システムサイドから見たこれまでの経験と将来像

数値風洞 NWT の開発　－1担当者の体験－

北村　俊明

（元富士通(株)　現広島市立大学）

講演者紹介

名前：北村　俊明（きたむら　としあき）

現職：広島市立大学　大学院情報科学研究科　情報工学専攻　教授

略歴：1955年　京都市生まれ

1978年　京都大学工学部情報工学科　卒業

1983年　京都大学大学院工学研究科博士後期課程情報工学専攻　研究指導認定退学

富士通株式会社入社　超大型汎用計算機の開発に従事

1988年　スーパーコンピュータ NWT の開発に従事

1994年　命令セットエミュレーションの研究に従事

1996年　京都大学博士（工学）

1997年　米国 HAL Computer Systems Inc.に出向 SPARC プロセッサの開発に従事

2000年　富士通株式会社を退社し、京都大学総合情報メディアセンター助教授

2002年　広島市立大学情報科学部教授

現在に至る。

情報処理学会、電子情報通信学会、ACM、IEEE 会員。大学院在学中から現在まで、
各種コンピュータの命令セット、マイクロアーキテクチャの研究開発に従事。
NWT においては、スカラ命令セットの開発を行い、システム論理仕様とスカラ部の
論理設計を担当した。

数値風洞NWTの開発 –1担当者の体験–

広島市立大学大学院　情報科学研究科　北村俊明

2011 / 9 / 10

講演内容

- 開発に参加したエンジニアとして、
  - 何が挑戦的課題だったか
  - 何を達成したいと思ったか
  - どんな状況で開発したか
  - そこから得られた教訓

  等をお話ししたい

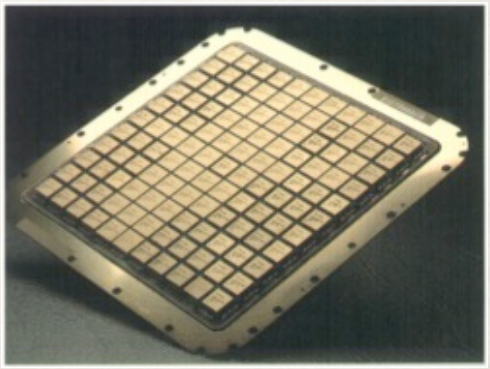
最大の課題は小さくすること

- システムイメージは現行VP-2000シリーズのベクトルプロセッサを数百台結合したシステム

VP-2000シリーズ

これを２００台並べる！？
設置できるサイズにしない
といけない

基板３～４枚で
構成されるベク
トルプロセッサ
を基板１枚に！



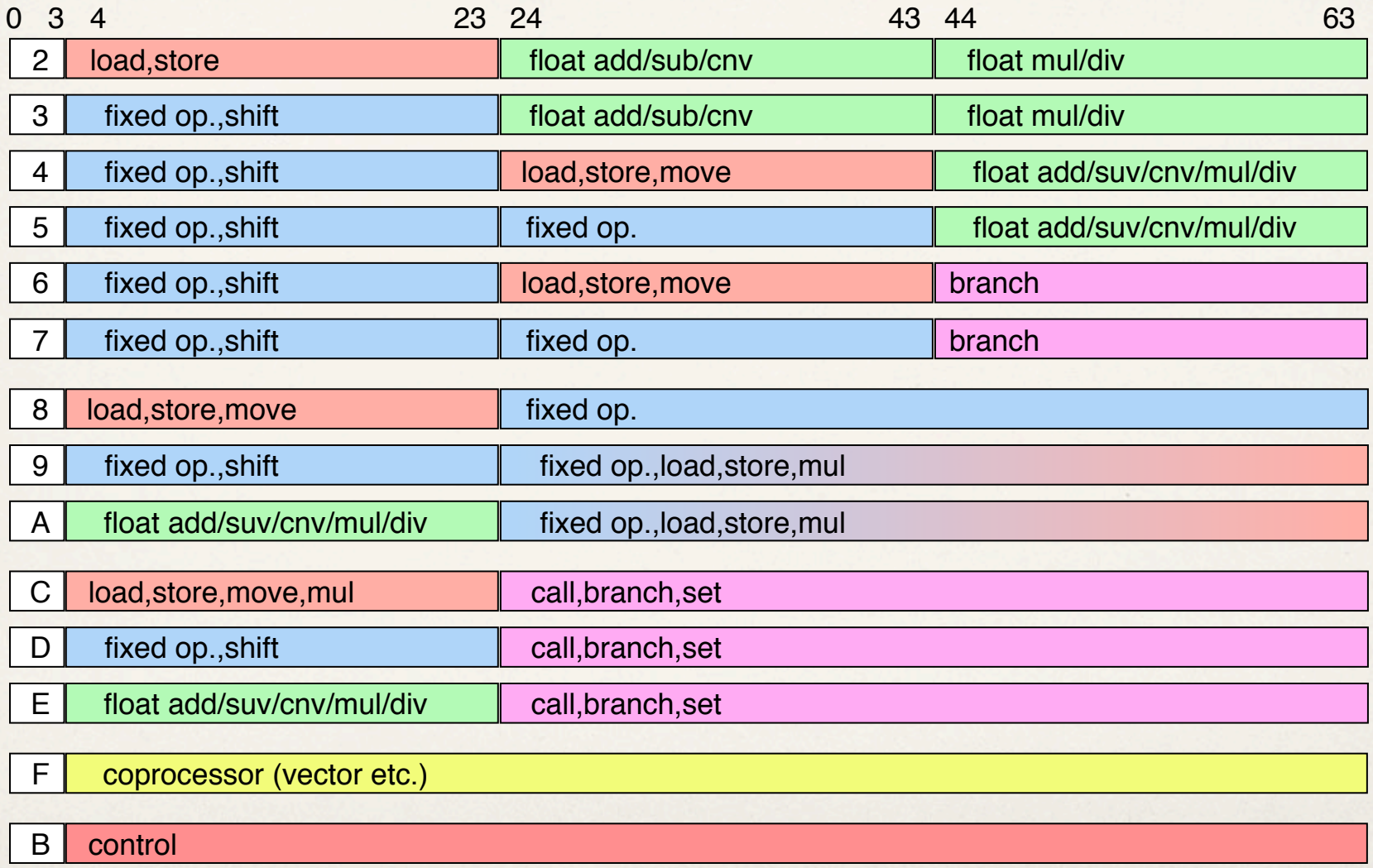
小さくするには…

- スカラプロセッサの大幅見直し
  - VPシリーズのスカラ命令は汎用計算機Mシリーズと共通
    - 命令セットの仕様が重い
    - スパコンに特化すると余分な命令も
  - １つの案として命令セットの部分セットとする
    - 利点は？？？
  - 並列実行RISCが脚光
    - RISC命令セットなら仕様も軽そう
    - 重装備ベクトルプロセッサの弱点であるショートベクトルをなんとかしたい（開発者の下心）
    - スーパスカラではなくLIWを選択

LIW命令セット：TaChyon

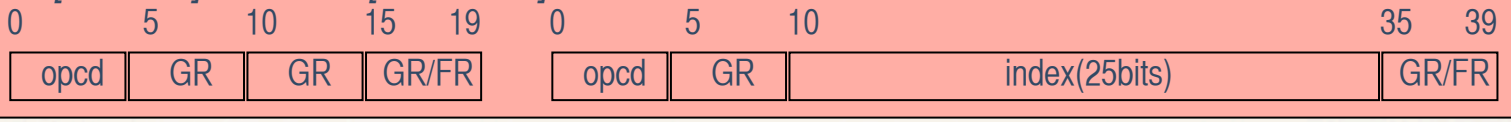
- パイプライン処理のノウハウとＲＩＳＣ／命令レベル並列処理の融合
- ＲＩＳＣの哲学：ハードウェアとソフトウェア（コンパイラ）のトレードオフの見直し
  - コンパイラによる最適化の可能性
  - ハードウェアはコンパイラではできないことに投資
  - スーパースカラよりＬＩＷ
  - メモリアクセスによる遅れに対して：メモリアクセスの非同期化
  - 浮動小数点演算のような複数サイクルの演算：非同期化
  - ＰＣ相対の分岐
- 命令長：固定64ビット、３～１操作（Veryと言うほどでないのでVLIWのVを抜いてLIWと勝手に言った）

