esa

European Space Agency

Design-dependent reliability tools and IDS tools improvement study

Decuzzi Filomena

YGT semi final presentation

TEC-EDM

Agenda

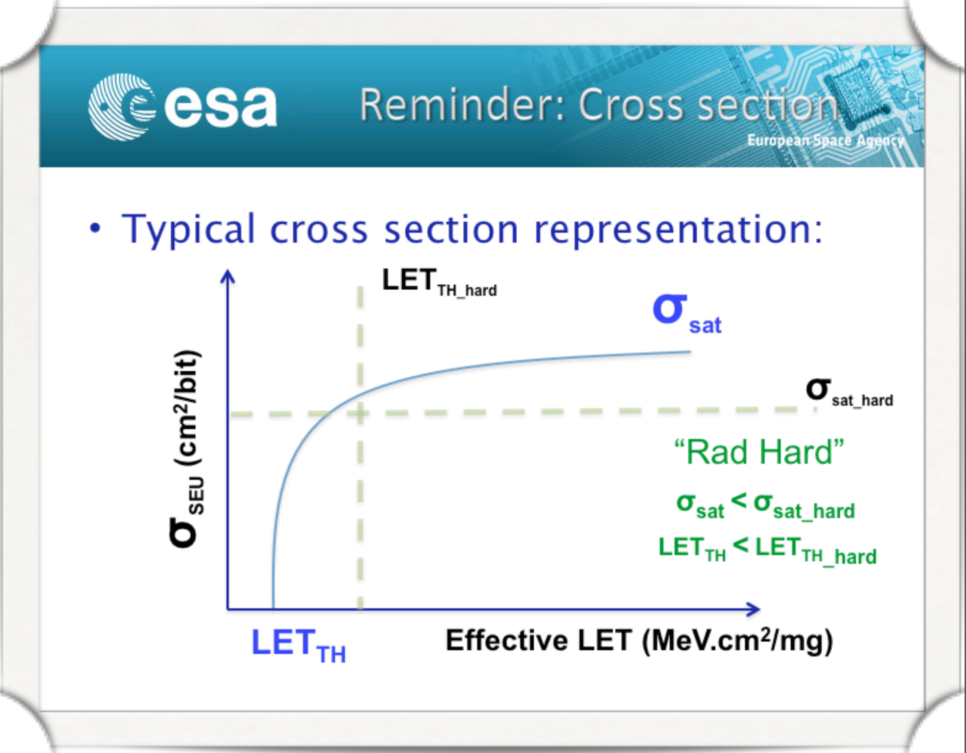
- Application-oriented SEU sensitiveness analysis
- Atmel technology library optimization
- Placement impact on the design performance

I

Study and development of an application-oriented analysis tool

Motivation

- Devices to be used in radioactive environments are qualified using accelerated radiation ground testing obtaining the device cross section
  - Cross section = probability for a ionizing radiation to produce an upset



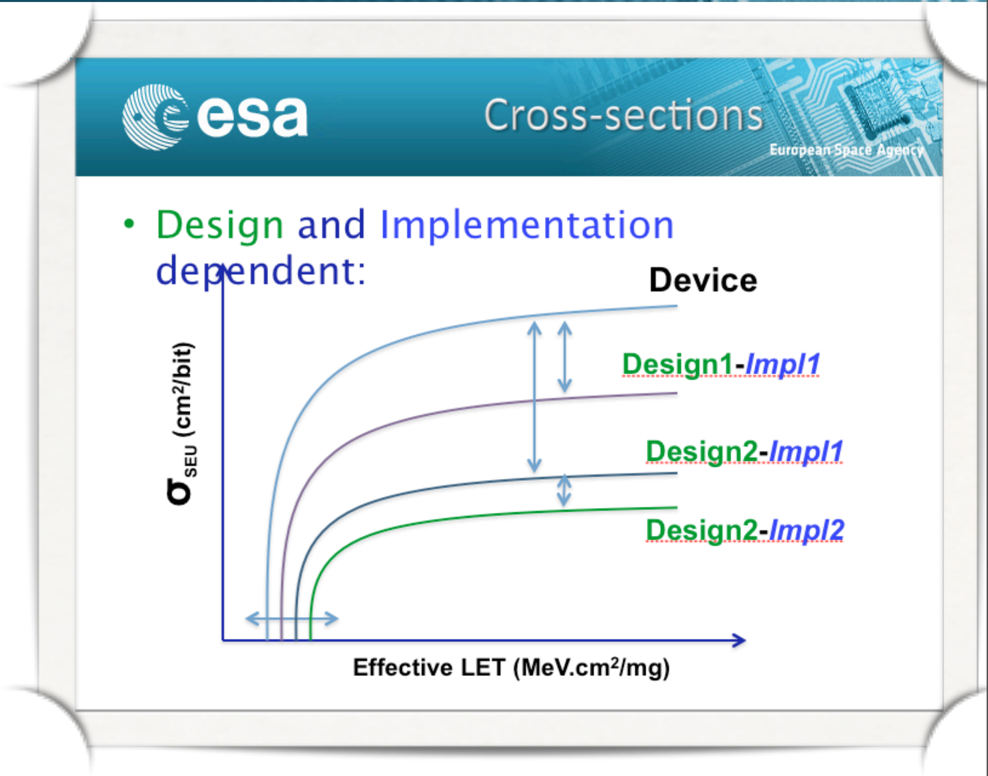
Reminder: Cross section

- Typical cross section representation:

- In case of FPGA different cross sections are measured:
  - Configuration memory cross section
  - Flip-Flop cross section
  - ...

Motivation

- Cross section are associated to the device, but ...
- An application (e.g. a design) seldom uses the whole device



Cross-sections

European Space Agency

- Design and Implementation dependent:

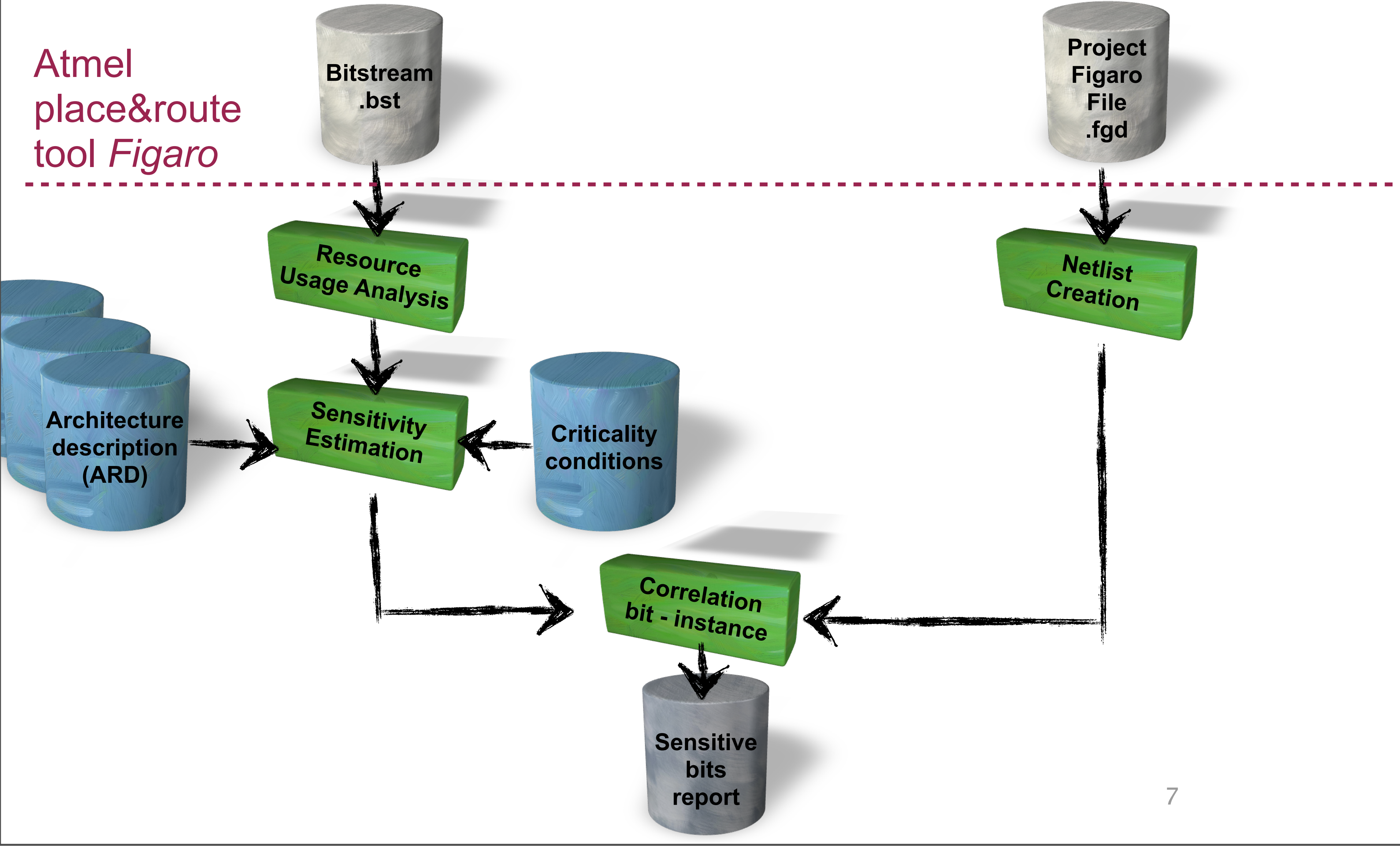
- Cross section should be de-rated to take into account for the real device usage to obtain application cross section

