算術演算アルゴリズムと実時間信号処理への応用

魏　書剛

研究背景

電子情報産業の拡大

- 情報化社会において、処理する情報量が激しく増大する
  高速性、信頼性、安全性
- 情報処理機器が一般の家庭電器となる
  計算機システム（PC，インターネット、デジカメ、印刷）
- 各分野および産業で電子部品の使用
  - 通信
  - 自動車
  - 医療機器
  - ほか

マイコン、信号処理プロセッサ、メモリなどの集積回路（IC）が産業の米

情報処理能力の比較

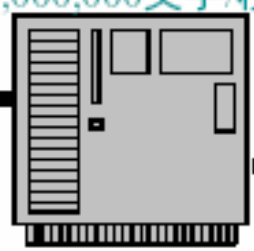
書く（100文字/分）

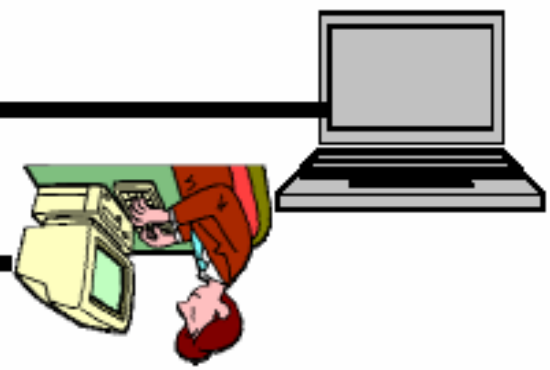
読む（1000文字/分）

話す（500文字/分）

ファクシミリ（2000文字/分）

インターネット
（1,000,000文字/秒）





専用集積回路

ソフトウェア上では間に合わないさまざまな信号処理：
- 画像処理、画像圧縮/伸長
- グラフィクス
- 暗号処理

VLSI上に実現される専用回路：
- 高速処理、IC内部の並列処理
  - 実時間の高速処理
  - プロセッサ（CPU）と専用回路での並列処理の実現
- 消費電力の低減
  - マイコンやDSPより動作周波数が低い
  - 個別ICの集積による消費電力の低減

研究内容

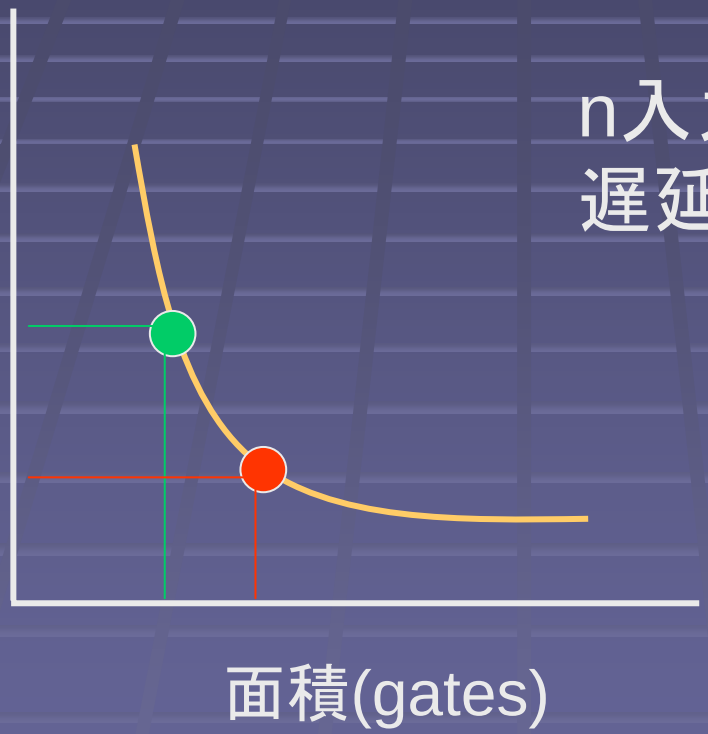
専用VLSIの実現：

- 膨大なデータを処理するため高速演算回路が必要
- ハードウェアアルゴリズムの開発

主な研究課題：

- 冗長な数系による高速演算
  - 桁上げ伝播を制限する演算方法
  - 冗長な数表現により並列剰余演算を実現
  - 従来の2進数と剰余数との高速変換回路
- 専用VLSI
  - デジタル音響レベル圧縮プロセッサ
  - FPGAによるハードウェア実現法

論理関数の合成結果(解)



n入力の論理関数の
遅延時間： O(log(n))