TaChyon命令フォーマット

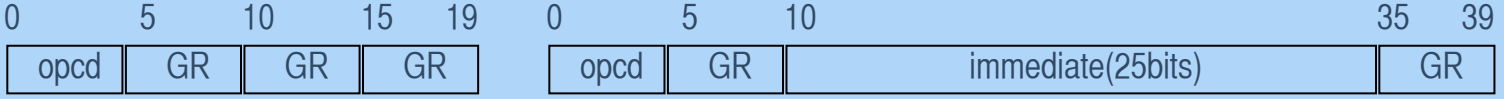


TaChyon操作形式(1)

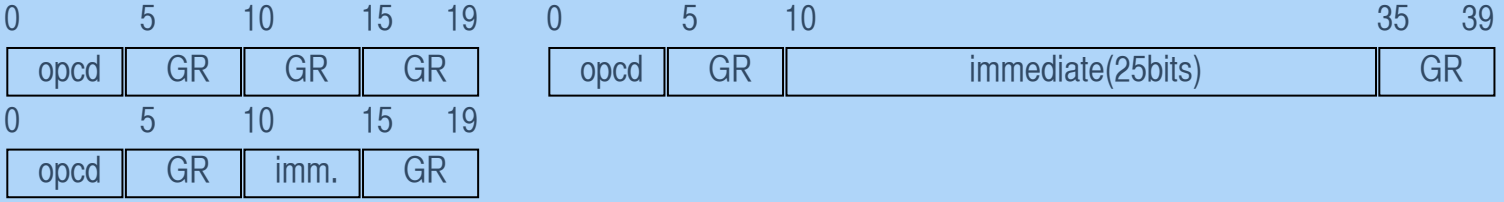
load,store *[GR+GR]→GR/FR [GR+index]→GR/FR*



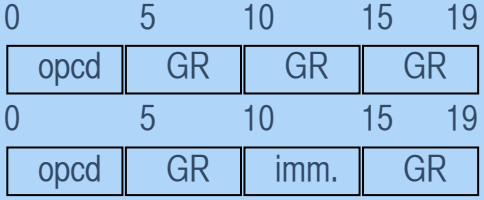
logical op.,multiply *GR∗GR→GR GR∗immediate→GR*



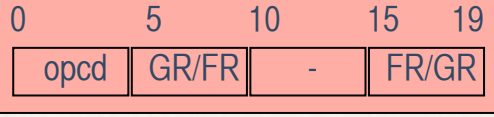
fixed op. *GR∗GR→GR GR∗immediate→GR*



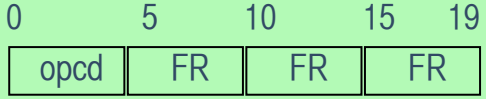
shift *GR∗GR→GR GR∗immediate→GR*



move *GR/FR→FR/GR*

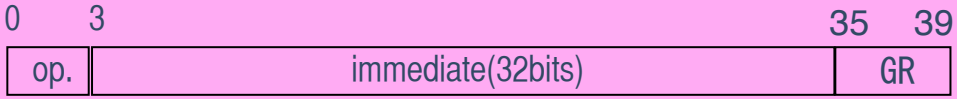


float op. *FR∗FR→FR*

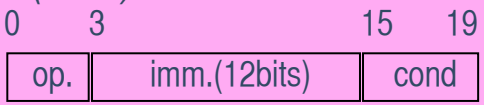


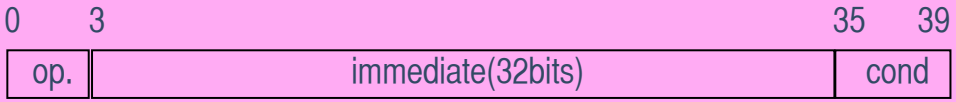
TaChyon操作形式(2)

