

高性能計算基盤

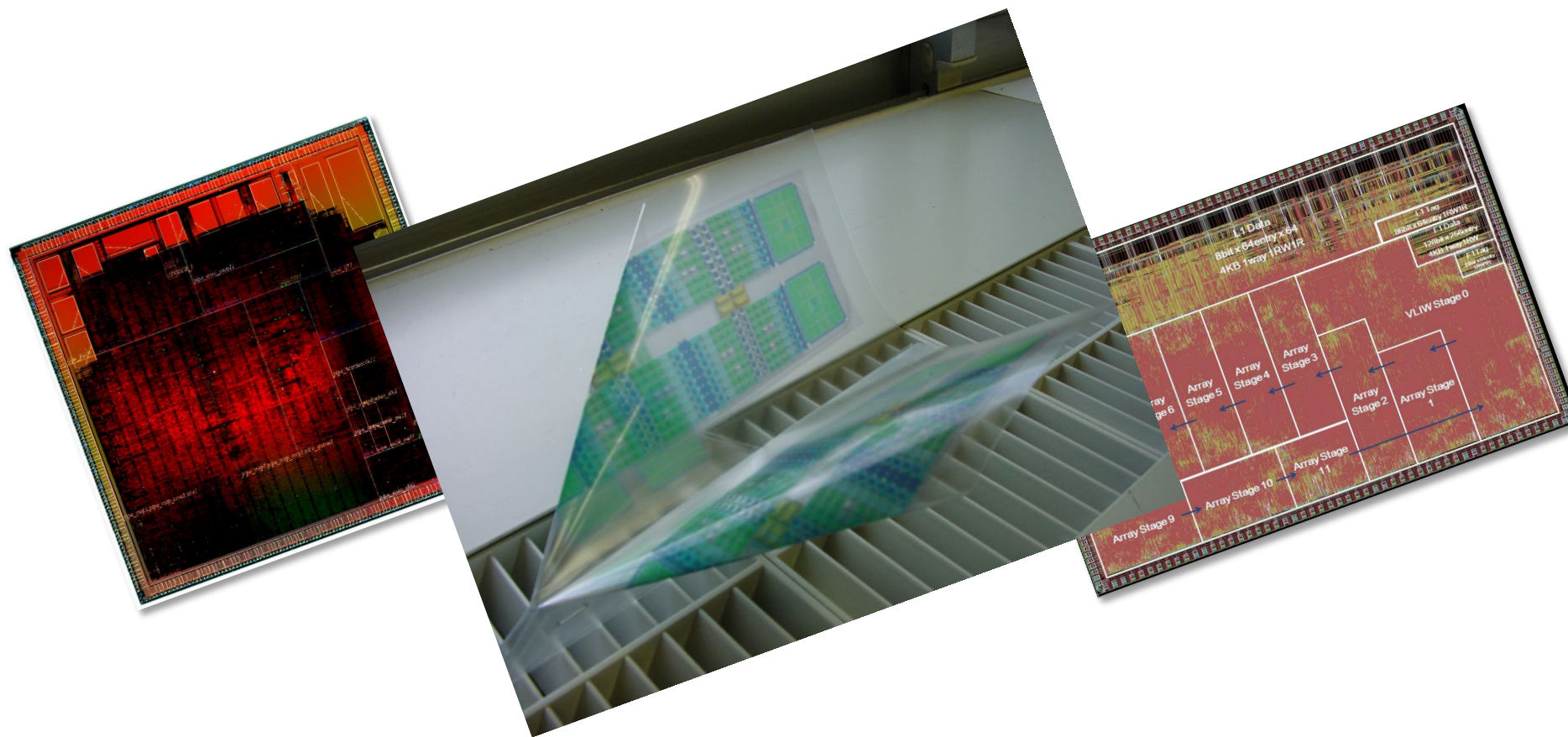
自習 APDX08:Tiny Emulation

<http://archlab.naist.jp/Lectures/ARCH/x08/apdx08j.pdf>

Copyright © 2021 奈良先端大 中島康彦

ナレーター VOICEVOX:もち子(cv 明日葉よもぎ)

フィルムコンピュータ実現に向けた エミュレータ専用小型CPUの検討

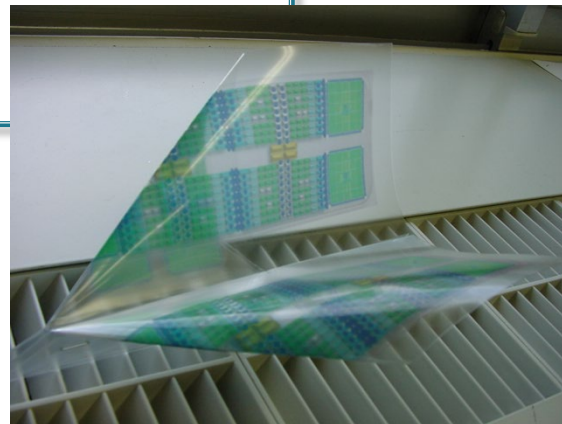
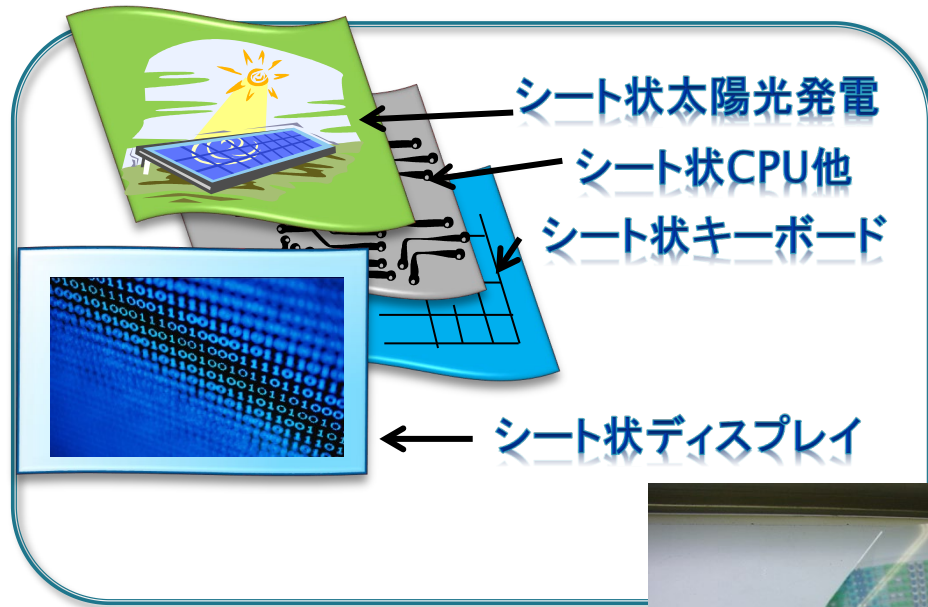


NAIST中島康彦

プリンタブル・フィルムコンピュータ

新サービスの起爆剤

捨てカメラ、忘れ物、見守り、リモコン、誘導、追跡、緊急
時通信、災害時センシング、大量散布＋ネットワーク形成



融合研究に熱心な大学なので

アドホックネットワーク等応用(情報)



この人はARMが
良いと言う

高信頼アーキ(情報)

テスト容易化設計(情報)

回路モデル化技術(物質)

生分解可能素材
(バイオ)

高分子材料
(物質)

酸化亜鉛TR
(物質)



