



EDA Technofair 2000

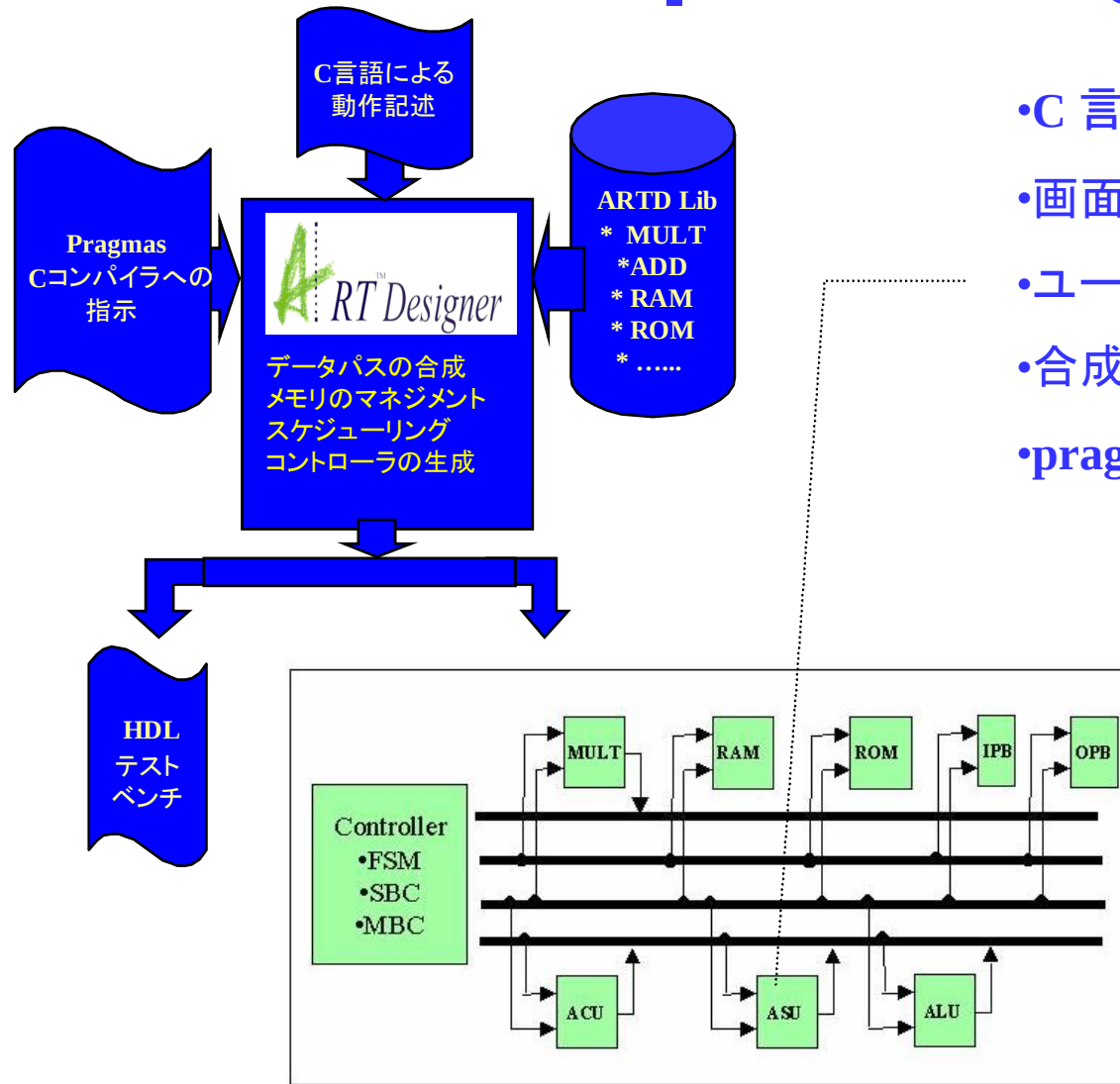
丸文株式会社
開発営業第2部
連絡先: TEL; 03-3639-9640
imamura@marubun.co.jp



A|RT Designer: From C to HDL

丸文株式会社
開発営業第2部
連絡先: TEL; 03-3639-9640
imamura@marubun.co.jp

A|RT Designer



- C 言語記述を動作合成する最新ツール
- 画面を通して対話的に最適化を探索
- ユーザ設計のブロックが指定可能
- 合成可能なVHDL/Verilogを生成
- pragmaによって機能の指定が可能

NEW !

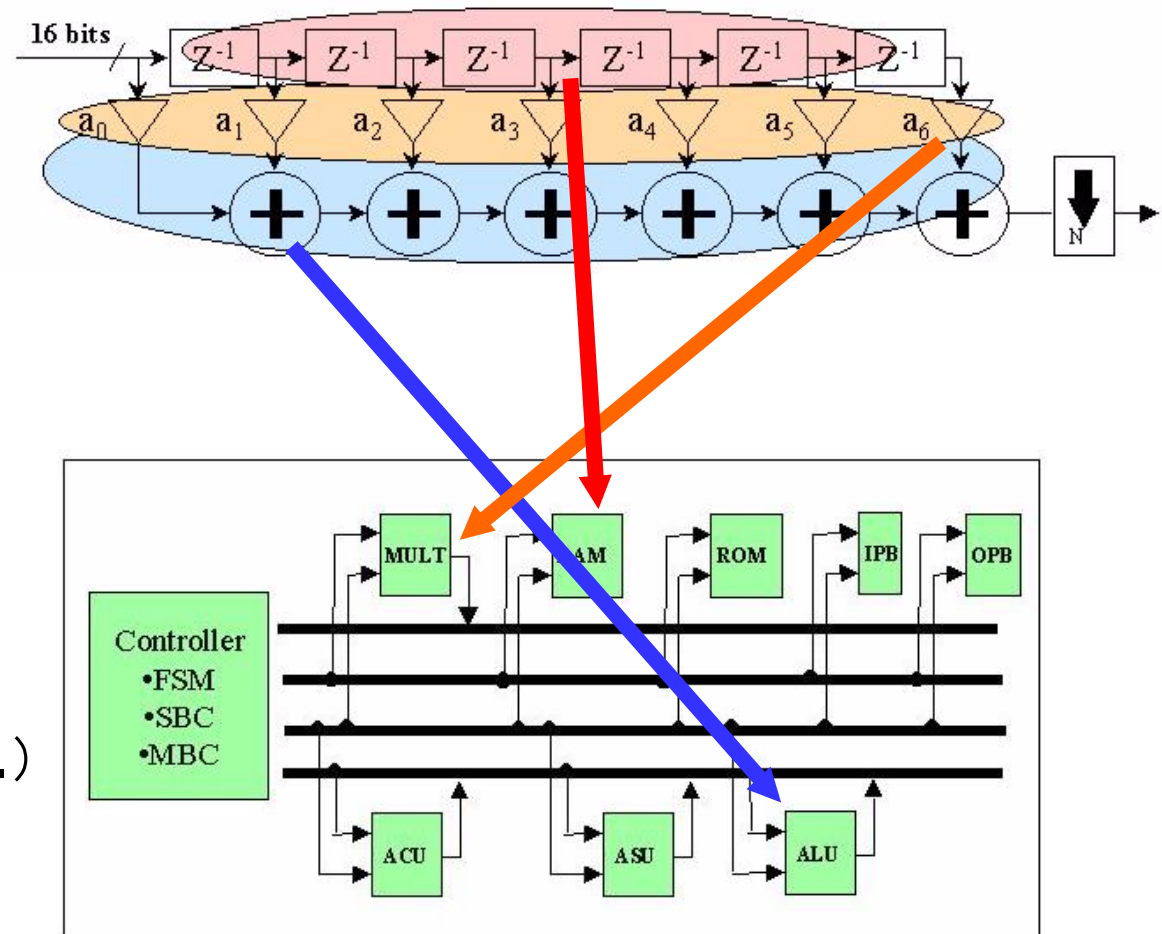
動作記述からアーキテクチャを生成

● ビヘイビア/アルゴリズム

- 動作, 機能の記述
- 入出力信号、サンプルレート

● アーキテクチャの生成

- コントローラ
- データパス
- リソース
 - (ALU, ROM, RAM...)
- クロック



最適なアーキテクチャを合成

システムの動作記述

資源の配置

資源の割当

スケジューリング

アーキテクチャの“RTL”
(合成可能)

