

計算機を創る

~~ やっぱり一番でしょう ~~

東京大学情報理工学系研究科

平木 敬

最終講義

Tokyo

IEEAF

Seattle

JGN2

Chicago

SURFnet

Amsterdam



はじめに

「最終講義は何のためにするか謎だった」

「なぜ最終講義の後が祝賀会かはもっと謎だった」

- 最終講義する側になって良く判った
 1. 取りあえず病気もせず、暗殺もされずに最終講義
 2. 講義をする重荷から解放される！
 3. 今後好きなことが出来ることがとても嬉しい！

お世話になった皆様に心から感謝します

担当している講義(今年)

1. 計算機構成論 アーキテクチャの基本
2. 分散並列論 アーキテクチャとネットワーク
3. Modern Computer Architecture 英語講義
4. 分散並列システム論 並列システムの基礎
5. 情報理工学倫理 研究倫理と情報倫理
6. プログラミング技法(駒場) プログラム最適化等
7. ハードウェア実験 論理回路とHDLの実験
8. プロセッサ実験 FPGAでプロセッサを作る

最終講義の前座として、過去最多の講義を行った

担当していた講義(過去)

1. アーキテクチャ I (アーキテクチャの初歩編)
2. アーキテクチャ II (アーキテクチャの応用編)
分岐予測、OoO、並列キャッシュは扱わなかった
3. データ構造とアルゴリズム(以前は本郷開講)
4. 情報システム論 (駒場4学期)
5. ネットワークシステム
6. ハードウェア構成法 (論理回路の基礎)

講義の便利屋？

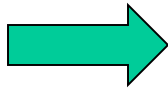
講義のねらいは全部同じ

- 情報処理・計算の中身を知って好きになってほしい
- 物を作り上げる楽しさと大変さを知ってほしい
- 計算機を作ることを仕事にする人が増えて欲しい

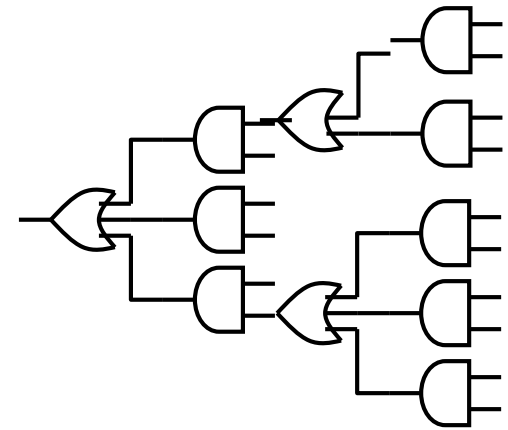
最初は「計算機構成論」に関する
最終講義

コンピュータを作る意義

- 世界と知の根本を
 - 解析的方法論
 - 合成的方法論で探求



各講義の出だしで使うスライド



+

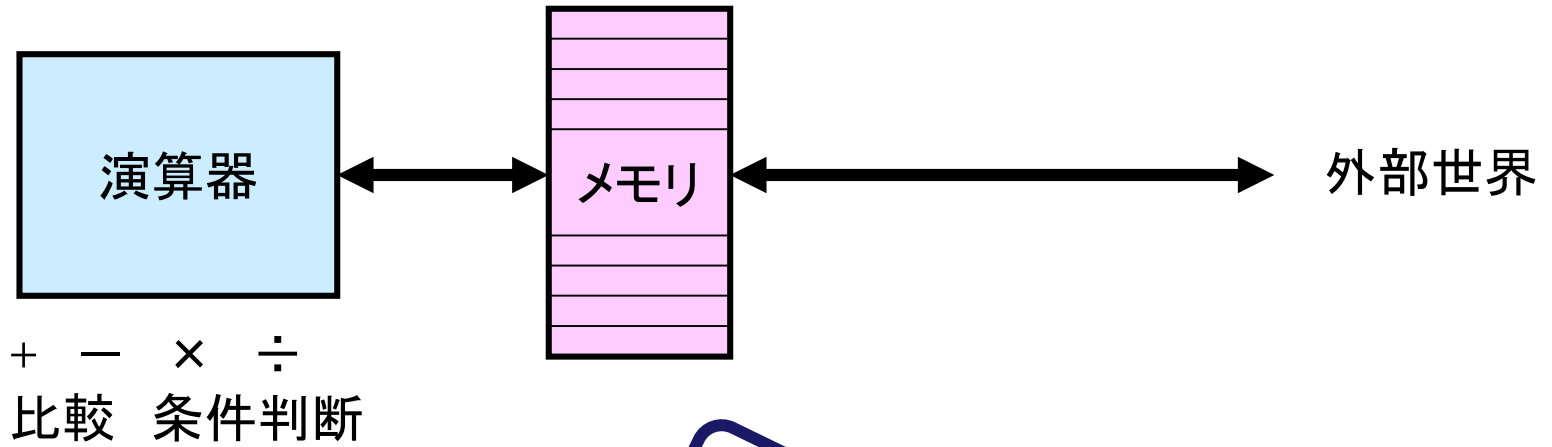
main:

```
!#PROLOGUE# 0  
save %sp,-120,%sp  
!#PROLOGUE# 1  
nop  
st %g0,[%fp-20]
```

.LL2:

```
ld [%fp-20],%o0  
cmp %o0,9  
ble .LL5  
nop  
b .LL3  
nop
```


コンピュータの万能性



1. 計算機に個性はない
2. 全分野の計算・情報処理は同一システムで処理可能
3. チューリングの意味での計算可能性

各講義の出だしで使うスライド

例えば自動車



各講義の出だしで使うスライド

1. 目的ごとに異なる自動車が必要
2. 速度、乗車人数、積載量、重量など多くの特徴
3. そもそも、性能が指数関数的に向上しない

コンピュータ作りの歴史

- 初期の非数値計算機たち

- 作表統計機

- 一つのテーブルの計算
 - 統計処理

- カード、紙テープ、タイプライタ

- 初期の数値計算機たち

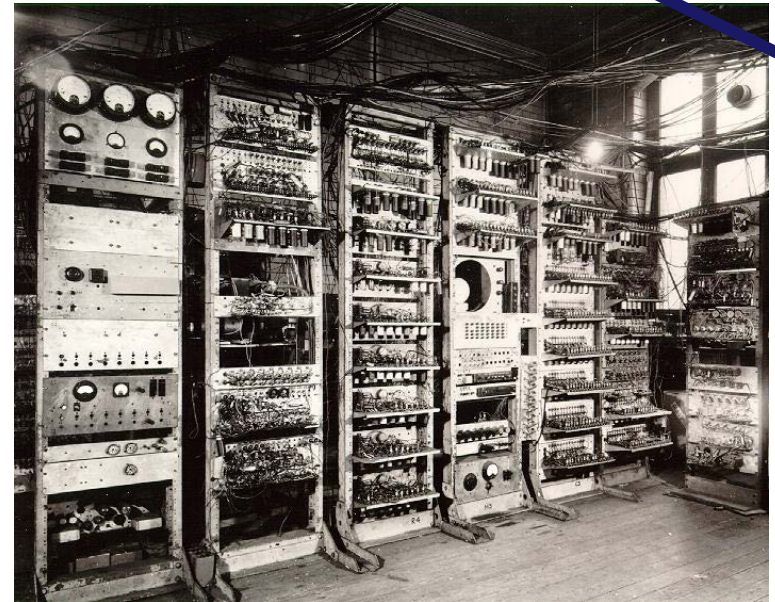
- 第一世代

- 方程式の求解
 - 数表作成

各講義の資料で使うスライド



ホレリス統計機



マンチェスターMARK-1

高速化の歴史を振り返る

- 20年 {
 - 1 MFLOPS
 - 1964, CDC 6600
 - ILP (パイプライン並列)、Out of order実行、スコアボード
- 13年 {
 - 1 GFLOPS
 - 1984, Cray XMP/4
 - ベクトルプロセッサ、SMP
- 11年 {
 - 1 TFLOPS
 - 1997, ASCI-Red
 - MPUを用いたクラスタ計算機
- 12年 {
 - 1 PFLOPS
 - 2008, IBM Roadrunner (Cell base)
 - On Chip Multi CPU, 超並列プロセッサ
- 13年? {
 - 1 EFLOPS
 - 2020年?
 - アクセラレータ付超並列プロセッサ? 積層半導体?
 - 1 Zflops
 - 2033年??
 - 超超メニーコア?

世界一のスーパーコンピュータの歴史

	マシン名	動作開始	Linpack性能	ピーク性能
1.	UNIVAC LARC	1960		(0.16Mflops)
2.	IBM STRECH	1961		(0.3Mflops)
3.	CDC-6600	1964	0.5Mflops	(3 Mflops) *N=100 Linpack
4.	CDC-7600	1969	3.3Mflops	(10 Mflops)*N=100 Linpack
5.	TI ASC	1972	~30 Mflops	(64 Mflops)
6.	ILLIAC IV	1975	~40 Mflops	(150 Mflops)
7.	Cray-1	1976	110 Mflops	(160 Mflops)*N=1000 Linpack
8.	Cray-XMP4	1982	714 Mflops	(800 Mflops)
9.	SX-2	1985	885 Mflops	(1.3Gflops)
10.	Cray-2	1985	1.4Gflops	(1.9Gflops)
11.	CM-2	1987	2.4Gflops	(5 Gflops)
	(ETA-10	1988	496 Mflops	(9.1 G (single)/4.6G(double) /8proc) 並列動作は不動 *N=1000 Linpack, 1 proc. 7ns)

- 世界一マシンの何れも、出来上がるまでに様々なドラマがある
- 世界一マシンの影には、名前も知られず消え去った数多くの2位以下のマシンがある

世界一のスーパーコンピュータの歴史

	マシン名	動作開始	Linpack性能	ピーク性能
12.	SX-3/44R	1990	23.2Gflops	(25.6Gflops)
13.	CM-5	1993	60 Gflops	(131 Gflops)
14.	Fujitsu NWT	1993	124 Gflops	(236 Gflops)
15.	Intel Paragon XP	1994	143 Gflops	(184 Gflops)
16.	Fujitsu NWT	1994	170 Gflops	(236 Gflops)
17.	Hitachi SR-2201	1996	220 Gflops	(307 Gflops)
18.	Hitachi CP-PACS	1996	368 Gflops	(614 Gflops)
19.	Intel ASCI RED	1997	1.1Tflops	(1.5Tflops)
20.	IBM ASCI White	2000	4.9Tflops	(12.4Tflops)
21.	NEC ES	2002	35 Tflops	(40.1Tflops)
22.	IBM BlueGene/L	2004	71 Tflops	(92 Tflops)
23.	IBM Roadrunner	2008	1.0Pflops	(1.4 Pflops)
24.	Cray XT-5	2009	1.8Pflops	(2.3 Pflops)
25.	天河-1A	2010	2.5 Pflops	(4.7 Pflops)
26.	K-computer	2011	10.5 Pflops	(11 Pflops)
27.	BlueGene/Q	2012	16 Pflops	(20 Pflops)
28.	Cray XK7	2012	17.6 Pflops	(27 Pflops)
29.	天河-2	2013	33.9 Pflops	(55 Pflops)
30.	神威・太湖之光	2016	93.9 Pflops	(125 Pflops)

私がコンピュータを作る動機

- 41年で、約1000万倍の速度向上、1000万倍のメモリ増大
- 大域ネットワーク
 - 50Kbps(1969)から40Gbps(2004))35年で100万倍の速度増加

速いものを作りたい！

高速化の方法は様々

ハードウェア

OS

言語

アプリケーションソフト

私が創った計算機たち

- 学部4年から現在まで
- 作るのは本当に大変だが、完成で忘れる
- 完成まで1年から10年前後まで様々
 - 論文執筆効率が本当に悪い
 - 出来た時は時代遅れの恐怖
- 今日は私の**黒歴史**を全披露します

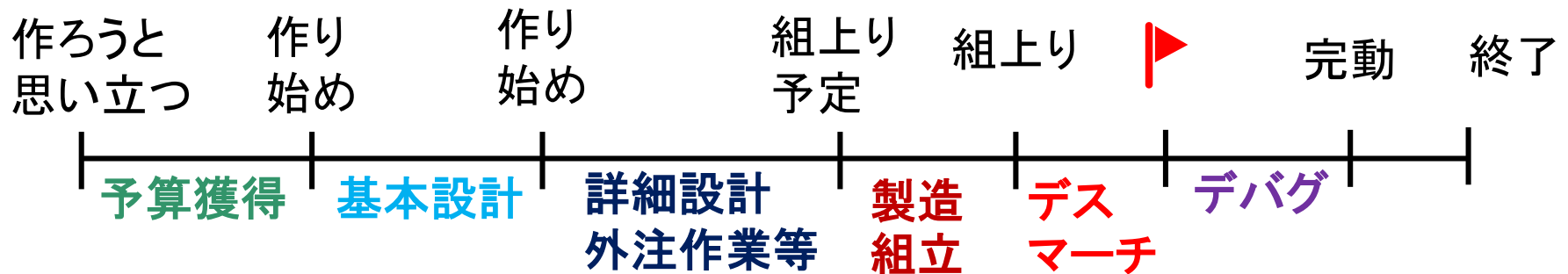
黒くない計算機は稀

- 最後まで動かなかった
- 完成が大きく遅れた(納入後も動かないとか)
- バグが最後まであった
- 不安定ですぐ止まる
- 予定の性能が出なかった
- 誰も使ってくれなかった

計算機ができるまで

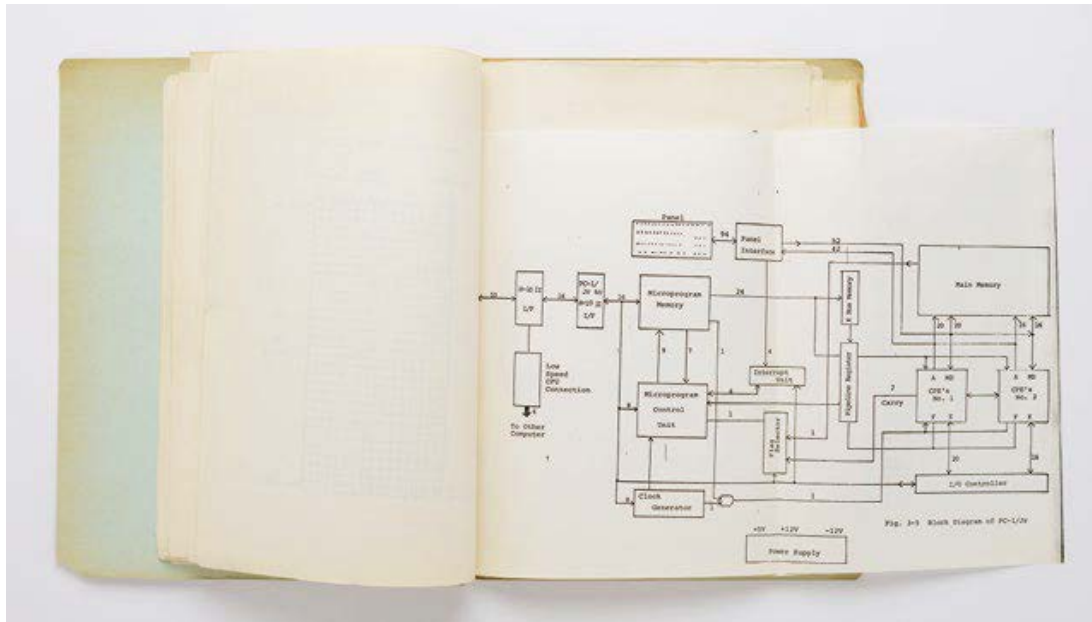
- 計算機により各区分の時間配分は様々
- 普通、2年から8年かかる
- 国のスパコンでは

— マル超	1966～1972	6年
— スーパーコンピュータ	1981～1989	9年
— 地球シミュレータ	1996～2002	7年
— 京コンピュータ	2005～2012	8年
— ポスト京	2014～2022?	9年？



最初に作ったもの

- 学部4年卒業研究(高橋秀俊研究室)
- ITVインタフェース装置(1975～76)
 - HITAC 10にテレビカメラから入力
 - 同期信号に合わせてACDでサンプリング
- 安定動作して10年くらい五月祭で使われた



卒業研究からブロック図

本体写真はありません

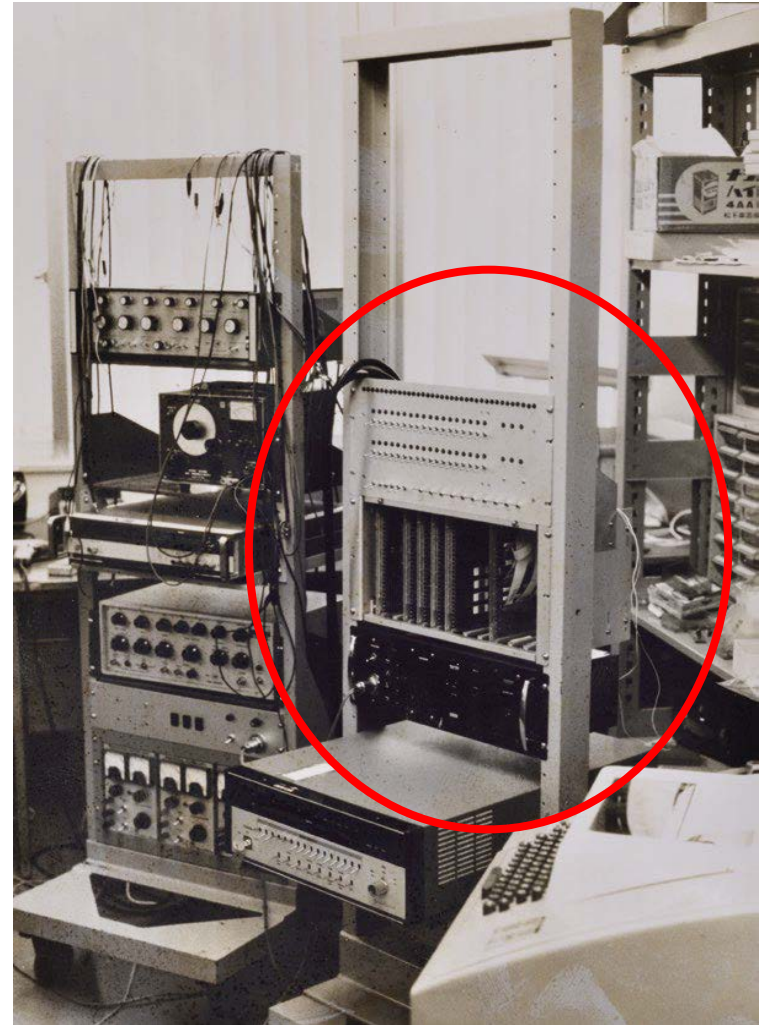
PC-1と高橋・後藤先生



(高橋滋子氏、高橋信夫氏のご好意による)