2進数

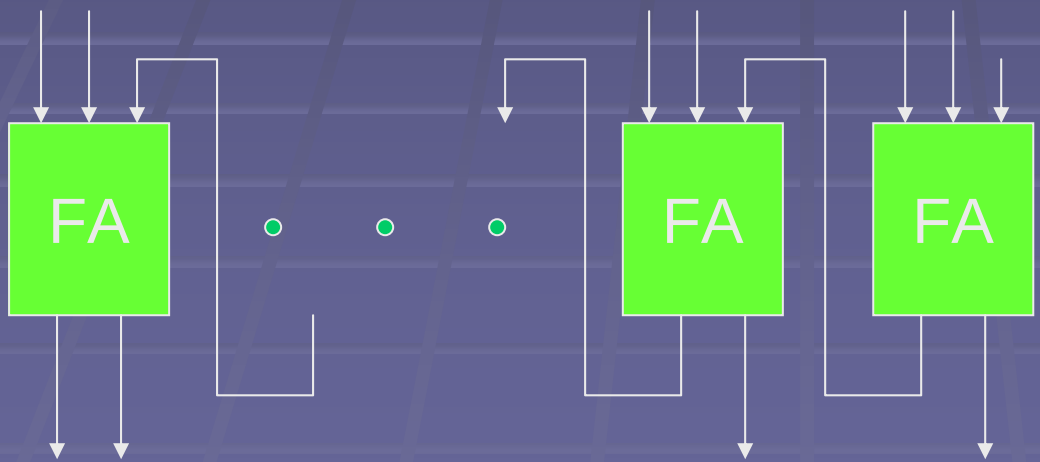
- 2値論理デバイスによるデータ表現
- 非冗長(nonredundant)
- 重み付き(weighted)
- 補数による負数の表現

殆どの数値計算システムに使用されている。

2進数の加算:桁上げ伝播

$$
\begin{array}{rl}
 & 1\,1\,0\,0\,0\,1\,0\,1 \\
+ & 0\,0\,0\,1\,1\,0\,1\,1 \\
c & 1\;1\;1\;1\;1 \\
\hline
s & 1\,1\,1\,0\,0\,0\,0\,0
\end{array}
$$

nビットの加算時間:O(n)
高速:O(log(n))



FA: 全加算器
Full Adder

冗長な2進(Signed-Digit)数表現

$$X = \sum x_i 2^i \qquad x_i \in \{\bar{1}, 0, 1\}$$

$\bar{1} = -1$

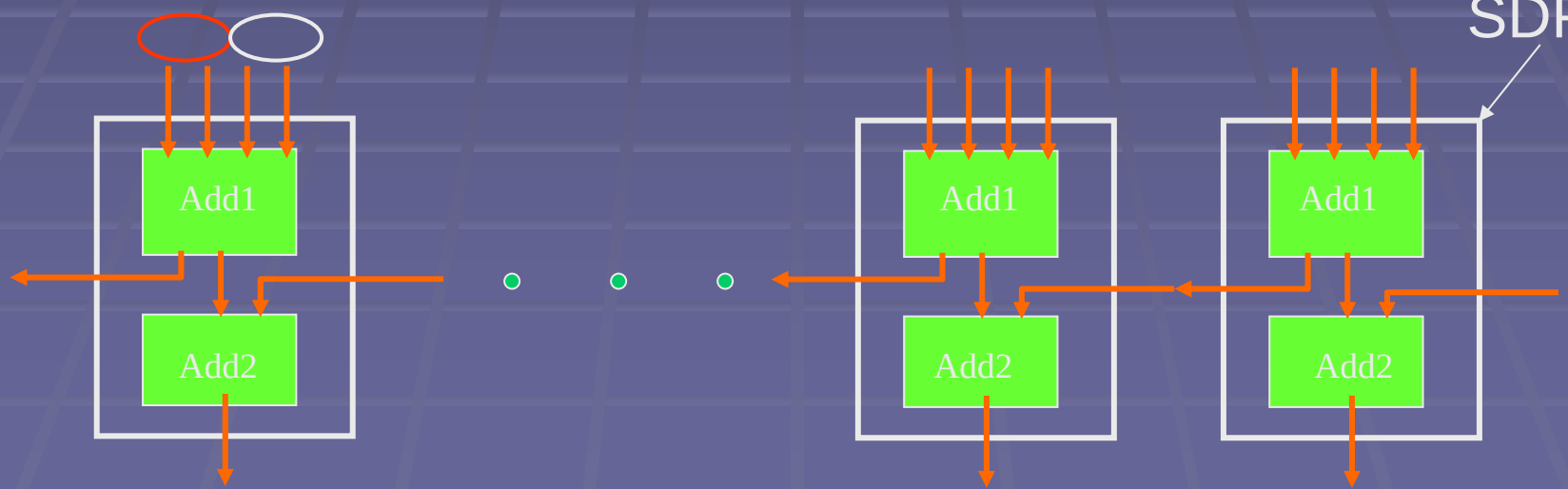
複数のSD数表現がある:

$5 = (0101) = (011\bar{1})$

数表現範囲: $[-(2^p - 1),\ 2^p - 1]$

冗長な数表現による加算（並列）

X:	1	1	0	0	0	1	0	1
+ Y:	0	0	0	1	1	0	1	1
W:	1	1	0	1	1	1	1	0
C:	1	1	0	1	1	1	1	1
S:	1	0	$\overline{1}$	1	0	0	0	0	0



加算の論理関数の比較

- 2進数関数（A,B:nビット）：語長に依存

$$C_n = f(A, B)$$

- 冗長な2進数関数：語長に依存しない

$$C_i = f(a_i, a_{i-1}, b_i, b_{i-1})$$

桁上伝搬がなくなる

剰余数系(RNS)

