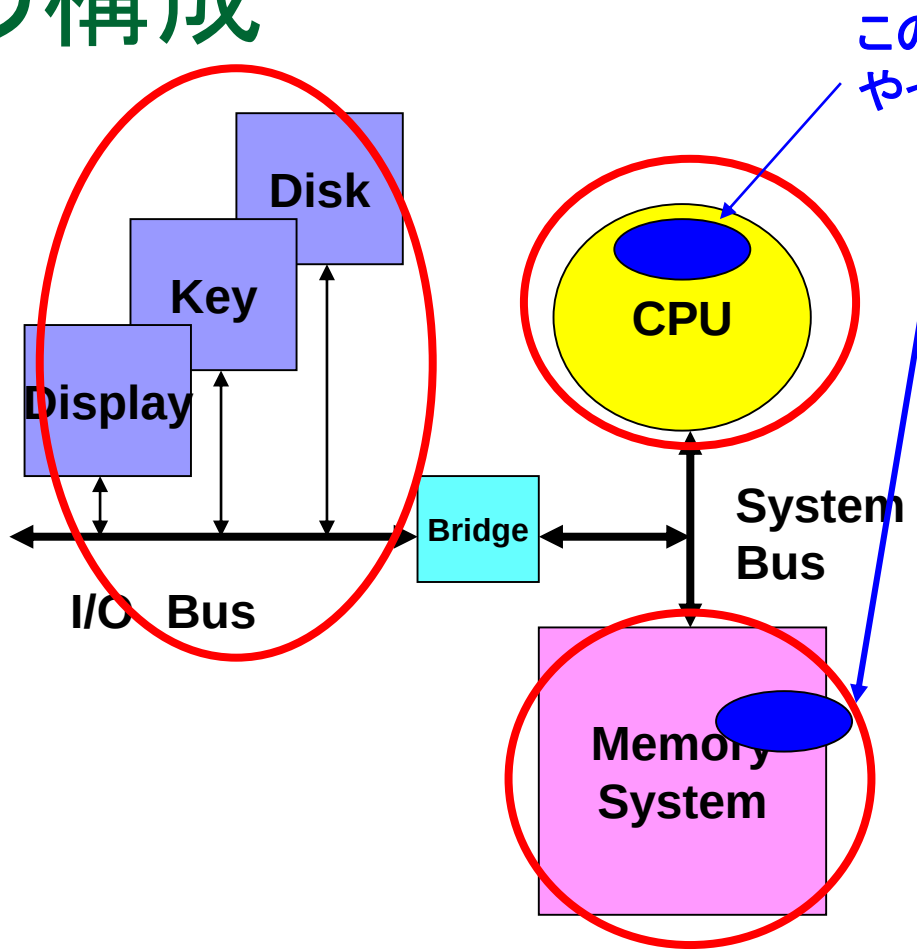
コンピュータアーキテクチャ:
ここでやったこと
これからやること

計算機構成同演習最終回

天野

コンピュータの構成



コンピュータの3要素

来年以降何をやっていくか？

- 3年春第2Q　コンピュータアーキテクチャ
  - I/Oはサボったため、最初から、割り込みも
  - POCOを32ビットにグレードアップ―>MIPSへ
  - パイプライン構成をきちんと勉強
  - CPUの高速化テクニックは設計コンテストで
- 3年秋　計算機実験
  - MIPSを用いる
  - コンパイラで機械語コードを生成
  - I/Oを実際に制御
  - FPGAを用いる

重要な入出力(I/O)

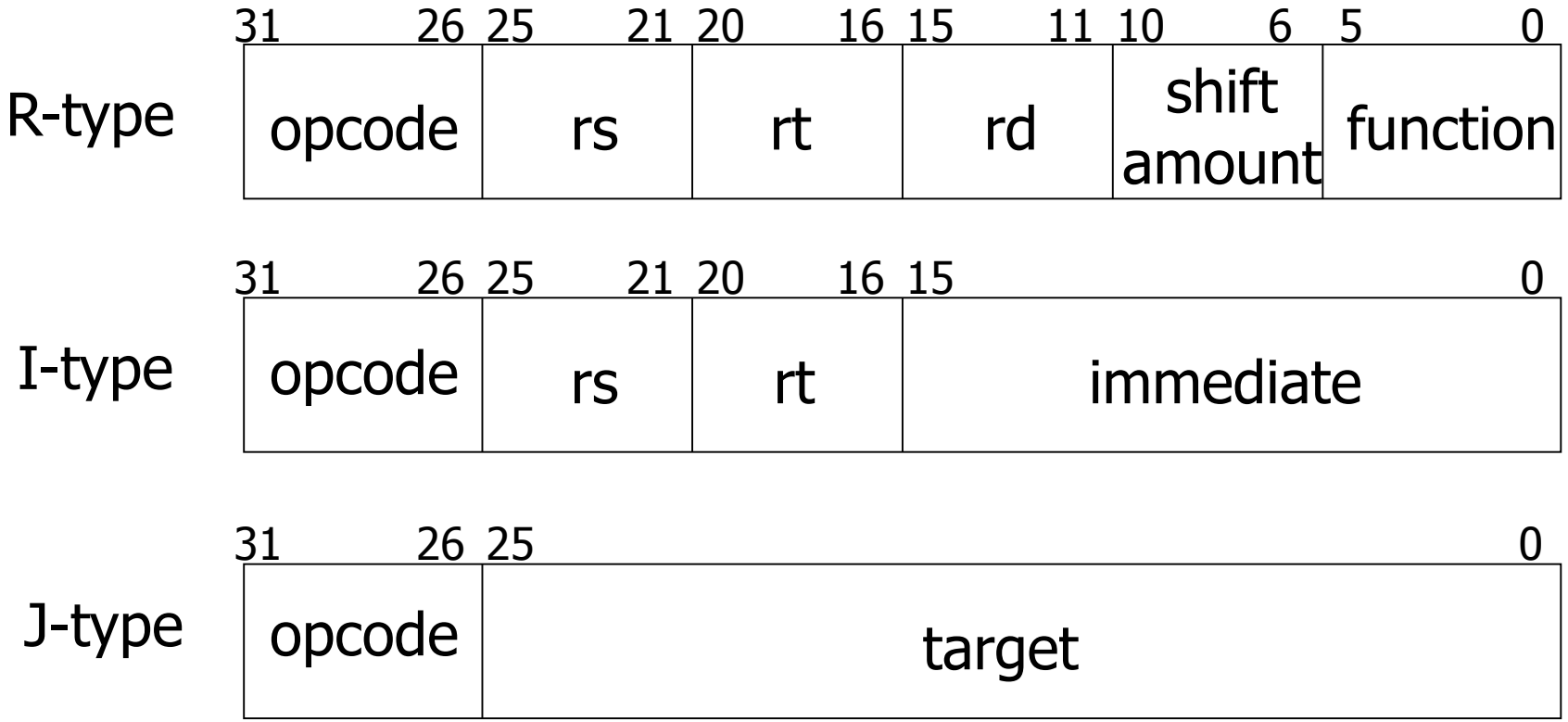
- バスブリッジを介してI/Oバスと接続
- メモリと同様な番地付けをする場合が多い（Memory-mapped　I/O)
- マルチメディア化により範囲が広がる
  - ディスク、テープ、CD、DVDなどの補助記憶
  - Ethernetなどのネットワーク
  - ビットマップディスプレイ、CDCビデオ入力
  - キーボード、マウス等の入力装置
  - 音声出力、入力

32ビットマイクロプロセッサ MIPS

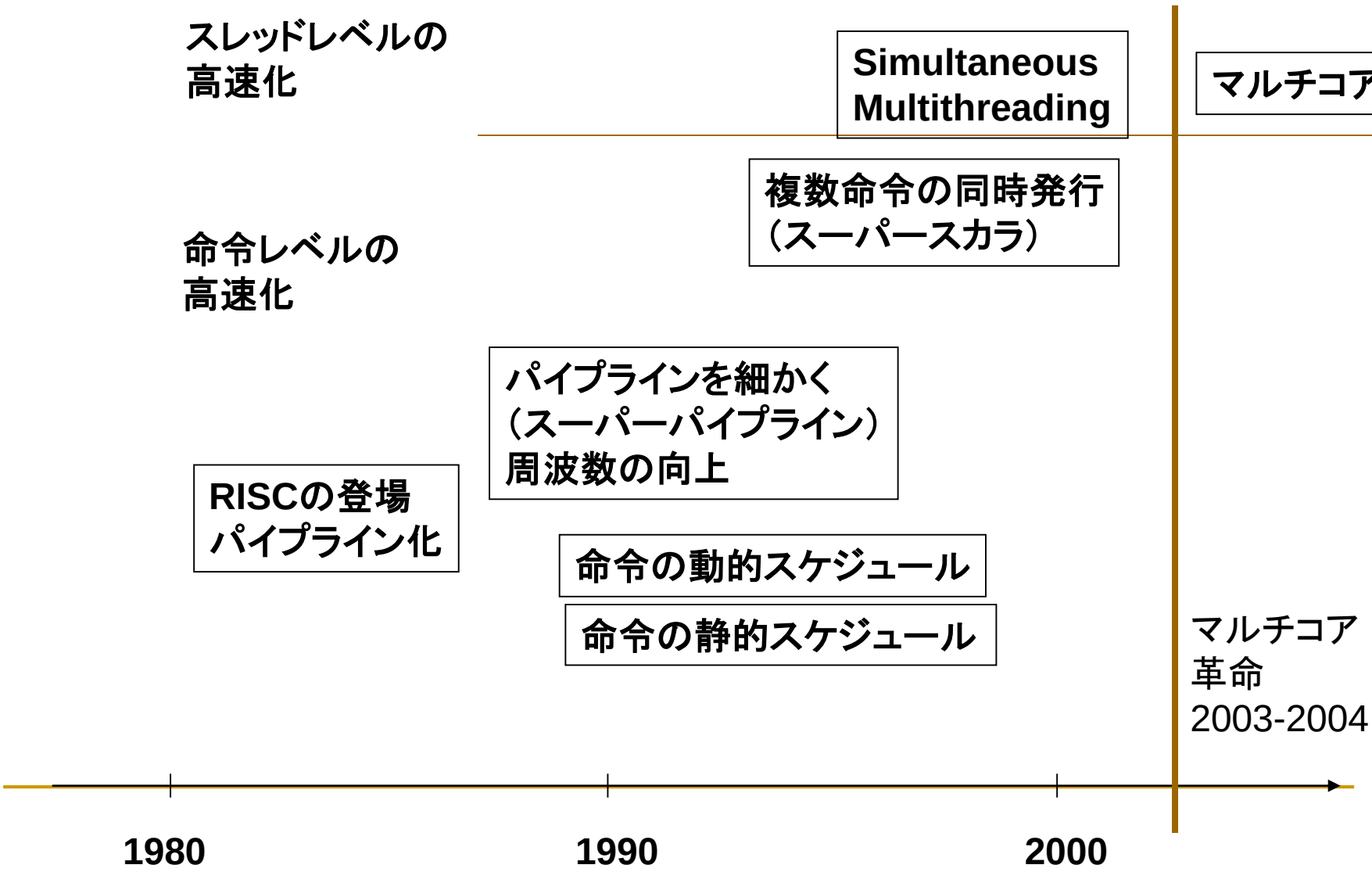
- 32ビットレジスタが32個ある
- 3オペランド方式
- LD，STはディスプレースメント付き
  - ADD　R3，R1，R2
  - ADDI　R3，R1，＃1
  - LD　R3，　10（R1）
  - ST　R1，　20（R2）
  - BEQ　R1，R2，　loop