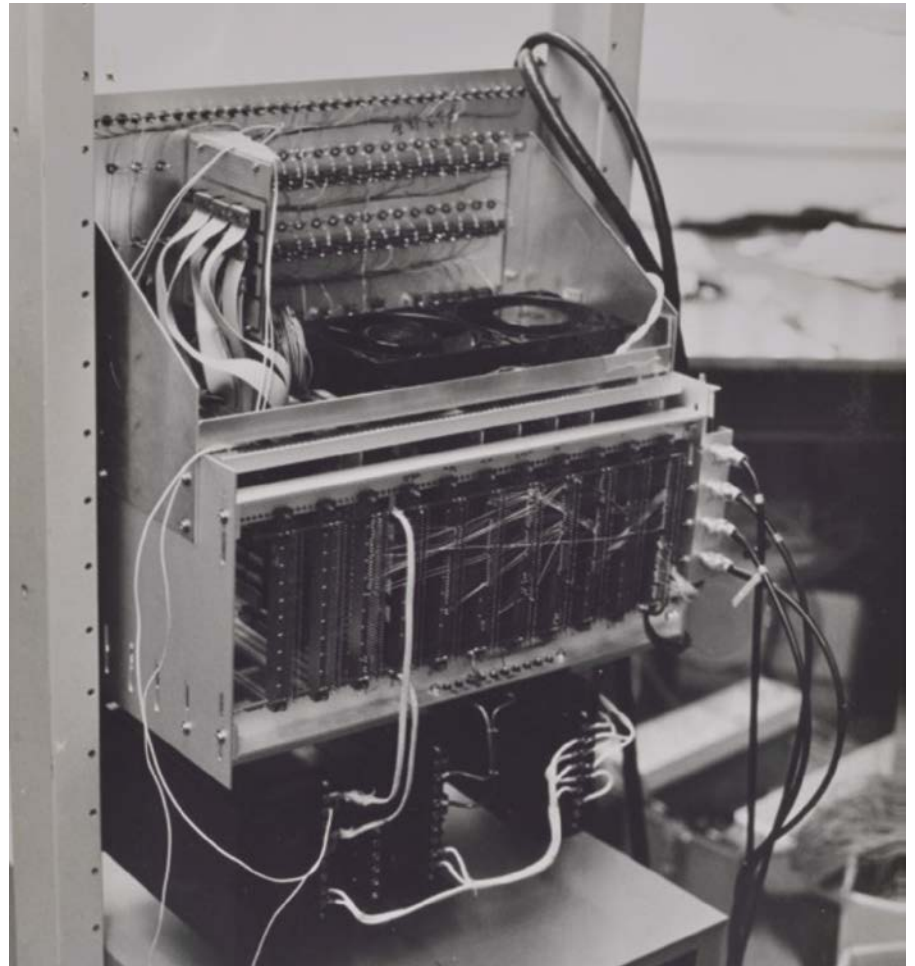
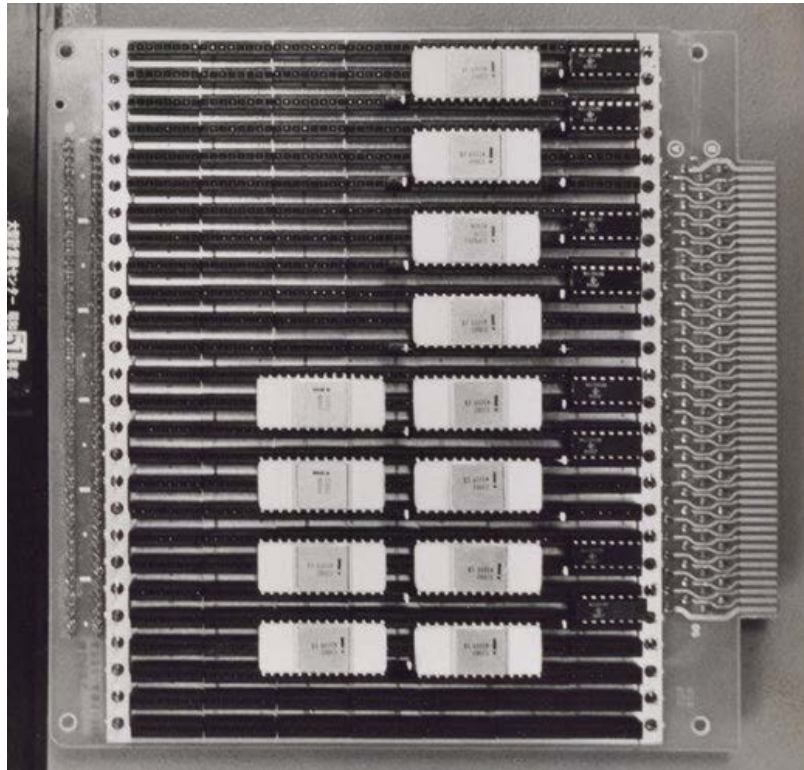
1976～1978年:PC-1Jr

- 修士論文のための作品(石田晴久研究室)
- 半導体技術でPC-1を作るとどのくらいの高速化が得られるか？
- Spec
 - Intel 3000シリーズ ビットスライスCPU
 - 全部手作り、
 - 配線20000本はワイヤラッピング
 - Clockは1MHz、
 - 0.5 VAX MIPS程度
 -
 -



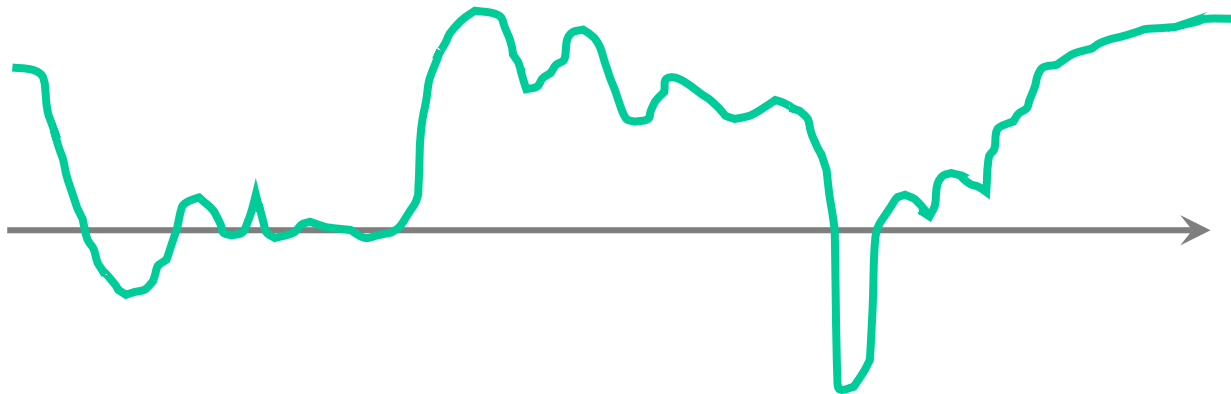
1976～1978年:PC-1Jr

- パネルはPC-1のものを模倣し、手作り
- バックプレーンにグランドプレーンがない
- PCBでも、GND, VCCが
細い線で配線



黒歴史1号

- 実は3分間くらいしか動かなかった
- 原因はGNDのノイズと信号波形の崩れ
 - GND、VCCの取り方が大切ということを学習
 - 手配線は大変なだけ、プリント基板使え
- 配線理論を博士に入ってから学んだ



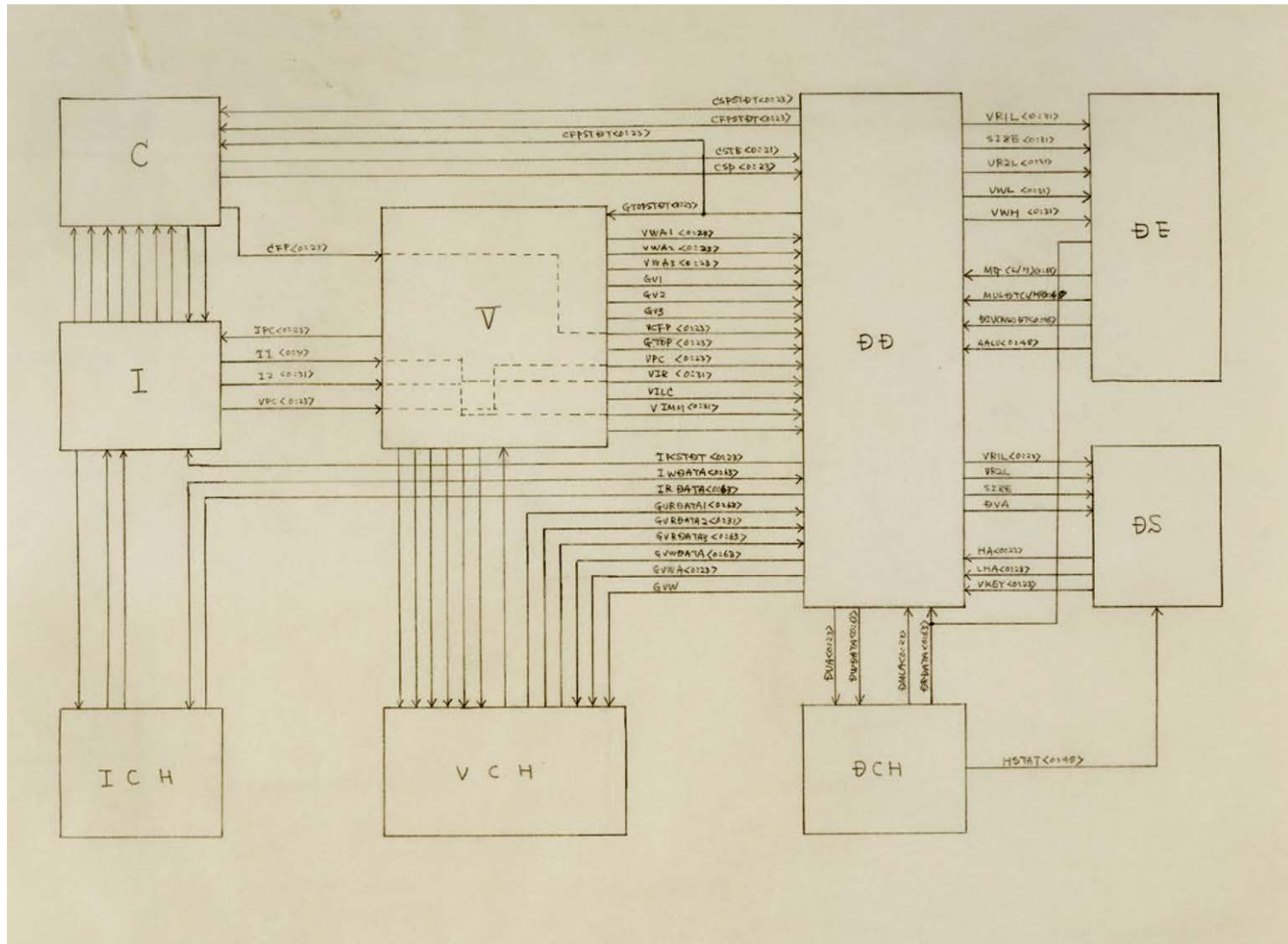
後藤研究室に転籍

- 修論審査会で、4月から後藤研でFLATS作れ、と
 - 昔は今では考えられない無茶があった
 - 私が計算機作りになった一番のきっかけ ⇒ 偶然
 - 高橋研に偶然入り、後藤研に強制的にコンバート
 - FLATSとは
 - Fortran, Lisp, Algol, Tuple and tree, Set 私が入った時
 - Formula, Lisp, Association, Tuple and Set プロジェクト終了時
- 一番困ったのが、開発目標が固定せず揺れ動いていたこと
(スパコン開発でも良く見られる困った現象)

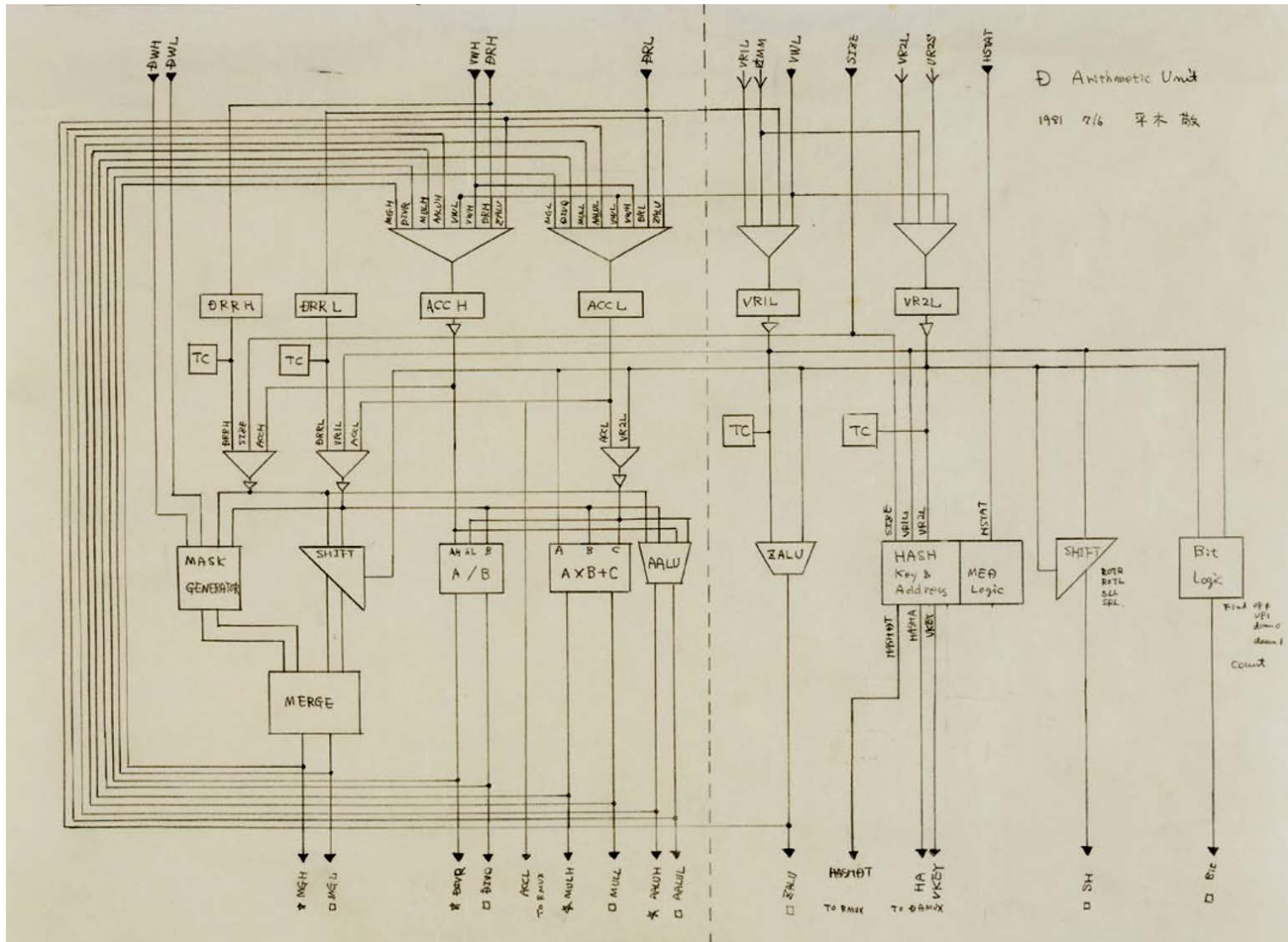
1978～1982年:FLATS

- 設計担当
 - 命令制御部 清水謙多郎（東京大学大学院農学生命科学研究科・教授）
 - 乗算器 後藤英一先生
 - 残り全部 平木 敬
- 回路素子 モトローラ ECL (CML)
 - 主にSSIを使用。全部のI/Oピンを外に出して、ボード間配線で論理回路を構築
 - キャッシュメモリだけは集積度の高いプリント基板作成
- 設計用CAD 清水さんが作成・とても便利だった
- 製作 三井造船・玉野製作所。
- クロック周波数 10 MHz

