# 高性能マイクロプロセッサの発展過程と今後

京都大学
大学院情報学研究科
富田眞治

# 目次

- １　マイクロプロセッサの発展
- ２　マイクロプロセッサの高速化技術と命令レベル並列
- ３　マルチコア型プロセッサ

# 1　マイクロプロセッサの発展

## (1)1970年代:オンチッププロセッサの幕開け

1970:　1KbitのDRAM

1971:　Intel4004:日本人の嶋　正利さんが関係

4ビットプロセッサ、2300個のトランジスタ,

750KHz、300mW、16ピンパッケージ

8μmデザインルール、8KB ROM、640B RAM

命令実行時間:10.8μsec／21.6μsec

嶋　正利:マイクロプロセッサの２５年、電子情報通信学会誌、８２巻、１０号、pp.997-1017,1999

1978:Intel　8086, IA-32の始まり

DEC　VAX-11/780、超CISC、高級言語マシン

## （２）1980年代：RISCの時代

1978：VAX　商用CISCコンピュータ

高級言語計算機に対抗して

Patterson and Ditzel: The Case for the Reduced Instruction Set Computer,Comp Arc News,1980

１チッププロセッサ：数万TRの時代

高級言語の普及

コンパイル技術との協調

高級言語計算機の反省：捨ての美学

MIPS,SPARC,PA…

# (3)1990年代:命令レベル並列処理の時代

## 1992年:DECAlpha21064

D.Dobberpuhl et al: A 200 MHz 64-b Dual-Issue CMOS Microprocessor, IEEE J of Solid State Circuits, 27,11, pp.1555-1567(1992)

DEC社　CISCからRISCへの転換
命令パイプライン:2多重、7ステージ
168万個のトランジスタ、200MHz、431ピンパッケージ、30W

## 2000年:Pentium4

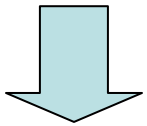
命令パイプライン:3多重、20ステージ
4200万個のトランジスタ、 1.4GHz、478ピンパッケージ、55W

DRAM:64Mbit～256Mbit

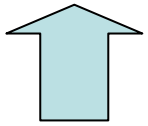
# (4)2000年代:マルチコア／省電力化の時代

①大域並列の利用
・パイプライン:時間並列
・乱実行、投機実行による時間並列の高速化
・局所並列:スーパスカラ、VLIW
・非常に複雑な構造
　IPCの頭打ち

プログラムカウンタの近傍にある命令の並列実行



・大域並列:マルチコア型プロセッサ



・周波数向上が困難
　深いパイプラインで性能向上:小さくなった

②省電力化
　動的消費電力∝$f^3$
　リーク電流の増大

③超高信頼、セキュアなプロセッサ

キャッシュ、分岐予測ミス
ステージ内ゲート段数

# ２ 高速化技術と命令レベル並列

## ①VLSIハードウェア技術

Mooreの法則：Tr数：2倍／1.5年：　メモリ10年で100倍の容量

スケーリング則　S：スケールファクタ／3年：0.7

## ②局所性の利用とメモリ階層

２，３階層キャッシュ：１次キャッシュ：0.25nsec、DRAM：50nsec

ギャップ：200倍！！

## ③機械命令レベルでの並列性の利用

## ④コンパイラとの協調

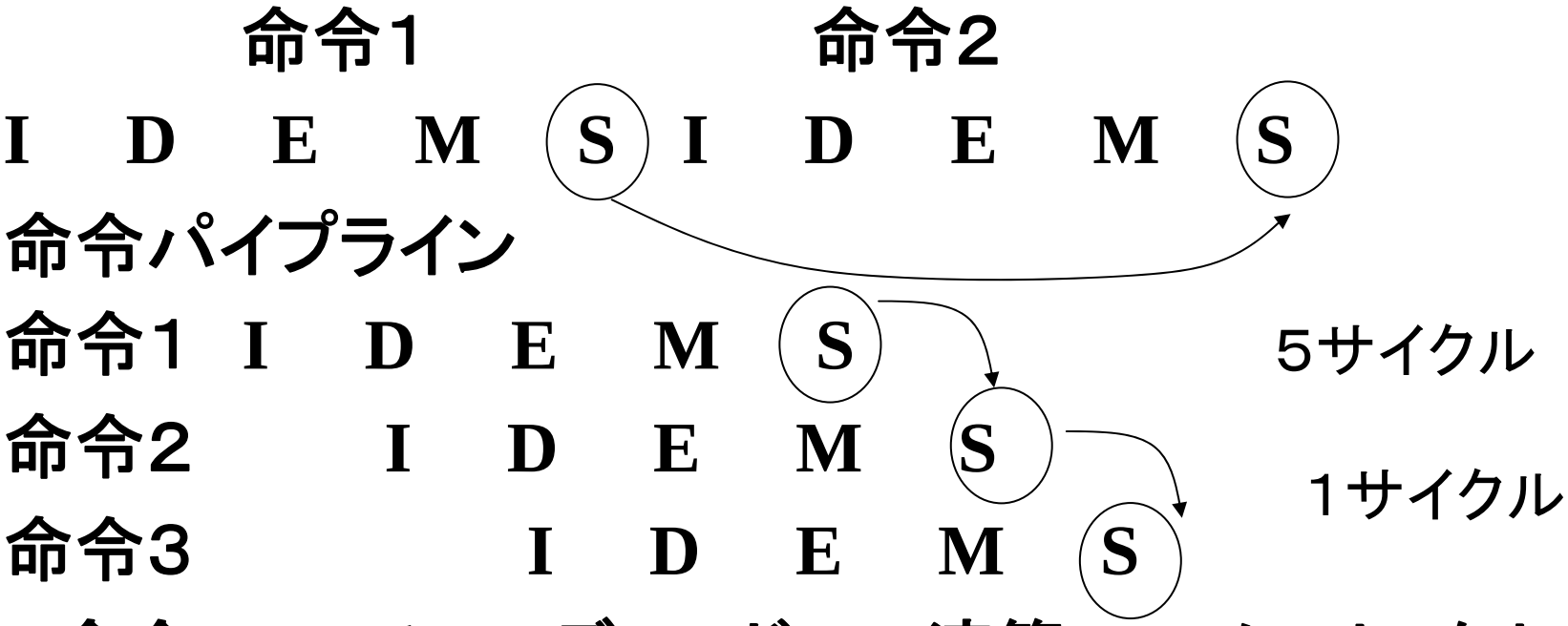
RISCの考え方、トレーススケジューリング、ソフトウェアパイプライン、ループアンローリング

- M.Lam:Software Pipelining: An Effective Scheduling Technique for VLIW Machines, Proc of Conf on Programming Language Design and Implementation, pp.318-328(1988)
- J.Fisher:Trace Scheduling: A Technique for Global Microcode Compaction, IEEE Trans. Computers, C-30,7, pp.478-490(1981)

## ⑤応用指向のハードウェア

マルチメディア機構：４，８要素のSIMD処理：MMX、SSE

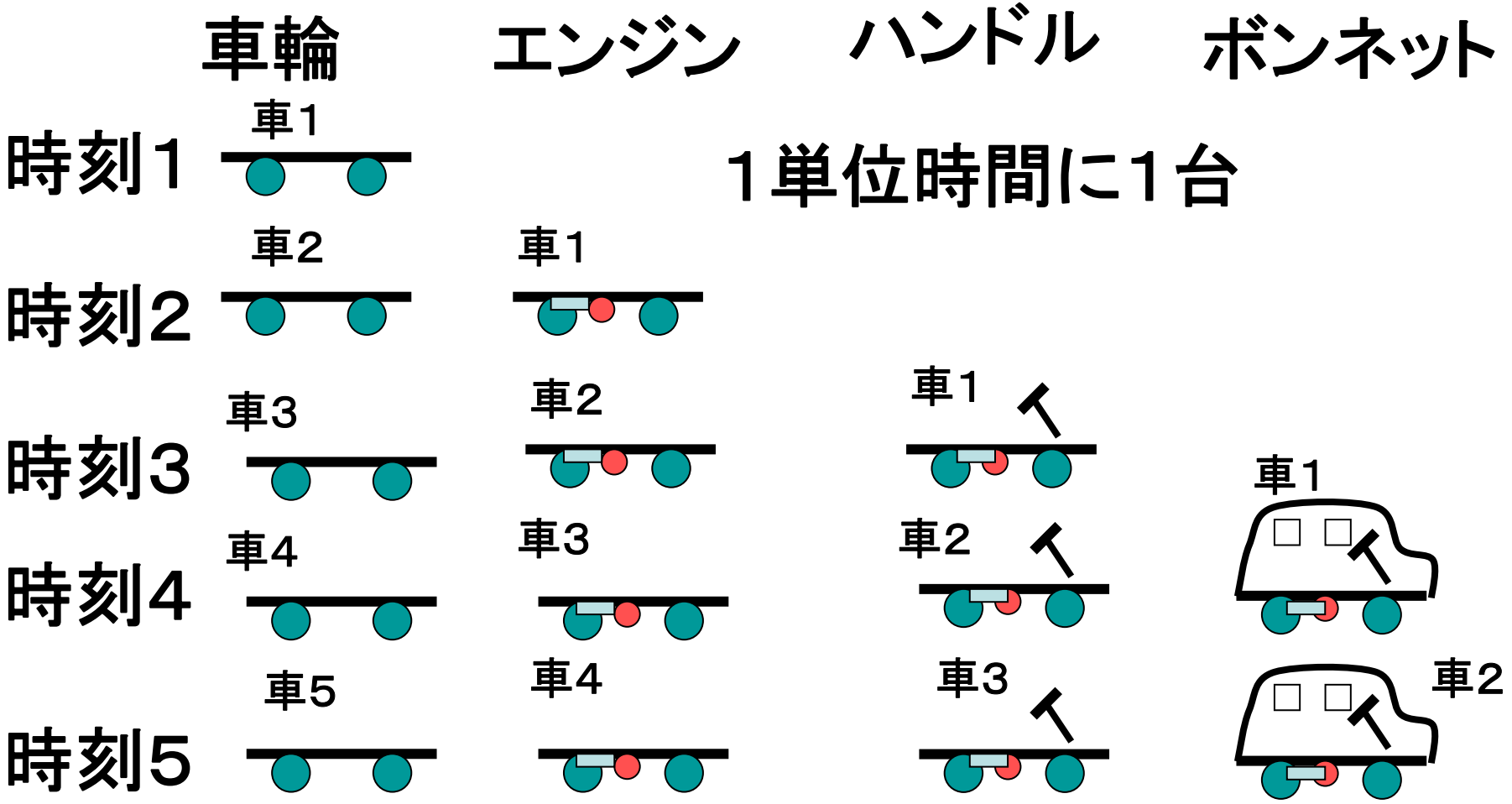
# （1）単純な時間並列性の利用
# パイプライン




I:命令フェッチ，D:デコード，E：演算，M:メモリアクセス，
S：格納

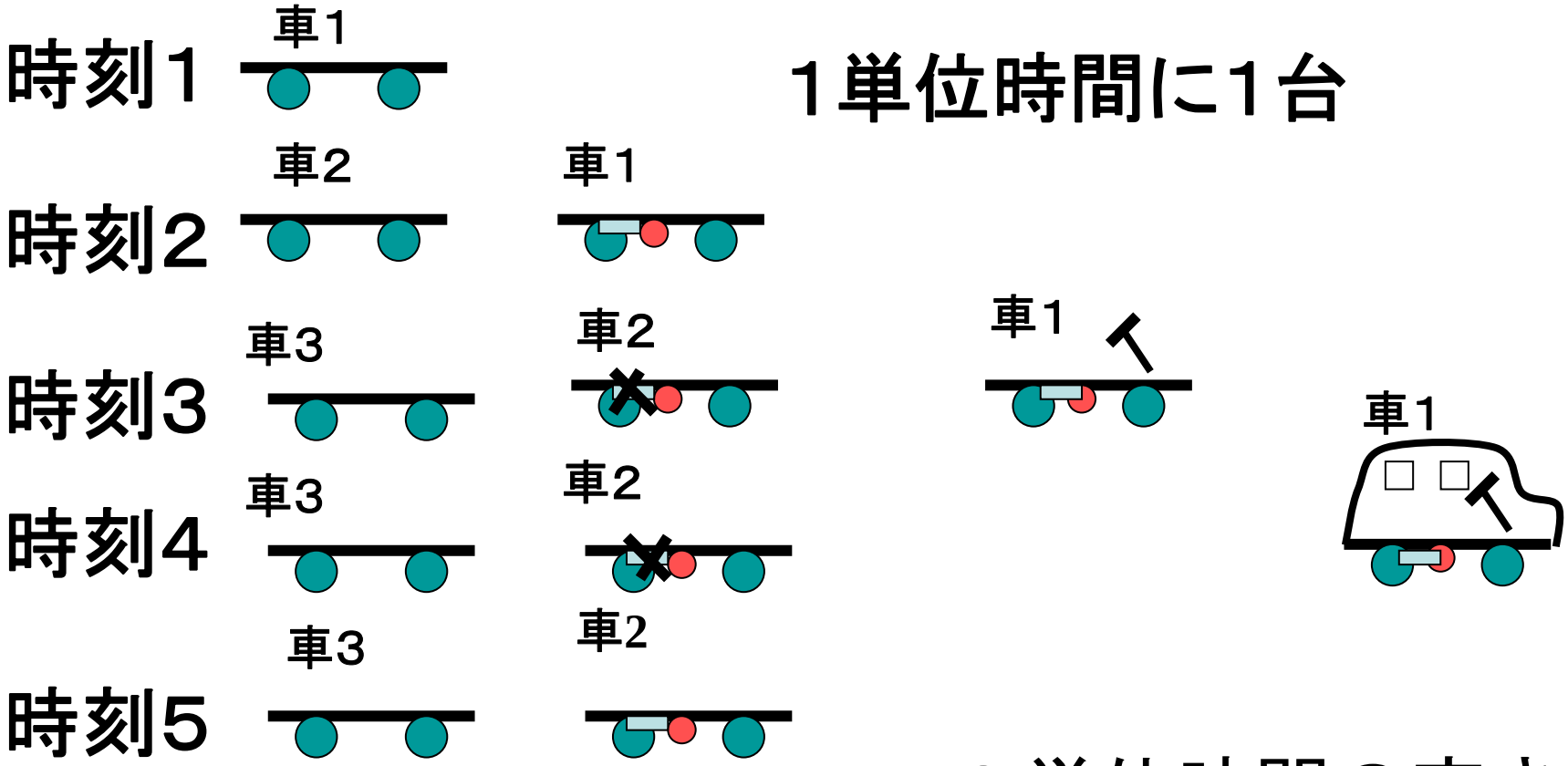
CPI： Cycles Per Instruction，理想的には1

# パイプライン＝流れ作業

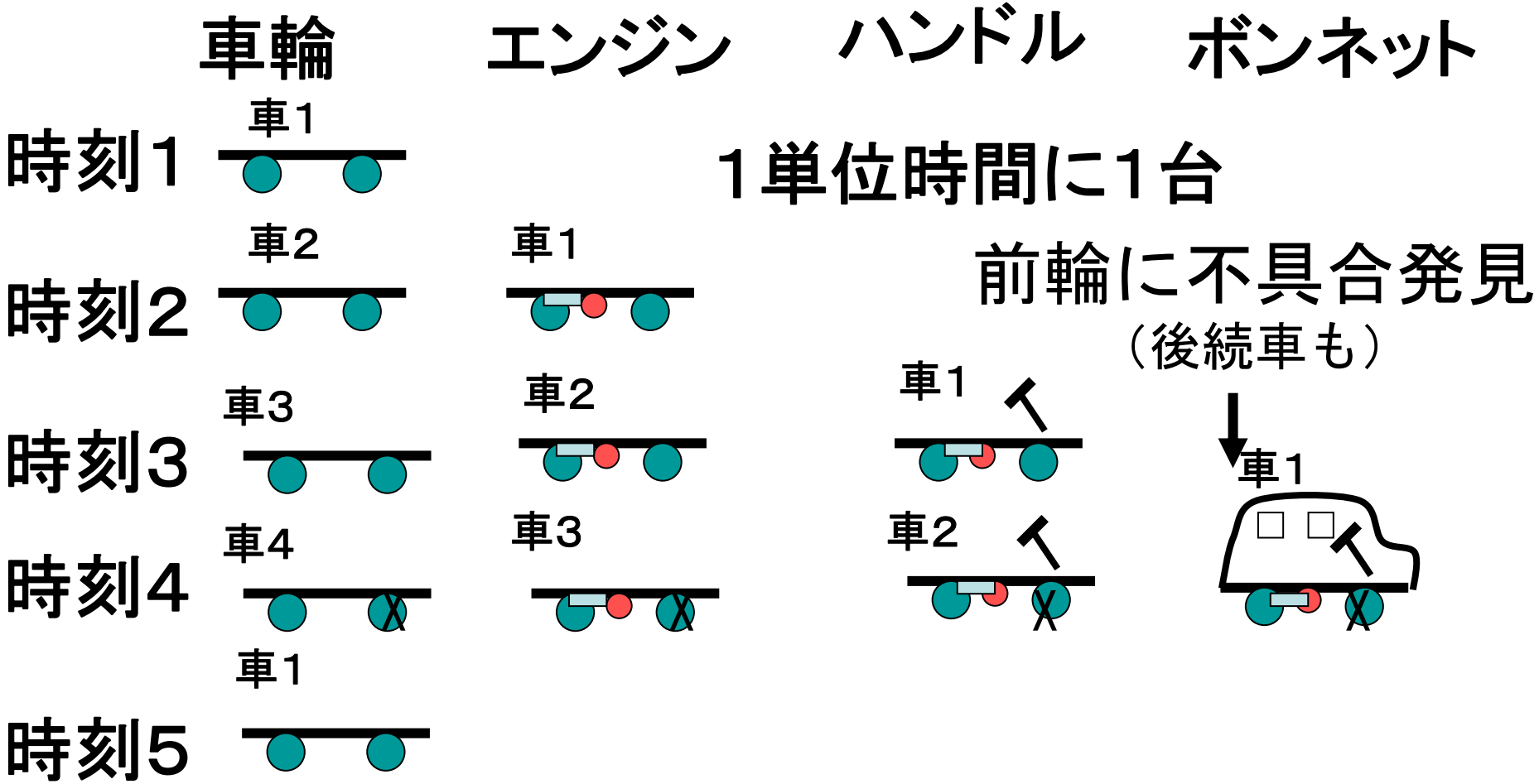

車輪
エンジン
ハンドル
ボンネット
時刻1
車1
1単位時間に1台
時刻2
車2
車1
時刻3
車3
車2
車1
時刻4
車4
車3
車2
車1
時刻5
車5
車4
車3
車2

