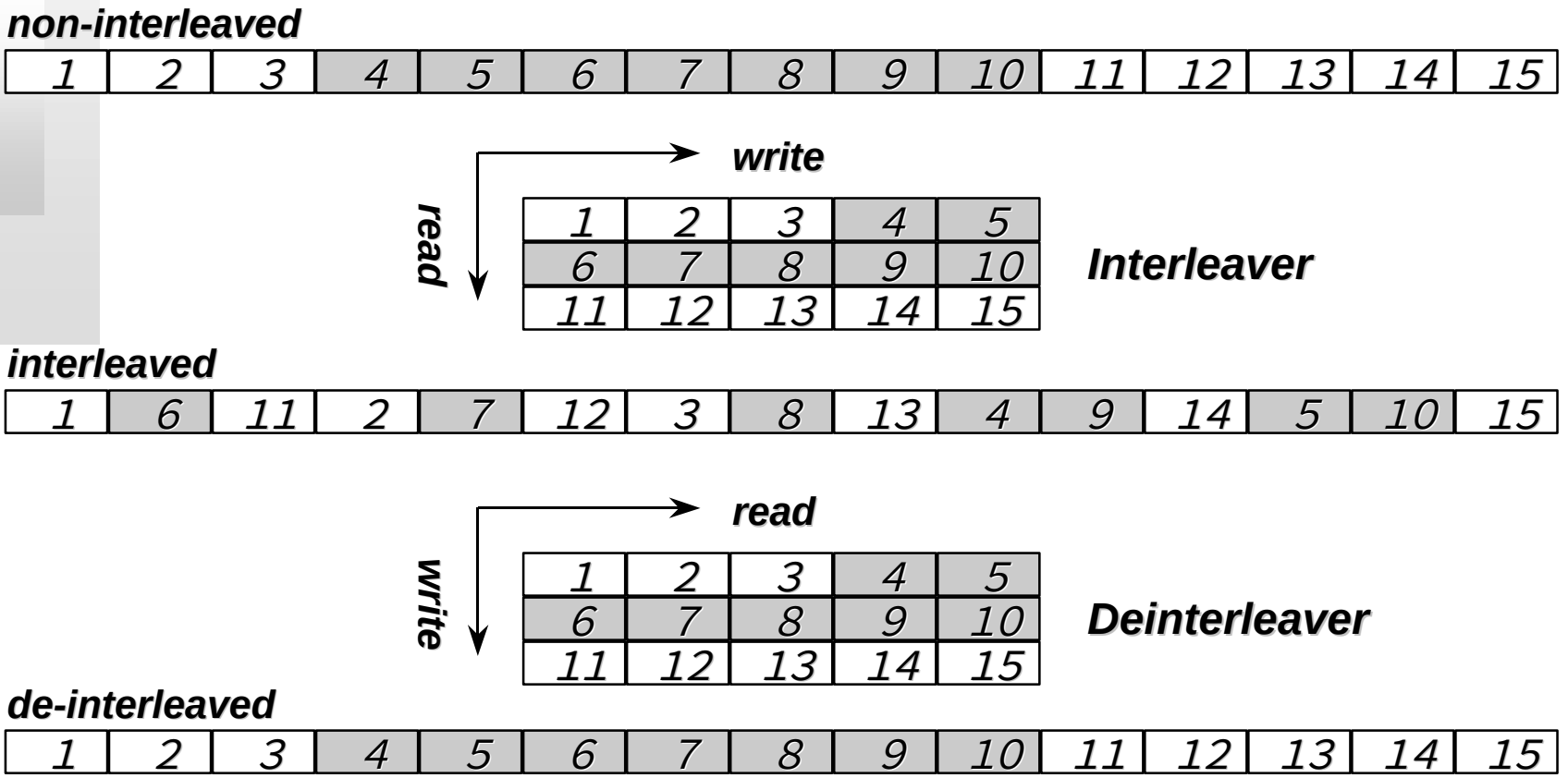
誤り訂正技術

- ランダムエラーとバーストエラー



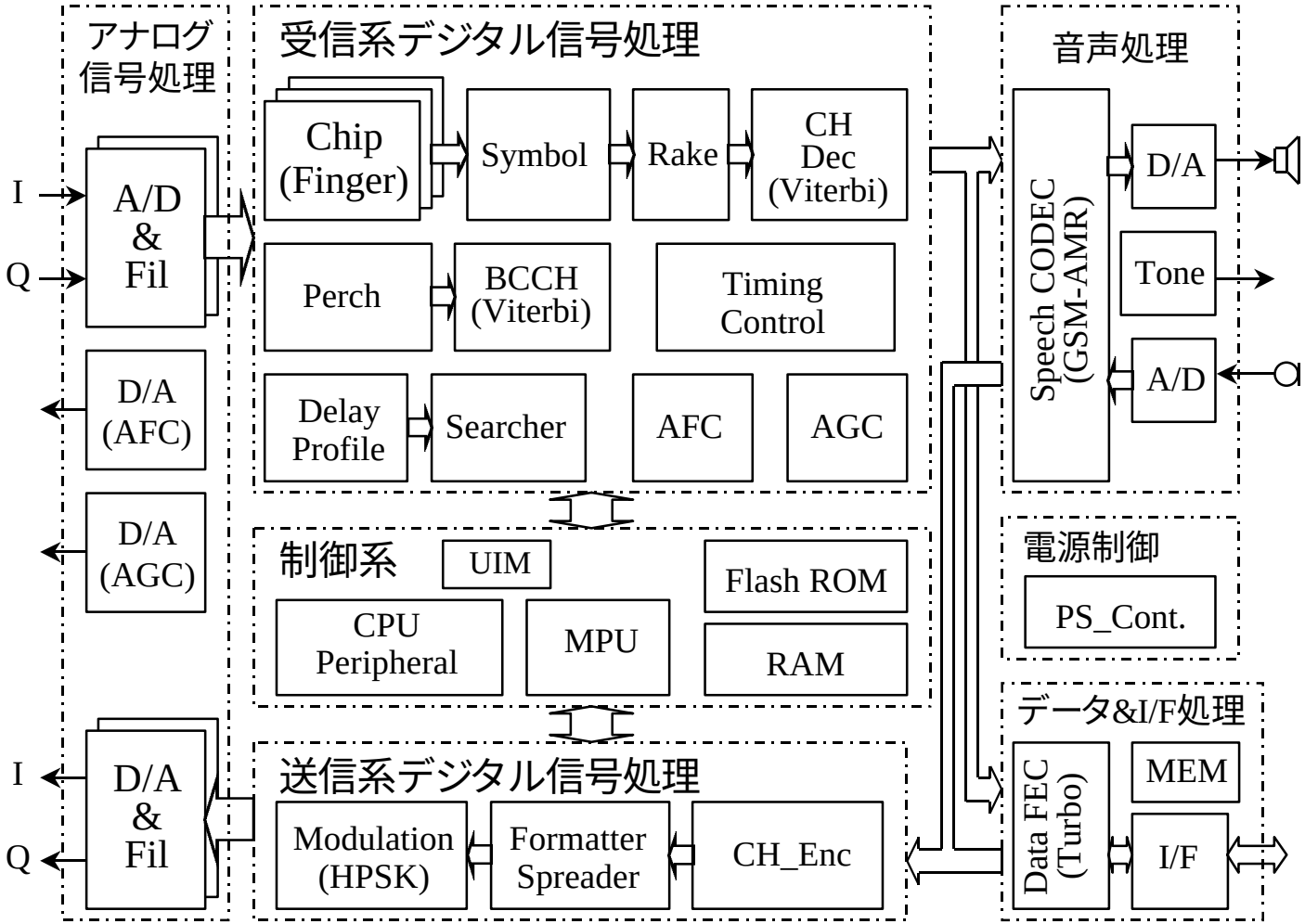
- インターリーブ
- フォワードエラーコントコレクションコード (FEC)

バーストエラー対策用インターリーブ



- 以上のようにメモリを使って実現できる。

W-CDMAのBB系ブロック図



携帯電話でのLSI技術

- 複雑なデジタル信号処理がいっぱい
- RFアナログ
- MPEG4などの画像コーデック
- USBやBlueToothなどのインターフェイス
- 低電圧・低パワー