この人はTr沢山
は無理と言う

酸化亜鉛IGZOをベースに

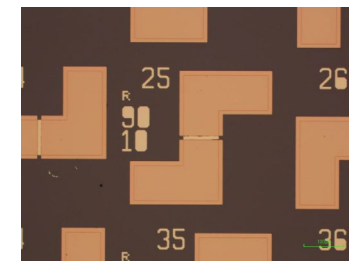
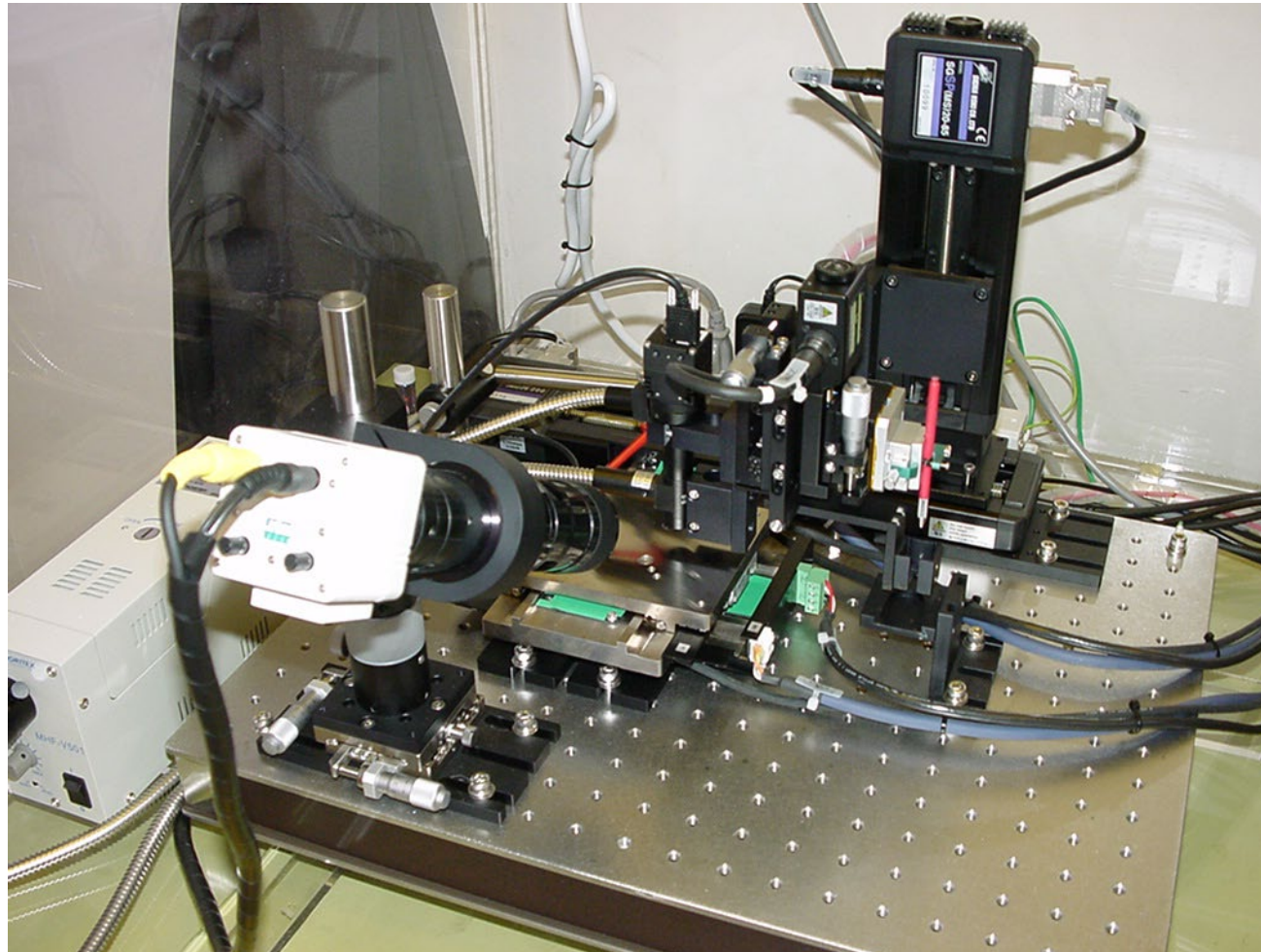
新素材: 如何にトランジスタを構成し

モデル: 如何にパラメタ抽出を行い

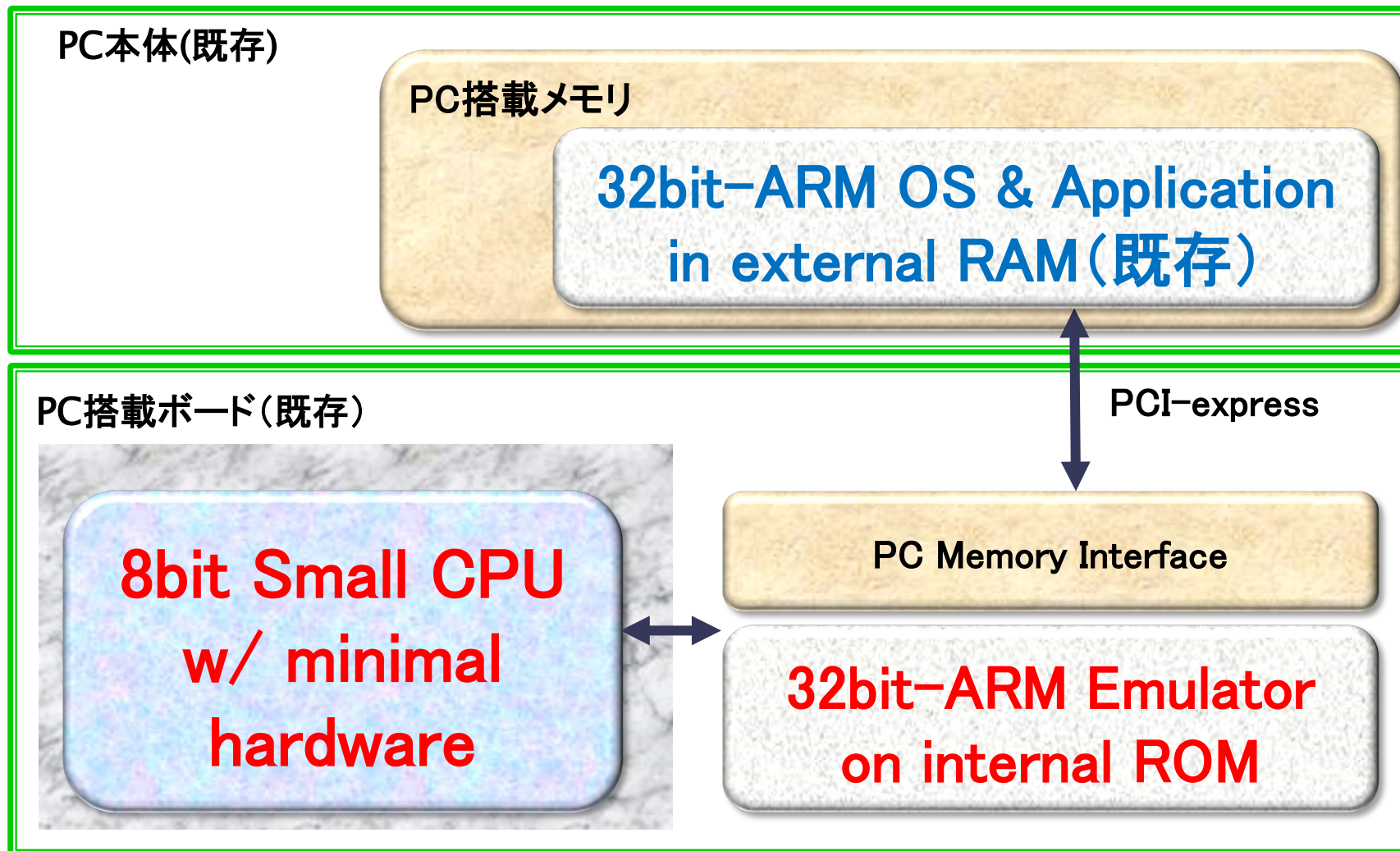
テスト・アーキ: 如何に安定回路を構成するか

回路描画用インクジェットプリンタ

- 線幅0.5 μ 程度、5センチ角まで
- 250 $^{\circ}$ C程度で焼けばよい(今はガラス、将来フィルム)