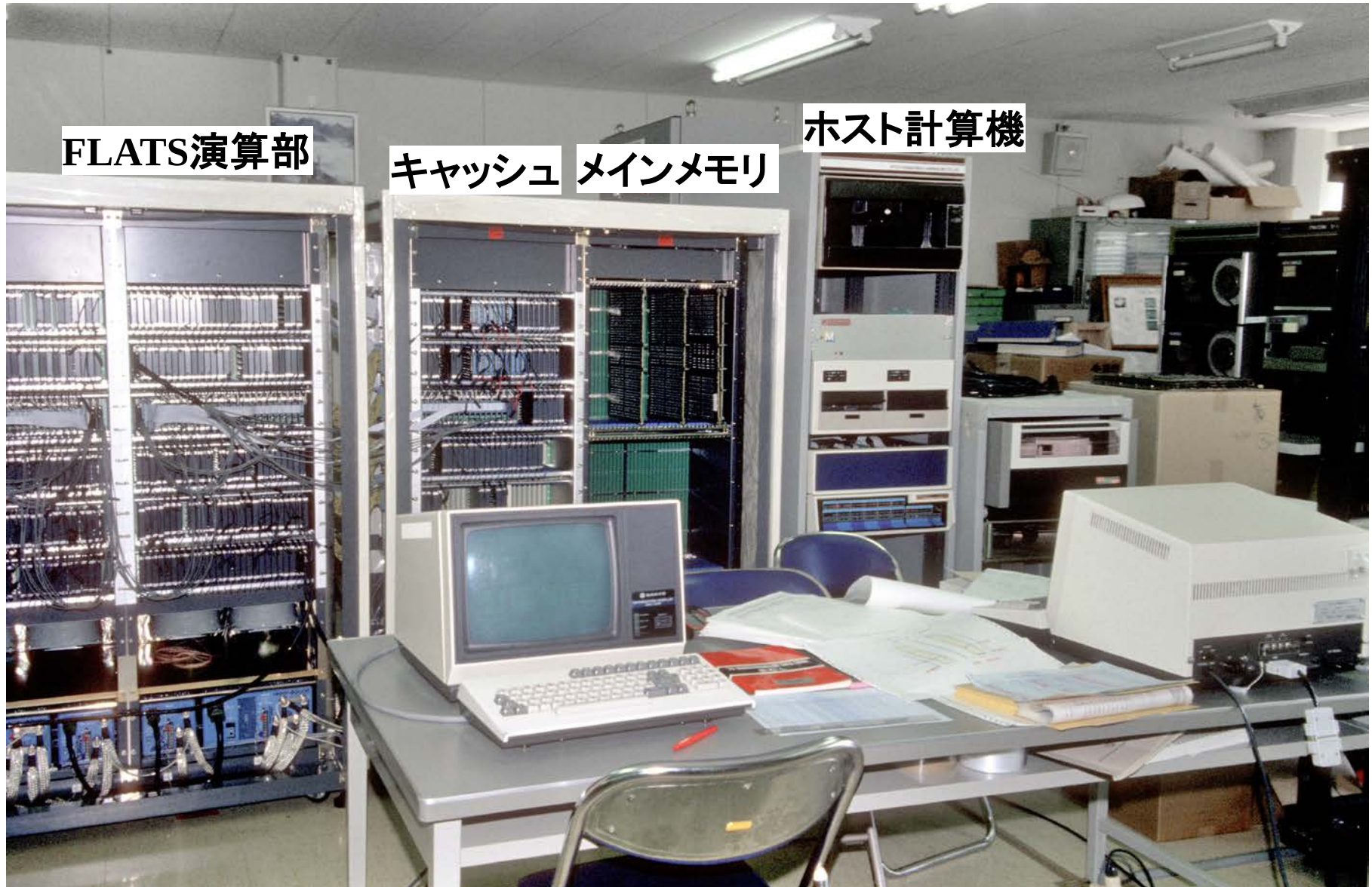
この図がトップレベルのブロック図



演算ユニット:ここから下はHDLで書いていました



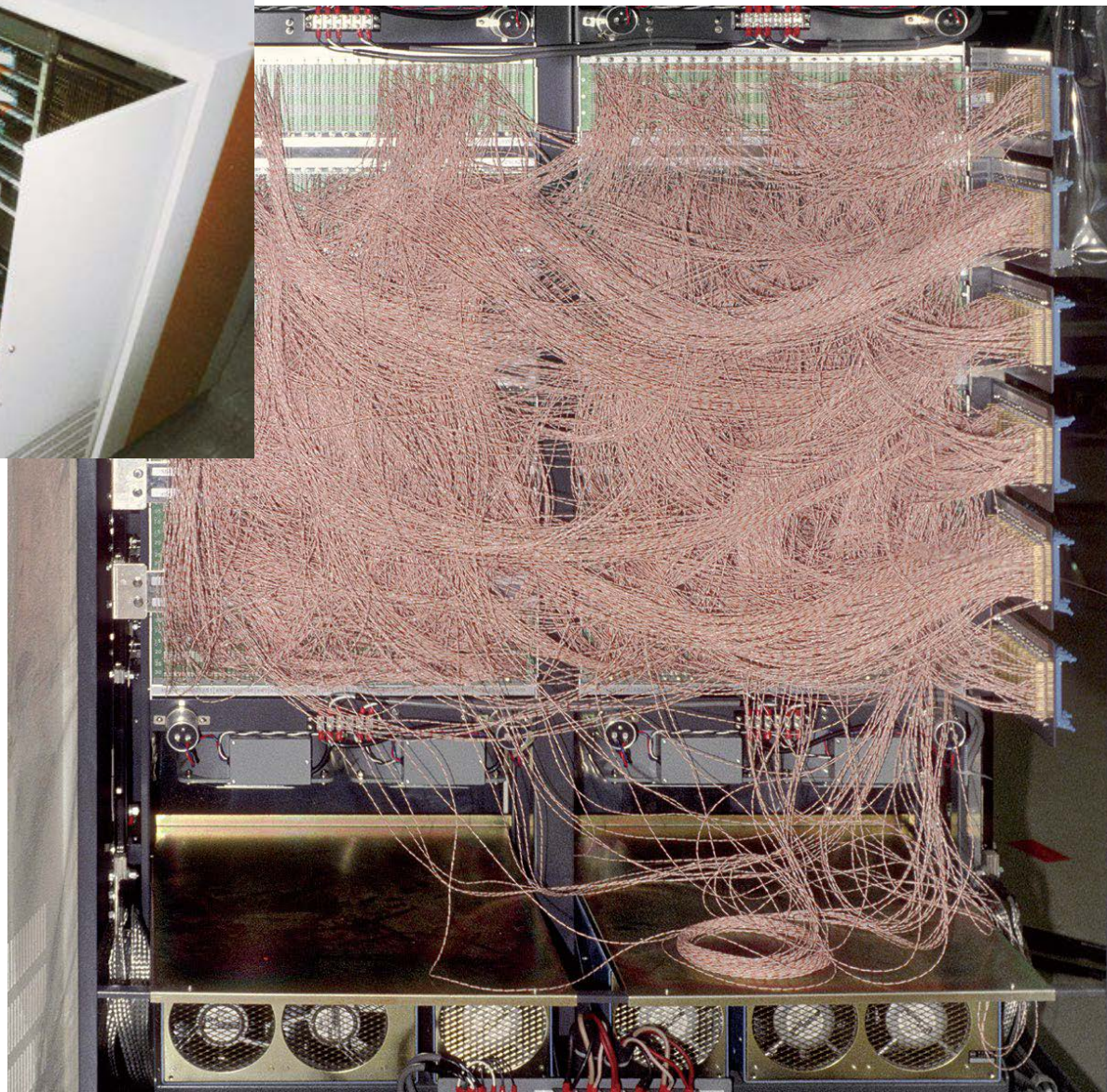
FLATS全景





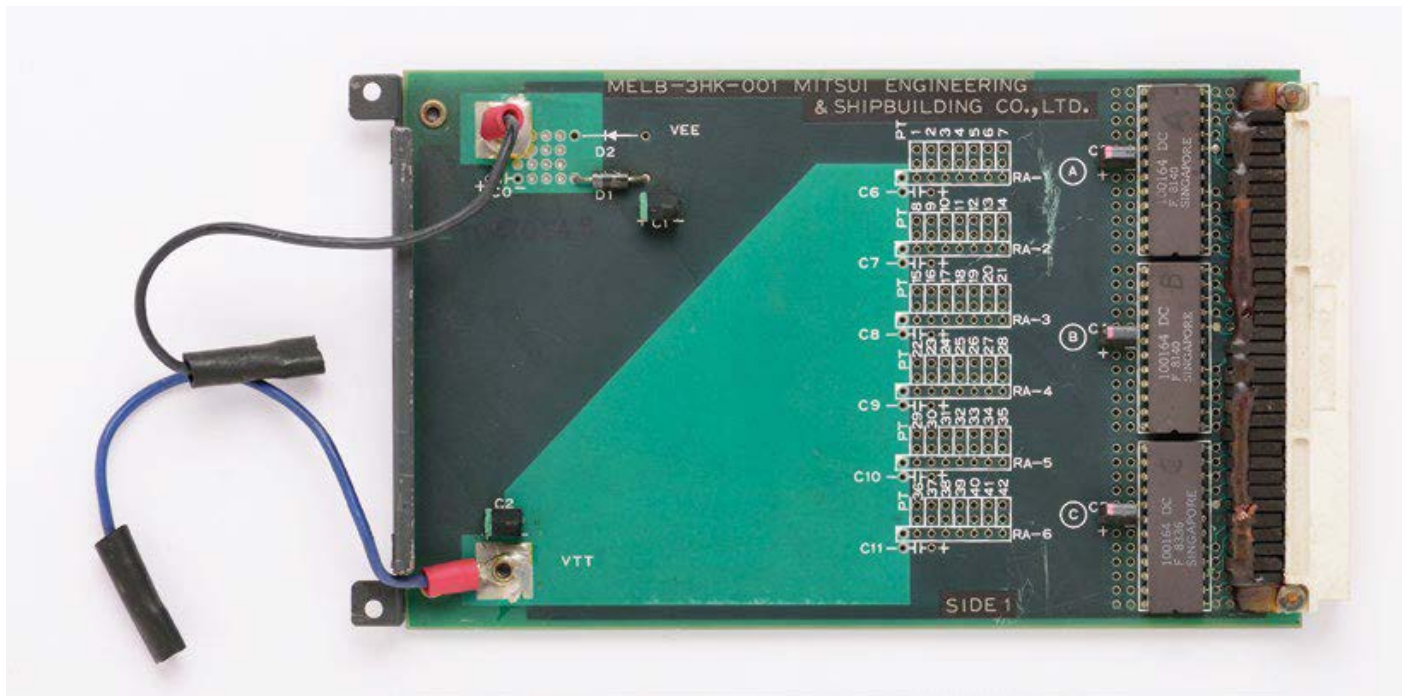
五角形に組んだ筐体

バックプレーンの配線



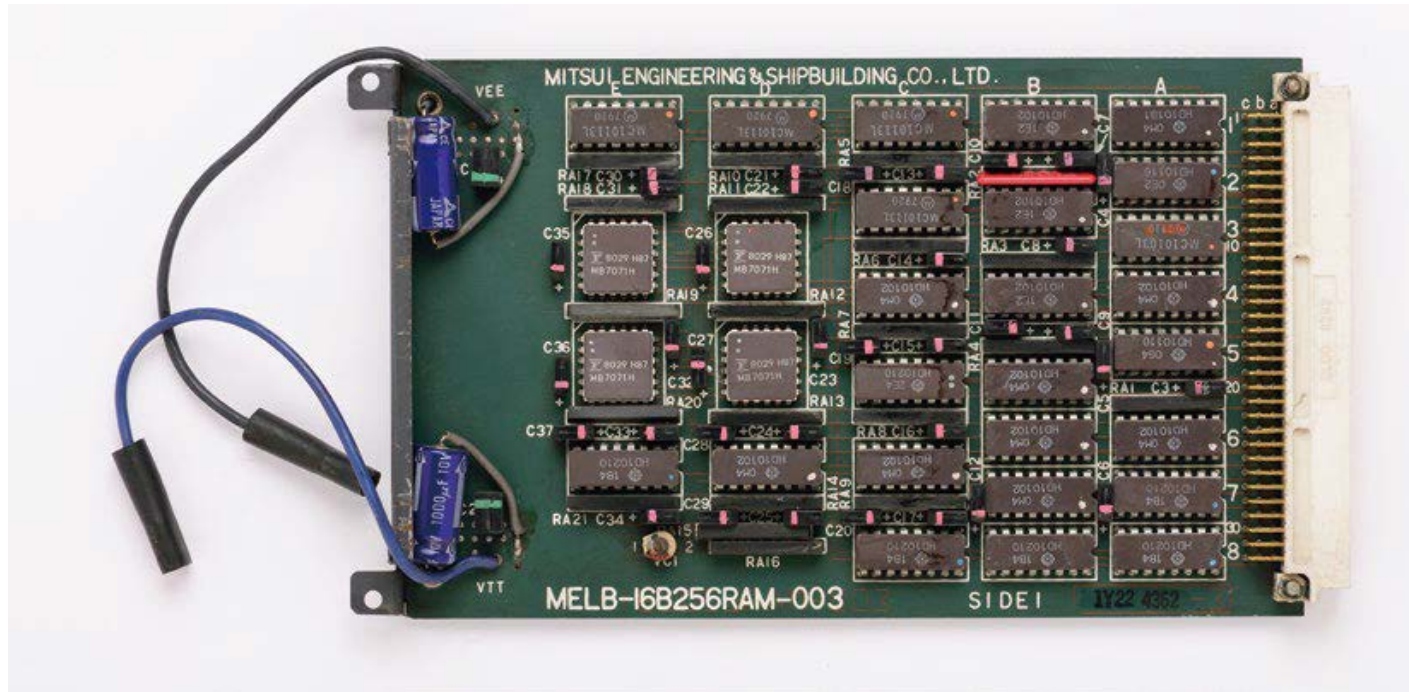
FLATSの論理回路ボード

- 1ボードにSSIが3個、ゲート12個
- 信号発振問題に悩まされる
- ECL回路は配線状態が悪いと発振する



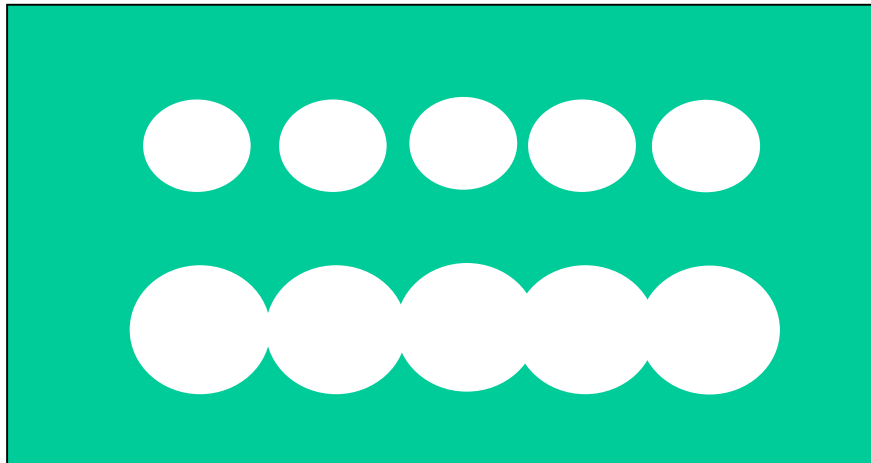
FLATSのキャッシュメモリボード

- 四角いのが富士通のECL SRAM
- SSIはヒット検出回路
- このボードは日立化成のマルチワイヤー。安定動作した

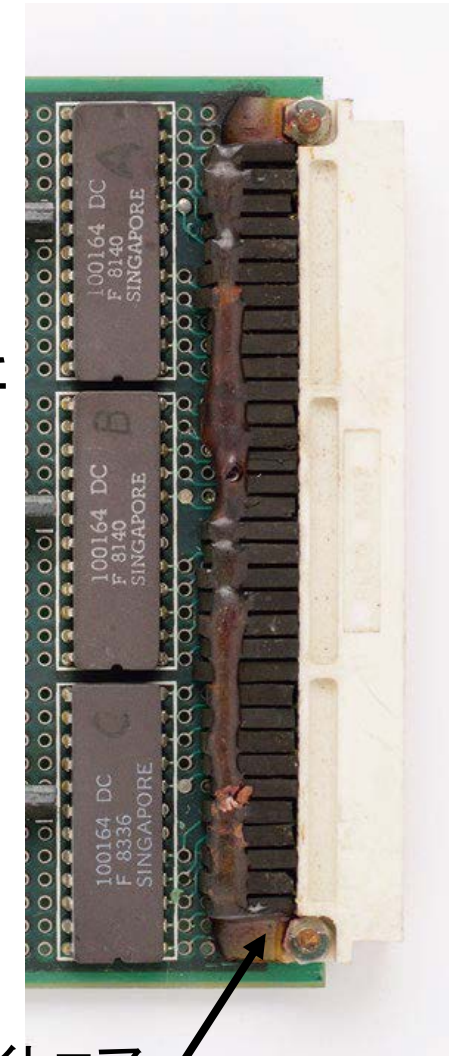


信号発信問題

- 原因は、プリント基板の内層
- クリアランスランドがつながってカット
- プリント基板のパターンを私は見なかった
- 深く反省、以後、パターンは必ず
詳細チェックを掛けることにした



正
誤



VMEコネクタ

フェライトコア
波形ダンプ用

黒歴史2号

- **プリント基板**のミスで3ヶ月遅れた
 - 原因はGNDのノイズと信号波形の崩れ
- LSI時代に**集積度の低いものは不適切**
 - クロックが上がらない原因(Mシリーズの1/10)
- Lisp実行でM380とほぼ同速度(クロックあたり10倍)
- 長時間安定動作した(当たり前だけど)
- 最後まで、**目的意識が定まらなかった**
- **QRP**(Questionable Research Practice)に悩まされた
 - ⇒ 私が研究倫理教育担当になった原動力か？

知られざるコンピュータ作りに

情報処理：特集：知られざる計算機 2002年2月 (Vol.43 No.2)

日立のHITAC 2010 (高橋 茂・浦城恒雄)

VLIWコンピュータのパイオニア QA-1 (富田眞治)

神戸大LISPマシン, PROLOGマシン (金田悠紀夫・瀧 和男・和田耕一・田村直之)

筑波大 PACS (星野 力)

通研 ELIS (日比野靖)

EVLISマシン (安井 裕)

数式処理計算機 FLATS (相馬 嵩)

FACOM α (林 弘)

SIGMA-1: データフロースーパーコンピュータ (平木 敬)

並列推論マシン PIE (田中英彦)

電総研 EM-4 (坂井修一)

SIMD型並列計算機 SM-1 (湯浅太一)

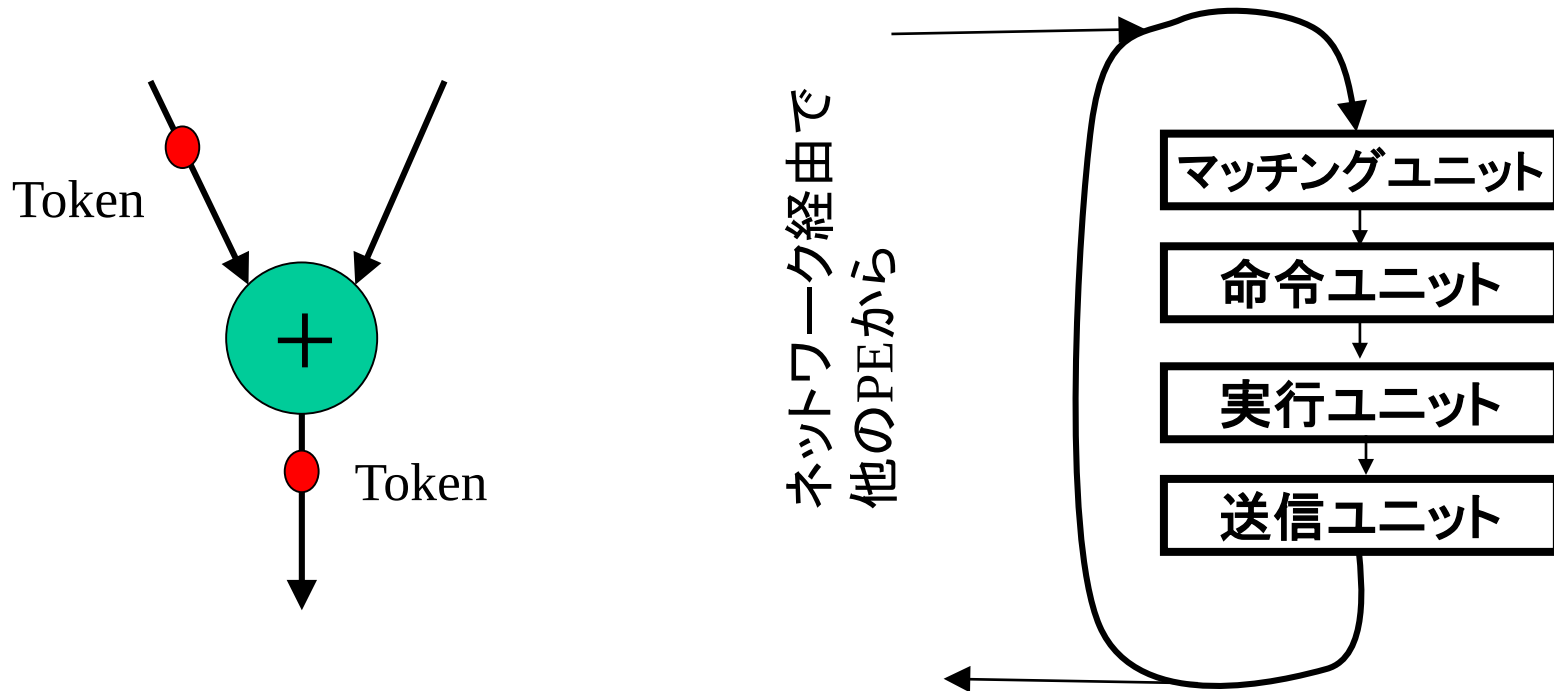
(私が創った計算機のほぼすべてがこのカテゴリーなのが悲しい)

1982～1988年：SIGMA-1作り

- 世界初、世界一性能のデータフロースーパーコンピュータ
- Spec
 - 富士通のCMOS ASIC使用、128プロセッサ、128同期するメモリ
 - Clockは10MHz、 110Mflopsを実測
- 体制
 - 製造は業者に出したが、設計・製作・デバッグは一人作業
 - 取りまとめは弓場敏嗣さん、島田俊夫さんが担当しました。
 - ソフトウェアは佐藤三久さん、関口智嗣さん他が担当しました
 - プロトタイプメモリ、浮動小数点演算器は西田さんが担当しました。
 - 予算は有名な「通産省大型プロジェクト・科学技術用高速計算システム」

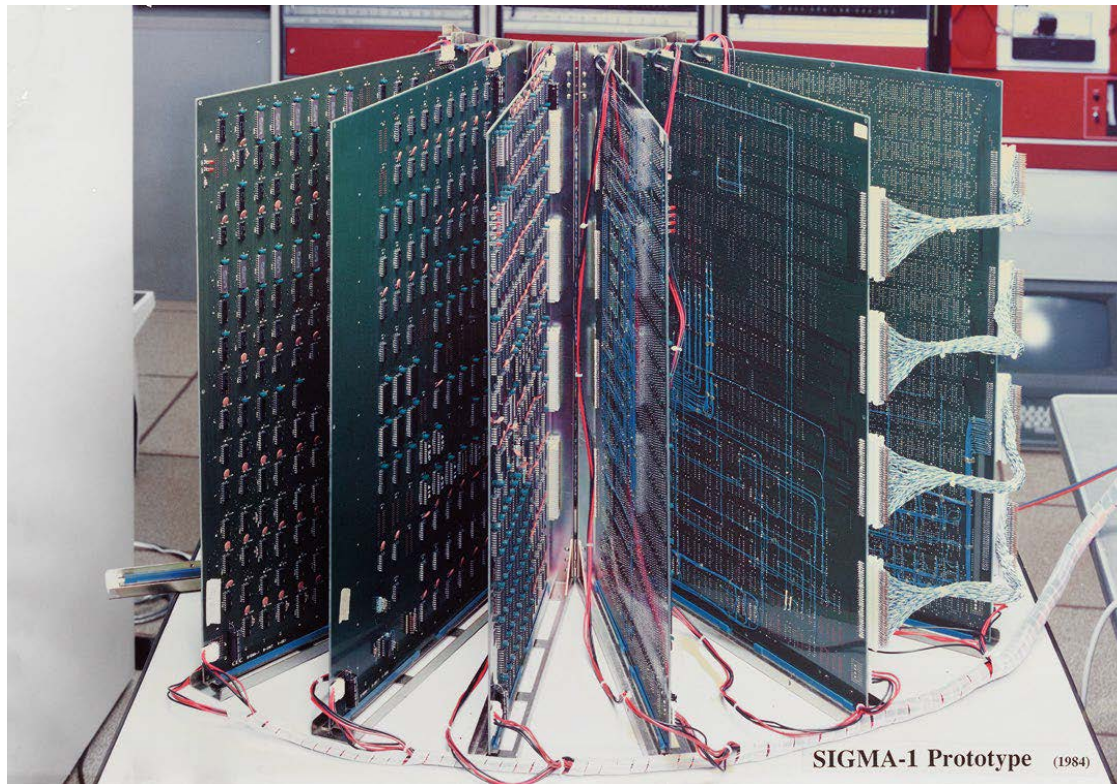
データフローコンピュータとは？

- 代表的非フォンノイマン計算機
- 命令への全データ到着で命令実行を開始
- プログラムをData Flow Graphとして表現する
(これを語り出すと2時間では終わらない)



