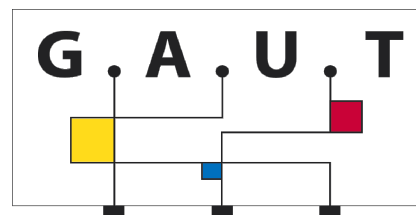
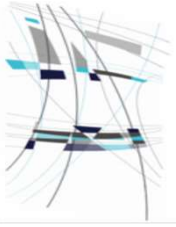


# GAUT Labs

Philippe COUSSY

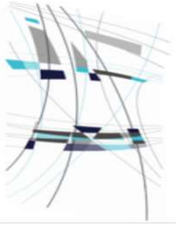
[philippe.coussy@univ-ubs.fr](mailto:philippe.coussy@univ-ubs.fr)





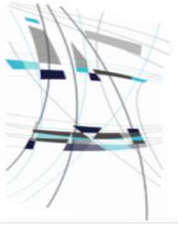
# Outline

- Lab1 “Introduction”
  - ◇ Discovering the GAUT IDE
  - ◇ First basic synthesis
    - ◇ Build project
    - ◇ Discover GAUT environment
    - ◇ Study GAUT Output files
- Lab1 “Simulation”
  - ◇ Testbench generation
  - ◇ Simulation
- Lab2 “Design Space exploration”
  - ◇ Synthesis options
    - ◇ Loop unrolling
    - ◇ Resource constraint
- Lab3 “Do It Yourself”



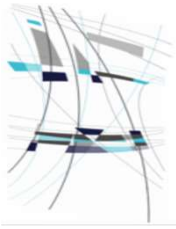
# Outline

- Lab1 “Introduction”
  - ◇ Discovering the GAUT IDE
  - ◇ First basic synthesis
    - ◇ Build project
    - ◇ Discover GAUT environment
    - ◇ Study GAUT Output files
- Lab1 “Simulation”
  - ◇ Testbench generation
  - ◇ Simulation
- Lab2 “Design Space exploration”
  - ◇ Synthesis options
    - ◇ Loop unrolling
    - ◇ Resource constraint
- Lab3 “Do It Yourself”



# Lab1 Discovering the tool

- Login
  - ◇ Login: UBS-30XXX\gaut
  - ◇ Password: gaut
- Enter eclipse folder **D:/GAUT/eclipse-gaut...../eclipse**
- Double click on eclipse.exe file
- By default the selected workspace is the one we provided you containing all the labs.
  
- You have launched the eclipse C++ including Gaut Tool plugin.
- You're in an eclipse environment!



# Lab1 GAUT IDE

The screenshot displays the GAUT IDE interface. The main editor shows a C program named `DCT64x64.c` with the following code:

```
* in: 64x64

short * dct(const char *in, int nbloc) {
    unsigned short i;
    int x0, x1, x2, x3, x4, x5, x6, x7, x8;
    short *out;
    dct_bloc: for (; nbloc; nbloc--) {

        /* All values are shifted left by 10
        * and rounded off to nearest integer
        */

        static const unsigned short c1 = 1420; /* cos PI/16 * root(2) */
        static const unsigned short c2 = 1338; /* cos PI/8 * root(2) */
        static const unsigned short c3 = 1204; /* cos 3PI/16 * root(2) */
        static const unsigned short c5 = 805; /* cos 5PI/16 * root(2) */
        static const unsigned short c6 = 554; /* cos 3PI/8 * root(2) */
        static const unsigned short c7 = 283; /* cos 7PI/16 * root(2) */

        static const unsigned short s1 = 3;
        static const unsigned short s2 = 10;
        static const unsigned short s3 = 13;

        inner_loop1: for (i = 8; i > 0; i--) {
            x8 = (int) in[0] + (int) in[7]; /* levelshift: - 128 - 128 */
            x7 = (int) in[1] + (int) in[6]; /* levelshift: - 128 - 128 */
            x6 = (int) in[2] + (int) in[5]; /* levelshift: - 128 - 128 */
            x5 = (int) in[3] + (int) in[4]; /* levelshift: - 128 - 128 */

            x0 = (int) in[0] - (int) in[7]; /* levelshift: - 128 + 128 */
            x1 = (int) in[1] - (int) in[6]; /* levelshift: - 128 + 128 */
            x2 = (int) in[2] - (int) in[5]; /* levelshift: - 128 + 128 */
            x3 = (int) in[3] - (int) in[4]; /* levelshift: - 128 + 128 */

            x8 -= 256; /* levelshift: - 128 - 128 */
            x7 -= 256; /* levelshift: - 128 - 128 */
            x6 -= 256; /* levelshift: - 128 - 128 */
            x5 -= 256; /* levelshift: - 128 - 128 */

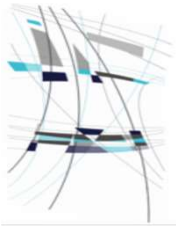
            x4 = x8 + x5;
            x8 -= x5;

            x5 = x7 + x6;
            x7 -= x6;
        }
    }
}
```