アーキテクチャ階層の当面の仕事



本発表の流れ

① ARM命令セットのおさらい

② エミュレータ専用8bit-CPUの考え方

論理の複雑さとプログラムサイズのトレードオフ

ゲスト命令デコード機能

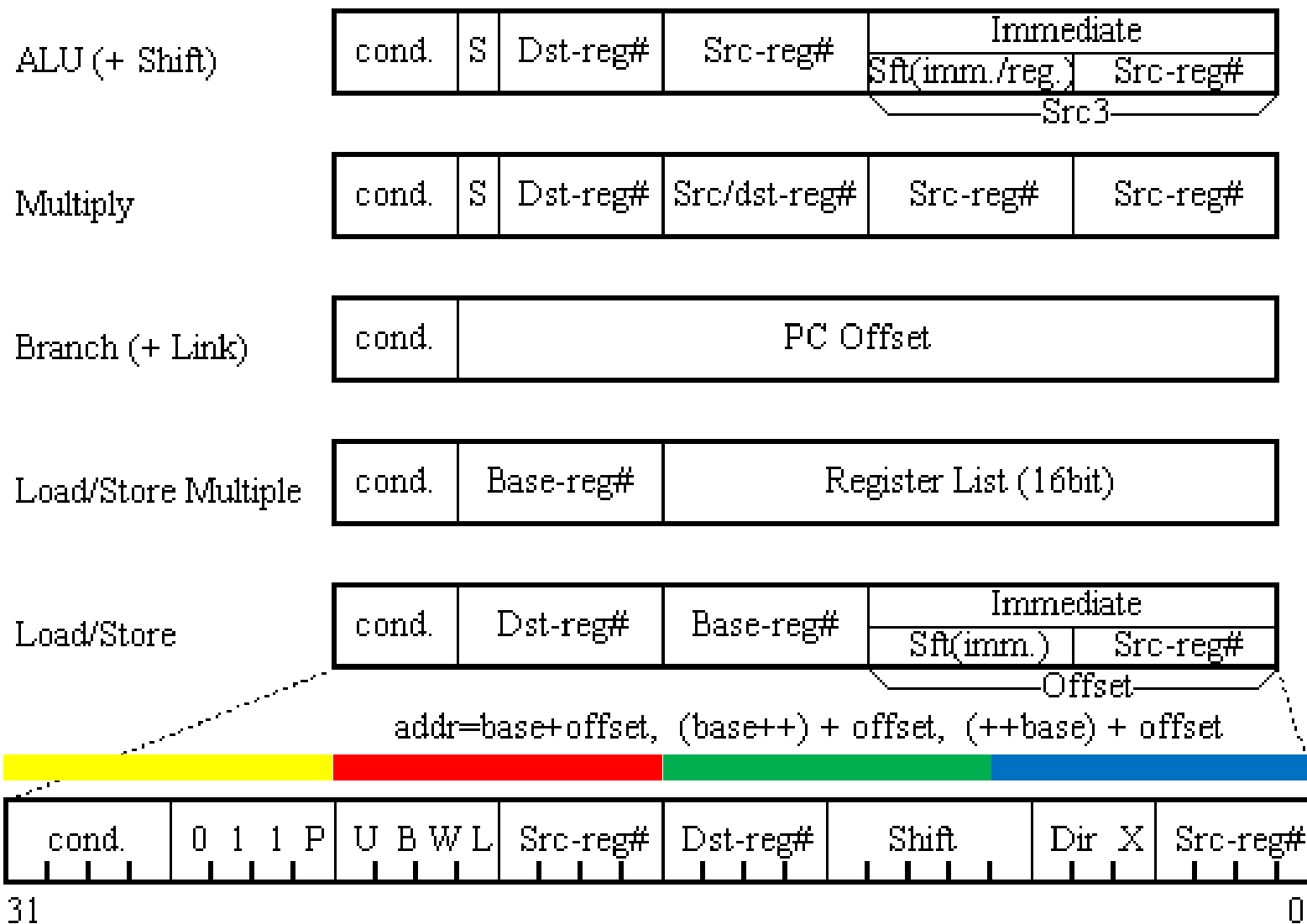
ゲスト演算機能

③ エミュレータの評価

静的ステップ数からわかったこと

動的ステップ数とサイクル数からわかったこと

① ARM命令セットのおさらい



② エミュレータ専用8bit-CPUの考え方

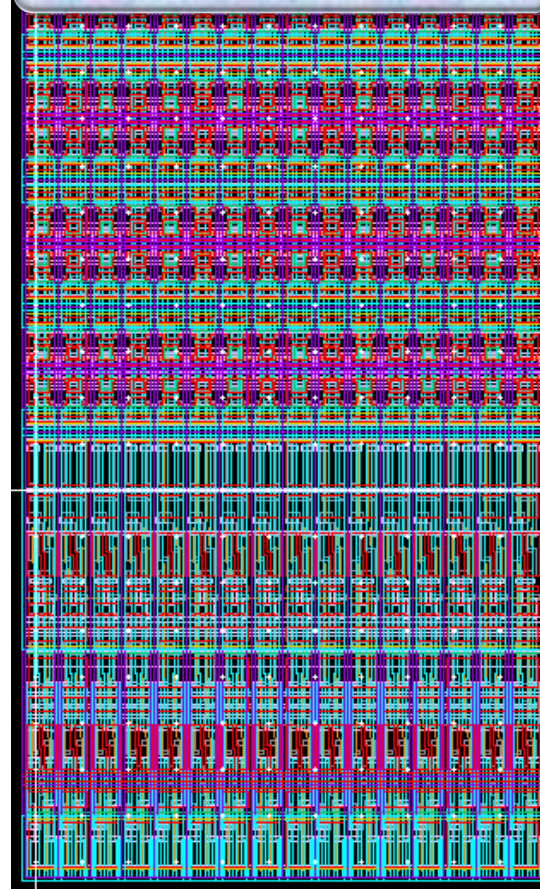
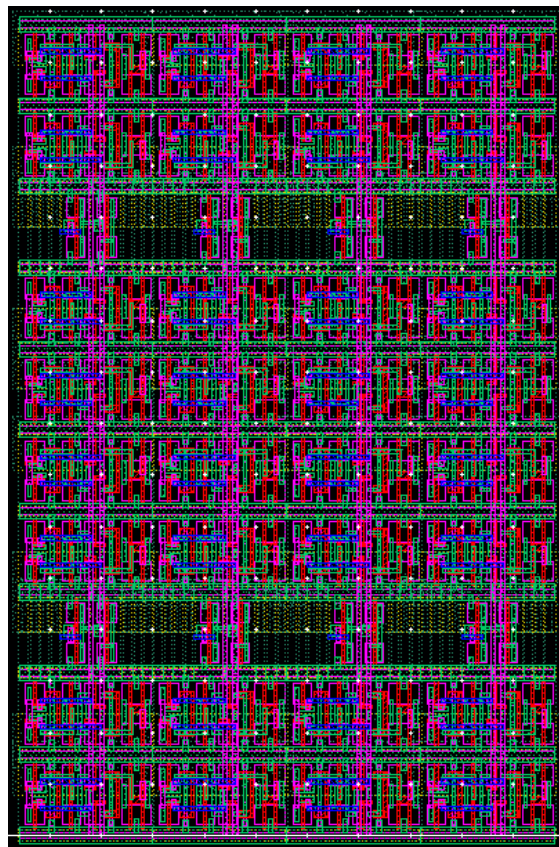
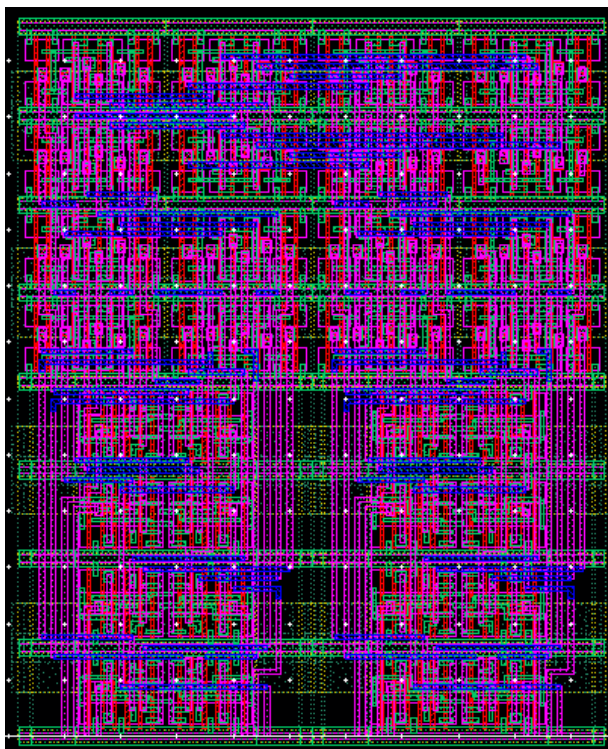
論理を増やすか