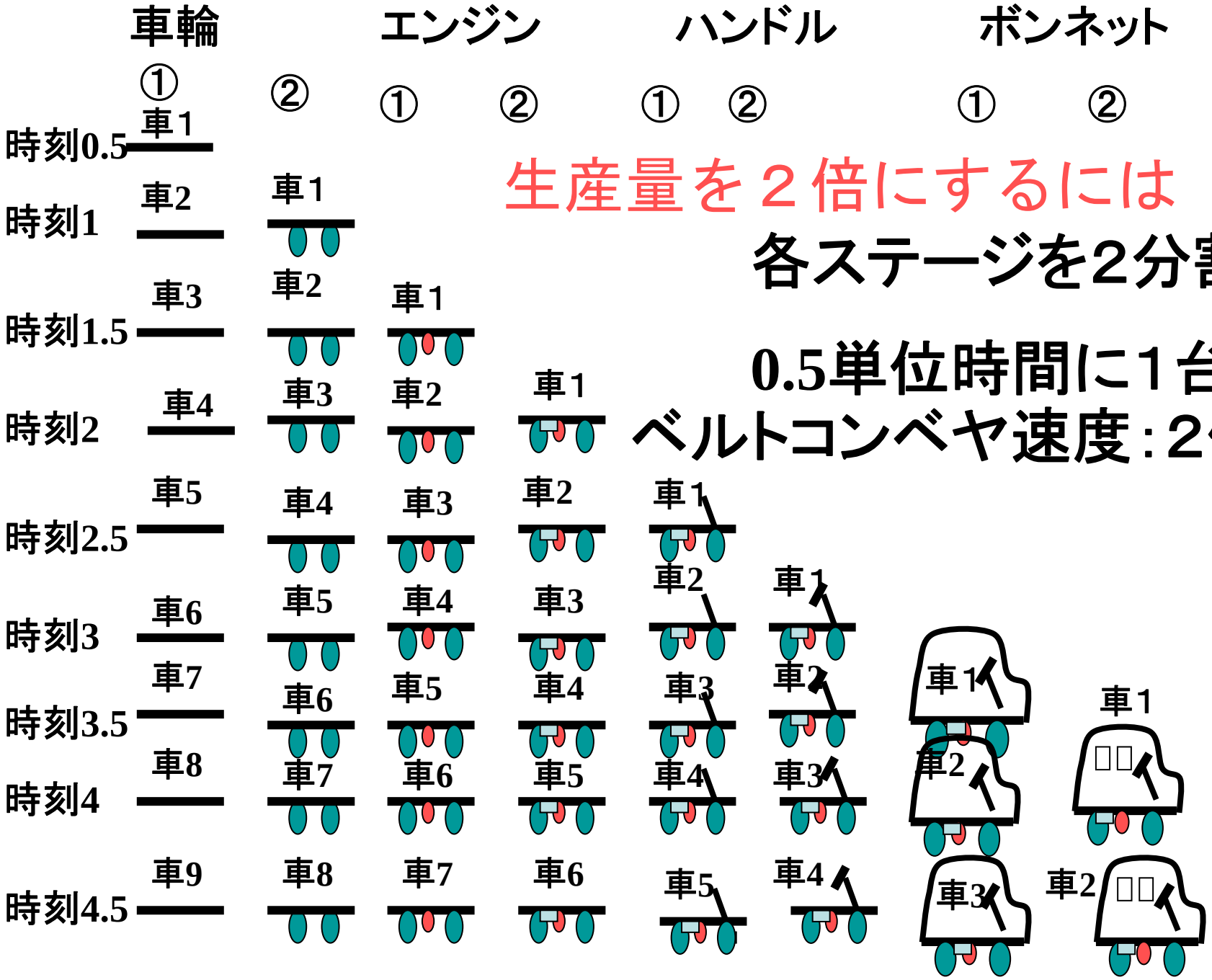
# 流れ作業の乱れ
## （1）部品調達遅れ

車輪　　エンジン　　ハンドル　　ボンネット



# 流れ作業の乱れ
## （2）検査不良による再取付け



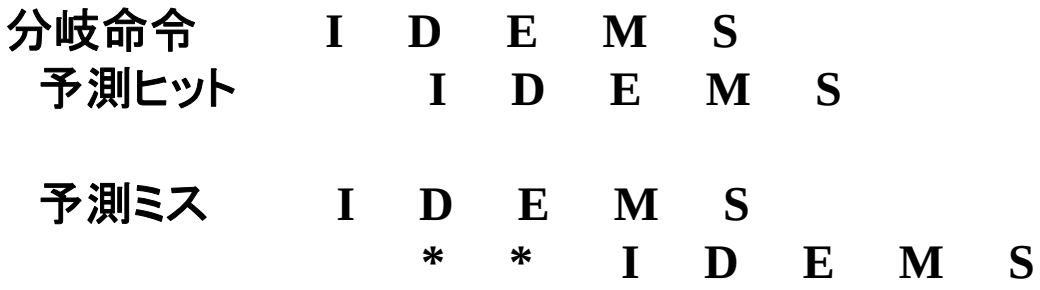

車輪
エンジン
ハンドル
ボンネット
①
②
①
②
①
②
①
②
時刻0.5
車1
時刻1
車2
車1
時刻1.5
車3
車2
車1
時刻2
車4
車3
車2
車1
時刻2.5
車5
車4
車3
車2
車1
時刻3
車6
車5
車4
車3
車2
車1
時刻3.5
車7
車6
車5
車4
車3
車2
時刻4
車8
車7
車6
車5
車4
車3
時刻4.5
車9
車8
車7
車6
車5
車4
車3
車2
車1
生産量を２倍にするには
各ステージを2分割
0.5単位時間に1台
ベルトコンベヤ速度：2倍

## データ依存

改良機構


バイパス、乱実行

## 制御依存




分岐予測、投機実行

分岐予測ミスによる性能低下=分岐出現確率＊パイプライン段数＊ミス率

=0.16*20*0.1=0.32

## 資源競合

### 演算器


パイプライン化、並列化

### キャッシュ

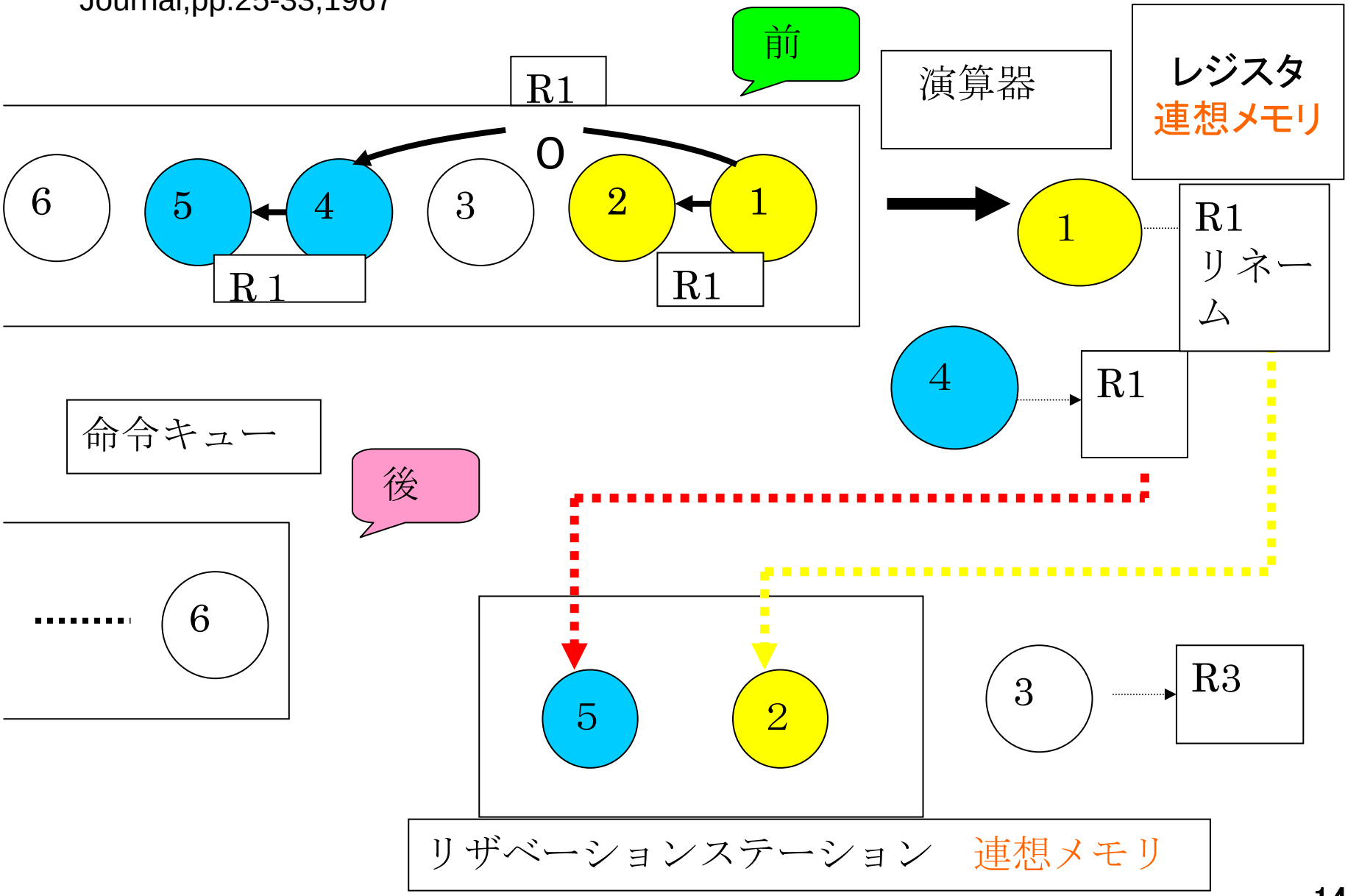

階層化、ノンブロッキング化

キャッシュミスによる性能低下＝L/S命令の確率＊ミス率＊転送時間

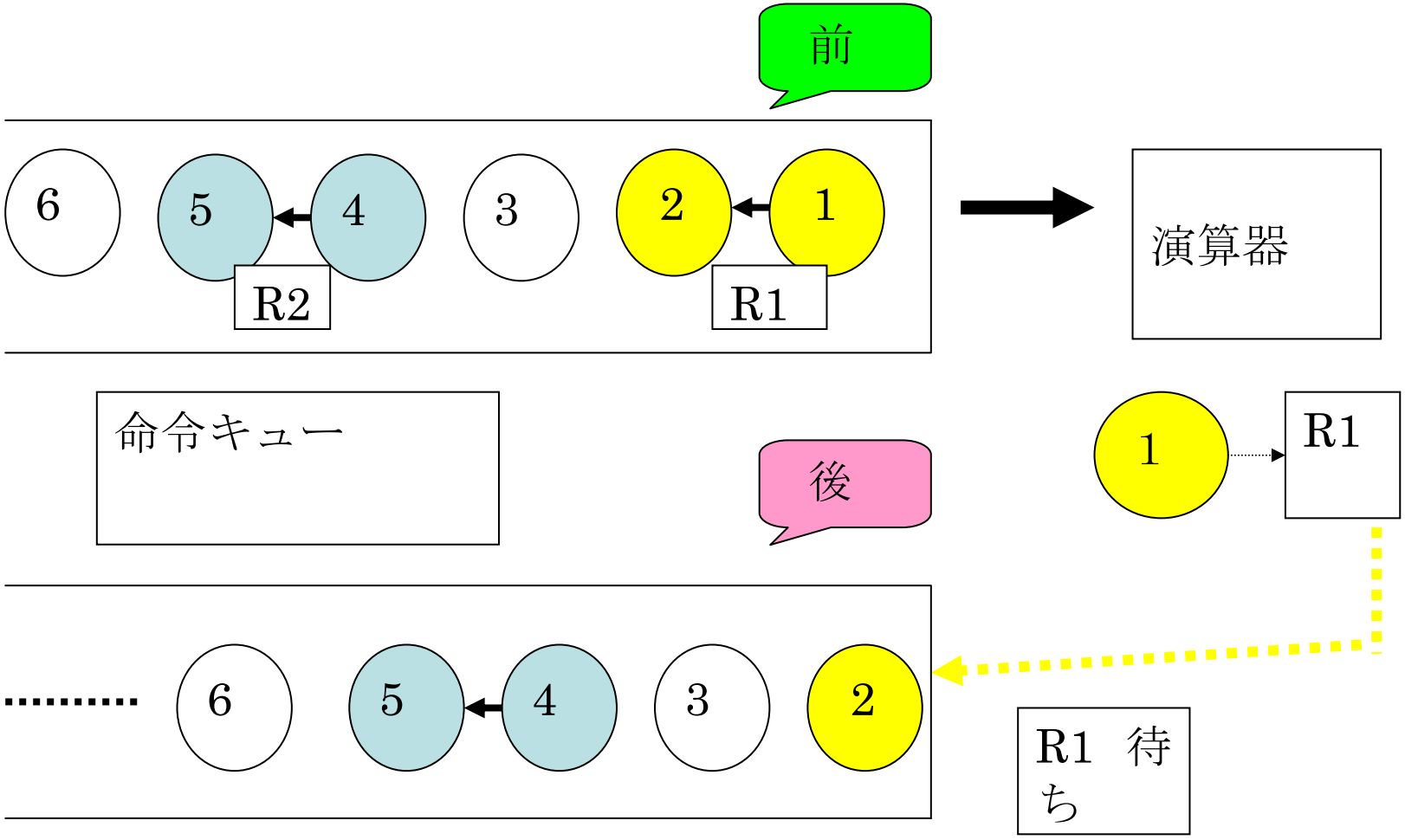
＝0.35＊0.02＊50＝0.35

# (2)Tomasuloの乱実行方式

R.M.Tomasulo:An Efficient Algorithm for Exploiting Multiple Arithmetic Units, IBM Journal,pp.25-33,1967



# 順発行



連想メモリ

レジスタ書込みのタグ比較

リネーミングレジスタ方式で解消

論理レジスタと物理レジスタの動的な対応付け

書き込み時に対応付けを変更

K.C.Yeager: The MIPS R10000 Superscalar Microprocessor, IEEE Micro,pp.28-40,April(1996)

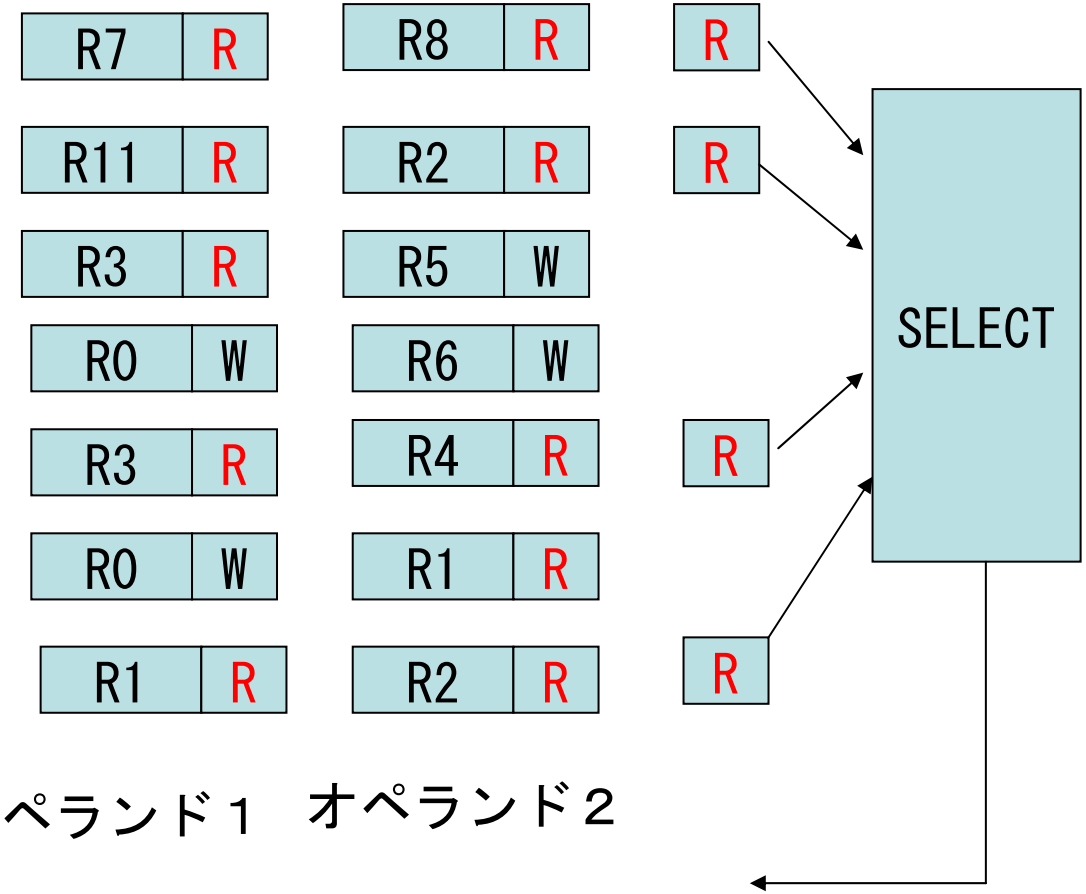
命令のWake UpとSelect

リザベーションステーションなどでの待ち合わせ

依存行列テーブル（DMT）法で解消

# 命令のWake UpとSelect：連想メモリ方式

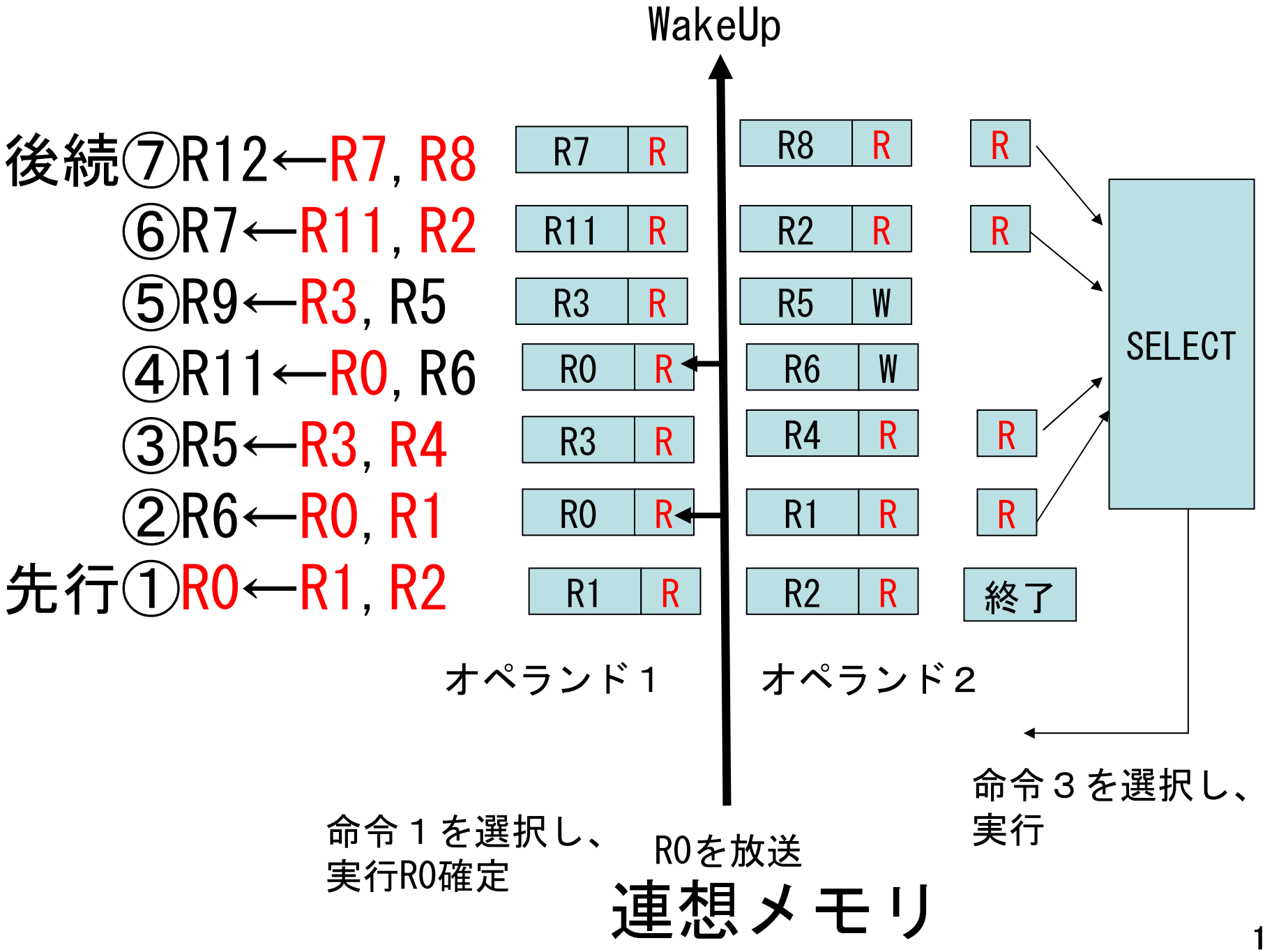
後続⑦R12←R7, R8
⑥R7←R11, R2
⑤R9←R3, R5
④R11←R0, R6
③R5←R3, R4
②R6←R0, R1
先行①R0←R1, R2