論理合成へ

アーキテクチャ合成

動作合成

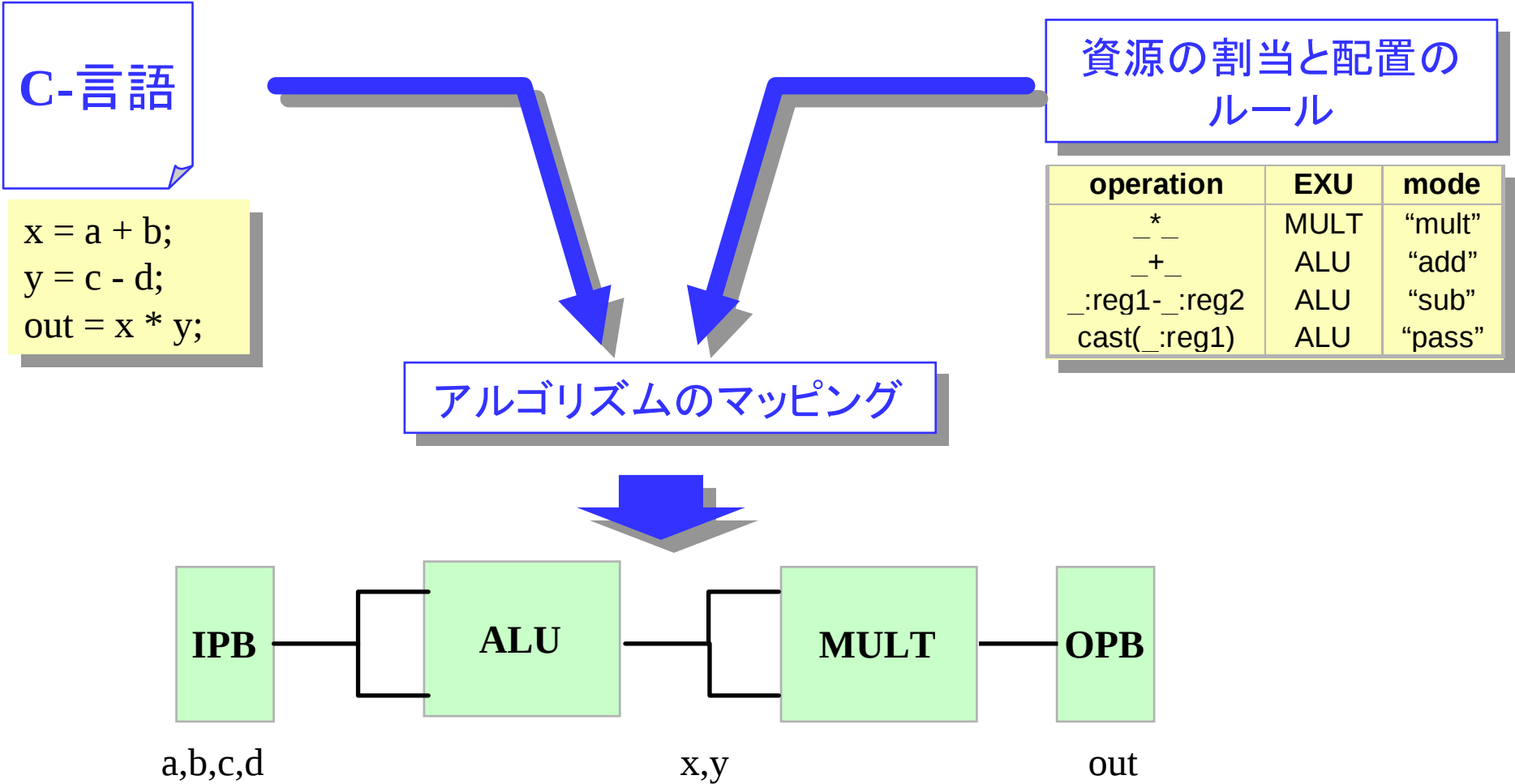
- ・自動配置
- ・自動スケジューリング
- ・データパスの自動生成
- ・制御部の自動生成
- ・HDL-RTLの自動生成

+

アーキテクチャの最適化

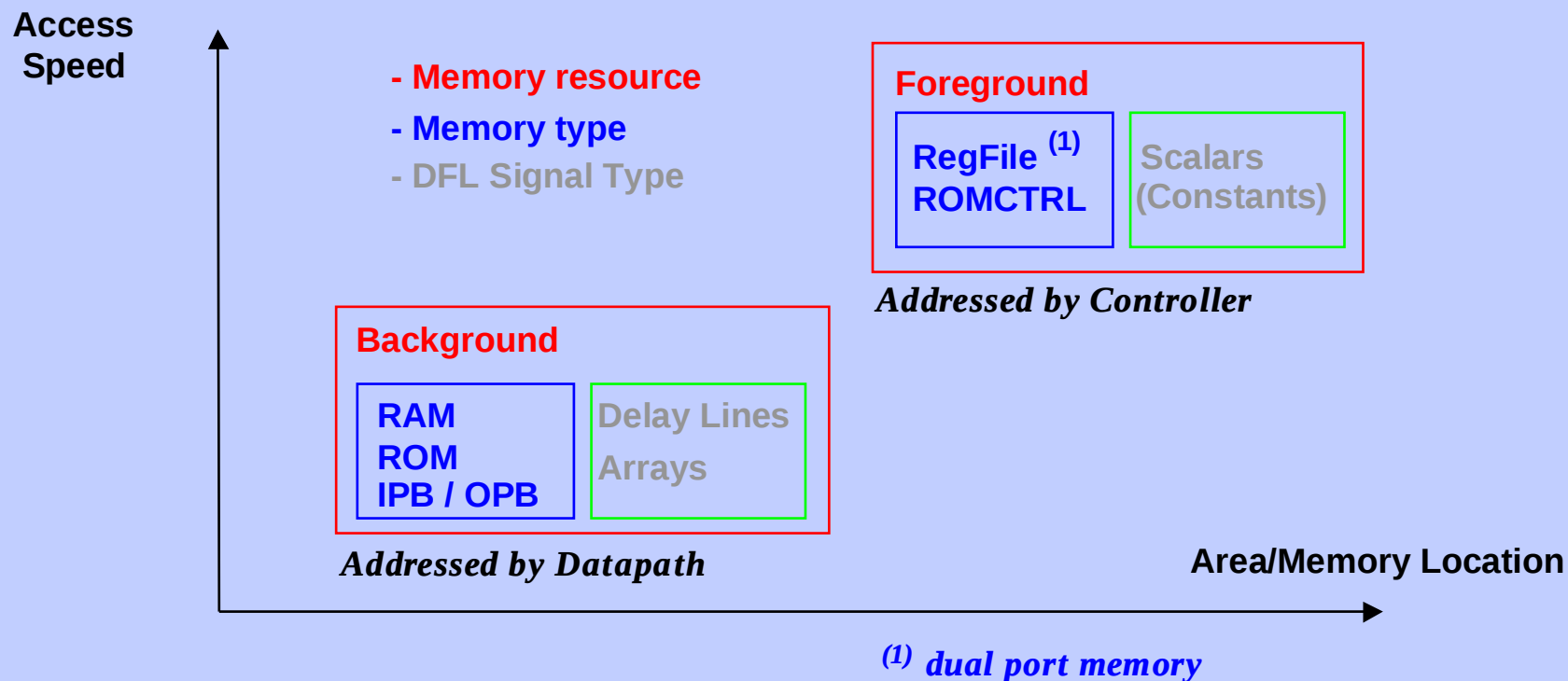
- ・性能/スピード
- ・面積/コスト
- ・消費電力

資源の配置と割当

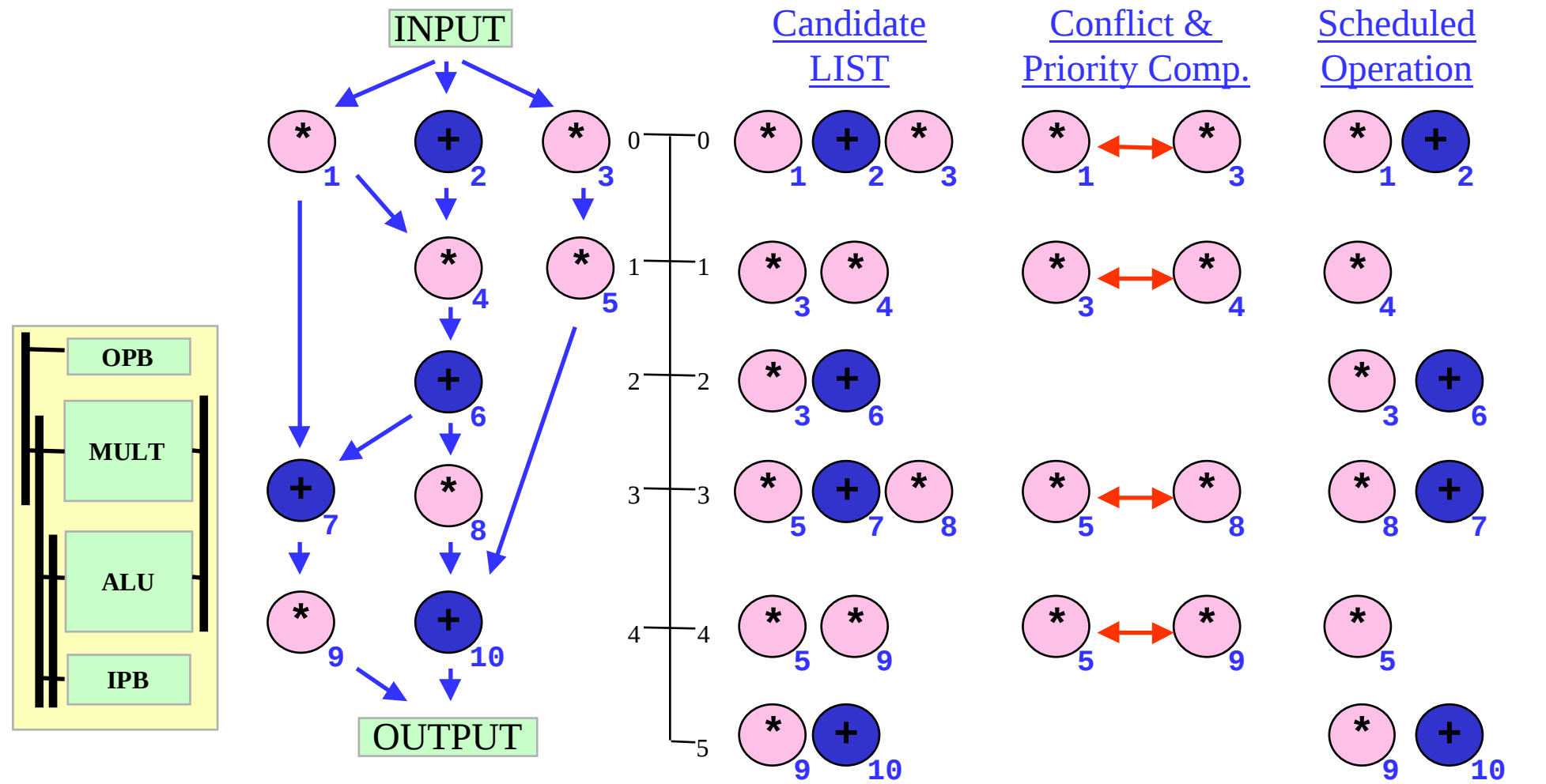


メモリアーキテクチャの生成

Available Memory Resources and Types.