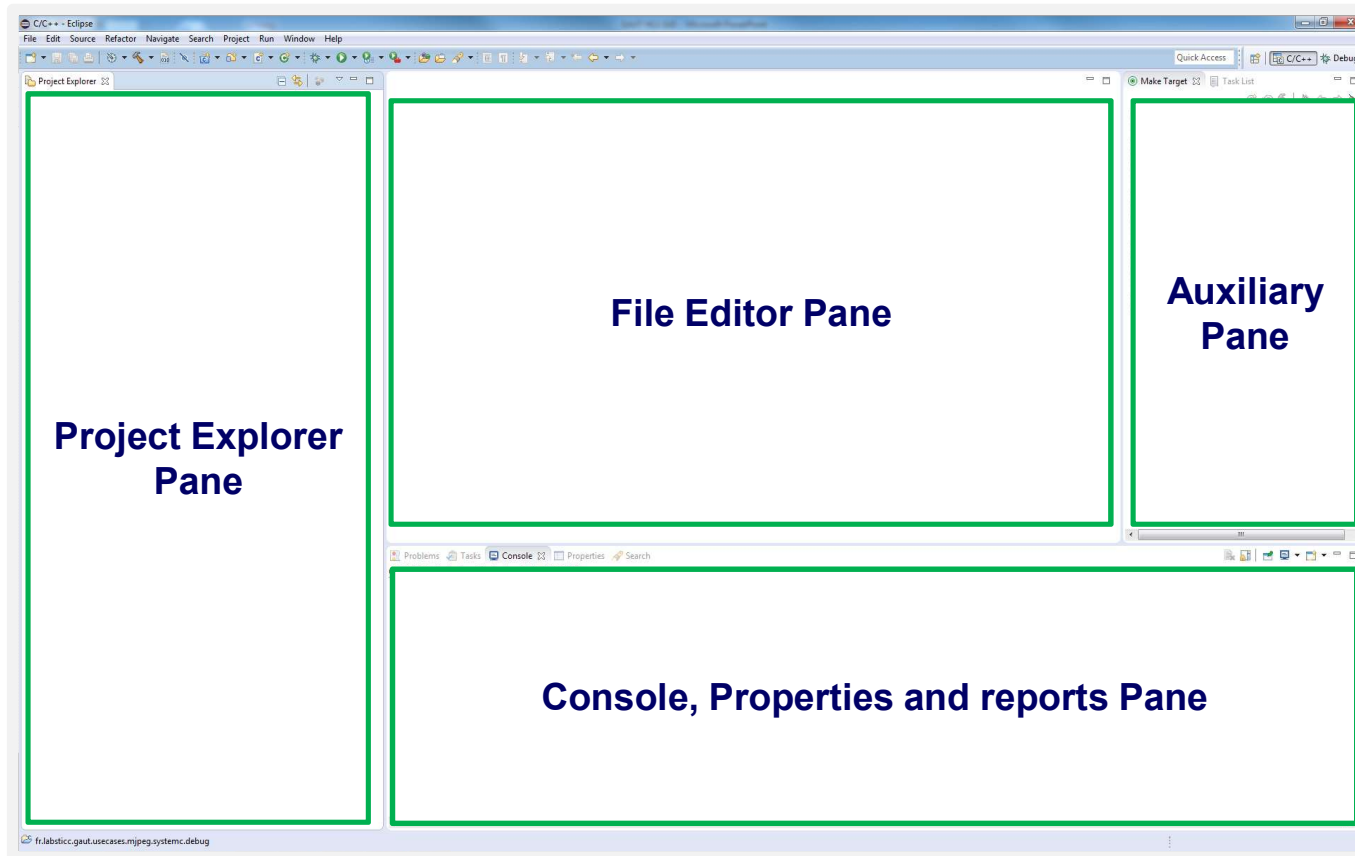
The Project Explorer on the left shows a project named `DCT64x64` with various source files and build artifacts. The Properties window on the right shows the following information:

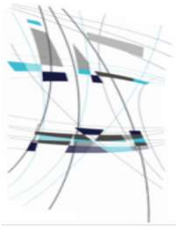
Property	Value
derived	false
editable	true
last modified	23 novembr...
linked	false
location	D:\workspa...
name	DCT64x64
path	/DCT64x64

The bottom of the IDE shows the Console window with the text "Gaut Console".



# Lab1 GAUT IDE





# Lab1 Build project

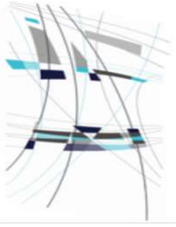
- In the project explorer pane
- Expand lab1 folder, go to “src” folder
  - ◇ Double-click on the C file to open and study the code

- Click right on lab 1 project
  - ◇ Build project



Alternately, use this Toolbar Button

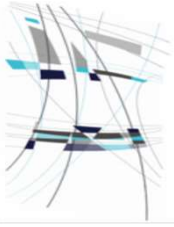
- ◇ This will report GCC build errors and GAUT HLS error messages
- ◇ You can move to the console to see build progress
- ◇ Rebuild project
  - ◇ To remove (false) errors/warnings...



# Lab1 Discover Gaut Outputs

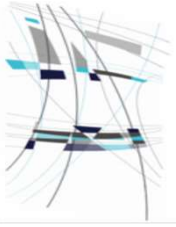
- New project directories are created
- C/C++ Project: the application to be synthesized
  - ◇ Source directory
  - ◇ Debug/Release directory
  - ◇ Binaries directory
- SystemC Project: the RTL systemC project
  - ◇ Source folder : \*.cpp and \*.hpp files
- VHDL Project: the RTL VHDL project
  - ◇ Source folder: \*.vhd files
  - ◇ Modelsim script \*.tcl





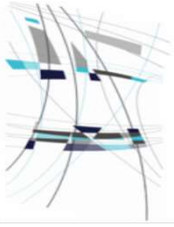
# Lab1 Discover Gant Outputs

- Expand Debug folders and discover ...
  - ◇ Dot subfolders and associated files
    - ◇ CDFG.dot, datapath.dot, FSM.dot
  - ◇ Pseudocode folder (file.txt)
  - ◇ SVG folder
    - ◇ Sheduling results (gantt diagram, file.svg)
  - ◇ Synthesis result statistics (labx.gstats)
  - ◇ Synthesis options (project.gsettings)
  
- Expand projects folders
  - ◇ SystemC (folder Lab1.systemc.debug)
  - ◇ VHDL (folder Lab1.vhdl.debug)
    - ◇ Datapath (\_dp), FSM (\_fsm), top (\_fun), lib (\_carac)
    - ◇ Have a look on the files
  
- Set the values of the constant b as follows
  - ◇ First 5 and next 3
  - ◇ Study the impact of such theses two modifications via the CDFG and the gantt files