R:Ready

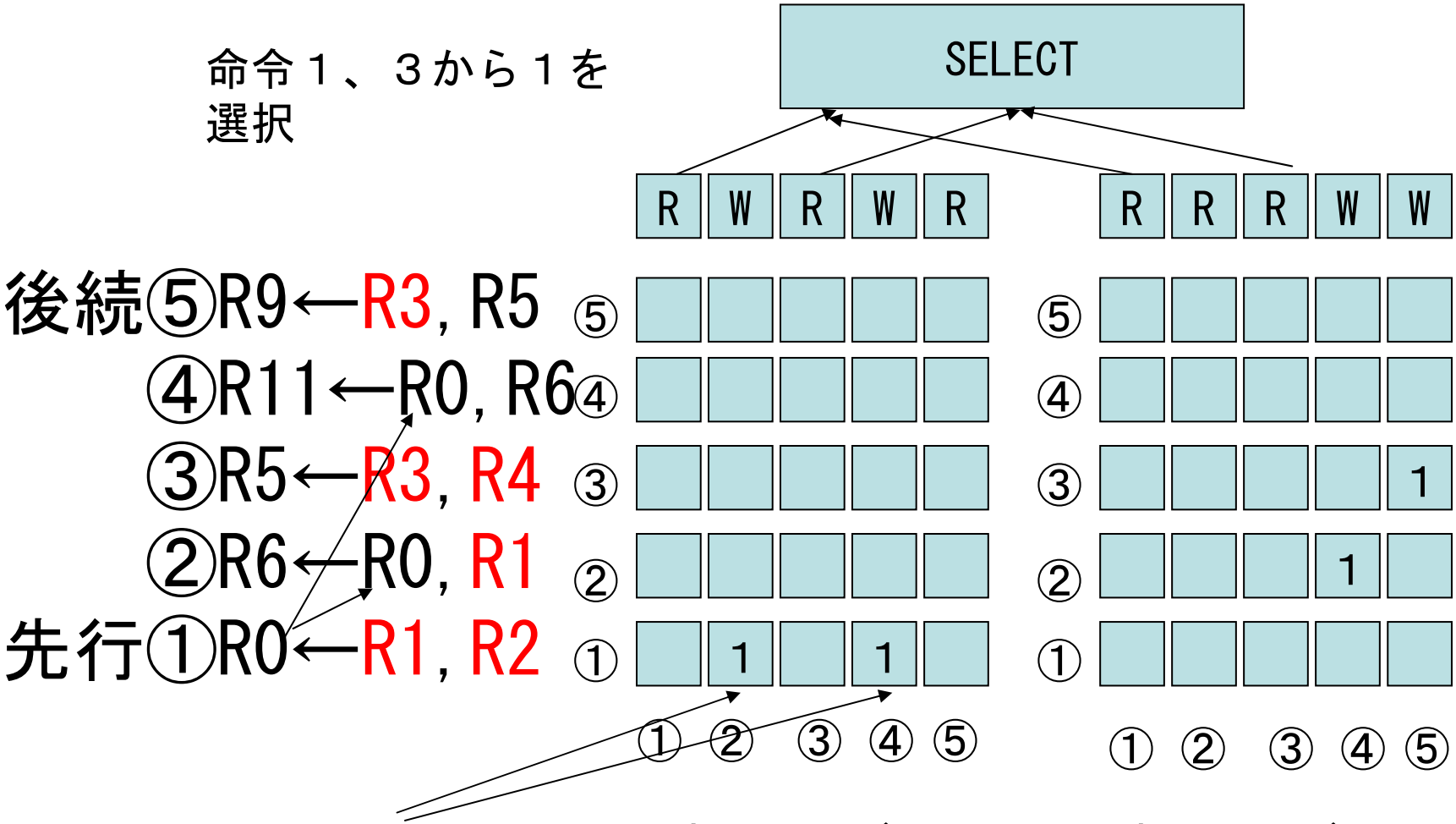
W:Wait

命令１を
選択し、
実行


WakeUp
後続⑦R12←R7, R8
⑥R7←R11, R2
⑤R9←R3, R5
④R11←R0, R6
③R5←R3, R4
②R6←R0, R1
先行①R0←R1, R2
R7
R
R8
R
R
R11
R
R2
R
R
R3
R
R5
W
R0
R
R6
W
R3
R
R4
R
R
R0
R
R1
R
R
R1
R
R2
R
終了
SELECT
オペランド１
オペランド２
命令１を選択し、
実行R0確定
R0を放送
命令３を選択し、
実行
連想メモリ

# 依存行列テーブル（DMT）法

命令１、３から１を選択

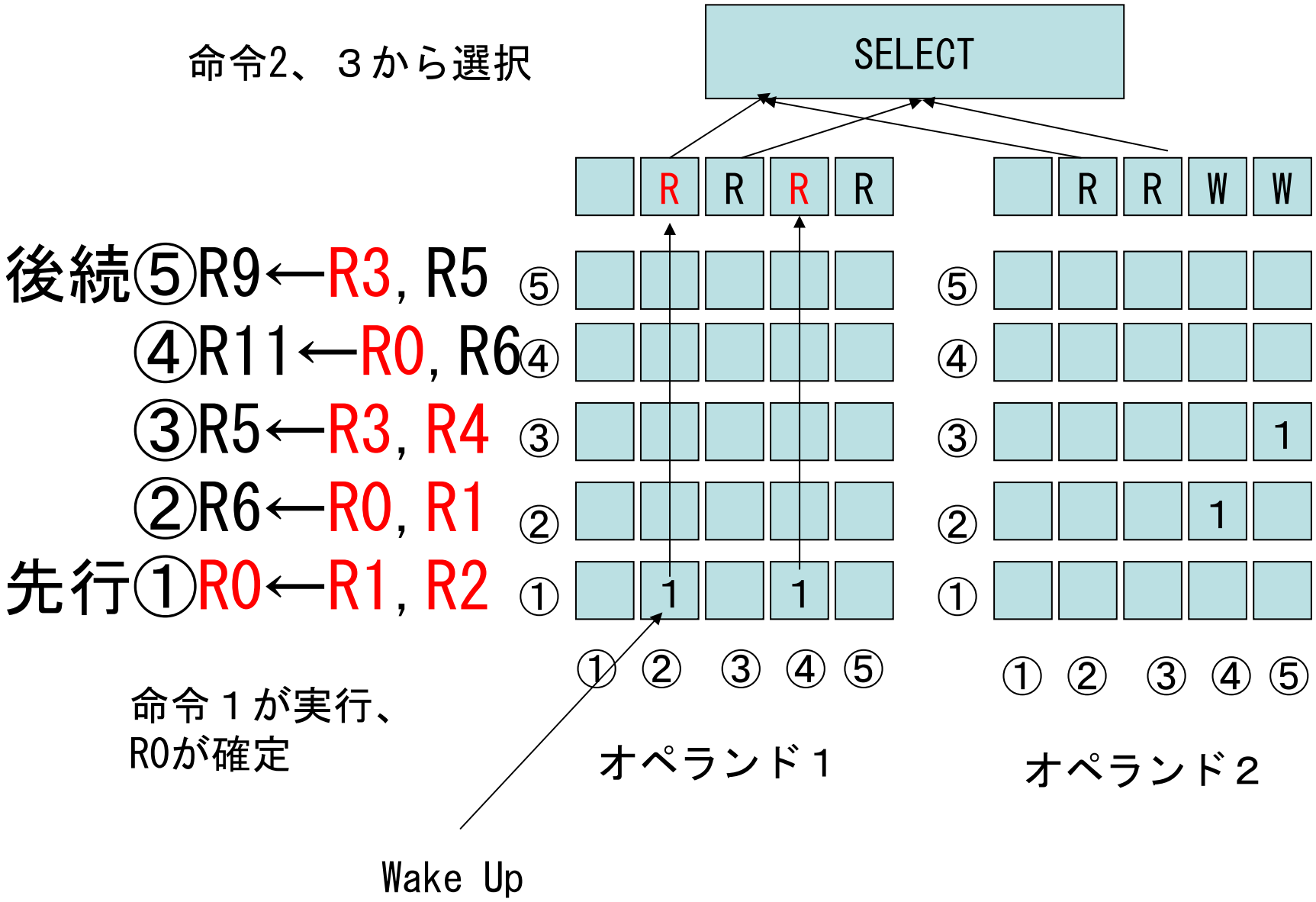



命令１の結果R0をオペランド１で使用する命令番号

オペランド１　　オペランド２

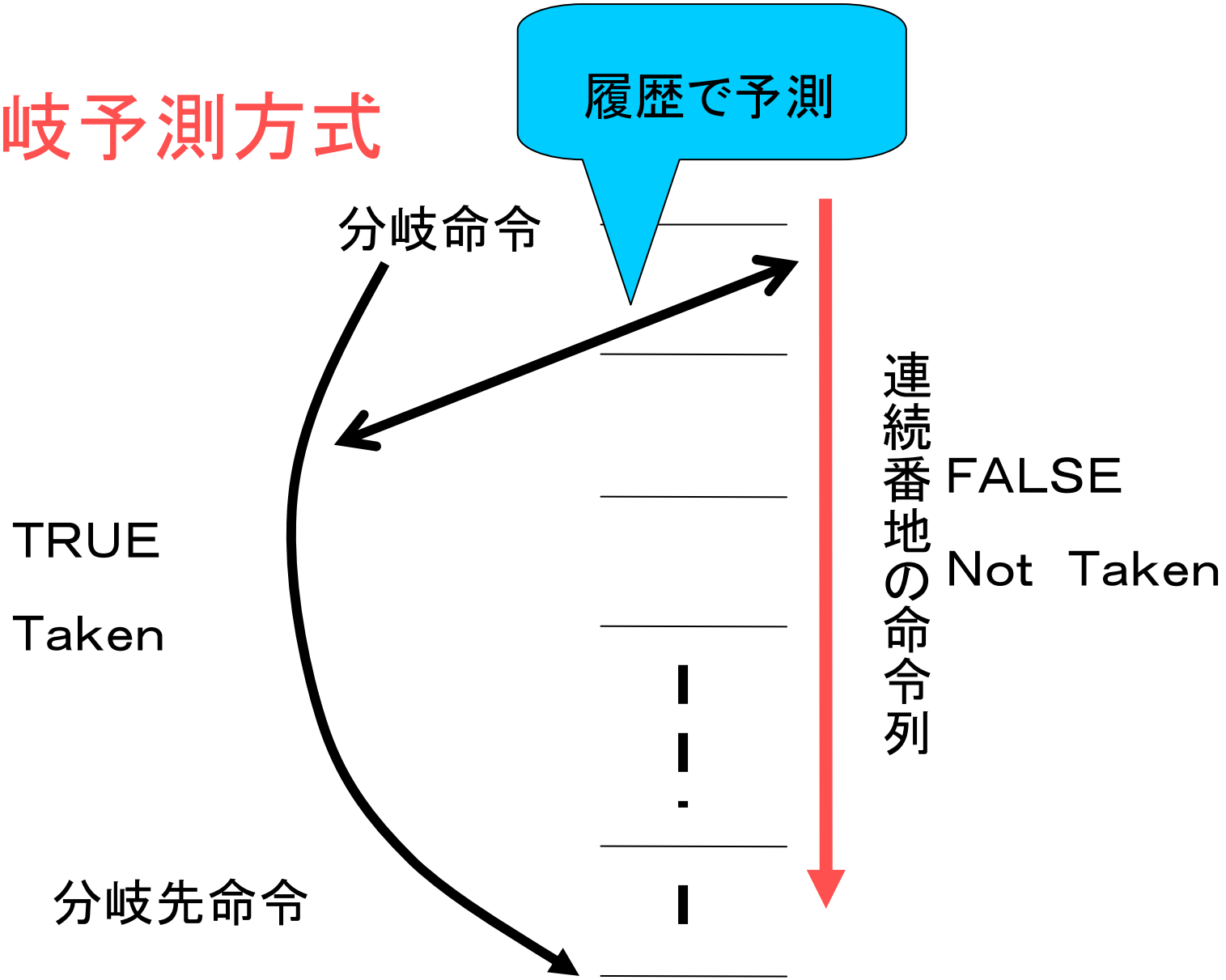
Goshima etal: A High Speed Dynamic Instruction Scheduling Scheme for Superscalar Processors, MICRO-34,pp.225-236,2001

命令2、３から選択


SELECT
R R R R
R R W W
後続⑤R9←R3, R5 ⑤
④R11←R0, R6 ④
③R5←R3, R4 ③
②R6←R0, R1 ②
先行①R0←R1, R2 ①
1 1
1
1
1
① ② ③ ④ ⑤
① ② ③ ④ ⑤
命令１が実行、
R0が確定
オペランド１
オペランド２
Wake Up

# （3）制御投機的実行：

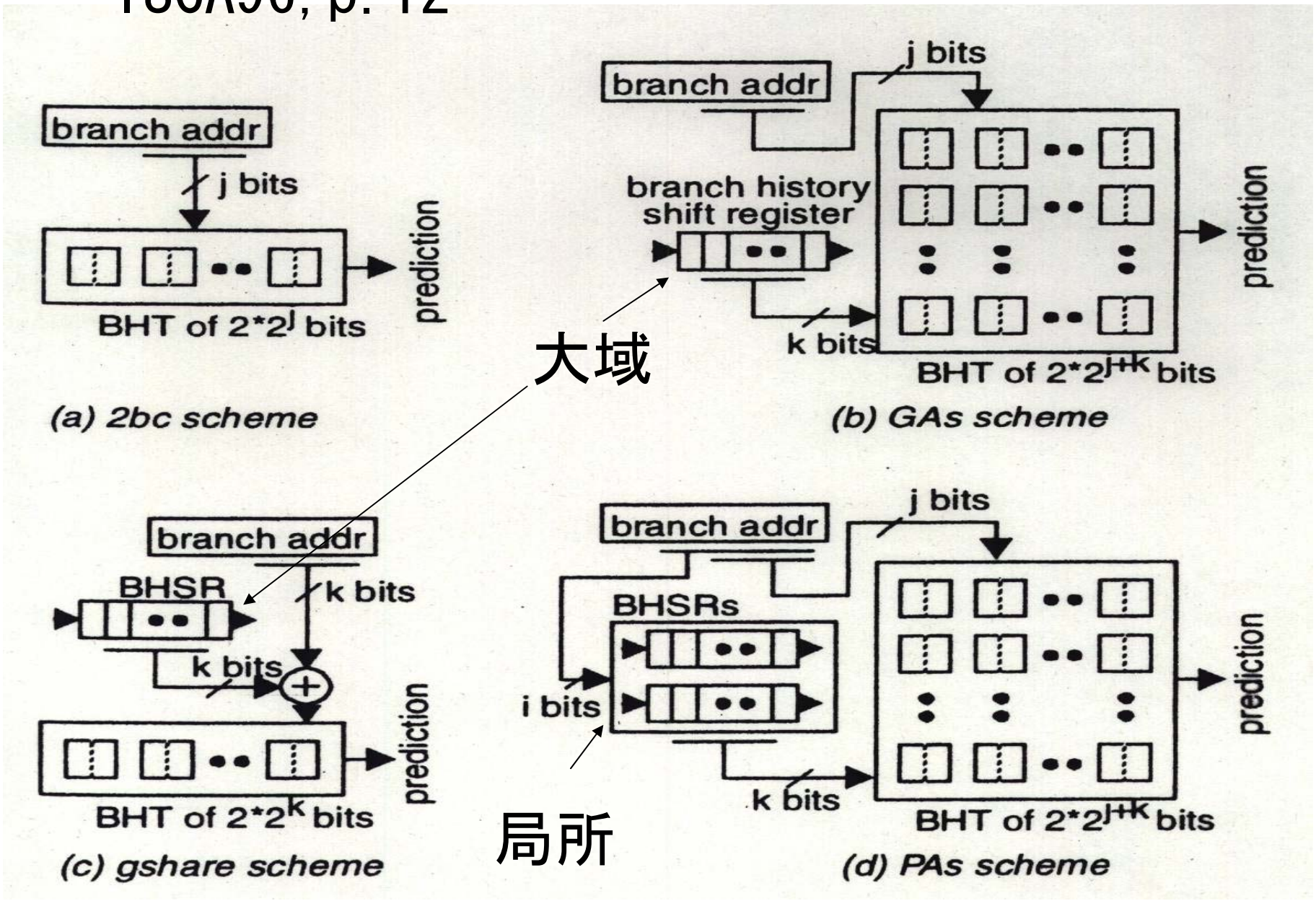
## ①分岐予測方式



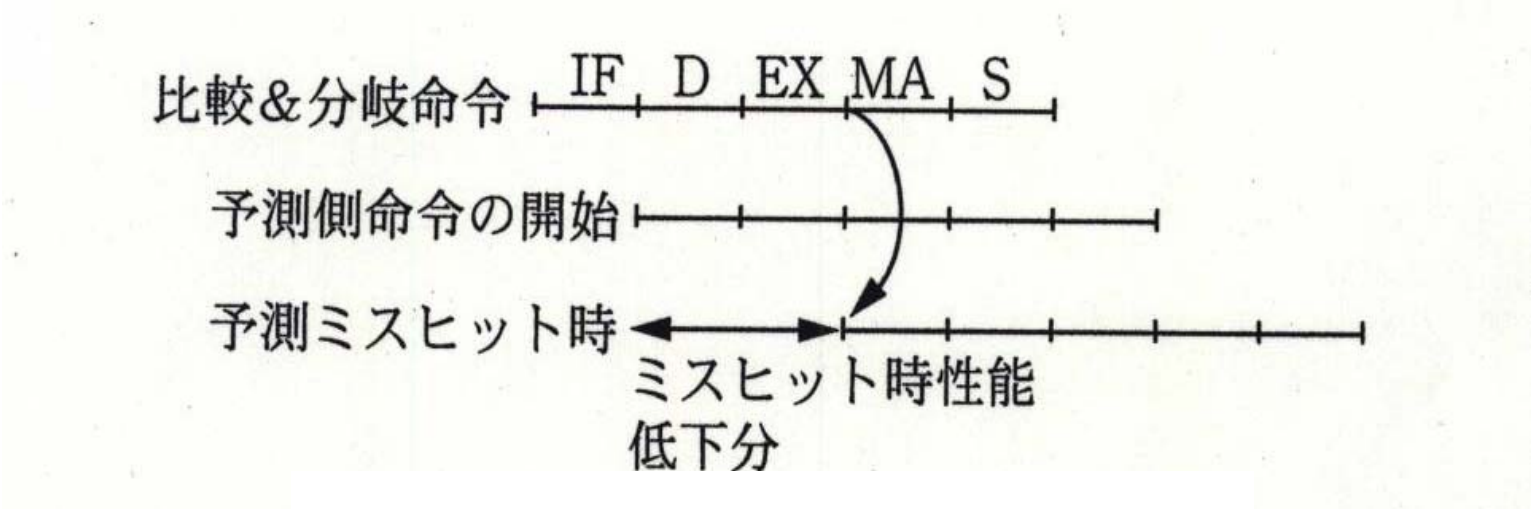
nビットカウンタ方式
Gshare方式
Gas方式
２レベル適応方式
パーセプトロン方式
ハイブリッド方式
　　局所＋大域
ヒット率：９０～９５％