call *PC→GR,PC+immediate→PC*
set *immediate→GR*

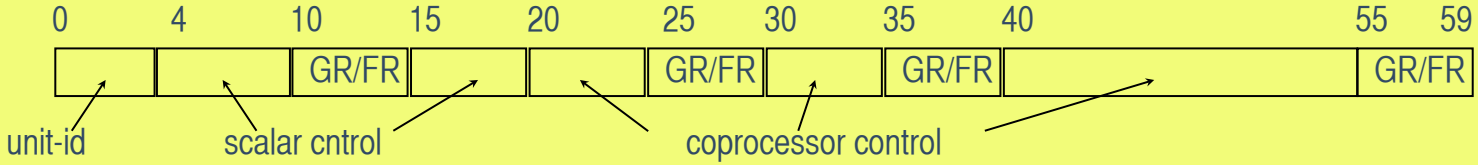


branch *if (cond) PC+immediate→PC*

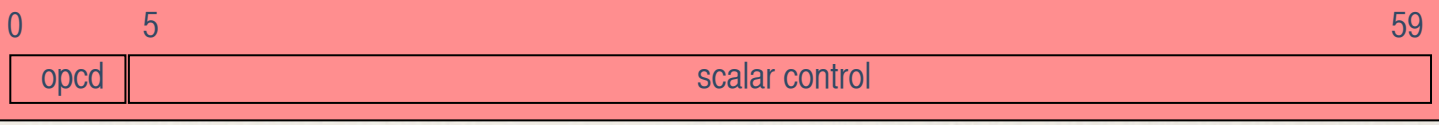




coprosessor



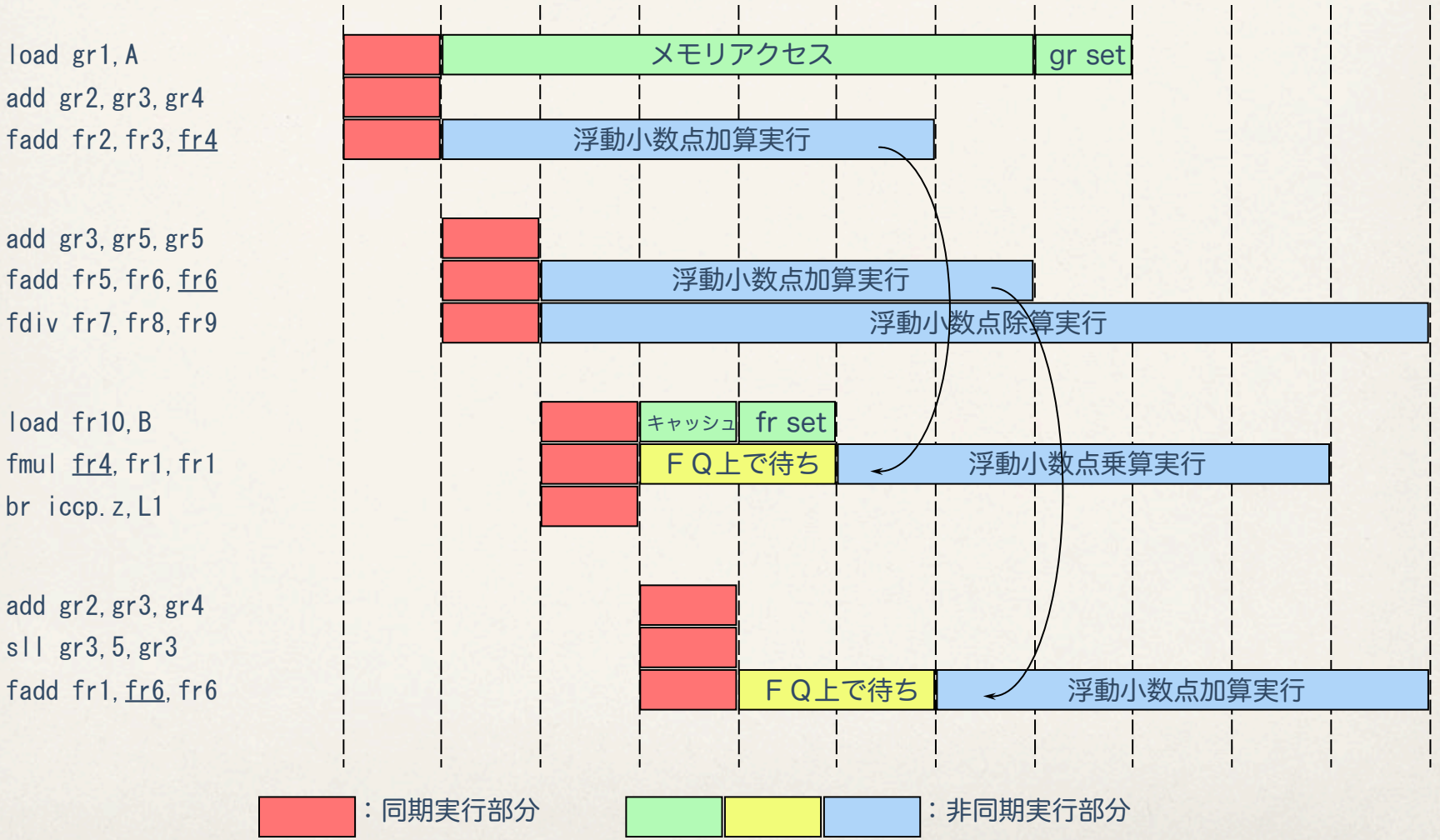
control



命令セットの特徴

- ６４ビット長ＬＩＷ（最大３操作を指定可能）
  - ハードウェアの軽量化
  - 命令サイズ増大の抑制
- ＰＣ相対アドレス指定による条件分岐操作
  - 分岐先アドレス計算、命令プリフェッチの高速化
- 非同期操作（乗算、浮動小数点演算、主記憶参照、ベクトル演算）
  - 複数サイクルを要する操作の突き放し
  - ハードウェアが実行順序を変更可能
  - ｇｒ０へのロード：ソフトウェア・プリフェッチ
- 例外および未実行操作の複数同時表示機構
  - 例外処理の際、割込み時のＰＣを必要としない
  - ハードウェアの軽量化／高速化

同期動作と非同期動作


load gr1,A
add gr2,gr3,gr4
fadd fr2,fr3,fr4
メモリアクセス
gr set
浮動小数点加算実行
add gr3,gr5,gr5
fadd fr5,fr6,fr6
fdiv fr7,fr8,fr9
浮動小数点加算実行
浮動小数点除算実行
load fr10,B
fmul fr4,fr1,fr1
br iccp.z,L1
キャッシュ
fr set
ＦＱ上で待ち
浮動小数点乗算実行
add gr2,gr3,gr4
sll gr3,5,gr3
fadd fr1,fr6,fr6
ＦＱ上で待ち
浮動小数点加算実行
：同期実行部分
：非同期実行部分

同期動作

- １命令中の操作が同時に行われる。
- 固定小数点演算、分岐操作

・ＸとＹの交換

	sub grX-0⇨grY	sub grY-0⇨grX	

・Ｘ１＝Ｙ１またはＸ２＝Ｙ２の場合、分岐

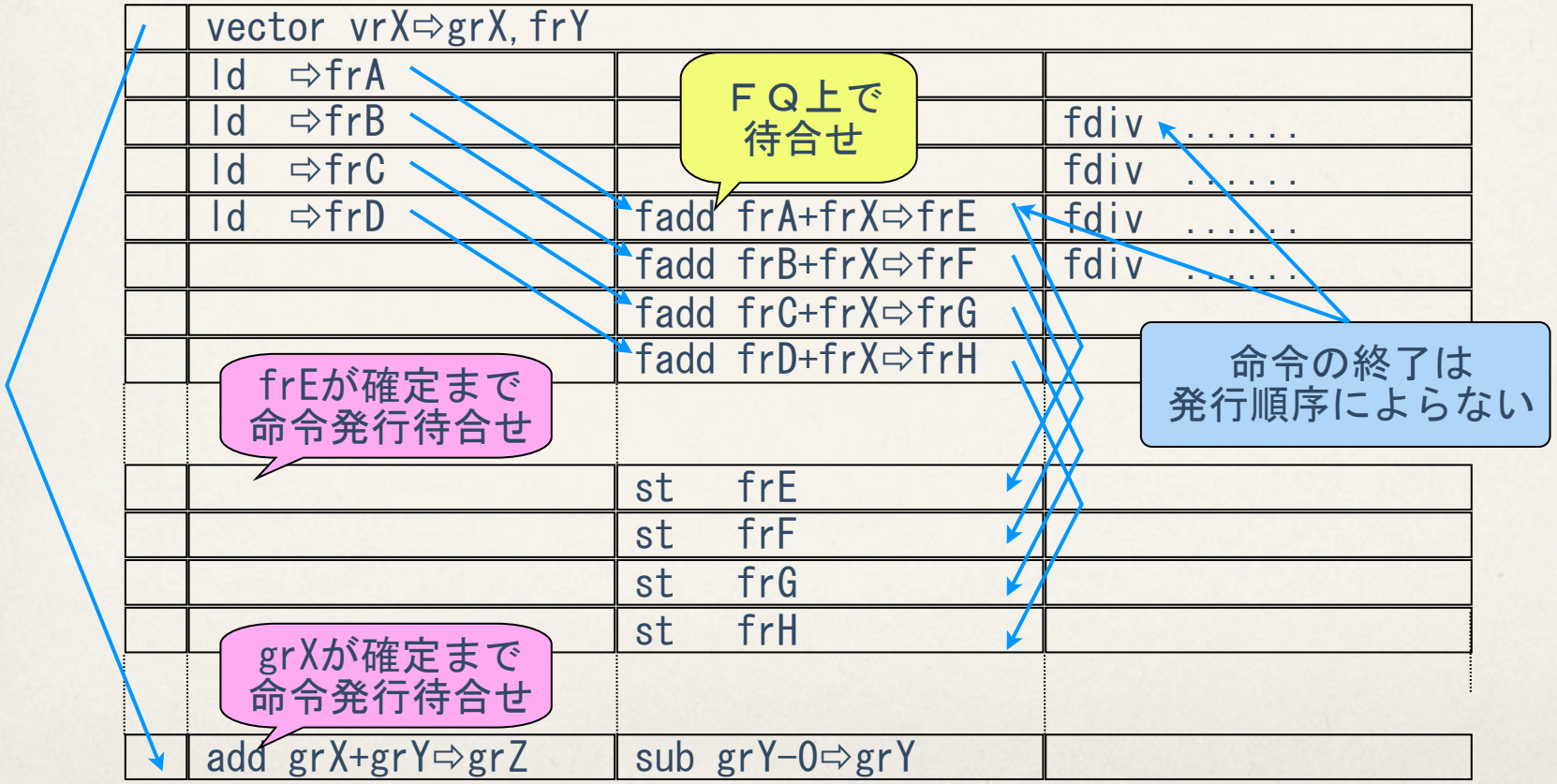
	sub grX1-grY1⇨gr0	sub grX2-grY2⇨gr0	
			brc oz,XXX