RAMを増やすか

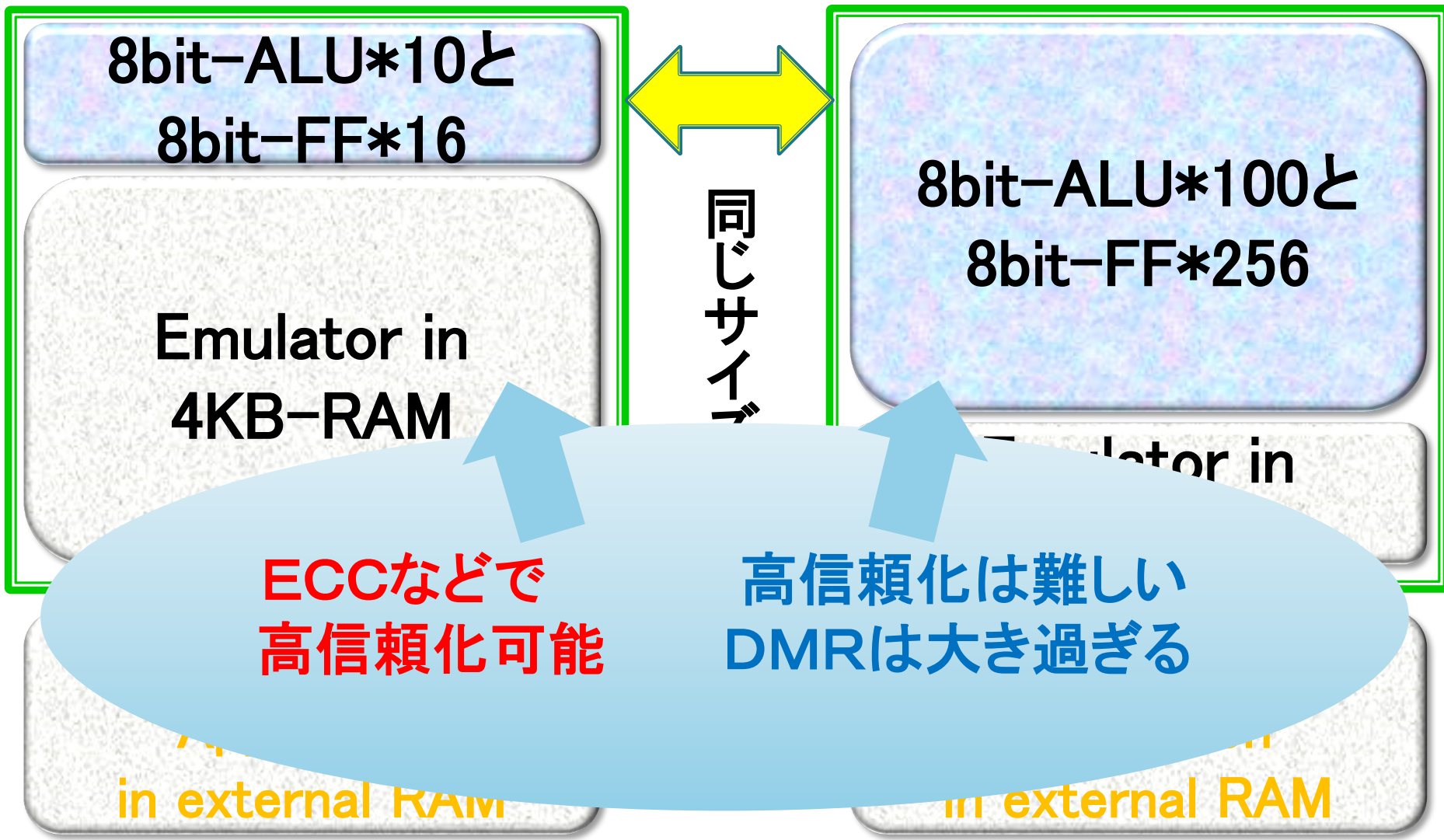
8bit-ALU+

32bit-FF

128bit-RAM

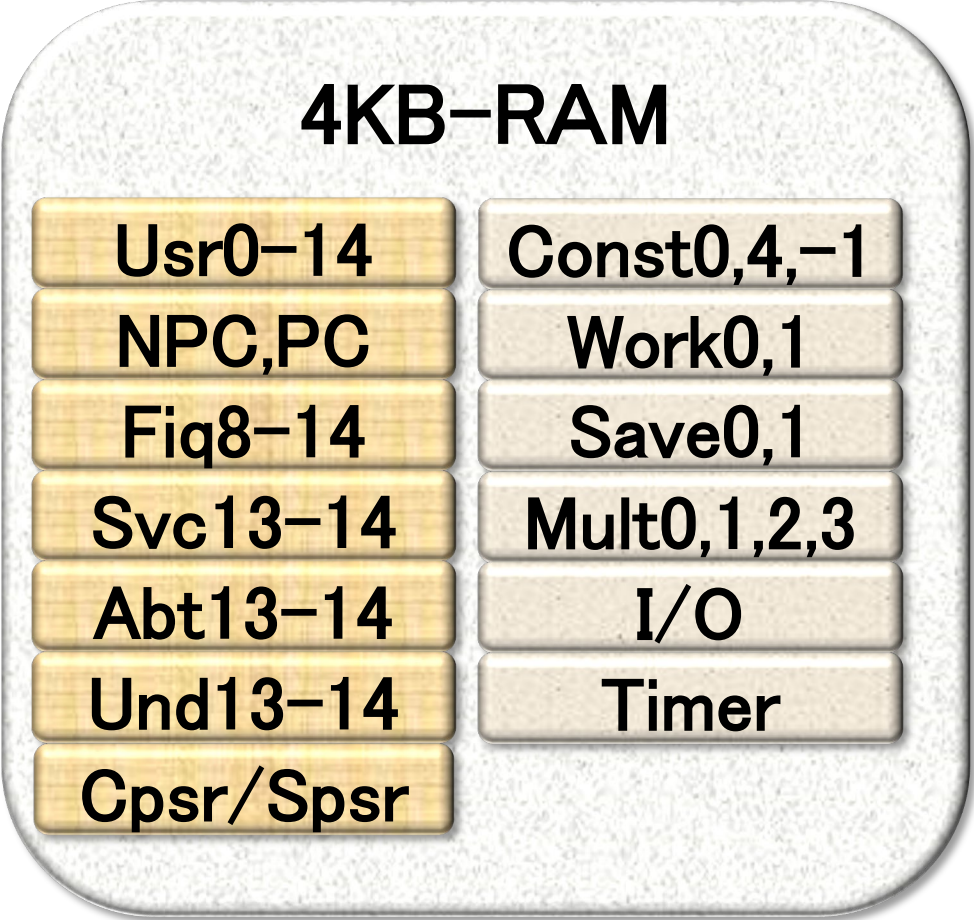
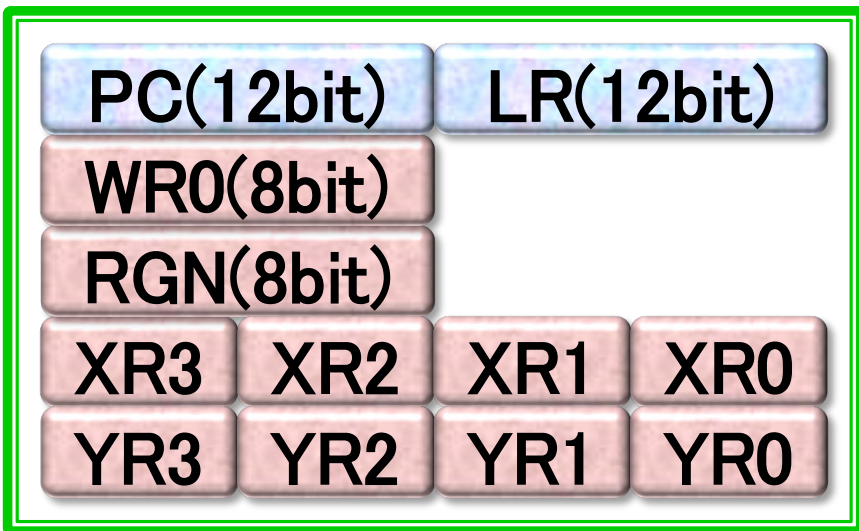