**N.Gloy et al:An Analysis of Dynamic Branch Prediction Schemes on System Workloads, Proc. of ISCA96, pp.12-21(1996)**

ISCA96, p. 12




BHS:Branch History Table, BHSR:Branch History Shift Register




**分岐予測ミスによる性能低下**
**Cf=0.16*(0+2*0.1)=0.032**

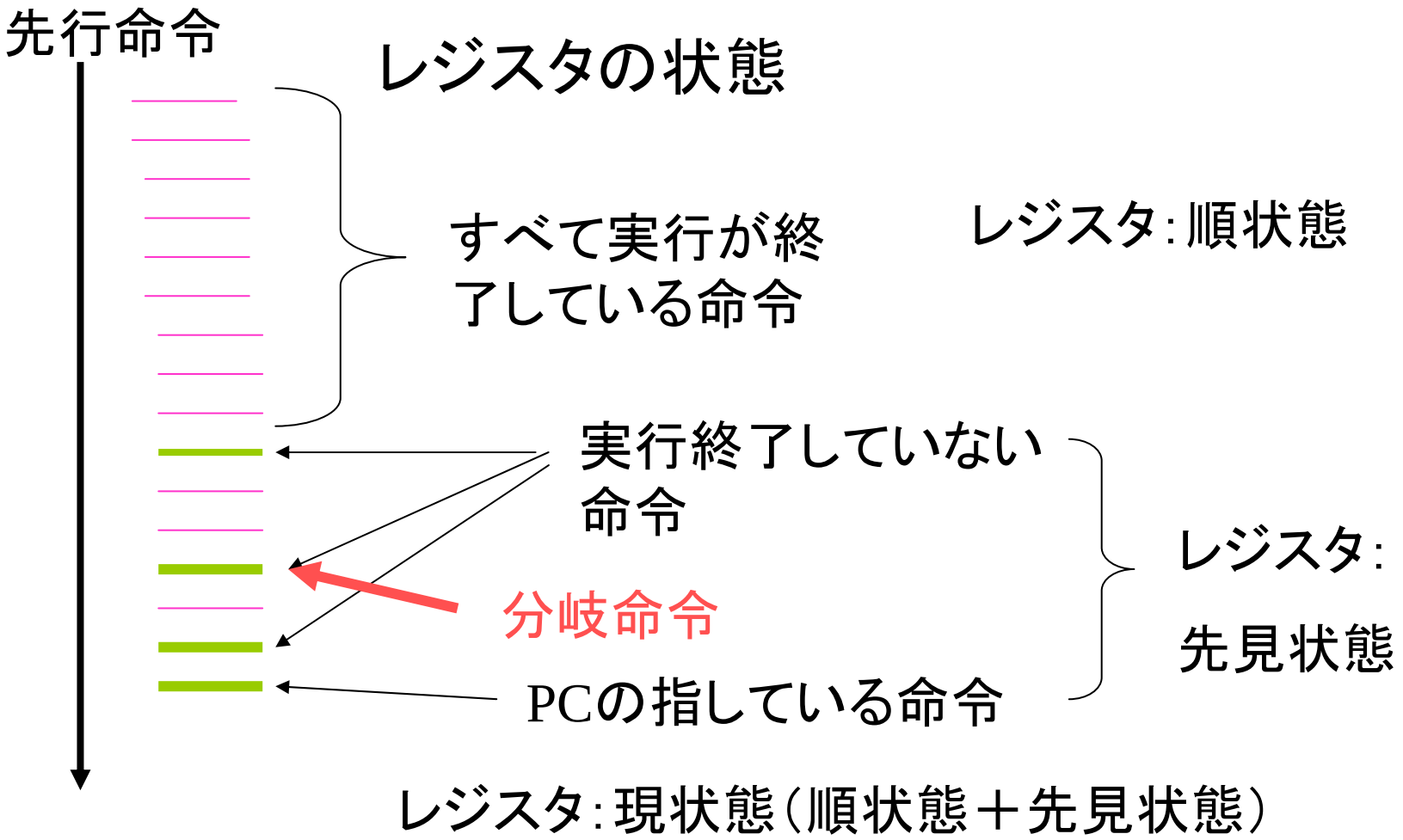
整数系分岐確率

分岐予測
ミス率

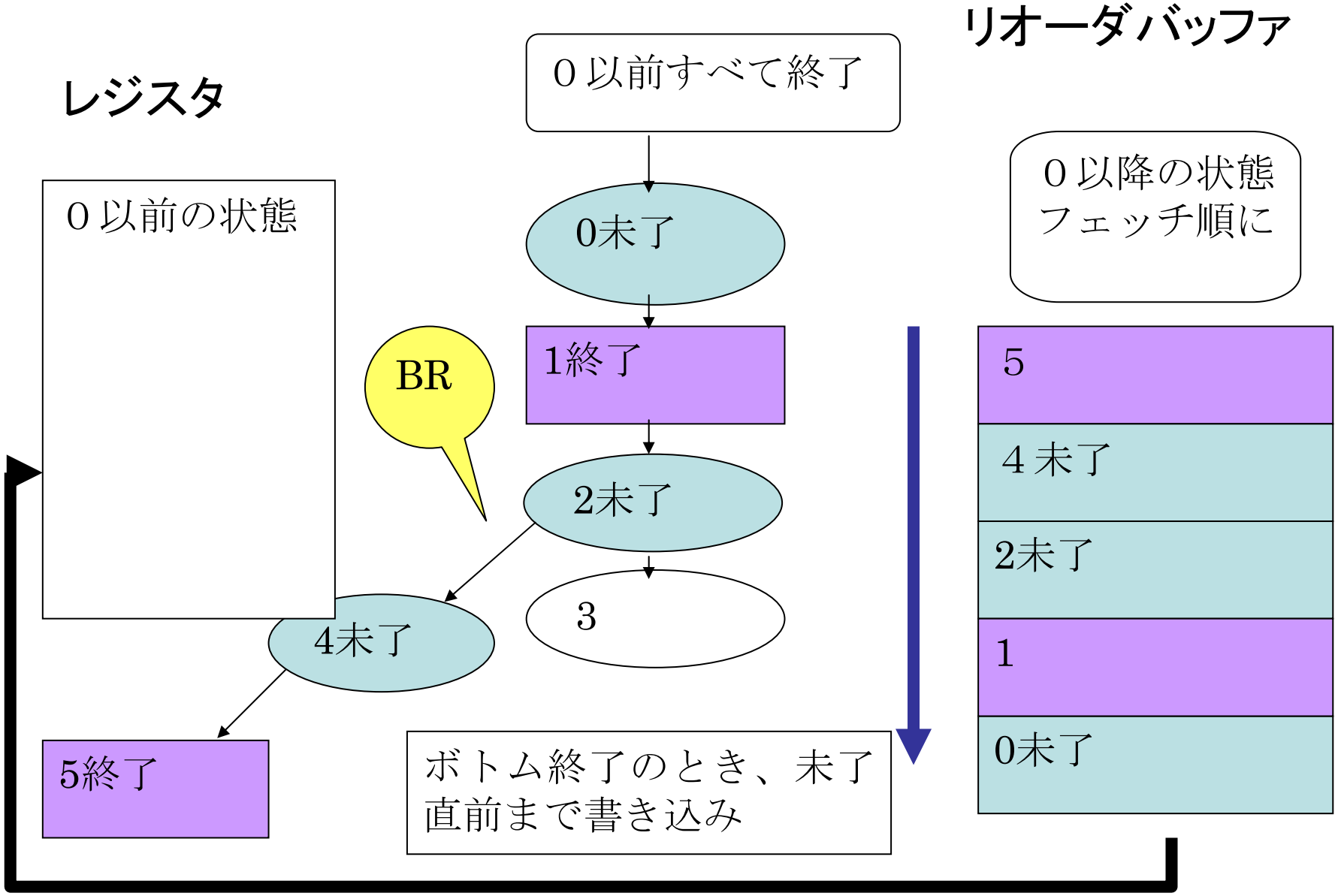
Pentium４では
２０
ペナルティ：0.35

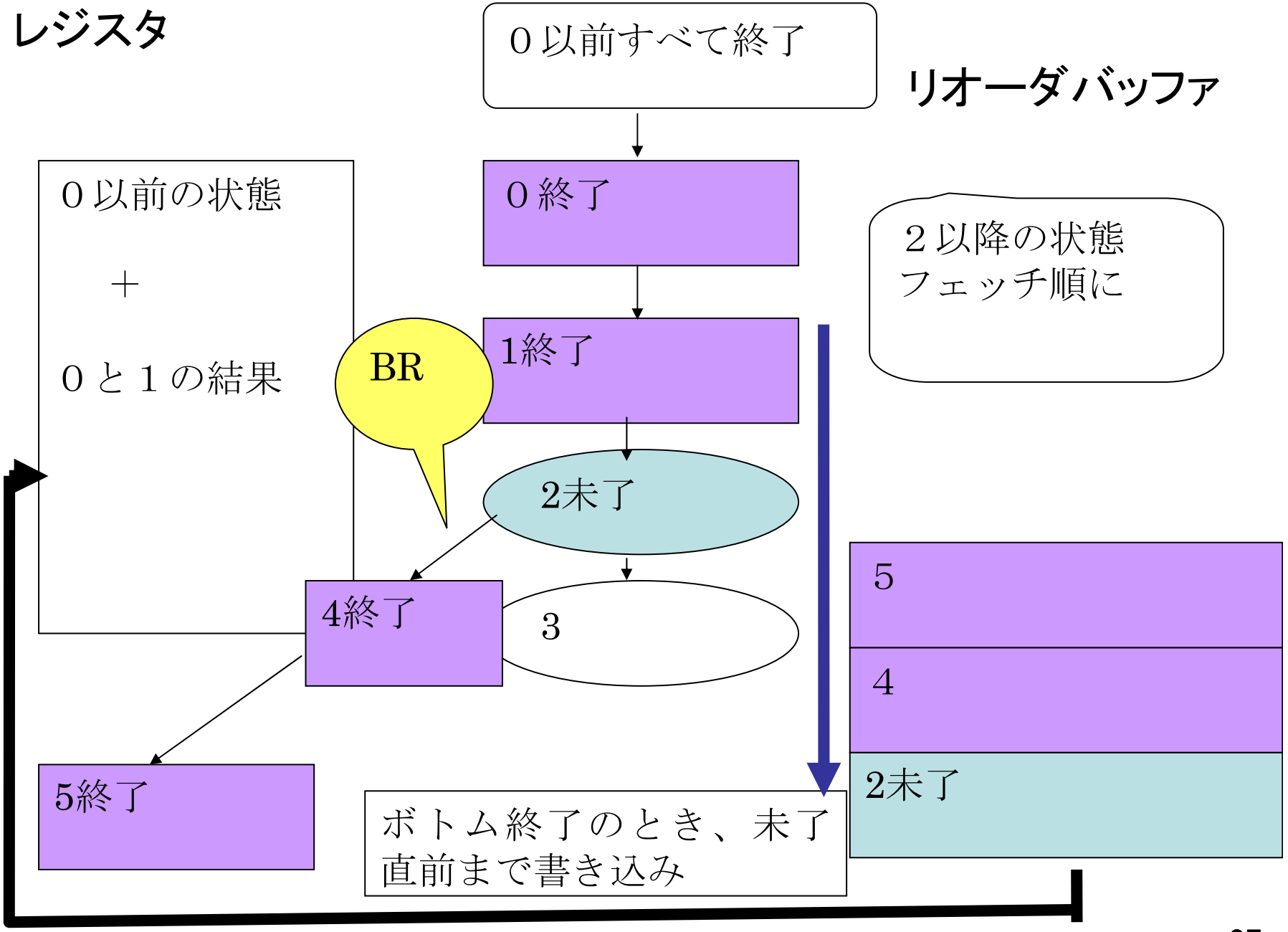
# ②復旧方法

J.E.Smith et al: Implementation of Precise Interrupts in Pipeline Processors,pp.36-44,ISCA,1985