1984年：SIGMA-1プロトタイプ

- データフローマシンは誰も作ったことが無いので、とりあえず動くものを作った
- 設計半年、製作3ヶ月(マルチワイヤー)、デバグ3ヶ月



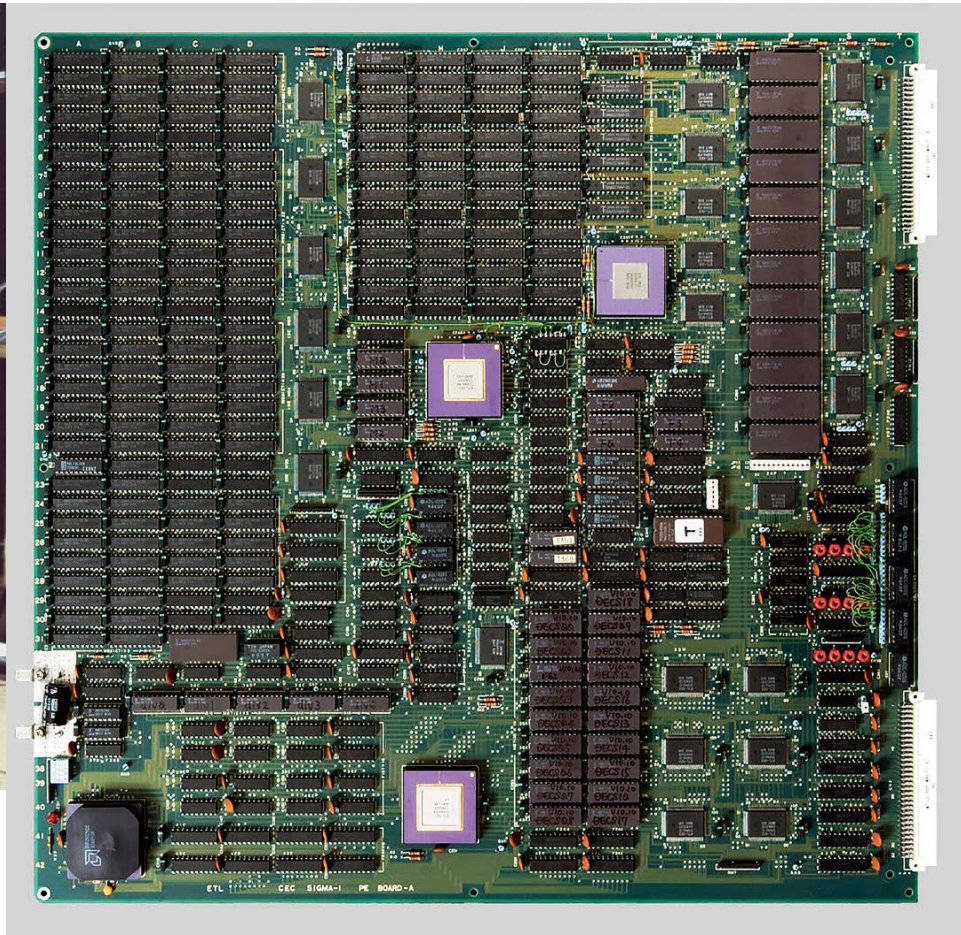
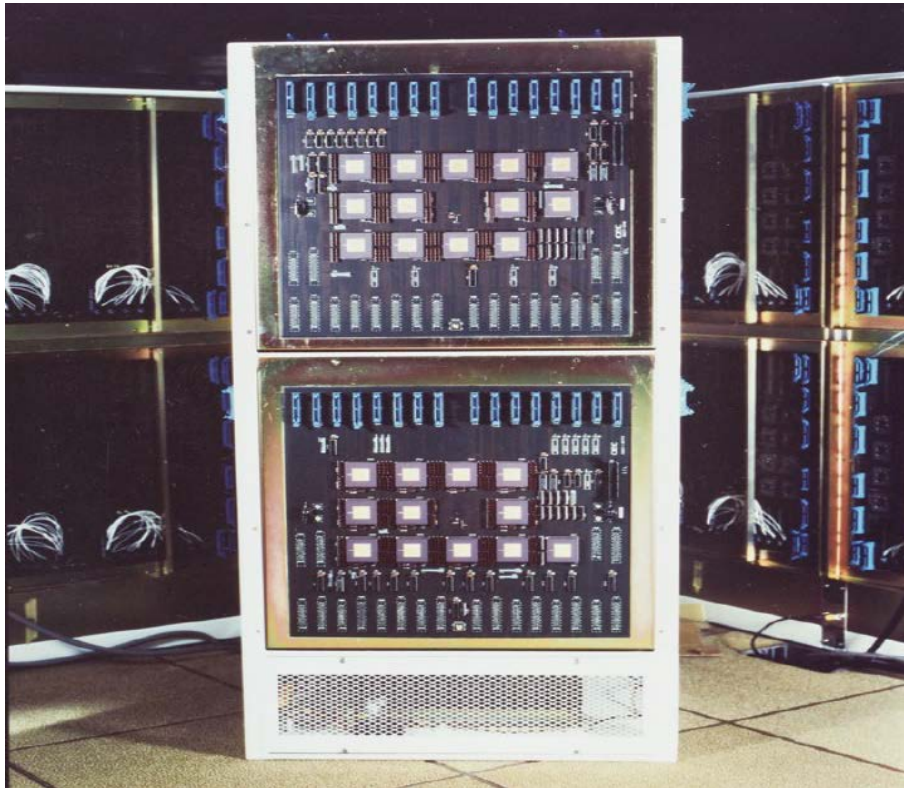
パッチワイヤの多さが少し恥ずかしい

1988年: SIGMA-1フル構成稼働

- プロセッサ 128台、構造メモリ 128台構成
- ネットワーク⇒クロスバーとオメガ網の2段構成
- Hilo CAD systemで設計(設計は楽だった)
- 富士通ASICでPE/SEを構成



SIGMA-1写真帳(話すと長くなるので)



SIGMA-1を作って学んだこと

- 高並列システムを作るのは大変
 - 1台が超安定にうごいても、128台だとぼろぼろ
 - 1本のパッチ線でも、128台あてる必要がある
 - ツール入れると時間関係が変わりデバグが難しい
 - 並列言語の重要性が良く判った

今日では常識になっている
並列コンピュータの難しさを初体験

SIGMA-1を作って学んだこと2

- 周囲の意見は参考にならないこと
 - 何度も頭おかしいと(メーカーの主要な技術者に)いわれた
 - 何の役にたつのかとも、常に言われ続けた
 - きっと動かないと不吉な予言を何度も聞かされた

1988～1990年：IBMでHybrid dataflow作り

- IBM経営危機でプロジェクトがキャンセルされて完成しなかった
- IBM Watson Research Center
- 7名のプロジェクト
- Spec
 - BossはBob Iannucci博士
 - DataflowとControl FlowのHybridアーキテクチャ
 - システムデザインと論理設計を担当
 - 多くのプロジェクトがキャンセルされた

作ってないから、黒歴史ではない

1992～2000年: JUMP-1作り

- 科研費重点領域研究: 代表者 田中英彦
- Spec
 - 64プロセッサ、SUNのSuperSPARCチップ使用
 - 分散共有メモリ、RDTネットワーク
 - 足回り、ネットワーク周りで2種のASIC作成(■、日立)
 - Clockは200MHz
 - Major player 東大、慶応大学、京都大学



(写真は五島正裕氏提供)



黒歴史3号

- ともかく完成が遅れに遅れた
 - 戦犯1号は■■■■社 下流設計が2年間進展なし
 - 設計が各大学に分散し工程管理が困難だった
 - プロジェクト予算が途中で切れて、後は各拠点の好意
注: 何かTACに似ている状況と言うと言い過ぎか

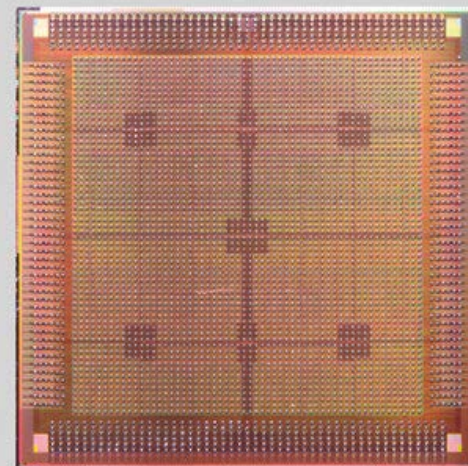
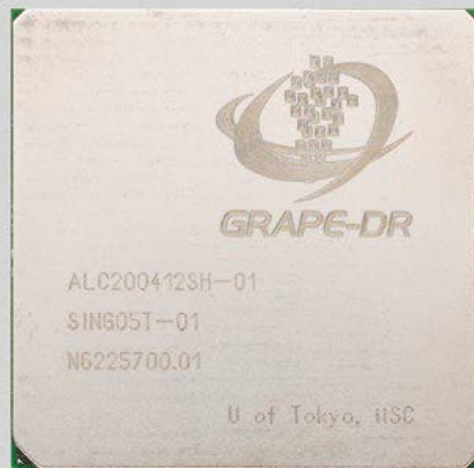
これと、次のGRAPE-DRのLSI
作成で、日本の半導体は大問
題を抱えていることを確信した。

2004～2010年:GRAPE-DR作り

- 科学技術振興調整費
 - プロセッサとインターネットの両方を高速化
- お金が無かったので、牧野さんと組んで大プロジェクトを提案した
- 超低消費電力・高性能が最初からの売り
 - これから、私と京、ポスト京との関係が始まった
 - 当時はGreen500もなく、性能・電力比の重要性は認められていなかった(特に日本)

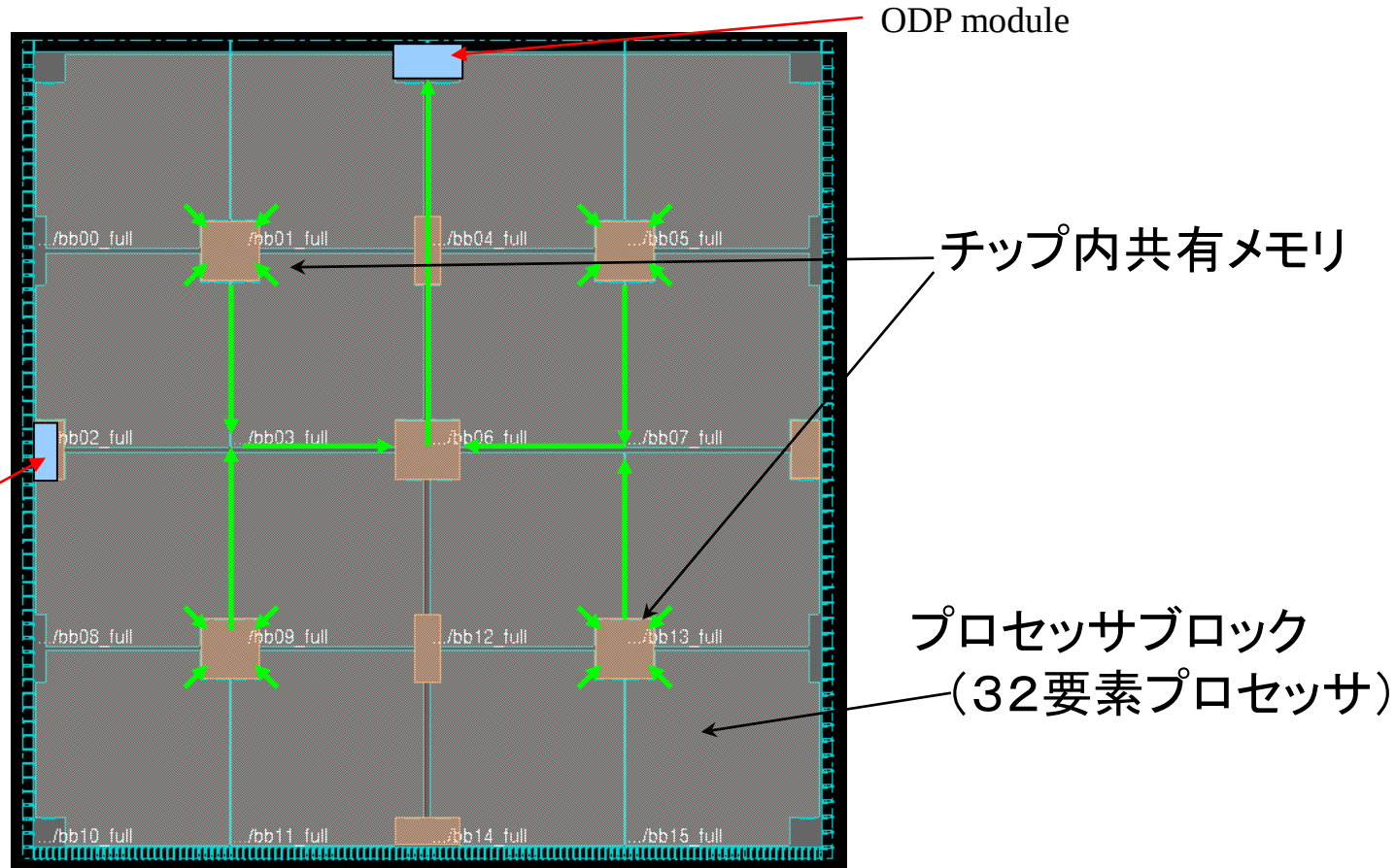
GRAPE-DRのプロセッサチップ

- SIMD演算プロセッサチップを作成
- Spec
 - 128ノード、512カスタムプロセッサ
 - 90nm CMOS (TSMC)
 - 超低消費電力
 - Clockは500MHz



プロセッサチップのフロアプラン

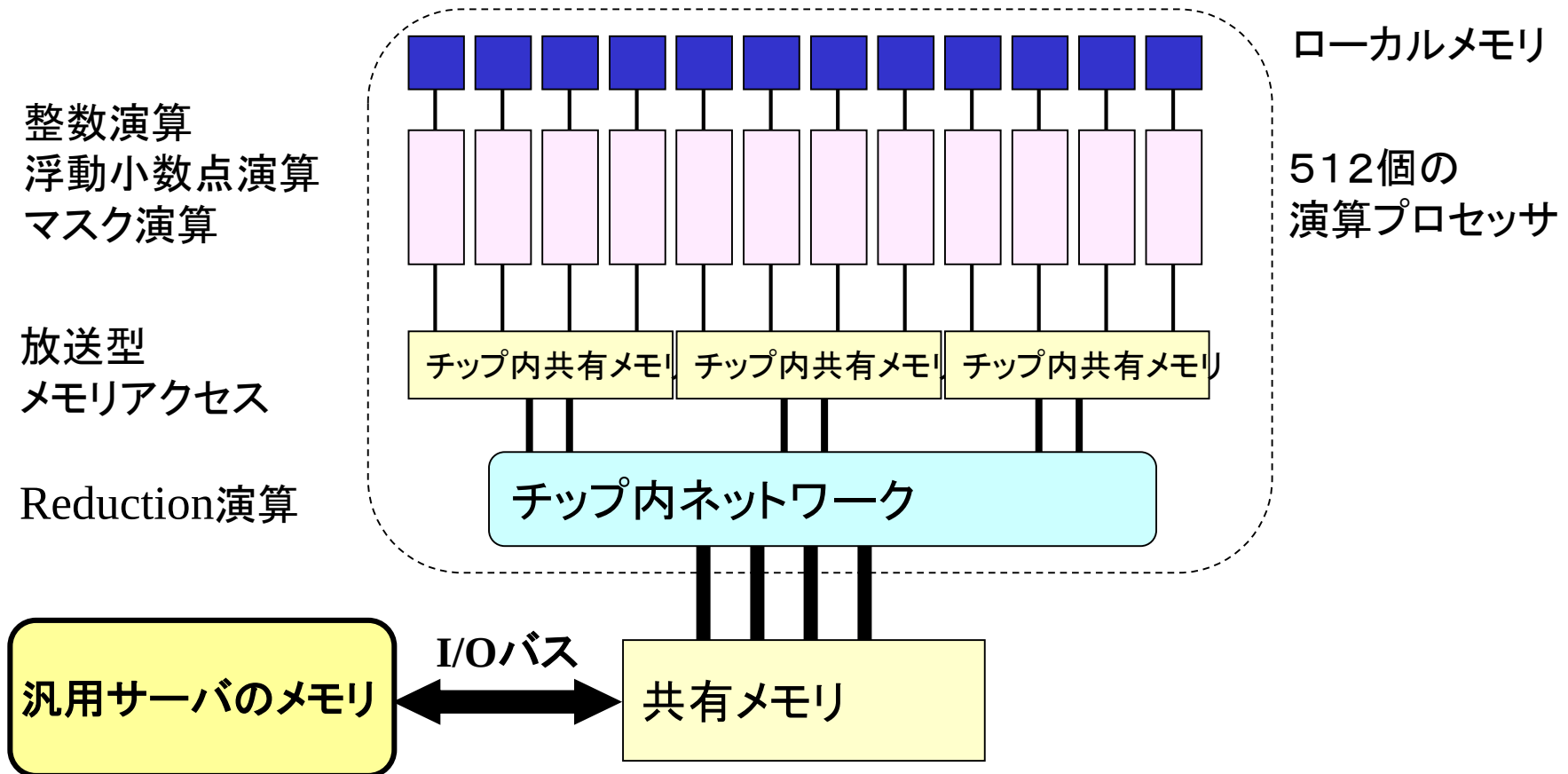
(Grape_weekly_meeting_20050815.ppt)



- 90nm CMOS
- 18mm x 18mm
- 約4億トランジスタ
- BGA実装

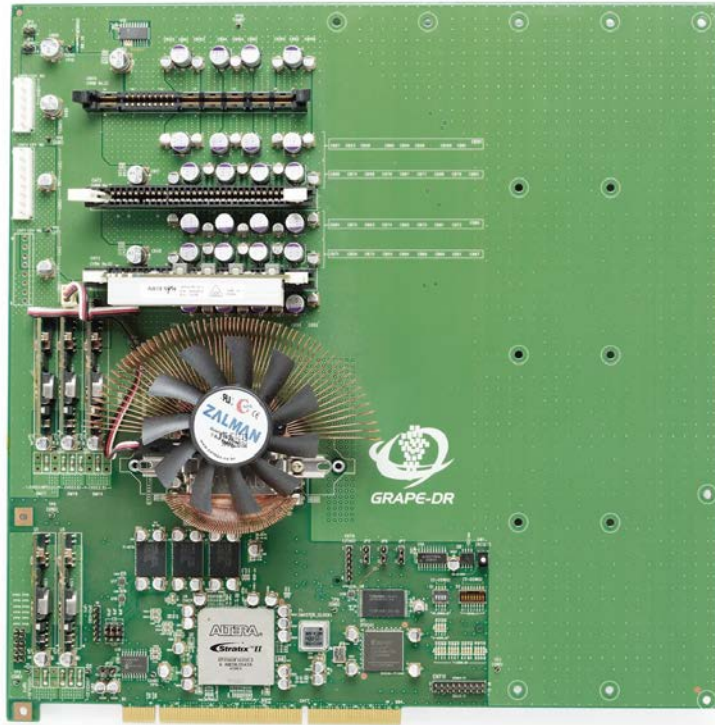
GRAPE-DRプロセッサ

- 超並列プロセッサ
- 512組の64ビット演算器を共有メモリと、ネットワークで接続
 - 演算器間の相互接続の排除
 - Reduction専用ネットワークの埋め込み



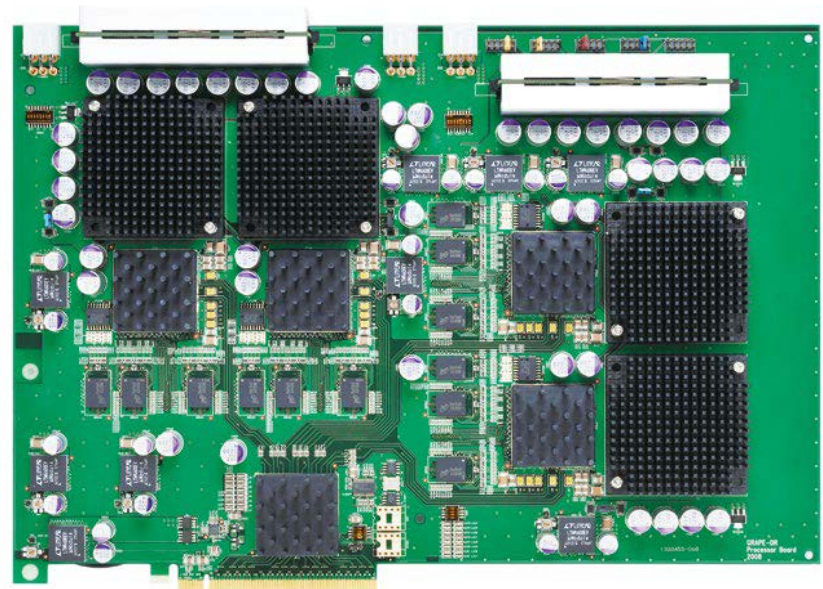
2004～2010年:GRAPE-DR作り

- PEボードは何枚も作った



最初の1PEボード
PCI-Xバス用

PCIEの1PEボード



4PEボード

おまけ、 ちょっただけ自慢

GRAPE-DR

128サーバ、1024チップのクラスタ



GRAPE-DRシステム

なぜコンピュータが丸裸なの？

1. 牧野さんの趣味

- 多くのGRAPEシリーズは裸だ
- 確かに便利、冷却も良い

2. 本当にお金が無かった、箱が買えない

- 為替差損、 105円⇒115円

Green 500

1. 初の1Gflos/W 越え

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
<u>1</u>	1684.20	IBM Thomas J. Watson Research Center	NNSA/SC Blue Gene/Q Prototype	38.80
<u>2+</u>	1448.03	National Astronomical Observatory of Japan	GRAPE-DR accelerator Cluster, Infiniband	24.59
<u>2</u>	958.35	GSIC Center, Tokyo Institute of Technology	HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows	1243.80
<u>3</u>	933.06	NCSA	Hybrid Cluster Core i3 2.93Ghz Dual Core, NVIDIA C2050, Infiniband	36.00
<u>4</u>	828.67	RIKEN Advanced Institute for Computational Science	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect	57.96
<u>5</u>	773.38	Universitaet Wuppertal	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
<u>5</u>	773.38	Universitaet Regensburg	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
<u>5</u>	773.38	Forschungszentrum Juelich (FZJ)	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
<u>8</u>	740.78	Universitaet Frankfurt	Supermicro Cluster, QC Opteron 2.1 GHz, ATI Radeon GPU, Infiniband	385.00
<u>9</u>	677.12	Georgia Institute of Technology	HP ProLiant SL390s G7 Xeon 6C X5660 2.8Ghz, nVidia Fermi, Infiniband QDR	94.40
<u>10</u>	636.36	National Institute for Environmental Studies	GOSAT Research Computation Facility, nvidia	117.15