- 法の集合

$$\{m_1, m_2, \cdots, m_n\}$$

- 数表現

$$A = (A_1, A_2, \cdots, A_n)$$

$$A_i \in \{0,1,\cdots, m_i - 1\}$$

剰余数系の演算例

法: {3,5,7}

a = (2,4,1) = 29

b = (0,3,6) = 48

$a+b$ = (2,2,0) = 77

2進数による剰余加算

$m = 31 = (1,1,1,1,1)$

$a = 23 = (1,0,1,1,1)$

$b = 25 = (1,1,0,0,1)$

$a+b = (1,1,0,0,0,0)$

$|a+b|_{31} = (1,0,0,0,1)$

$= |48|_{31} = 17$

冗長な数表現導入の問題点

- 大きさの判定
  p桁のSD数は、正数か？
  法mより小さいか否か？
- 剰余演算方法
  p桁のSD数表現で、剰余演算はどうのように行えるのか？

冗長な剰余数表現

従来の剰余数集合

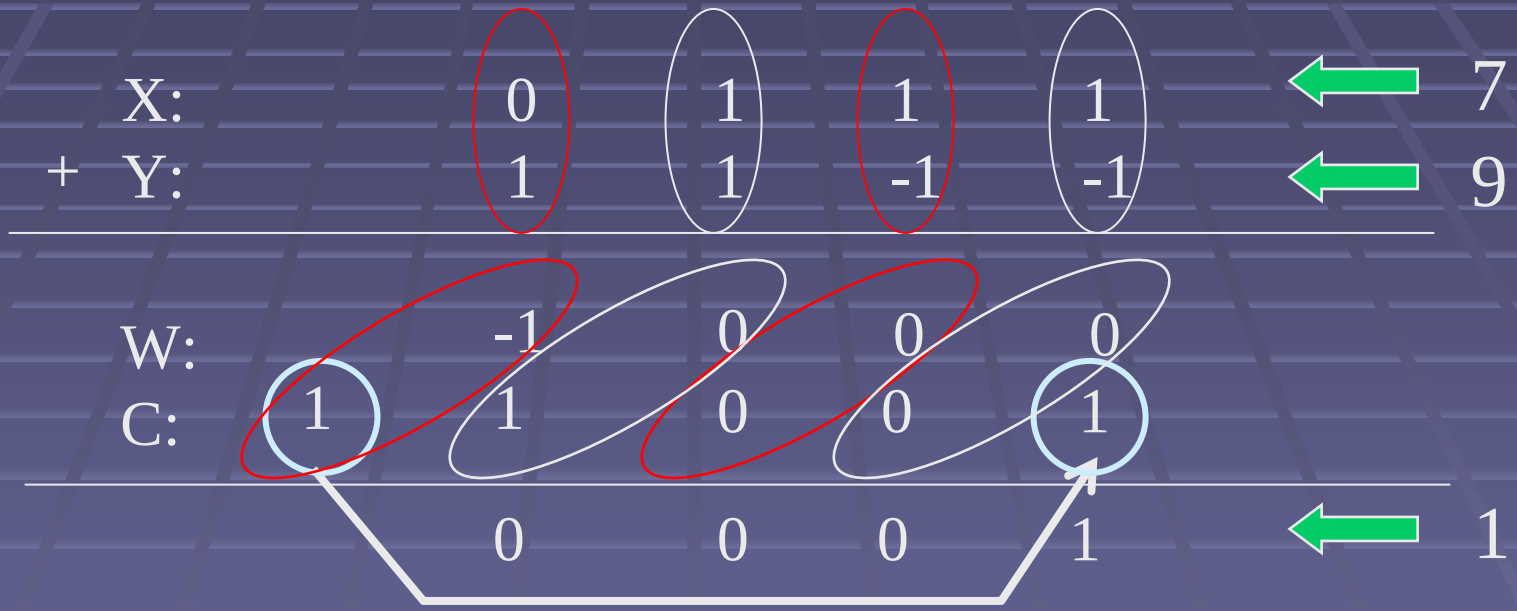
$$l_{m_i} \in \{0,1,\cdots,m_i-1\}$$

拡大の剰余数集合

$$L_m \in \{-(2^p-1),\cdots,-(m_i-1)/2,\cdots,0,$$
$$\cdots,(m_i-1)/2,\cdots,2^p-1\}$$