命令フォーマット

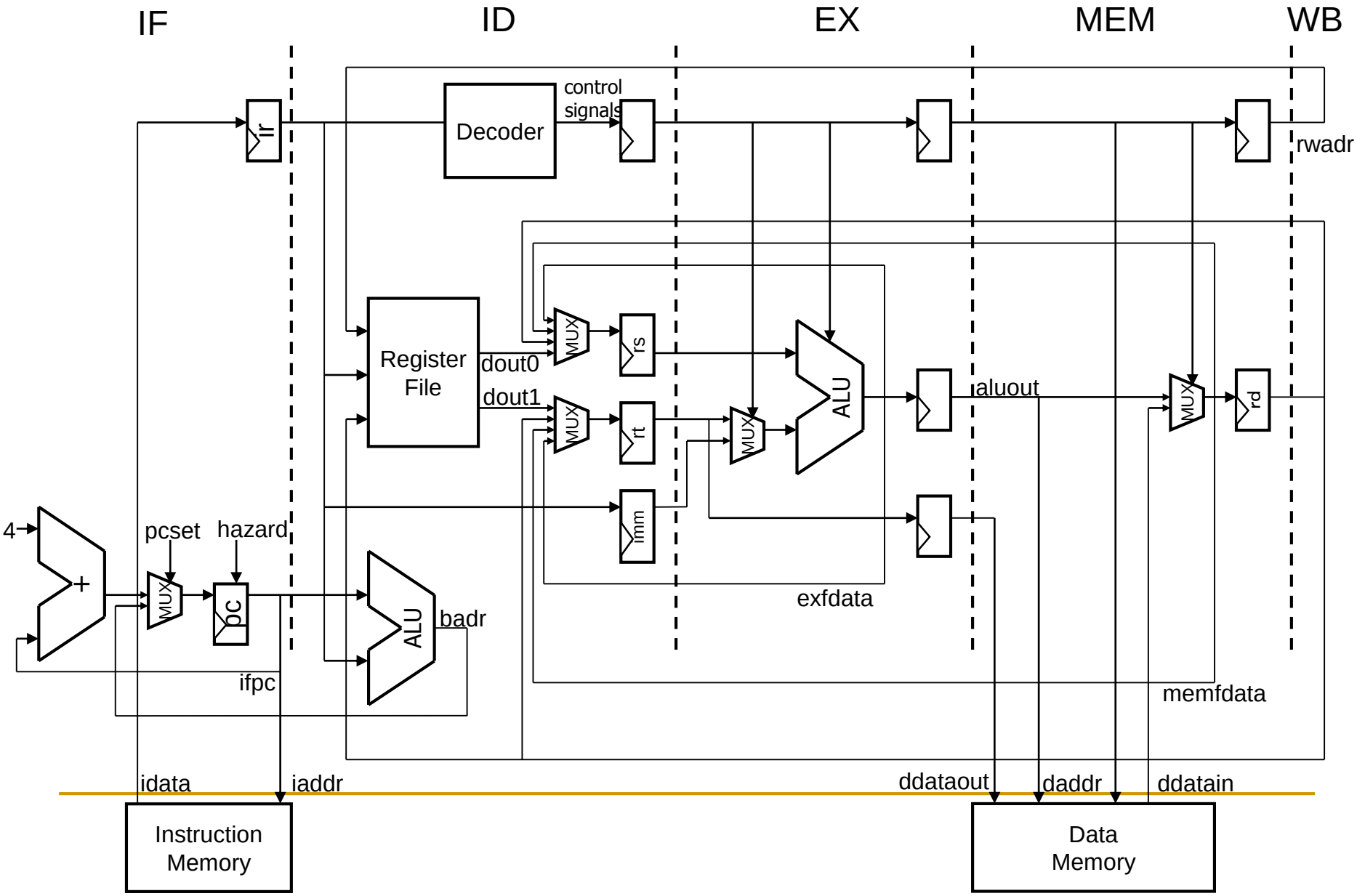
- 3種類の基本フォーマットを持つ



高速化の流れ


スレッドレベルの
高速化
Simultaneous
Multithreading
マルチコア化
複数命令の同時発行
（スーパースカラ）
命令レベルの
高速化
パイプラインを細かく
（スーパーパイプライン）
周波数の向上
RISCの登場
パイプライン化
命令の動的スケジュール
命令の静的スケジュール
マルチコア
革命
2003-2004
1980
1990
2000

パイプラインの概観


IF
ID
EX
MEM
WB
ir
Decoder
control
signals
rwadr
Register
File
dout0
dout1
MUX
rs
rt
imm
ALU
aluout
rd
exfdata
memfdata
4
+
pcset
hazard
pc
badr
ifpc
idata
iaddr
ddataout
daddr
ddatain
Instruction
Memory
Data
Memory

MIPS R4000の8ステージパイプライン

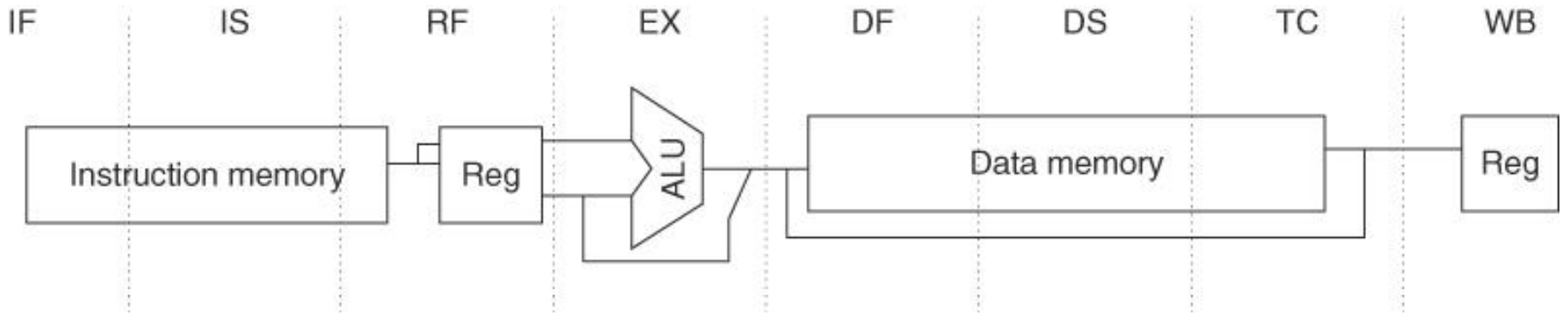


Figure C.41 The eight-stage pipeline structure of the R4000 uses pipelined instruction and data caches. The pipe stages are labeled and their detailed function is described in the text. The vertical dashed lines represent the stage boundaries as well as the location of pipeline latches. The instruction is actually available at the end of IS, but the tag check is done in RF, while the registers are fetched. Thus, we show the instruction memory as operating through RF. The TC stage is needed for data memory access, since we cannot write the data into the register until we know whether the cache access was a hit or not.

ループアンローリング

```
Loop:  L.D F0,0(R1)    ; 配列要素をF0にロード
       ADD.D F4,F0,F2   ;スカラ値を加算
       S.D   F4,0(R1)       ;結果をストア
       L.D F6,-8(R1)
       ADD.D F8,F6,F2
       S.D  F8,-8(R1)
       L.D  F10,-16(R1)
       ADD.D F12,F10,F2
       S.D F12,-16(R1)
        L.D F14,-24(R1)
        ADD.D F16,F14,F2
        S.D F16,-24(R1)
       DADDI R1,R1,#-32;  ポインタを4回分デクリメン
       BNE R1,R2,Loop   ;
       NOP
```

ループを4回分開いてやる

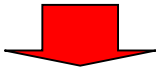
ループアンローリング

```
Loop:  L.D F0,0(R1)    ; 配列要素をF0にロード
       L.D F6,-8(R1)
       L.D  F10,-16(R1)
      L.D F14,-24(R1)
      ADD.D F4,F0,F2   ;スカラ値を加算
      ADD.D F8,F6,F2
      ADD.D F12,F10,F2
      ADD.D F16,F14,F2
      S.D   F4,0(R1)       ;結果をストア
      S.D  F8,-8(R1)
      S.D F12,-16(R1)
      DADDI R1,R1,#-32;  ポインタを4回分デクリメン
      BNE R1,R2,Loop   ;
      S.D F16,8(R1)
```

ループを4回分開いてやる

ソフトウェアパイプライン

```
Loop:  L.D F0,0(R1)    ; 配列要素をF0にロード
       ADD.D F4,F0,F2   ;スカラ値を加算
       S.D   F4,0(R1)       ;結果をストア
       DADDI R1,R1,#-8 ;ポインタをデクリメント（倍精度データを想定）
       BNE R1,R2,Loop   ;
```