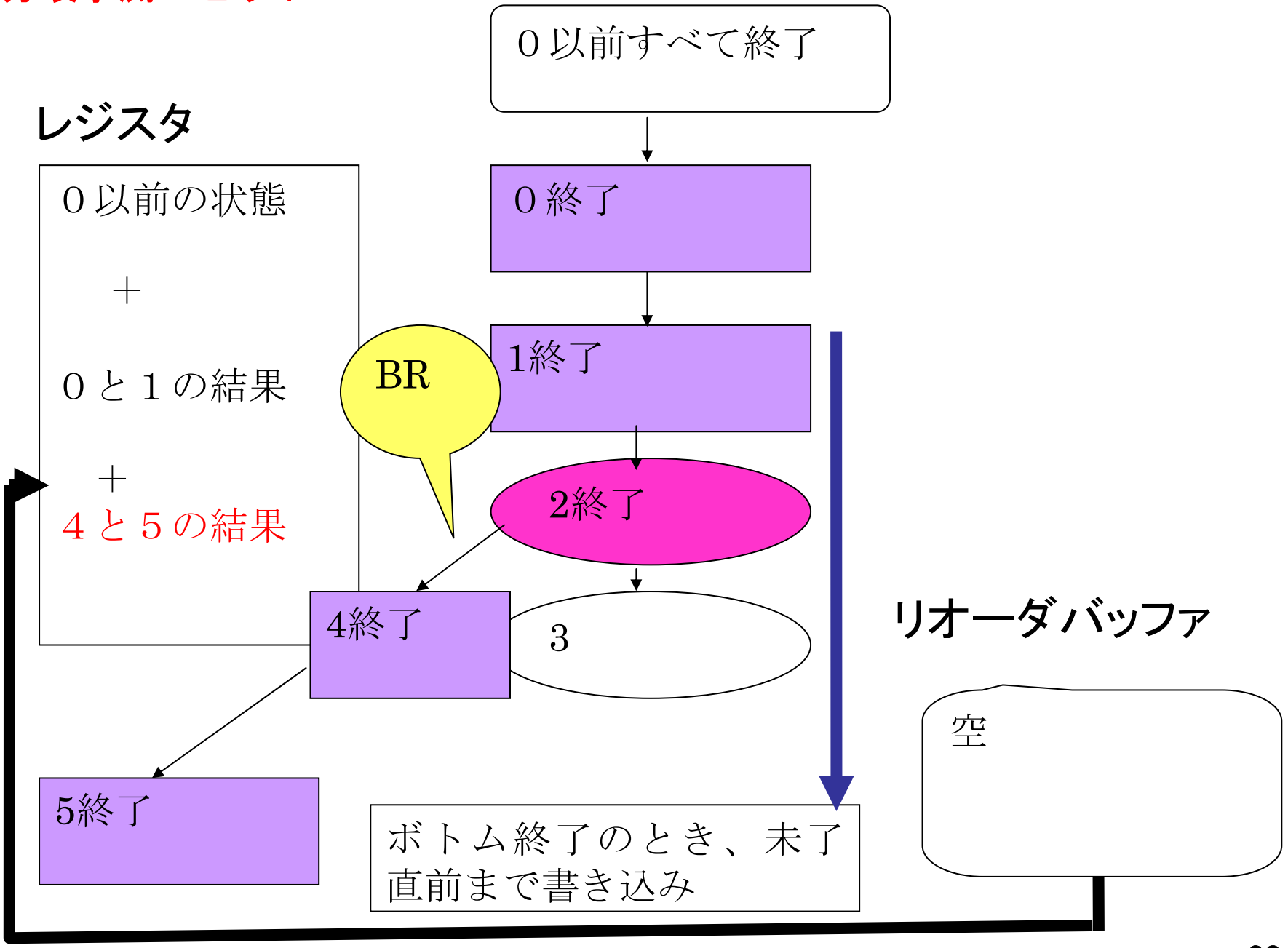


# 投機実行のメカニズム:リオーダバッファ法

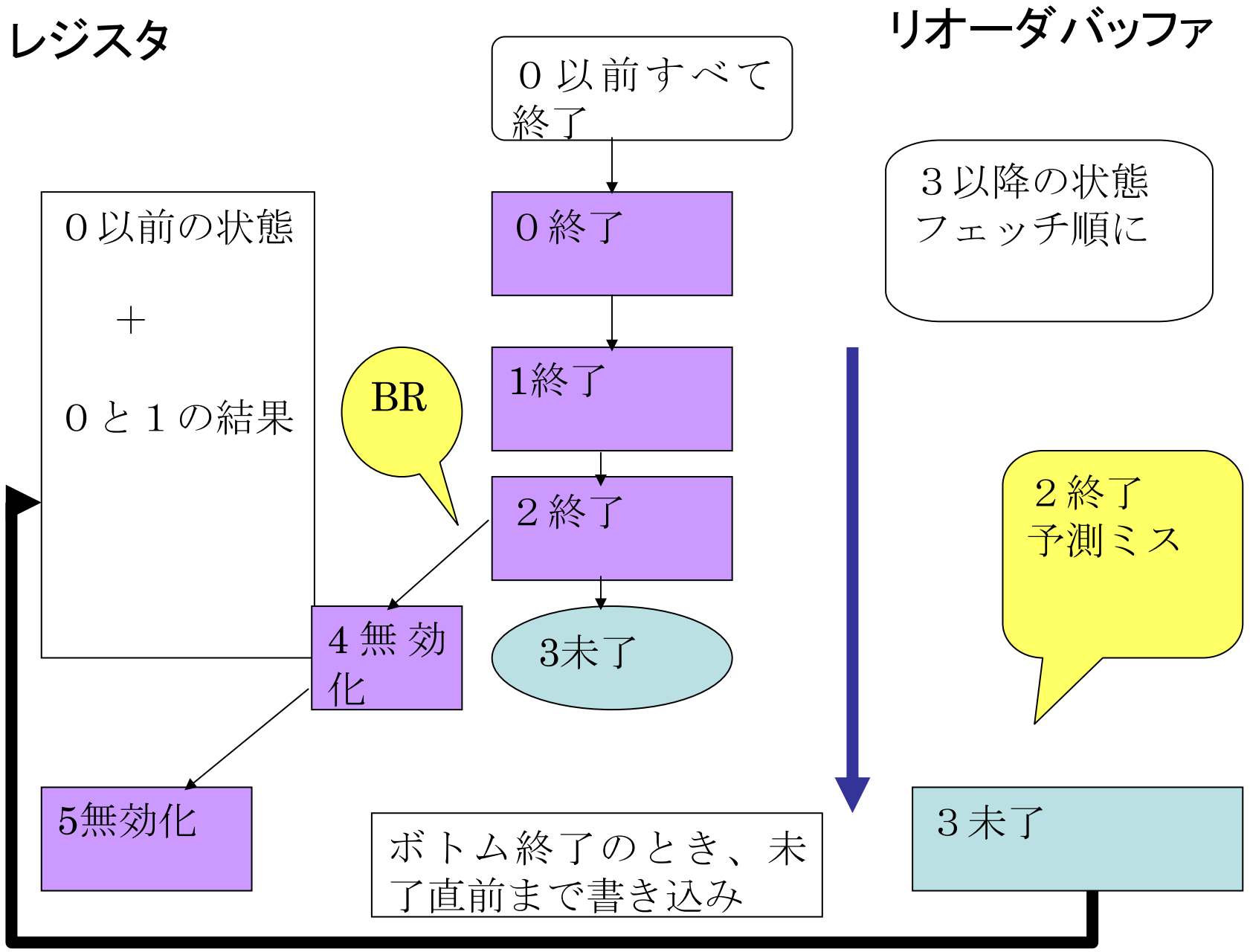



レジスタ
０以前すべて終了
リオーダバッファ
０以前の状態
＋
０と１の結果
０終了
１終了
BR
２以降の状態
フェッチ順に
2未了
4終了
3
5
4
2未了
5終了
ボトム終了のとき、未了
直前まで書き込み

# 分岐予測：ヒット


０以前すべて終了
レジスタ
０以前の状態
＋
０と１の結果
＋
４と５の結果
０終了
BR
1終了
2終了
4終了
3
5終了
リオーダバッファ
空
ボトム終了のとき、未了
直前まで書き込み

## 分岐予測：ミス


レジスタ
リオーダバッファ
０以前すべて終了
０以前の状態
＋
０と１の結果
０終了
1終了
２終了
3未了
BR
4 無効化
5無効化
３以降の状態フェッチ順に
２終了予測ミス
３未了
ボトム終了のとき、未了直前まで書き込み

# (3)局所空間並列の利用: PC近傍

## ①スーパスカラ

ルーツ:データフローコンピュータ
局所データフロー制御

ハードウェア指向:動的，実行時並列性検出

命令レベル互換性とスケーラビリティ

命令の並びと演算器間で多対多の
結合網必要

汎用プロセッサで利用

1990年代の主要技術

４命令同時フェッチして実行

M.S.Lam etal: Limits of Control Flow on Parallelism, pp.46-57, ISCA, 1992

# ②VLIW（Very Long Instruction Word）

ルーツ：水平型マイクロプログラム制御方式

コンパイラ指向： 静的、コンパイル時並列性検出

長命令間および内にNOP操作

互換性とスケーラビリティの欠如

1970年代研究

QA-1/2,Trace, AP-120B, Cydra5

1990年代：メディアプロセッサ, ベクトルプロセッサで利用

2000年代：IntelのItaniumに採用

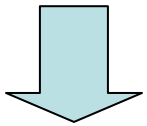

- A.E.Charlesworth: An Approach to Scientific Array Processing, The Architectural Design of the AP-120B/FPS-164 Family, IEEE Computer 14,9,pp.18-27(1981)
- J.A.Fisher:Very Long Instruction Word Architecture and ELI-512, pp.140-150,ISCA,1983
- S.Tomita etal: A User Microprogrammable, Local Host Computer with Low-Level Parallelesim, pp.151-157, ISCA,1983
- M.S. Schlansker et al:EPIC: Explicitly Parallel Instruction Computing, IEEE Computer, Feb. pp.37-45(2000)

# ３　マルチコア型プロセッサ

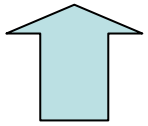
## ①大域並列の利用

・パイプライン：時間並列
・乱実行、投機実行による時間並列の高速化
・局所並列：スーパスカラ、VLIW
・非常に複雑な構造
　　IPCの頭打ち

プログラムカウンタの近傍にある命令の並列実行



・大域並列：マルチコア型プロセッサ



・周波数向上が困難
　深いパイプラインで性能向上：小さくなった

## ②省電力化

　動的消費電力∝$f^3$
　リーク電流の増大

## ③超高信頼、セキュアなプロセッサ

キャッシュ、分岐予測ミス
ステージ内ゲート段数

# 3-1　諸制約

## ①ステージ内ゲート段数

周波数向上：40%/年

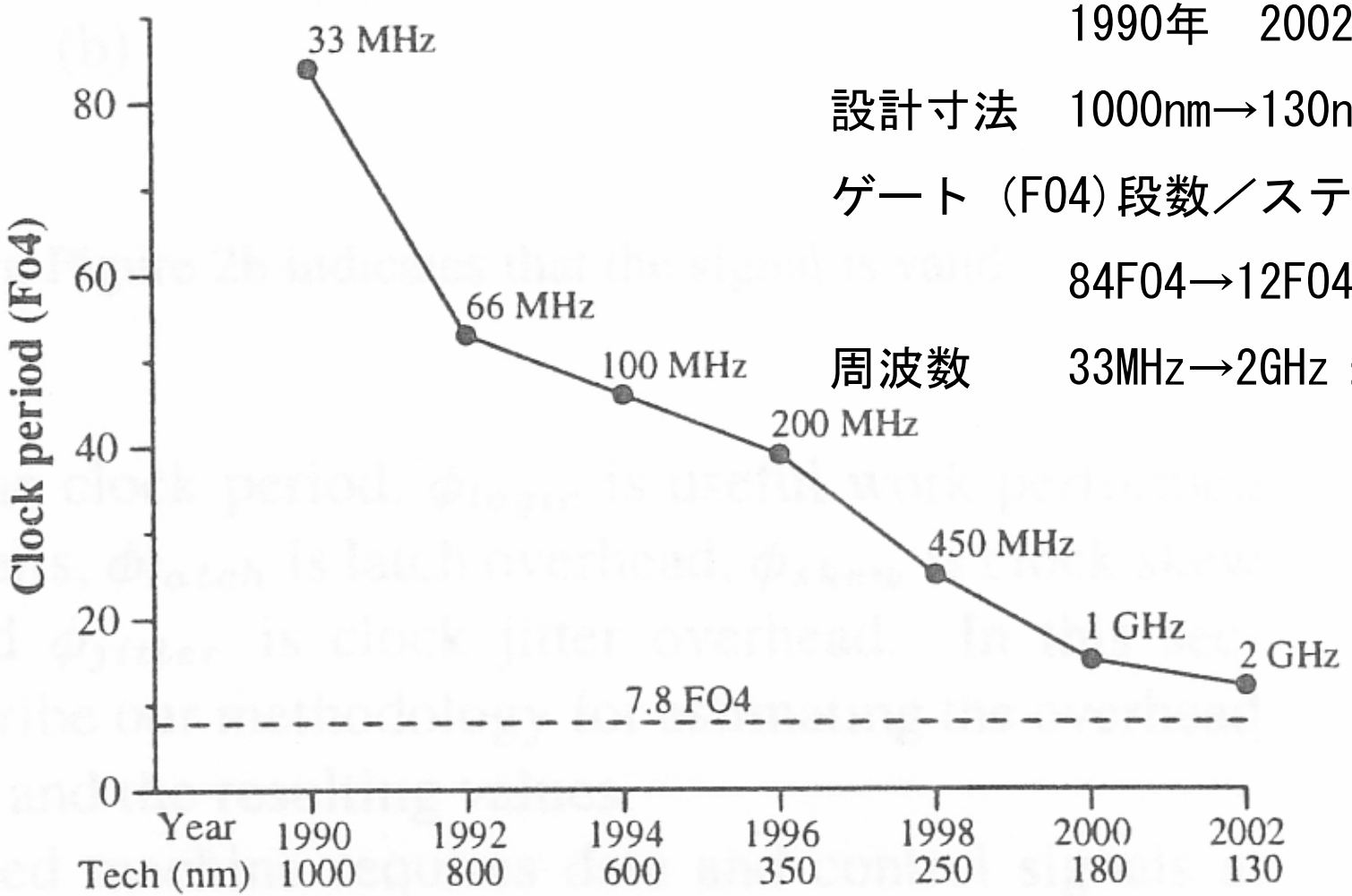
1990年　2002年

設計寸法　1000nm→130nm：1/8

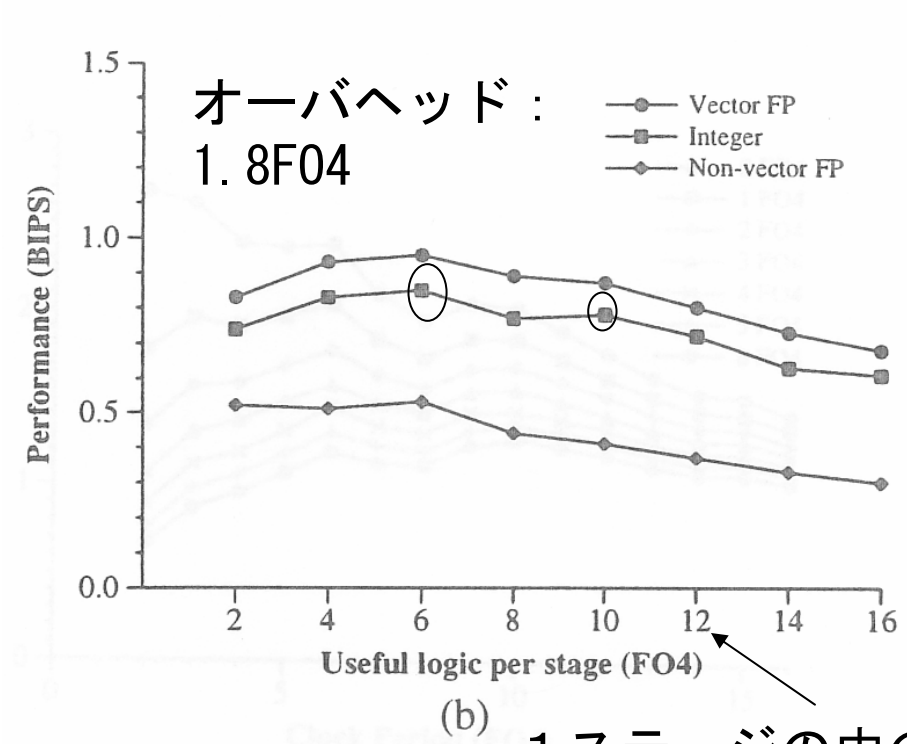
ゲート（FO4）段数／ステージ

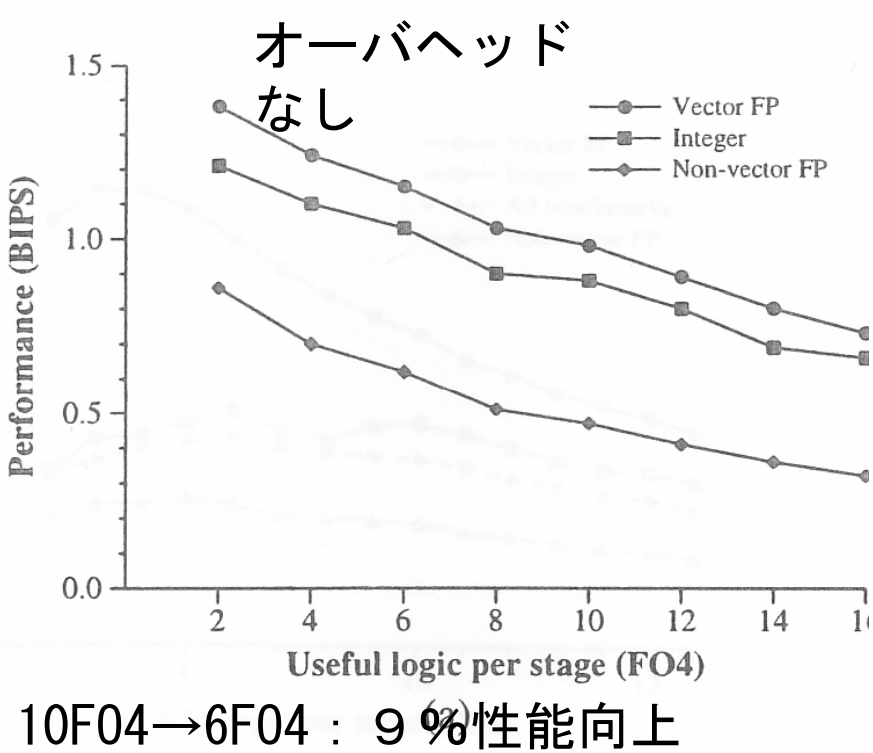
84FO4→12FO4：1/7

周波数　33MHz→2GHz：60倍




N. Jouppi et. al., The Optimal Logic Depth Per Pipeline Stages is 6 to 8 FO4 Inverter Delays, pp. 14-24, ISCA, 2002

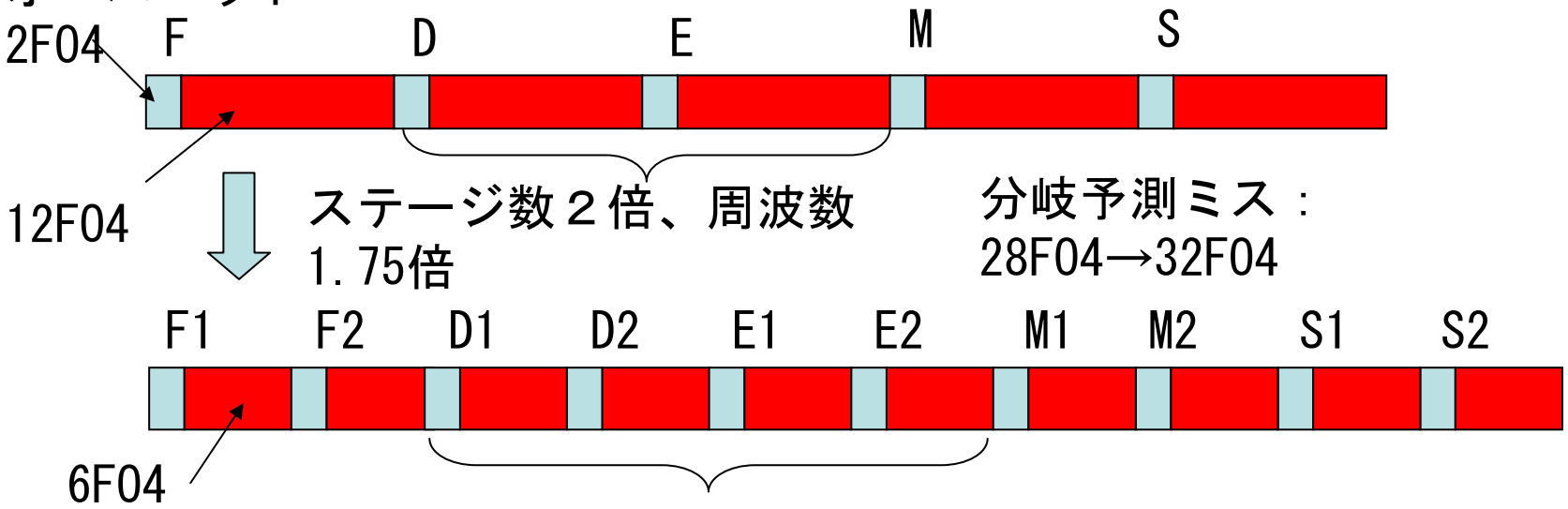



１ステージの中の
有効ゲート数

10F04→6F04：９％性能向上

11.8F04→7.8F04：周波数1.5倍



# ②省電力化

**CMOSの電力消費**

- **動的**

  **回路がON、OFFするとき　$\alpha f C V^2$**

- **漏れ電流　　$V I_{leak}$**
- **貫通電流　　$\alpha f t_{st} I_{short} V$**

**$\alpha$：ゲート動作率、f：周波数、C：ゲート総容量、V：電源電圧、$t_{st}$：スイッチング時間**

$$P = \alpha f C_L V^2 + V I_{leak} + \alpha f t_{st} I_{short} V$$

$$F_{max} \propto (V - V_{threshold})^2 / V \fallingdotseq V$$

$$I_{leak} \propto \exp(-qV_{threshold}/kT)$$

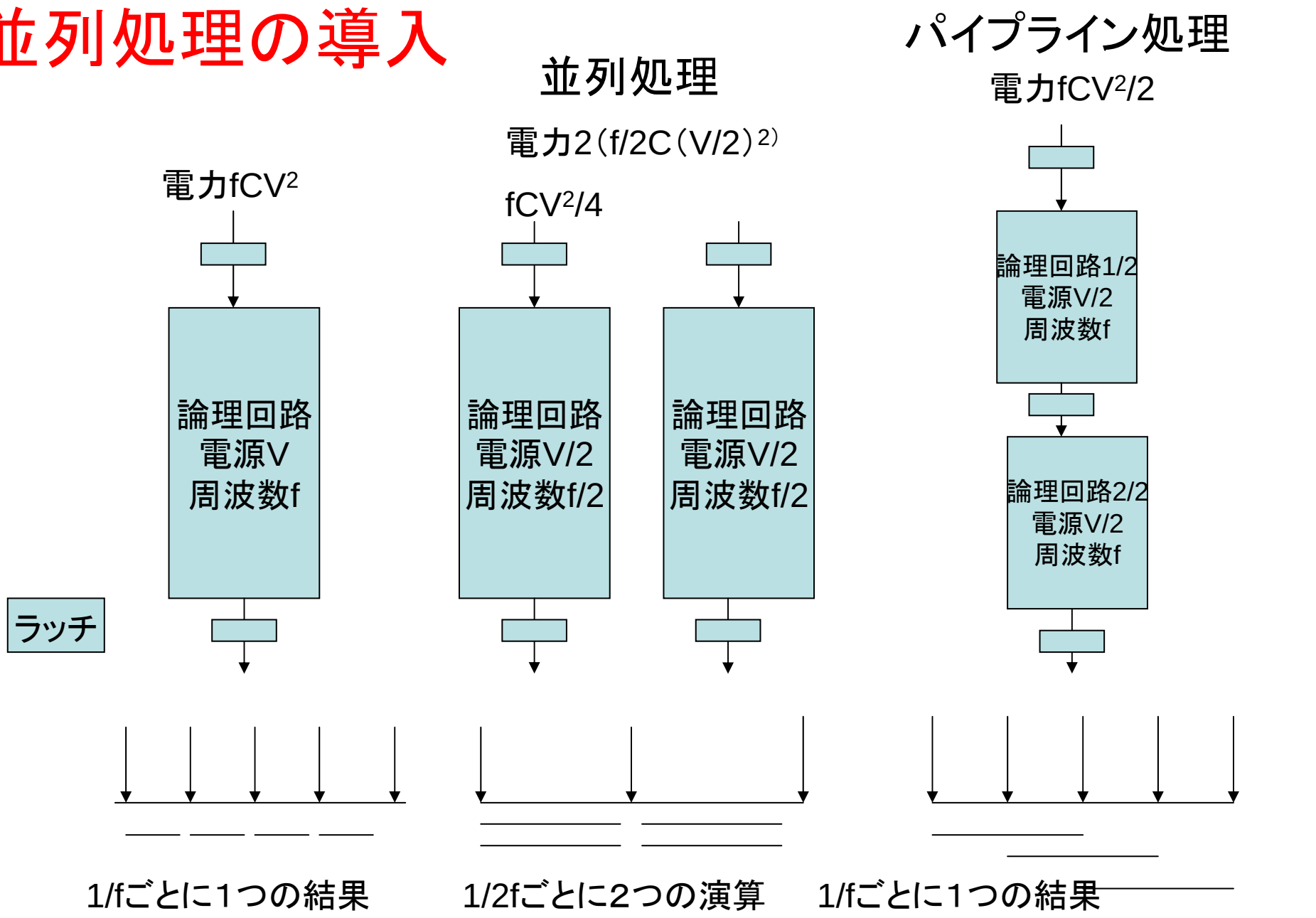
T.Mudge: Power: A First-Class Architectural Design Constraint, IEEE Computer, pp.52-58, April 2001

