

システムLSIのシステムデザインの現状と問題点、今後の展望について：コストモデルの観点から

安浦，寛人
九州大学大学院システム情報科学研究院 | 九州大学システムLSI研究センター

<https://hdl.handle.net/2324/9159>

出版情報：SLRC プレゼンテーション，2007-04-06. 九州大学システムLSI研究センター
バージョン：
権利関係：



システムLSIのシステムデザインの現状と問 題点、今後の展望について ーコストモデルの観点からー

安浦寛人

九州大学システムLSI研究センター

システムデザインの問題点

- システムの複雑化による設計コストの高騰
 - ◆ マスク費用（微細化による高騰）
 - ◆ 個別設計部分の設計コスト
 - ◆ ソフトウェア（並列化による高騰）
- 汎用品（Commodity）の利用との競争
 - ◆ マイクロプロセッサ＋メモリ＋SW
 - ◆ FPGAなどの再構成可能デバイス

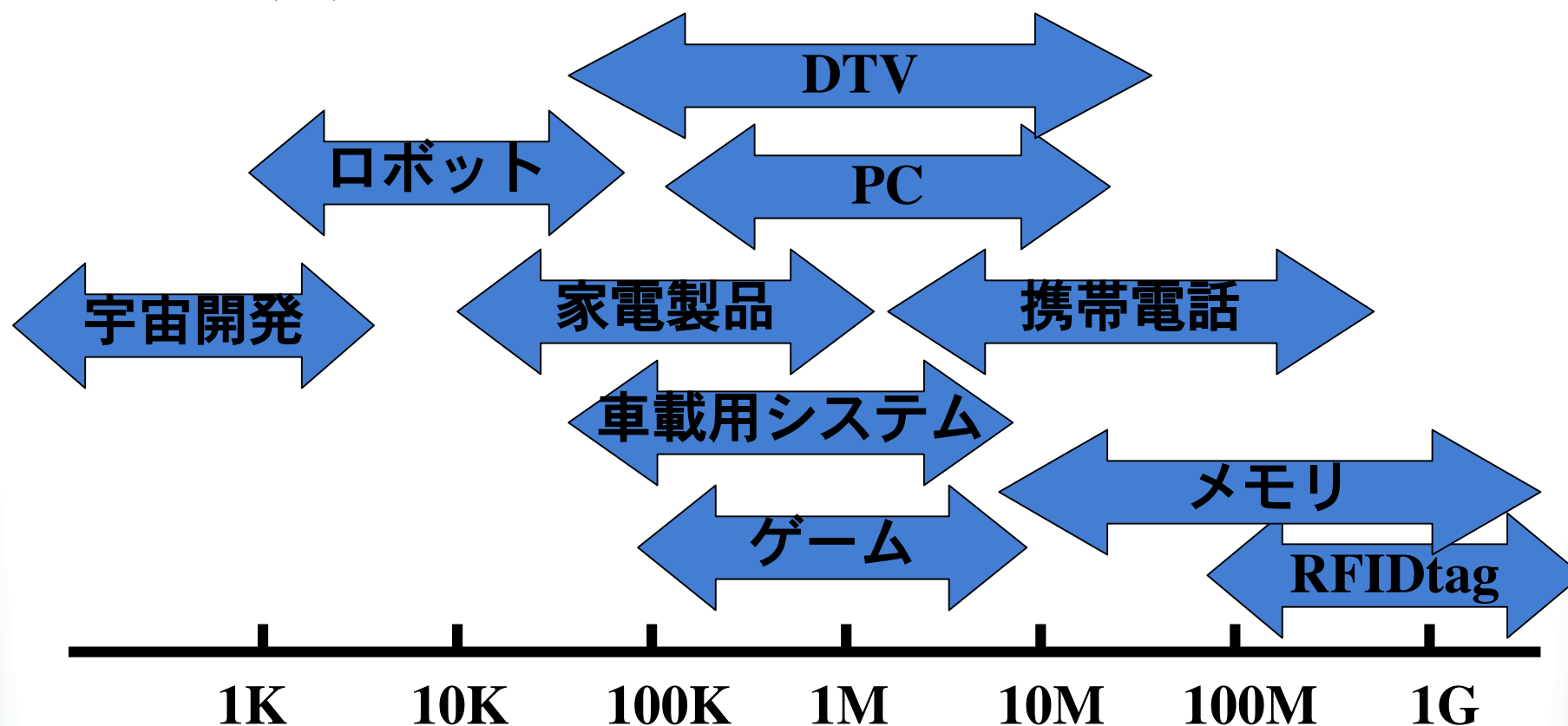
問題のポイント

- 設計コストは，製品に転化されるが，製品1個あたりのコストは製品数に反比例する．
- 同じプロセス技術を利用し，製品数を100万個と仮定したとき
 - ◆ ASIC的アプローチ
 - Chip設計費・マスク費用は全額100万個で回収しなければならない
 - ソフトウェアも個別開発が必要
 - ◆ FPGA的アプローチ
 - Chip設計費・マスク費用はFPGAベンダが他製品と共有して回収する
 - 個別回路の設計分のみ100万個で回収する必要がある
 - ソフトウェアは個別開発が必要
 - ◆ プロセッサ的アプローチ
 - Chip設計費とマスク費用はプロセッサベンダやメモリベンダが膨大な数の他製品と共有して回収する
 - ソフトウェアも基本ソフト部分は他製品と共有で回収が可能
 - 固有部分のソフトウェア開発費のみ100万個で回収が必要