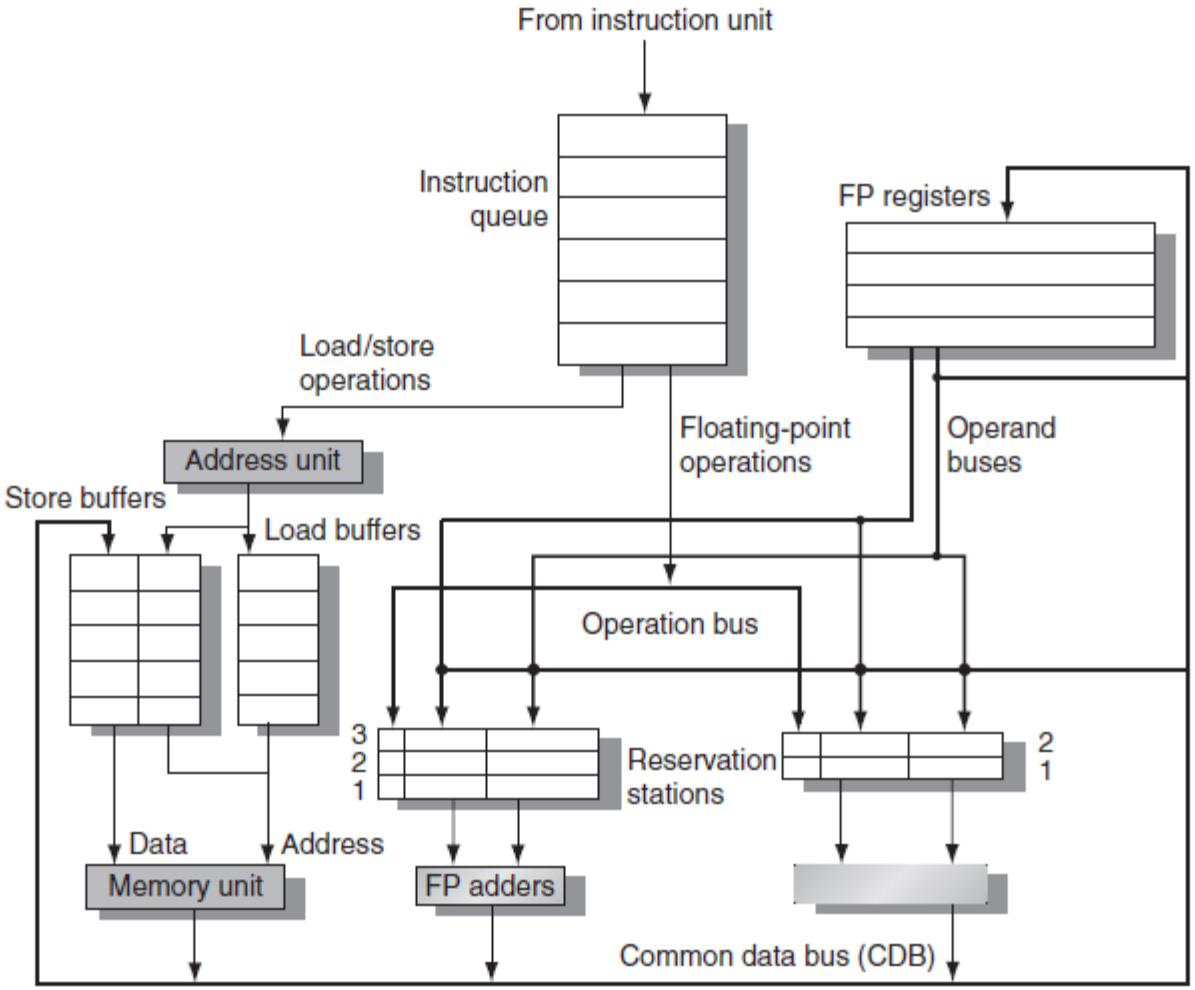
逆の順番にスケジュール

プロローグ（前処理）が必要

```
Loop:  S.D F4,0(R1)      ; M[i]のストア  2回前にLDされ1回前に加算された値
       ADD.D F4,F0,F2  ; M[i-1]に対する加算  1回前にLDされた値
       L.D F0,-16 (R1)       ; M[i-2]のロード  2つ先の値をLD
       BNE R1,R2,Loop  ; 遅延分岐
       DADDI R1,R1,#-8 ;ポインタをデクリメント（倍精度データを想定）
```

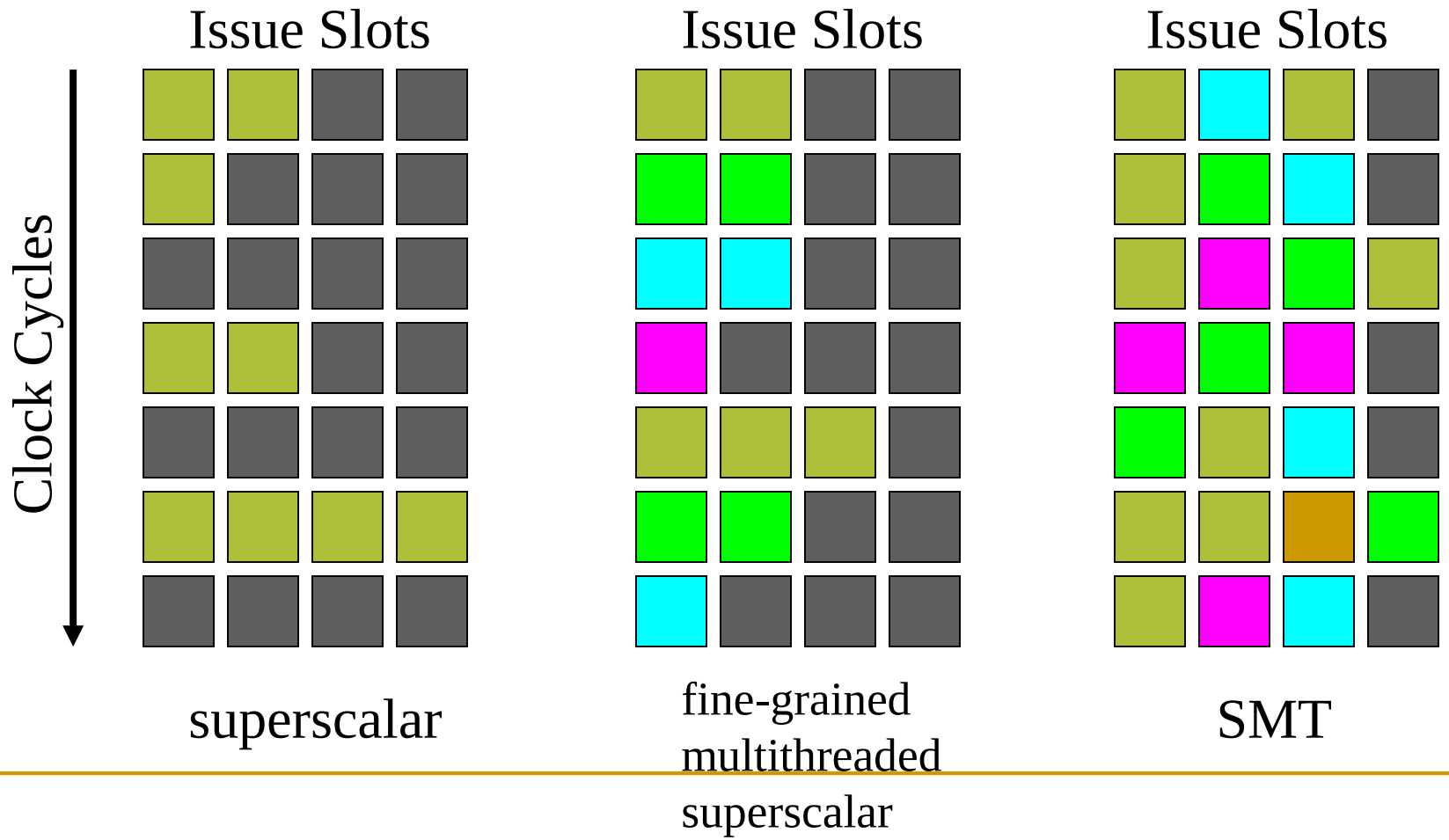
エピローグ（後処理）も必要

命令のアウトオブオーダ実行
トーマスローのアルゴリズム



Hennessy & Patterson Computer Architecture より

マルチスレッドとSMT(Simultaneous Multi-Threading)



マルチコア、メニーコア

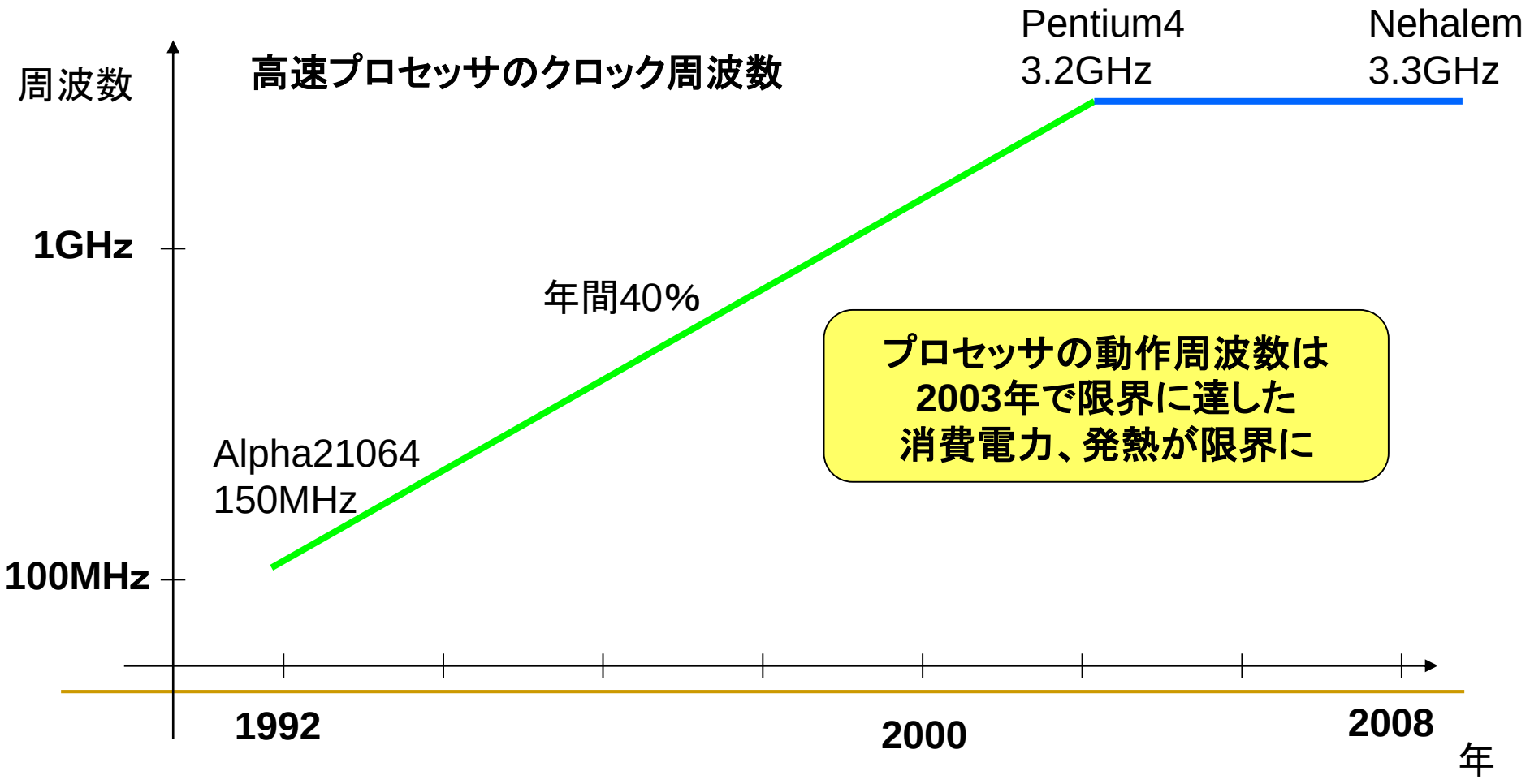
- 動作周波数の向上が限界に達する
  - 消費電力の増大、発熱の限界
  - 半導体プロセスの速度向上が配線遅延により限界に達する
- 命令レベル並列処理が限界に達する
- メモリのスピードとのギャップが埋まらない