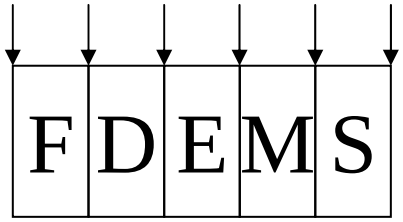
# 基本的な考え方

- スイッチング回数を少なくする
- 動作をしない（と予想される）回路には
  クロック供給しない
  電源を供給しない
- 電源電圧を制御して、必要十分な処理速度で実行
- 電源電圧、周波数を落として並列処理、パイプラインで行う
- 基盤バイアス印加による閾値制御：リーク電流削減
  高速部分：低閾値、
  低速部分や待機時：高閾値（バイアス印加）

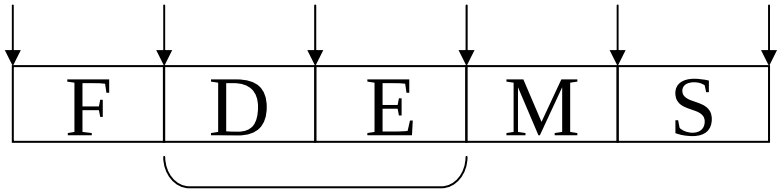
- デバイスレベル
  低電源電圧化、低ゲート容量化
- 回路レベル
  パストランジスタ論理
  ゲート付きクロック
  グリッチの削減
  動的電源遮断
  非同期回路
- アーキテクチャレベル
  データパスの最適化：必要な演算幅の決定など
  並列処理、パイプライン処理
  キャッシュメモリ
  バス: アドレスの反射2進符号化、データ圧縮
- OS、コンパイラ、アルゴリズムレベル
  符号化
  ビット変化の少ないコード生成
  動的電源電圧制御
  動的周波数制御

# 並列処理の導入


並列処理
パイプライン処理
電力fCV2
電力2(f/2C(V/2)2)
電力fCV2/2
fCV2/4
論理回路
電源V
周波数f
論理回路
電源V/2
周波数f/2
論理回路
電源V/2
周波数f/2
論理回路1/2
電源V/2
周波数f
論理回路2/2
電源V/2
周波数f
ラッチ
1/fごとに1つの結果
1/2fごとに2つの演算
1/fごとに1つの結果



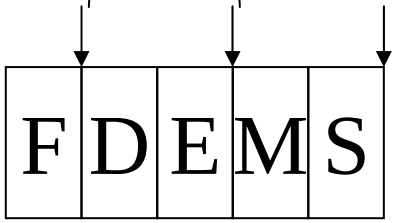

周波数f、電源V：電力：$fV^2$




ボルテージスケーリング

周波数f/2、電源V/2：電力：1/8

分岐ミスのとき



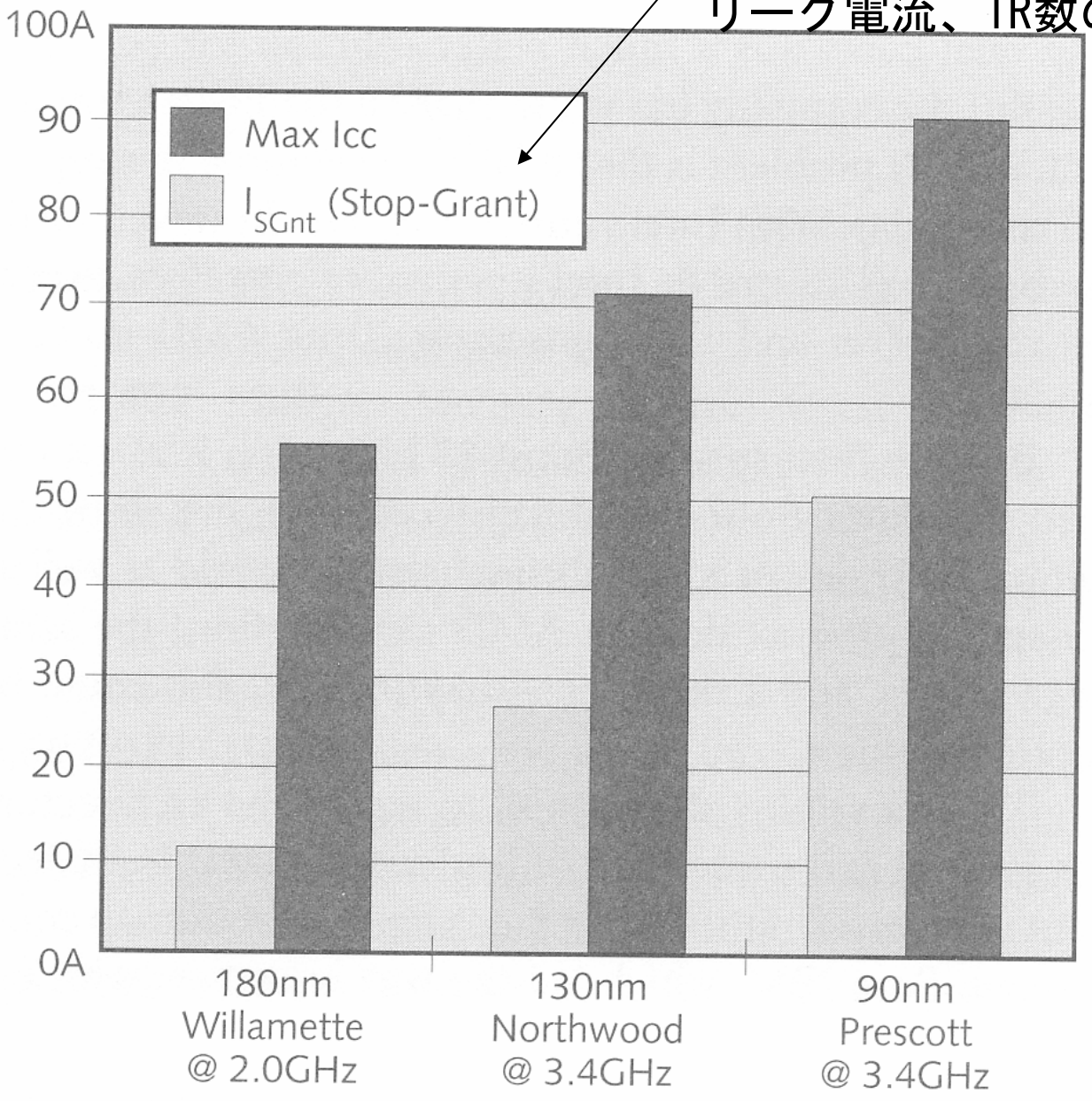

パイプラインステージ統合

周波数f/2、電源V：電力：1/2

低電源電圧化が困難なとき、有効（リーク電流）

アイドル状態での電力消費

リーク電流、TR数の増大




Microprocessor Report

May 2004

# ③超高信頼、セキュアなプロセッサ

高信頼、セキュアなプロセッサ

要因

・製造バラツキ

・ソフトエラー

・経年変化

・セキュリティ・アタック

高信頼性

- デバイスレベル
- 論理回路レベル

  パリティ、ECC、ラッチ／フリップフロップの２重化
- アーキテクチャレベル

  機能装置の２重化、マルチスレッドでの２重実行、
  データパスの２重化
  ランタイムチェッカ
  イリノイ大学RSE（Reliability and Security Engine）

高セキュア化

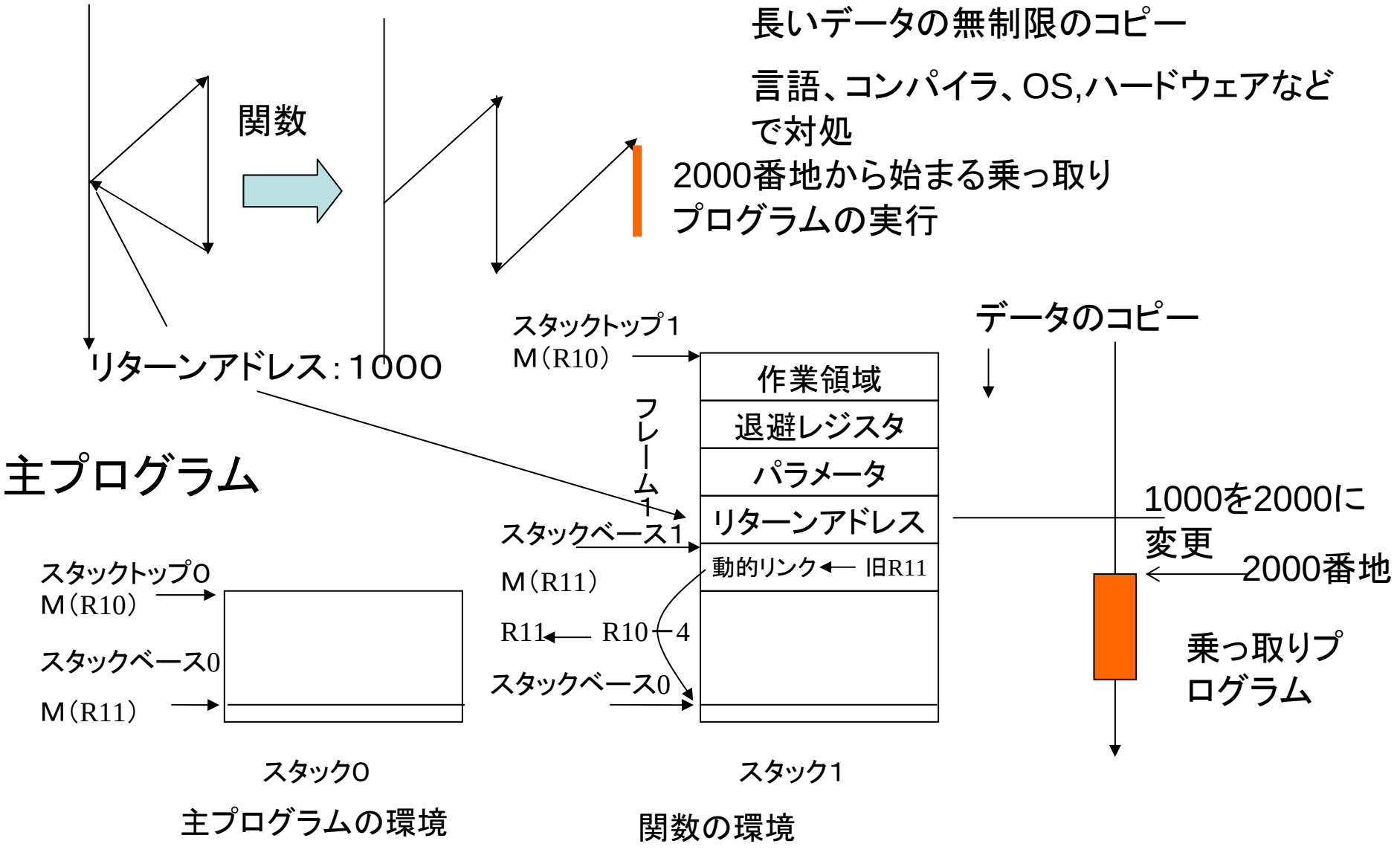
- 暗号化
- スタックオーバフローアタック対策

B.A.Kuperman etal: Detection and Prevention of Stack Buffer Overflow, Vol.48,No.11,pp.51-56,CACM, 2005

# ソフトウェアの脆弱性の例　バッファーオーバフロー問題

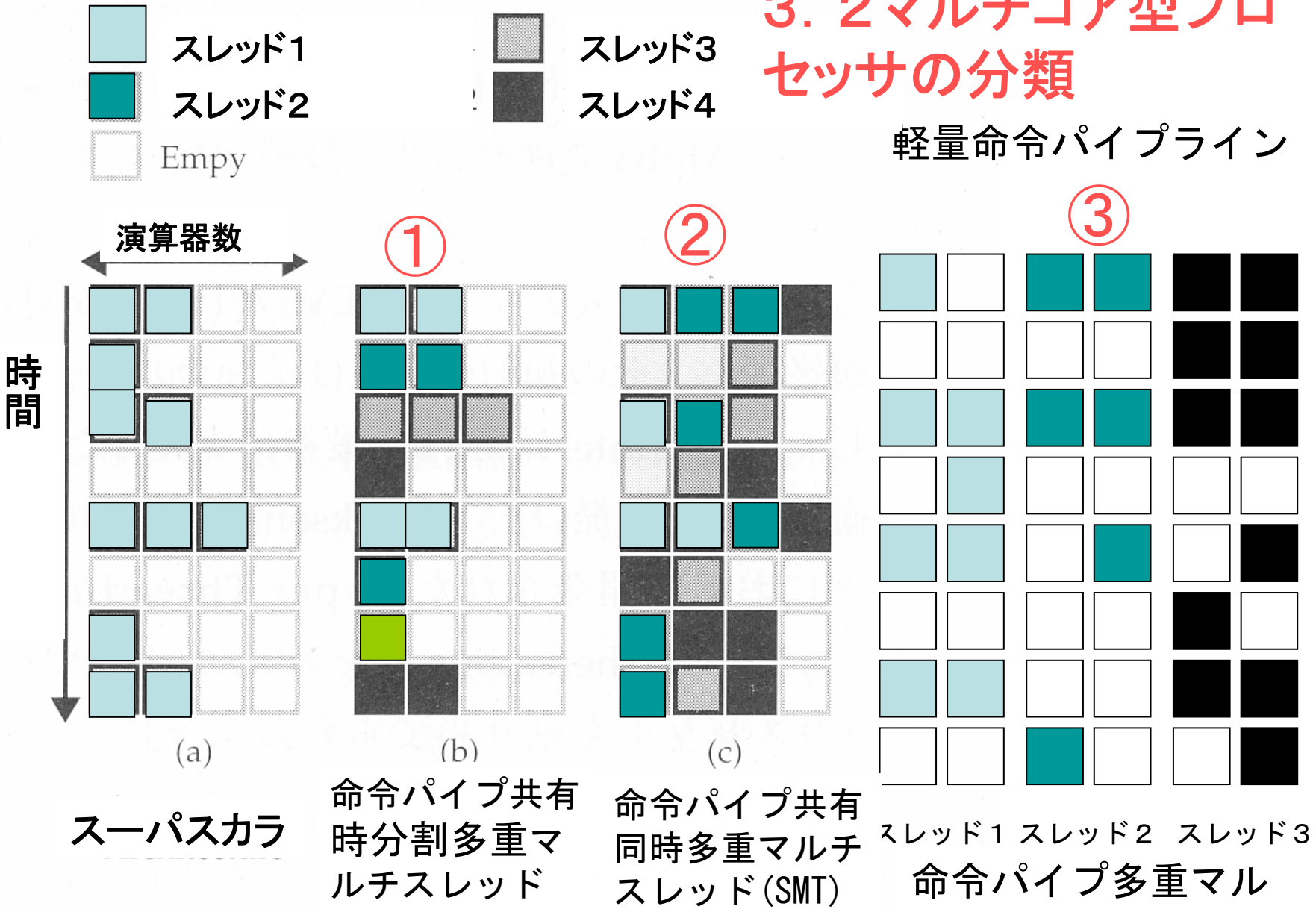
長いデータの無制限のコピー

言語、コンパイラ、OS,ハードウェアなどで対処




# 関数呼出しの制御

# 3．2マルチコア型プロセッサの分類




スーパスカラ

命令パイプ共有時分割多重マルチスレッド

命令パイプ共有同時多重マルチスレッド(SMT)

命令パイプ多重マルチスレッド／CMP

均質Vs非均質、汎用Vs専用

①命令パイプ共有時分割多重マルチスレッド

・原型：HEP(B.Smith,Tera Computer):1974

②SMT　Simultaneous　Multithreading

Intel Hyper Threading：15−25%性能向上、
チップサイズ5%増、2スレッド実行

H.Hirata etal: An Elementary Processor Architecture with Simultaneous Instruction Issuing from Multiple Threads ISCA−19,pp.136−145,1992,