P桁のSD数系の数表現範囲となる

SD数の剰余加算例

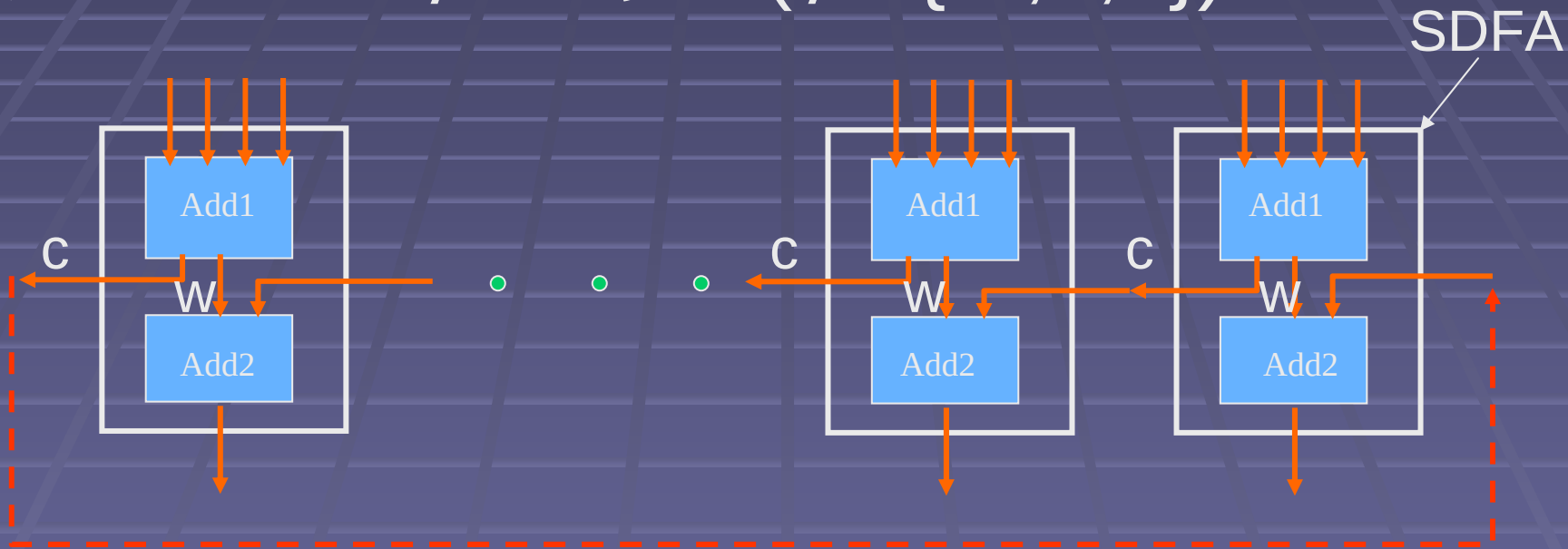


$$\langle 7+9 \rangle_{15} = 1$$

P桁の演算結果はp桁

回路構成

法　$m=2^p+\mu$ の場合($\mu \varepsilon \{-1,0,1\}$)



法　$2^p-1<m<2^p+2^p-1$の場合

2段のSD数加算で実現 – 並列加算

剰余乗算

$$\langle X \times Y \rangle_m = \langle (x_{p-1}2^{p-1} + x_{p-2}2^{p-2} + \cdots + x_0) \times (y_{p-1}2^{p-1} + y_{p-2}2^{p-2} + \cdots + y_0) \rangle_m$$

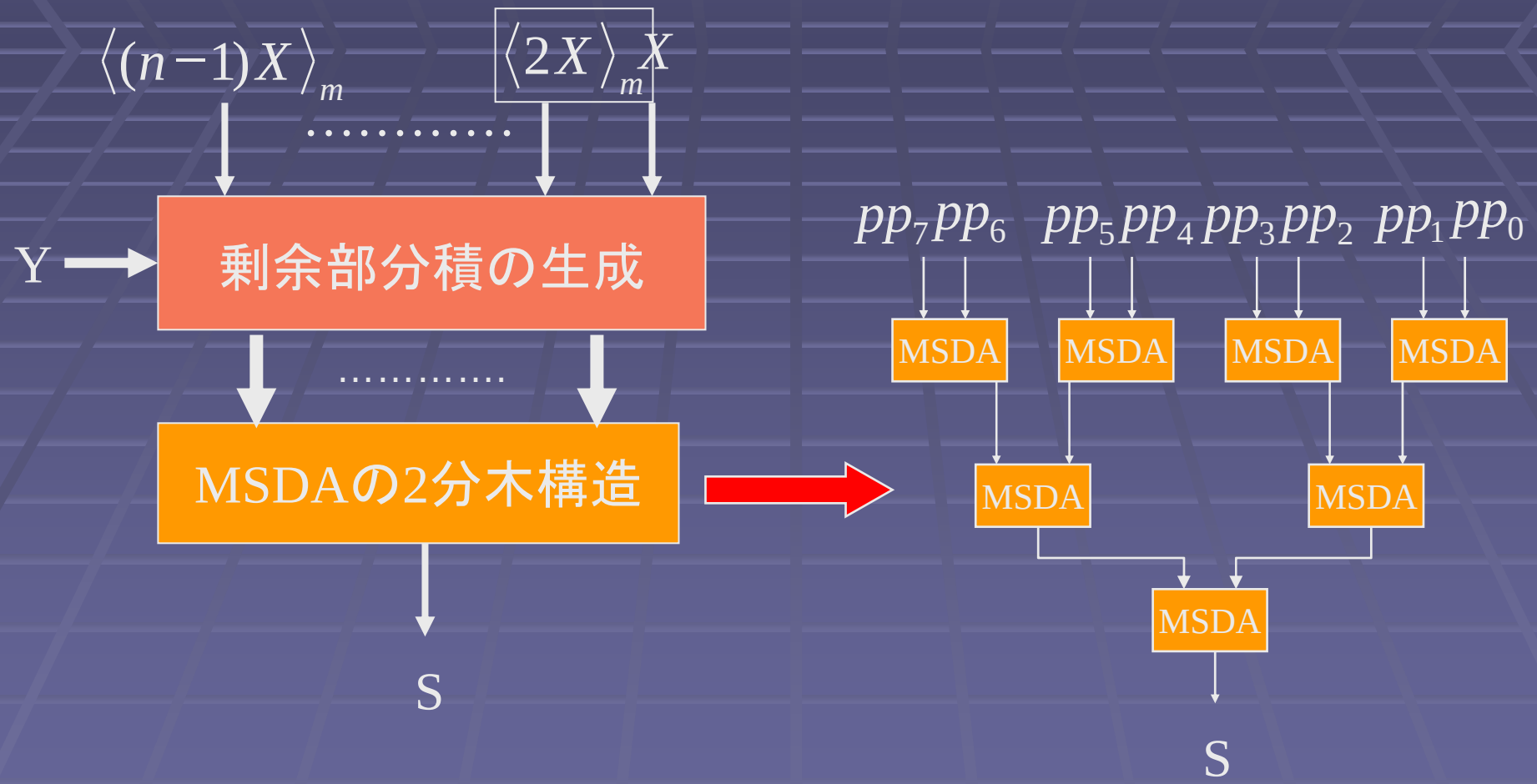
$$= \left\langle \sum_{i=0}^{p-1} \left\langle y_i \times 2^i \times (x_{p-1}2^{p-1} + x_{p-2}2^{p-2} + \cdots + x_0) \right\rangle_m \right\rangle_m$$