・ＮＵＬＬ文字の検出

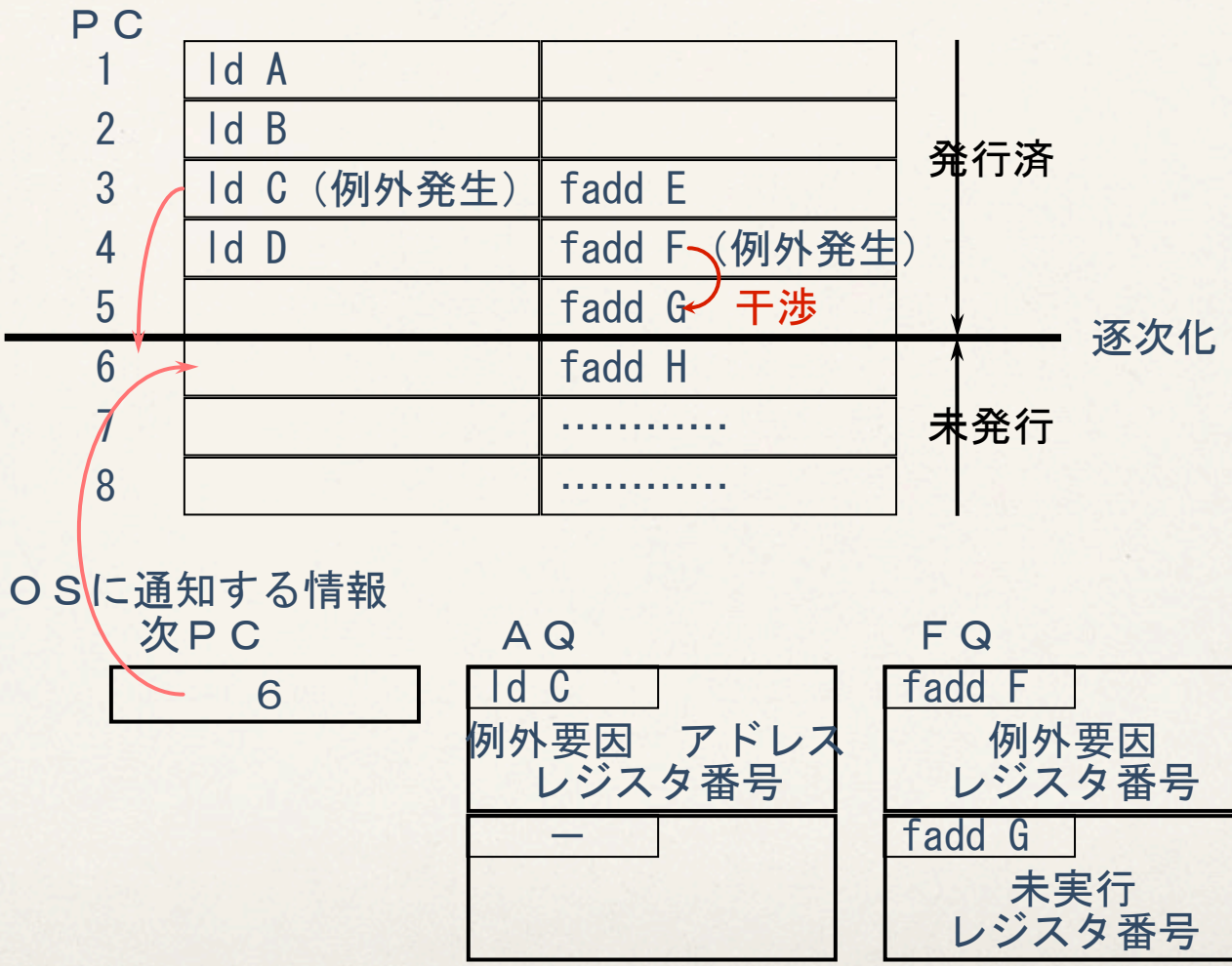
	xor grX1^0⇨gr0		
	xor grX2^0⇨gr0		brc iccp.v,XXX
			brc iccp.v,XXX

非同期動作

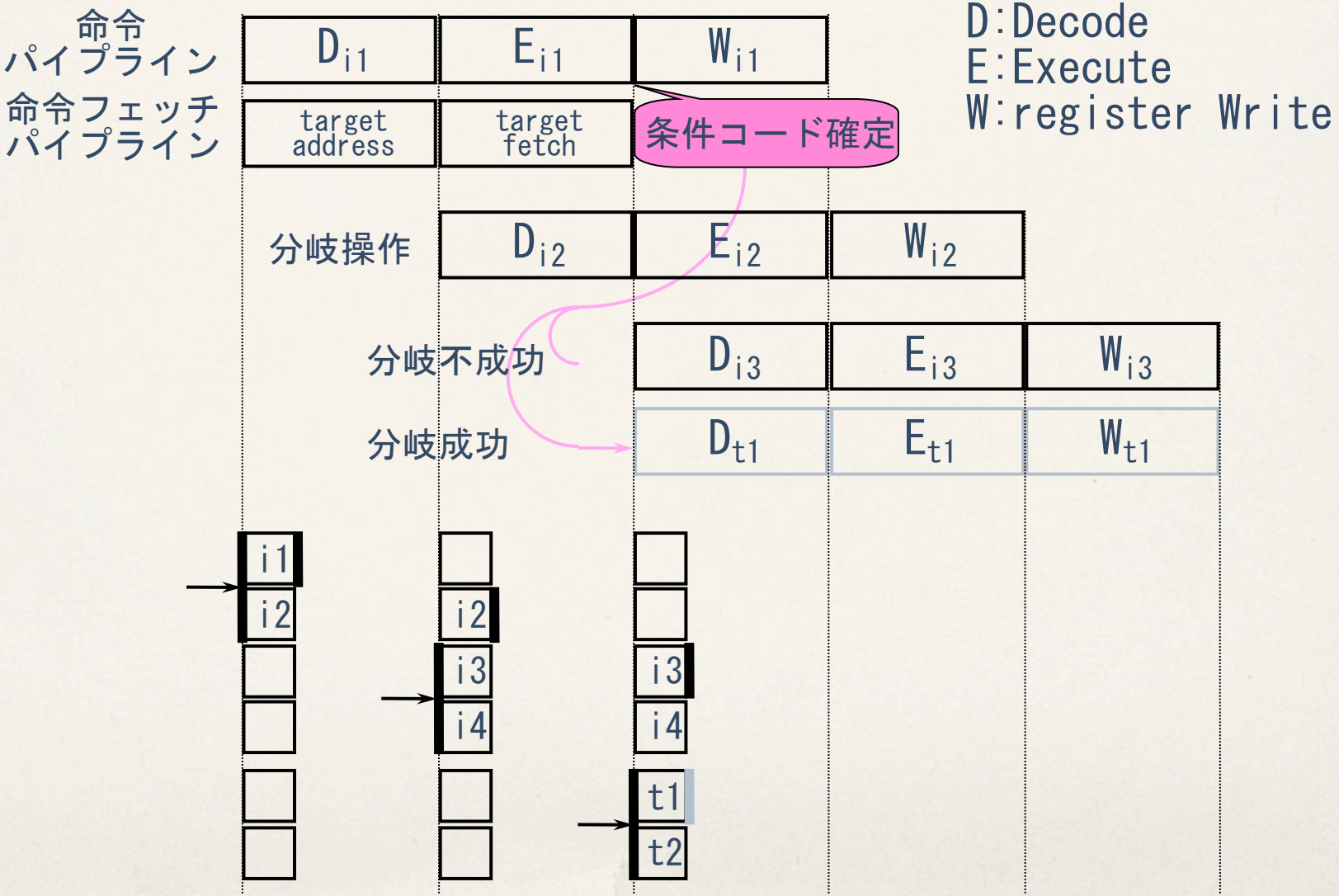
- 同期部分で終了しない操作
- 乗算、浮動小数点演算、主記憶操作、ベクトル演算