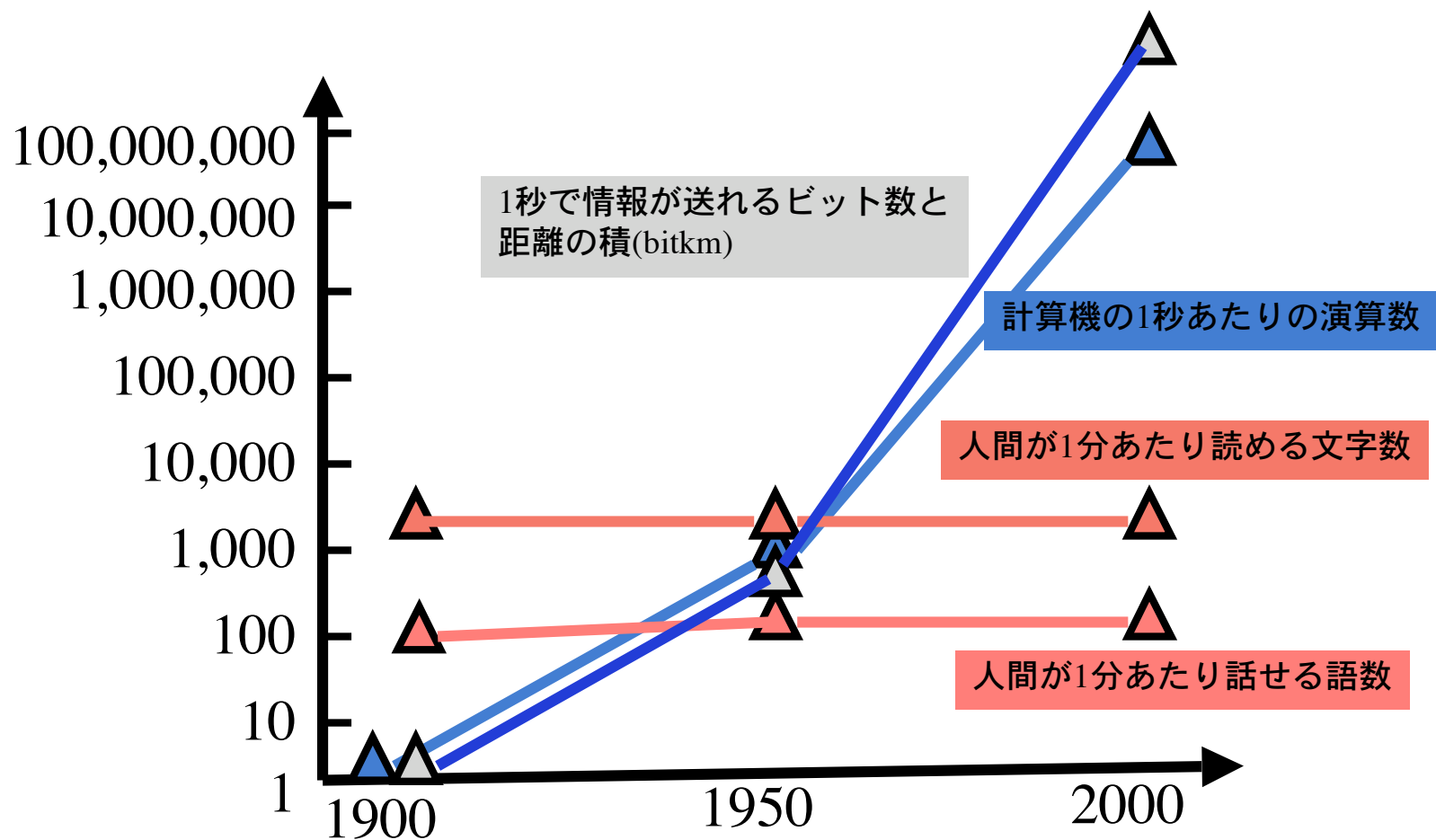
複数のSolutionの必要性



- 製品によって、製品の個数（市場）は大きく異なる



情報の通信・処理の変化



基本的な問題

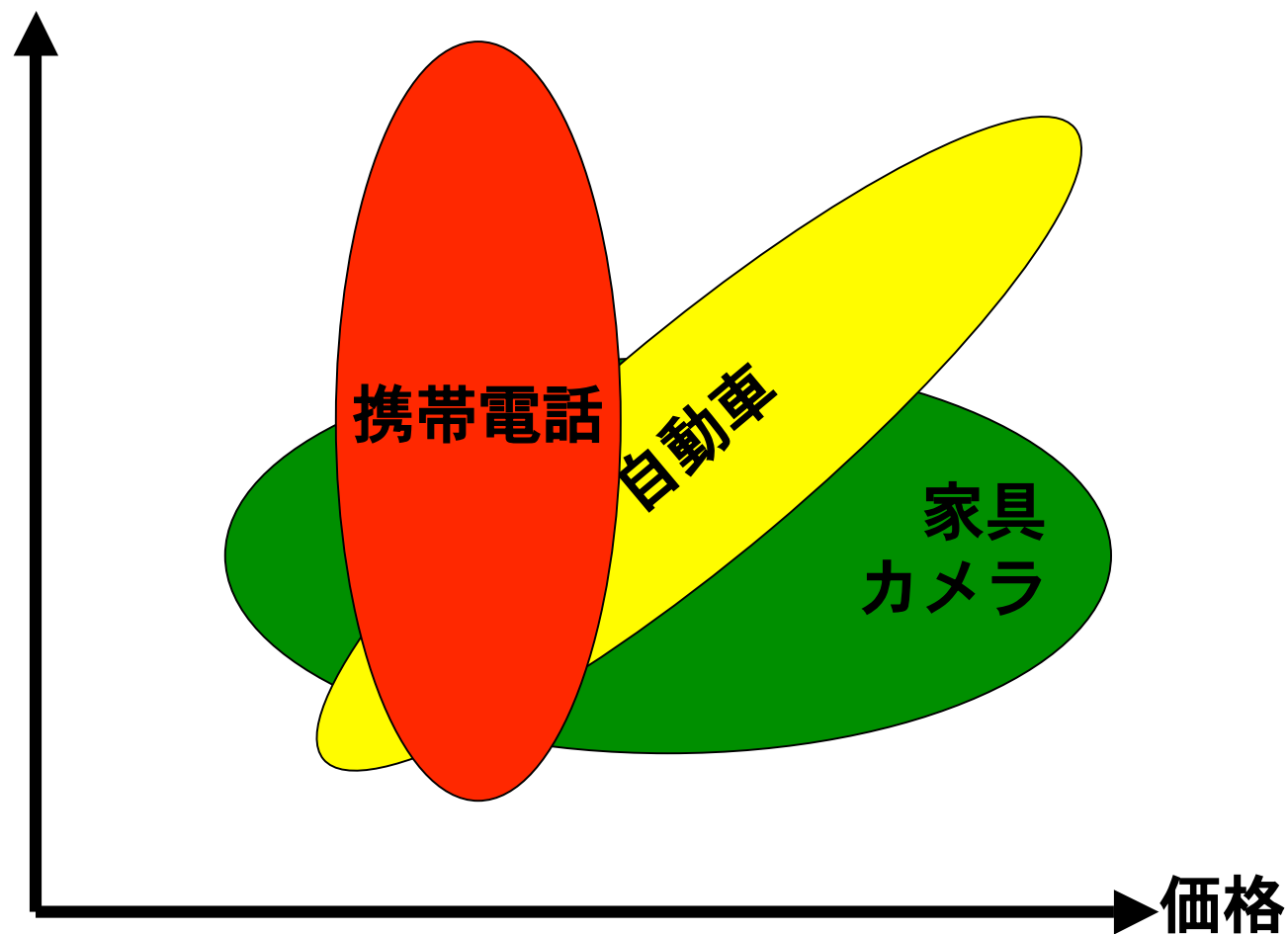
- 製品の市場価値を決める決定的な要素は？
 - ◆ 価格
 - ◆ 機能
 - ◆ 性能
 - ◆ 消費電力
 - ◆ 信頼性
- 携帯電話で20%の消費電力の削減と20%の価格削減はどちらが魅力的か？
- 製品市場の成熟度によっても市場価値を決定する要素は異なる

市場と製品の特性

- 製品の市場価値を決める決定的な要素
 - ◆ 価格：コモデティ化した大衆向け製品
 - ◆ 機能：プロフェッショナル向け、高級品、新規市場向け製品
 - ◆ 性能：性能要求が満たされていない分野、プロフェッショナル向け製品
 - ◆ 消費電力：電力供給がコスト要因となる製品（RFICタグ、ICカード、ペースメーカーなど）、電池駆動の携帯機器
 - ◆ 信頼性：人命や財産に関わる製品（自動車、電子マネー）
- 市場価値の地域的な違い
 - ◆ 成熟した先進国：米国、日本、西欧など（約6億人）→機能、信頼性
 - ◆ 急成長をしている国々：中国、インド、ロシア、韓国、台湾、東ヨーロッパ、ブラジル、メキシコ、東南アジアなど（約30億人）→性能、価格
 - ◆ 成長から取り残された国々：アフリカ諸国、アラブ諸国、アジアの一部、南米の一部（約30億人）→価格