Goal

- To develop a tool for performing application-oriented sensitivity analysis of design implemented using radiation-hardened FPGA (AT40K, ATF280E and ATFS450 from Atmel)
- To correlate sensitive configuration bits to design instance
  - to detect most sensitive module of the design
  - to support debug

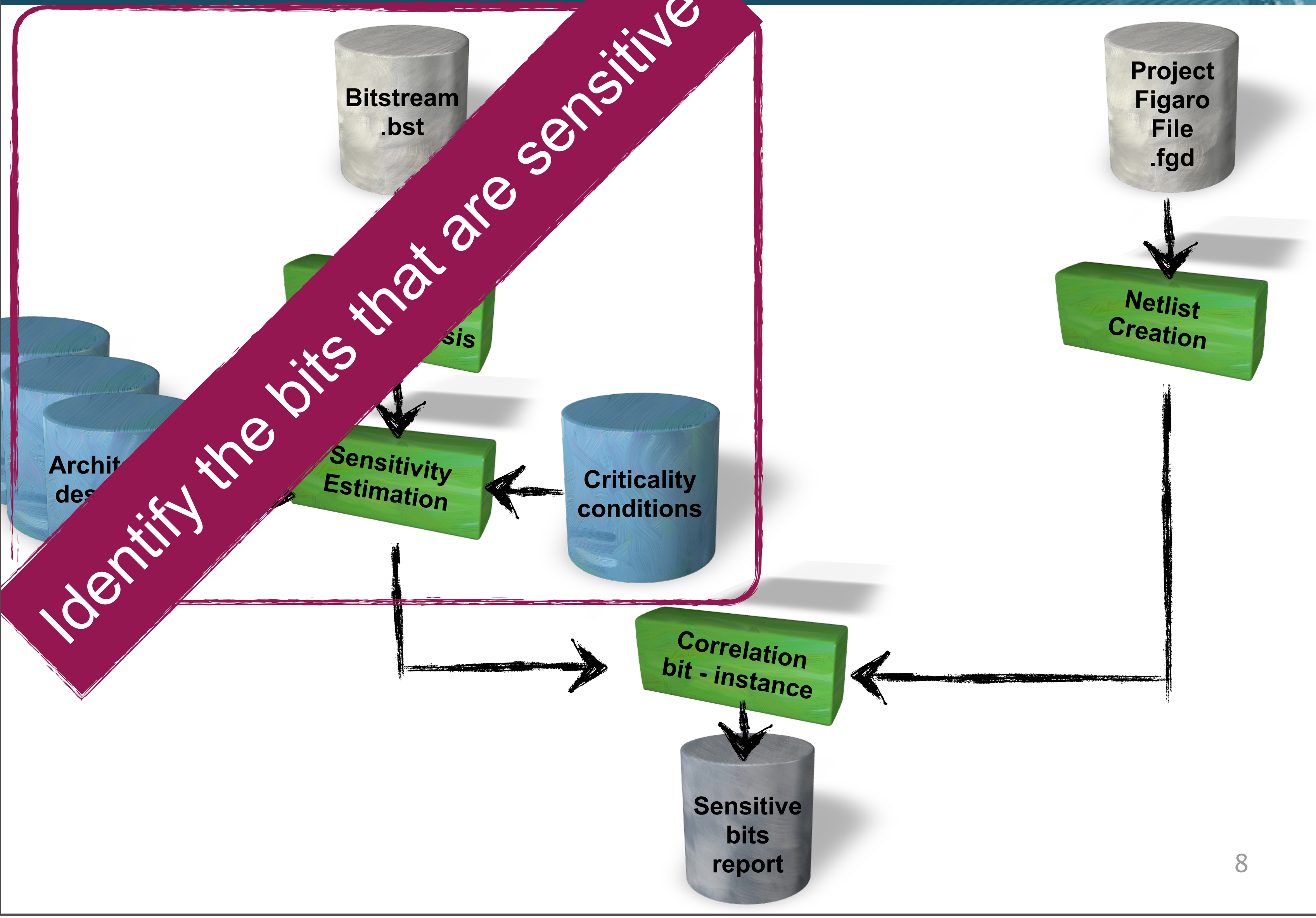
Analysis flow

Atmel place&route tool *Figaro*

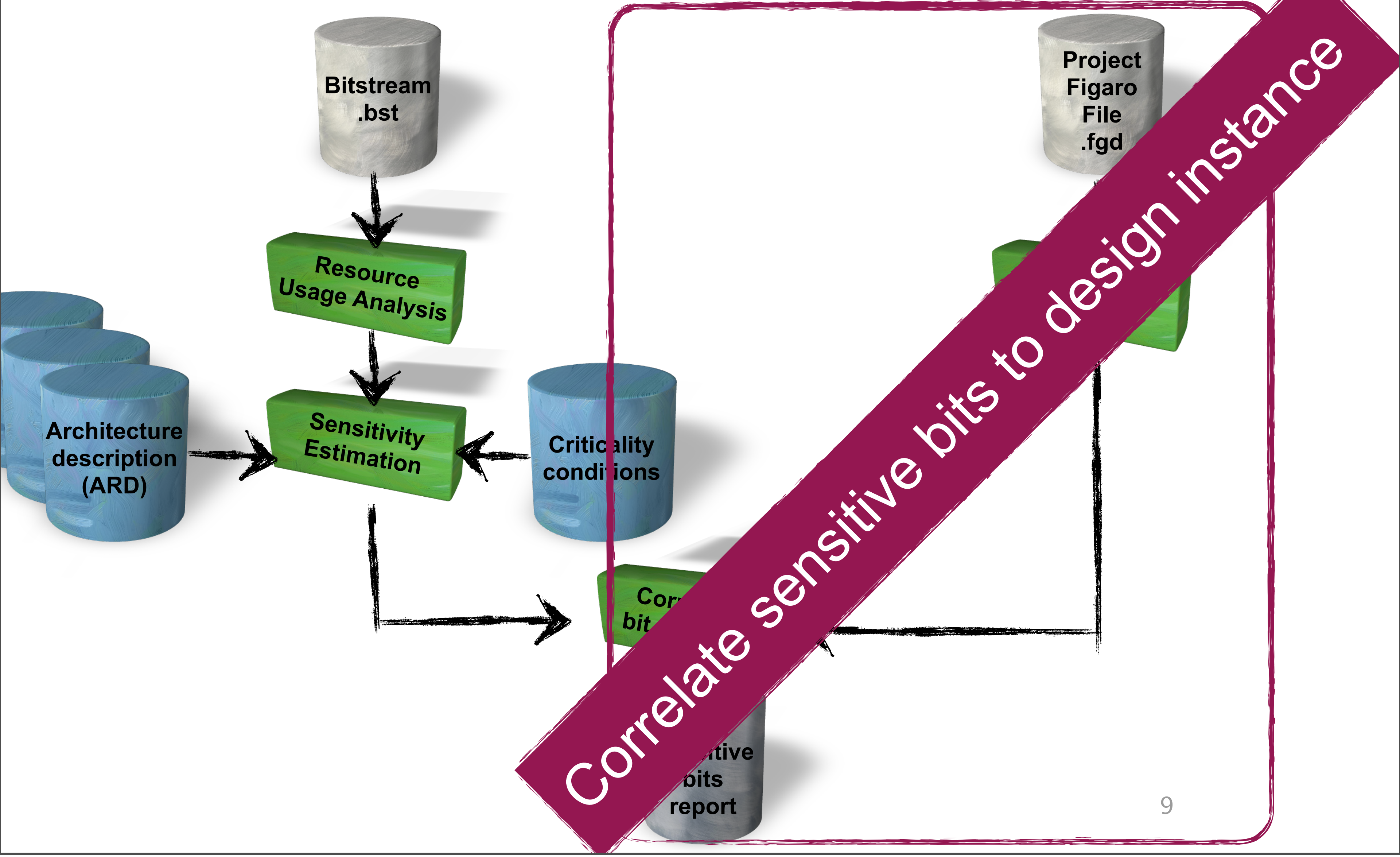


Susanna

European Space Agency