* Performance data obtained from publicly available sources including TOP500

いきなり話は変わります

高速インターネット研究

2001年～:いきなりインターネット研究

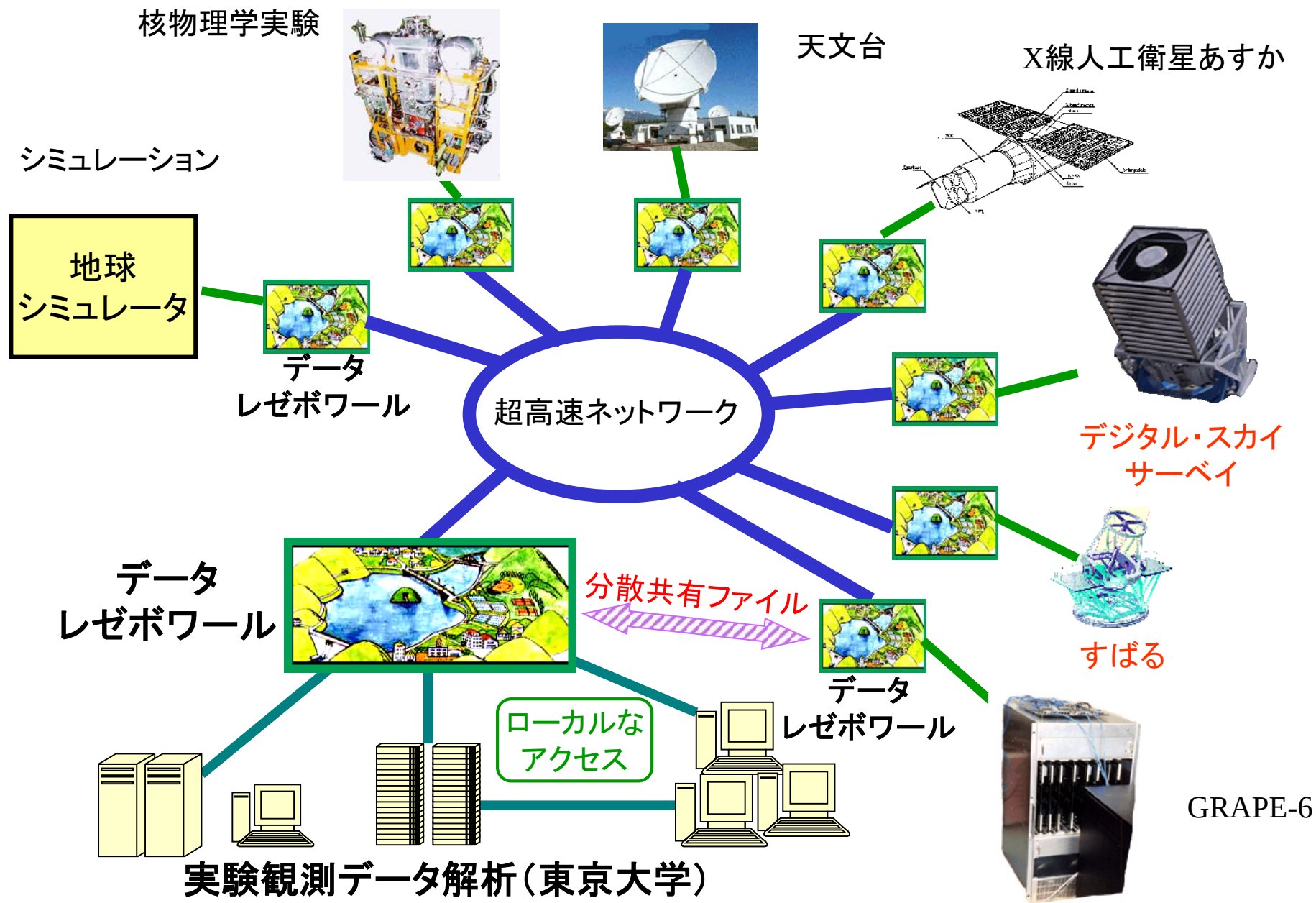
- 新分野に入った理由 ⇒ お金が無かった。。。。
- アーキテクチャ研究冬の時代
- SINETの10G幹線が完成したが殆ど使われてなかった



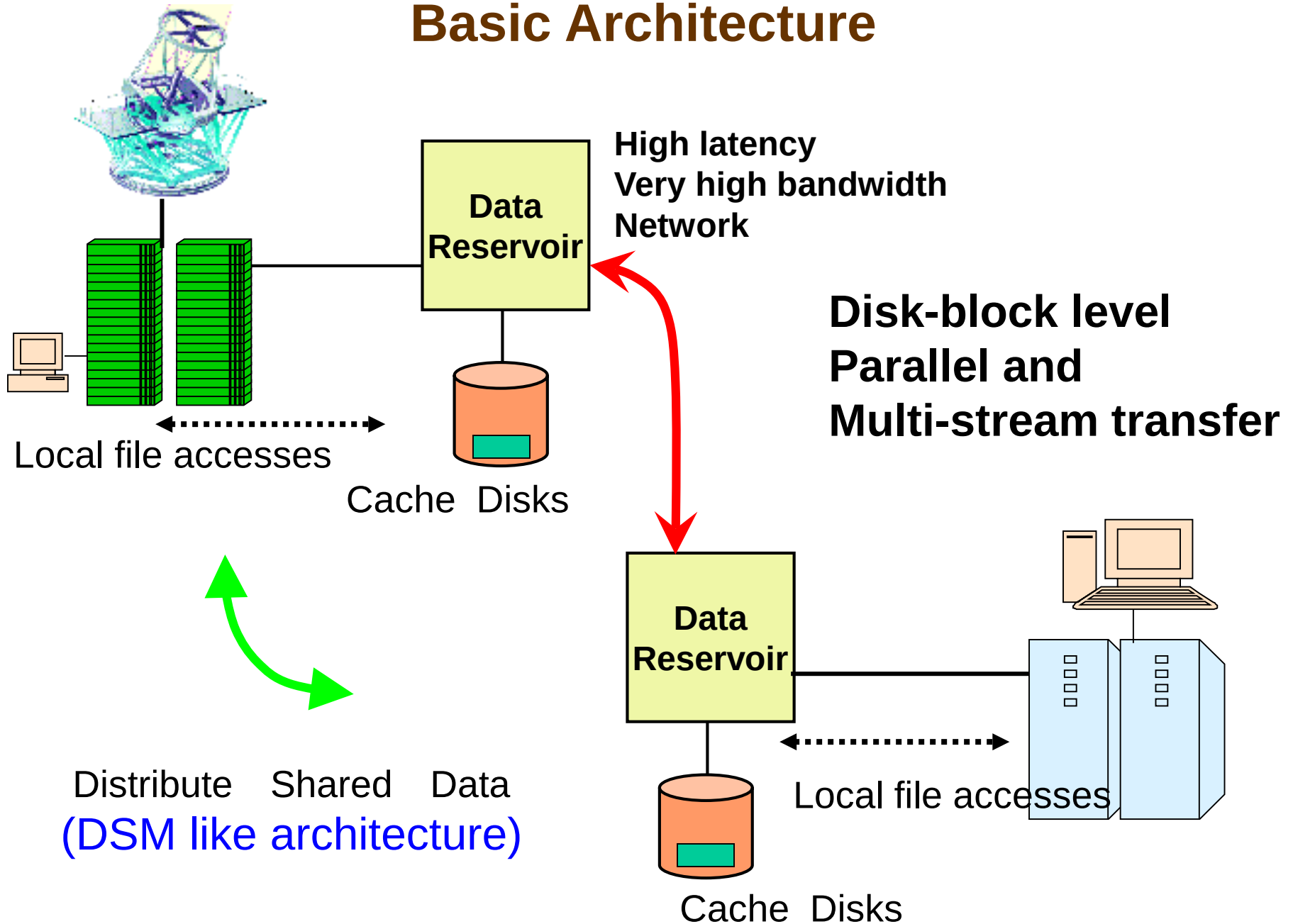
チャンス到来。ハードウェア技術を生かして高速化しよう！

本当にこれだけの理由でネットワークやさんになりました

超高速ネットワークを用いた科学技術実験観測データ解析



Basic Architecture

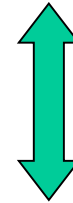
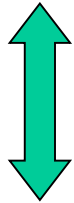


設置されたデータレゼボワール

国立天文台

JAXA

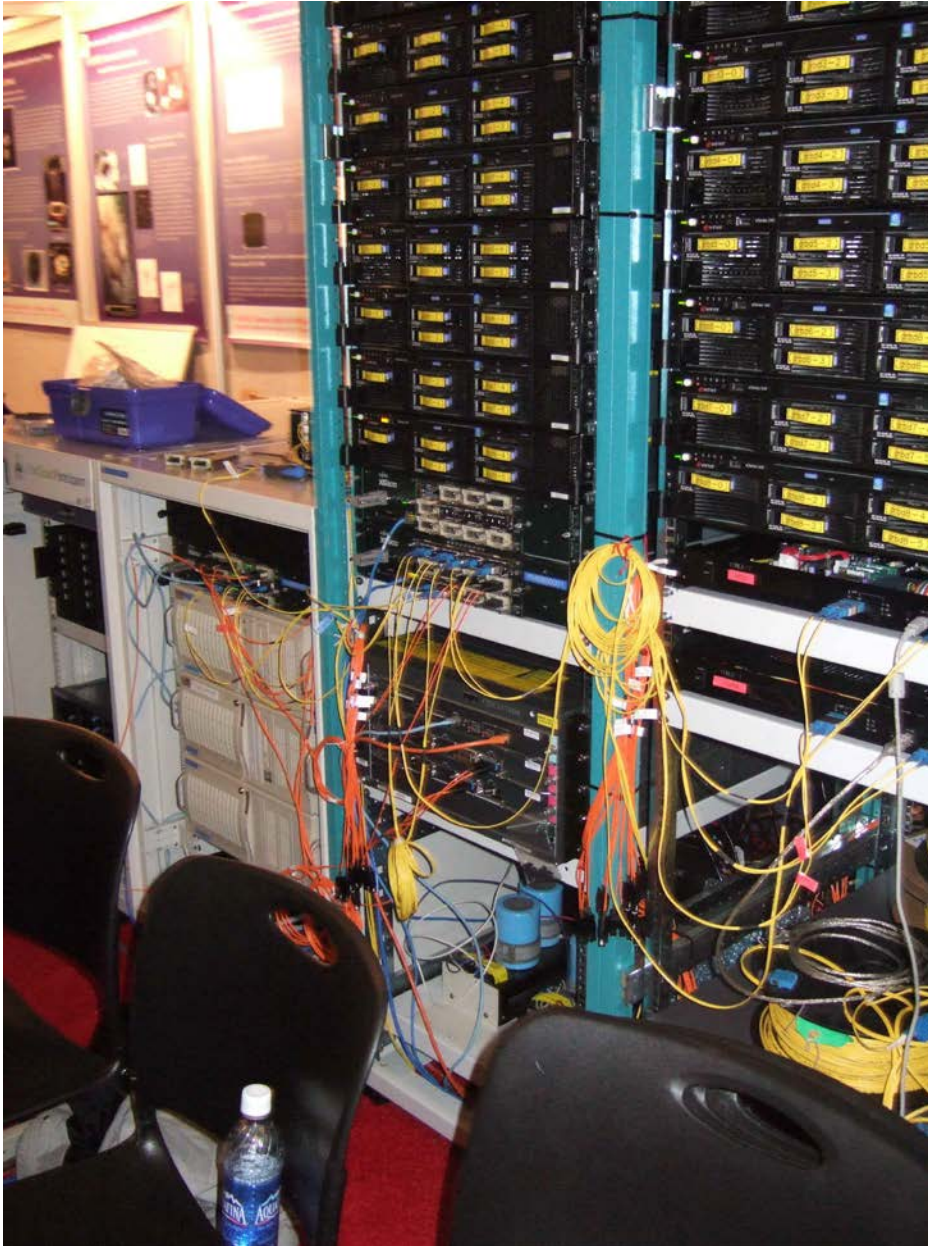
東大プレハ
ブ実験室



東京大学

東京大学

Devices at SC|05



Devices at Pacific Northwest
Gigapop



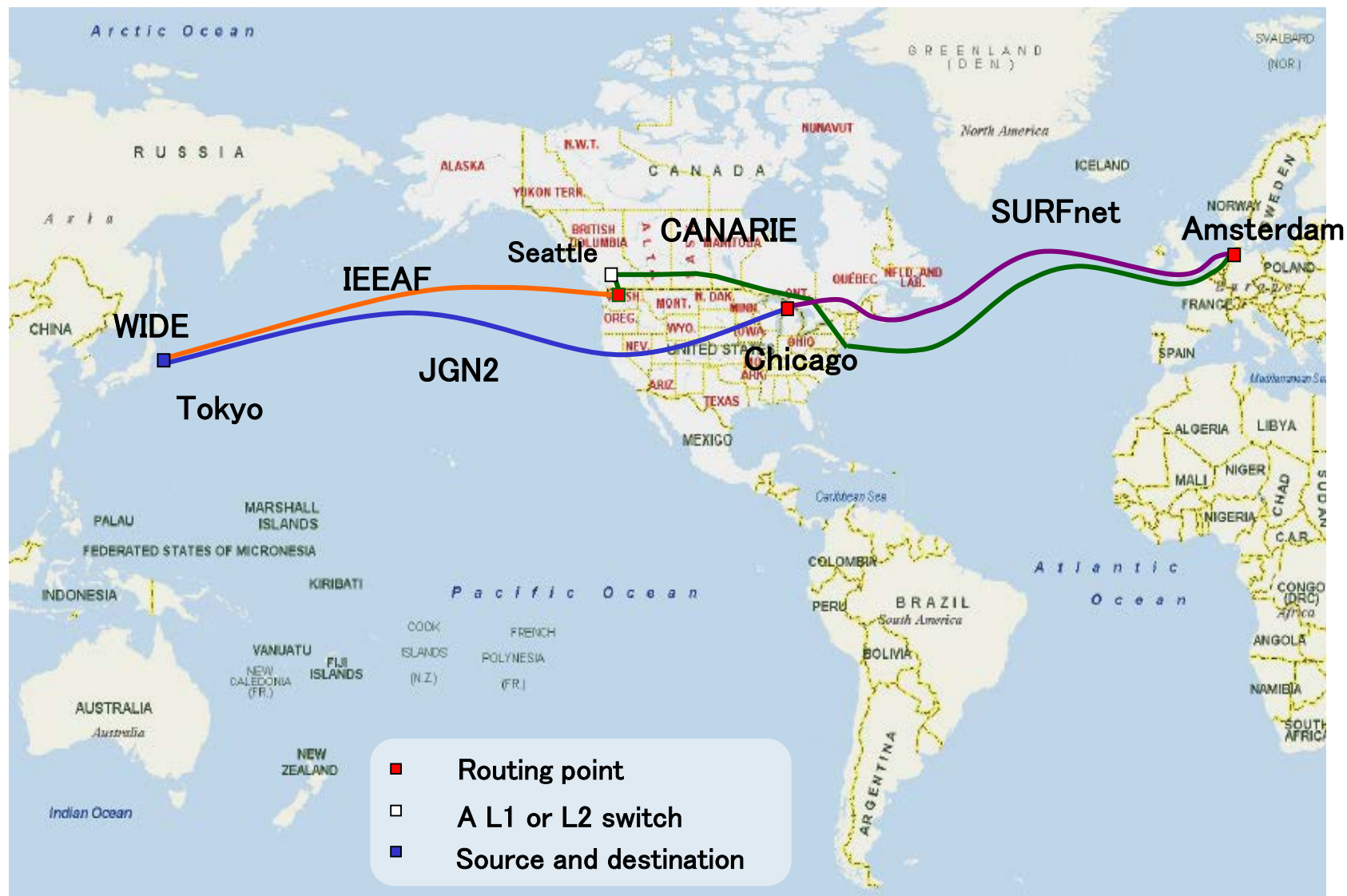
LSRへの挑戦について

- ・ データレゼボワールプロジェクト(2001年から)
 - 遠距離研究機関間の研究データ共有
 - 現在、天文台、JAXA、地球シミュレータセンタで利用
- ・ 海外での試験使用(2002) → 低いTCP性能に悩む
- ・ システム規模の縮小が必須となった

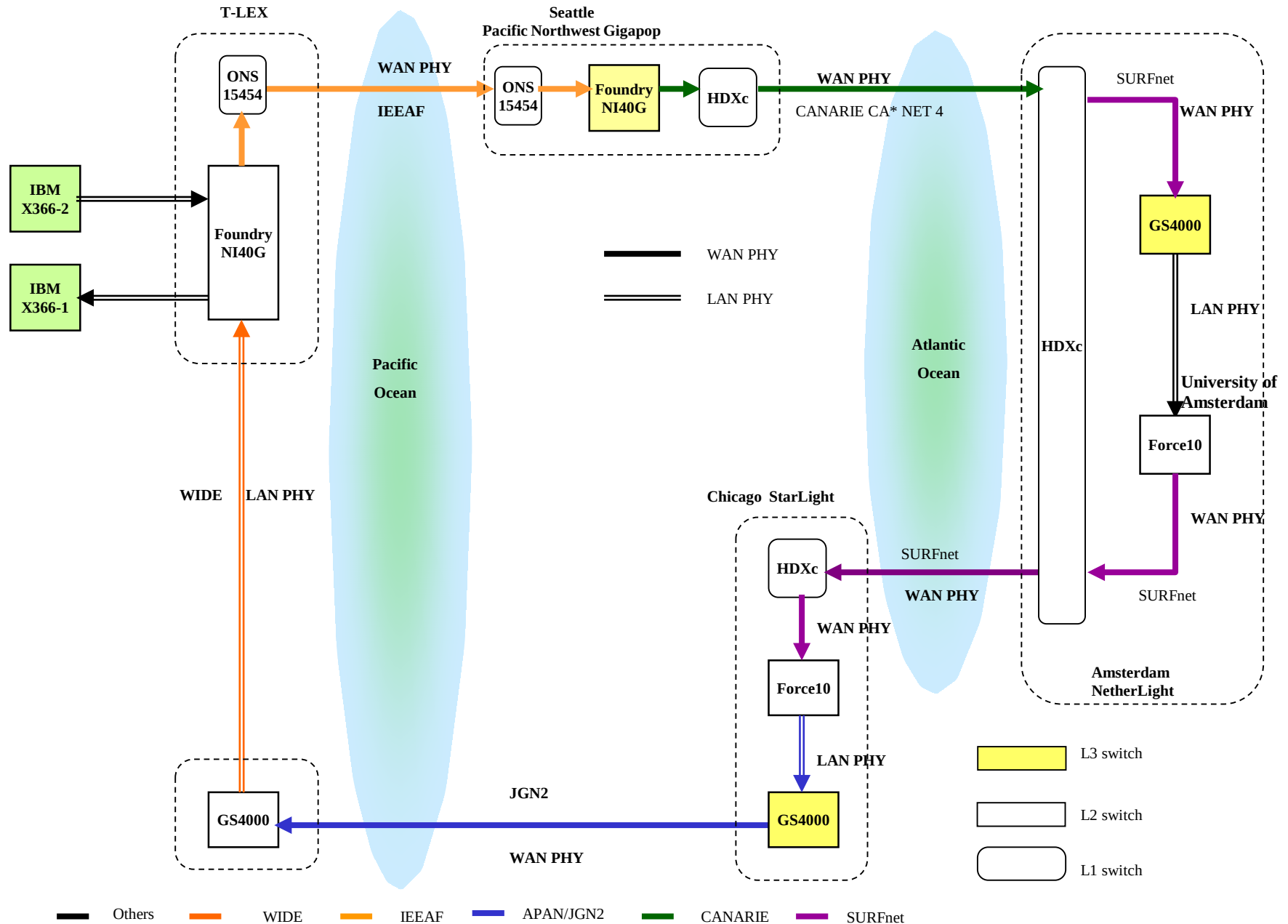


- ・ 遠距離TCP通信高速化への研究開発取り組み

2006年2月実験 ネットワーク経路



2006年2月実験 ネットワーク構成



5世代のData Reservoir

長距離のディスク間データ転送システム

- 1st Generation – 26 servers, 26 disks for 500Mbps
- 2nd Generation – 16 servers, 64s for 10Gbps
- 3rd Generation – 8 servers, 32 disks for 10Gbps
- 3rd Generation – 1 server, 32 disks for 10Gbps
- **5th Generation – 1 CPU PC, 6 SSDs for 10Gbps**



**5th generation
2009**



1st generation
2001



2nd generation
2003



3rd generation
2005



4th generation
2006

ちょっと自慢2

10回更新したInternet2 Land Speed Record



私たちはやりすぎました。LSRが無くなってしまいました
(スパコンでもLinpackは危機に瀕しています)

今取り組んでいること

- セレンディピターの構築
 - 血液細胞やミドリムシの細胞群から希少な細胞を探索する
 - 多数の超高速センサからの情報を機械学習で弁別
 - All IP architecture を基本アーキテクチャにする
 - 理学系、工学系、千葉大学、名古屋大学、京都大学と共同研究

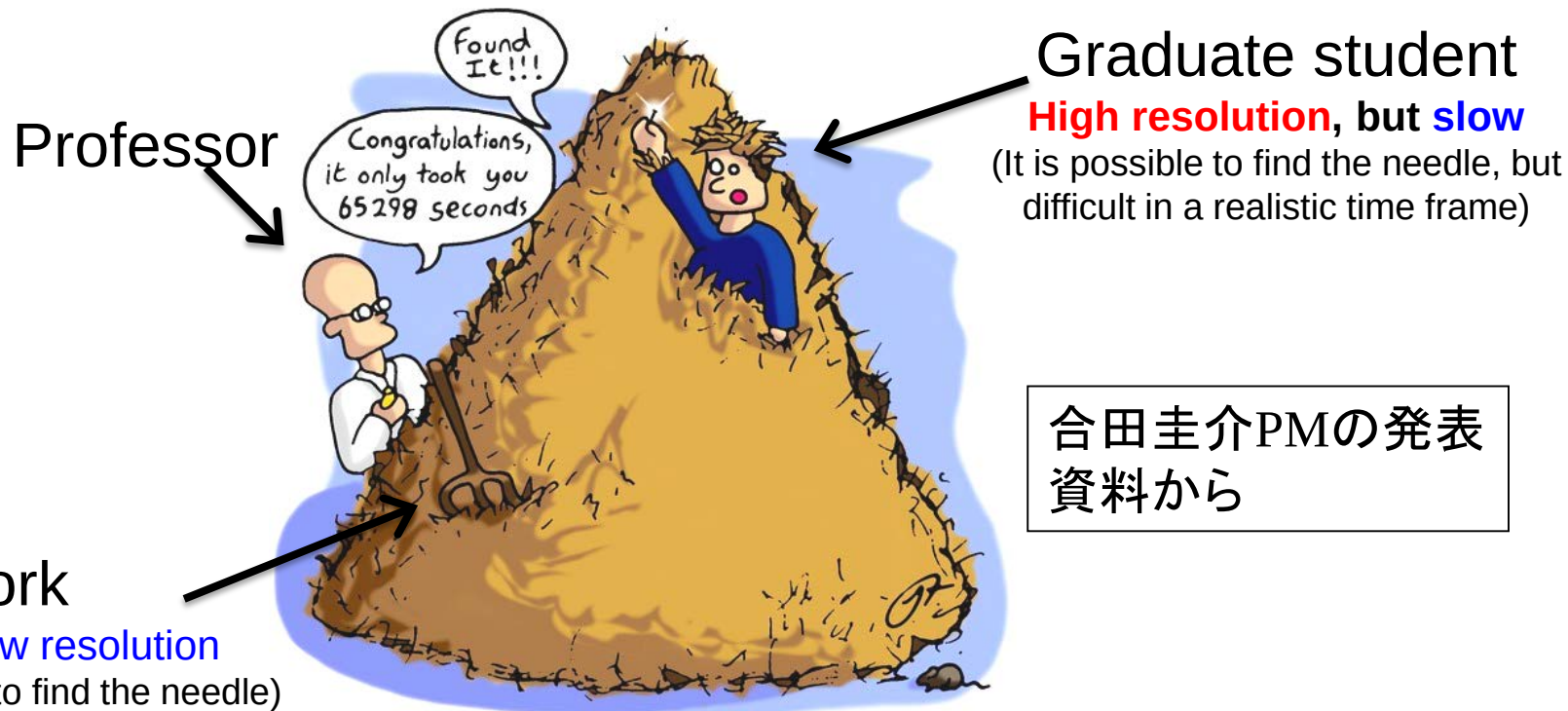
.....