論理の複雑さとプログラムサイズのトレードオフ



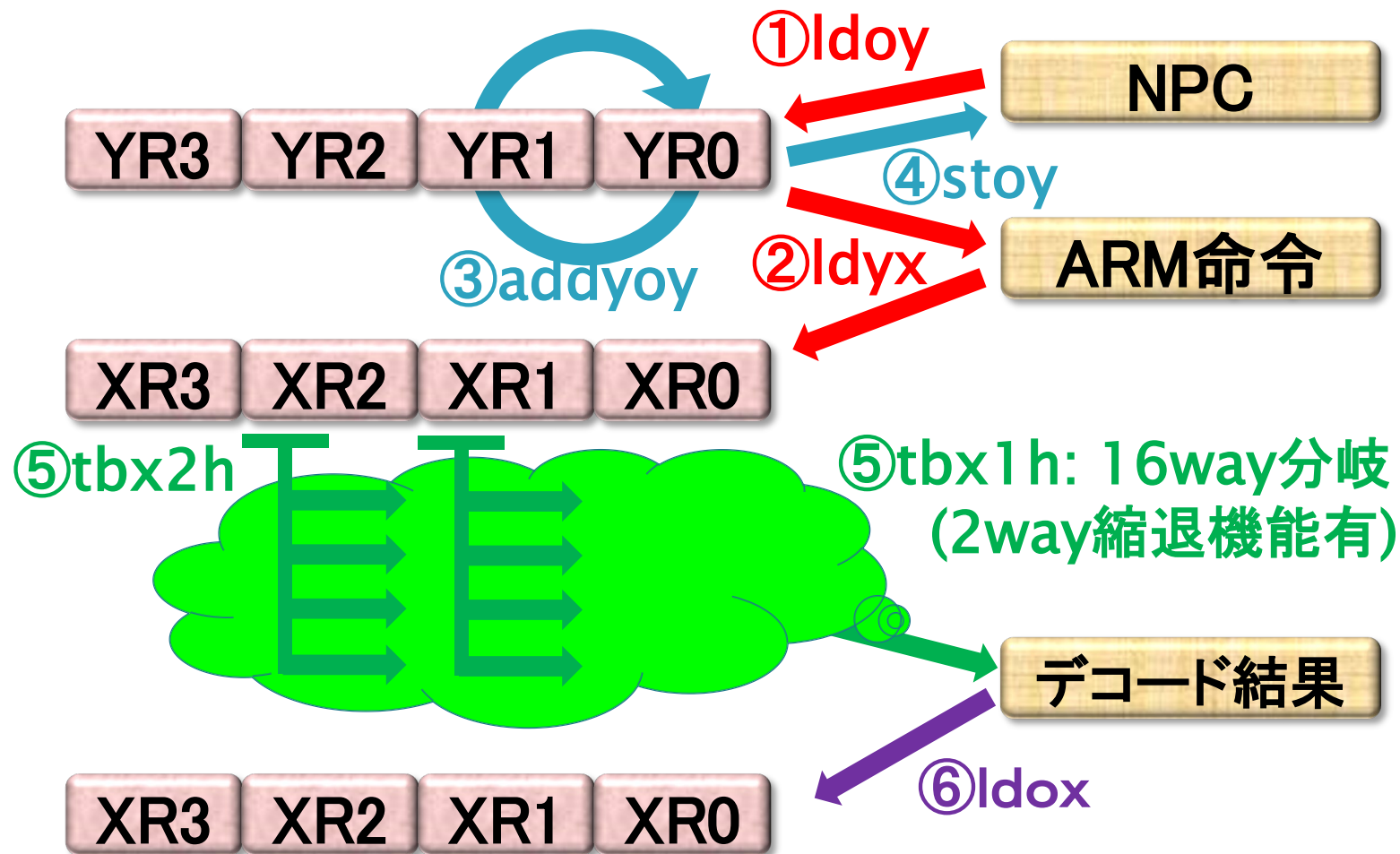
レジスタの配置

- 最小限の演算器・FFで作ることに
- 若干のハード+マルチサイクル動作を許容



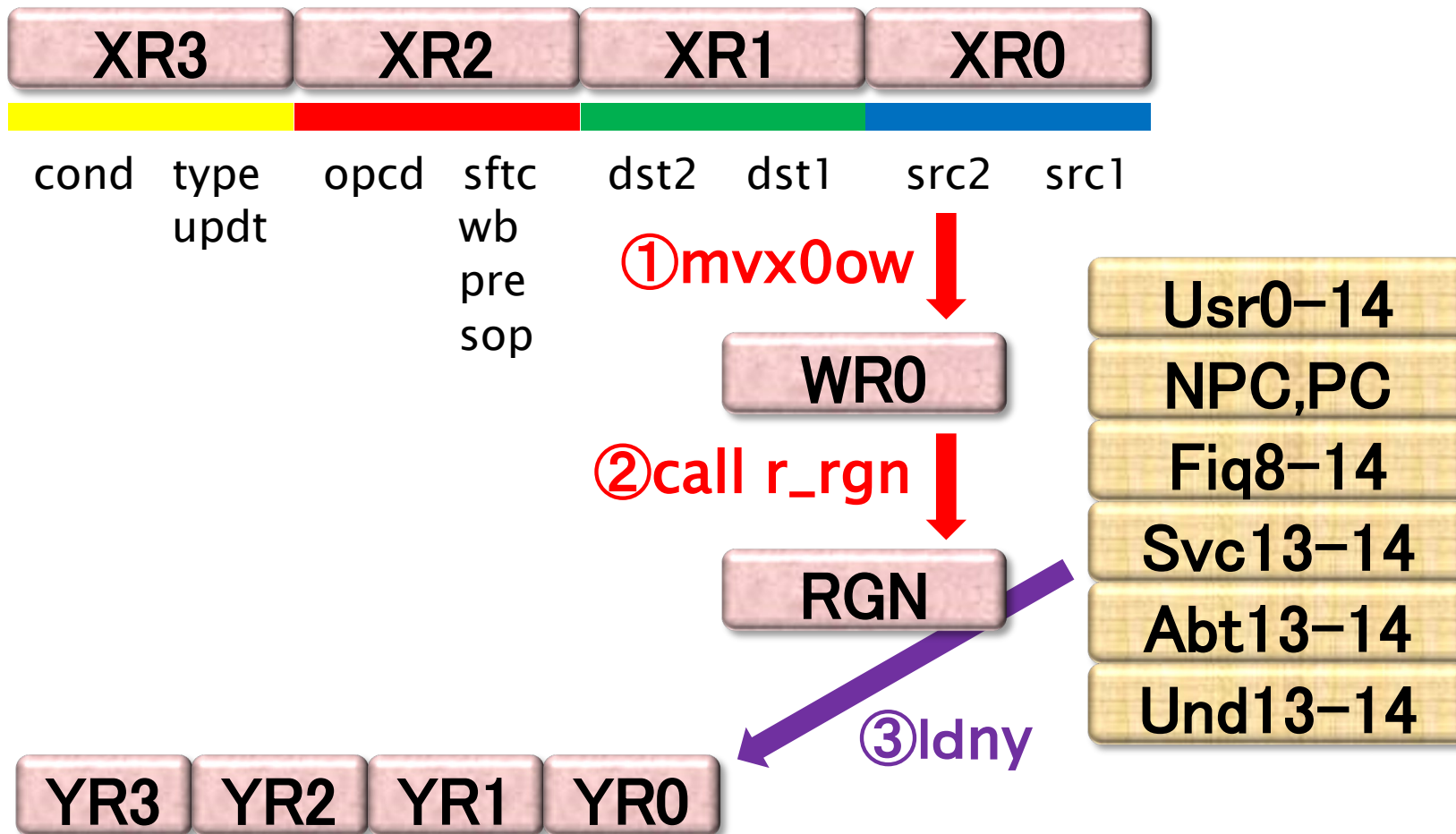
ゲスト命令デコード機能

➤ エミュレータサイズ抑制のため2段階デコード



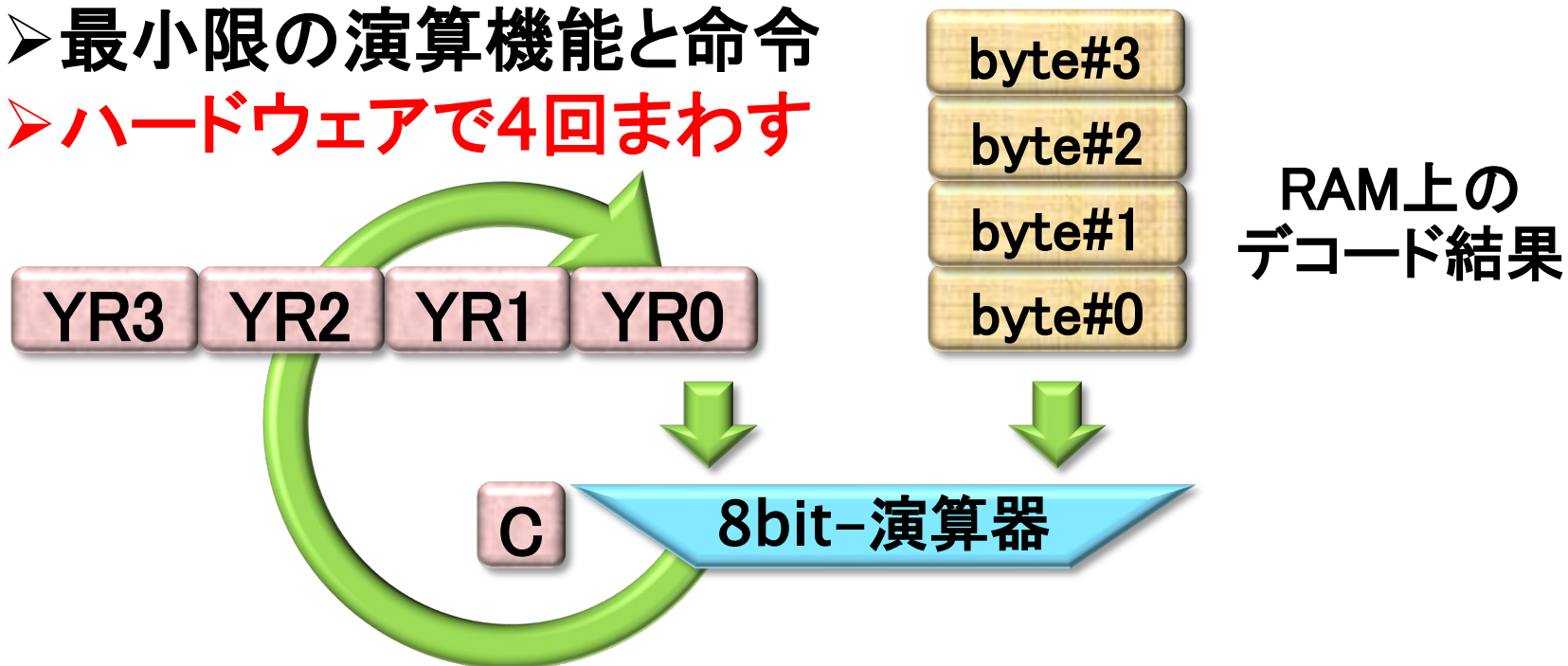
ゲストレジスタ参照機能

➤ レジスタ番号指定レジスタ(RGN)の利用



ゲスト演算機能(加減算と論理演算)