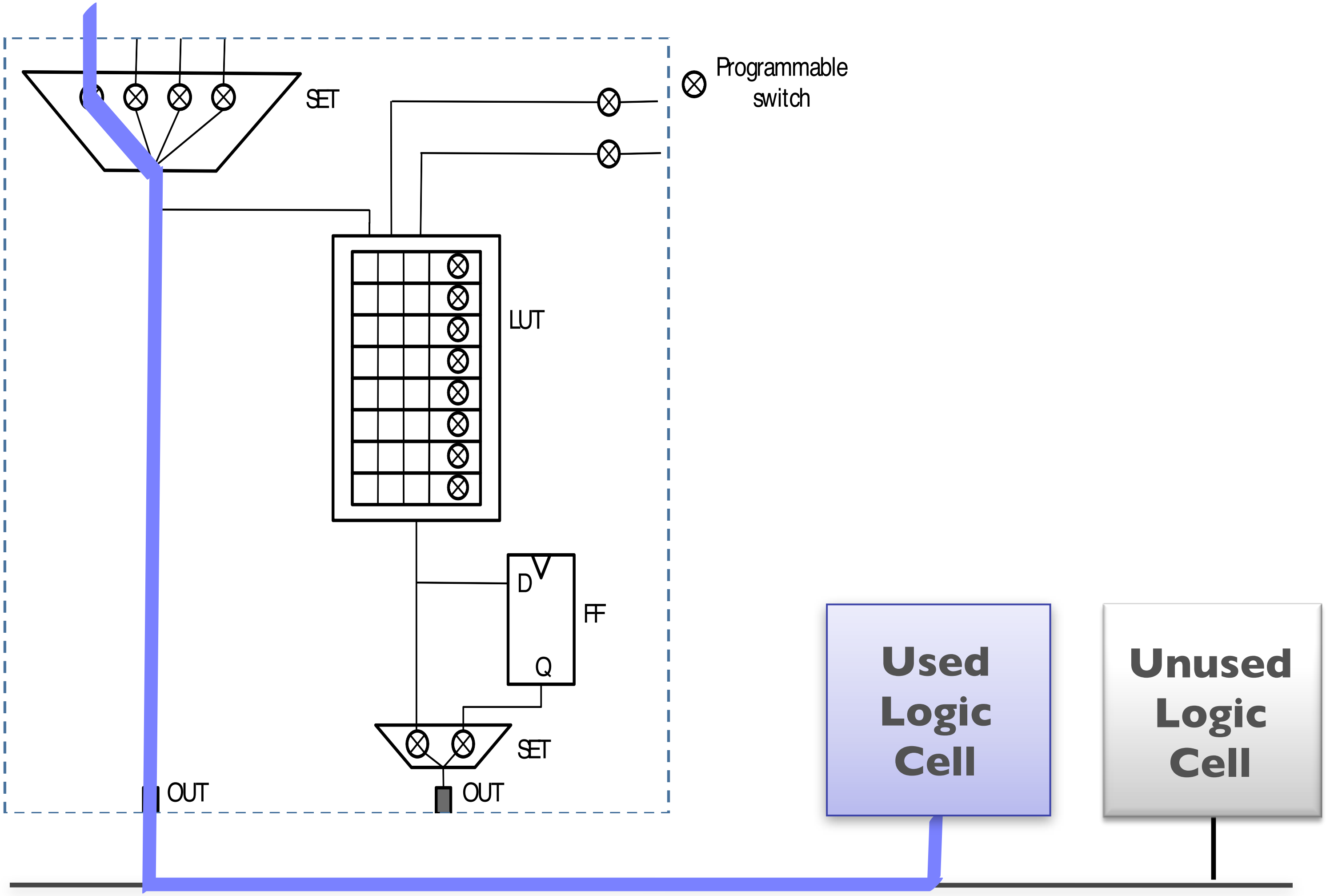


Jonathan

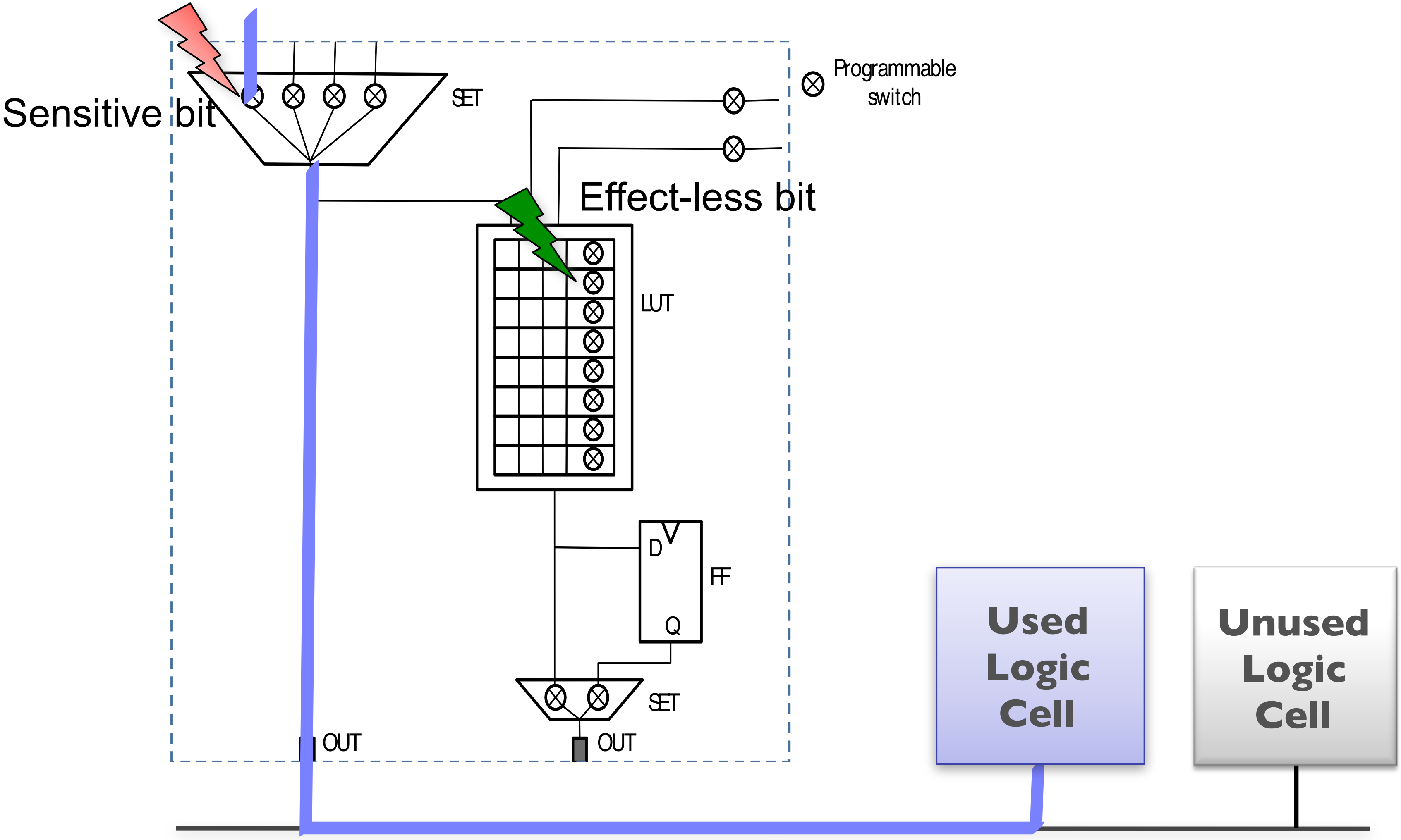


Correlate sensitive bits to design instance

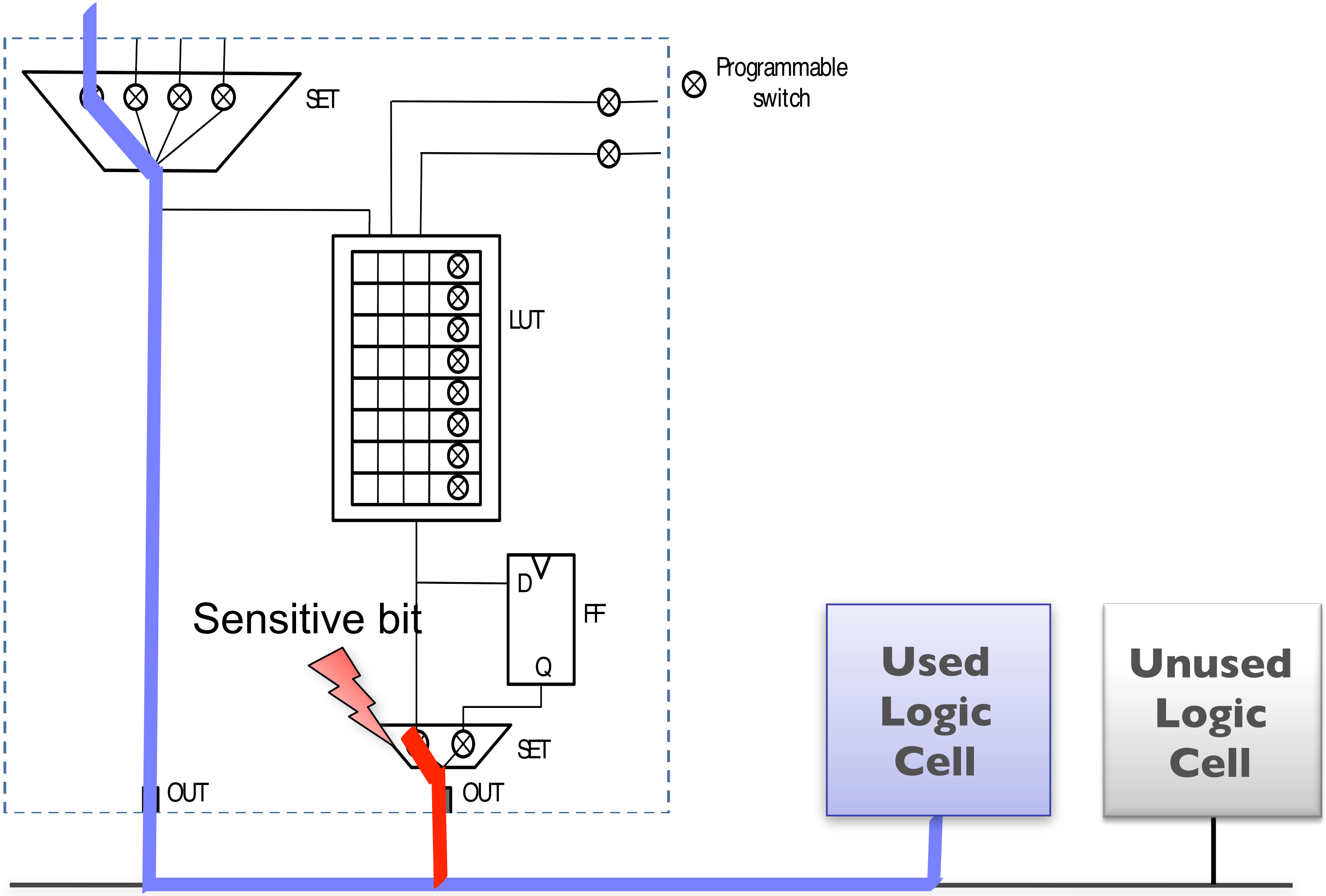
Faulty case


Programmable
switch
SET
LUT
D
Q
FF
SET
OUT
OUT
Used
Logic
Cell
Unused
Logic
Cell

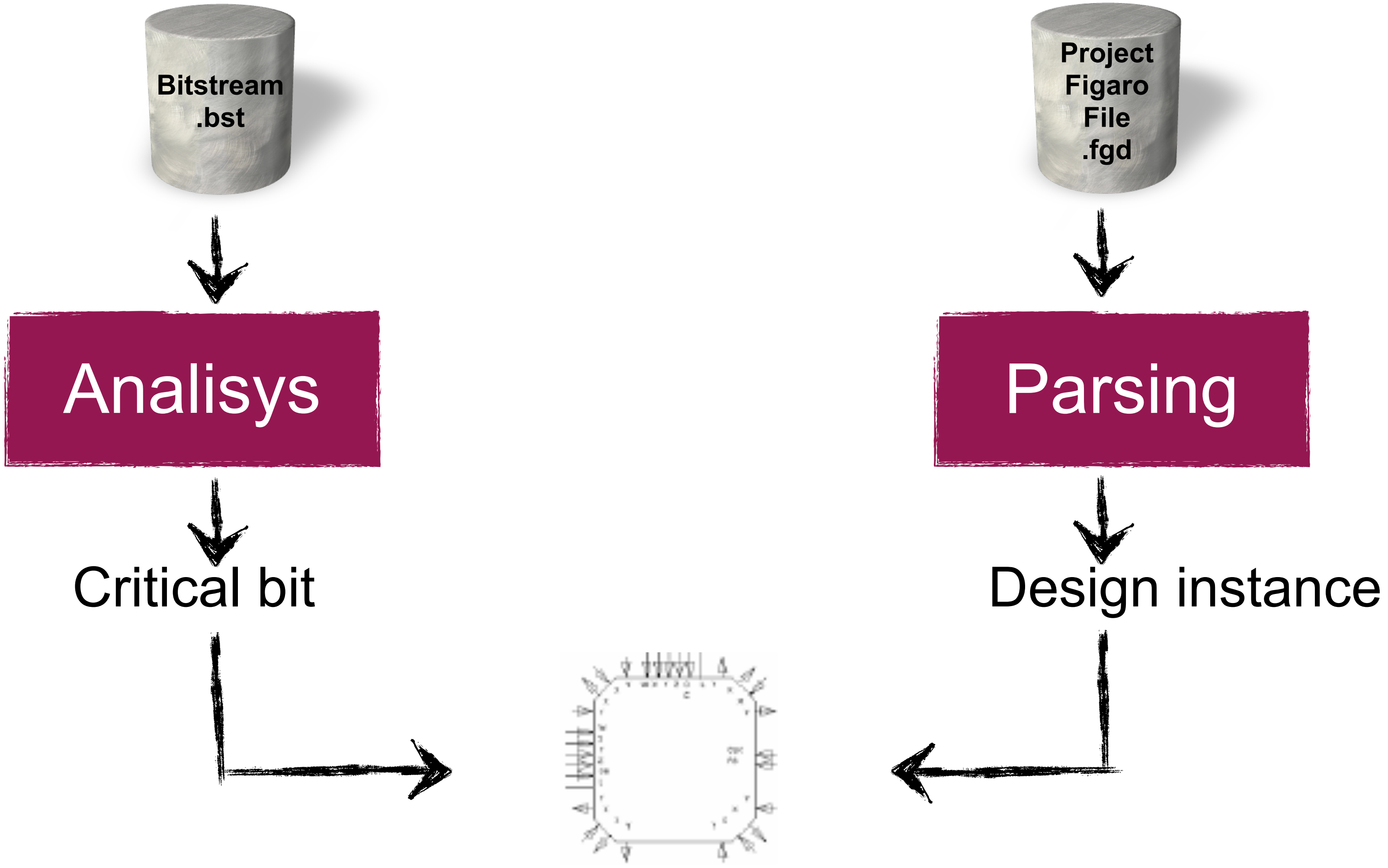
Faulty Intra-resource



Faulty Inter-resource



Bits – design instance



Bitstream .bst

Analisys

Critical bit

Project Figaro File .fgd

Parsing

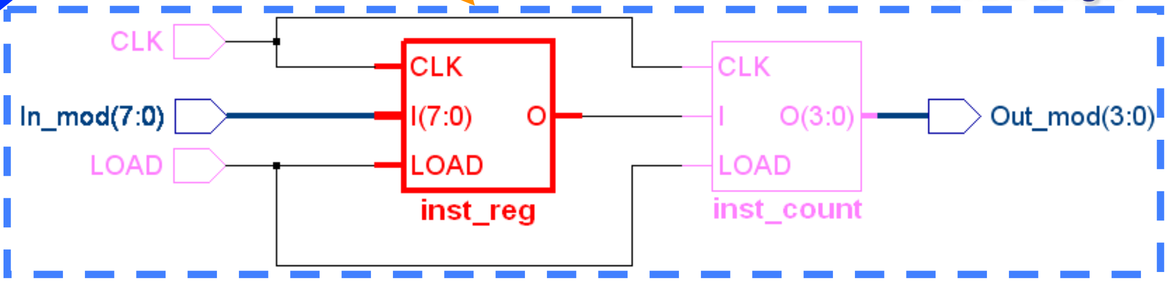