私の2つの研究の流れの合流

Background: Why is serendipity accidental?

Searching is one of the most important behavioral activities for animals and humans (e.g., searching for food, a marriage partner, work)

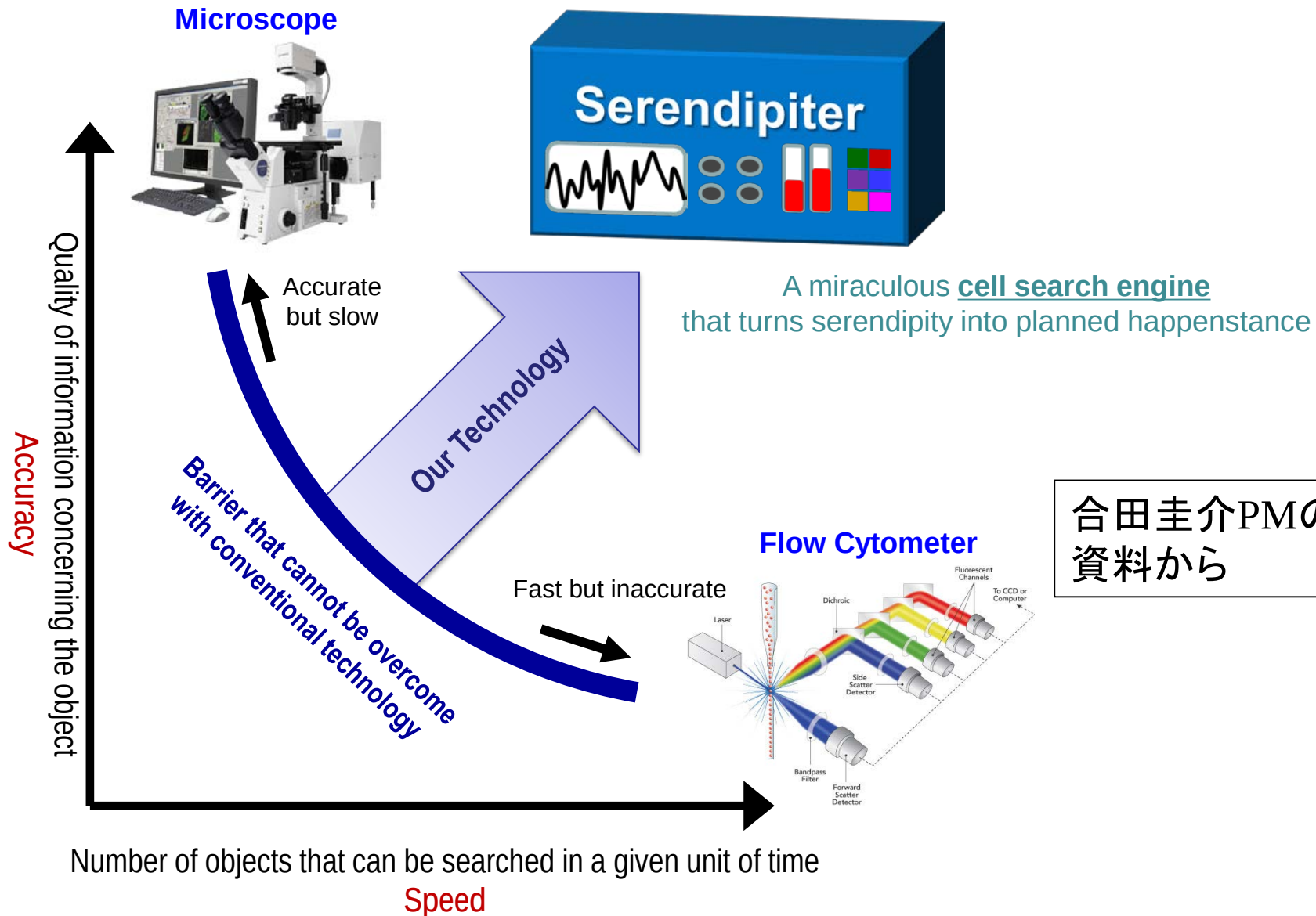
Serendipity = Needle in the Haystack



www.jolyon.co.uk

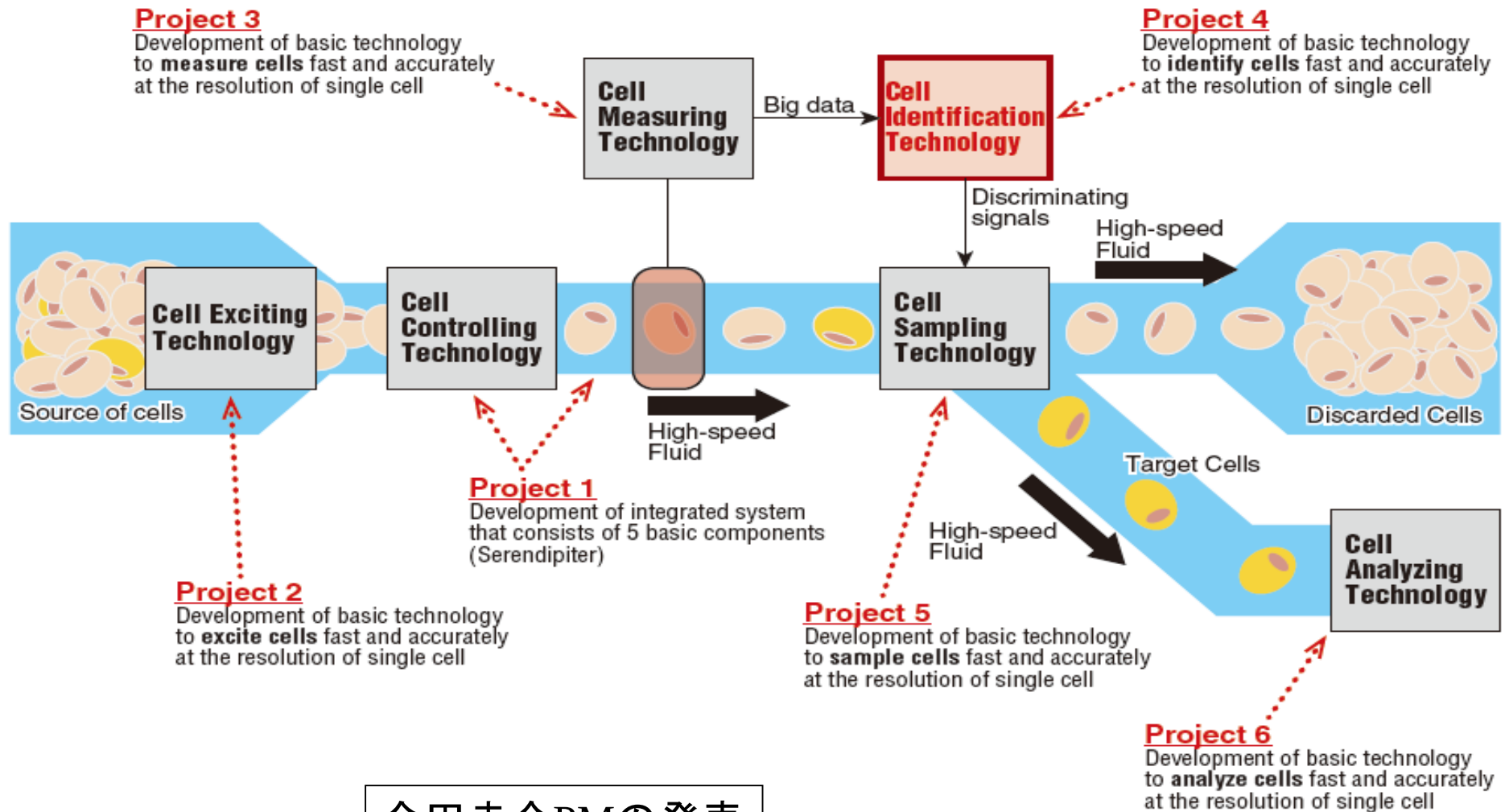
In order to find the needle, it is necessary to conduct an accurate search, but since the probability of finding it is very low, the search resorts to trial and error, and much time is needed before serendipity is reached. Phenomena with low reproducibility are not regarded as scientific, and they cannot be industrialized.

Serendipiter – a miraculous cell search engine



合田圭介PMの発表
資料から

Block diagram of a “Serendipiter”



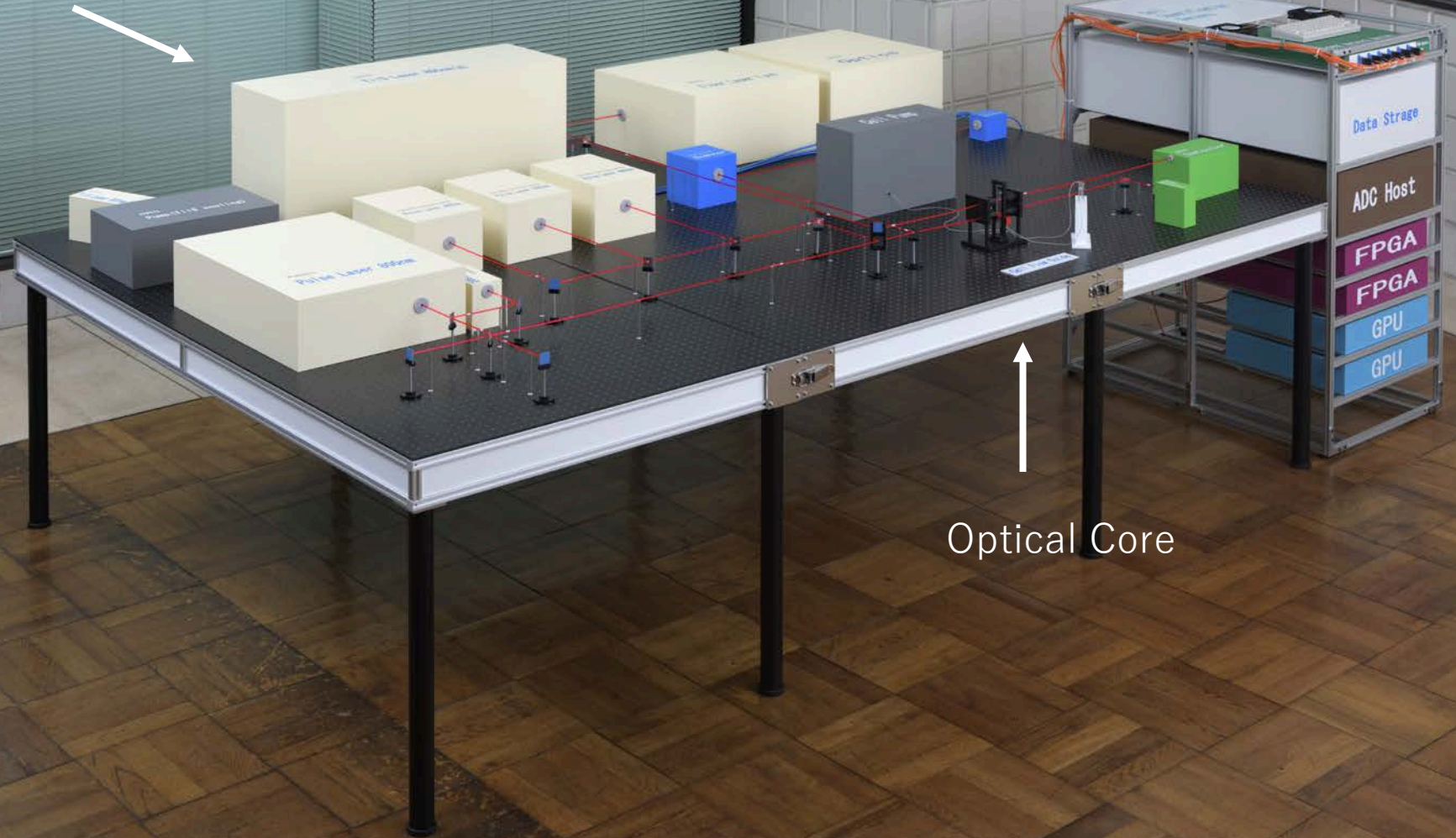
合田圭介PMの発表
資料から

Serendipiter Prototype Model

Lasers

FPGA/GPU based
Selection system

Optical Core



Realtime AIシステム構築

- ・ 超高速Opticsによる連続・多量データの解析と弁別
 - － 1000FPS超の高速度動作実現
 - － 超少数サンプルを多量データから抽出
 - － 小型組み込みシステムとして実現
 - － データレートは約10GB/s



FPGAベース分散並列マシンラーニングシステム
＋
小型超高速ビッグデータシステム

学生たちが創った コンピューター

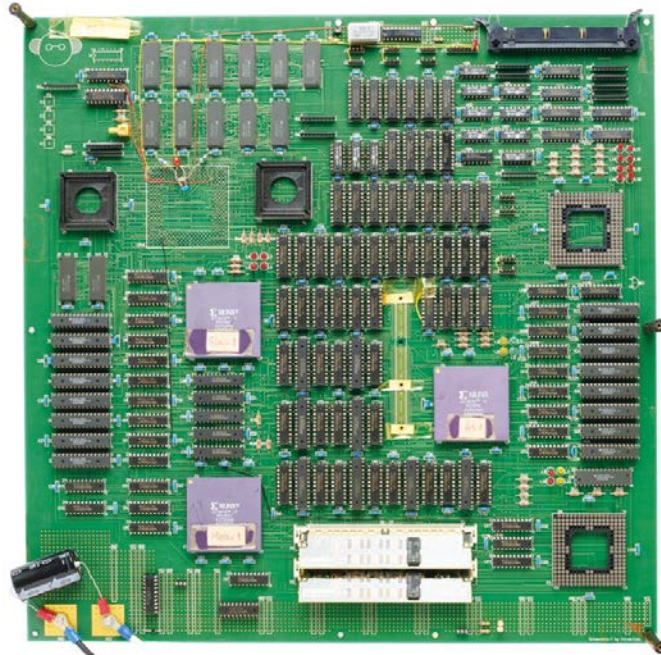
なぜ学生がコンピュータを創るか

- 実システムを作ることは労多くして功少なし
- システムを作ることの重要性
 - 私は高橋・石田・後藤先生のおかげで今日がある
- FPGAを主体としたプロセッサ等が多い

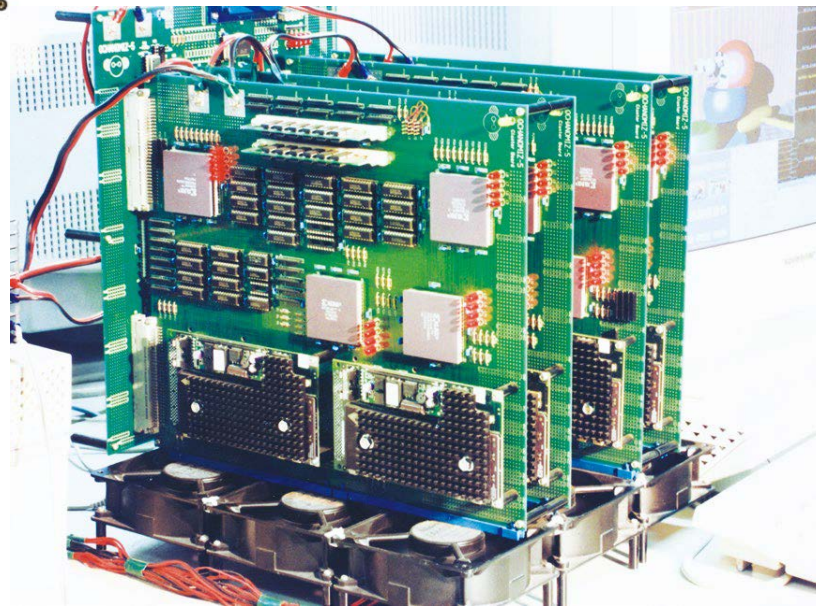
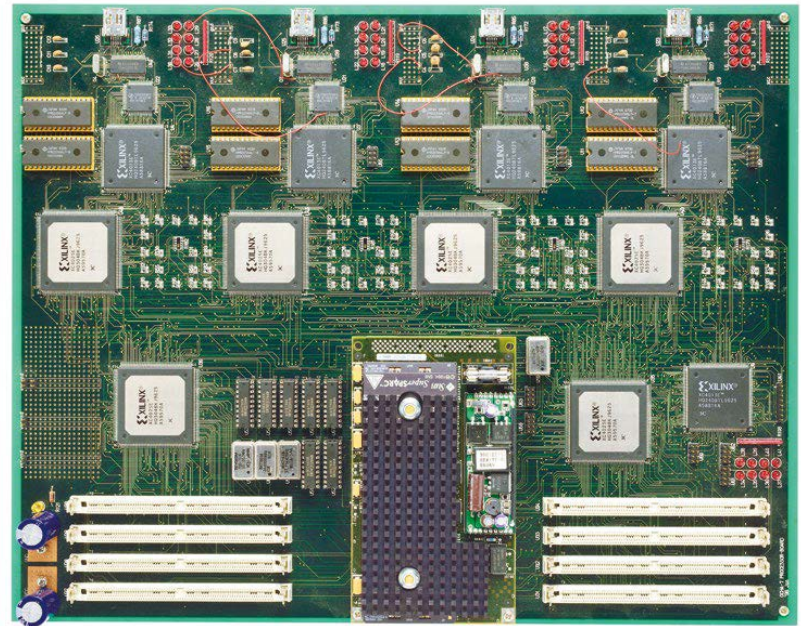
お茶の水シリーズ