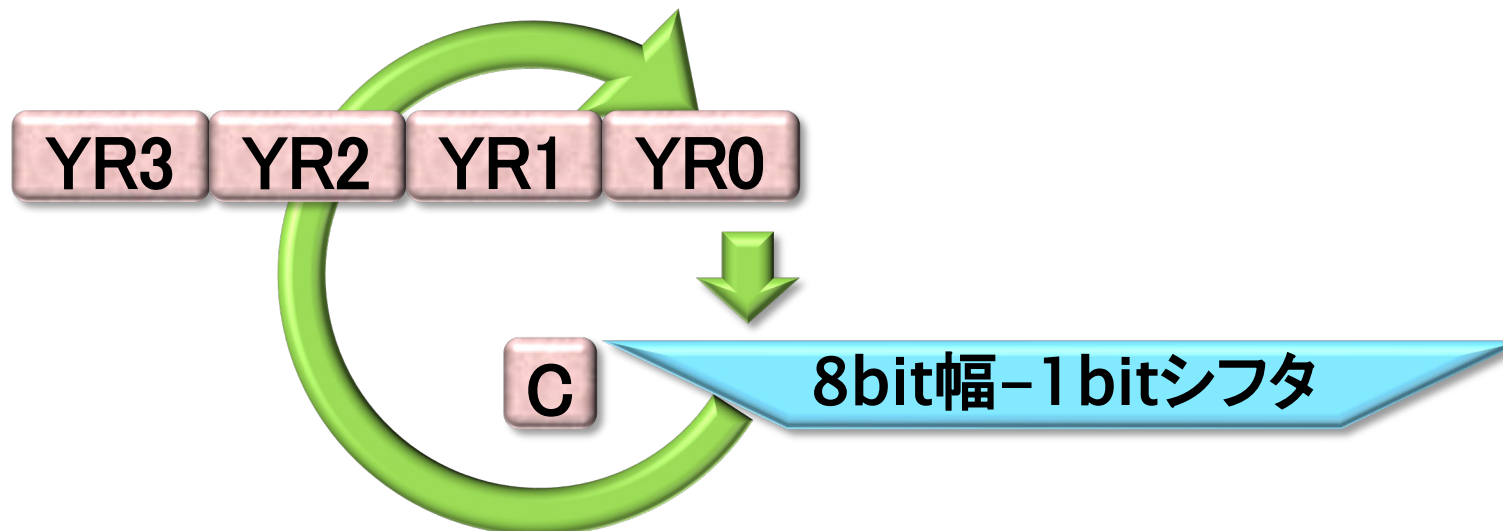
- 最小限の演算機能と命令
- ハードウェアで4回まわす



Addyoy, Subyoy, Rsbyoy (加減算)
 Adcyoy, Sbcyoy, Rscyoy (with Carry)
 Andyoy, Oryoy, Xoryoy (論理演算)
 ⇒ 4サイクルと仮定

ゲスト演算機能(シフト)

- シフトも最小限 (Vacated-bitは1 bitのみ)
- **ハードウェアで4回まわす**



Lsry1y, Asry1y (右1 bitシフト)

Rory1y, Rory0y (右1 bitローテート)

Lsly1y, Lsly0y (左1 bitシフト)

⇒ 4サイクルと仮定

エミュレータの実際(冒頭部分)

LOOP:

```
ldo0w ARM_STAT;
cmpwow 0x00;
brnz LOOP;
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Fetch ARM-insn
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
ldoy ARM_NPC;
ldyx;
addyoy CONST_FOUR;
stoy ARM_NPC;
addyoy CONST_FOUR;
stoy ARM_PC;
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Check Condition
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
ldo3w ARM_CPSR;
tbx3h 0x00;
.byte C1_EQ;
.byte C1_NE;
.byte C1_CS;
.byte C1_CC;
.byte C1_MI;
.byte C1_PL;
.byte C1_VS;
```

命令フェッチとPC更新

YR ← [NPC]

XR ← [YR]

YR+4 → [NPC]

YR+8 → [PC]

4ビット条件コードをロードし、

命令先頭4ビットに基づき

16方向分岐

エミュレータの実際(最初のデコード)

```

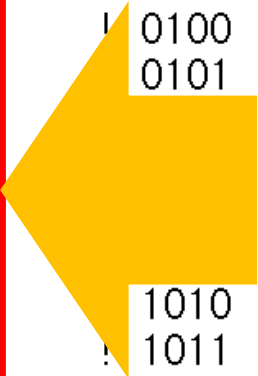
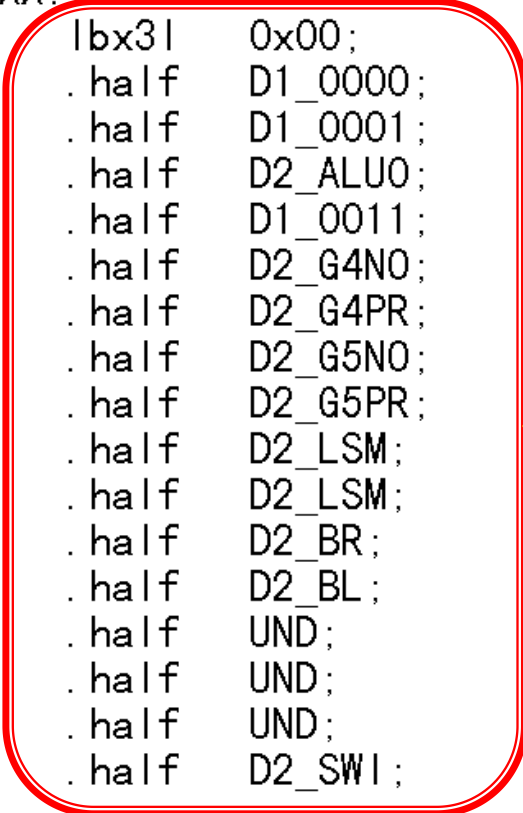
D1_XXXX:
    lbx3l    0x00;                /* 00000000:4bit-16way ... 16way-branch */
    .half   D1_0000;            ! 0000
    .half   D1_0001;            ! 0001
    .half   D2_ALU0;            ! 0010
    .half   D1_0011;            ! 0011 ! to find MSRO
    .half   D2_G4NO;            ! 0100
    .half   D2_G4PR;            ! 0101
    .half   D2_G5NO;
    .half   D2_G5PR;
    .half   D2_LSM;
    .half   D2_LSM;
    .half   D2_BR;              ! 1010
    .half   D2_BL;              ! 1011
    .half   UND;                ! 1100
    .half   UND;                ! 1101
    .half   UND;                ! 1110
    .half   D2_SWI;            ! 1111

UND:
    halt;

D1_0011:
    tbx2h   0x32;                /* 0011xxxx:4bit-eor+or ... 0011_1001 1001-
    .byte   D1_00110110;        ! xxxx
    .byte   D1_00110X10;        ! 0010

D1_00110110:

```



**主オペコードによる分類
届かないので
16bit-オフセット**

エミュレータの実際(詳細なデコード)

```

D2_LDST:
    mvx0ow 0x60;          /**** move sop0:1 sop1:2 sop2:1|rm -
    srlwow  1;           /* convert to opcd */
    tbx3l   0x21;        /* 0010xxxx:4bit-and+or check wb */
    .byte   D2_LDSTPRE1;
    .byte   D2_LDSTPRE0;
D2_LDSTPRE1:
    orwow   0x04;        /* pre=1 */
D2_LDSTPRE0:
    tbx2h   0x22;        /* 0010xxxx:4bit-and+or check wb */
    .byte   D2_LDSTWB1;
    .byte   D2_LDSTWB0;
D2_LDSTWB1:
    orwow   0x02;        /* wb=1 */
D2_LDSTWB0:
    sto2w   ARM_DECODE0; /**** store opcd:4 sop:1 pre:1 wb:1
    ldo3w   ARM_DECODE0; /**** load cond:4 updt:1 type:3 -> w
    tbx2h   0x21;        /* 0010xxxx:4bit-and+or check sls */
    .byte   D2_LDSTSLS1;
    .byte   D2_LDSTSLS0;
D2_LDSTSLS0:
    orwow   0x01;        /* type=3 ST */
D2_LDSTSLS1:
  
```

! pre=1
! pre=0

**ロードストアの
詳細デコード**

! wb=1
! wb=0

! sls=1
! sls=0

エミュレータの実際(演算)

EX_ADD:

```
mvx0ow 0x0f;
call get_r_rgn;
```

XR0の下位4ビットをWRへ

```
ldny;
addyoy ARM_DECSR3;
```

WR→RGN変換しYRへロード+加算

```
lbr EX_ALU_ARITH_WRITE_CC_REG;
```

EX_ADC:

```
mvx0ow 0x0f;
call get_r_rgn;
```

XR0の下位4ビットをWRへ

```
ldny;
ldo3w ARM_CPSR;
srlwow 5;
adcyyoy ARM_DECSR3;
```

WR→RGN変換しYRへロード

CARRYをロード+加算

```
lbr EX_ALU_ARITH_WRITE_CC_REG;
```

EX_SBC:

```
mvx0ow 0x0f;
call get_r_rgn;
```

同様に減算

```
ldny;
ldo3w ARM_CPSR;
srlwow 5;
sbcyyoy ARM_DECSR3;
```

```
lbr EX_ALU_ARITH_WRITE_CC_REG;
```

③ エミュレータの評価

➤ 目的は、

- ◆ エミュレータサイズの検証
- ◆ 性能予測
- ◆ 命令セットの妥当性
- ◆ 効率化のヒント

➤ Mibench

➤ uClinux (4MB nommu)