Design instance

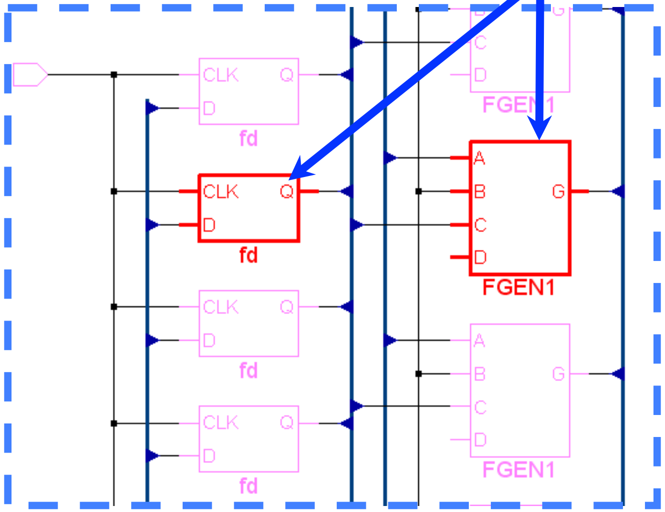
Physical Resource

Report

```
Byte address: 0x00011408; bit number: 0
Bit value: 1
Resource: CC(8, 20)
Hierarchical Design Instance: inst_reg
Map instance:  layer_5_ffdmap_reg_Y (FGEN1R)
Design Instance:
    layer_5_ffdmap_reg_Y (FD)
    ix7923z1498 (FGEN1)
-----------------------
```

RTL Design



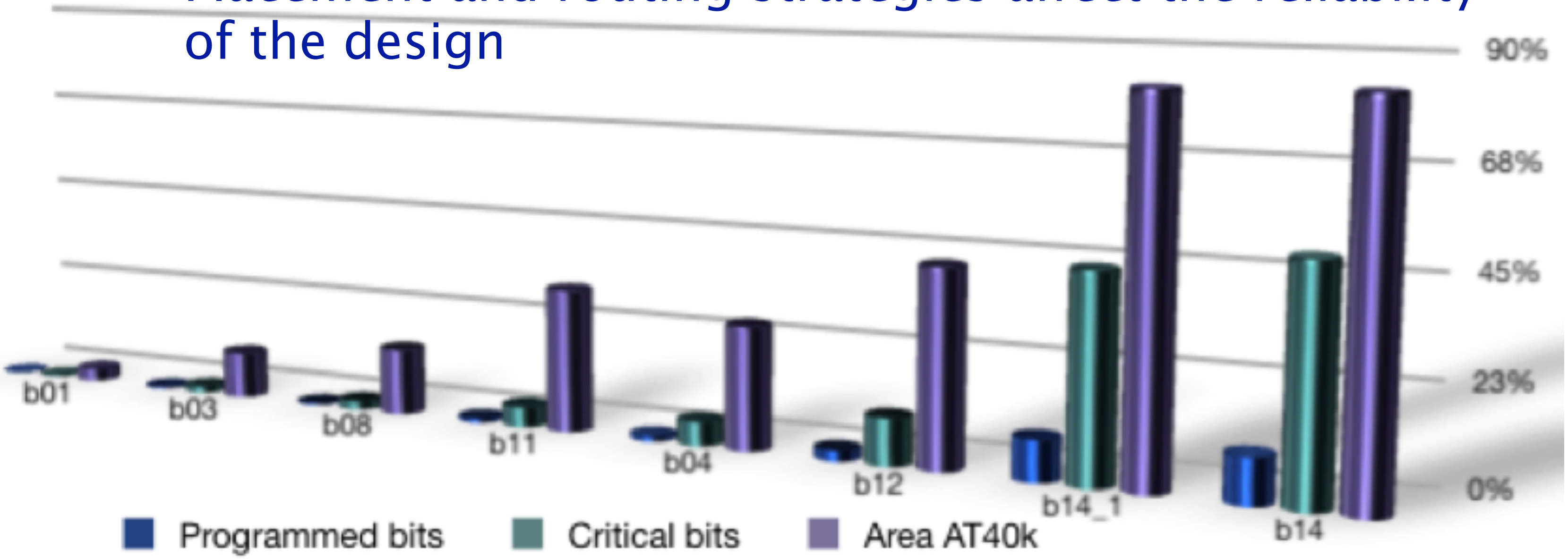


Netlist

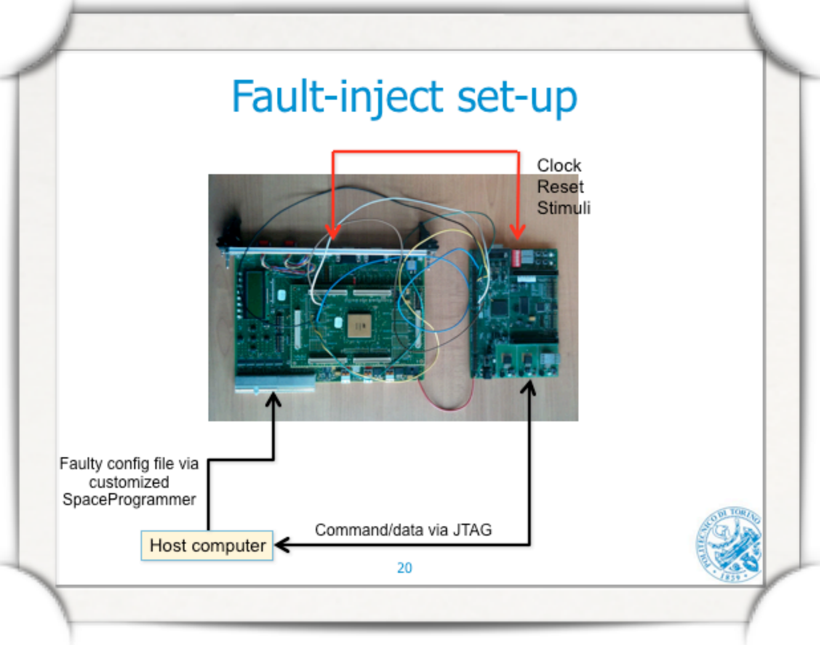
Analysis results

Several benchmark circuit of the ITC’99 set:

- Not all the programming bits for a used resource are potential critical
- A not programmed bits can induce an error in the circuit
- Placement and routing strategies affect the reliability of the design



Validation



Fault-inject set-up

- Design implemented in on Aerospace Development Kit
- Injection of Single Bit Upset in bits identified as sensitive:
  - around 63% detected
- Express bits analysis not accurate
- Injection in not-sensitive bits:
  - No mismatch found