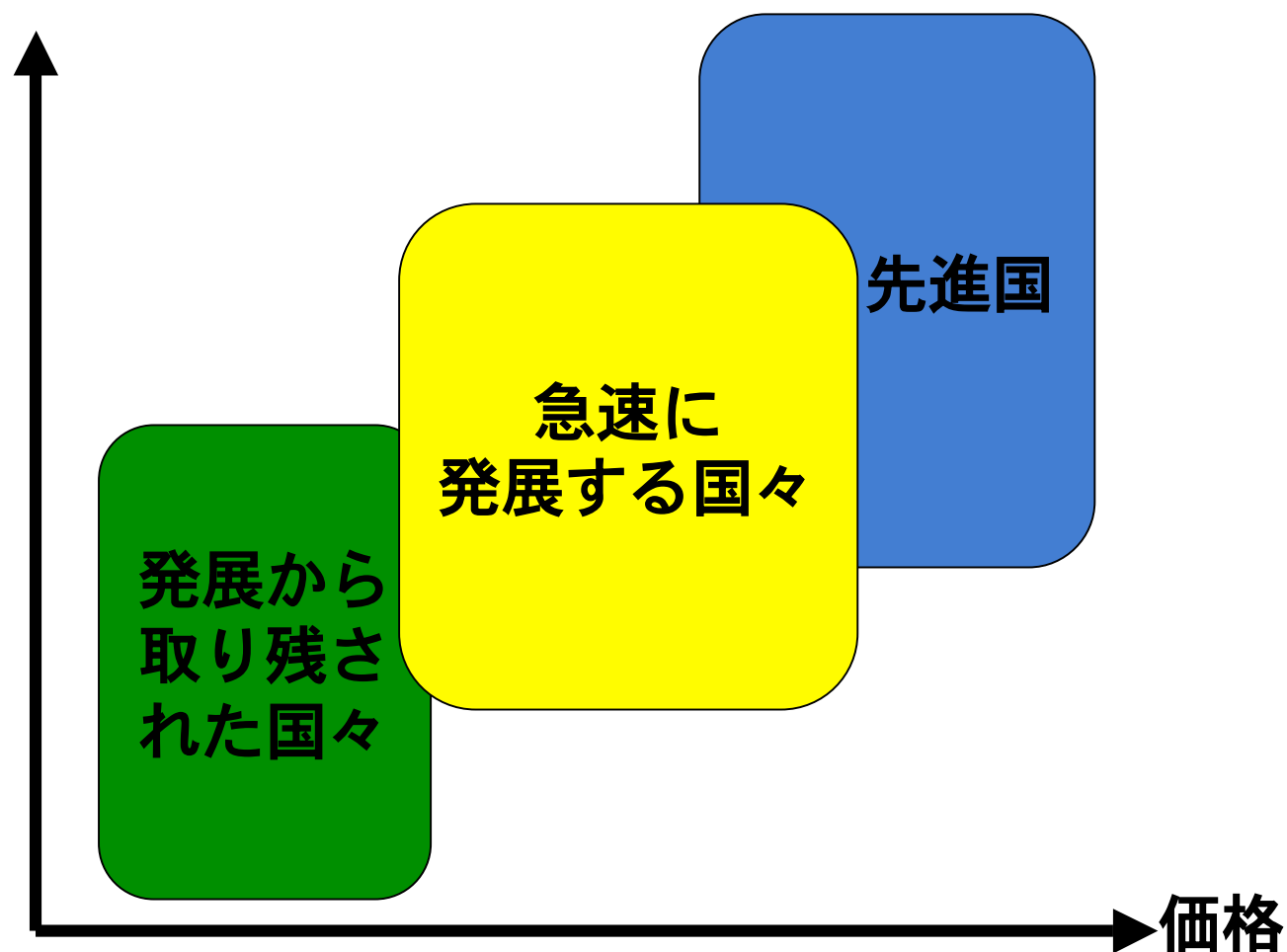
価格と他の尺度の関係

性能・機能・
信頼性など

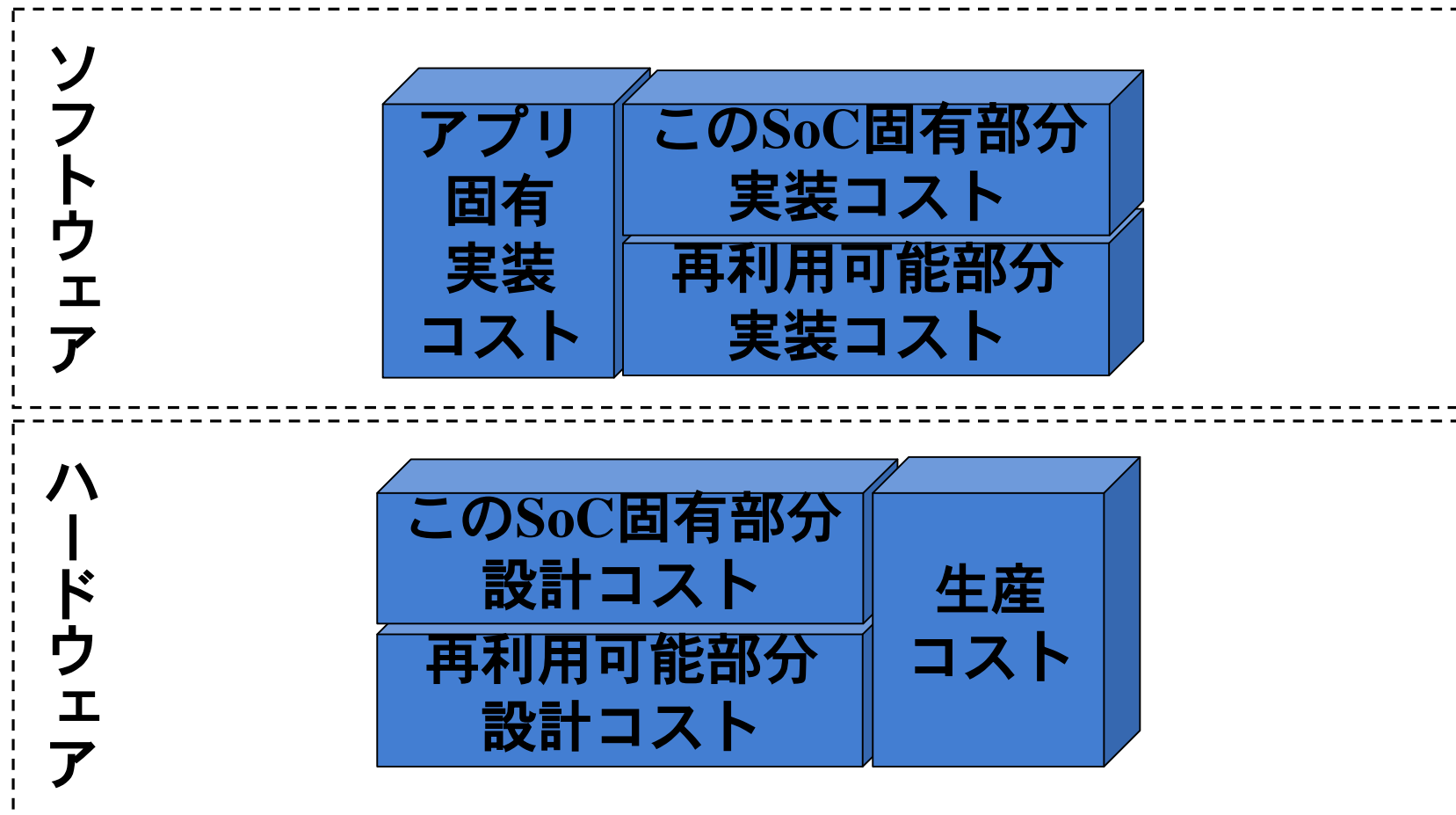


地域による市場の違い

性能・機能・
信頼性など



SoCのコスト



ソフトウェアを含めたSoCコスト



ソフトウェア込みの
チップあたりのコスト

チップあたりの
ハードウェアコスト

チップあたりの
ソフトウェアコスト

$$CPC_{SoC_{ij}} = CPC_{hw_i} + CPC_{sw_{ij}}$$

ソフトウェア
バリエーション数

$$N_{good_i} = \sum_j N_{good_{ij}}$$

ハードウェアiの
ソフトウェア
バリエーションjの
出荷個数

ハードウェア
バリエーション数

$$N_{good} = \sum_i N_{good_i}$$

ハードウェア
バリエーションjの
出荷個数

Ni, Njの増加で
チップあたりの
ソフトウェア単価
設計単価・製造単価
が小さくなる

ハードウェアコスト

$$CPC_{hw_i} = \frac{COST_{design_i}}{N_{good_i}} + \frac{COST_{design_{common}}}{N_{good}} + \frac{mCOST_{mask_i}}{N_{good_i}} + \frac{COST_{fab_i}}{YN_{wafer_i}} + TEST_{chip_i} + Pack_i + IP_i$$

CPC_{hw_i} : Chip Cost

$COST_{design_i}$: Design Cost

N_{wafer} : # of Chips on a Wafer

$TEST_{chip_i}$: Test Cost per Chip

$COST_{design_{common}}$: Design Cost for Reusable Components

$COST_{mask_i}$: Mask Cost

m : # of Original Masks

N_{good_i} : # of Shipped Chips

$Pack_i$: Package Cost per Chip

N_{good} : # of Shipped Chips using Reusable Components

$COST_{fab_i}$: Fabrication Cost per Wafer

Y : Yield

IP_i : Intellectual Property (Designed Block)

$$Y = (1 - q)^{A_{chip}}$$

A_{chip} : Area of Chip

$$N_{wafer} = A_{wafer} / A_{chip}$$

A_{wafer} : Area of Wafer

$$TEST_{chip} = TEST_{design} / N_{good} + (V_T * T_{test}) / Y$$

設計とマスクのコスト

$$\frac{COST_{design_i}}{N_{good_i}} + \frac{COST_{design_{common}}}{N_{good}} + \frac{mCOST_{mask_i}}{N_{good_i}}$$

$COST_{design_i}$ あるSoC iの設計コスト

$COST_{design_{common}}$ 複数のSoCで共通の設計コスト

設計者の人件費、設計用計算機、自動設計ツール、
部屋代、管理費など

m : 固有のマスクの費用

20 ~ 30枚(金属層数や技術に依存)

$COST_{mask_i}$: マスクコスト

¥500,000 (0.25 μ m) > ¥5,000,000 (0.09 μ m)

N_{good} : 設計を再利用した最終製品のチップ数

N_{good_i} : あるSoC iの最終製品のチップ数

生産するチップ数 - 不良チップ数

赤字は設計者が決定できるパラメータ

製造コスト



COSTfab / (YNwafer)

COSTfab: ウェハ1枚の製造コスト

ウェハ材料費、工場設備投資、ガスや化学薬品、工場の運営費

工場設備投資: ¥100,000,000,000

Y: Yield (歩留り)

$$\frac{(1 \text{ ウェハ上のチップ数} - \text{不良チップ数})}{1 \text{ ウェハ上のチップ数}}$$

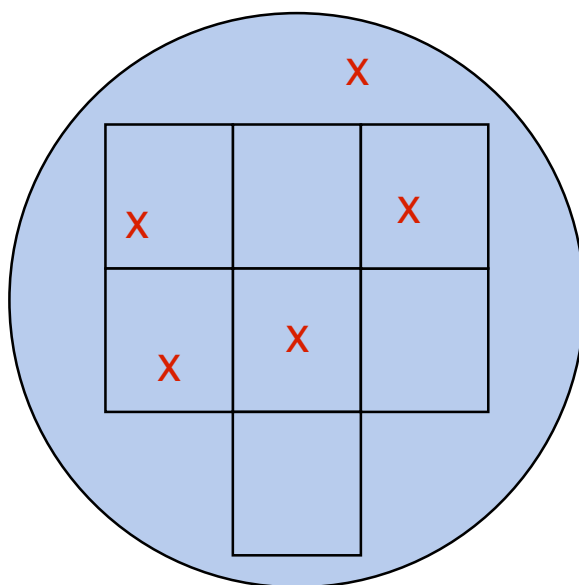
$$Y = (1 - q)^{A_{\text{chip}}} \quad q: \text{単位面積あたりの欠陥確率}$$

Nwafer: 1 ウェハ上のチップ数

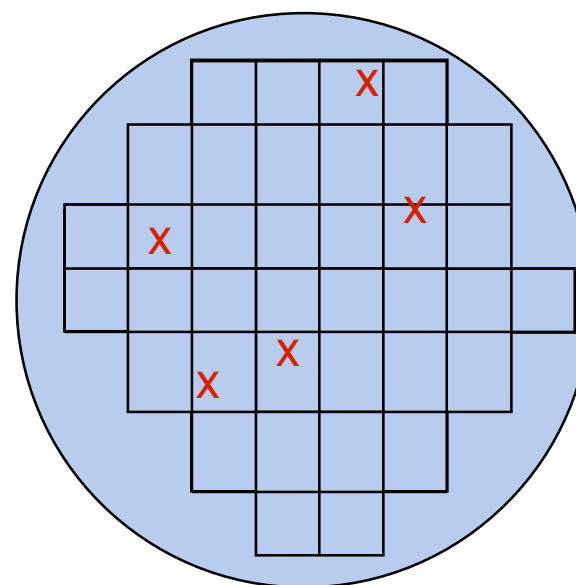
$$A_{\text{wafer}} / A_{\text{chip}}$$

$$A_{\text{wafer}} = \pi r^2, \quad 2r = 3, 5, 8, 12 \text{ inches}$$

チップ面積とコスト



チップ数7、良品3
歩留り42%



チップ数37、良品32
歩留り86%

$$Y = (1 - q)^{A_{chip}}$$
$$N_{wafer} = A_{wafer} / A_{chip}$$

テストコスト

$$TEST_{chip_i} = \frac{TEST_{design_{common}}}{N_{good}} + \frac{TEST_{design_i}}{N_{good_i}} + \frac{V_T T_{test}}{Y}$$

$TEST_{design_{common}}$: 設計を再利用したテスト設計のコスト

$TEST_{design_i}$: あるSoC i 固有のテスト設計のコスト

テスト設計者の人件費、計算機、テスト生成ツール

V_T : 1秒当たりのテスト利用コスト(償却費)

$V_T = \text{Tester Cost} / \text{Total Active Time (sec)}$

T_{test} : テスト時間(sec)

テスト時間を短くする努力

Reliability: テストによる見逃しを最小化する要請

Package and IP Cost



Package Cost + IP_i

パッケージコストは、パッケージの材質とピン数に依存する

ピン数: 入出力信号、Vdd と GND

パッケージの外との信号伝達 (DIP, PGA, BGA, QFP など)

材質: 安いプラスチック、高価なセラミック

発熱による温度によって選択肢が決まる

IP (Intellectual Property) コスト

IP (設計された部品) の購入費やライセンス費

搭載ソフトウェアの購入費やライセンス費

チップコストの概算

単位：円

$$\begin{aligned}
 CPC_{hw_i} = & \frac{10^6 \sim 10^9}{10^5 \sim 10^8} \frac{COST_{design_i}}{N_{good_i}} + \frac{10^6 \sim 10^9}{10^5 \sim 10^{10}} \frac{COST_{design_{common}}}{N_{good}} \\
 & + \frac{10^5 \sim 10^7}{N_{good_i}} mCOST_{mask_i} \\
 & + \frac{10^4 \sim 10^7}{YN_{wafer_i}} COST_{fab_i} + \frac{10 \sim 10^5}{10^2 \sim 10^4} TEST_{chip_i} + \frac{10 \sim 10^4}{10 \sim 10^3} Pack_i + IP_i
 \end{aligned}$$