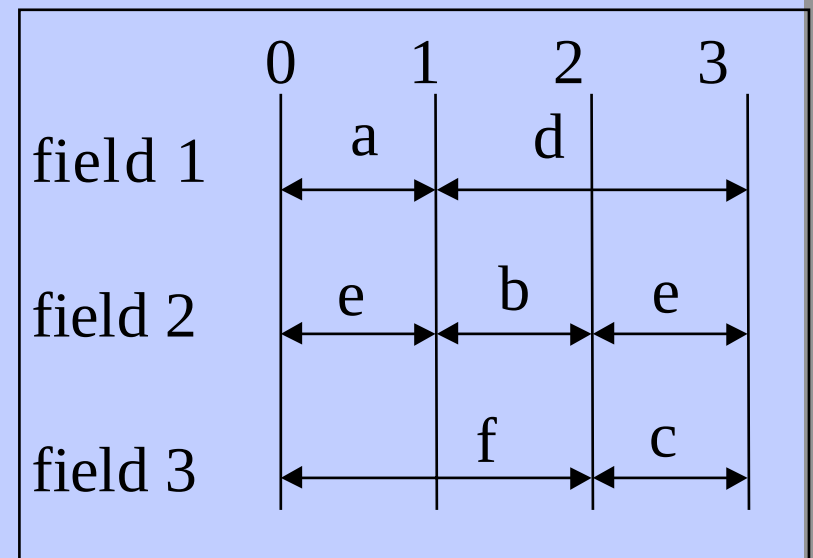
スケジューリング



スケジューリング : レジスタの共有化

Register Assignment & Register Optimization.

variable	lifetime	conflict graph
a	[0,1]	
b	[1,2]	
c	[2,3]	
d	[1,3]	
e	[0,1] U [2,3]	
f	[0,2]	



スケジューリング : Loop Folding

- **Faster Schedule Through more Parallelism**
- **MicroROM Area Increases (#potentials)**
- **Size of Register Files Increases**
- **Example**

```
pot 0    s = 0;  
          FOR i = 1 to 10 {  
pot 1      p[i] = c[i]*in[i];  
pot 2      s = s+p[i];  
          }
```

21 cycles

Loop Folding

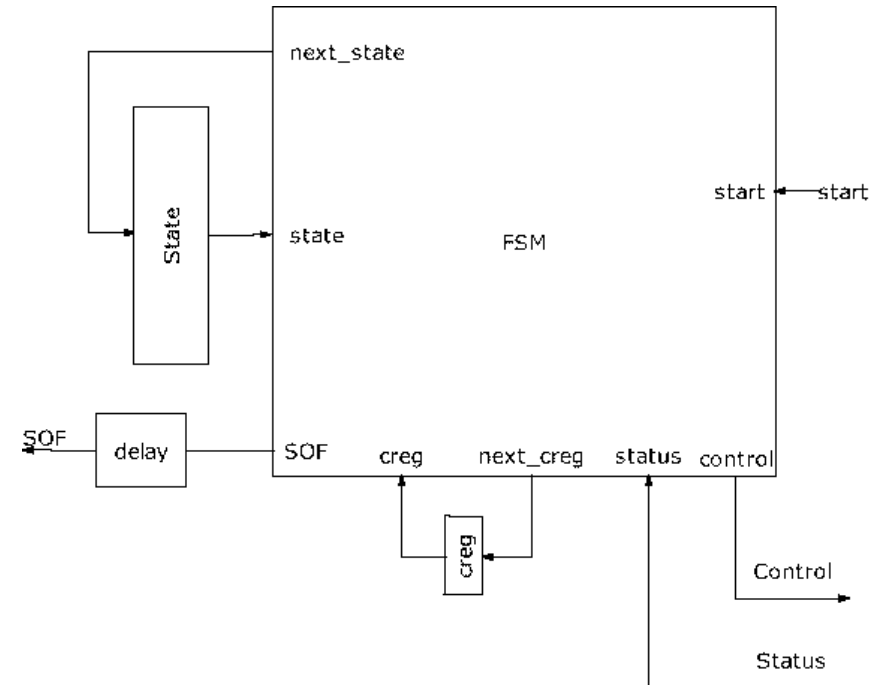
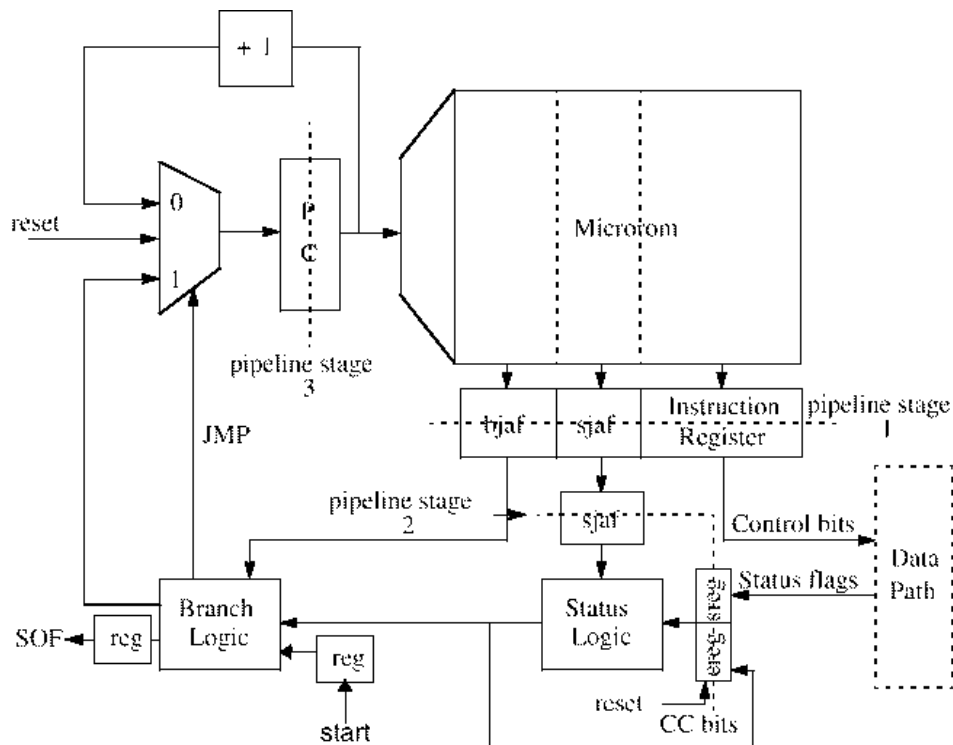
```
pot 0    s = 0;  
pot 1    p[1] = c[1]*in[1];  
          FOR i = 2 to 10 {  
pot 2      s = s+p[i-1]; p[i] = c[i]*in[i];  
          }
```

12 cycles



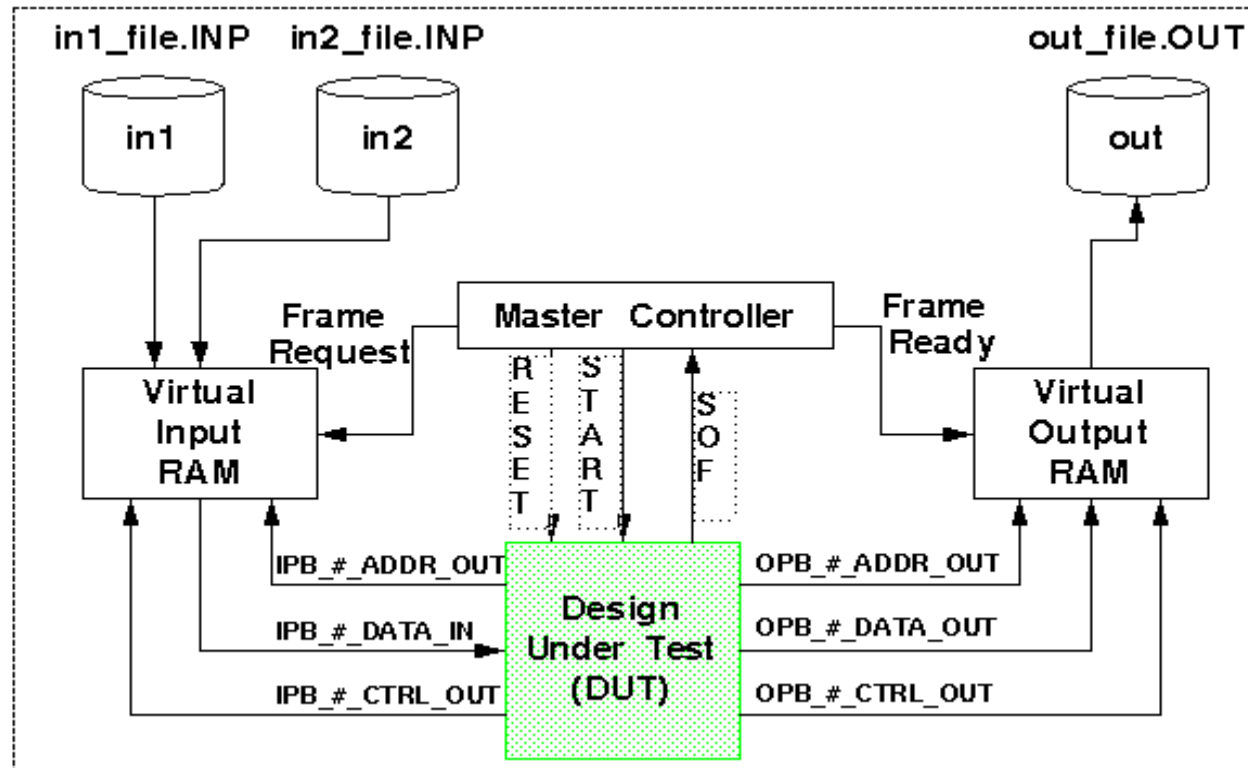
コントローラの生成

- User can choose from 6 possible controller architectures.