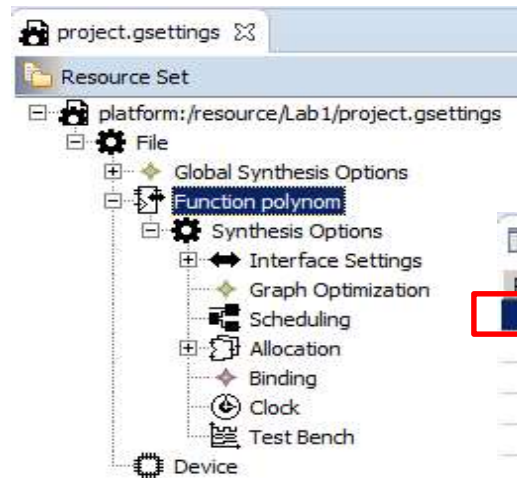
# Outline

- Lab1 “Introduction”
  - ◇ Discovering the GAUT IDE
  - ◇ First basic synthesis
    - ◇ Build project
    - ◇ Discover GAUT environment
    - ◇ Study GAUT Output files
- Lab1 “Simulation”
  - ◇ Testbench generation
  - ◇ Simulation
- Lab2 “Design Space exploration”
  - ◇ Synthesis options
    - ◇ Loop unrolling
    - ◇ Resource constraint
- Lab3 “Do It Yourself”



# Lab1 Generate Test bench

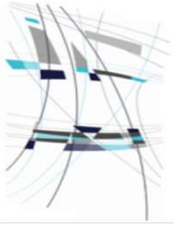
- Set the constant  $b$  to its initial value 2
- Open the synthesis option file (project.gsettings)
- Select “Function polynom”
  - ◇ Generate Test Bench option
    - ◇ Click true, **save** (file-> save or ctrl+s)



The screenshot shows the 'Properties' window with a table of properties. The 'Generate Test Bench' property is highlighted with a red box and has a checked checkbox next to the value 'true'.

Property	Value
Generate Test Bench	<input checked="" type="checkbox"/> true
Id	
Ignored	<input type="checkbox"/> false
Interface	CTRL_HandSh
Name	polynom

- ◇ Re-build the project
  - ◇ A testbench has been created in the VHDL project folder
    - ◇ File lab1\_tb.vhd
  - ◇ Study the code



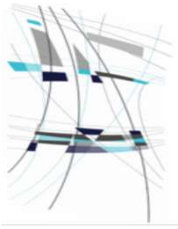
# Lab1 First Simulation Setup

In Modelsim,

- Create the library gaut for this first time simulation in the workspace.
  - ◇ Click on File -> New -> Library
  - ◇ In the frame create, chose a new library
  - ◇ Enter gaut as Library Physical Name
- Now, compile the hdl files of gaut operator library
  - ◇ Click on Compile -> Compile
  - ◇ Select gaut as Library
  - ◇ Browse for gaut folder in the library and select all files **ctrl+a**
  - ◇ Click on Compile until you obtain no error message

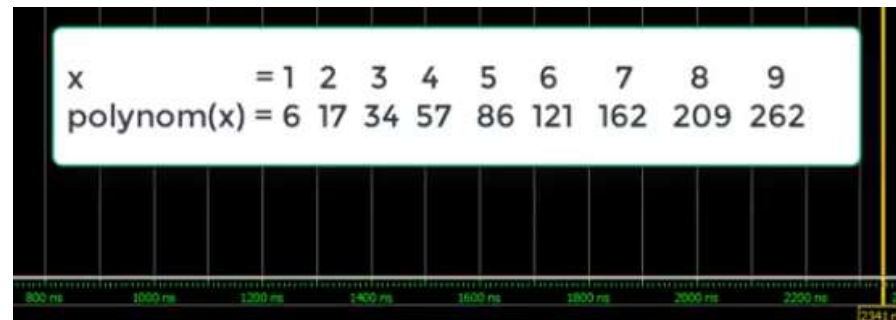
- Otherwise, execute the TCL script
  - ◇ We have generated a script tcl for you
  - ◇ Go Tools -> Tcl -> Execute Macro ...
  - ◇ Select the workspace directory
  - ◇ Open **Makefile\_gautlib.tcl**