チップコストの例

$$\text{COSTdesign}_i = 50,000,000\text{円} \quad m = 30\text{枚}$$

$$\text{COSTdesign} = 200,000,000\text{円}$$

$$\text{COSTmask} = 500,000\text{円}$$

$$\text{COSTfab} = 200,000\text{円} \quad Y = 0.8$$

$$N_{\text{wafer}} = 500\text{個} \quad \text{TESTchip} = 100\text{円}$$

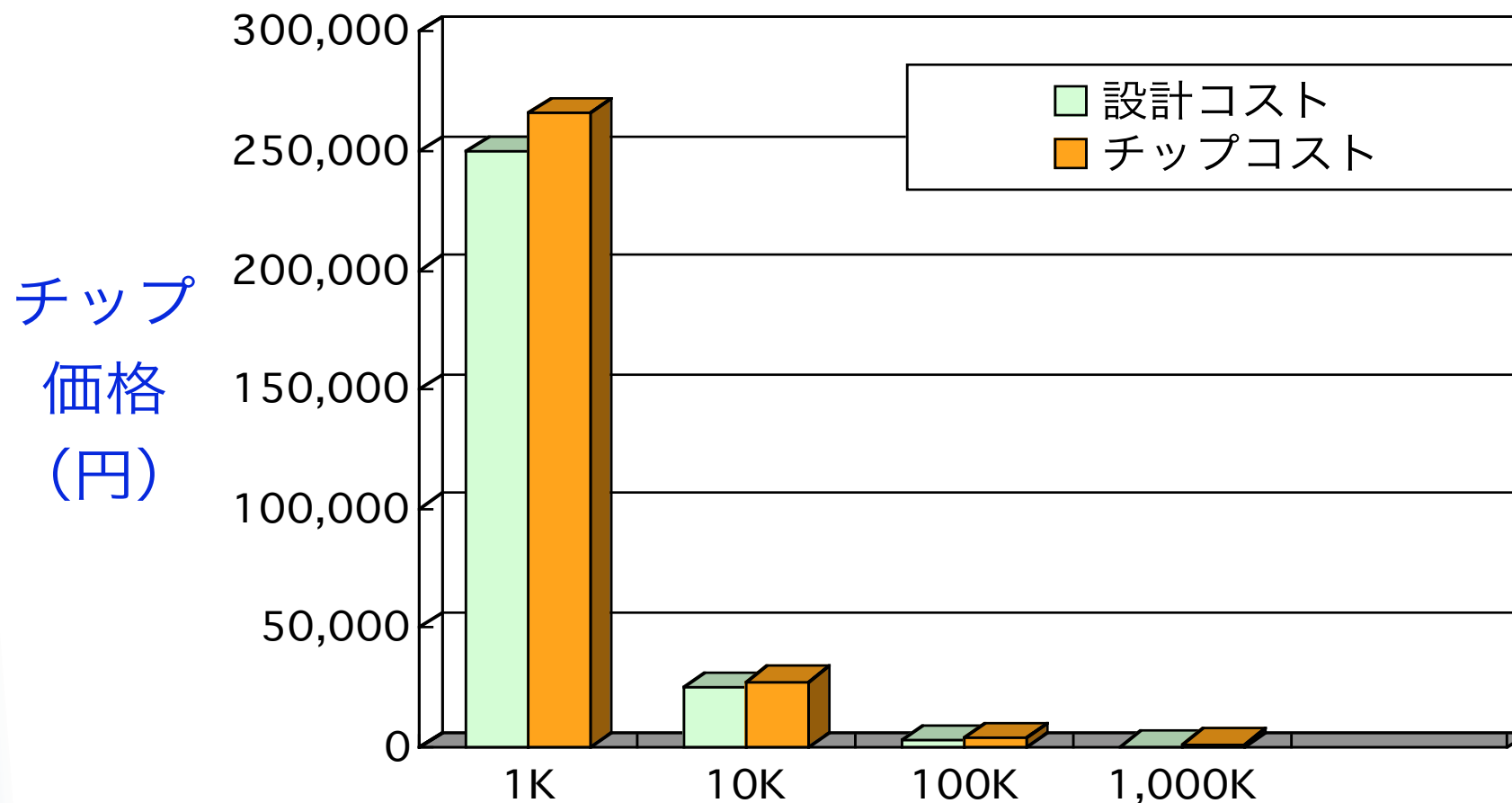
$$\text{Pack} = 300\text{円} \quad \text{IP} = 100\text{円}$$

$$\begin{aligned} \text{CPChw}_i &= (50,000,000 + 15,000,000) / \text{Ngood}_i \\ &\quad + 200,000,000 / \text{Ngood} \\ &\quad + (200,000 / (0.8 * 500)) + 100 + 300 + 100 \\ &= 50,000,000 / \text{Ngood}_i + 200,000,000 / \text{Ngood} \\ &\quad + 1000 (\text{円}) \\ &= 265,000,000 / \text{Ngood} + 1000 (\text{円}) \\ &\quad (\text{バリエーションがない場合}) \end{aligned}$$

市場規模とチップコスト



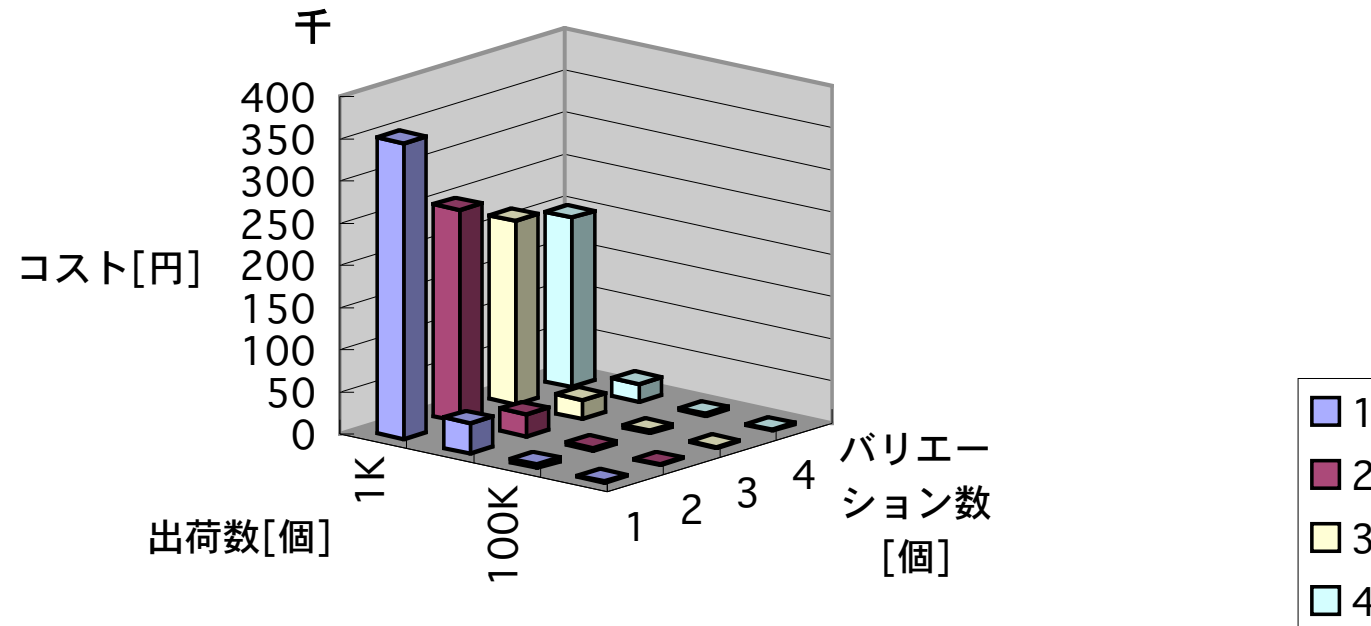
Chip Cost vs. # of Products



製品数 Ngood

九州大学システムLSI研究センター

設計共通化とチップコスト

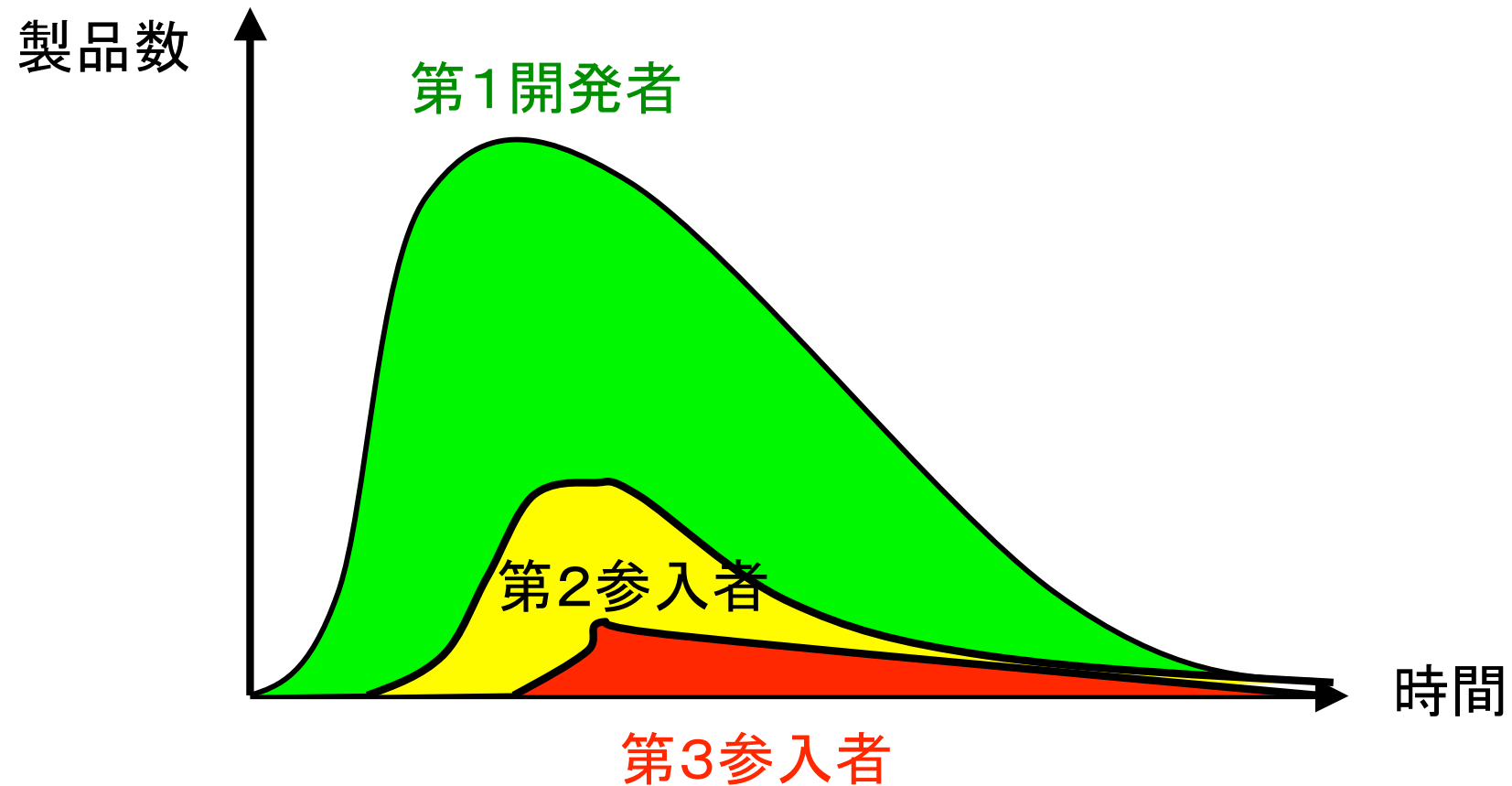


	1K	10K	100K	1,000K
1	351,000	36,000	4,500	1,350
2	251,000	26,000	3,500	1,250
3	217,667	22,667	3,167	1,217
4	201,000	21,000	3,000	1,200