③オンチップマルチプロセッサ　CMP

・IBM POWER4：2台のSMP、スヌープキャッシュ

・Pentium EE840デュアルコアプロセッサ、

・Itanium2-Montecito：2個のItanium2プロセッサ、2スレッド、時分割多重マルチスレッド命令パイプ共有

・SPARCNiagara：2多重の簡単なSPARCプロセッサ　8台
各プロセッサ：4スレッド、時分割多重マルチスレッド命令パイプ共有、クロスバー網

・Intel Core 2 Duo　(2006 7 27発表)：デュアルコア、14段パイプライン、65nmデザインルール、29,100万Tr、L2：4MB

# Intel Core 2 Duo 2006 7 27発表

デュアルコア

14段パイプライン

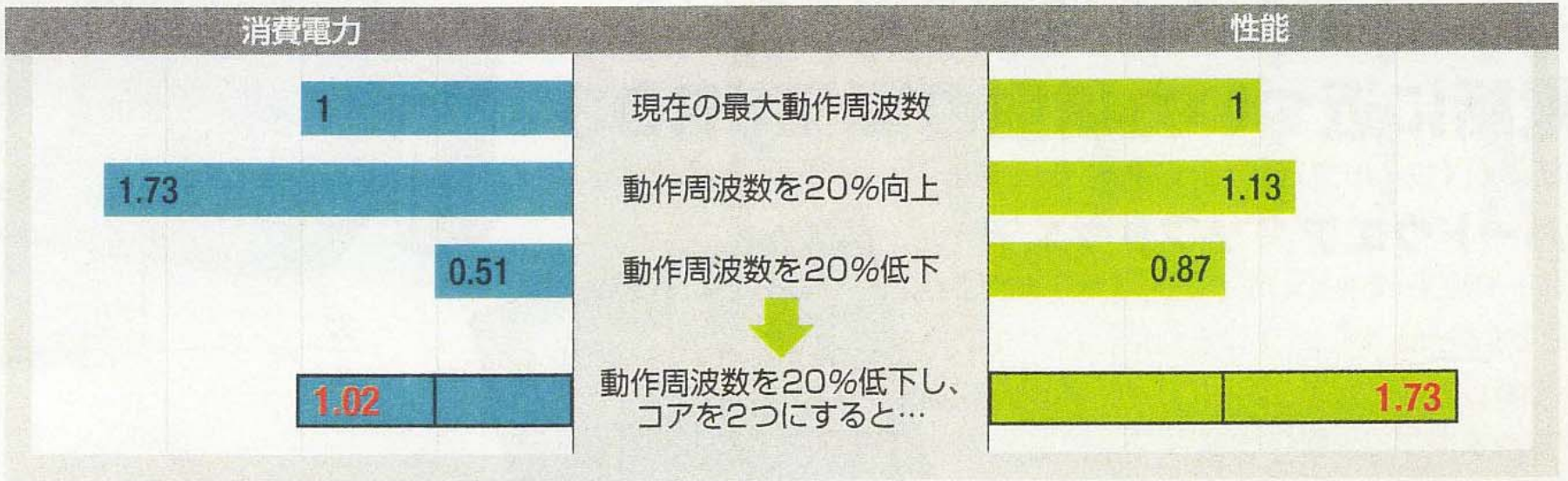
65nmデザインルール、29,100万Tr、L2:4MB

●Coreマイクロアーキテクチャーが採用した技術

| 技術名 | 概要 |
|---|---|
| ワイド・ダイナミック・エグゼキューション | 1クロック当たりの命令実行数を向上させる技術。同時実行可能な命令数を4つに増やした。また、連続して使われることの多い命令を合体させ、まとめて実行する機能（マクロフュージョン）も搭載 |
| インテリジェント・パワー機能 | CPUの各部分の消費電力をリアルタイムで管理し、あまり利用されていない部分の電力を最小限に抑える |
| アドバンスト・スマート・キャッシュ | 2つのコアが、1つの2次キャッシュを共有する。一方のコアが使ったデータをもう一方のコアが使う場合、別途メモリーにアクセスする必要がなくなるなどのメリットがある |
| スマート・メモリー・アクセス | データの書き出しと読み込みの命令が続いた場合、書き出しの命令を実行する前に、次の命令が利用するデータを読み込む。またデータの利用パターンを分析・予測することで、必要とされるであろうデータを事前にキャッシュに取り込んでおくことも可能 |
| アドバンスト・デジタル・メディア・ブースト | 画像や暗号の処理など高度な数値計算で用いられる、SIMD拡張命令（SSE）の処理効率を、従来の倍に向上させた |

## ●Core 2 Duoの設計思想

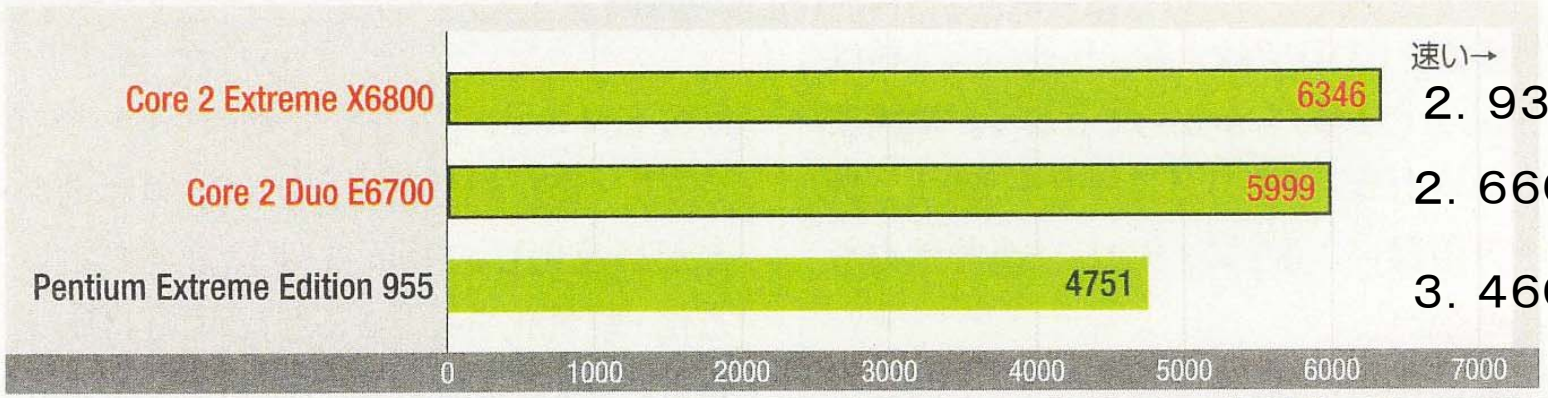
現在の動作周波数での性能と消費電力を1としたときの相対値で表示




インテルが開発者会議で発表した値。現在の動作周波数からさらに20%周波数を上げても、性能は1.13倍にしかならないが、消費電力は1.73倍と大幅に増えてしまう。逆に20%周波数を下げると、性能は0.87倍になるが、消費電力は0.51倍とほぼ半減する。それならば、動作周波数を落としてコアを複数にすることで消費電力と性能のバランスをとろうというのがCore 2 Duoの設計思想だ

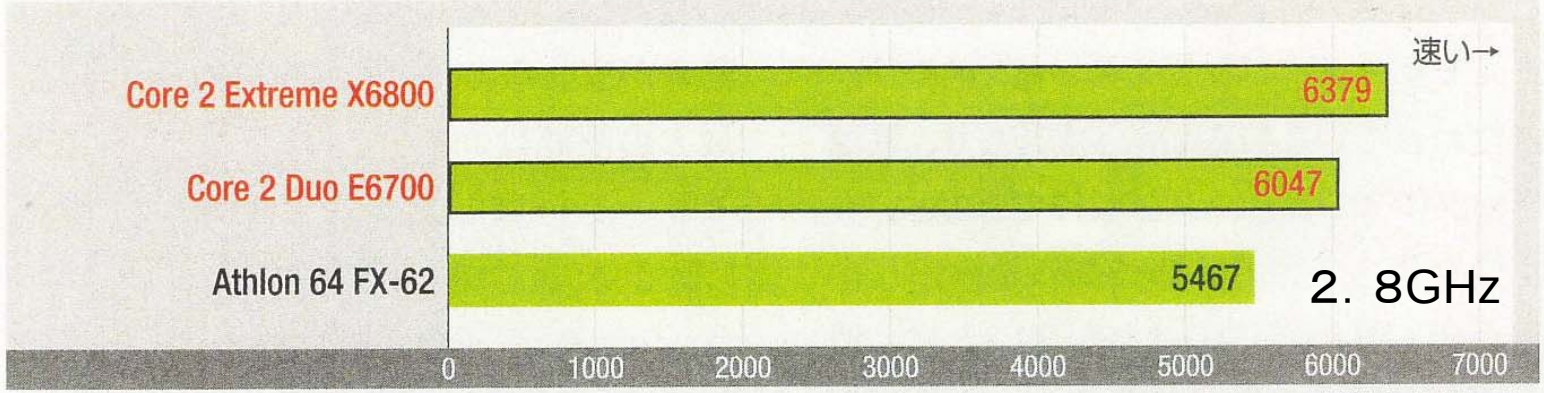
日経パソコン2006．8．14

## ●従来版デュアルコアCPUを引き離す性能



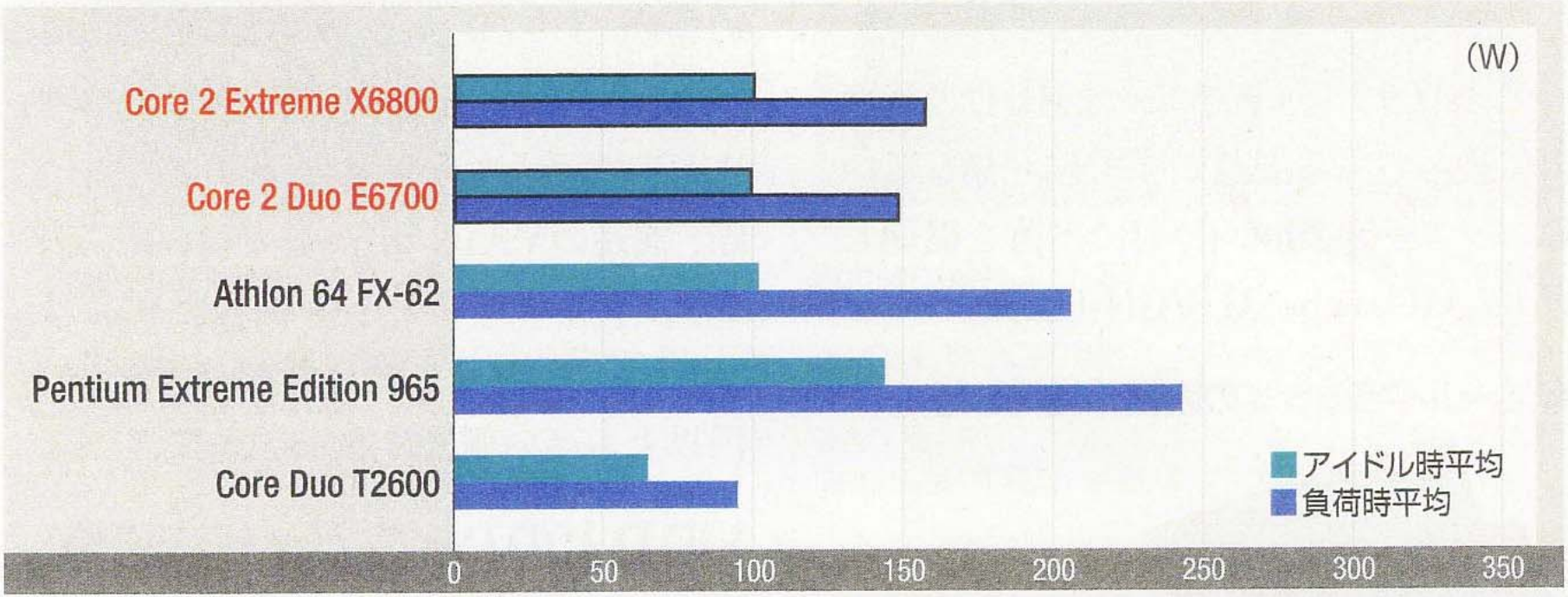

Core 2 Extreme X6800およびCore 2 Duo E6700と、従来のハイエンドデスクトップ向けデュアルコアCPU「Pentium Extreme Edition 955」(3.46GHz)の性能を比較した。Core 2系の2製品はPentium Extreme Editionの性能を大きく上回った。測定には「PCMark05」を利用。いずれも、マザーボードはインテルの「D975XBX」を使用し、2GBのメモリー(DDR2-667)を搭載して測定した

## ●AMDのハイエンドCPUと比較しても優位




米AMDのハイエンドCPU「Athlon 64 FX-62」(2.8GHz)とCore 2 Extreme、Core 2 Duoの性能を比較した。Core 2系の2製品はAthlonの性能を上回った。測定には「PCMark05」を利用。マザーボードは、Athlon 64 FX-62は「M2N32-SLI Deluxe」(アスーステック・コンピュータ製：AMD提供の評価キット)、Core 2系はインテルの「D975XBX」を使用した。いずれも、1GBのメモリー(DDR2-800)を搭載して測定した

●消費電力の測定結果




Core 2系の2製品はノートパソコン向けの「Core Duo」よりも消費電力は大きいが、Pentium Extreme EditionやAthlon 64 FX-62よりも低い。特に、負荷時の差が大きいのが分かる　　データ提供「日経WinPC」編集部

日経パソコン2006．8．14

# Sony:Cell

## マルチコア：ヘテロジニアス構成

1個のPPE:Power Processor Element

IBM　Power　Architecture

２多重スーパスカラ、乱実行なし、

分岐予測なし、非常にシンプル

8個のSPE：Synergistic Prosessing Element

SIMDでの４並列積和演算（８演算）

４GHz×８×８：256GFLOPS

相互結合：リングバス、256GB/s

応用：マルチメディア

SPE

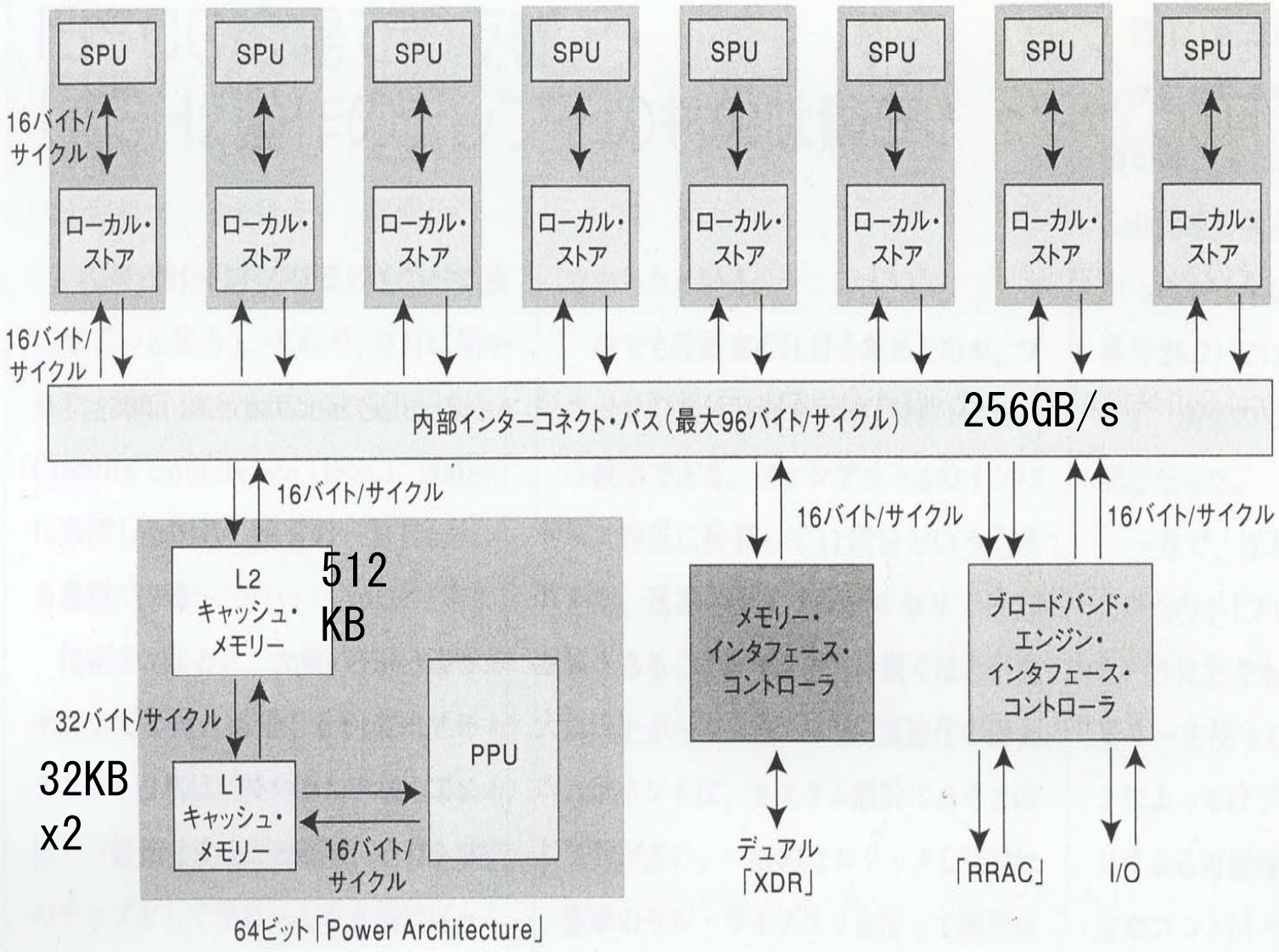



**図2 「Cell」の構成**
「SPU（Synergistic Processor Unit）」にDMAコントローラを含めた領域がSPEとなる。「RRAC」は「Redwood Rambus ASIC Cell」。ソニー・グループ，米IBM Corp.，東芝のデータ。

PPE
シンプルな２多重
スーパスカラ

日経エレクトロニクス
2005. 2. 28

# 3-3　マルチコアの相互結合網

## 評価項目

①スループット

②レイテンシ

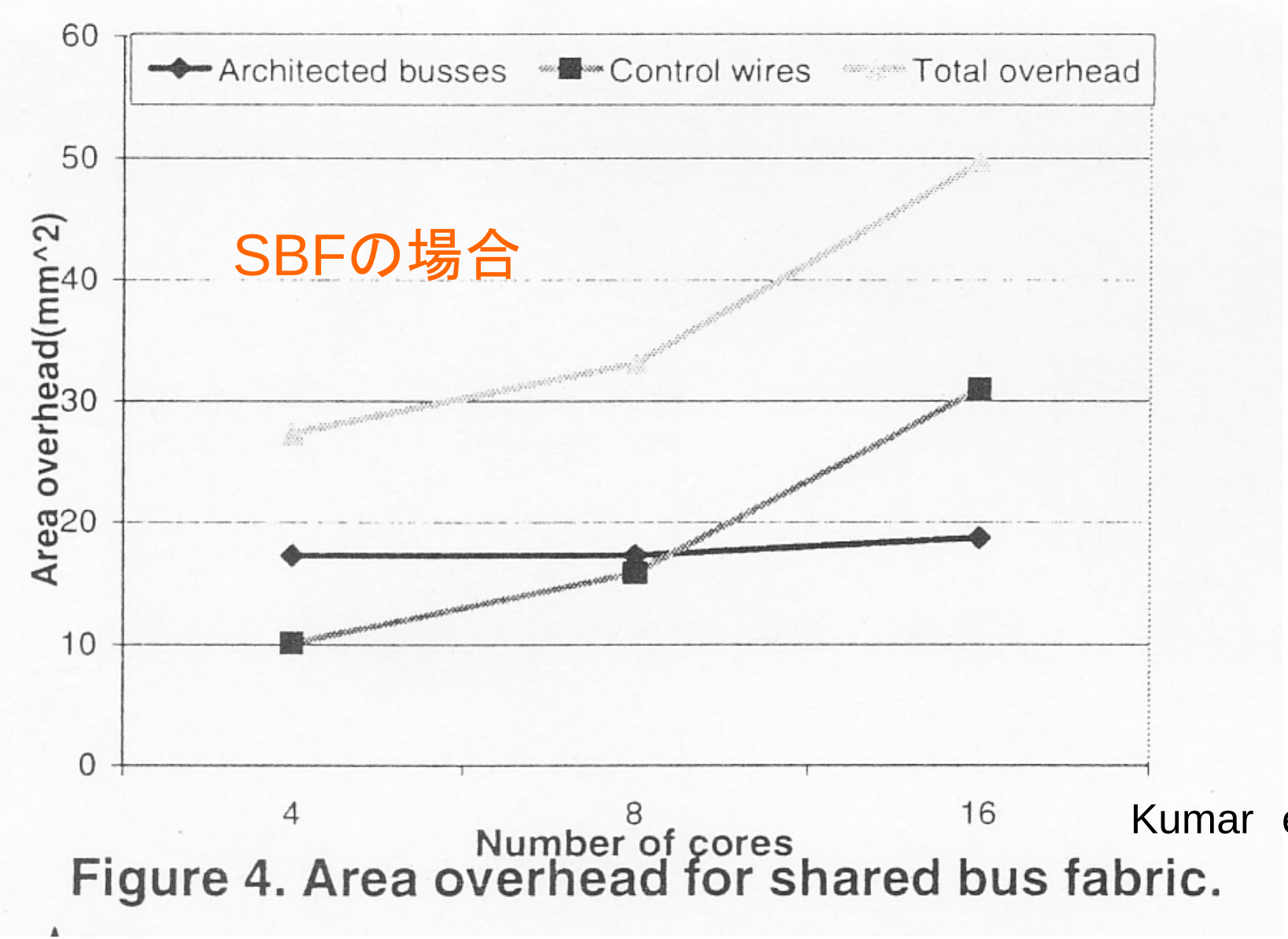
③消費電力：　スイッチ＋配線

④面積

⑤配線層数

・P.P.Pande  etal: Performance Evaluation and Design Trade-offs for Network-on-Chip Interconnect Architectures, IEEE Trans. Computers, Vol54, No.8,pp.1025-1040,2005

・R.Kumar etal: Interconnections in Multi-core Architectures: Understanding Mechanisms, Overhead, and Scaling, pp.408-419, ISCA, 2005




Kumar etal

Figure 4. Area overhead for shared bus fabric.