RTL-HDL(ネットリスト)の生成

- Synthesizable VHDL or Verilog
- Processor Netlist
- Testbench



対話型設計と最適化の探求

The screenshot displays the A|RT Designer software interface, which includes several key components:

- 基本ユーザインタフェース (Basic User Interface):** Located at the top left, it features a menu bar (Project, Run, Options, Pragmas, Reports, Views, Hdl, Misc) and a toolbar with icons for New, Open, Import, Save, Editor, Compile, Build, and Stop. Below these are tabs for Arch, Flg, Mem, Mux, Types, and Sched. A 'Run' button is also present.
- Cのソースエディタ (C Source Editor):** A central window showing C code for a project named 'marovc/iterate'. The code includes variable declarations, a loop, and conditional logic for signal processing.
- VHDL/Verilog 生成 (VHDL/Verilog Generation):** A window on the right showing the generated VHDL code, which mirrors the logic of the C source code.
- ハードウェア資源の動作状況、バスの負荷状況など (Hardware Resource Operation Status, Bus Load Status, etc.):** A window at the bottom left displaying a Gantt chart and a table of resource usage. The table lists resources like ram_1, romctrl_1, rom_1, opb_1, ipb_1, mult_1, acu_1, and alu_1, along with their usage counts.
- メモリやレジスタ内の変数の寿命 (Lifetime of Variables in Memory or Registers):** A window at the bottom right showing a detailed view of the program counter and variable lifetimes.

Cのソースエディタ

基本ユーザインタフェース
配置、割当、スケジューリング

VHDL/Verilog 生成

ハードウェア資源の動作状況、
バスの負荷状況など

メモリやレジスタ内の変数の寿命

A|RT Designerの特徴

- **C-based behavioral synthesis tool. Offers true “Algorithm to RTL” solution**
- **No ‘blackbox’ tool. By using pragmas the user has full control over:**
 - datapath composition (arithmetic, memory, IO, address / databusses..)
 - operation assignment
 - scheduling & register generation
 - controller design : choose from 6 possible controllers
- **Very fast. Facilitates extensive “what-if” analysis at the architecture level.**



A|RT Designer の特徴(2)

- **Allows to synthesize a customized datapath & controller for an algorithm specified in C**
- **ART Designer generates an algorithm specific processor with a user defined instruction set.**
 - The ultimate in flexible instruction set processors
 - Beyond approaches such as ARC, Tensilica, etc..
- **ART Designer is an open system. Allows to add new instructions and datapath elements to the library**
- **Runs on PC/NT and Workstation**



A|RT Designer の応用例

- Design of co-processors for compute intensive tasks in a SOC design
- Implement high performance algorithms on FPGA
- Ultra low power DSP ASIC Design
- Migration of DSP processor based solutions to high performance FPGA solutions.
- Reconfigurable computing