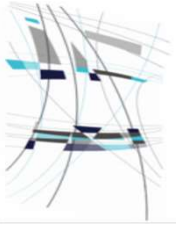




# Lab1 Simulation With Gaut Script

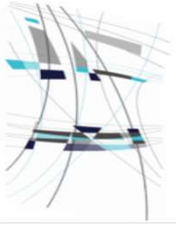
- Launch the simulation
  - ◇ Use the automatically generated script tcl
    - ◇ Go Tools -> Tcl -> Execute Macro ...
  - ◇ Browse for the Makefile\_\*.tcl
    - ◇ This will automatically launch the simulation
      - ◇ Take care about the time units (ps -> ns)
      - ◇ Simulate->Runtime options->decimal
      - ◇ Run
- Study the wave forms
  - ◇ By default, inputs are counters from 0 to 9
  - ◇ Each rising edge of the signal done, new result is available
  - ◇ Result for





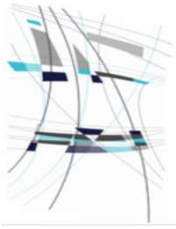
# Outline

- Lab1 “Introduction”
  - ◇ Discovering the GAUT IDE
  - ◇ First basic synthesis
    - ◇ Build project
    - ◇ Discover GAUT environment
    - ◇ Study GAUT Output files
- Lab1 “Simulation”
  - ◇ Testbench generation
  - ◇ Simulation
- Lab2 “Design Space exploration”
  - ◇ Synthesis options
    - ◇ Loop unrolling
    - ◇ Resource constraint
- Lab3 “Do It Yourself”



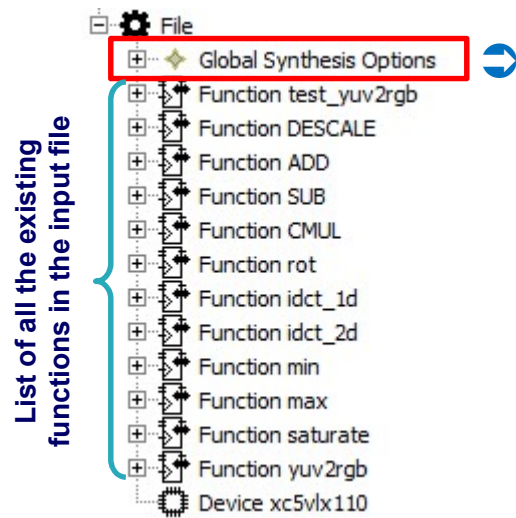
## Lab2 Synthesis options to explore

- Configuration #1
  - ◇ Synthesis option by default
    - ◇ No loop unrolling
- Configuration #2
  - ◇ Synthesis option by default
  - ◇ Loop unrolling enabled, factor = 2
- Configuration #3
  - ◇ Synthesis option by default
  - ◇ Loop unrolling enabled, factor = 4
- Configuration #4
  - ◇ Manual allocation, 1 operator of each type
  - ◇ Loop unrolling enabled, factor = 4



# Lab2 Synthesis options

- Open the \*.gsetigns file and modify the default configuration
  - ◇ Global Synthesis Options for all file functions
  - ◇ Synthesis options for each function, by default the global settings are used



## Default settings are:

- ⇒ No communication interfaces (point to point)
- ⇒ No loop unrolling
- ⇒ List scheduling policy
- ⇒ Automatic mean lower bound allocation policy
- ⇒ Maximum Weighted Bipartite Matching binding policy
- ⇒ No test bench generation