Resource type	Sensitive bits	
	Susanna	fault injection
Logic – CC	284	154
Routing – Local/CC	36	22
Routing – Express/CC	114	2
Routing – H repeater	110	75
Routing – V repeater	124	86

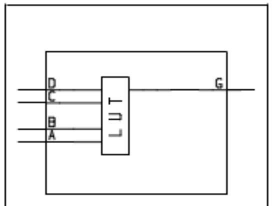
Macro Study

II

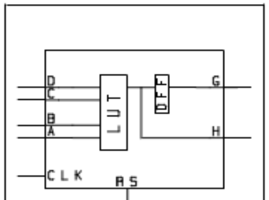
Study of the use of optimized macros

Atmel technology library

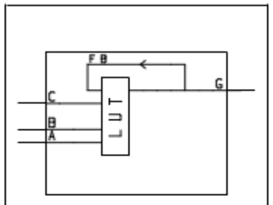
- Functional macros are components with fixed functionality (Gate, IO, RAM...)
- Dynamic macros are designed to given better control over the implementation of specific function in a single logic cell



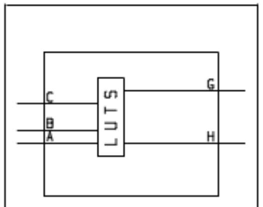
FGEN1



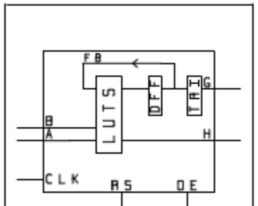
FGEN1R



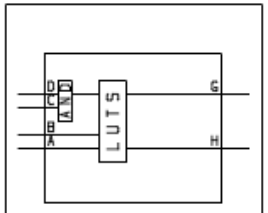
FGEN1F



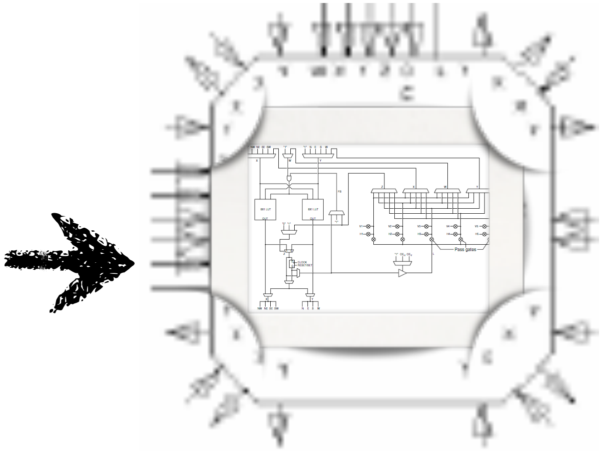
FGEN2



FGEN2RFT



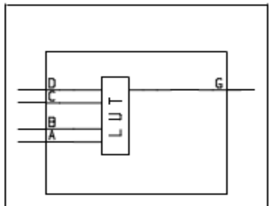
MGEN



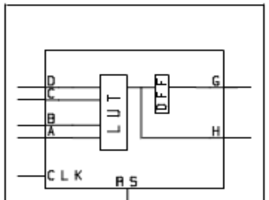
Atmel technology library

European Space Agency

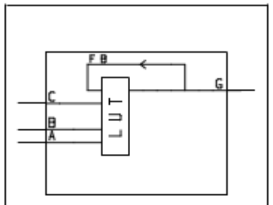
- Functional macros are components with fixed functionality (Gate, IO, RAM...)
- Dynamic macros are designed to given better control over the implementation of specific function in a single logic cell