i 桁のシフト

$$= \left\langle \sum_{i=0}^{p-1} pp_i \right\rangle_m$$

剰余部分積

2分木構造の乗算器


$\langle (n-1)X \rangle_m$
$\langle 2X \rangle_m X$
Y
剰余部分積の生成
MSDAの2分木構造
S
$pp_7 pp_6$
$pp_5 pp_4$
$pp_3 pp_2$
$pp_1 pp_0$
MSDA
MSDA
MSDA
MSDA
MSDA
MSDA
MSDA
S

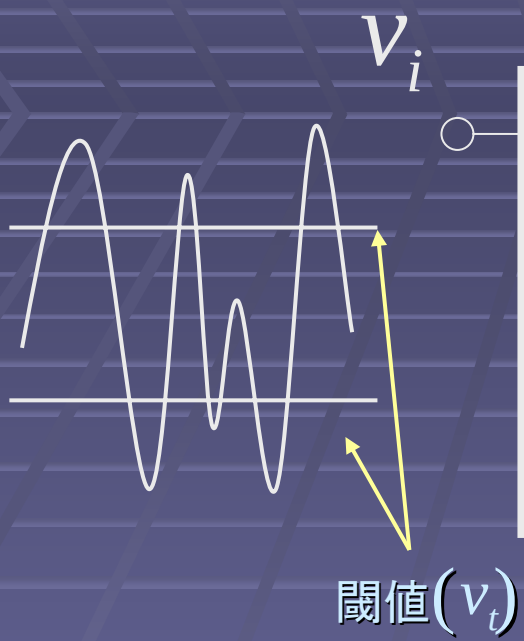
主な剰余演算の研究成果

- 剰余乗算の高速化
  - 並列剰余乗算器の構成
  - 直列剰余乗算器の構成
  - 多値回路による剰余乗算回路の実現
  - 新しいBooth方法の提案:部分積減少
- 剰余演算による算術演算の誤り検出
- 多入力の剰余加算回路
  - 3入力加算器による剰余乗算器の構成

音響レベルコンプレッサ

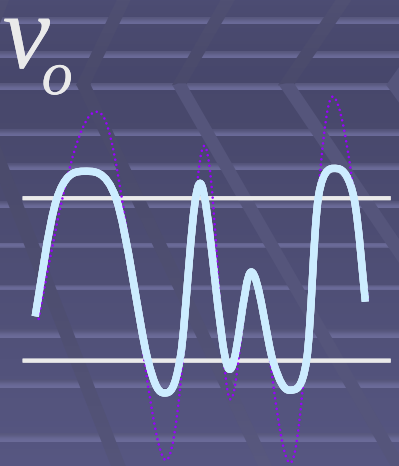
- ダイナミックレンジ制御器 (コンプレッサ/リミッタ) の役割 .
  - I. 過大信号を扱える数値に変換する
  - II. スピーカを保護する
  - III. 理想的な音響効果を得る

コンプレッサの役割



べき乗演算

$$|v_o| = \begin{cases} v_t^{(p-1)/p} \cdot |v_i|^{1/p} & (|v_i| \geq v_t) \\ |v_i| & (|v_i| < v_t) \end{cases}$$