→ マルチコア、メニーコアの急速な発達

マルチコア革命　2003-2004年

- プログラマが並列化しないと単一プログラムの性能が上がらない

クロック周波数の向上

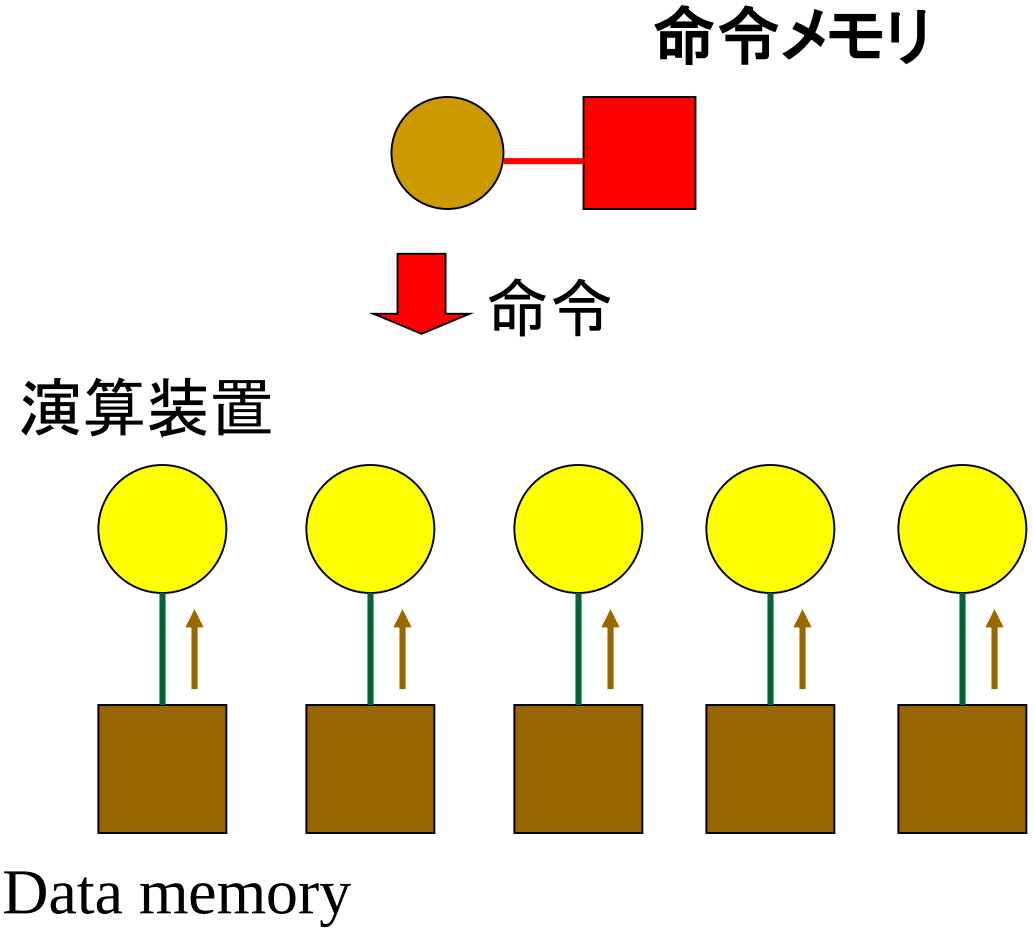

高速プロセッサのクロック周波数
周波数
1GHz
100MHz
Alpha21064
150MHz
年間40%
Pentium4
3.2GHz
Nehalem
3.3GHz
プロセッサの動作周波数は
2003年で限界に達した
消費電力、発熱が限界に
1992
2000
2008
年

Flynnの分類

- 命令流(Instruction　Stream)の数：M(Multiple)/S(Single)
- データ流（Data　Stream）の数：M/S
  - SISD
    - ユニプロセッサ（スーパスカラ、VLIWも入る）
  - MISD：存在しない（Analog　Computer）
  - SIMD
  - MIMD

一人の命令で皆同じことをする

SIMD



半導体チップ内でたくさんの演算装置を動かすには良い方法

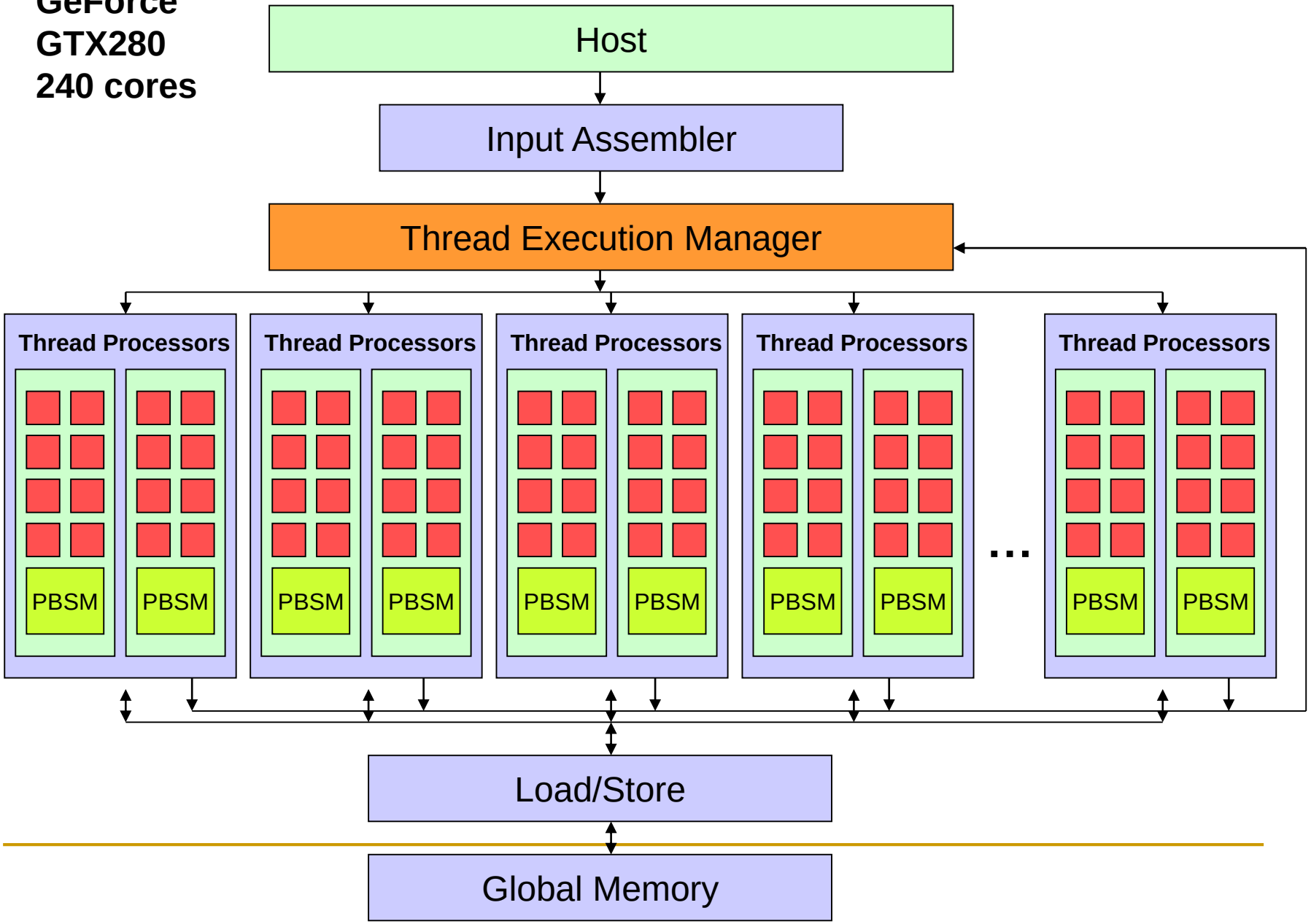
アクセラレータ（普通のCPUにくっつけて計算能力を加速する加速装置）の多くはこの方式