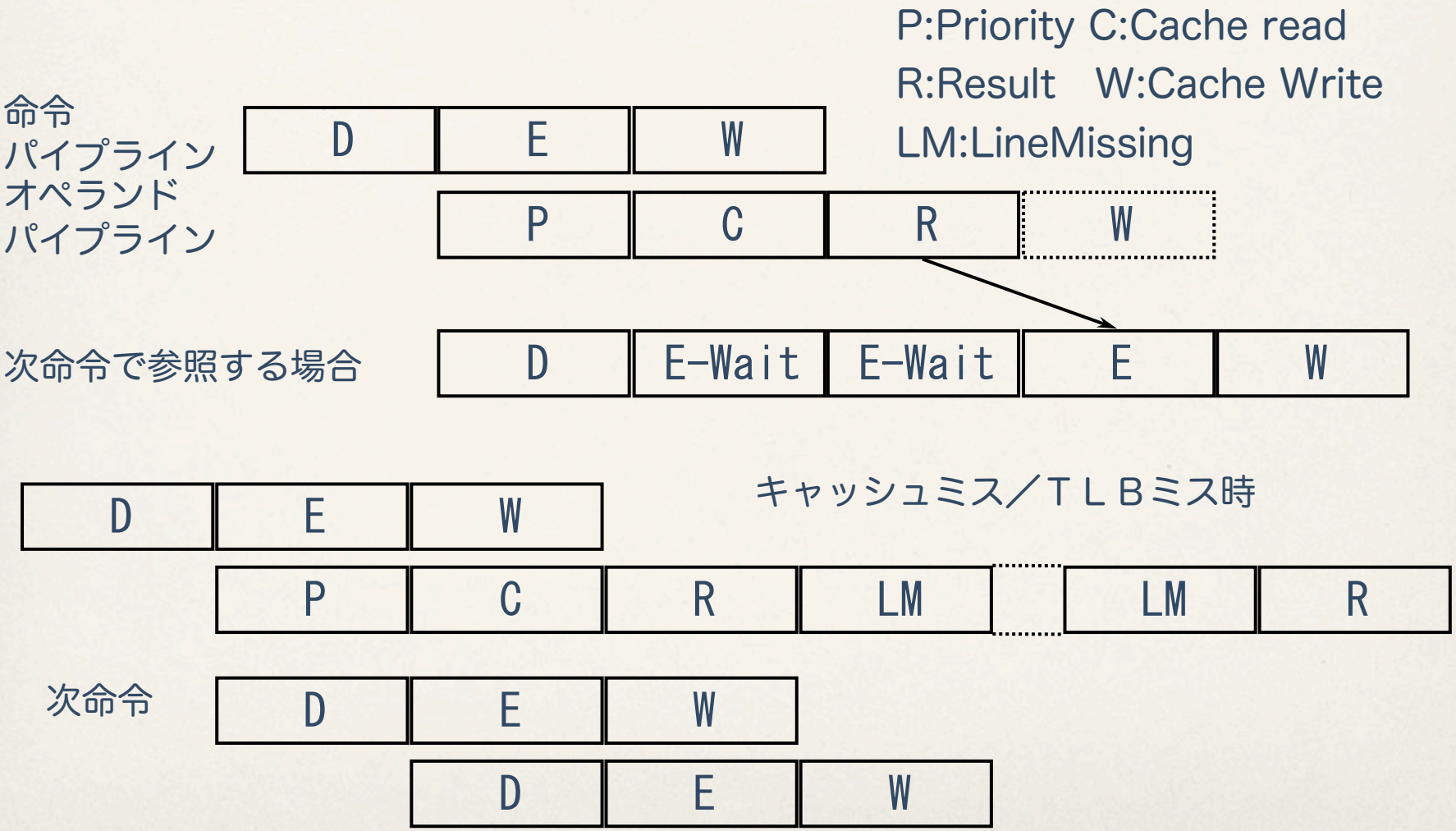
非同期動作の割込み


ＰＣ
1
2
3
4
5
6
7
8
ld A
ld B
ld C（例外発生）
ld D
fadd E
fadd F（例外発生）
fadd G
干渉
fadd H
.....
.....
発行済
逐次化
未発行
ＯＳに通知する情報
次ＰＣ
6
ＡＱ
ld C
例外要因　アドレス
レジスタ番号
ー
ＦＱ
fadd F
例外要因
レジスタ番号
fadd G
未実行
レジスタ番号

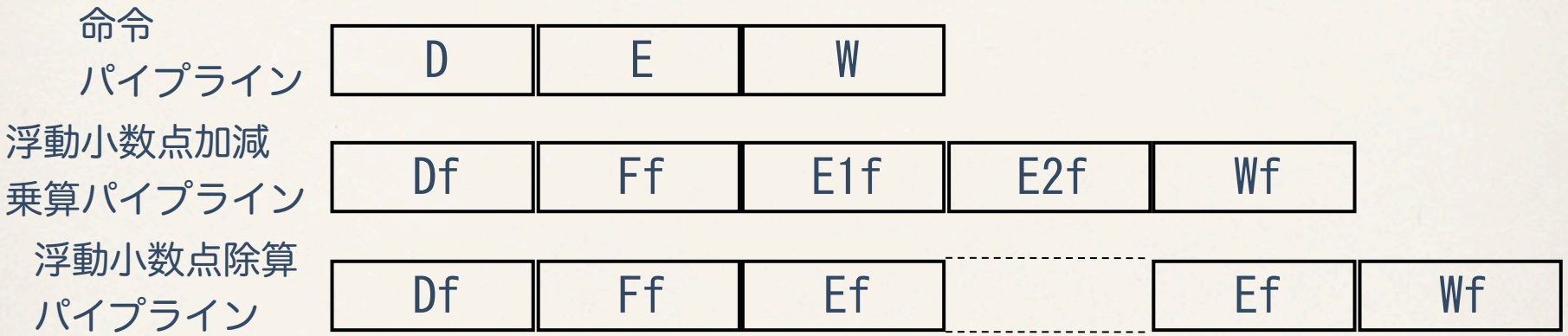
命令パイプライン(通常)


命令
パイプライン
命令フェッチ
パイプライン
D:Decode
E:Execute
W:register Write
D_{i1}
E_{i1}
W_{i1}
target address
target fetch
条件コード確定
分岐操作
D_{i2}
E_{i2}
W_{i2}
分岐不成功
D_{i3}
E_{i3}
W_{i3}
分岐成功
D_{t1}
E_{t1}
W_{t1}
i1
i2
i3
i4
t1
t2