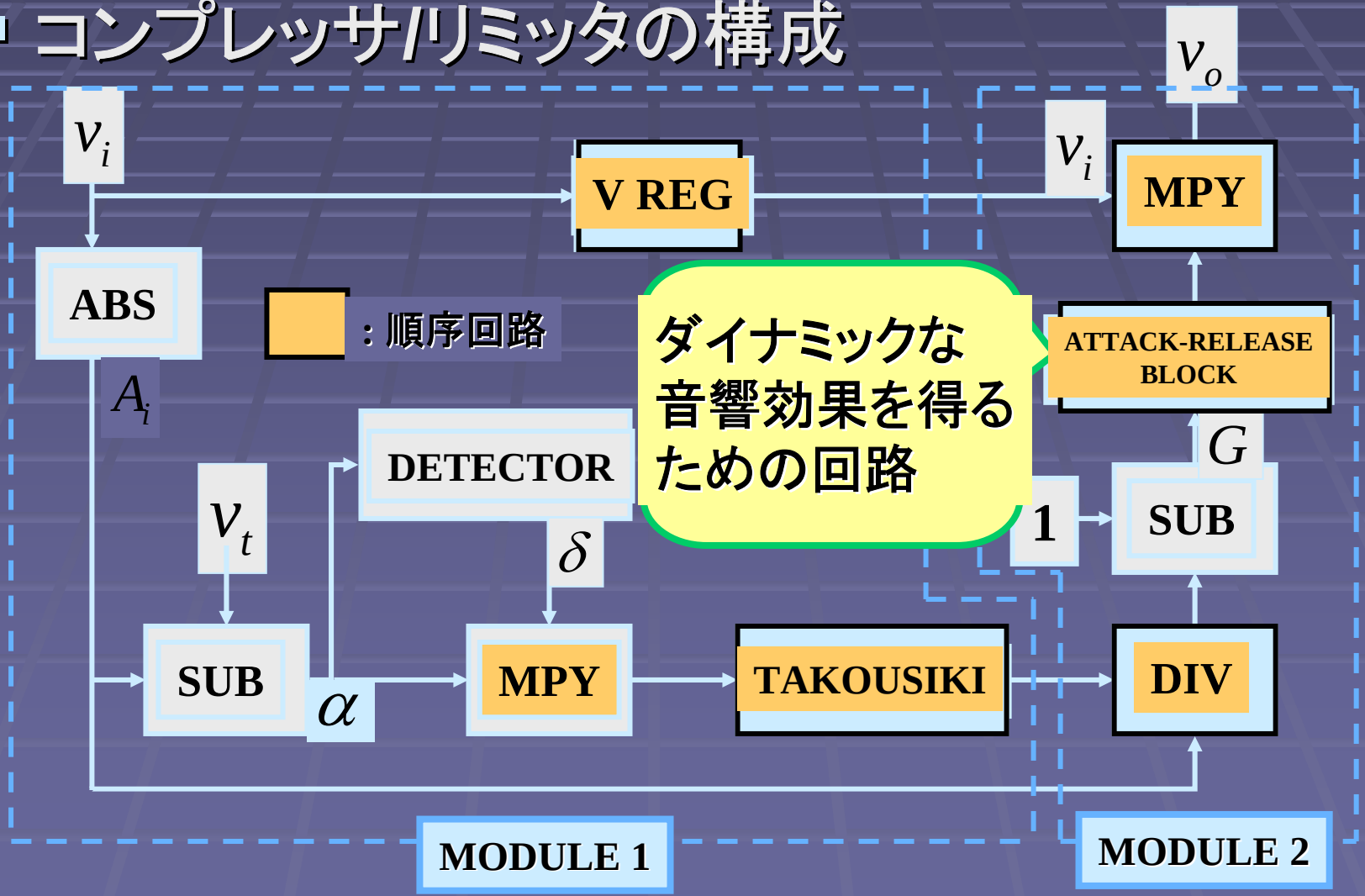
- $x^y\ (0<x, y<1)$ の演算はディジタル信号処理システムで実現することが困難

Approach1

除算や多項式演算を用いて上式を近似する

音響信号レベル圧縮処理

- コンプレッサ/リミッタの構成



大規模集積回路(LSI)設計の流れ

機能設計
(HDLコード)

シミュレーション
（機能検証）

論理合成

配置配線

シミュレーション
（タイミング検証）

製造

ASIC
半導体メーカ
の仕事

魏研究室

魏研究室、FPGA

大規模集積回路(LSI)設計の流れ

機能設計
(HDLコード)

シミュレーション
（機能検証）

論理合成

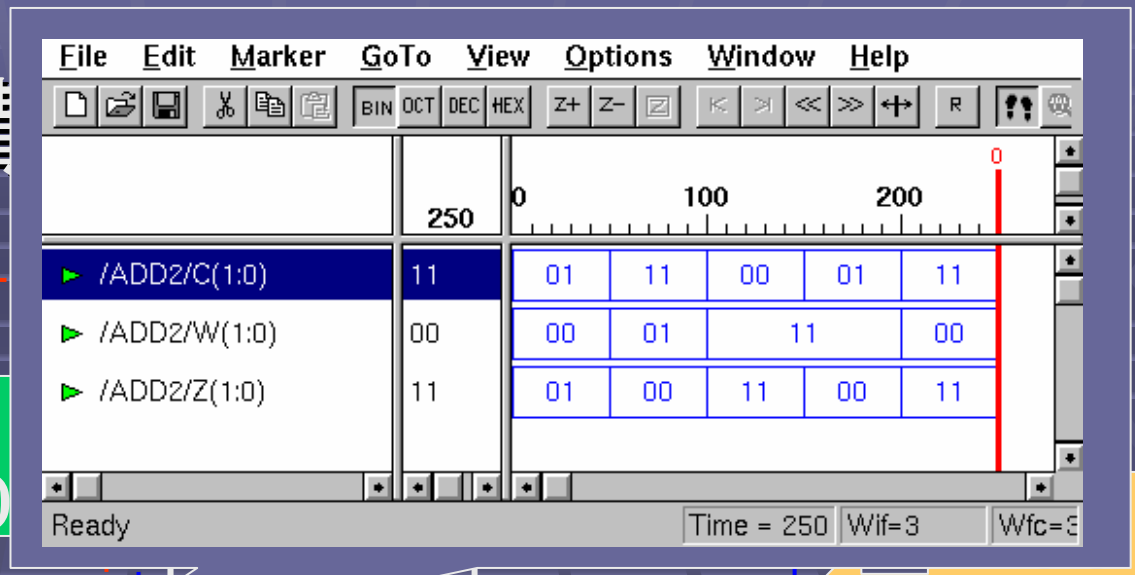
魏研究室

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity ADD2 is
 port(
       c,w:in std_logic_vector(1 downto 0);
       z:out std_logic_vector(1 downto 0));
end ADD2;