- ◆ Binutils-2.17
- ◆ Elf2flt-20050308
- ◆ Genromfs-0.5.1
- ◆ Gcc-4.4.1
- ◆ -march=armv4 -msoft-float -O2

```
Linux version 2.4.32-uc0 (nakashin@arch00) (gcc バージョン 4.1.1) #2 2012年 2月2
Processor: Atmel AT91M4000x revision 0
Architecture: EB01
Ignoring unrecognized tag 0x54410001
Ignoring unrecognized tag 0x54410002
On node 0 totalpages: 1024
zone(0): 0 pages.
zone(1): 1024 pages.
zone(2): 0 pages.
Kernel command line: root=/dev/ram0
Calibrating delay loop... 15.82 BogoMIPS
Memory: 4MB = 4MB total
Memory: 3412KB available (438K code, 153K data, 36K init)
Dentry cache hash table entries: 512 (order: 0, 4096 bytes)
Inode cache hash table entries: 512 (order: 0, 4096 bytes)
Mount cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 1024 (order: 0, 4096 bytes)
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Starting kswapd
Atmel USART driver version 0.99
ttyS0 at 0xffffd0000 (irq = 2) is a builtin Atmel APB USART
ttyS1 at 0xffffc0000 (irq = 3) is a builtin Atmel APB USART
OROCHI-FRV attached (irq = 7) svc_opcd at f004000 svc_gr4-7 at f004010-f00401c
Bikmem copyright 1998,1999 D. Jeff Dionne
Bikmem copyright 1998 Kenneth Albanowski
Bikmem 1 disk images:
0: 1400000-14AE7FF [VIRTUAL 1400000-14AE7FF] (RD) <ROMFS>
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
VFS: Mounted root (romfs filesystem) readonly.
Shell invoked to run file: /etc/rc
Command: hostname OROCHI
Command: /bin/expand /etc/ramfs.img /dev/ram0
Command: mount -t proc proc /proc
Command: mount -t ext2 /dev/ram0 /var
Command: mkdir /var/tmp
Command: mkdir /var/log
Command: mkdir /var/run
Command: mkdir /var/lock
Command: mkdir /var/empty
Command: cat /etc/motd
Welcome to

FCOMP support by <nakashin@is.naist.jp>
Execution Finished, Exiting

Sash command shell (version 1.1.1)
/>
```

静的ステップ数からわかったこと

- **Table Jumpが20%**
- **1 byteオフセットで概ね届く**
- **16→2方向縮退機能
アドレス数平均=5
バイト数平均=6.5**
- **サイズ<4KB
PCが12ビットで済む**

機能	Table Jump の各アドレス込み命令数
基本デコード	190
分岐命令	131
単純ロード/ストア デコード/実行	141/208
マルチロード/ストア デコード/実行	55/157
一般演算 デコード/実行	121/156
乗算 デコード/実行	23/173
ARM レジスタ番号変換	111
32bit 幅 1bit シフト	119
uClinux 対応制御レジスタ	183
uClinux 対応 I/O 空間	230
合計命令数	1998
うち Table Jump 命令数	86
うち Table Jump アドレス数	438
合計 byte 数	3492
うち Table Jump アドレス部分	566

動的ステップ数/サイクル数からわかったこと

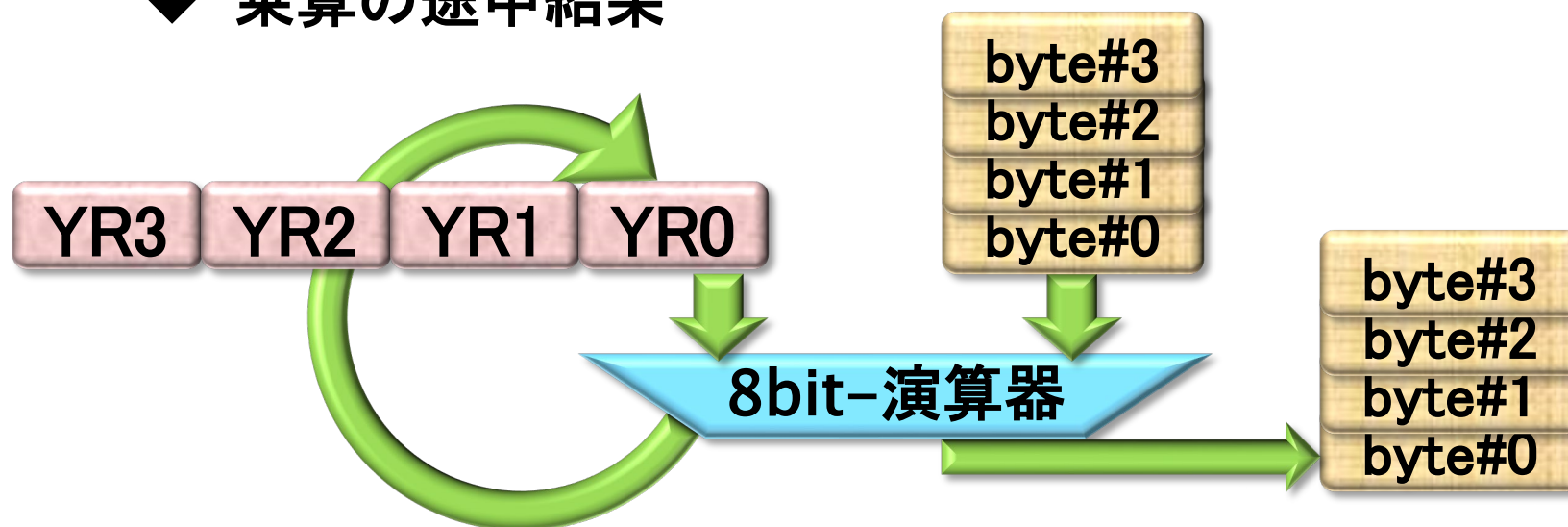
- ARM 1命令 ⇒ RISC 1.7命令
- RISC 1命令 ⇒ HOST 38命令、64サイクル
- **ARM 1命令に対して、HOST 109サイクル**
(参考 IBM370 ⇒ SPARC-V8の命令数比
JITは1:10、インタプリタは1:100)

【動的ステップ数の内訳】

- LD/ST:25% 分岐:20%
- Table-Jump、Reg間転送、8bit演算:各15%
- 残り32bit演算等
- **単体では、32bitストアと条件分岐が各5%**

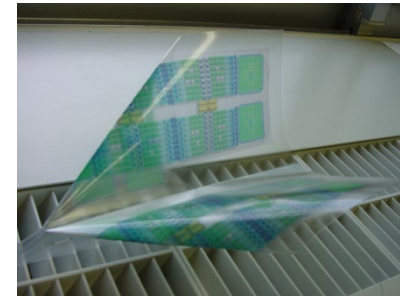
サイクル数からわかったこと

- YR3-0のロード、ストア、加算、1 bitシフトが上位に
 - ◆ 予定通りハードで4回まわす作戦で、よさそう
 - ◆ 1 bitシフトも他と同程度で多くない。もろみ通り
- 2命令連続では、**addyoy + stoy** が6-10%
 - ◆ ゲストPC更新(+4が多い)
 - ◆ ロード/ストアのベースレジスタ更新(+4が多い)
 - ◆ 乗算の途中結果



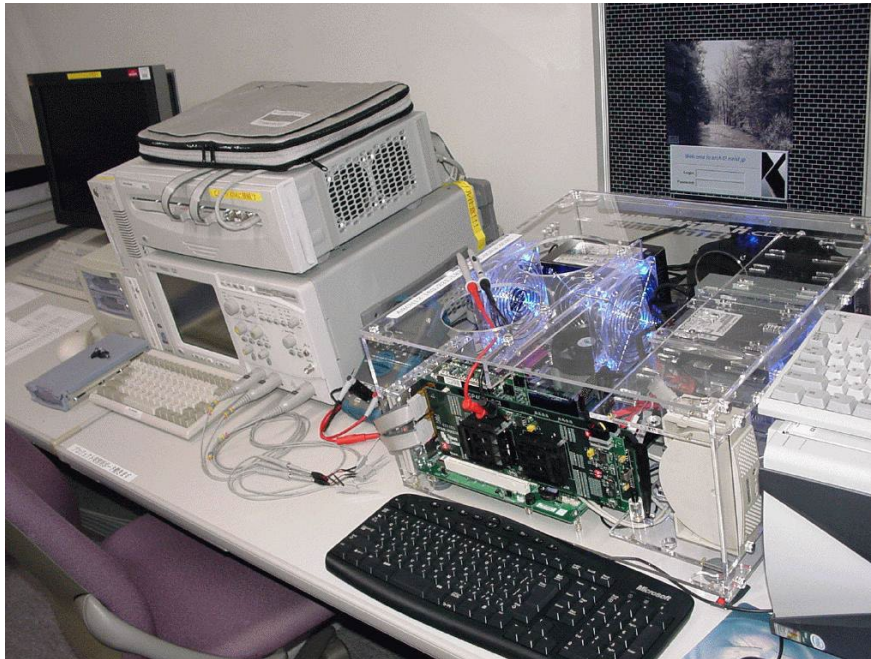
まとめ

- Table Jumpがstaticの20%
 - ◆4KB以下で書ける
- ハードで4回まわす32bit演算でよさそう
 - ◆1bitシフトで十分
- 連続命令に最適化のヒント
 - ◆4byte加算+ストアが6-10%
- 今後の予定
 - ◆ 既存8bit-CPUと比較
 - ◆ ハードの設計と評価
 - ◆ 高信頼化高位合成のネタに使う
 - ◆ いずれ制作 できるといいなあ