Frontier

↑↑↑↑↑ *Design*

<http://www.frontierd.com>

A TM *RT Designer*

- 応用例: **FM demodulator** -

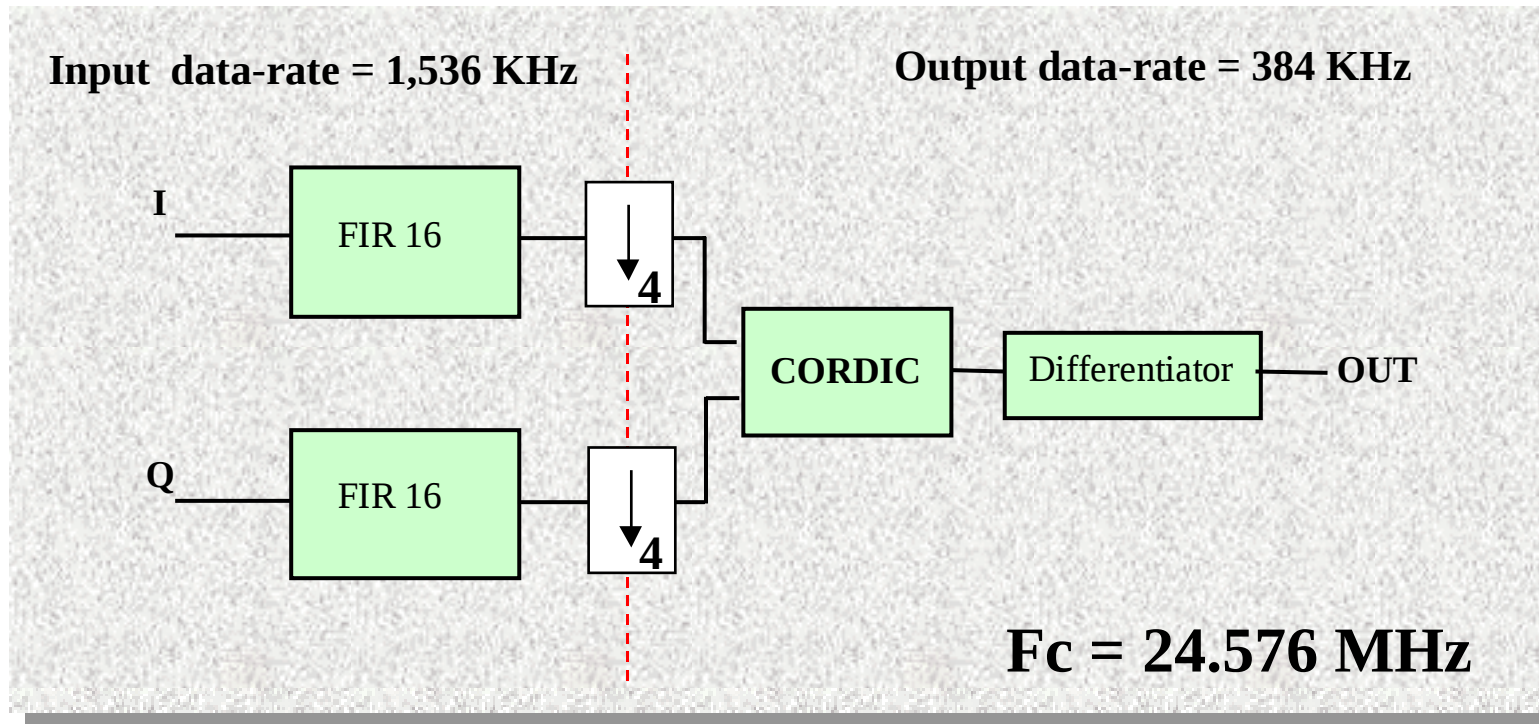
丸文株式会社

開発営業第2部

連絡先: TEL; 03-3639-9640

imamura@marubun.co.jp

FM復調器のアルゴリズム

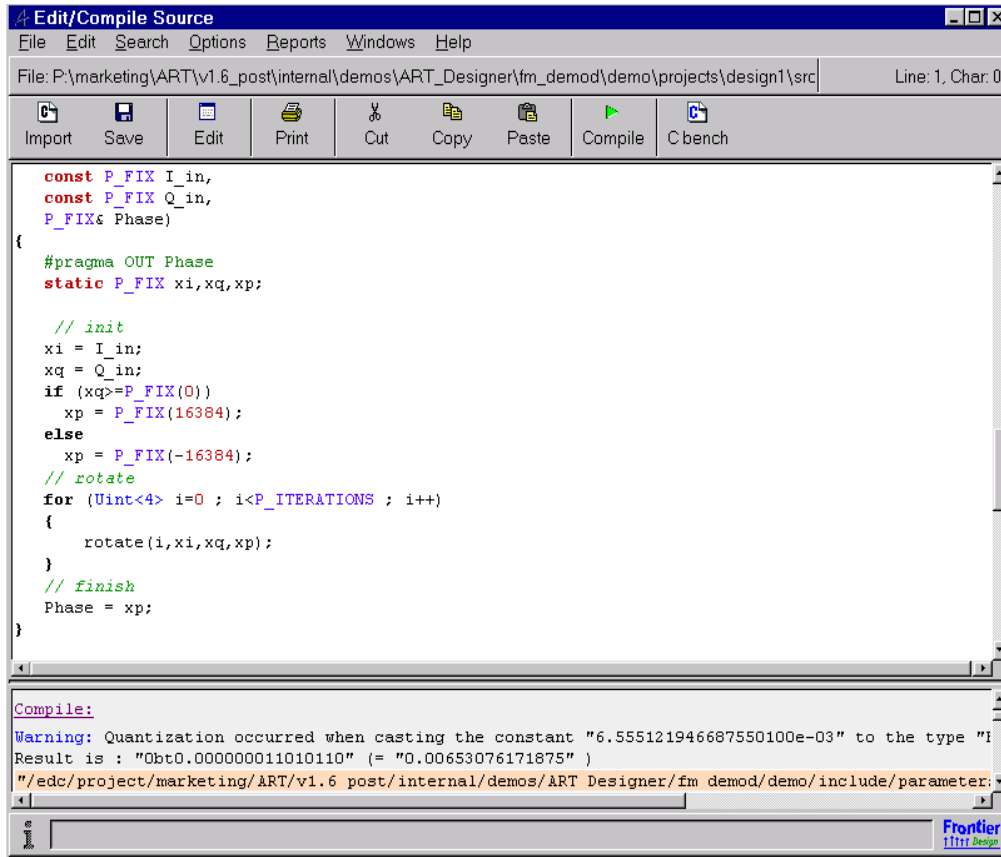


$$\frac{24.576 \text{ MHz}}{384 \text{ KHz}} \rightarrow \text{設計目標}$$

64 cycles to compute every output

Case Study

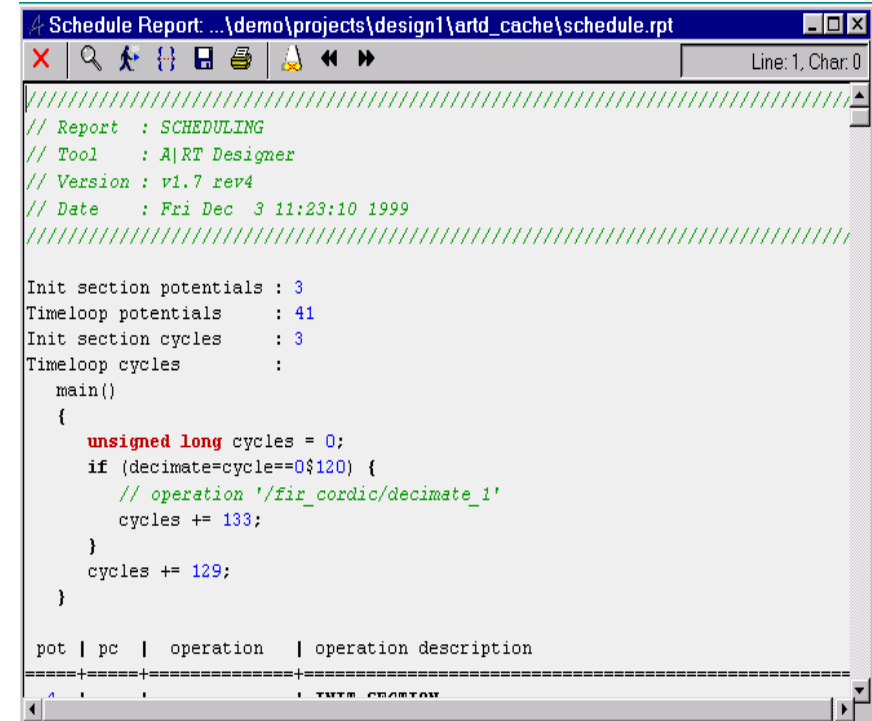
デフォルト設定での結果



```
const P_FIX I_in,
const P_FIX Q_in,
P_FIX& Phase)
{
    #pragma OUT Phase
    static P_FIX xi,xq,xp;

    // init
    xi = I_in;
    xq = Q_in;
    if (xq>P_FIX(0))
        xp = P_FIX(16384);
    else
        xp = P_FIX(-16384);
    // rotate
    for (UInt<4> i=0 ; i<P_ITERATIONS ; i++)
    {
        rotate(i,xi,xq,xp);
    }
    // finish
    Phase = xp;
}

Compile:
Warning: Quantization occurred when casting the constant "6.555121946687550100e-03" to the type "I
Result is : "0bt0.000000011010110" (= "0.00653076171875" )
"/edc/project/marketing/ART/v1.6 post/internal/demos/ART_Designer/fm_demo/demo/include/parameter:
```



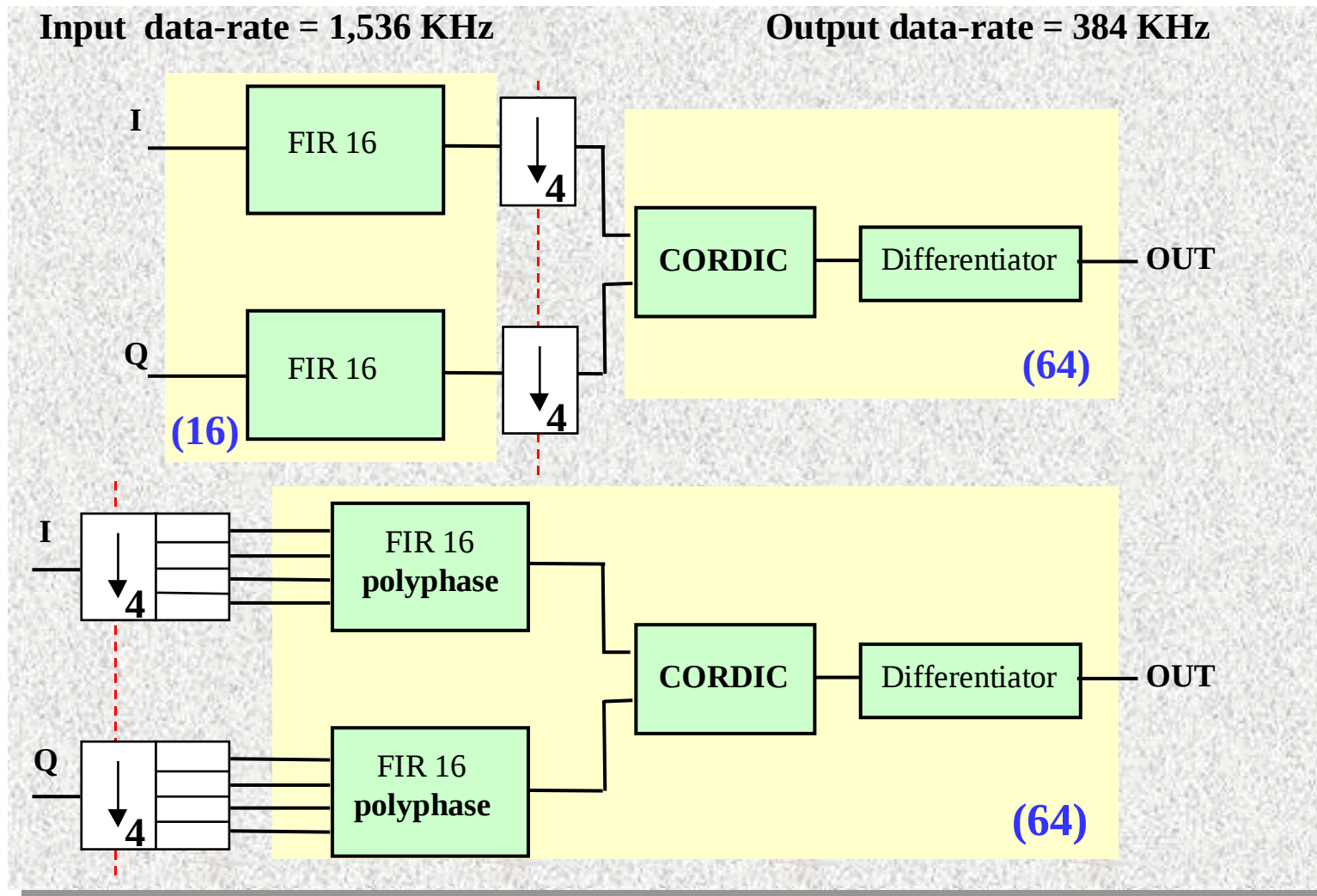
```
// Report : SCHEDULING
// Tool : A|RT Designer
// Version : v1.7 rev4
// Date : Fri Dec 3 11:23:10 1999

Init section potentials : 3
Timeloop potentials : 41
Init section cycles : 3
Timeloop cycles :
    main()
    {
        unsigned long cycles = 0;
        if (decimate=cycle==0$120) {
            // operation '/fir_cordic/decimate_1'
            cycles += 133;
        }
        cycles += 129;
    }

pot | pc | operation | operation description
=====
INTL SECTION
```

129 cycles for filtering
133 cycles for cordic

アルゴリズムの変更



Case Study

アルゴリズムの変更 -ポリフェーズフィルタの使用結果-

```
diff_delay=diff_in;
return(Output);
}

void fir_cordic(
    const P_FIX I_in[4],
    const P_FIX Q_in[4],
    P_FIX& Output)
{
    #pragma OUT Output
    P_FIX I_out, Q_out, Phase;

    // compute I and Q filters
    I_out = fir16<0>(I_in);
    Q_out = fir16<1>(Q_in);
    // compute angle
    cordic(I_out, Q_out, Phase);
    Output=differentiator(Phase);
}
```

Compile:

Warning: Quantization occurred when casting the constant "6.555121946687550100e-03" to the type "I".
Result is : "0bt0.000000011010110" (= "0.00653076171875")
"/edc/project/marketing/ART/v1.6 post/internal/demos/ART Designer/fm demod/demo/include/parameter:

Report : SCHEDULING
Tool : ART Designer
Version : v1.7 rev4
Date : Tue Nov 30 16:00:09 1999

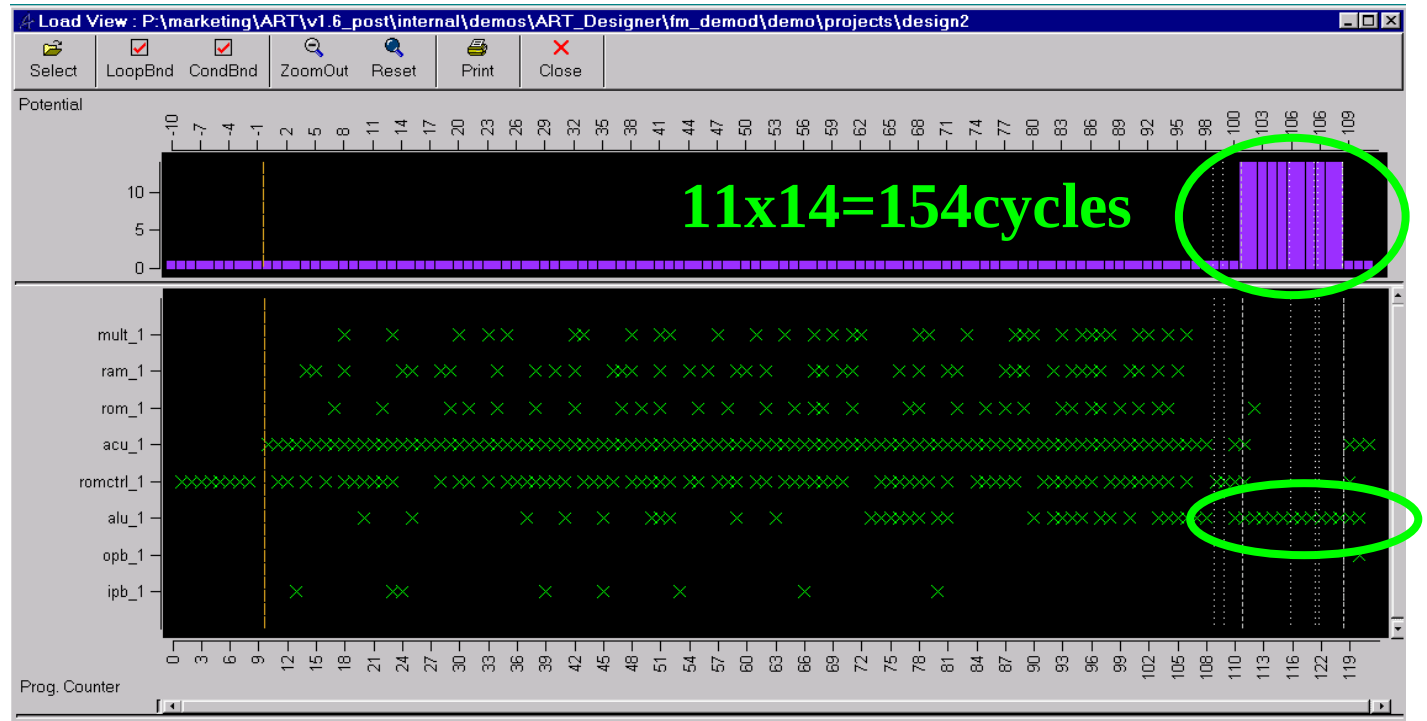
Init section potentials : 8
Time loop potentials : 112
Init section cycles : 8
Time loop cycles : 216

pot	pc	operation	operation description
-9			INIT SECTION
-9	1	RT391	BEGIN baseptr:reg_Xbus_acu_1 <- 'Int<6>(0)':romctrl_1 romctrl_1=const1[0];
-9	1	RT389	diff_delay:reg_Ybus_alu_1 <- 'Int<6>(0)':romctrl_1 romctrl_1=const1[0];