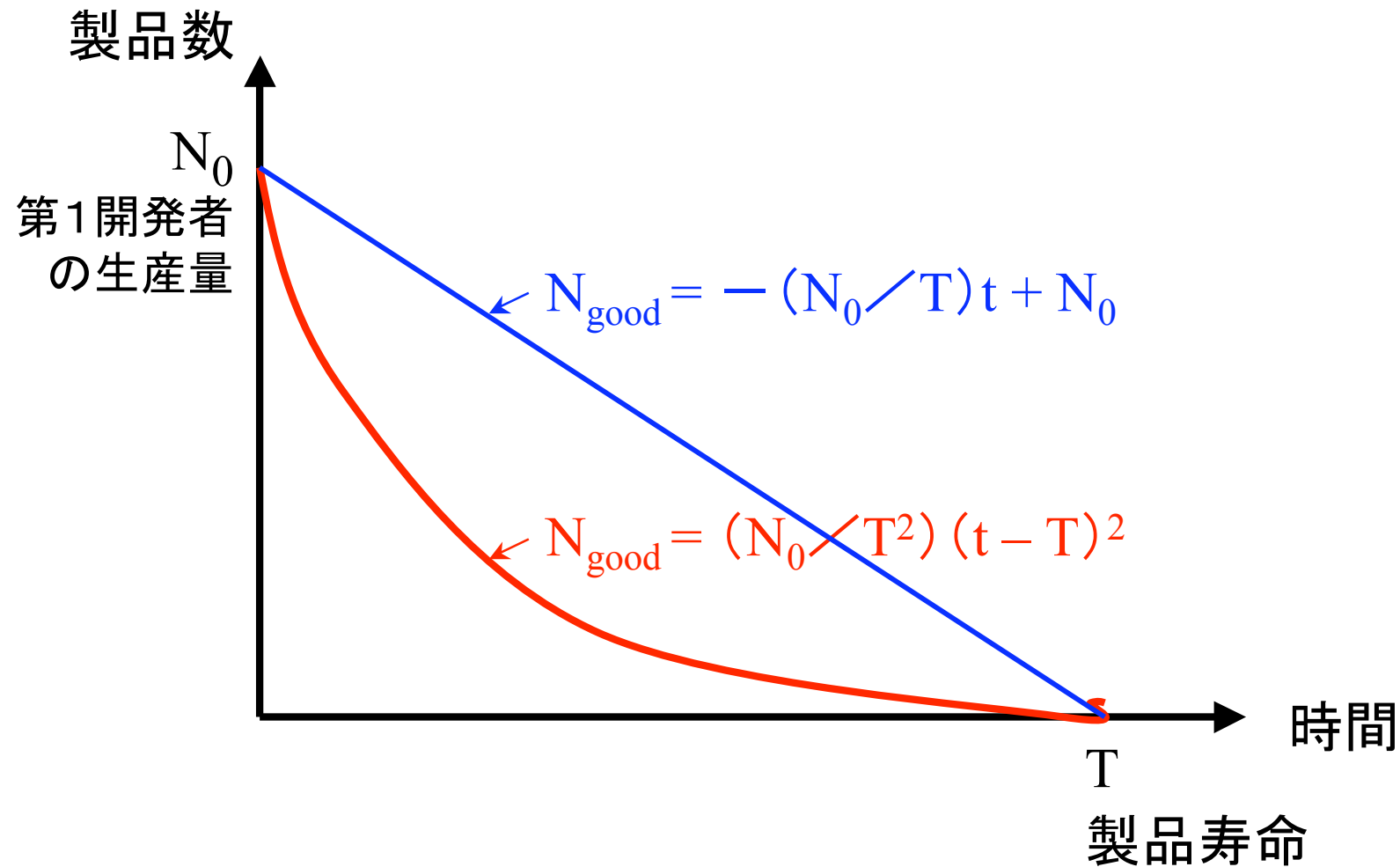
出荷個数は1バリエーションあたり

九州大学システムLSI研究センター

開発期間の影響



近似モデル

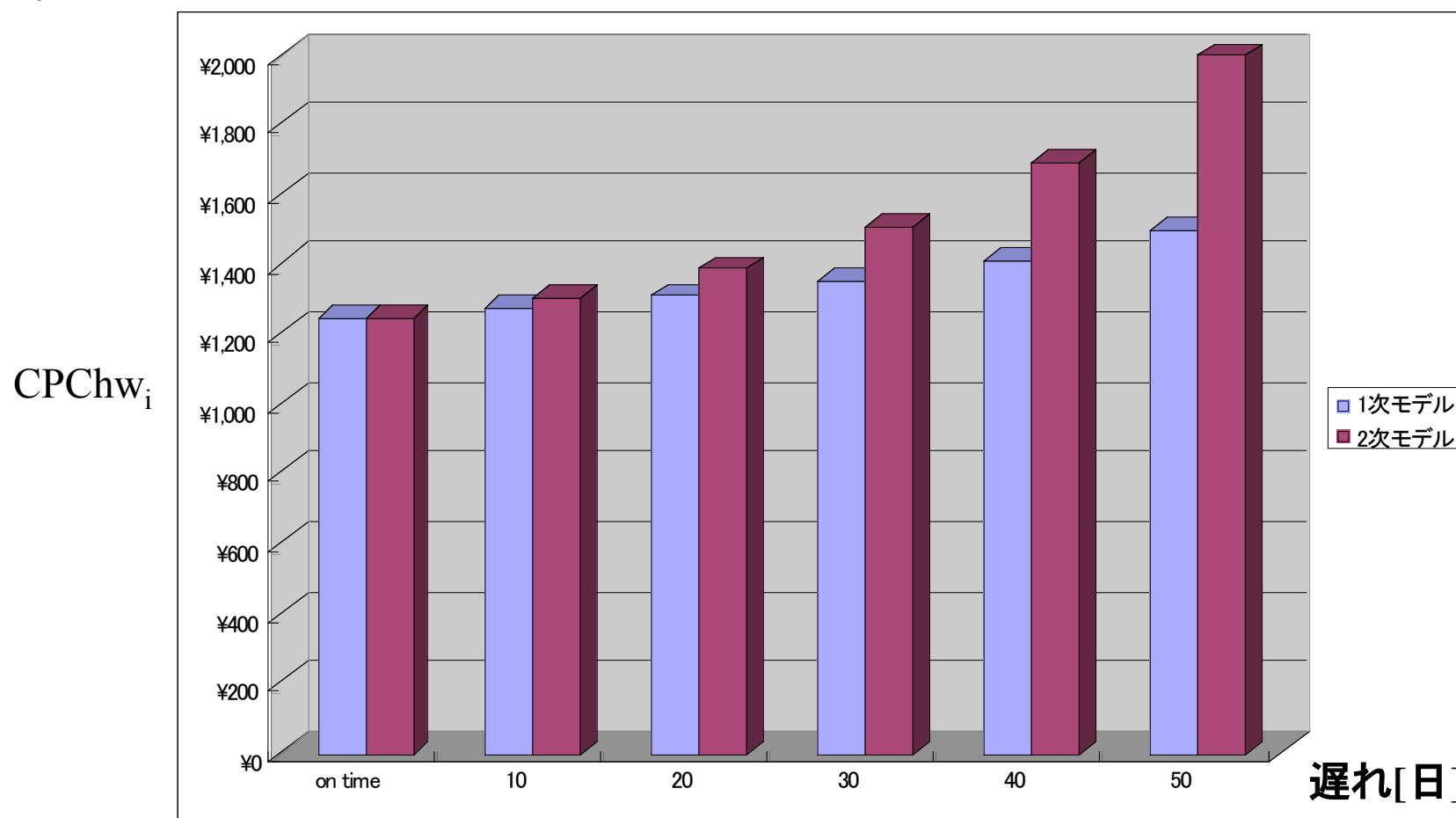


開発期間とチップコスト

$$\text{CPChw}_i = 250,000,000 / \text{N}_{\text{good}} + 1000 (\text{円})$$

$$T = 100 \text{ 日}$$

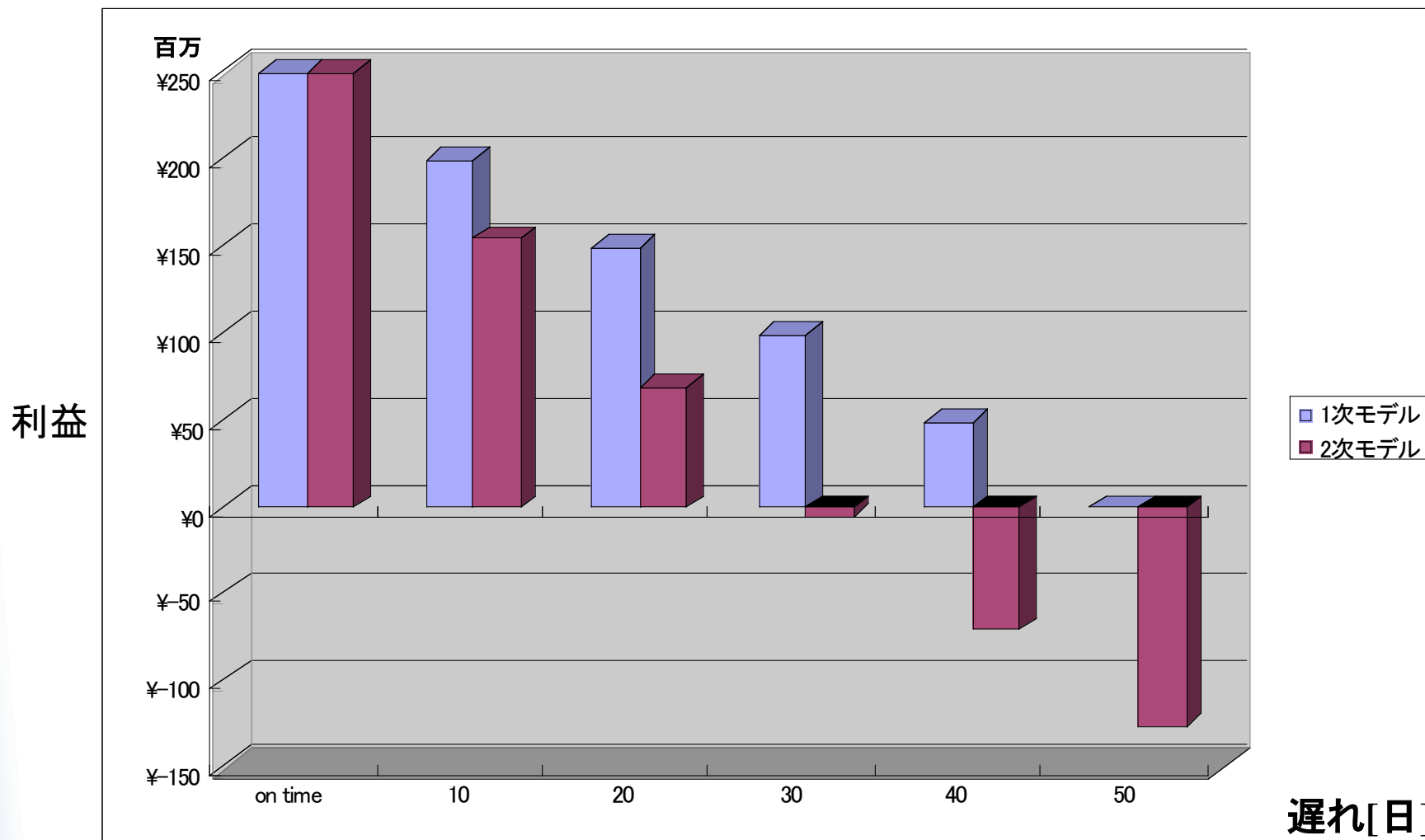
$$N_0 = 1\text{M}$$



開発期間と利益

販売価格を1300円と仮定する.