Instructions:88,34pins,Addr/Data 24/8bit

in RST,CLK,C_XINT
out C_XACK
out M_XREQ,M_ADSX,M_WEX,M_BSTMX
in M_XGNT,M_RRDY
io[23:0] M_AD



```
Linux version 2.4.32-uc0 (nakashim@arch00) (gcc バージョン 4.1.1) #2 2012年 2月2
Processor: Atmel AT91M40xxx revision 0
Architecture: EB01
Ignoring unrecognized tag 0x54410001
Ignoring unrecognized tag 0x54410002
On node 0 totalpages: 1024
zone(0): 0 pages.
zone(1): 1024 pages.
zone(2): 0 pages.
Kernel command line: root=/dev/rom0
Calibrating delay loop... 15.82 BogomIPS
Memory: 4MB = 4MB total
Memory: 3412KB available (438K code, 153K data, 36K init)
Dentry cache hash table entries: 512 (order: 0, 4096 bytes)
Inode cache hash table entries: 512 (order: 0, 4096 bytes)
Mount cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 1024 (order: 0, 4096 bytes)
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Starting kswapd
Atmel USART driver version 0.99
ttyS0 at 0xffffd000 (irq = 2) is a builtin Atmel APB USART
ttyS1 at 0xffffcc00 (irq = 3) is a builtin Atmel APB USART
OROCHI-FRW attached (irq = 7) svc_opcd at f004000 svc_gr4-7 at f004010-f00401c
Bikmem copyright 1998,1999 D. Jeff Dionne
Bikmem copyright 1998 Kenneth Albanowski
Bikmem 1 disk images:
0: 1400000-14AE7FF [VIRTUAL 1400000-14AE7FF] (RO) <ROMFS>
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
VFS: Mounted root (romfs filesystem) readonly.
Shell invoked to run file: /etc/rc
Command: hostname OROCHI
Command: /bin/expand /etc/ramfs.img /dev/ram0
Command: mount -t proc proc /proc
Command: mount -t ext2 /dev/ram0 /var
Command: mkdir /var/tmp
Command: mkdir /var/log
Command: mkdir /var/run
Command: mkdir /var/lock
Command: mkdir /var/empty
Command: cat /etc/motd
Welcome to

      GNU
    GNU/Linux
  GNU/Linux
GNU/Linux

FCOMP support by <nakashim@is.naist.jp>
Execution Finished, Exiting

Sash command shell (version 1.1.1)
/>
```

8bitハードウェアで32bitOSを効率よく動かす 開発途上デバイス向け超小型アーキテクチャ

INV, 2in-NAND, NOR, XOR, FF

Total: 4300 cells

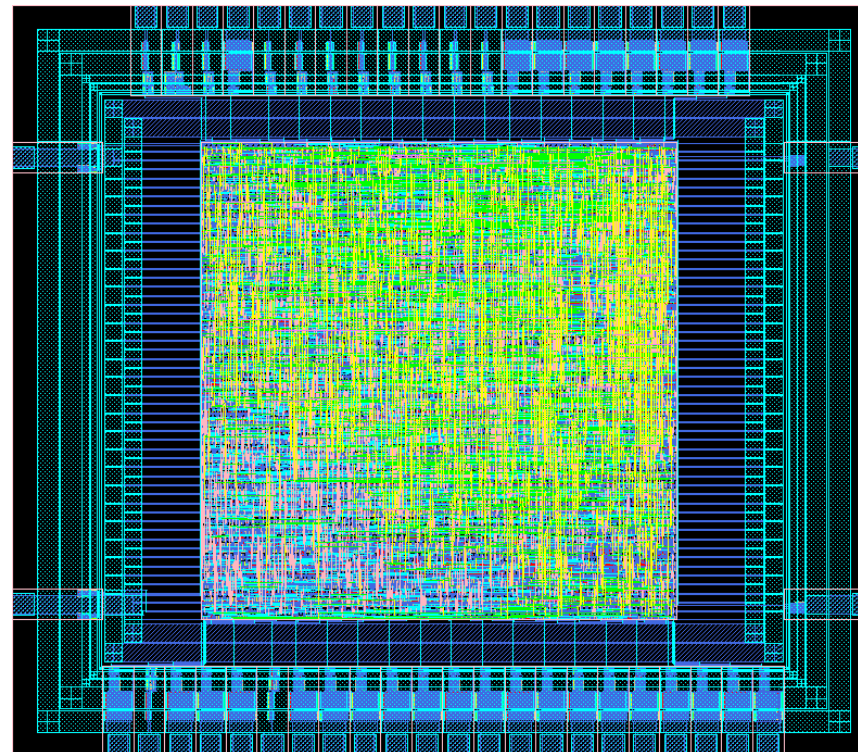
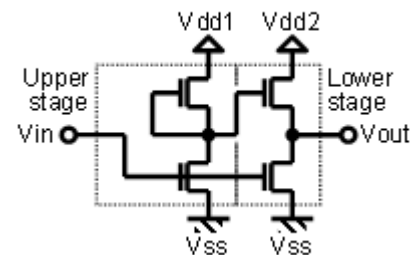
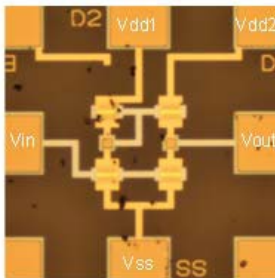
Tr#...21000

0.35 μ -5layers 0.78mm \square , 2layers 0.96mm \square

10 μ Inkjet Printer:

2layers 27mm \square

IGZO will be available

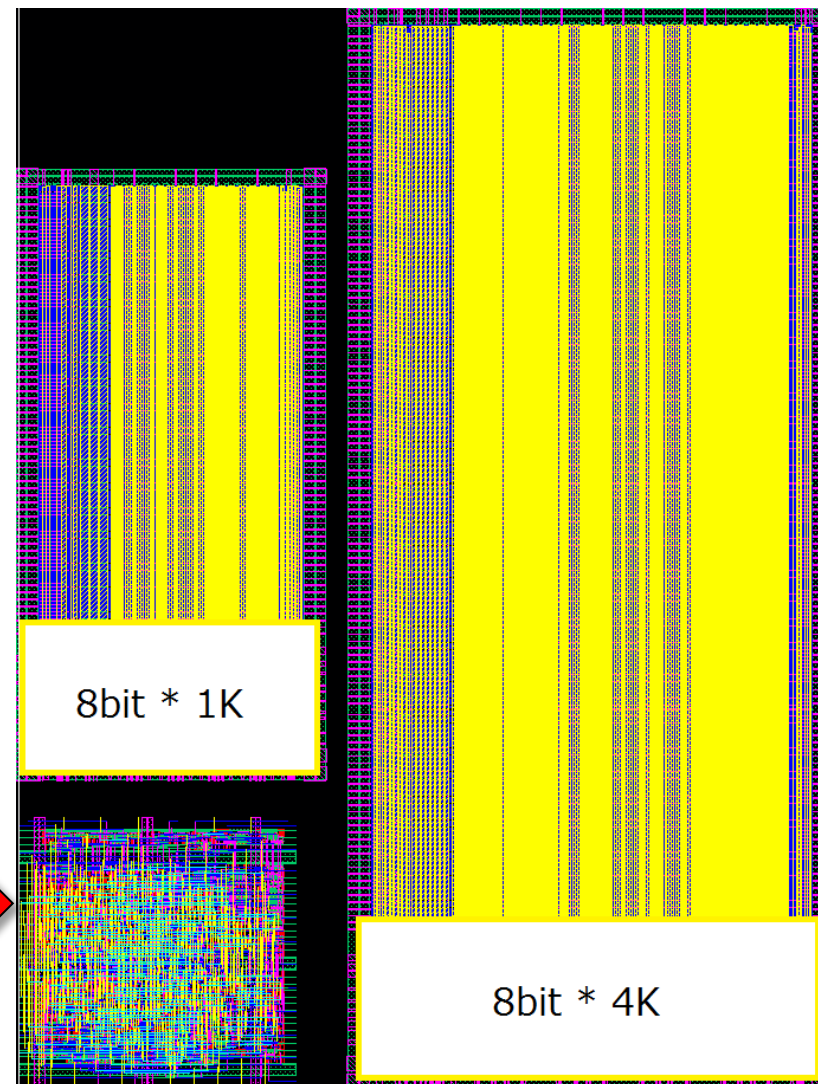


Emulator oriented Minimal Instruction set CPU EMIN

Instructions: 88
PC,LR(12bit),WR,CC
XR3-0,YR3-0
0.18μ...0.24mm□
2in-NAND...2643
Tr#...10000

Instructions: 55
PC(10bit),TR3-0,CC
0.18μ...0.20mm□
2in-NAND...1748
Tr#...7000(<M6502)

ARM7TDMI: 74000Tr



予備