# Lab2 Loop unrolling

## ◆ Enabling loop unrolling

The screenshot shows the 'Function foo' settings. Under 'Synthesis Options', 'Graph Optimization' is expanded to show 'Loop Unroll loop1'. Below this, a table lists various properties:

Property	Value
Decompose Slow Mul Operator	false
Find Mac Operator	false
Id	
Internal Global Variable	false
Loop Unrolling	false
Remove Mul Operator With Constant	false

## ◆ Setting unrolling factor

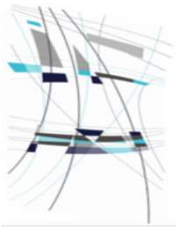
The screenshot shows the 'Loop Unroll loop1' settings. Below the navigation tabs, a table displays the configuration for the loop unrolling factor:

Property	Value
Factor	1
Id	
Line Number	11
Loop Label	loop1

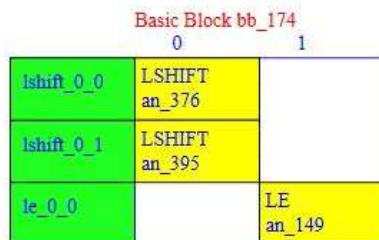
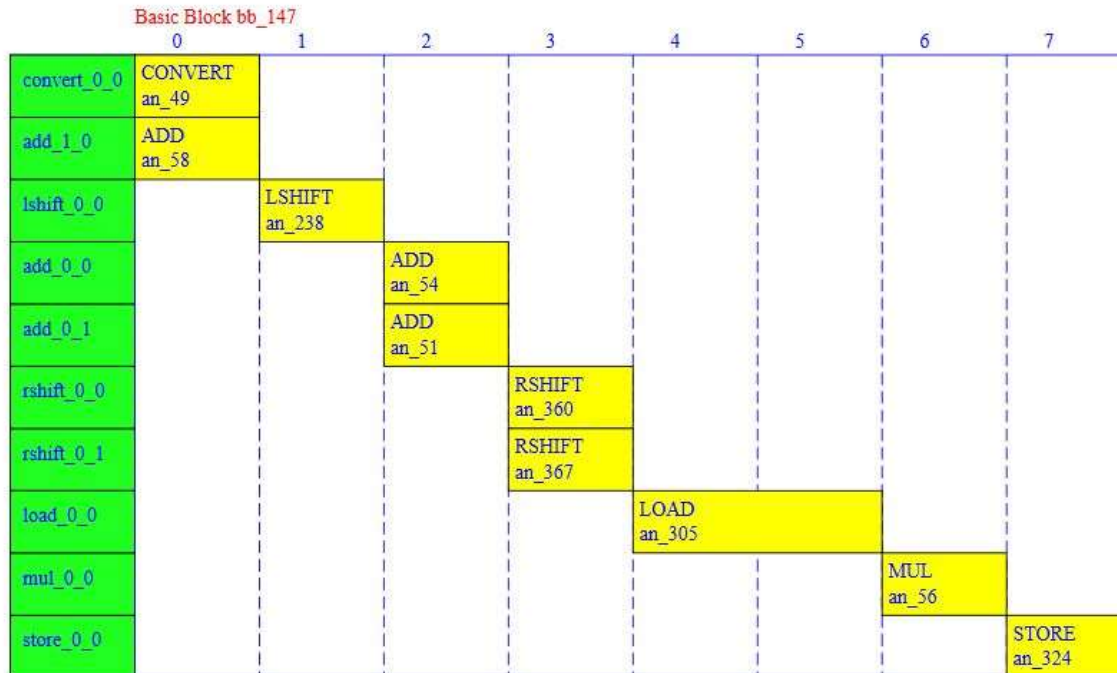
## ◆ Setting manual allocation

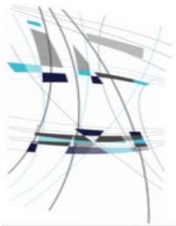
The screenshot shows the 'Allocation' settings. Below the navigation tabs, a table displays the allocation policy:

Property	Value
Id	
Policy	Manual



# Configuration #1





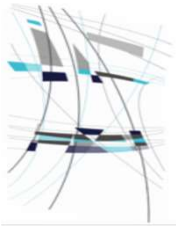
# Configuration #2

Basic Block bb\_147

	0	1	2	3	4	5	6	7	8
convert_0_0	CONVERT an_49								
add_1_0	ADD an_58	ADD n_365							
lshift_0_0		LSHIFT an_238	LSHIFT n_336						
convert_0_1		CONVERT n_333							
add_0_0			ADD an_54	ADD n_351					
add_0_1			ADD an_51	ADD n_340					
rshift_0_0				RSHIFT an_586	RSHIFT an_607				
rshift_0_1				RSHIFT an_593	RSHIFT an_600				
load_0_0					LOAD an_485	LOAD an_523			
mul_0_0							MUL an_56	MUL n_358	
store_0_0								STORE an_504	STORE an_542

Basic Block bb\_174

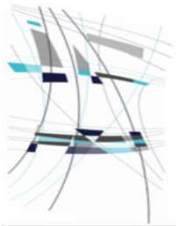
	0	1
lshift_0_0	LSHIFT an_616	
lshift_0_1	LSHIFT an_635	
le_0_0		LE an_149



# Configuration #3

Basic Block bb\_174

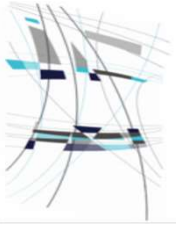
	0	1	2	3	4	5	6	7	8
lshift_0_0	LSHIFT an_1238								
lshift_0_1	LSHIFT an_1219								
add_0_0		ADD n_351							
add_0_1		ADD n_477							
add_0_2		ADD n_603							
rshift_0_0		RSHIFT an_1253	RSHIFT an_1295						
add_0_3		ADD n_340							
add_0_4		ADD n_466							
add_0_5		ADD n_592							
rshift_0_1			RSHIFT an_1260						
rshift_0_2			RSHIFT an_1274						
rshift_0_3			RSHIFT an_1288						
store_0_0			STORE an_1040						
rshift_0_4			RSHIFT an_1267						
rshift_0_5			RSHIFT an_1281						
load_0_0				LOAD an_1059	LOAD an_1097	LOAD an_1135			
store_1_0						STORE an_1078		STORE an_1116	STORE an_1154
lshift_1_0							LSHIFT an_923		
mul_0_0								MUL n_610	



# Configuration #4

Basic Block bb\_174

	0	1	2	3	4	5	6	7	8
lshift_0_0	LSHIFT an_1238	LSHIFT an_1219							
add_0_0		ADD n_351	ADD n_477	ADD n_603	ADD n_340	ADD n_466	ADD n_592		
rshift_0_0			RSHIFT an_1260	RSHIFT an_1274	RSHIFT an_1288	RSHIFT an_1267	RSHIFT an_1281	RSHIFT an_1295	
load_0_0				LOAD an_1059	LOAD an_1097	LOAD an_1135			
rshift_0_1					RSHIFT an_1253				
store_0_0						STORE an_1040			
store_1_0							STORE an_1078	STORE an_1116	STORE an_1154
lshift_1_0							LSHIFT an_923		
mul_0_0								MUL n_610	



# Outline

- Lab1 “Introduction”
  - ◇ Discovering the GAUT IDE
  - ◇ First basic synthesis
    - ◇ Build project
    - ◇ Discover GAUT environment
    - ◇ Study GAUT Output files
- Lab1 “Simulation”
  - ◇ Testbench generation
  - ◇ Simulation
- Lab2 “Design Space exploration”
  - ◇ Synthesis options
    - ◇ Loop unrolling
    - ◇ Resource constraint
- Lab3 “Do It Yourself”

# GAUT Labs

Philippe COUSSY

[philippe.coussy@univ-ubs.fr](mailto:philippe.coussy@univ-ubs.fr)