利益 = $N_{\text{good}} * (1300 - \text{COST}_{\text{chip}})$ (円)



コスト削減のための設計



- チップ面積の削減
 - ◆ ゲート数, レイアウト面積削減
 - ◆ チップ数増加, 歩留り向上
- 設計コスト削減
 - ◆ 設計時間の短縮, 設計自動化, 部品化などの設計効率改善
- マスク数削減
 - ◆ 配線層数削減, マスク共有(ゲートアレイ(拡散・ポリシリ層共有))
 - ◆ プログラマブルデバイス(FPGA)
- テストコスト削減
 - ◆ 短いテストパターン
- パッケージコスト削減
 - ◆ ピン数削減
 - ◆ 消費電力削減
- 売れるチップ
 - ◆ 機能, 性能, 価格, 信頼性

その他のコスト削減

$$CPC_{hw_i} = \frac{COST_{design_i}}{N_{good_i}} + \frac{COST_{design_{common}}}{N_{good}} + \frac{mCOST_{mask_i}}{N_{good_i}} \\ + \frac{COST_{fab_i}}{YN_{wafer_i}} + TEST_{chip_i} + Pack_i + IP_i$$

- COSTmask削減: マスク製造工程の改良
- Ngood / Ngood_i増加: 微細化による高性能化、高機能化
- Ngood増加: 設計の共通化
- COSTfab削減: 大口径ウェハの導入
- Yの向上: 加工精度の向上、工場の品質管理
- Nwafer : 大口径ウェハの導入 (12インチ=30cm wafer)
- TESTchipの削減: テスト機器の改良
- Packの削減: 新しい材料、パッケージ手法

再構成可能なHW

- COSTdesignとCOSTmaskの増加
 - ◆ 設計の複雑化によるバグの混入
 - ◆ 設計仕様の変更・不備
 - ◆ 微細化によるマスク費用の高騰(1枚数百万円)
- 設計変更や設計ミスによる手戻りは大きなコスト負担となる

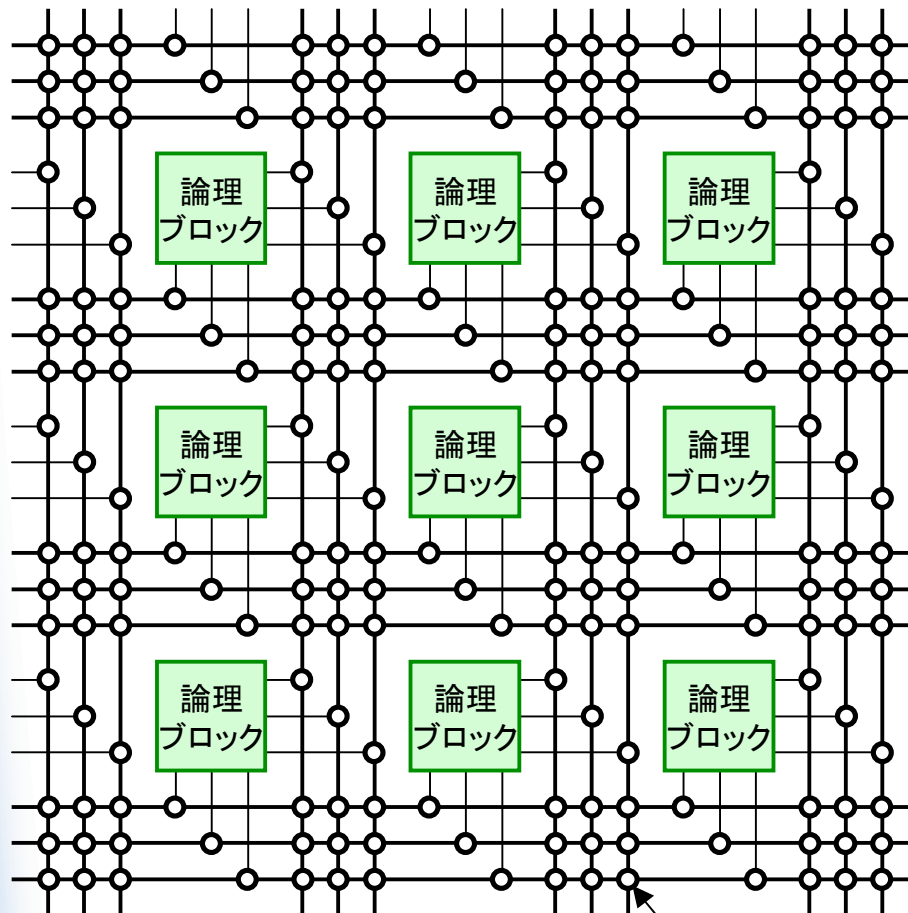
$$CPC_{hw_i} = \frac{COST_{design_i}}{N_{good_i}} + \frac{COST_{design_{common}}}{N_{good}} + \frac{mCOST_{mask_i}}{N_{good_i}} \\ + \frac{COST_{fab_i}}{YN_{wafer_i}} + TEST_{chip_i} + Pack_i + IP_i$$

- 解決策として、製造後に回路の構造を変えることができる可変構造論理の利用が行われている
 - ◆ FPGA: Field Programmable Gate Array
 - ◆ 試作機、少量生産製品などに利用

FPGA: Field Programmable Gate Array



- ・ユーザの手許でカスタム化(プログラム)可能なLSI
- ・カスタム化のためのLSIの試作期間や開発費が不要



小規模論理ブロックを規則的に配置し、その周囲にはプログラム可能な配線を配置したもの

- ・論理ブロックの形式には、ルックアップテーブル、マルチプレクサ、ゲートアレイなどのタイプがある。
- ・配線の交差点には、SRAM、EEPROM、アンチヒューズ等を利用したプログラム用デバイス(スイッチ)が組込まれている。

SRAM: Static Random Access Memory

EEPROM: Electrical Erasable and Programmable Read Only Memory

FPGAの利用によるコスト削減

通常のシステムLSI

$$CPC_{hw_i} = \frac{COST_{design_i}}{N_{good_i}} + \frac{COST_{design_{common}}}{N_{good}} + \frac{mCOST_{mask_i}}{N_{good_i}} + \frac{COST_{fab_i}}{YN_{wafer_i}} + TEST_{chip_i} + Pack_i + IP_i$$

FPGAを用いた設計

$$CPC_{hw_i} = \frac{COST_{design_i}}{N_{good_i}} + \frac{COST_{design_{common}}}{N_{good}} + COST_{fpga} + IP_i$$

- ◆ $COST_{fpga}$: FPGAのチップコスト
 - マスク費用を多くの設計でシェアできる
 - 大量生産できるので、FPGAの物理設計費用は大きくしても影響が小さい
 - 大量生産するので製造コストも小さい(先端技術が利用できる)
 - 一部故障していても問題ない場合もある
 - 規則的な構造でテストがしやすい
- ◆ $COST_{design}$: 論理設計とFPGAへのマッピングまででよく、物理設計が不要。
- ◆ 仕様変更や設計ミスがあってもあとで修正が可能
- ◆ 性能は通常のシステムLSIよりは悪い($\times 10^{-1} \sim 10^{-2}$)
- N_{good} が小さく、性能要求に余裕がある場合には有利になる。

チップコストの例

$\text{COST}_{\text{design}_i} = 50,000,000\text{円} \rightarrow 30,000,000\text{円}$ $m = 30\text{枚} \rightarrow 0\text{枚}$

$\text{COST}_{\text{design}} = 200,000,000\text{円} \rightarrow 50,000,000\text{円}$

$\text{COST}_{\text{mask}} = 500,000\text{円}$ $\text{COST}_{\text{fab}} = 200,000\text{円}$

$Y = 0.8$ $N_{\text{wafer}} = 500\text{個}$ $\rightarrow \text{FPGA購入費用} 10,000\text{円}$ $\text{TEST}_{\text{chip}} = 100\text{円}$
 $\rightarrow 0\text{円}$

$\text{Pack} = 300\text{円} \rightarrow 0\text{円}$

$\text{IP} = 100\text{円} \rightarrow 0\text{円}$

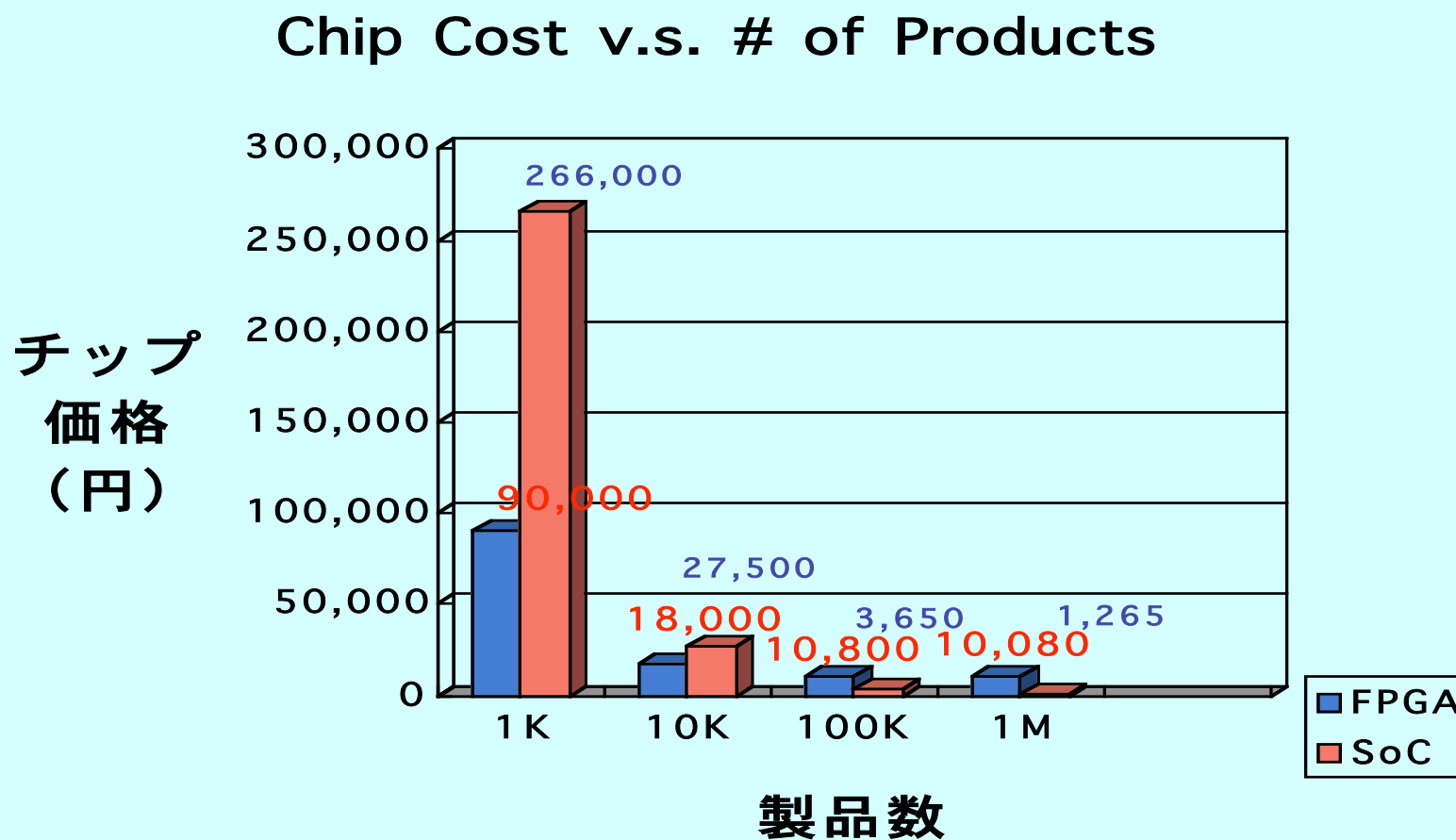
ASIC的SoCのコスト

$$\begin{aligned} \text{CPC}_{\text{hw}_i} &= (50,000,000 + 15,000,000) / N_{\text{good}_i} + 200,000,000 / N_{\text{good}} \\ &\quad + 500 + 100 + 300 + 100 \\ &= 265,000,000 / N_{\text{good}} + 1000 \text{ (円)} (\text{バリエーションがない場合}) \end{aligned}$$