- 1, 2, 5, 7号 欠番は計画倒れ

お茶の水7号



お茶の水1号



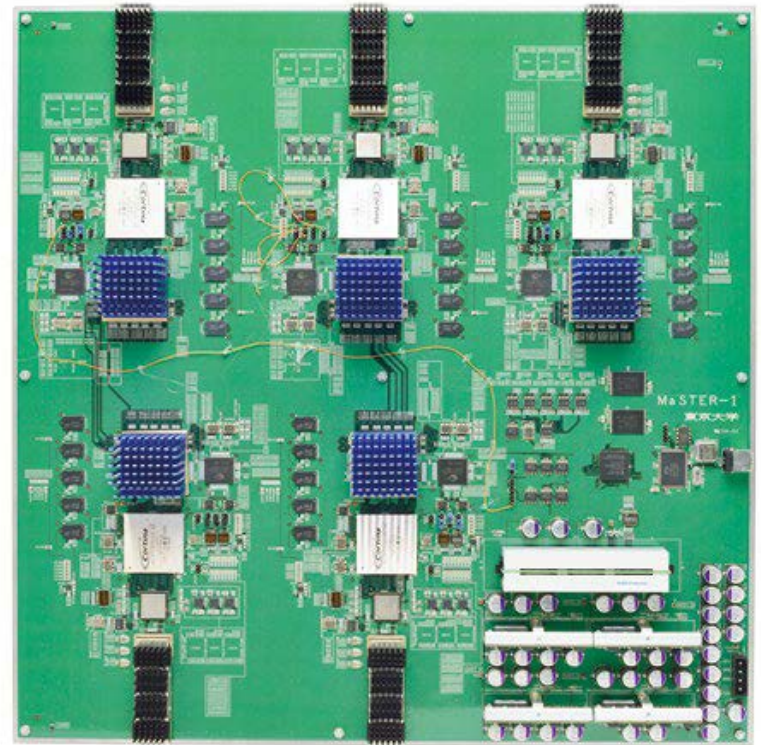
お茶の水5号

10Gネットワークシリーズ

- 実用ツールとして大活躍。今でも使っている。



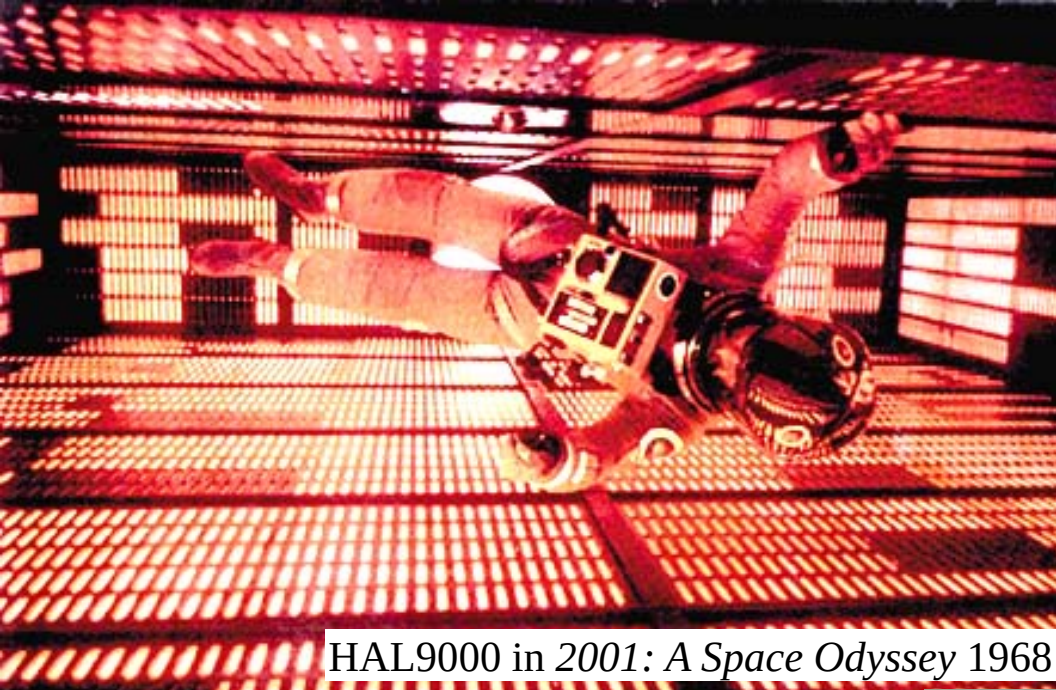
TGLNE-1 ネットワーク
テストベッド



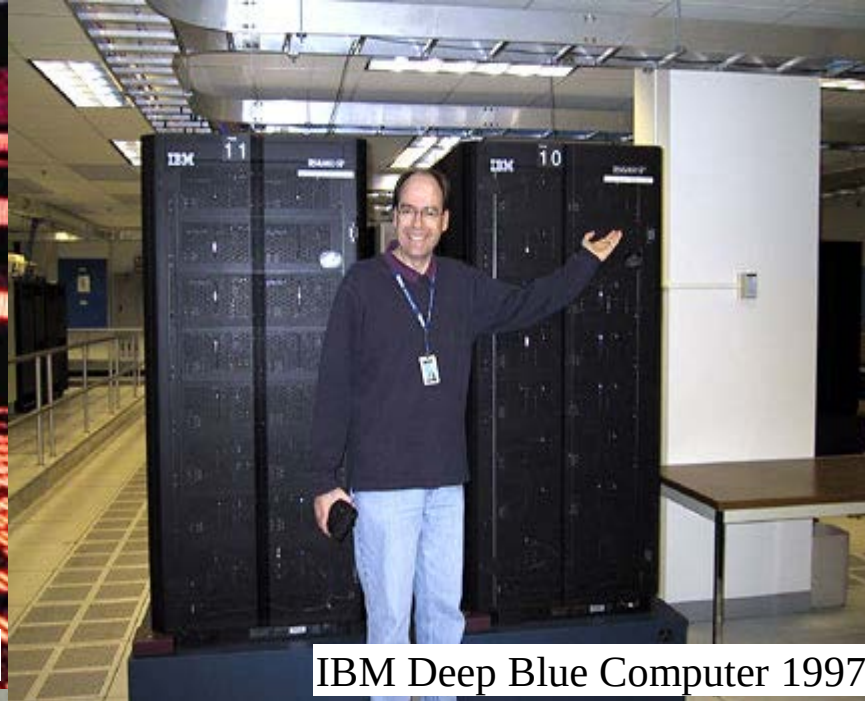
Master-1 5ポートネットワーク
テストベッド

今何が面白い？

- コンピュータ高速化は今が一番面白い
 - 2035年に人間に並ぶコンピュータが出来る
 - 持ち歩けるサイズで役立つ人工知能が出来る
- キーは常に計算速度と消費電力
- 人間という実例があるから出来ない訳がない



HAL9000 in 2001: A Space Odyssey 1968



IBM Deep Blue Computer 1997

コンピュータに人間を 超させたい



IBM Watson Computer (Power7) 2011



GPS将棋(東大) 2013

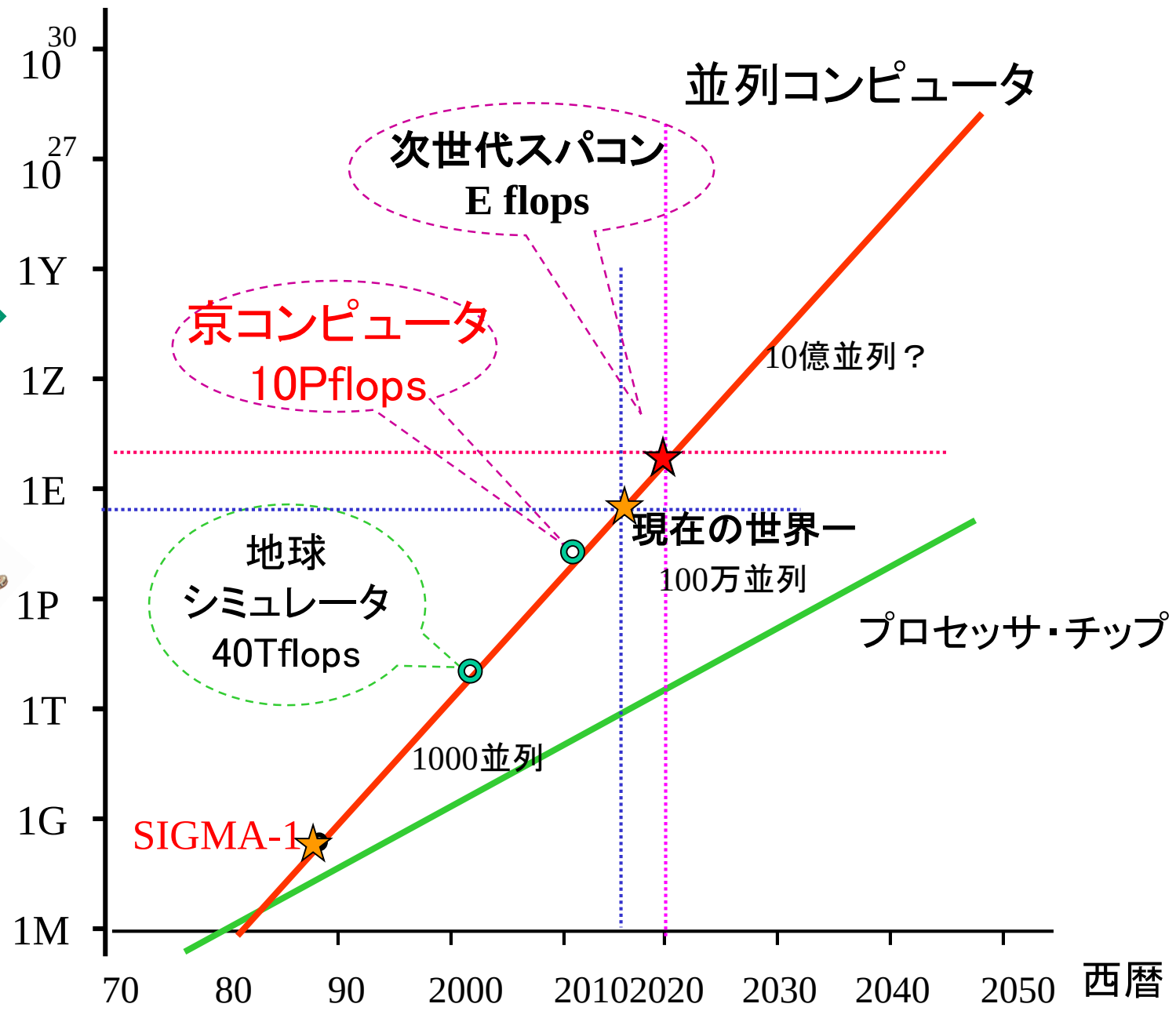
コンピュータが人間を超す

- 計算機が最初から出来たもの
 - 計算、シミュレーション
 - 作表・統計
 - データの蓄積・検索
- 今までにやっと実現したもの
 - 2本足歩行
 - 動画像生成
 - チェス（世界チャンピオンに勝利）
 - 将棋、囲碁
 - インターネット（仮想行動範囲の著しい拡張）
- まだまだ実現しないもの
 - 文章理解、翻訳
 - 音声認識、画像認識
 - 設計、診断
 - 著述、作曲

何故、次第に強くなるか？

- 3側の側面
 - コンピュータの速度が高速化、メモリが増加
 - ソフトウェア(アルゴリズム:戦い方)の進歩
 - 多量の知識ベースの蓄積(ビッグデータ)
- 速度は指数関数的に進歩 → 今後も100万倍以上

FLOPS



これからの30年

- チェス チャンピオンに勝利
- 将棋 プロ棋士に勝利
- 文字認識 実用化。人間並みに近づいている
- 音声認識 実用化。でも、まだまだ能力は低い
- クイズ解答 TV番組でチャンピオンを破る

----- 2017年の現在 -----

- 囲碁 もうすぐ？
- 翻訳 あと10年？
- 東大入試 手強かった。あと20年？
- 視覚・聴覚からの知識や経験の自動取得

- 音声認識、画像認識
- 設計、診断
- 著述、作曲

ロボットの的確な運動制御は、
人間の知能より10倍以上難しい

ここまでは「計算機構成論」に関する 最終講義

次は「情報理工学倫理」に関する 最終講義

としたいが、全部の講義をしめくくるにはあと8倍の時間が必要

ソフトウェア人材の養成

- ソフトウェア教育に取り組んだきっかけは偶然
 - 申請書を書くのが速い
 - 科学技術振興調整費の申請書式に慣れていた
 - 他先生代表者の教育プロジェクト申請書を24時間で代筆した
 - 申請がなぜか通ってしまった

これが切掛けとなって、
以後17年ソフトウェア教育に携わる

ソフトウェア人材養成の取り組み

1. 科学技術振興調整費・新興分野人材養成
 - 「戦略ソフトウェア創造人材養成」 2002～2006
2. 先導的ITスペシャリスト育成推進
 - 「情報理工実践プログラム」 2006～2010
3. 高度IT人材を育成する産学協働の実践教育ネットワーク(enPiT)
 - 「クラウド分野実践教育プログラム」2012～2016
4. (成長分野を支える情報技術人材の育成拠点の形成(enPiT2))
 - 初の学部レベルの情報技術人材教育プログラム 2016～2020
5. プログラミング技法講義 (駒場1年生対象)
 - 高度な情報科学技術人材を駒場から発掘 2015～



http://nc.ice.or.jp

※はじめて 参加される方は、左記のメニューの「参加方法について」をお読みください。

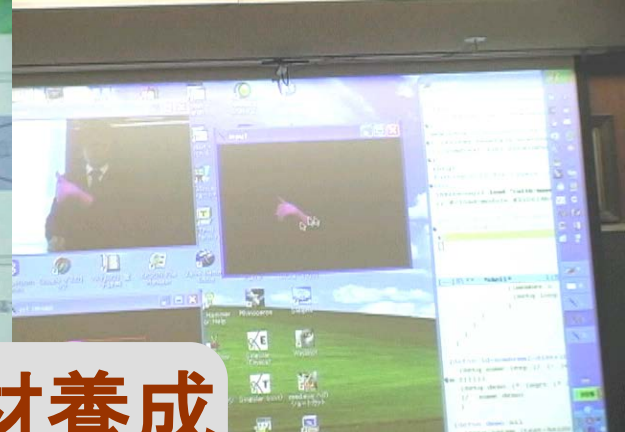
「ことだまレクチャー」を使って授業革命！

ことだまレクチャーは、東京大学大学院生の栗原一貴氏開発による電子白板ソフトウェアで、あいちコミュニティ会員にはGroupRoom内で、フリーにダウンロードできるよう提供しています。このソフトは、コンピュータ上で、黒板とチョークを使い筆紙を扱うような感覚で、マウスだけで操作でき、書き込み、移動、拡大・縮小、画像の貼り付け、リンク、保存等が自由にできる様々な可能性を秘めたソフトです。

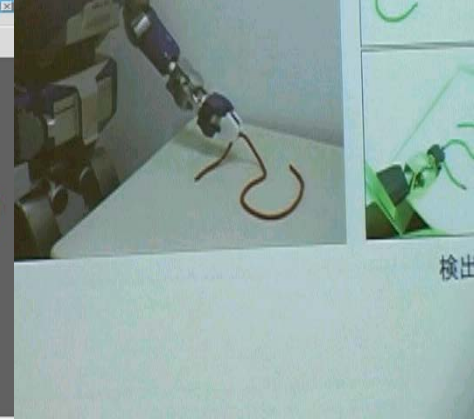
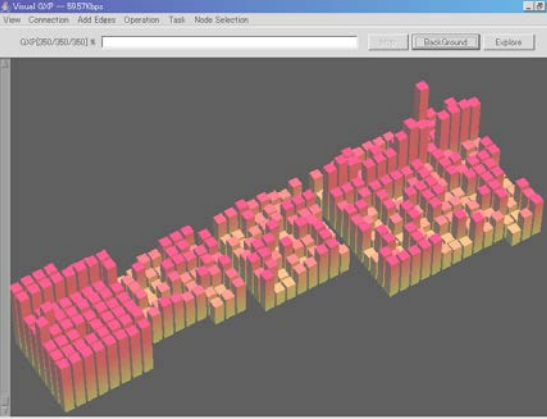
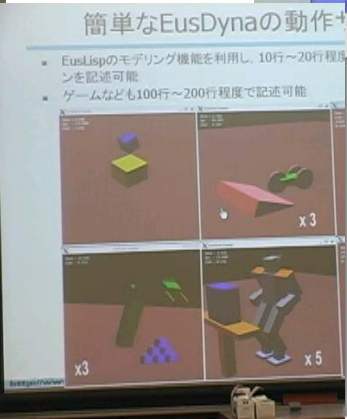
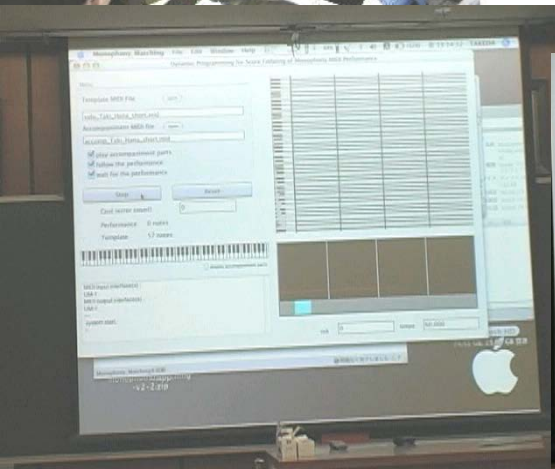
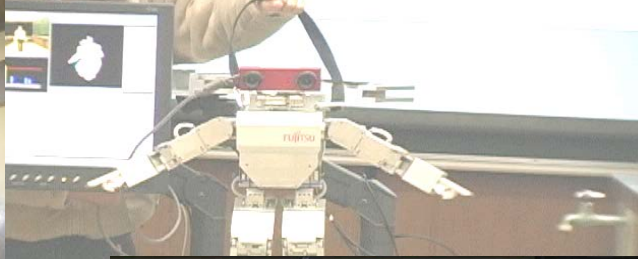
本センターでは、県内の教職員に授業で活用していただき、カスタマイズ要望の意見を募集しています。出された意見を開発者に要望し、今後さらに使いやすいようにカスタマイズされていく開発ツールです。まずは、あいちコミュニティに新規登録してください。

「みんなの広場」で情報の共有！

現在GroupRoom内のみんなの広場では、あいちコミュニティに参加している全会員が閲覧・投稿・掲示板を2つ用意してあります。

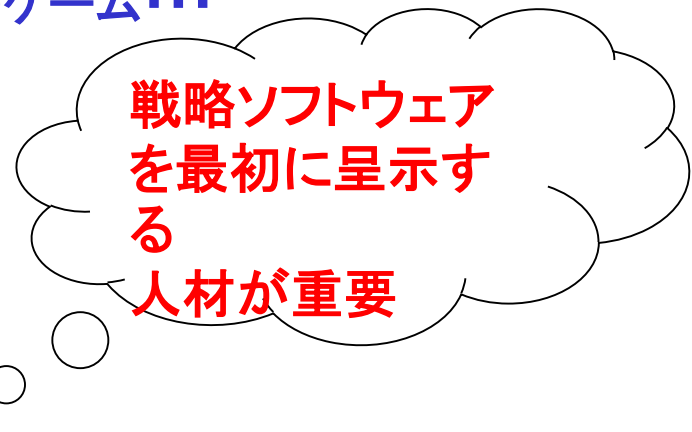


戦略ソフトウェア創造人材養成 最終評価



人材養成ユニットの目的

- 我が国からの戦略ソフトウェアの発信
 - キラーアプリケーション
 - スプレッドシート, ワープロ, ブラウザ, 3Dゲーム...
 - 基盤ソフトウェア(OS, 言語, UI など)
 - デペンダブルシステム?
 - 自己学習ロボット?
 - 大域移動分散基盤ソフトウェア?
 - 起業も視野に入れたソフトウェア戦略
- 戦略ソフトウェア創造者の養成 ← 「研究者」ではない
 - 戦略的分野の選択眼の育成
 - 創造者がもつ生来の創造力の発揮
- 手間をかけた少数教育の実現: 芸術(才能+研鑽)
 - 量的充足を目的とした情報教育と対極
 - OJT型実践教育による創造手段の熟成



戦略ソフトウェア
を最初に呈示す
る
人材が重要

戦略ソフトウェア創造人材養成の概要

- 養成の対象 **情報理工学の専門教育は前提**
 - 博士後期課程学生(修士課程学生も含める)
 - 博士研究員
- 養成コース
 - 2年間(1年後に資格再審査)
 - 1年目 …… 戦略ソフトウェア創造の基礎力, 基礎文筆力涵養
 - 2年目 …… 戦略ソフトウェア創造最終課題への取り組み
 - 養成人数 各対象分野2名, 合計6名/年
 - 全体で18名(3分野各6名)をプロジェクトリーダーレベルに育成
 - 第一年度は募集なし
 - 修了
 - 要件: 戦略ソフトウェア創造講義群 ⇒ 1単位
 - 戦略ソフトウェア創造最終課題の審査をもって修了とする

講義

- a. 戦略ソフトウェア講究
- b. ディペンダブル情報システム論
- c. 大域移動分散システム論
- d. 創造情報学特論
- e. キラーアプリケーションと戦略 ソフトウェア論
- f. Windows Internals
- g. Javaの世界
- h. ブルージーンシステムとシステムソフトウェア
- i. アプリケーションフォーラム

養成する人材のレベル	5年間の実績(目標)
修士課程	120(目標設定なし)
博士課程	40(目標設定なし)
社会人	40(目標設定なし)
創造課題修了者	15(18)