architecture RTL of ADD2 is
begin
 process(c,w) begin
  z <= c+w;
 end process;
end RTL;
```

-力

魏研究室、FPGA

大規模集積



機能設計
(HDLコード)

シミュレーション
（機能検証）

論理合成

シミュレーション
（タイミング検証）

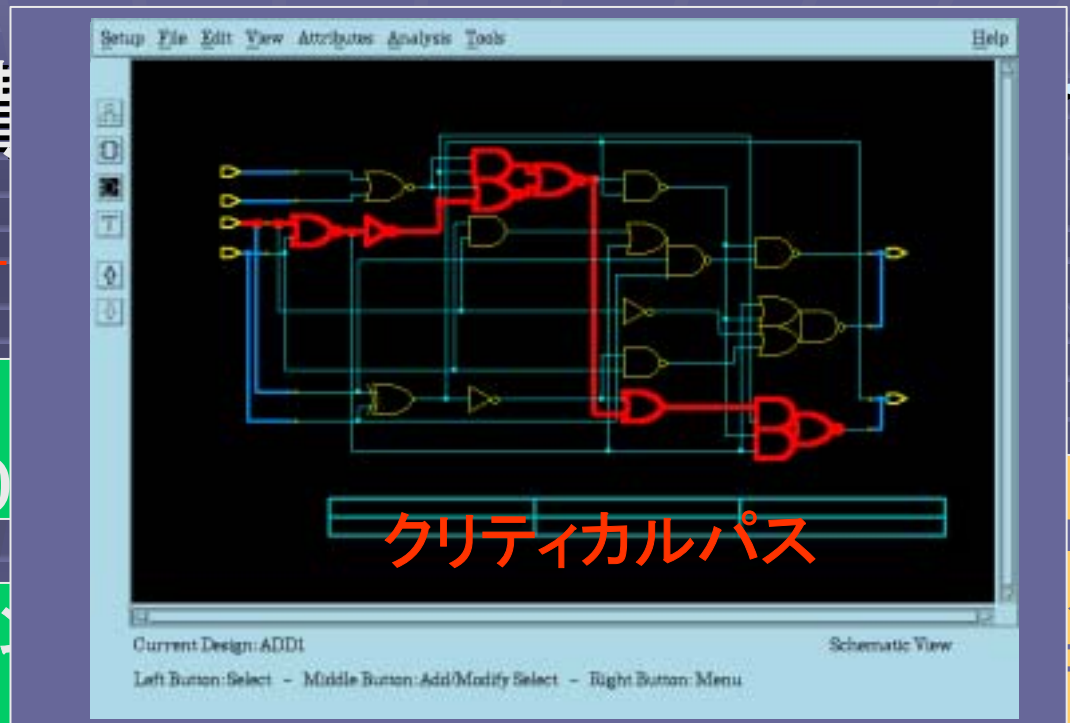
製造

半導体メーカ
の仕事

魏研究室

魏研究室、FPGA

大規模集積



機能設計
(HDLコード)

シミュレーション
（機能検証）

論理合成

製造

IC
メーカ
事

魏研究室

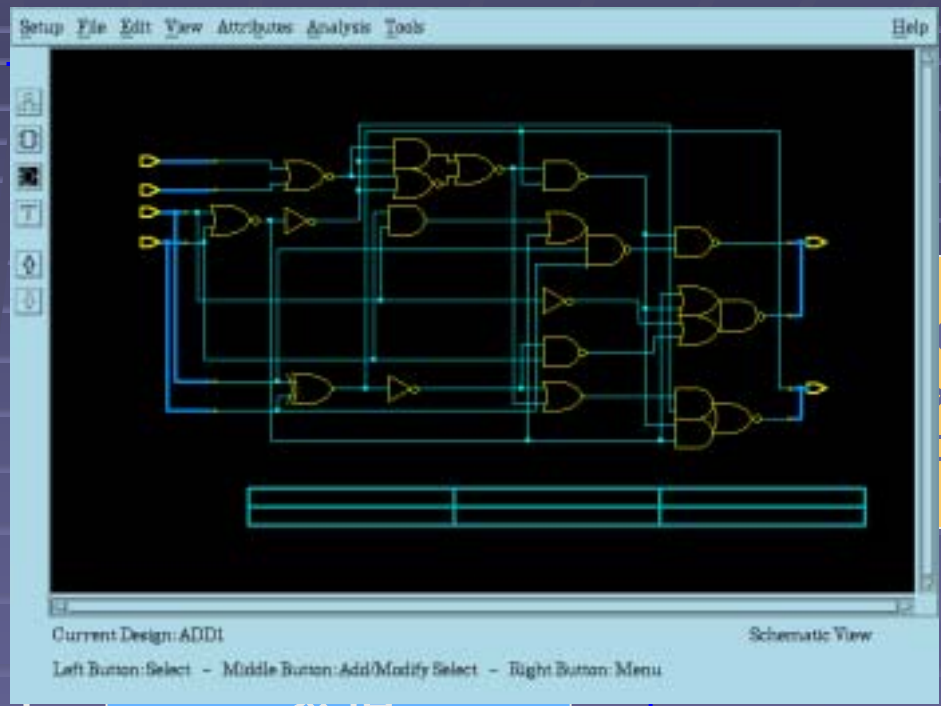
魏研究室、FPGA