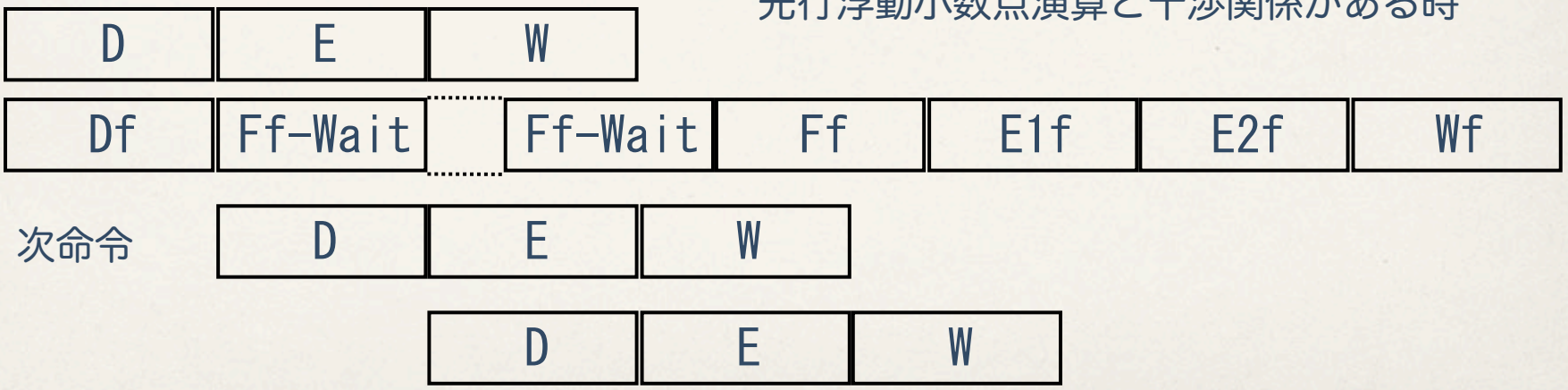
命令パイプライン（load）



命令パイプライン（浮動小数点）



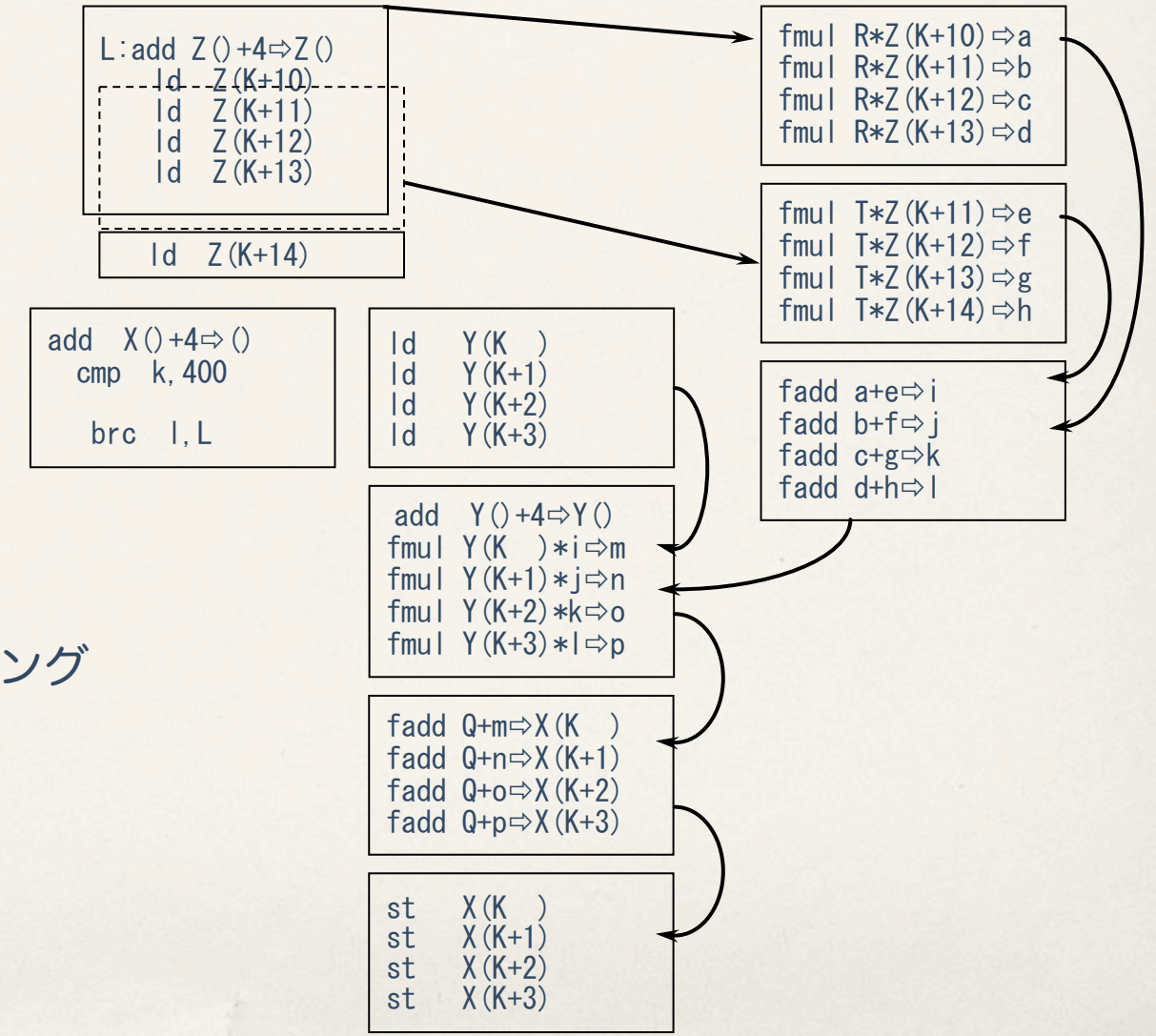
先行浮動小数点演算と干渉関係がある時



Livermore14ループ（1）

```
DO 1 K=1,400
1 X(K)=Q+Y(K)*
  (R*Z(K+10)+T*Z(K+11))
```

４重ループアンローリング
ソフトウェアパイプライニング



Livermore14ループ

L:

2	ld　Z(K+10)	fmul Y(K　)*i⇨a	
2	ld　Z(K+11)	fmul Y(K+1)*j⇨a	
2	ld　Z(K+12)	fmul Y(K+2)*k⇨a	
2	ld　Z(K+13)	fmul Y(K+3)*l⇨a	
2	ld　Z(K+14)	fadd Q+m⇨X(K　)	fmul R*Z(K+10)⇨a
3	add　Z()+4⇨Z()	fadd Q+n⇨X(K+1)	fmul R*Z(K+11)⇨b
3	add　Y()+4⇨Y()	fadd Q+o⇨X(K+2)	fmul R*Z(K+12)⇨c
2	ld　Y(K+3)	fadd Q+p⇨X(K+3)	fmul R*Z(K+13)⇨d
2		st　X(K　)	fmul T*Z(K+11)⇨e
2		st　X(K+1)	fmul T*Z(K+12)⇨f
2		st　X(K+2)	fmul T*Z(K+13)⇨g
2		st　X(K+3)	fmul T*Z(K+14)⇨h
4	add　X()+4⇨X()	ld　Y(K　)	fadd a+e⇨i
4	cmp　K,400	ld　Y(K+1)	fadd b+f⇨j
4		ld　Y(K+2)	fadd c+g⇨k
E	brc　l,L		fadd d+h⇨l

・コンパイラによる詰め込みを適用
１２５ＭＦＬＯＰＳ　３８操作／１６命令（１２８バイト）＝２．４
スーパースカラ方式：３８命令（１５２バイト）

性能

リバモア１４ループ

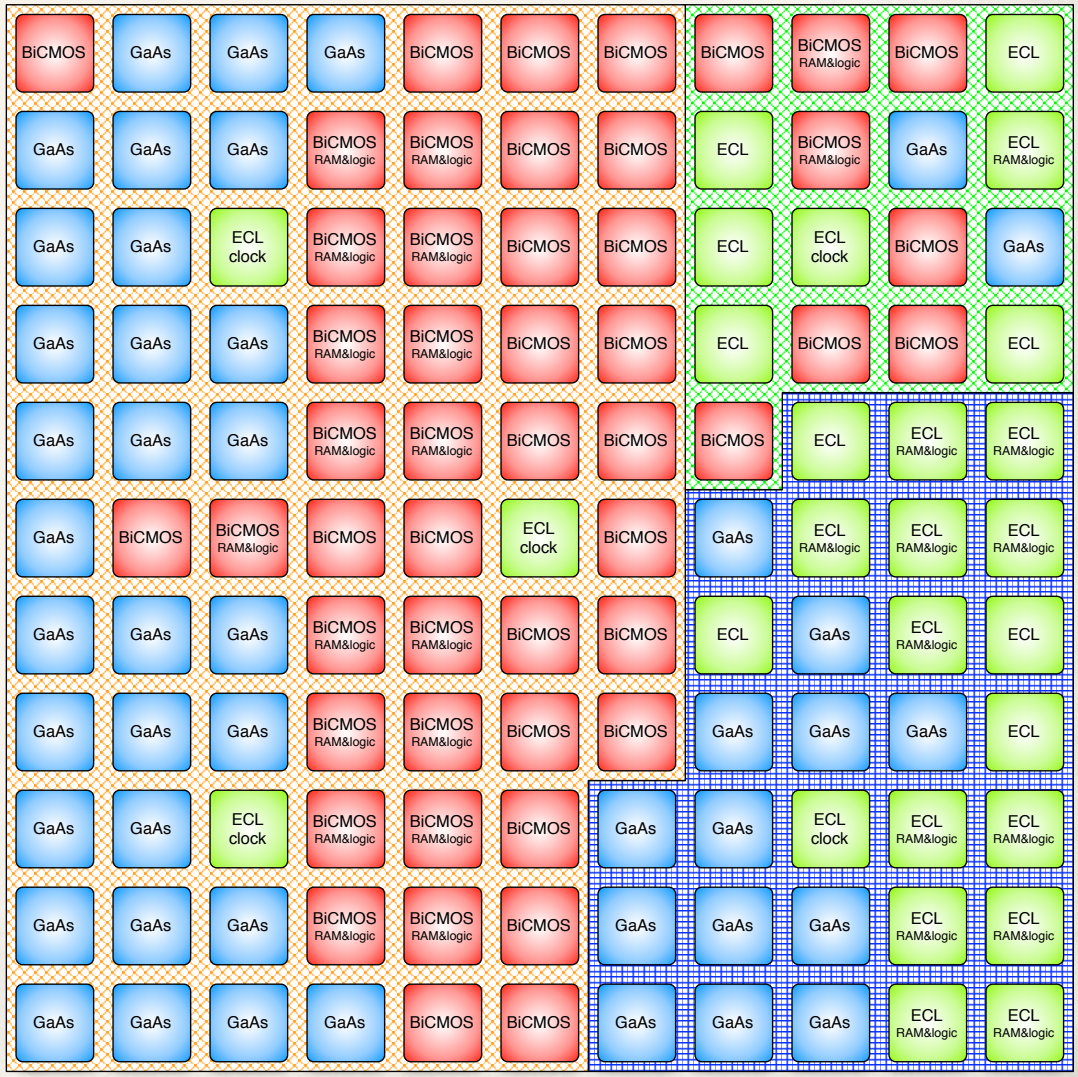
リバモア Loop#	VP2600 スカラ（6.4ns） MFLOPS	VPP500 スカラ（10ns） MFLOPS
1	47.3	135.5
2	47.2	72.5
3	50.1	72.8
4	35.1	38.7
5	42.5	20.0
6	40.1	22.8
7	55.1	103.6
8	43.5	72.5
9	48.2	73.5
10	32.5	21.5
11	43.9	16.7
12	28.1	43.5
13	9.5	8.7
14	19.5	16.9
単純平均	38.8	51.4