■ Windows Internals(Microsoft社)

Windowsの内部アーキテクチャ
Windowsにおける仮想メモリ管理機構
Windowsが用いる同期機構
Windows・デバイスドライバの入出力の実現
Windowsカーネルプログラミング

■ Latest Topics in Java(SUN microsystems社)

Javaにおける動的記憶管理とガーベジコレクション
J2meと、モバイル機器へのJavaの組み込み
インターネットからのサービスの実現
Javaプログラミング技法

■ BlueGene/L Supercomputer(IBM T.J.Watson研究センタ)

BlueGene/Lのシステムアーキテクチャ
BlueGene/Lのシステムソフトウェア
BlueGene/L用数値アルゴリズム

◎意義

世界における重要システムの開発者から、直接話を聞くことにより

- ・教科書から学ぶソフトウェアと、流通ソフトウェアの設計・作りこみの差を体感
- ・国際的な競争力を持つことの大切さを学習

Dr. D. Probert



Dr. Tony Printezis

Keith Hazelton

Dr. Simon See

Dr. G.Chiu

Dr.M. Gupta

Dr.J Saxton

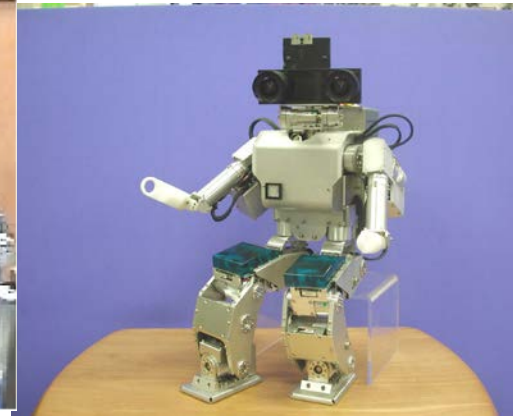
■ソフトウェア演習（認識行動ロボットソフトウェア教育）

ヒューマノイドソフトウェア:

モータ制御から知能まで, 多層大規模な戦略的統合ソフトウェアの好例

- ・ ヒューマノイド用実時間OSに関する講義
- ・ 特任教員本人が創造したヒューマノイド行動ソフトの理解
- ・ 特任教員による, ロボット・ソフトウェアにおける実時間OSの実習

◎意義: TOPレベルの若手開発者本人により, 戦略的ソフトウェアの実例を学ぶ



戦略ソフトウェア創造人材養成まとめ

- 科学技術振興調整費を切掛けとした2学科・専攻
 - 情報理工学系・創造情報学専攻
 - 理学部・生物情報科学科

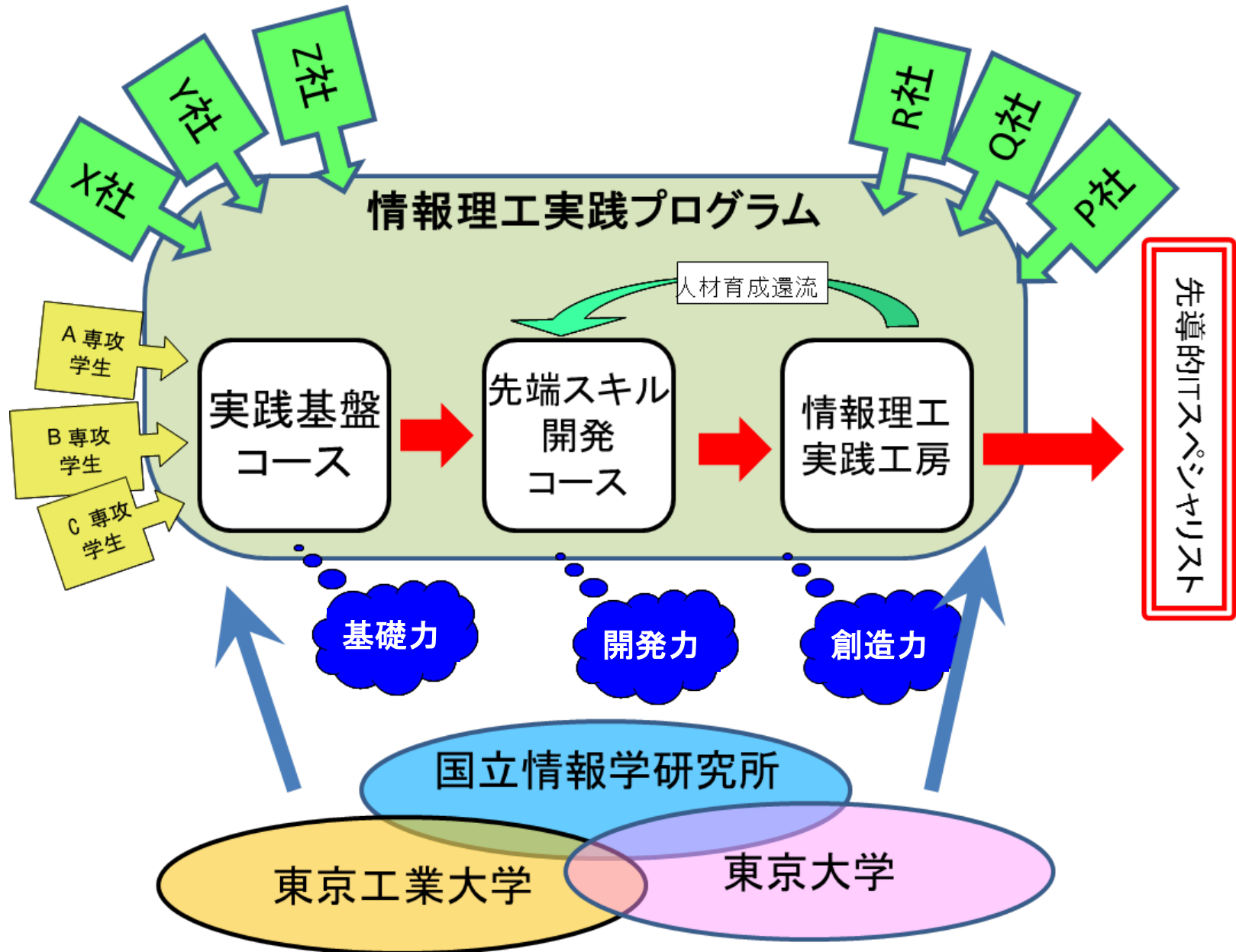
こうして私は
創造情報学専攻の教員になりました



先導的ITスペシャリスト育成推進 「情報理工実践プログラム」

東京大学情報理工学系研究科
東京工業大学情報理工学研究科
国立情報学研究所

情報理工実践プログラムの概要



質問①世界最高水準のIT人材育成が可能となったのか. また, その将来性 それを実現するための具体的な取り組みとその成果

- 三位一体の教育方法

- 『実践基盤コース』

- IT技術者が体得しておくべき基礎科目
 - 基礎力, 開発力, 創造力の要素をバランスよく習得

- 『先端スキル開発コース』

- NIIの蓄積による最新教材を開発
 - 東大・東工大合同の集中講義として実施(東大秋葉原拠点)
 - 最先端のソフトウェア工学による開発スキルを獲得

- 『情報理工実践工房』

- 実務家教員との緊密に連携した, プロジェクトを基本としたグループワーク(PBL)によるソフトウェア開発実習
 - 実務家による特別講演
 - 海外研修
 - 実践的創造力を涵養

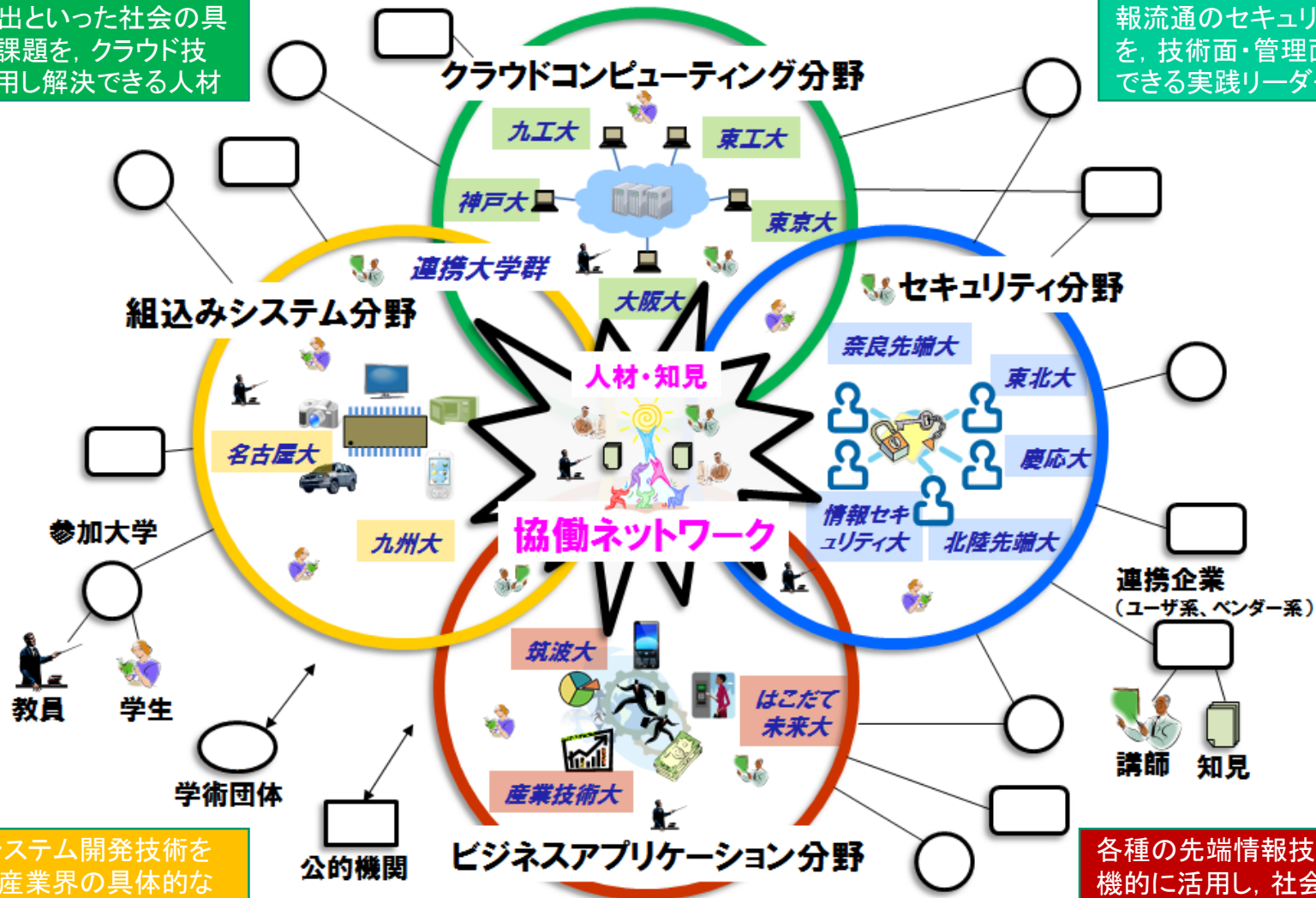
情報理工実践プログラム まとめ

- 一番苦労したのは、事業仕分け(2009)
 - 残念ながら仕分けられてしまいました (大幅な見直し)
 - 同じ事業仕分けでスパコンも仕分けられてしまいました(大幅見直し)
名言「二位じゃ駄目なんではないですか？」が生まれた
- 次のIT教育立ち上げが大変だった & 文部科学省からの数字締め付けが大変になった
- 創造情報学専攻での教育スタイルが確立した

高度IT人材を育成する産学協働の実践教育ネットワーク

いわゆるビッグデータの分析手法、新しいビジネス分野の創出といった社会の具体的な課題を、クラウド技術を活用し解決できる人材

社会・経済活動の根幹にかかわる情報資産および情報流通のセキュリティ対策を、技術面・管理面で牽引できる実践リーダー



組み込みシステム開発技術を活用して産業界の具体的な課題を解決し、付加価値の高いサイバー・フィジカル・システムズを構築できる人材

各種の先端情報技術を有機的に活用し、社会情報基盤の中核となるビジネスアプリケーション分野の実践的問題解決ができる人材。

大阪大学のスライドから

教育内容・方法

- 必要な基礎知識を習得した学生に対して，課題解決型学習(PBL)を中心とした短期集中合宿，分散PBLを柱とした教育プログラムを構築

4月～7月

基礎知識学習
短期集中合宿
受講のための
事前準備

連携大学，参
加大学の講義
，edubase教
材の利用

7月末～9月の2週間程度

短期集中合宿
各分野の講義
PBL
分散PBLに向けた準備等

分野内の拠点(複数可)に
集まって実施。連携大学，
参加大学教員，実務家教
員が指導

10月～12月

分散PBL
連携大学，参加大
学の学生が分散
環境でPBLを実施

主に連携大学・
参加大学の教
員が指導

発表会



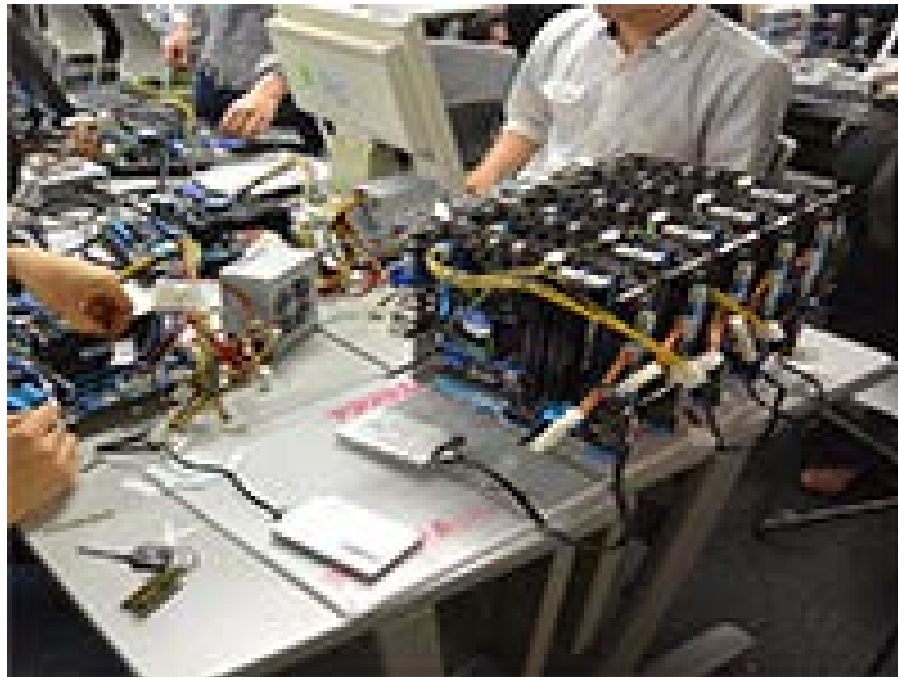
修了

注文の多い料理店状況

- 教員のFDの促進
 - 若手教員を実際の授業や演習に積極的に参加, 実践的情報教育の内容や指導の仕方を習得させる
 - FD委員会やワーキンググループ等を整備し, その知見を共有できる環境を整備
- 知見の収集と共有
 - ソフトウェアの開発環境や実行環境の提供(edubase Cloud)
 - 各連携校で行われる授業や演習をビデオ記録し, それをビデオ配信サーバに蓄積し公開
 - 国立情報学研究所が提供するedubase Streamを利用
- 広報を通じた裾野の拡大
 - 本協働ネットワークへの参加大学, 連携企業を拡大するために, 積極的に広報活動を行う.
 - ポータルサイト, パンフレットを整備
 - シンポジウムの定期的な実施

enPiT クラウド実習風景

- クラウドを創る実習 パーツから組立、クラウドソフトウェアを入れて自分のクラウドを構築する
- その後、このプライベート・クラウドを使うアプリ開発実習



enPiT東日本クラウド まとめ

- 阪大・井上先生に大変お世話になりました
- 事業仕分けの経験をへて、注文の多い料理店状態
- 質より数字の注文が一番困った
- 同じことをするための予算がどんどん減少
- 情報システムを構築することに燃える学生多数

千葉先生

enPiT2をよろしくお願いいたします

プログラミング技法

- プログラミングを専門としたい学生の発掘が目的
- 万人向けの講義ではなく、深い関心を持つ学生向け
- 後半は秋葉さん、岩田さんが担当

(前半)

- 計算機システムの理解、実行メカニズムの理解
- Java, C++, Rubyによる簡単なプログラミング
- 高速化技法と、その効果の測定 など。

(後半)

- 競技プログラミング入門
- ペアプログラミング法
- 実践的競技プログラミング

課題4

- 行列積のプログラム、またはバブルソートではないソートのプログラムを出来るだけ高速化せよ
 - 高速化前のプログラムに(コンパイラの最適化オプション以外の)高速化を施し、実行時間差を表にまとめる。
 - 注意: 高速化しやすいように最初のプログラムに駄目プログラムを使わないように

実行結果例 (on Mac/Linux)

Mac環境 = Xeon W3520, OSX 10.10.5, Apple LLVM 7.0.0 (clang-700.0.72) -
Ofast, JDK 1.6.0_65 (Server VM), ruby 2.0.0p481

C				
500,	0.000184,	0.000184,	0.000183,	0.000184,
2000,	0.002883,	0.002933,	0.002883,	0.002919,
4000,	0.011710,	0.011664,	0.011635,	0.011806,
6000,	0.025798,	0.025814,	0.026381,	0.025532,
8000,	0.046585,	0.045872,	0.044729,	0.045478,
10000,	0.071115,	0.071065,	0.069996,	0.071442,

Java				
500,	0.003559,	4.5E-4,	2.03E-4,	1.97E-4,
2000,	0.002943,	0.002998,	0.00296,	0.002939,
4000,	0.011844,	0.011839,	0.011813,	0.011724,
6000,	0.026704,	0.026679,	0.026775,	0.026665,
8000,	0.047838,	0.047576,	0.047828,	0.04762,
10000,	0.075231,	0.074106,	0.074147,	0.074319,

Ruby				
500,	0.041676,	0.041382,	0.041463,	0.041565,
2000,	0.653054,	0.651173,	0.651091,	0.652389,
4000,	2.608098,	2.603893,	2.649164,	2.641159,
6000,	5.869905,	5.920909,	5.938819,	5.904991,
8000,	10.525021,	10.507519,	10.531742,	10.473472,
10000,	16.40252,	16.862218,	16.603004,	16.376962,

Linux環境 = Core i7-920, CentOS 7.1, GCC 4.8.3 -O3, JDK 1.8.0_60, Ruby 2.0.0p598

C				
500,	0.000185,	0.000190,	0.000185,	0.000185,
2000,	0.002933,	0.002934,	0.002937,	0.002937,
4000,	0.011639,	0.011679,	0.011641,	0.011637,
6000,	0.026111,	0.026138,	0.026152,	0.026173,
8000,	0.046522,	0.046382,	0.046380,	0.046361,
10000,	0.072390,	0.072400,	0.072404,	0.072378,

Java				
500,	0.005704471,	0.001819815,	5.43581E-4,	2.07921E-4,
2000,	0.003087148,	0.003086791,	0.00295379,	0.003080792,
4000,	0.011716112,	0.011834578,	0.011823093,	0.011849949,
6000,	0.026388724,	0.02649999,	0.026641345,	0.026638378,
8000,	0.047312579,	0.047174071,	0.047170846,	0.047277527,
10000,	0.073556909,	0.073694132,	0.074268603,	0.074842909,

Ruby				
500,	0.050006323,	0.042032965,	0.037384864,	0.033574135,
2000,	0.495958805,	0.49685297,	0.496542461,	0.496584893,
4000,	1.984901537,	1.984239613,	1.98564696,	1.983702763,
6000,	4.469238888,	4.473405921,	4.510673699,	4.468921489,
8000,	7.941280614,	7.940602592,	7.9413131,	7.946932947,
10000,	12.40263134,	12.37210187,	12.36273734,	12.379538527,

プログラミング技法講義の感想

- 前期課程でのプログラミング教育の不足を実感した
- 学生のレベルの高さに驚いた。
- プログラミング関係教育の一層の充実を期待
- 大変だったけれど、やりがいがある講義だった

萩谷先生よろしくお願いいたします

まとめ

- 約40年間で5.5システム構築にかかわれ幸せ
- 情報システム作り教育は、作らないと絵に描いた餅
- ソフトウェア教育は効果が目に見えて良かった
- 職業上の必要性から論文も書いています
 - Kei Hiraki DBLPで検索
 - Google scholarで検索 すれば重要なものは出ます

今後ともよろしく願ひいたします

質問！

超高速計算システム、
情報教育に
関することならば
何でも回答します

使用した写真について

- 使用した写真は、主に三浦健司氏が撮影しました
- お借りした写真については、キャプションに記載しました
- インターネット研究で用いた写真は平木撮影
- ホレリス統計機、Manchester Mark-1の写真は日本国内では著作権保護期間が終了しています。
- 私が作成していないスライドについては注を入れました
- 本講演資料作成に当たり、赤嶋映子氏、本城剛毅氏のご協力を頂きました。