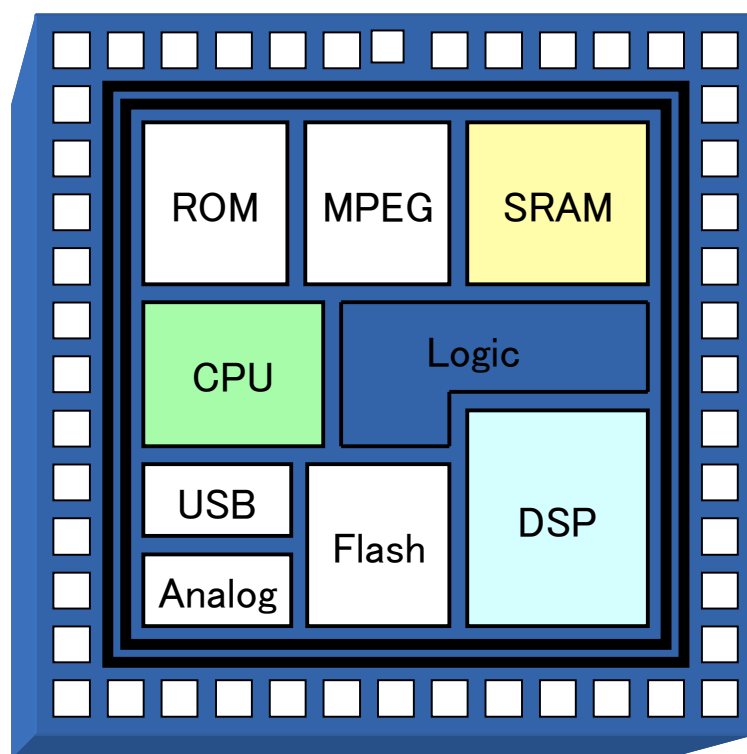
FPGA利用の時のコスト

$$\begin{aligned} \text{CPC}_{\text{hw}_i} &= 30,000,000 / N_{\text{good}_i} + 50,000,000 / N_{\text{good}} \\ &\quad + 10,000 \\ &= 80,000,000 / N_{\text{good}} + 10,000 \text{ (円)} (\text{バリエーションがない場合}) \end{aligned}$$

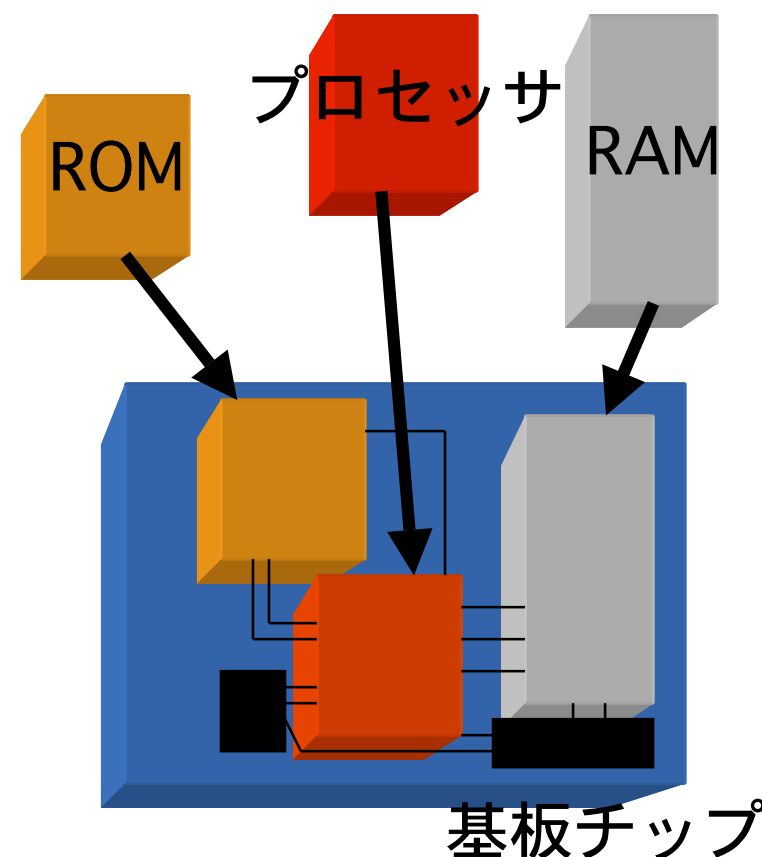
製品数とコスト



SoCとSiP



- 単一チップ方式
 - モノリシックシステムLSI



- 複数チップ方式
 - System in Package

チップコストの例

$\text{COST}_{\text{design}_i} = 50,000,000\text{円} \rightarrow 50,000,000\text{円}$

$m = 30\text{枚} \rightarrow 20\text{枚}$

$\text{COST}_{\text{design}} = 200,000,000\text{円} \rightarrow 40,000,000\text{円}$

$\text{COST}_{\text{mask}} = 500,000\text{円} \rightarrow 500,000\text{円}$

$\text{COST}_{\text{fab}} = 200,000\text{円} \rightarrow 200,000\text{円}$ $Y = 0.8 \rightarrow 0.9$

$N_{\text{wafer}} = 500\text{個} \rightarrow 400\text{個}$ $\text{TEST}_{\text{chip}} = 100\text{円} \rightarrow 70\text{円}$

$\text{Pack} = 300\text{円} \rightarrow \text{実装費}800\text{円} + \text{搭載チップ価格}2,000\text{円}$

$\text{IP} = 100\text{円} \rightarrow 0\text{円}$

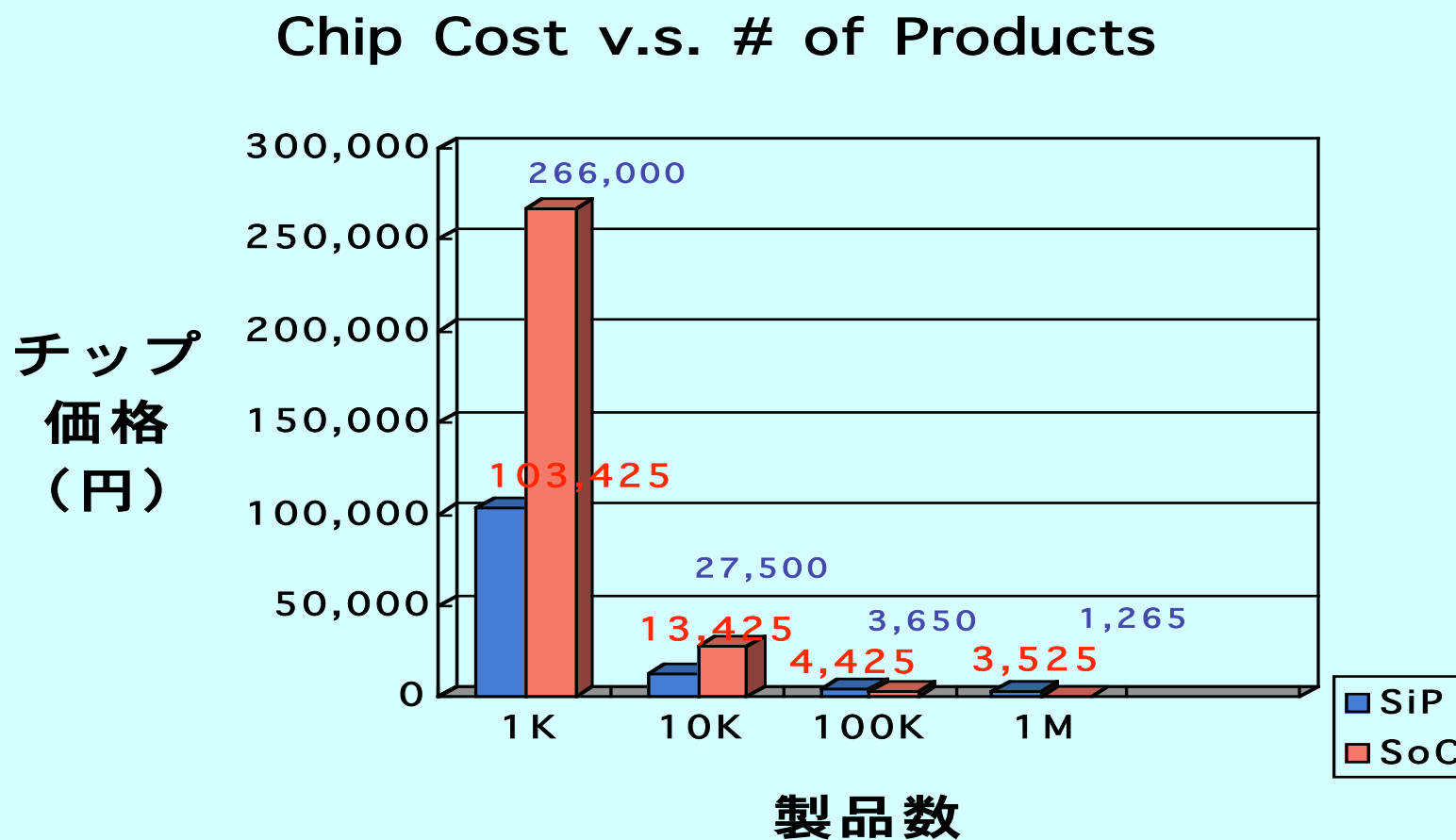
SoCのコスト

$$\begin{aligned}\text{CPC}_{\text{hw}_i} &= (50,000,000 + 15,000,000) / \text{N}_{\text{good}_i} + 200,000,000 / \text{N}_{\text{good}} \\ &\quad + 500 + 100 + 300 + 100 \\ &= 265,000,000 / \text{N}_{\text{good}} + 1000 \text{ (円)} (\text{バリエーションがない場合})\end{aligned}$$