SPECfp92

	052.alvinn	078.swm256
命令／サイクル	0.63	0.71
操作／サイクル	1.05	1.23
操作／命令	1.66	1.73
命令キャッシュヒット率	99.69%	99.99%
オペランドキャッシュヒット率	96.40%	96.27%
ＭＦＬＯＰＳ	38.88	66.38
SPEC ratio	307.6	165.5

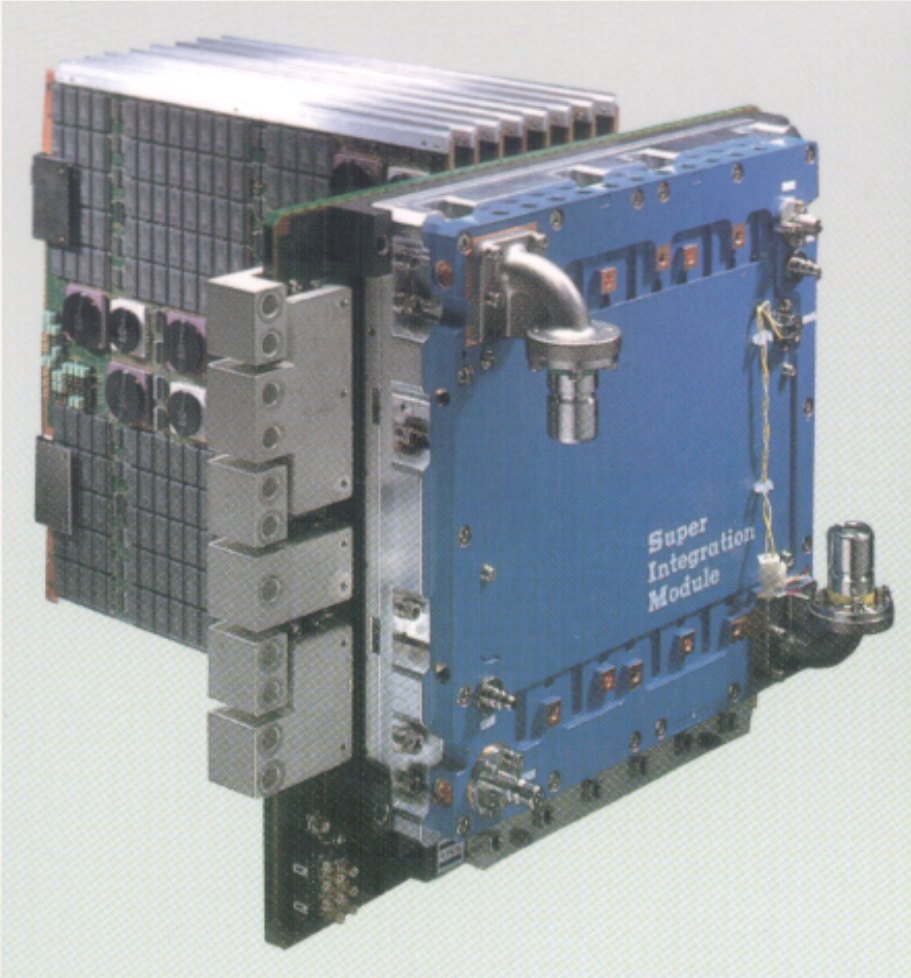
GaAs-LSI：期待と泥沼…

- いくら命令セットが簡単になっても、LSIの集積度が欲しい…、しかも現行機並の速度も欲しい
- ECL LSI
  - 15,000ゲート、遅延時間70ps
  - 64kビット／アクセス1.6ns＋3,500ゲート
- GaAs LSIは？
  - 25,000ゲート、遅延時間60ps
  - RAMは用意されてない
- BiCMOS LSIは？
  - 72,000ゲート、遅延時間200ps
  - 72kビット／アクセス7ns＋24,000ゲート
- 適材適所？無節操？テクノロジー博物館？
- GaAs LSIのFirst userは痛い目に…



エンジンが良くても足回りが…

- ベクトルプロセッサの鉄則
- 高性能の処理装置にデータをよどみなく供給する！
- 高スループット→バス幅特大、距離最短
- メモリ素子：汎用機の二次キャッシュ用SRAMで主記憶を構成
- 無茶なことをしたツケが…



パスコンは安定動作祈願のお札？

- メモリボードの安定動作が確保できない
- いろいろ調べると……ノイズ問題

↓

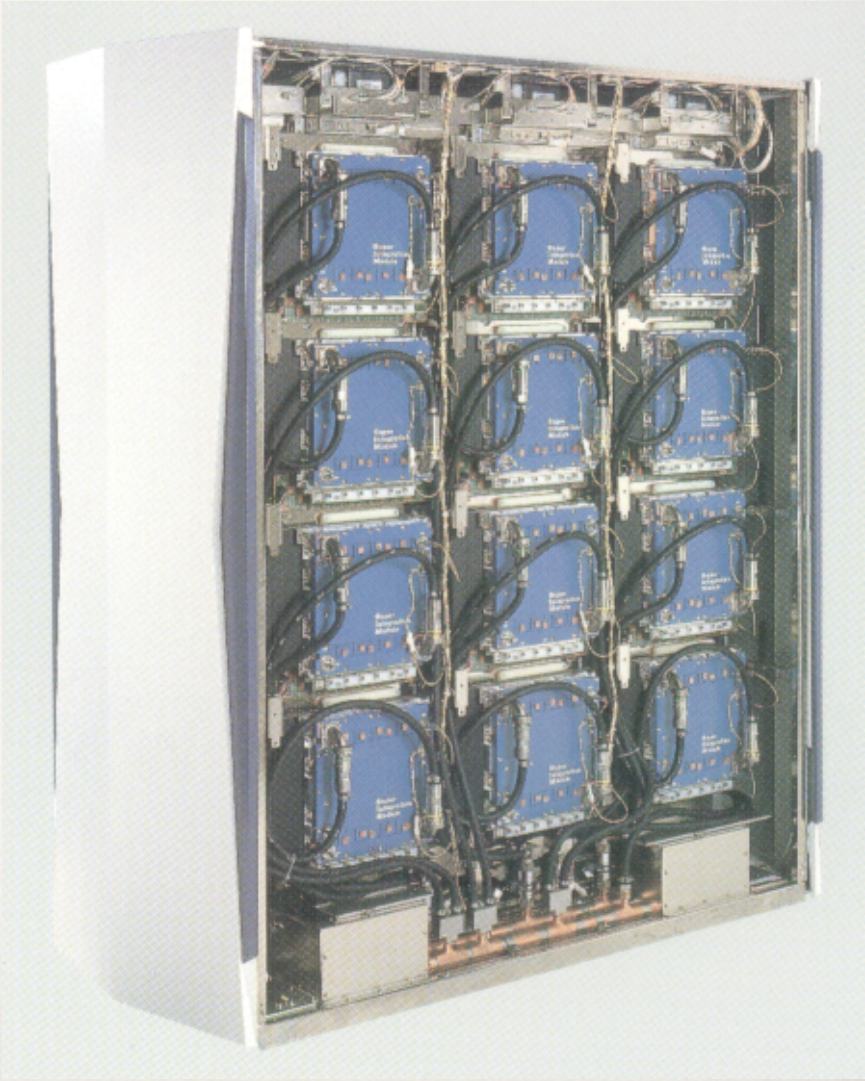
- ノイズにはコンデンサを付けるしか無い
- しかし、どこに？　試行錯誤の繰り返し
  - 付けては測定、外して違う場所につけたり、容量を変えたり…
  - ついにパスコンのサブ基板を基板の上にのせたり…