大規模集積回路(LSI)設計の流れ

機能設計
(HDLコード)

シミュレーション
（機能検証）

論理合成



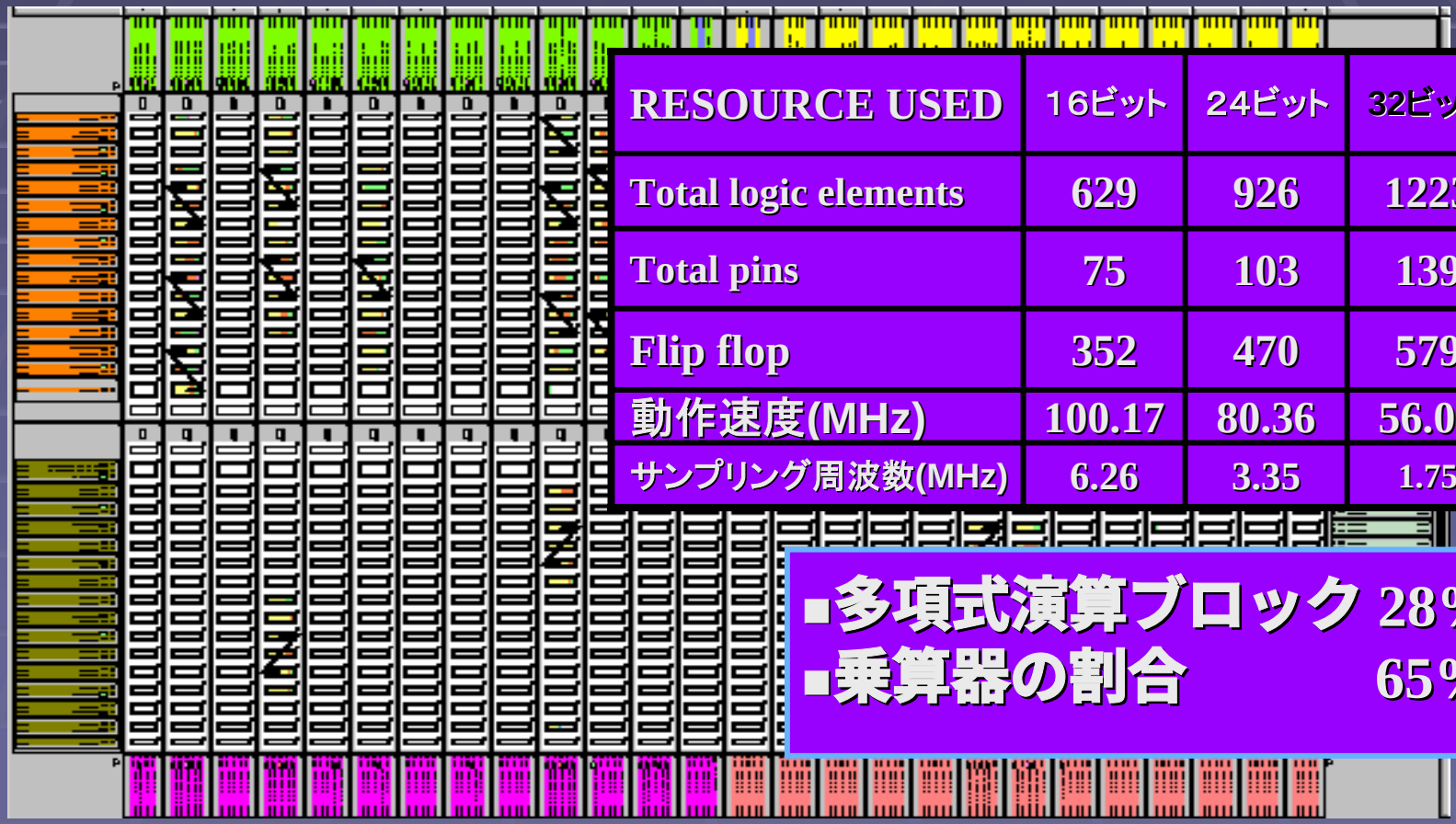
C
メーカ
事

魏研究室

魏研究室、FPGA

大規模集積回路（VLSI）の設計

- 32ビットコンプレッサのFPGA実現例



RESOURCE USED	16ビット	24ビット	32ビット
Total logic elements	629	926	1223
Total pins	75	103	139
Flip flop	352	470	579
動作速度(MHz)	100.17	80.36	56.08
サンプリング周波数(MHz)	6.26	3.35	1.75

- 多項式演算ブロック 28％
- 乗算器の割合 65％