安くて高いピーク性能が得られる
→パソコン、ゲーム機と共用

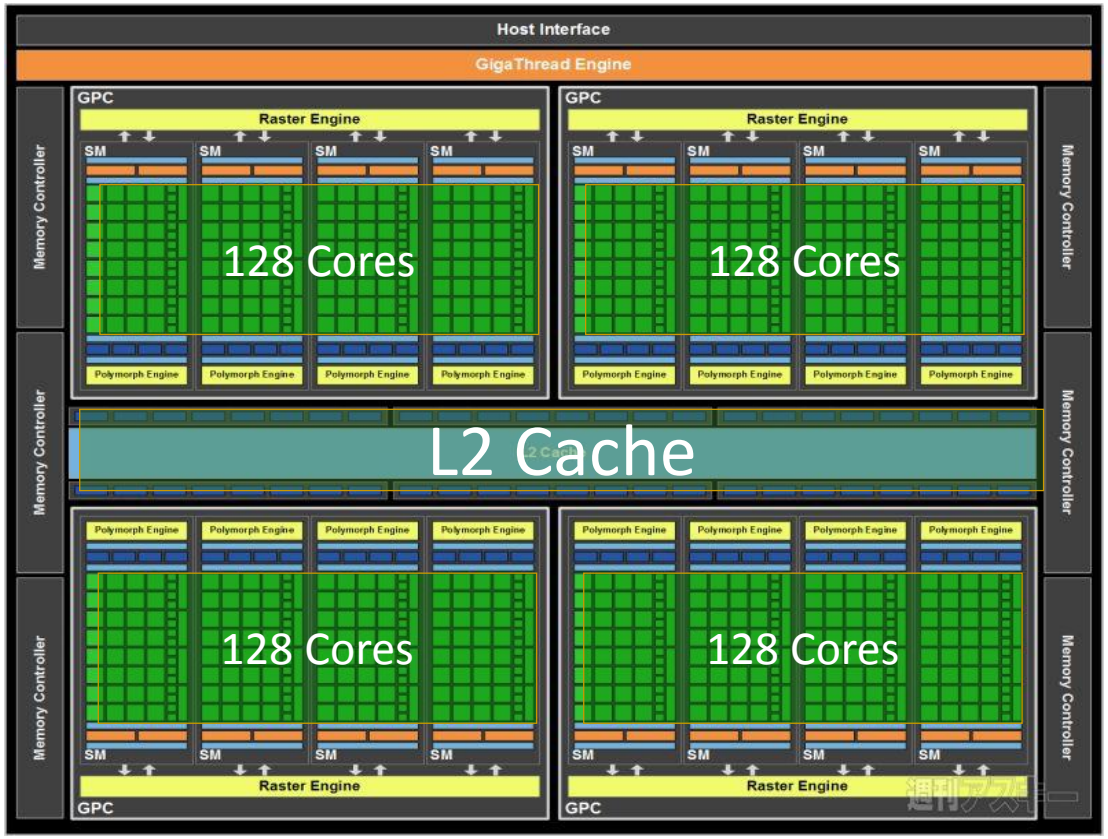
GPGPU：PC用
グラフィックプロセッサ

- TSUBAME2.0(Xeon+Tesla, Top500 2010/11 4th)
- 天河一号(Xeon+FireStream, 2009/11 5th)

GeForce GTX280 240 cores



GPU (NVIDIA's GTX580)



128個のコアは
SIMD動作をする

4つのグループは
独立動作をする

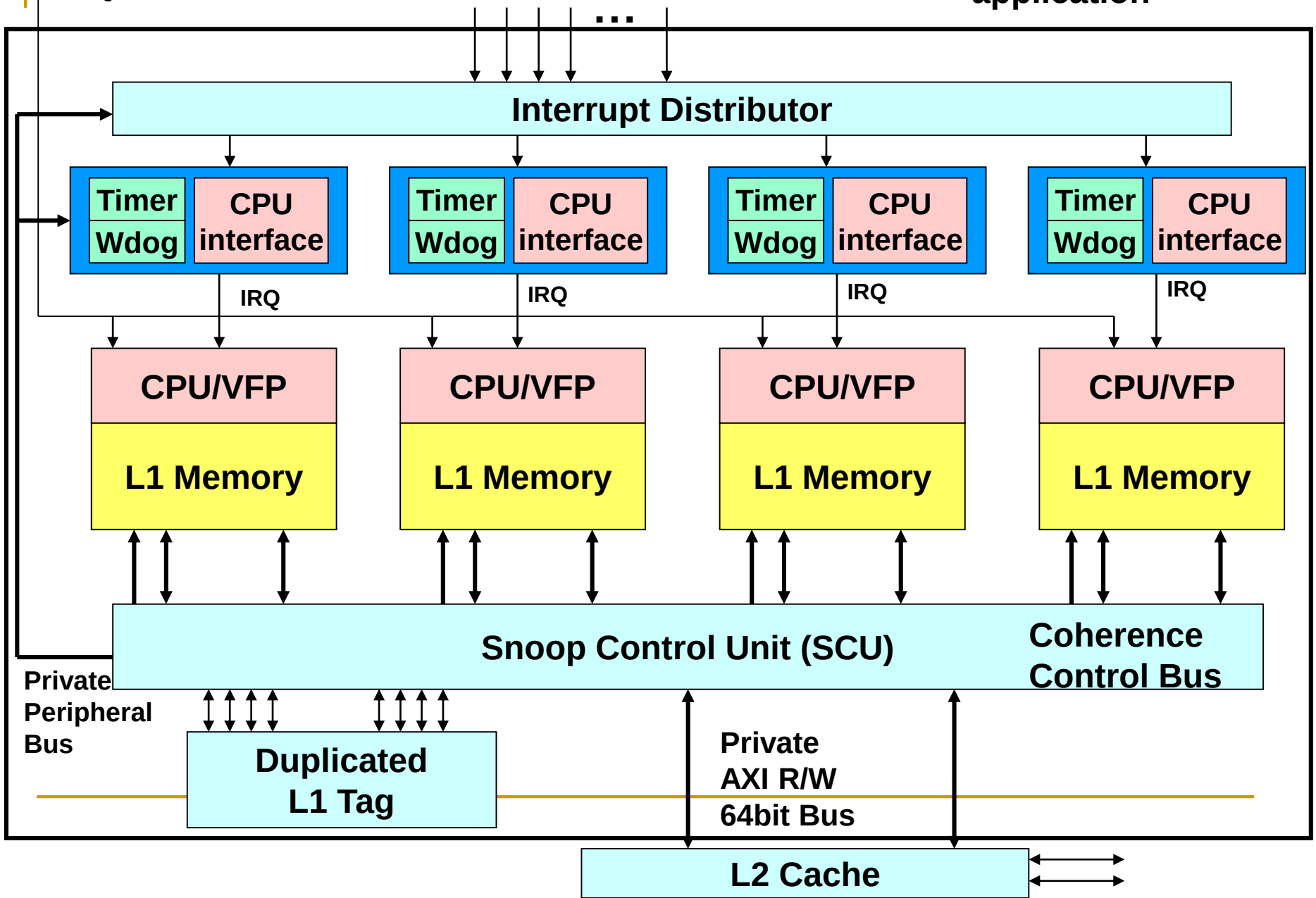
もちろん、このチップを
たくさん使う

512 GPU cores (128 X 4)
768 KB L2 cache
40nm CMOS　550 mm^2

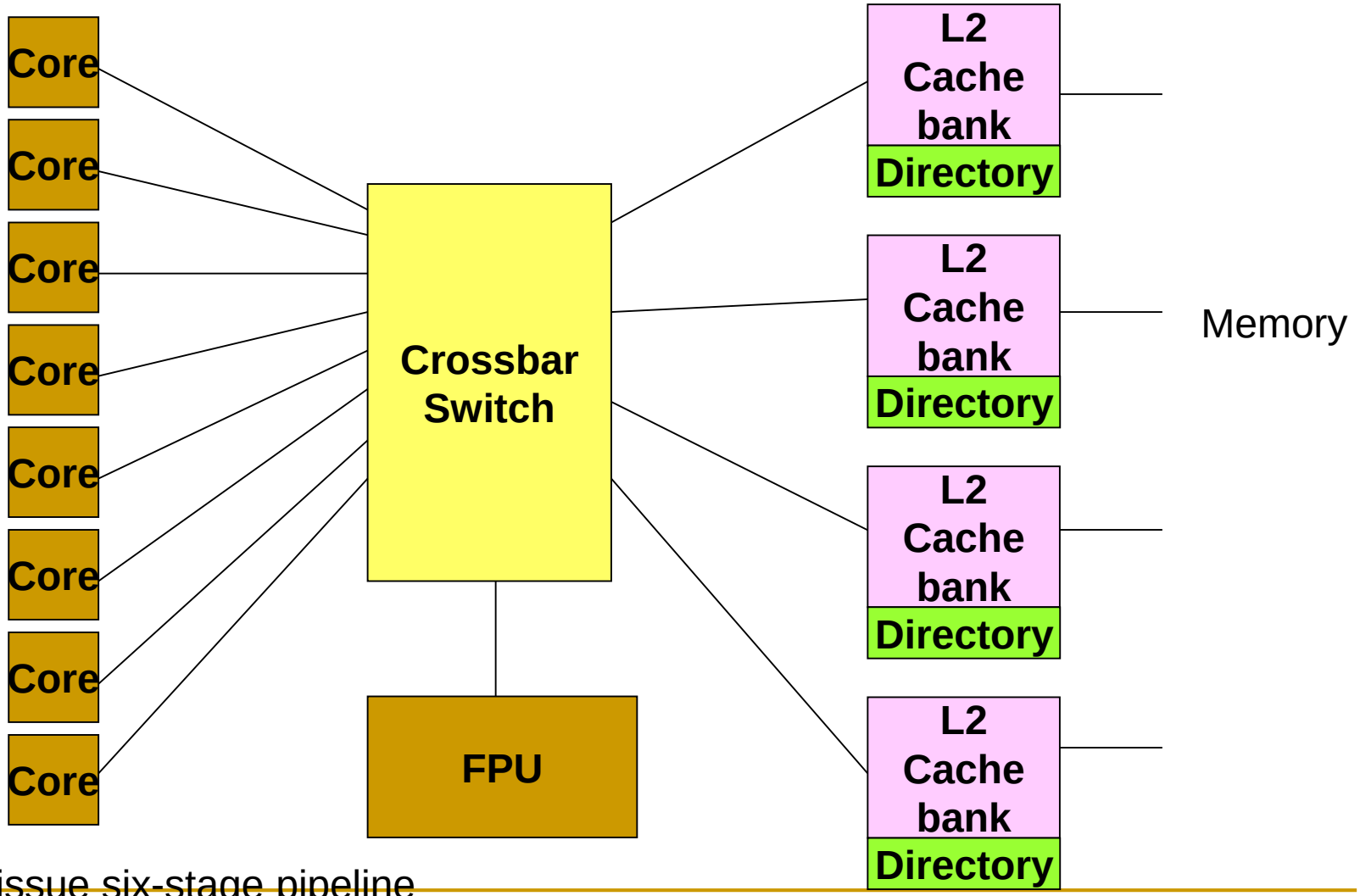
MIMD（Multipe-Instruction Streams/ Multiple-Data Streams)の特徴