↓

- 極限的な使用方法には、極限の設計ノウハウが必要で蓄積される

陽炎の立つマシン：熱との戦い

- プロセッサエレメントを縦４段に実装
- 水冷のCPU側は、２枚のモジュールで１ループを構成、水さえ回れば冷却可能
- メモリボードは空冷
  - ４段分の熱が上へ吹き出す…
  - 天板の穴空きパネルをさわると熱い！
  - 冷却設計に渡した発熱量を上回っている…
  - 冷却の改造を行って何とか規定値に
  - それでも天井の低い計算機室では陽炎が立つ…



力ずくのクロスバー結合！？

- プロセッサエレメント間の結合網に対する一般的評価
  - ハイパーキューブ：台数の増加に対して$\log_2$ N本の接続で、メッシュよりバンド幅取れそうだし自由度もそれなりに有りそう
  - ２次元メッシュ：台数が多くなるとこれしかなさそうでも、バンド幅とか転送の自由度とかは…
  - クロスバー：自由度高いけど、数百台でこれはないよね

↓

- 解きたい問題はこれ。そのために一番良い解法としてこれを使いたい。だから、各プロセッサに分散配置した配列部分を簡単に転置できないと困る。
  - 何が何でもクロスバーしか有り得ない…

プロセス間の同期

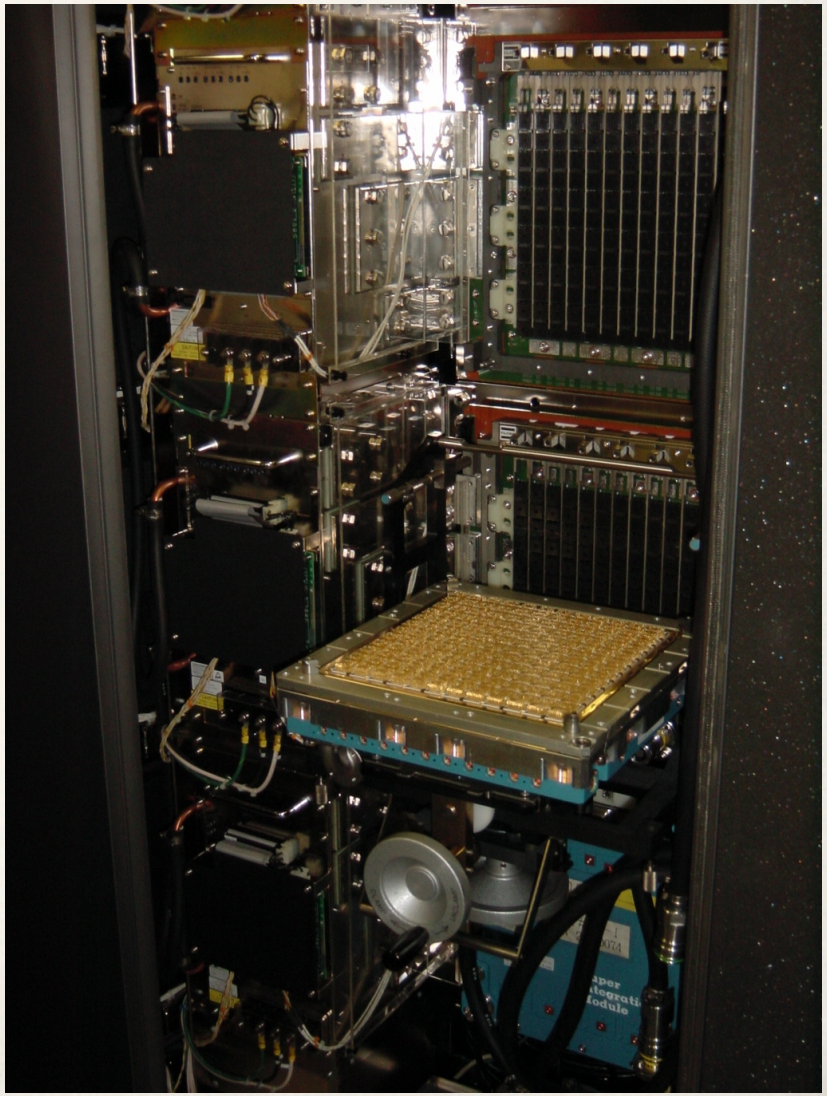
- 仕様的に目処がついた頃、「並列計算機は繋いであれば良いと言うもんじゃないでしょう」と並列仕様グループにちょっかいを出す
- 「バリア同期の高速化」を考える
- バリア同期ハードウェア⇔マルチ並列プロセス（@PE、@システム）
  - 論理的なプロセスと物理的な同期機構とのマッピング
  - 仮想計算機アシストの考え方を用いて、自由度を持たせる
  - これはちょっと趣味に走りすぎたかな…

稼働開始

- 並列計算機の製造問題
  - 汎用機ならシステムあたりのCPU基板はせいぜい４枚
  - 並列計算機は、一挙に１４０枚！
  - しかもシステムの受注が平均化しているわけではない
  - 製造が平準化できない→工場泣かせ（しわ寄せは技術部）
- やっと稼働に漕ぎ着けた
  - 皇太子ご成婚パレードの日に、LINPACKを測定
  - 念のために、保守部品を抱えて計算機室待機
  - 第１位／TOP500も頂けた
- 部品点数の多さに、保守では大変ご迷惑をかけてしまった

NWT開発で得られた知識と技術

- 水冷モジュールの交換が早くなった
  - フィールドCEに感心された…
- 会社のオペレーションに詳しくなった
  - スパコンは小所帯のグループで何でもやる。半導体の入荷状況／工務／製造工程／試験／営業支援
  - お客様に近い：生の声が聞ける



NWT開発で得られたもの

- 当時の技術の限界
  - 高速論理素子：ECLの次は？
  - CMOSやカスタムLSI設計にますます興味を持つ
  - テクノロジーの牽引車はやっぱりスパコンか
- 「できそうなことをやる」じゃなくて、「実現したらこんなに良いことがある」ということに挑戦する
  - アメリカのベンチャースピリットに通じるものがあるような…
  - ユーザがやりたい事を実現するスパコンを作る
  - 無茶と思っても、案外解決策が見つかる

今後…

- 「京」は１位／Top500を取れたが、「世界一」の日本製品は？
  - 最近のエンジニアは、「世界一」の成功体験を持つことが難しそうで、気の毒…
- このごろの学生を見ていると将来が心配
  - ますます衰退していくのではと心配になる
  - 「使い物になるエンジニア」を育てたい

ご清聴ありがとうございました