16コアで配線面積50mm$^2$ ⇔65nmデザイン、ダイサイズ：400mm$^2$、1コア：10 mm$^2$、L2キャッシュ：0.125MB／mm$^2$を仮定

ロジック（L2の下）： 5．6mm$^2$（4コア）、8．6mm$^2$（8コア）、17．94mm$^2$（16コア）

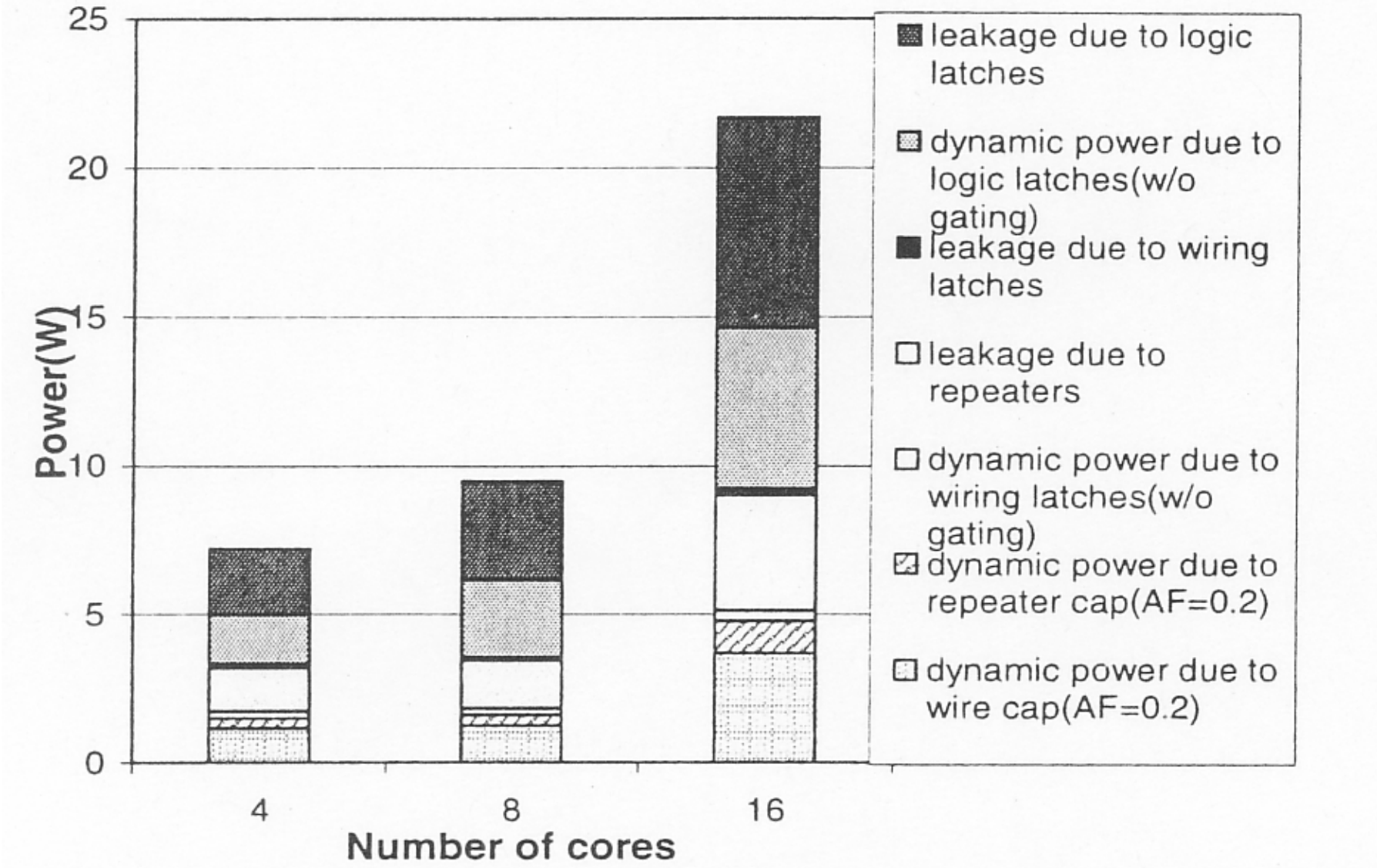



**Figure 5. Power overhead for shared bus fabric.**

1コア（Power4）：10W

オーバヘッド：2コア分に相当（22mm$^2$ 16コアの場合）

# 8コアX8L2キャッシュバンクのクロスバスイッチ

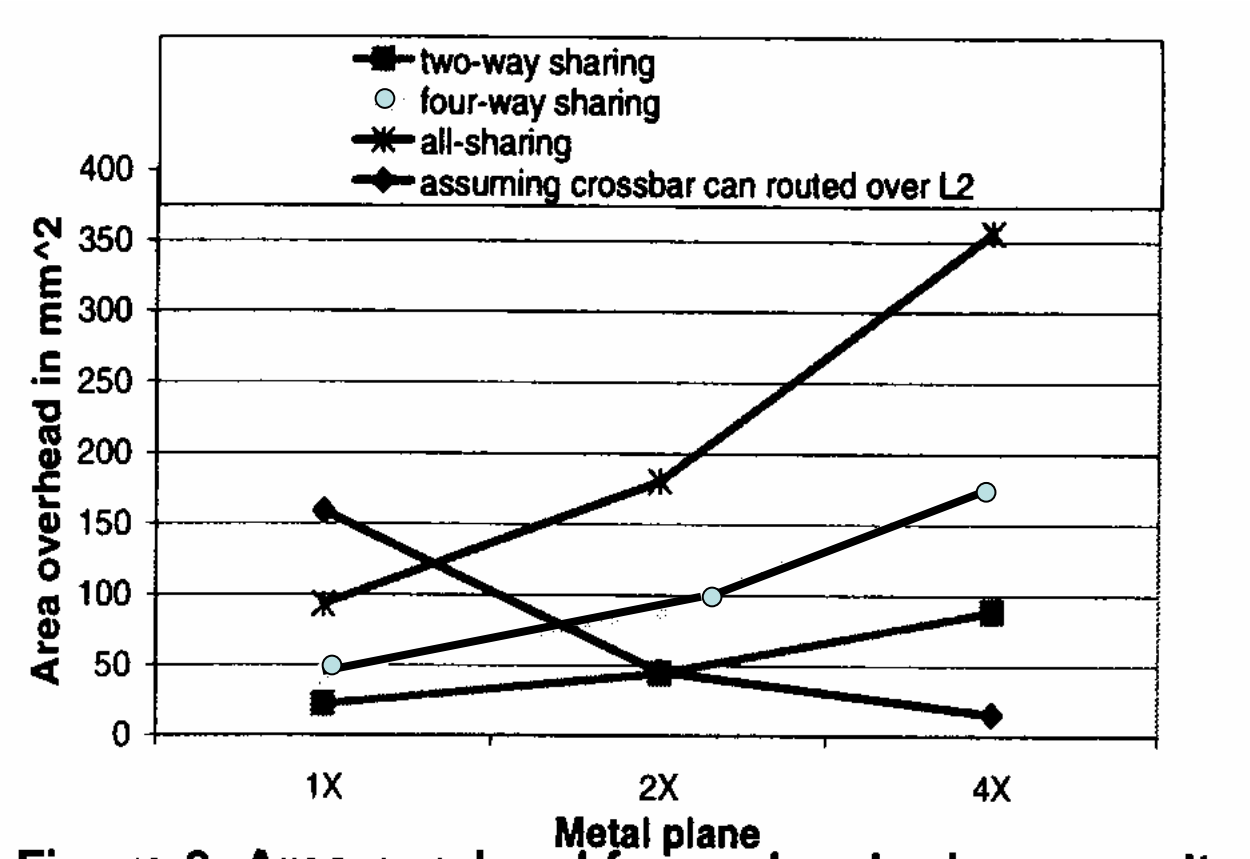



**Figure 8. Area overhead for cache sharing – results for crossbar routed over L2 assume uniform cache density.**

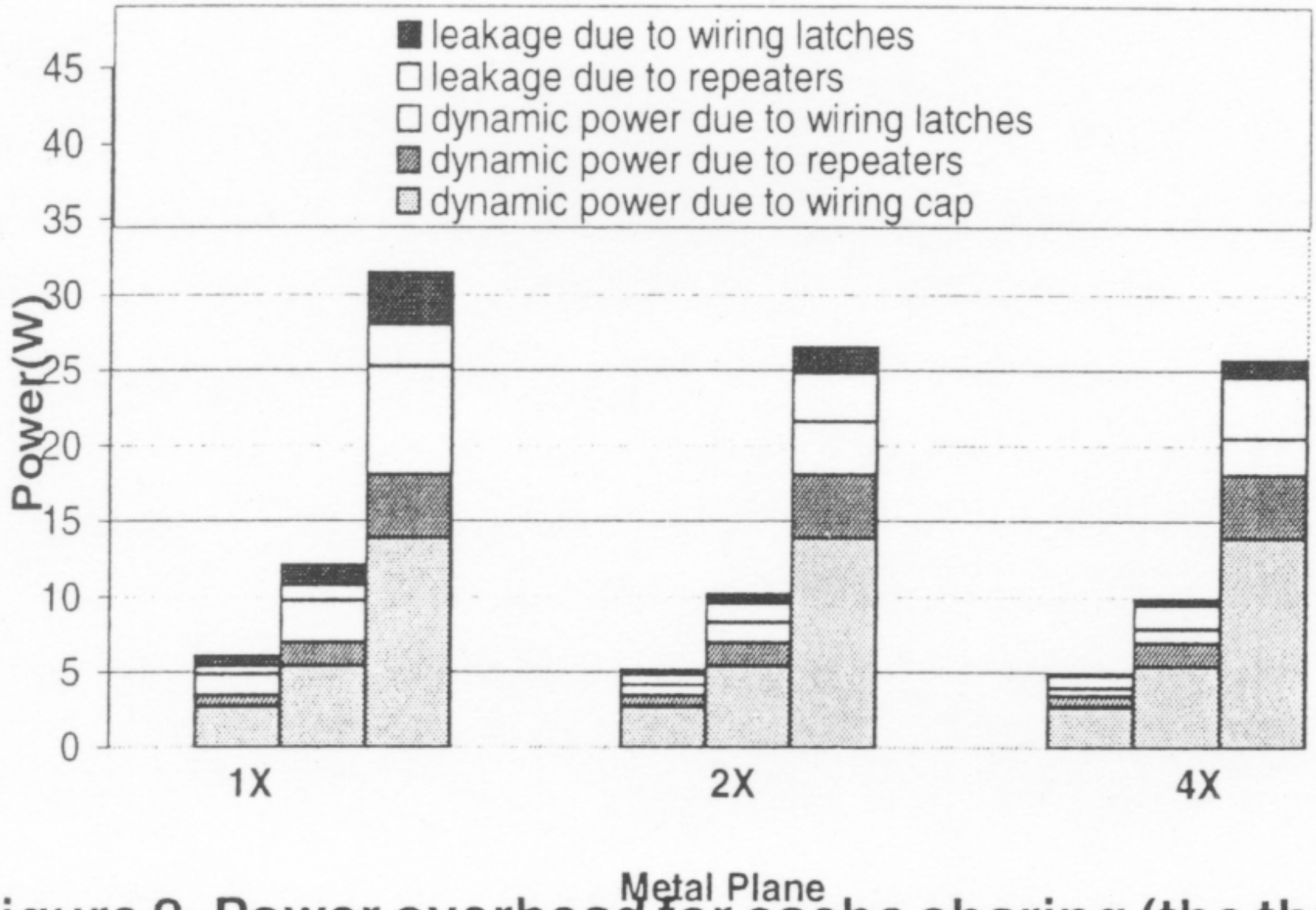
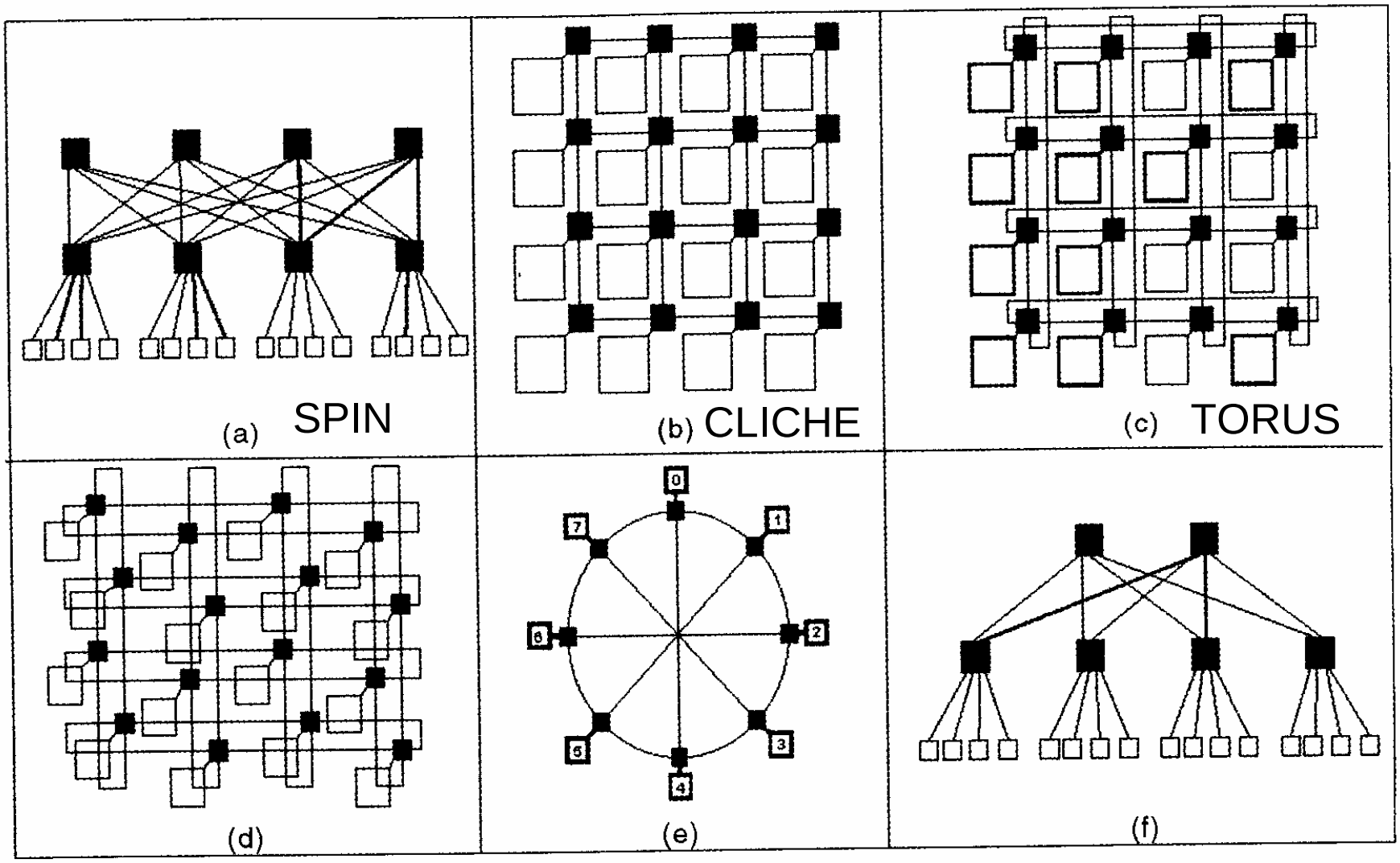



**Figure 9. Power overhead for cache sharing (the three bars, left to right, correspond to 2-way, 4-way and full sharing).**




Folded TORUS OCTAGON BFT Pande etal

SPIN:Scalable,Programmable,Integrated Network, CLICHÉ: Chip-Level Integration of Communicating Heterogeneous Elements, BFT:Butterfly Fat Tree

# 参考文献


- 富田眞治：コンピュータアーキテクチャ、第2版、丸善、2000
- 安藤秀樹：命令レベル並列処理、コロナ社、2005
- 坂井修一編：新世代マイクロプロセッサアーキテクチャ、（前編、後編）、情報処理学会誌46巻、10号、11号、2005
- 前川、木村編：マルチコアにおけるソフトウェア、情報処理学会誌47巻、1号、2006