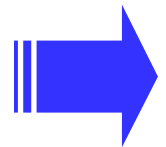
Case Study

専用ユニット(ASU)の割当

```
void rotate(  
  cons . P_INDEX iter,  
  P_FIX& xi,  
  P_FIX& xq,  
  P_FIX& xp)  
{  
  #define BOOL Uint<1>  
  
  BOOL xi_sign_bit,xq_sign_bit,different_sign;  
  P_FIX angle,xi_inc,xq_inc;  
  
  xi_inc = xq >> iter;  
  xq_inc = xi >> iter;  
  angle = lookup_angle_coeff[iter+1];  
  
  xi_sign_bit = BOOL(xi < P_FIX(0));  
  xq_sign_bit = BOOL(xq < P_FIX(0));  
  different_sign = xi_sign_bit ^ xq_sign_bit;  
  
  if (different_sign) {  
    xi = xi+xi_inc;  
    xq = xq-xq_inc;  
    xp = xp+angle;  
  } else {  
    xi = xi-xi_inc;  
    xq = xq+xq_inc;  
    xp = xp-angle;  
  }  
}
```

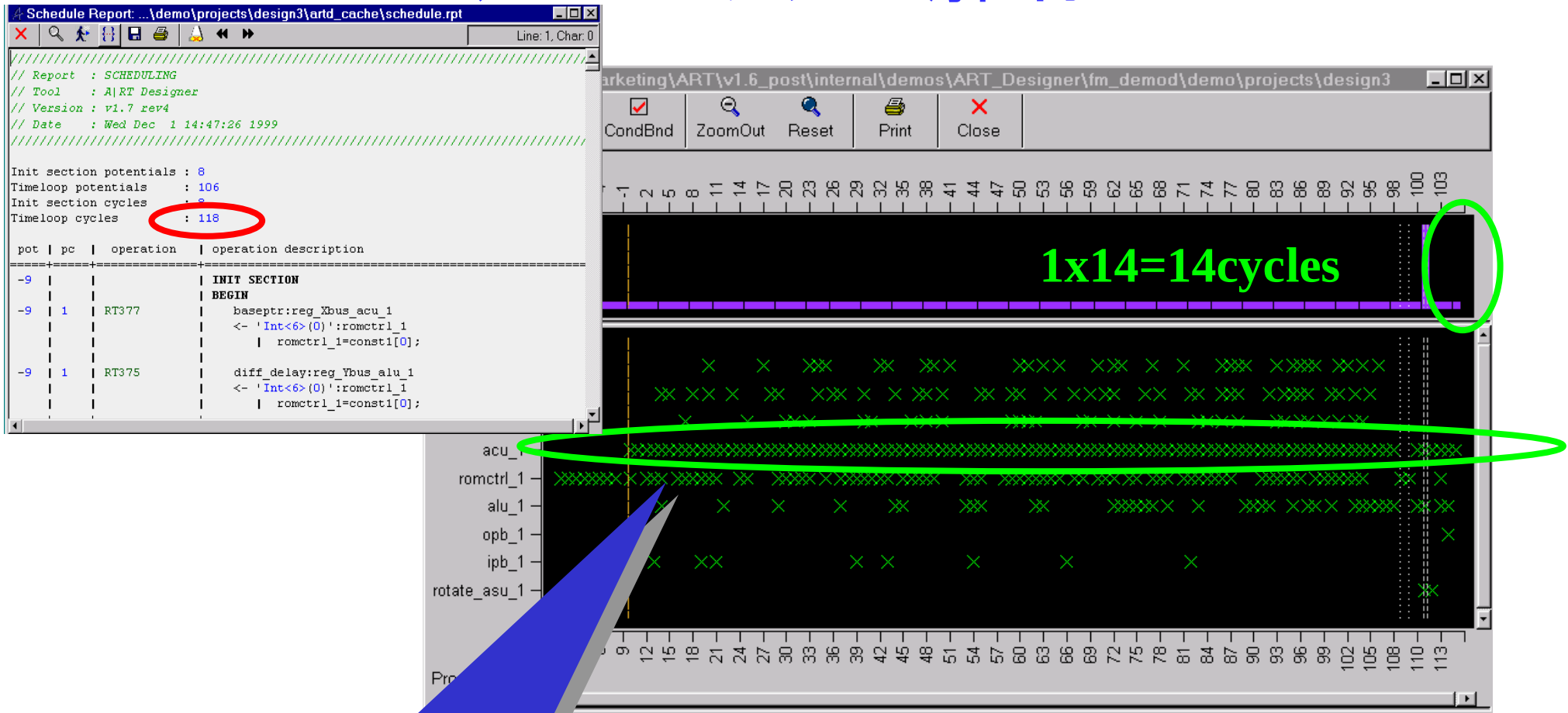


```
assign_fcalletoasu("/.../*=rotate(*)","rotate_asu1");
```



Case Study

処理ネックの解消

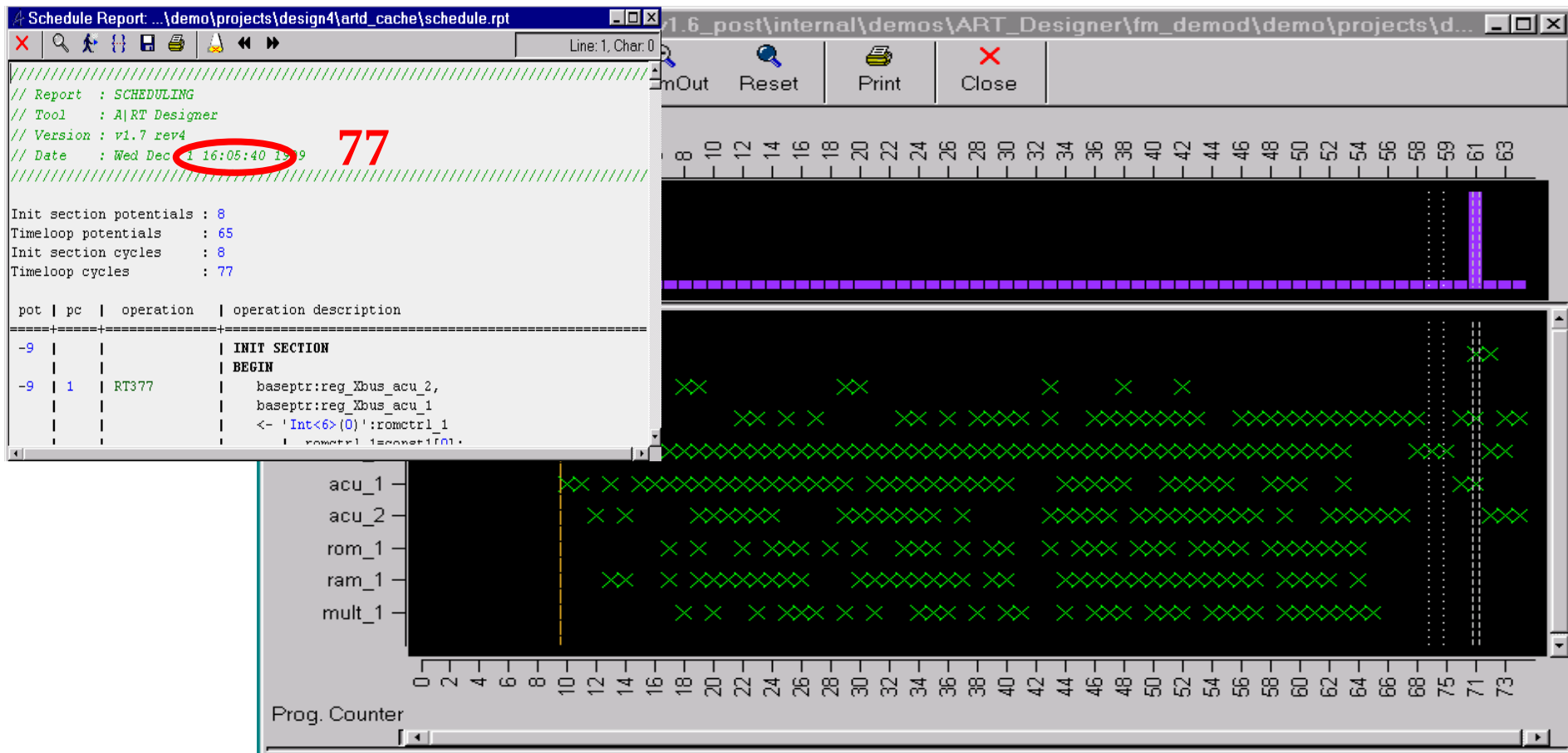


Address Computation Bottleneck

instantiate("acu_2");