- 自分のプログラムで動けるプロセッサ（コア）を多数使う
  - 同期：足並みを揃える
  - データ交信：共通に使うメモリを持つなど
- 最近のPC用のプロセッサは全部この形を取っている
- 最近はスマートフォン用のCPUもマルチコア化
- 集中メモリ型　UMA(Uniform Memory Access Model)
- 分散メモリ型 NUMA(Non-Uniform Memory Access Model)
- 共有メモリを持たない型　NORMA(No Remote Memory Access Model)

MPCore (ARM+Renesas)



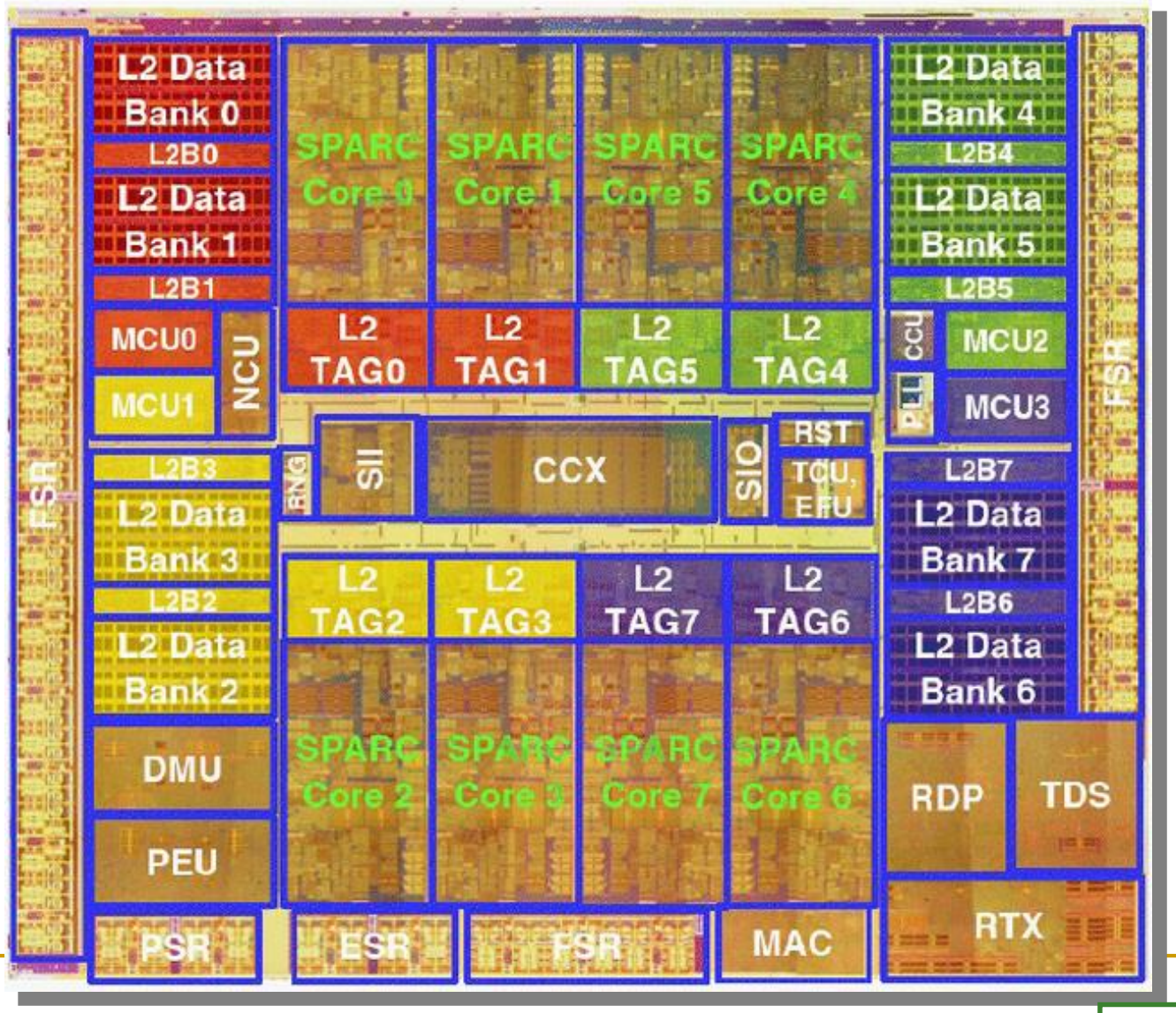
SUN T1



Single issue six-stage pipeline RISC with 16KB Instruction cache/ 8KB Data cache for L1

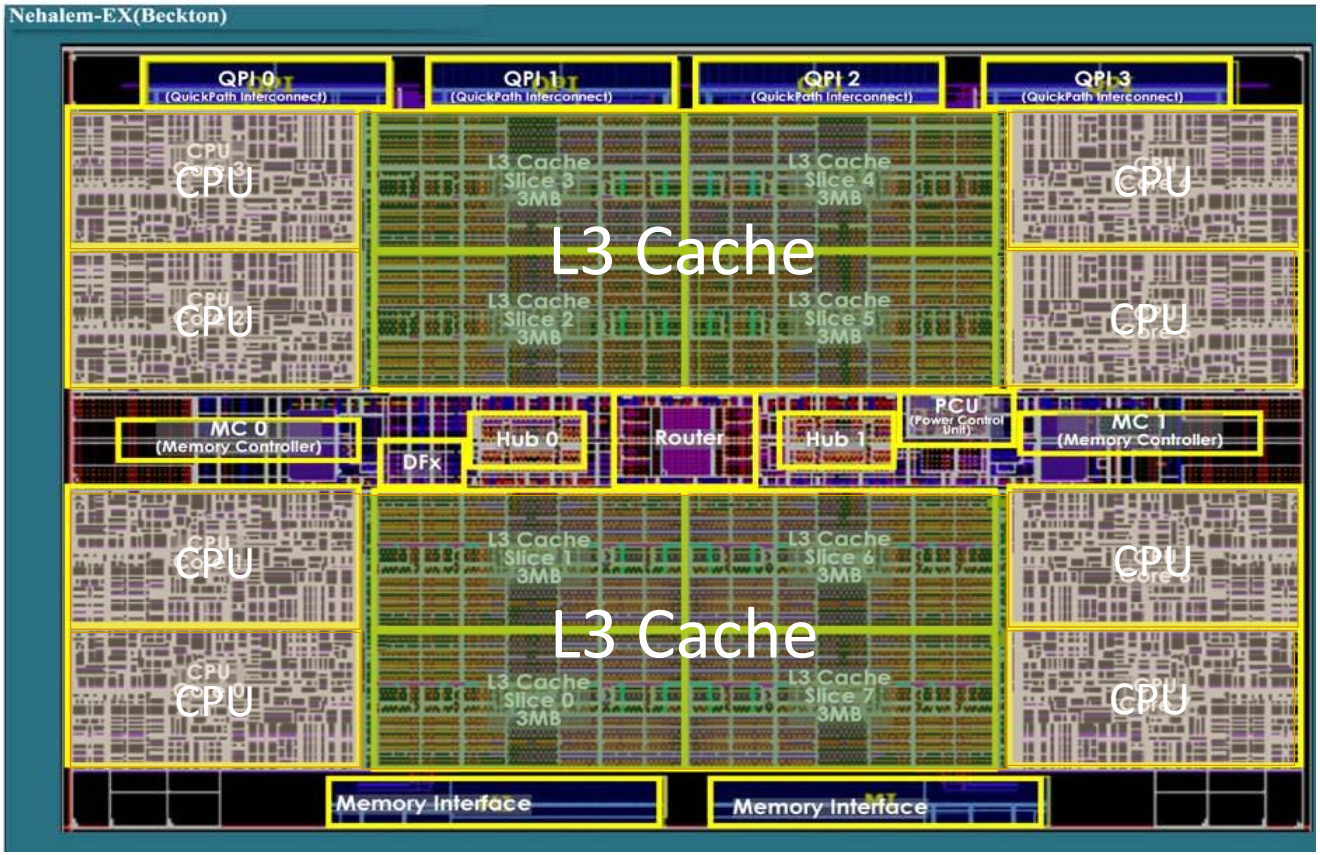
Total 3MB, 64byte Interleaved

SUN Niagara 2



Niagara 2

Multi-Core (Intel's Nehalem-EX)