SiPのコスト

$$\begin{aligned}\text{CPC}_{\text{hw}_i} &= (50,000,000 + 10,000,000) / \text{N}_{\text{good}_i} + 40,000,000 / \text{N}_{\text{good}} \\ &\quad + 555 + 70 + 2,800 \\ &= 100,000,000 / \text{N}_{\text{good}} + 3425 \text{ (円)} (\text{バリエーションがない場合})\end{aligned}$$

製品数とコスト



SiP方式の利点

- 汎用コア（メモリ，プロセッサ，通信インタフェースなど）の大量生産
 - ◆ コスト削減
 - ◆ IP化のリスク回避
- 基板とコアのプロセス技術の使い分け
 - ◆ 複数電源への対応，耐圧など
- チップ単位でのセキュリティ確保
- コア単位の個別テスト

ソフトウェア開発の問題点



- 圧倒的に大きなシステム複雑度
 - ◆ 数百万ステップ
 - ◆ 完全な仕様が書けない
- 低い生産性
 - ◆ 正当性の検証の難しさ（数ステップ/人日）
 - ◆ インターフェースの検証
- 互換性の保証
 - ◆ OS：対HW、対応用ソフトウェア
 - ◆ Open Software Communityとの関係

ソフトウェアコスト

ソフトウェア
バリエーションjの
SW開発コスト

ハードウェア
バリエーションiの
SW開発コスト

再利用可能部分の
SW開発コスト

ソフトウェア
ロイヤリティ

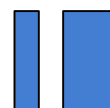
$$CPC_{sw_{ij}} = \frac{COST_{sw_{ij}}}{N_{good_{ij}}} + \frac{COST_{sw_i}}{N_{good_i}} + \frac{COST_{sw_{common}}}{N_{good}} + IP_{sw_j}$$

あるアプリケーションij
固有のソフトウェア

あるSoCi固有の
ソフトウェア

複数のSoCで使える
再利用可能な
ソフトウェア

小



Nの大きさ

大

対象LSI共通のSW開発コスト

$COST_{SW_{common}}$

- 要求定義、分析
- アーキテクチャ設計
- 基盤ソフトウェア導入
- 開発環境構築
- コンポーネント開発

etc

- テスト設計
- テスト環境構築
- 基盤ソフトウェアテスト
- コンポーネントテスト
- テストパターン開発

etc

バリエーション*j*のSW開発コスト

 $COST_{SW_i}$

SoC *i* 固有のソフトウェア開発コスト
ドライバなど

- 要求定義、分析
- 設計
- 開発
- テスト設計
- テストパターン開発
- テスト

 $COST_{SW_{ij}}$

SoC *i* 上で動作する、ある応用*j*のソフトウェア開発コスト
ユーザインターフェースなど

etc

コストの関係

$$CPC_{sw_{ij}} > CPC_{hw_i}$$

ソフトウェアコストの方が
高くなりつつある

$$COST_{sw_{ij}} + COST_{sw_i} + COST_{sw_{common}} \gg COST_{design_i} + COST_{design_{common}}$$

設計・実装コストは
ソフトウェアのほうが高い

製造コストが低ければ
ハードウェアでバリエーションを出し
てもペイする

なるべく共通化した方がお得
なるべくポータブルに実装すると
もっとお得

$$CPC_{sw_{ij}} = \frac{COST_{sw_{ij}}}{N_{good_{ij}}} + \frac{COST_{sw_i}}{N_{good_i}} + \frac{COST_{sw_{common}}}{N_{good}} + IP_{sw_j}$$

小



Nの大きさ

大

SW再利用と開発コスト / 仮定



仮定

変数	個数	仮定
Ngood	7,500,000	5年 / シリーズ 5機種 / 年
Ngoodi	300,000	3応用(バージョン) / SoC
Ngoodij	100,000	10万個 / 1応用

	個別に実装	SoCごとに実装	複数のSoCで共有
COST _{sw common}	¥50,000,000	¥50,000,000	¥500,000,000
COST _{swi}	¥50,000,000	¥500,000,000	¥100,000,000
COST _{swij}	¥100,000,000	¥10,000,000	¥10,000,000

OSなどは再利用
他は応用ごとに実装

SoCごとに再利用可能なソフトウェアを実装

複数のSoCで再利用可能なソフトウェアを実装

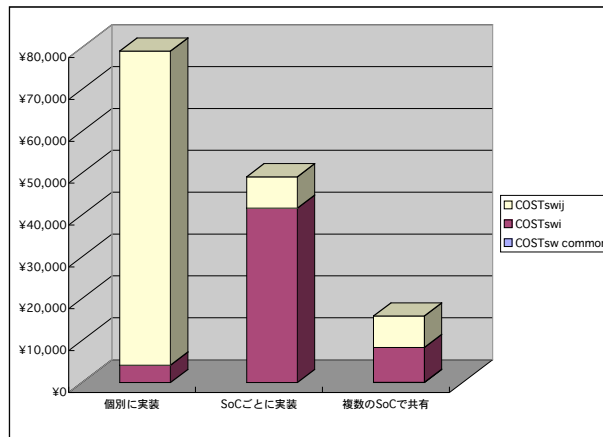
1応用の開発コストは低い

SoCごとに集中投資

プラットフォーム開発に集中投資

久住@QUBE

SW再利用と開発コスト



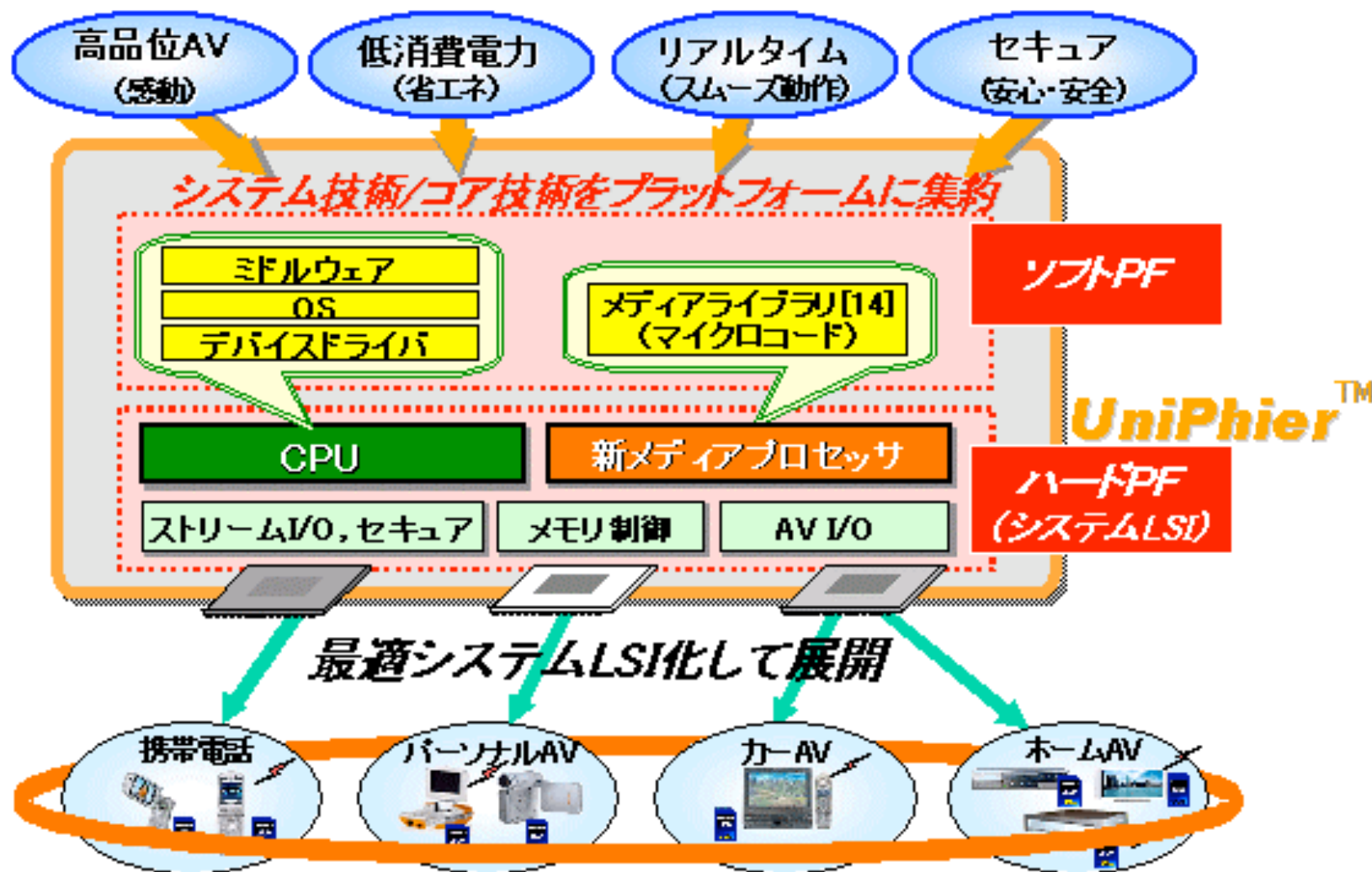
1台あたり	個別に実装	SoCごとに実装	複数のSoCで共有
COSTsw common	¥7	¥7	¥67
COSTswi	¥4,167	¥41,667	¥8,333
COSTswij	¥75,000	¥7,500	¥7,500
合計	¥79,173	¥49,173	¥15,900

- $\text{COST}_{\text{sw common}} / N_{\text{good}}$ は N_{good} が大きいため、小さくなる
- $\text{COST}_{\text{sw common}}$ への投資は結局は安くつく
- 一時的な出荷時期遅延の痛み
- なるべく再利用すべき
- 独自開発には注意が必要

設計プラットフォーム

- 製品系列に共通なHWやSWの共有を目指したシステム設計基盤 ⇒ **COSTdesign_iとIP費の削減**
 - ◆ ソフトウェアのインターフェースと構造の共通化
 - ミドルウェア、OS、デバイスドライバなど
 - 共通のプロセッサアーキテクチャ
 - ◆ ハードウェアアーキテクチャの共通化
 - プロセッサとコンパイラ(対SW)
 - バスなどのインタフェース(対HW)
 - 専用ハードウェア(専用プロセッサ、専用回路)
 - ◆ 開発効率の向上と設計資産の再利用
 - 半導体集積回路技術への統一的なインターフェース
 - 低消費電力化、高品質化、高信頼化、セキュリティ対策
 - 各種アプリケーションソフトウェアや標準プロトコルへの統一的な対応

設計プラットフォーム



出典: 松下電器産業(株) デジタル家電に対応する統合プラットフォーム
UniPhier™ (Universal Platform for High-quality Image-Enhancing Revolution)
九州大学システムLSI研究センター

まとめ



- 製品・製品群・関連製品の中でいかに設計コスト，マスク費用，ソフトウェア実装コストを共有化するかの問題
- ソフトウェア開発費の高騰をいかに抑制するか？
- プロセッサベースやFPGAベースの設計とASIC的発想の比較
- SiPも一つの解

参考図書：安浦寛人監修：SoC開発講座，丸善，
2006.