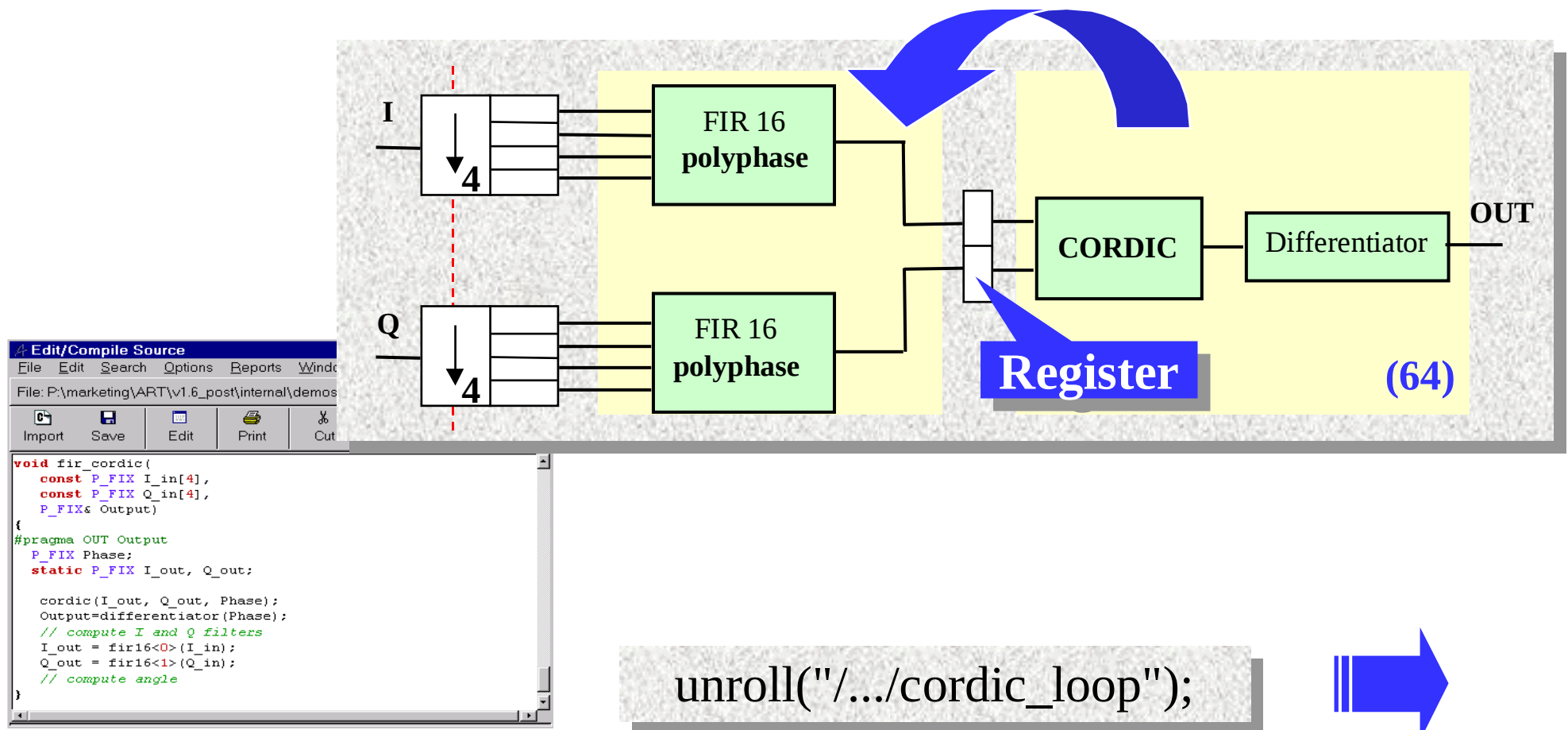


処理ネックの解消 -ACU_2の割当結果-

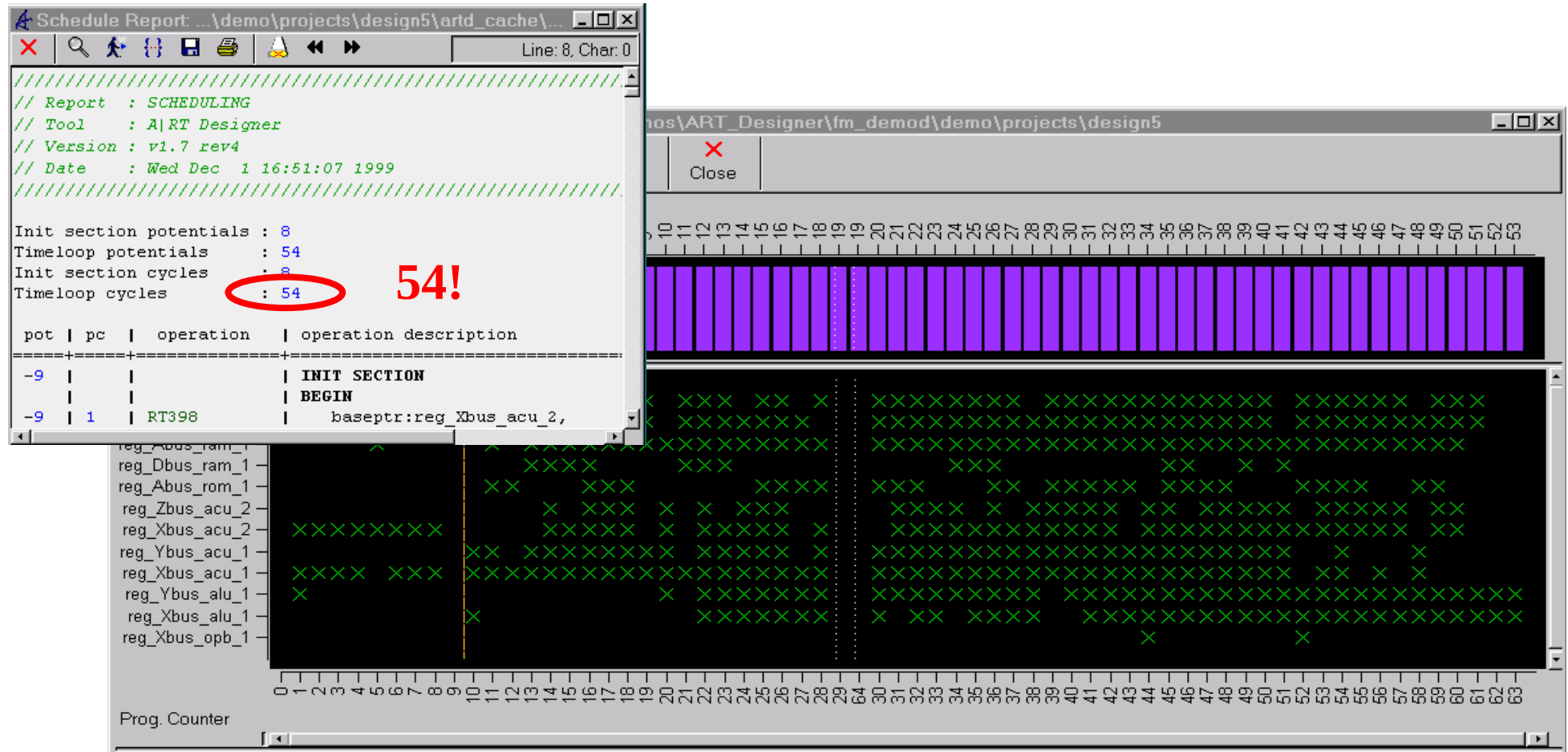


Case Study

パイプライン化



パイプライン化の結果



Total number of fields and bits:
 datapath : 51 fields, 558 bits
 creg : 1 fields, 1 bits

Case study

結論

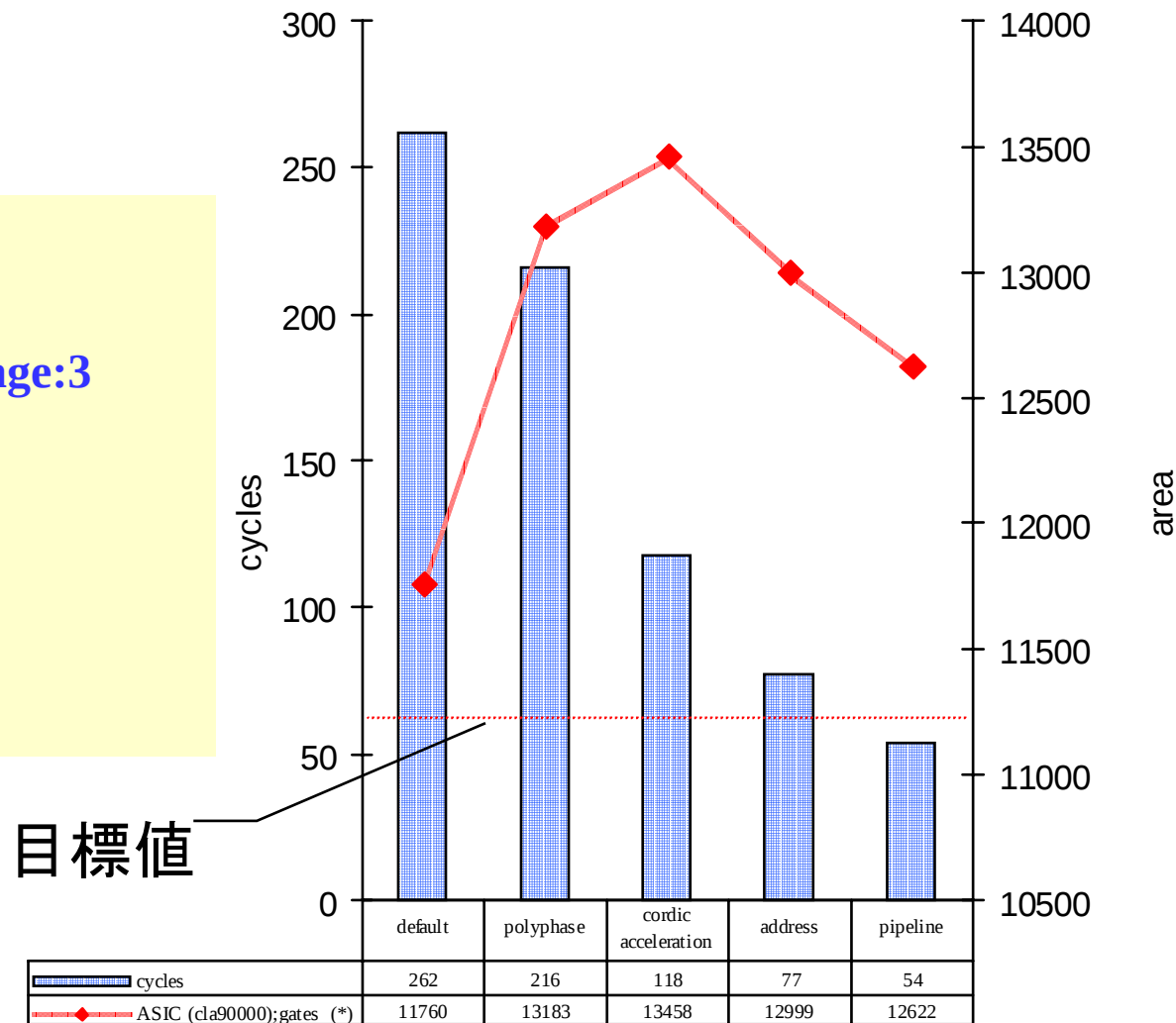
C-source:140

source code change:3

lines edited:4

hours spent:2

HDL-output:7750



(*) excluding RAM, ROM



Frontierの A|RT と SystemC ?