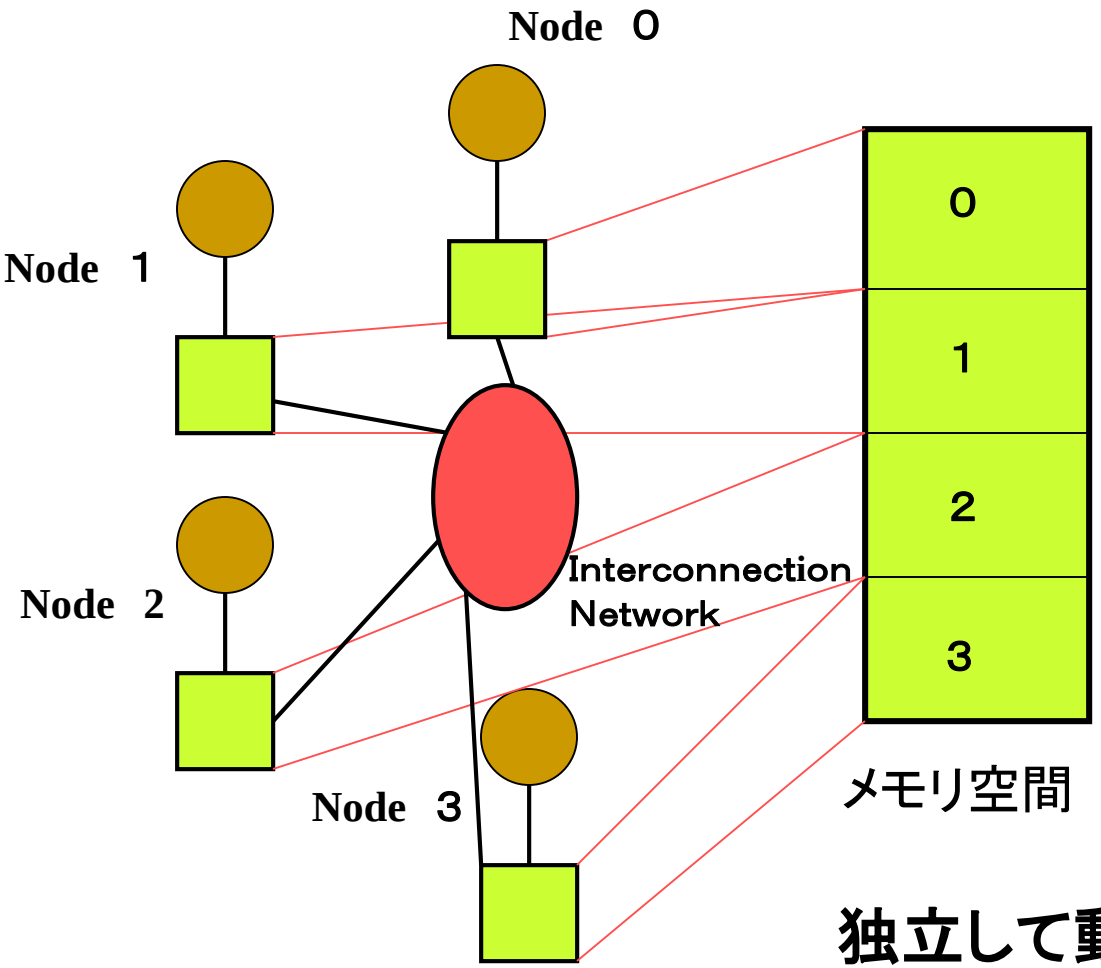


8 CPU cores

24MB L3 cache

45nm CMOS 600 mm^2

分散共有メモリ型

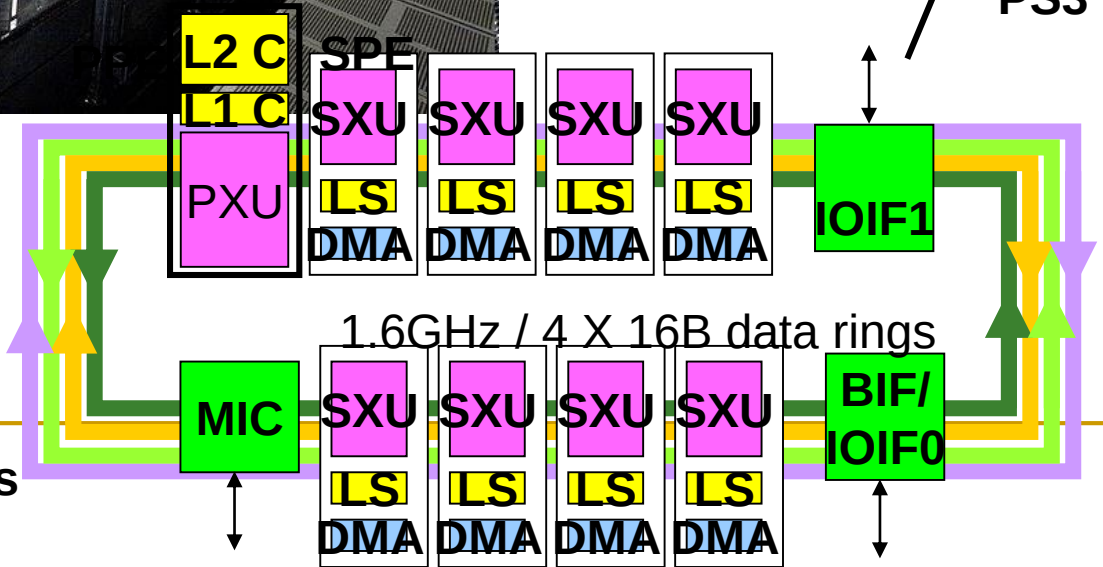


独立して動けるプロセッサを複数使う

Cell Broadband Engine

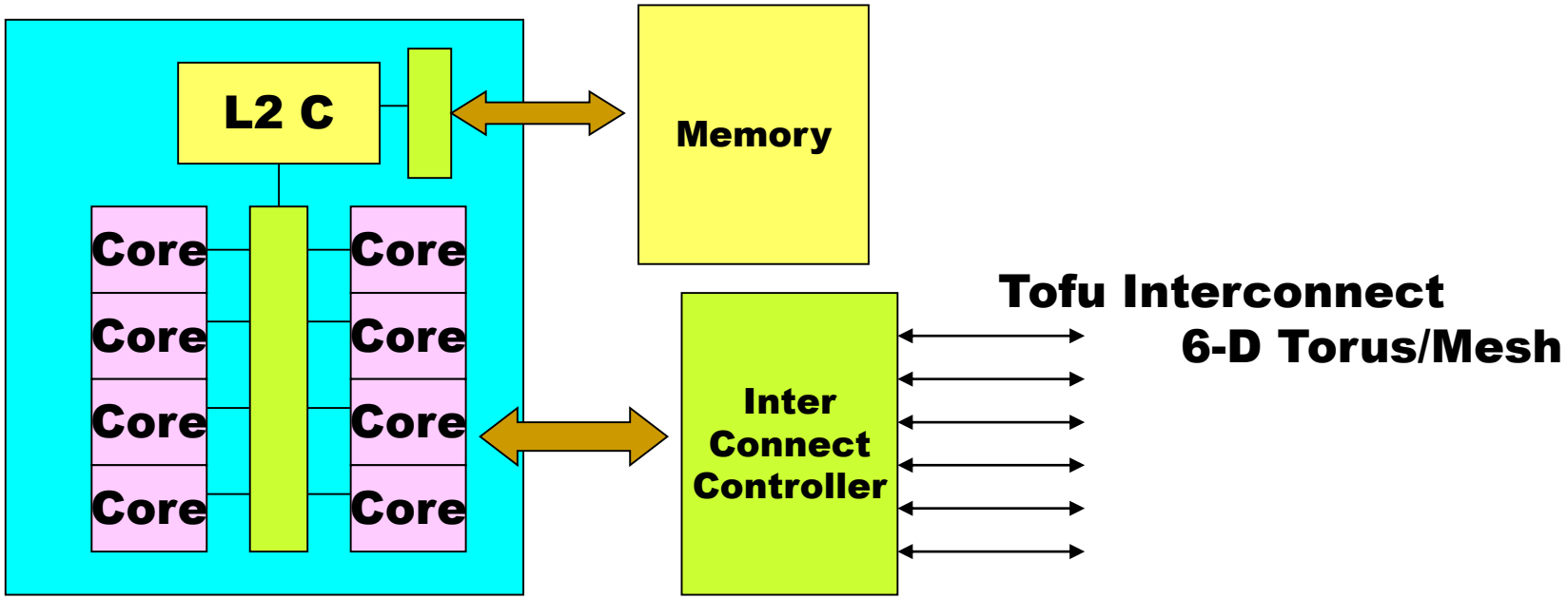
PS3

IBM Roadrunner



Common platform for supercomputers and games

Supercomputer「K」



SPARC64 VIIIfx Chip

4 nodes/board

24boards/Lack

96nodes/Lack

RDMA mechanism
NUMA or UMA+NORMA

「京」のシステム構成

システム全体

計算ラック×864

1京回/秒≒10.6ペタフロップス
1PB以上

計算ラック群

計算ラック×8
ディスクラック×2

98.4兆回/秒
12TB

計算ラック

システムボード×24
IOシステムボード×6

12.3兆回/秒
1.5TB

システムボード

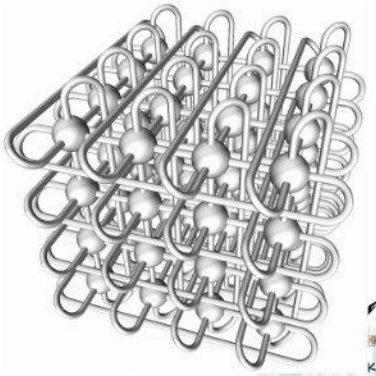
ノード×4

5120億回/秒
64GB

ノード

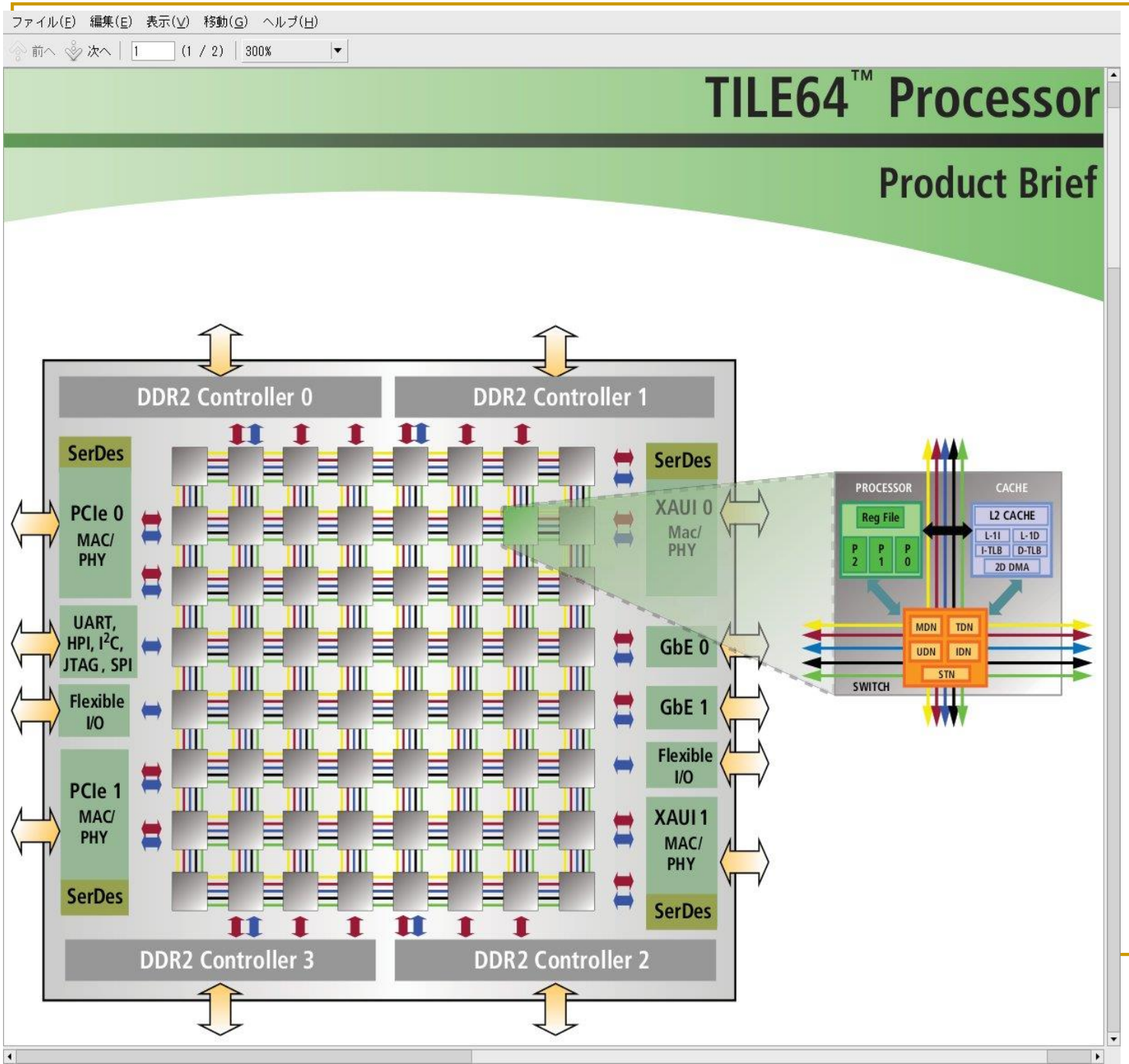
CPU×1
ICC×1
メモリ

演算性能:1280億回/秒
メモリ容量:16GB



SACSIS2012 Invited speech


クラスタコンピュータ
共有メモリを持たないものも多い

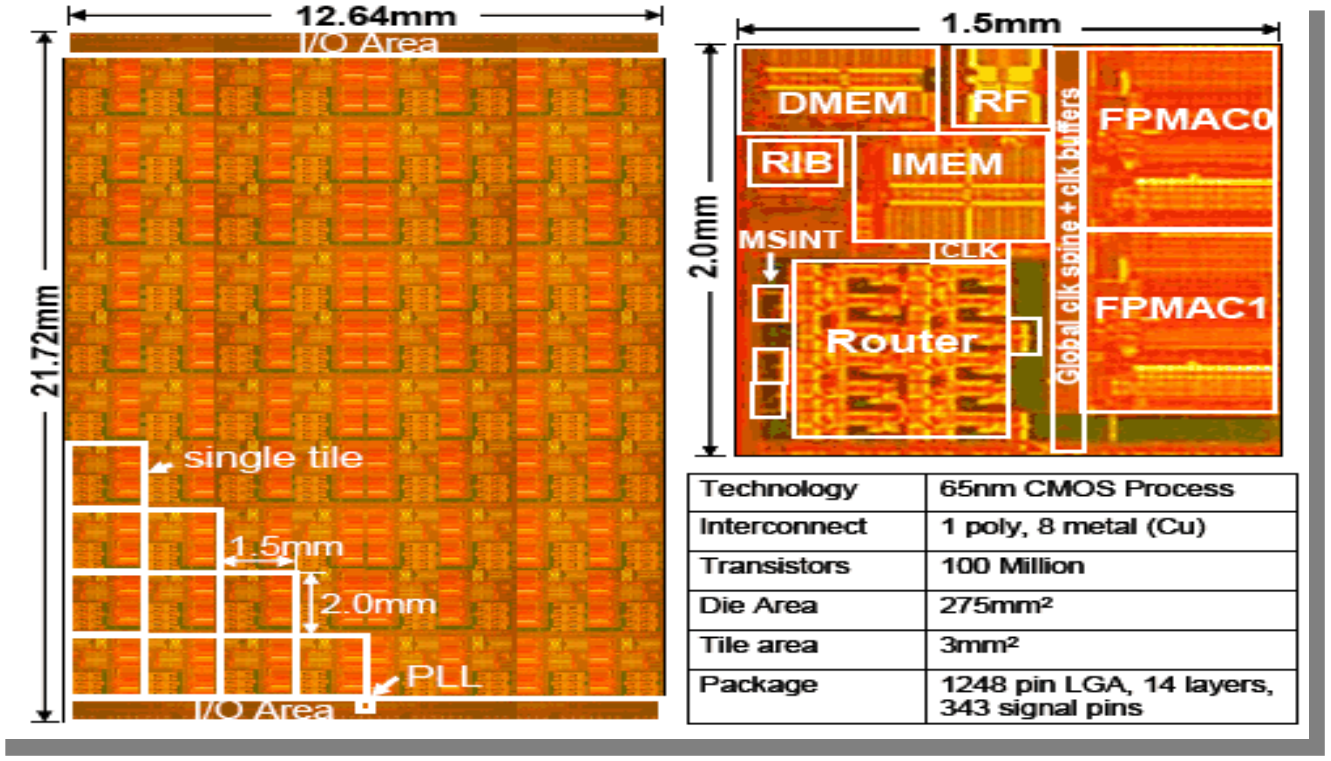


Tilera's Tile64

Tile Pro, Tile Gx

Linux runs in each core.

Intel 80-Core Chip



Technology	65nm CMOS Process
Interconnect	1 poly, 8 metal (Cu)
Transistors	100 Million
Die Area	275mm²