- **Frontier is co-developer of SystemC together with Synopsys and Coware**

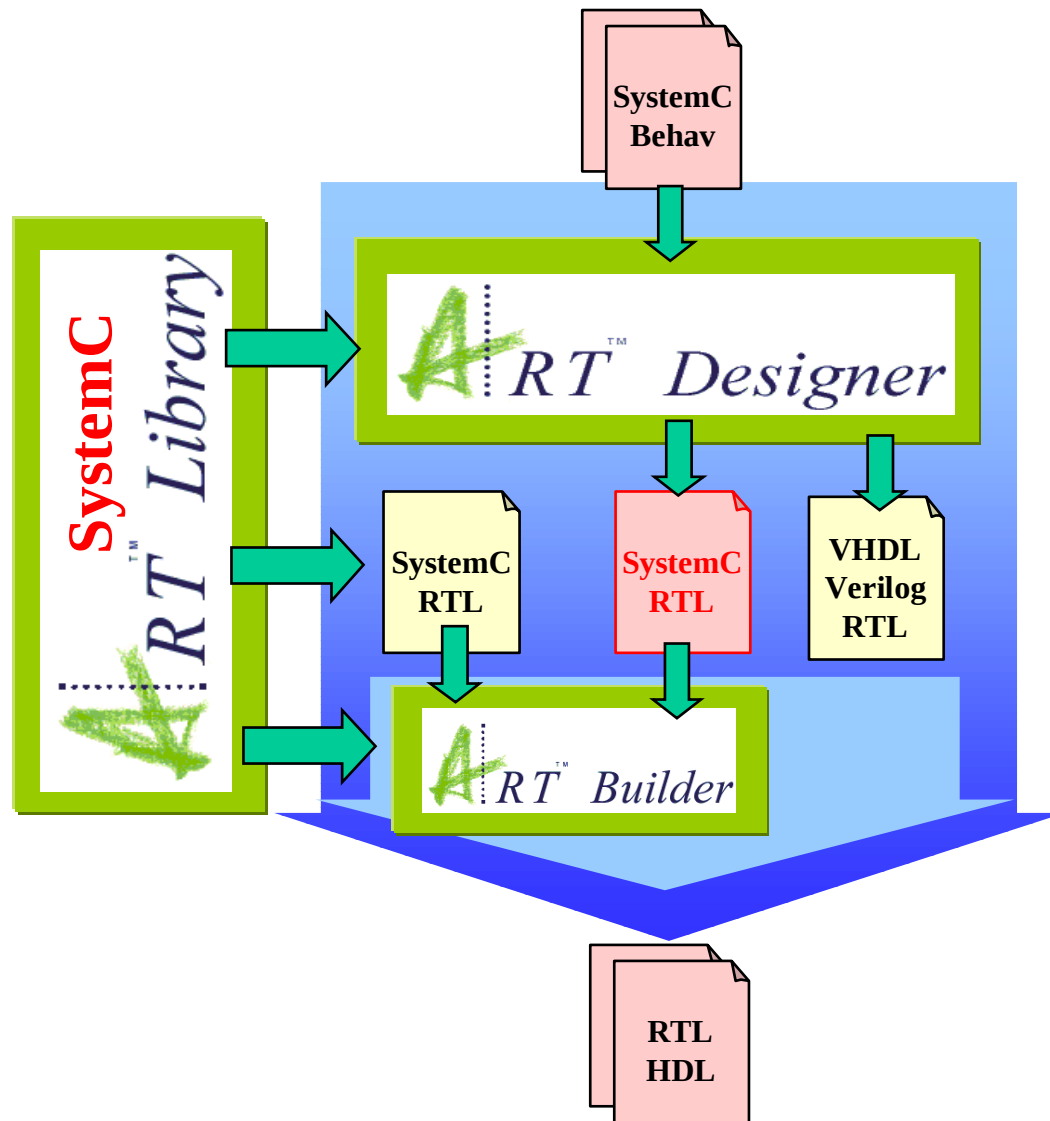
- Frontier brings in the Fixed Point modeling library (ART Library)
- Synopsys brings in the Scenery environment
- Coware will provide its interprocess communication technology

- **Frontier is focussing on providing synthesis solutions based on SystemC**

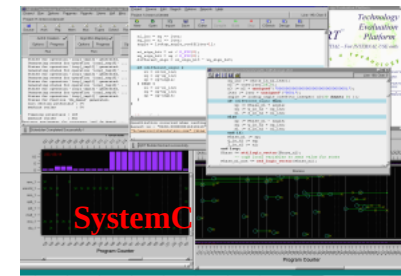
- ART Builder : RTL-C to RTL-HDL
- ART Designer : Behav-C to RTL-HDL



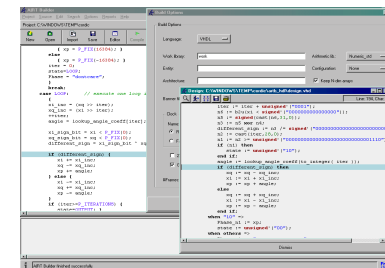
A|RT と SystemC



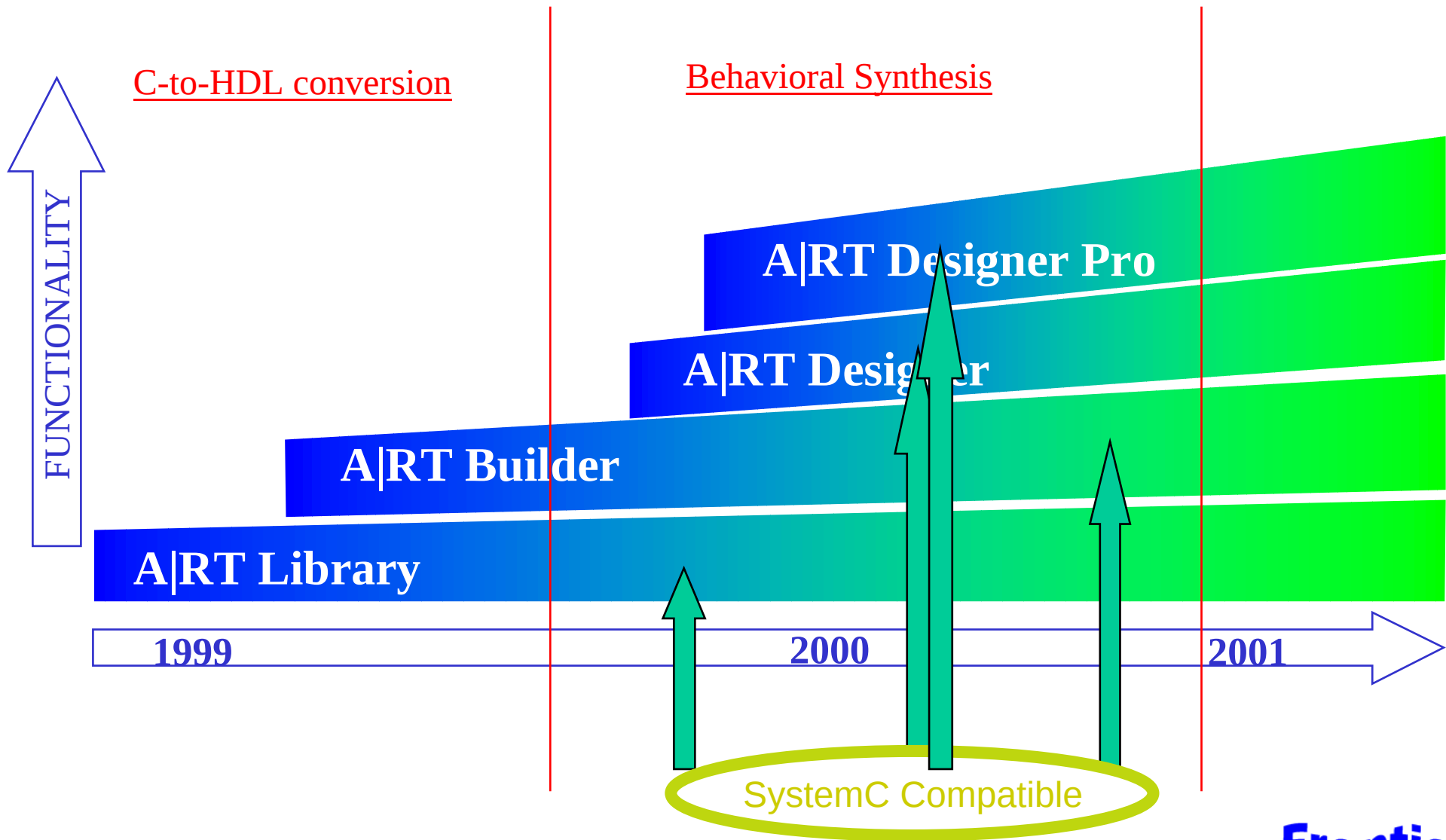
System Design



RTL Design



製品ロードマップ



まとめ

- **Frontier**社はC言語からの設計、アーキテクチャ合成ツールのリーダーです。
 - ART Library : C based hardware modeling
 - ART Builder : “RTL-C to RTL-HDL” translation
 - ART Designer : “Behav-C to RTL-HDL” synthesis
- **Frontier**社は従来のVHDL/Verilog設計の上流設計フローとして、C言語での設計手法を提案しております。
- **Frontier**社は **SystemC standard**に準拠した設計ツールの開発を推進しております