Tile area	3mm²
Package	1248 pin LGA, 14 layers, 343 signal pins

Intel 80-core chip [Vangal,ISSCC'07]

Amdahlの法則

高速化の効果はそれが可能な部分の割合によって制限される

高速化後の時間＝　高速化前の時間×（（1－高速化が効く割り合い）＋
高速化が効く割り合い／スピードアップ）

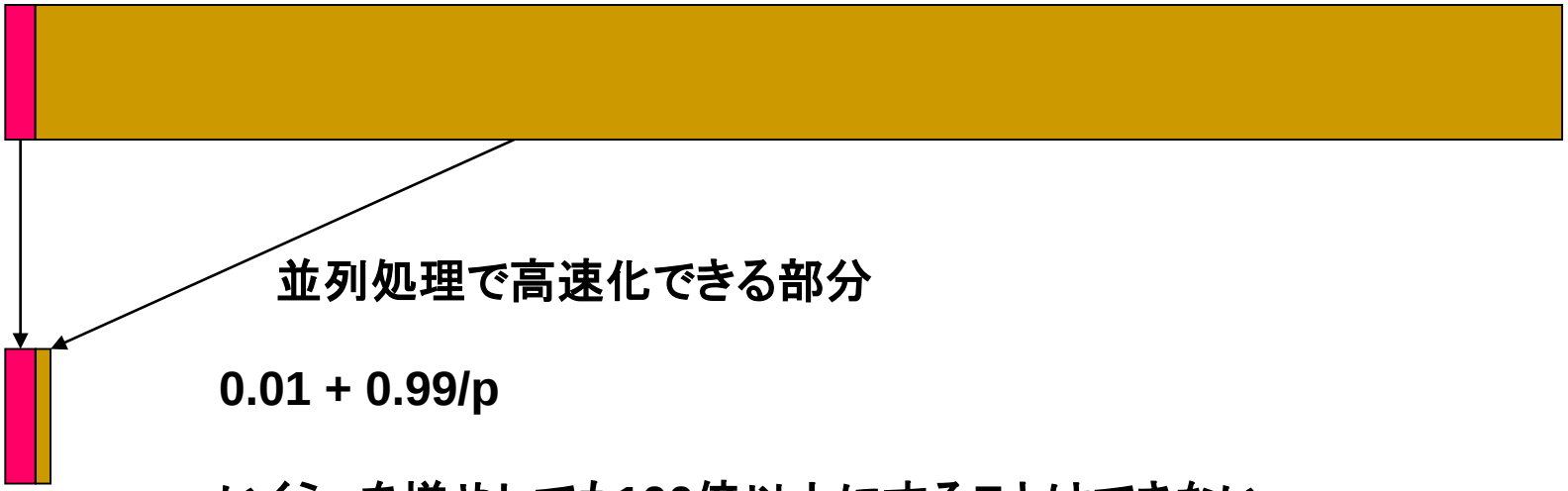
高速化後の時間／高速化前の時間＝

（1－高速化が効く割り合い）＋高速化が効く割合／スピードアップ

Amdahlの法則

シリアルな部分part
1%

並列処理が可能な部分 99%



並列処理で高速化できる部分

0.01 + 0.99/p

いくらpを増やしても100倍以上にすることはできない

高速化の効果はそれが可能な部分の割合によって制限される

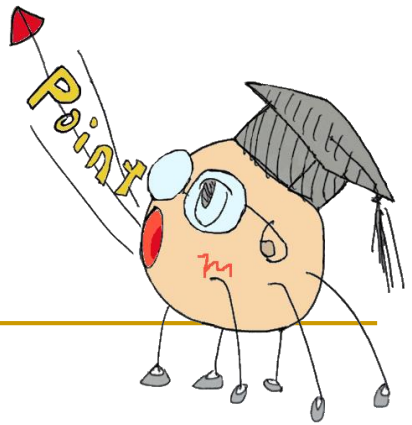
多くの並列処理にとっては限界になる

まとめ

- 汎用プロセッサのマルチコア化は現在絶好調進行中
  - 世代が進む毎に2ずつ増えている
  - しかし、コア数をこれ以上増やしても良いことがないかも、、、
- メニーコア
  - GPUなどのアクセラレータの性能向上は進む
  - メニーコアによるクラウドコンピューティング

おわりに

- コンピュータ自体の設計は、それを含むシステムの設計の一部として行わなければならない
- スマートフォン、タブレット、ウェアラブルデバイス、自動車、ロボット、センサ、コンシューマ機器、サーバ、データセンターなど対象システムは様々な性能、コスト、電力が要求される
  - 様々なコンピュータが成功する可能性がある
- 大変面白い時代になってきている



最後の演習

- 演習１：並列化できない部分が1%存在し、その他は全て並列処理が可能なとき
  - 100台で並列処理を行うと性能は何倍になるか？
  - 10000台ならばどうなるか？
- 演習２：ではなぜ、プロセッサ数10000を越えるスーパーコンピュータを作るのだろうか？その理由を考えよ。

- 3年履修した全科目のアンケート調査に入力せよ
  - 凝ったコメントを入れる必要はない