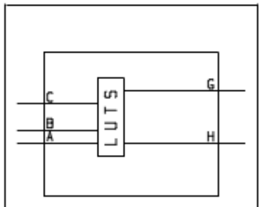
FGEN1



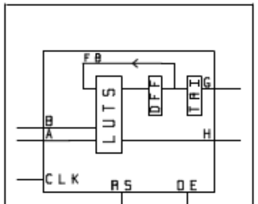
FGEN1R



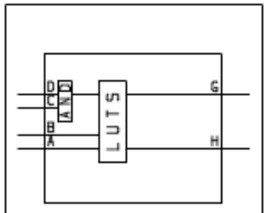
FGEN1F



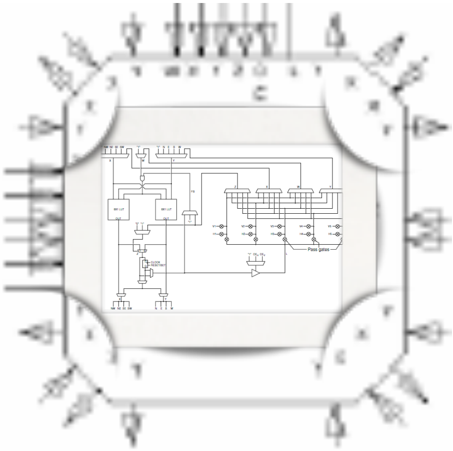
FGEN2



FGEN2RFT



MGEN

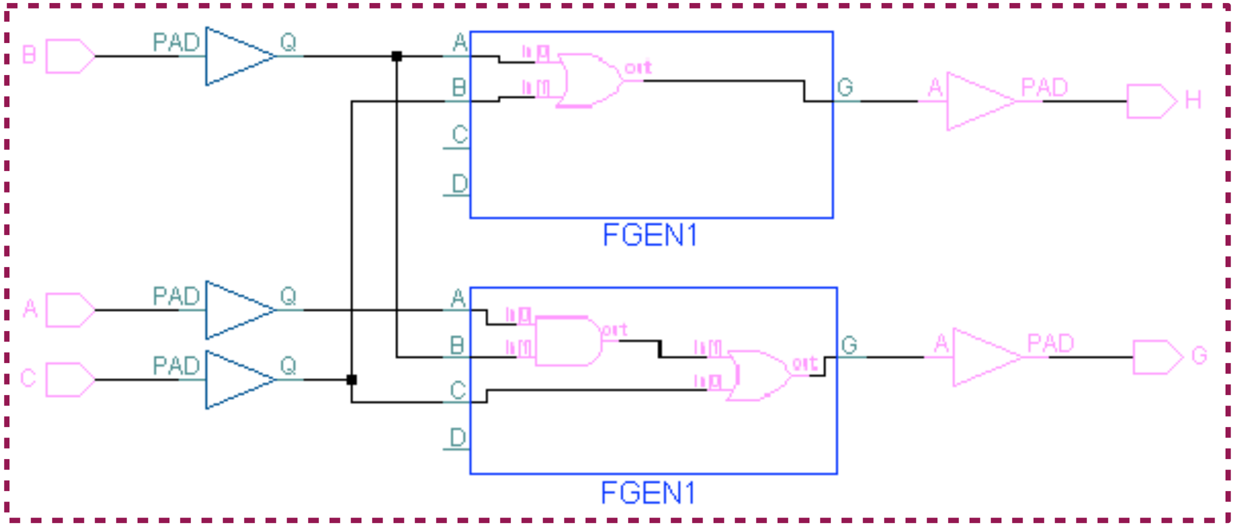


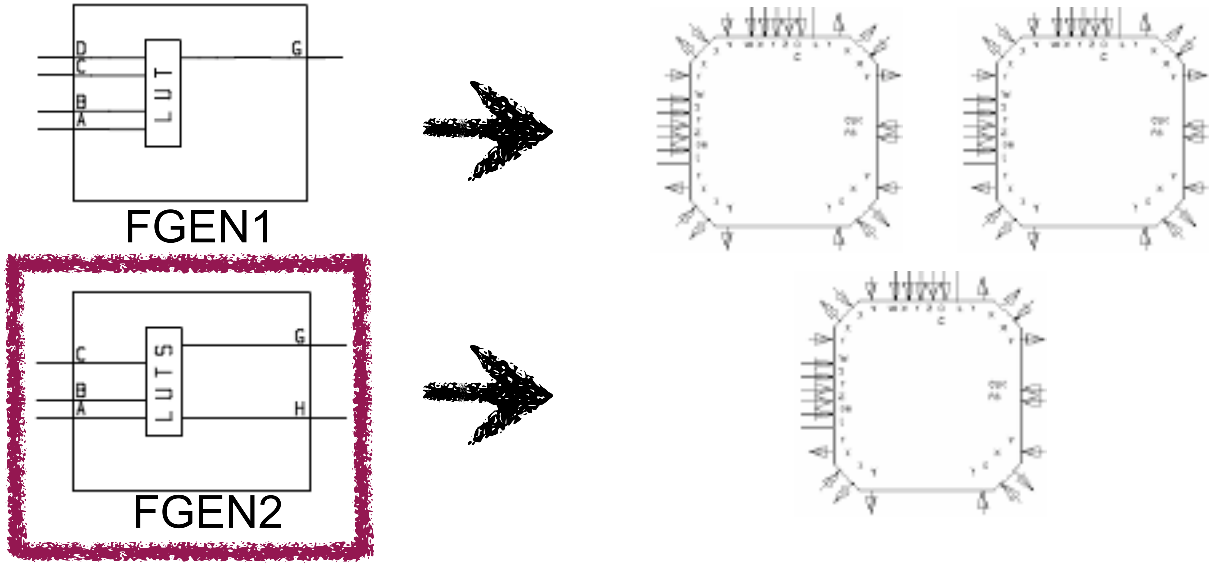
Not implemented

Motivation

Circuit:

G<= (A*B) + C
H<= (A+B)



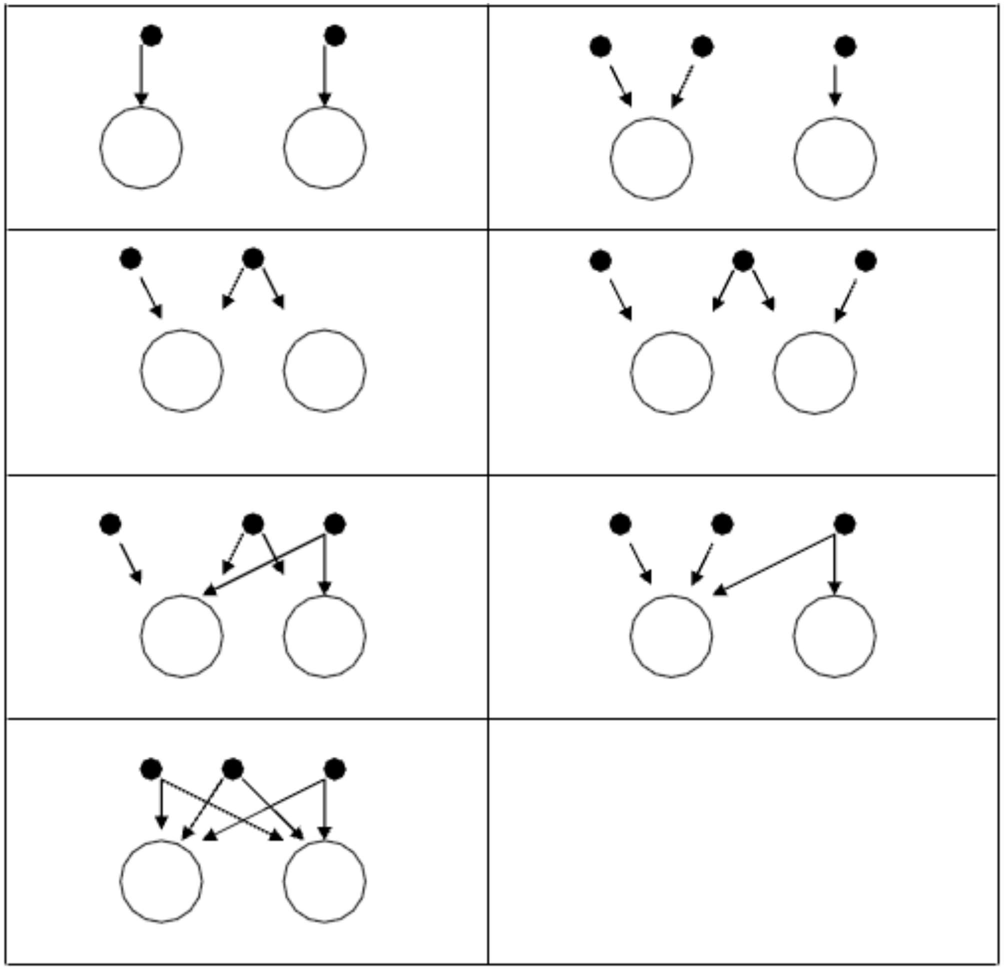


Manual optimization

- Custom logic functions in single logic cell can be created like macros by means of Figaro's "macro generator"
- The macros are stored in the user library
- The user have to change the VHDL description to use user-macro as black box during the synthesis

Analysis

Configurations optimization:



- 2 FGEN1->1 FGEN2

FGEN1:
1 function
4 input

FGEN2:
2 functions
3 input

- Area occupancy
- Memory usage and criticality from Susanna/Jonathan

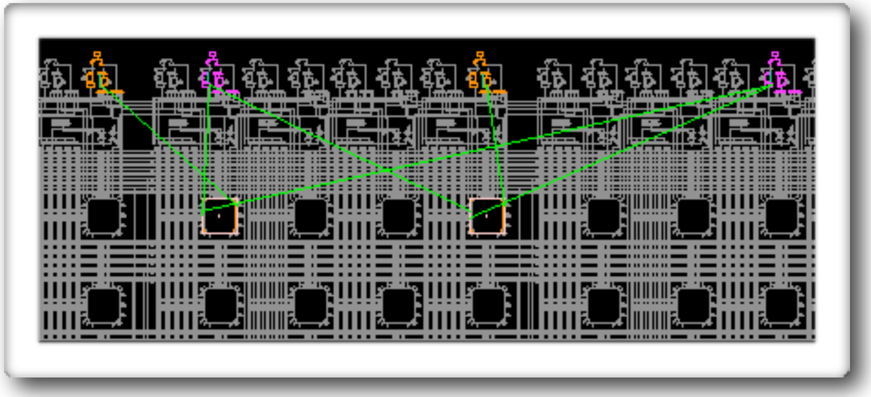
○ function

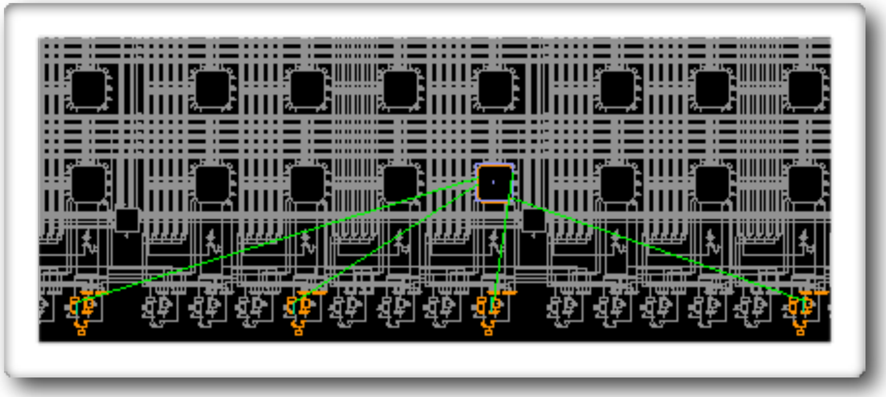
● Input

esa European Space Agency

Manual Analysis Results

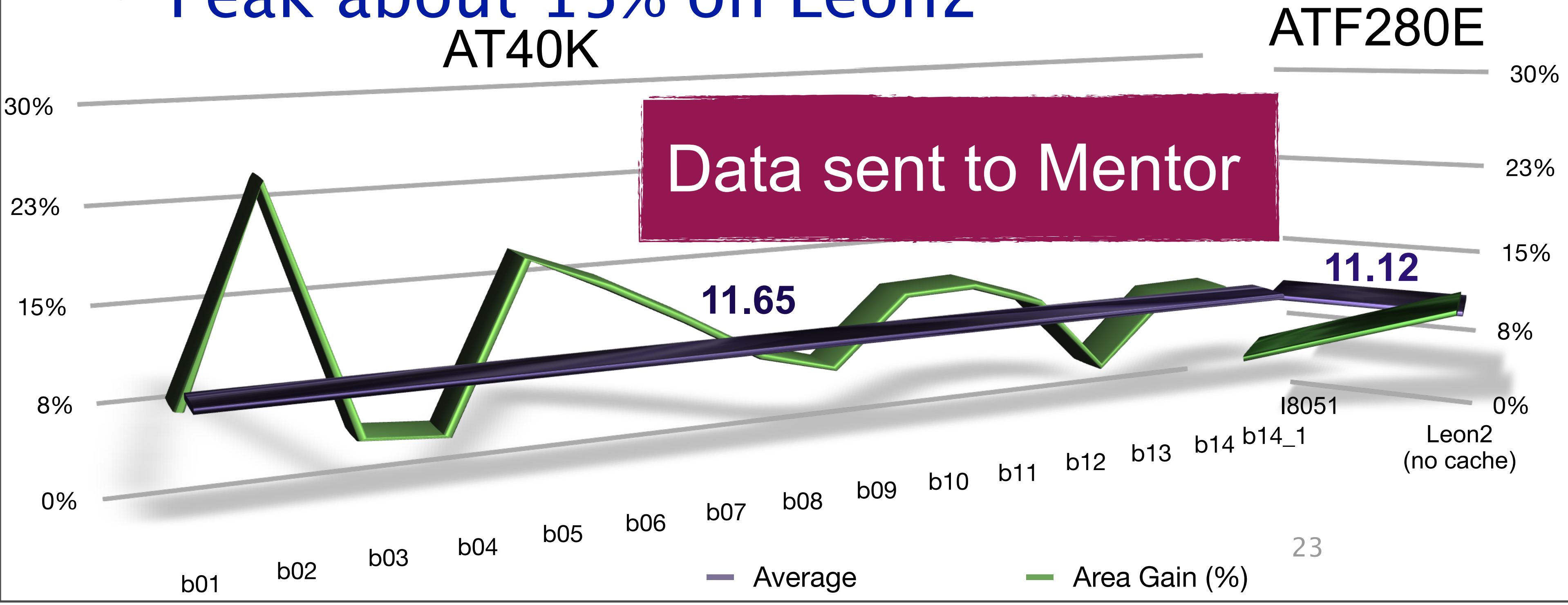
Function	used logic cell		Sensitive bits	
	Original [#]	Gain [%]	Original [#]	Gain [%]
And-or	2	50%	52	44%
bits-counter	27	0%	977	2%
multiple bits-counter	77	1.31%	3721	7%





Automatic Analysis

- Analysis of EDIF netlist using PAHT tool
- Realistic circuit under test
- More than 11% average area gain
- Peak about 15% on Leon2



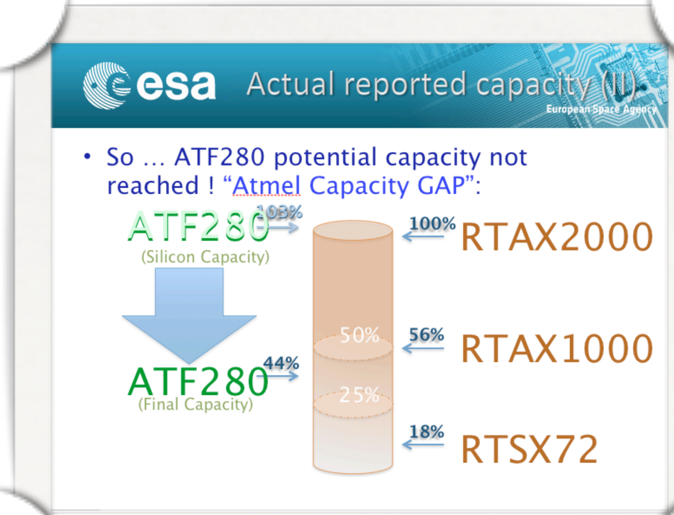
Data sent to Mentor

III

Platform to apply a new placement algorithm on the Atmel AT40K, ATF280E and ATFS450 series

Motivation

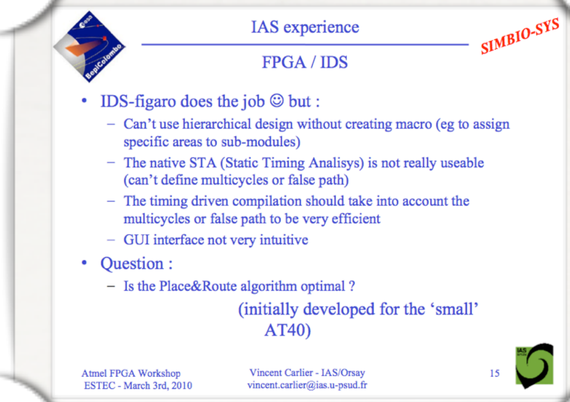
- Router can only work as good as the place
- Figaro doesn't always achieve fulfilling solution
- Feedback from industries



Actual reported capacity (II)

European Space Agency

- So … ATF280 potential capacity not reached ! "Atmel Capacity GAP":



IAS experience

FPGA / IDS

SIMBIO-SYS
BepiColombo

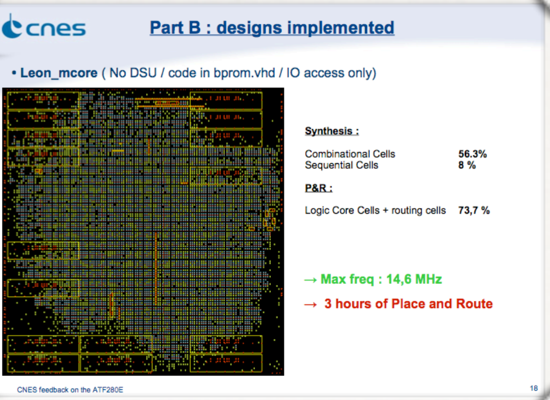
- IDS-figaro does the job ☺ but :
  - Can't use hierarchical design without creating macro (eg to assign specific areas to sub-modules)
  - The native STA (Static Timing Analisys) is not really useable (can't define multicycles or false path)
  - The timing driven compilation should take into account the multicycles or false path to be very efficient
  - GUI interface not very intuitive
- Question :
  - Is the Place&Route algorithm optimal ?

(initially developed for the 'small' AT40)

Atmel FPGA Workshop
ESTEC - March 3rd, 2010

Vincent Carlier - IAS/Orsay
vincent.carlier@ias.u-psud.fr

15



Part B : designs implemented

- **Leon_mcore** (No DSU / code in bprom.vhd / IO access only)

Synthesis :

Combinational Cells	56.3%
Sequential Cells	8 %

P&R :

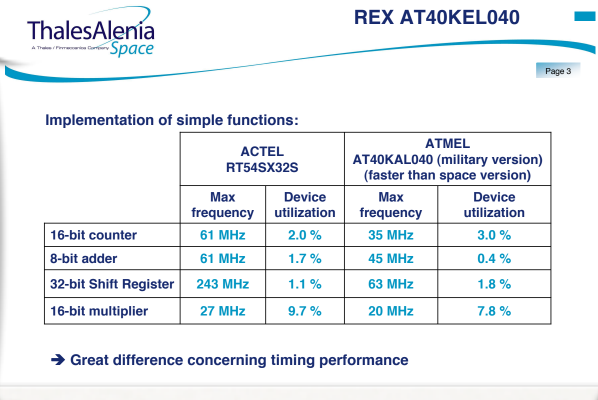
Logic Core Cells + routing cells	73,7 %

→ Max freq : 14,6 MHz

→ 3 hours of Place and Route

CNES feedback on the ATF280E

18



ThalesAlenia Space
A Thales / Finmeccanica Company

REX AT40KEL040

Page 3

Implementation of simple functions:

	ACTEL RT54SX32S		ATMEL AT40KAL040 (military version) (faster than space version)	
	Max frequency	Device utilization	Max frequency	Device utilization
16-bit counter	61 MHz	2.0 %	35 MHz	3.0 %
8-bit adder	61 MHz	1.7 %	45 MHz	0.4 %
32-bit Shift Register	243 MHz	1.1 %	63 MHz	1.8 %
16-bit multiplier	27 MHz	9.7 %	20 MHz	7.8 %

➔ Great difference concerning timing performance

Motivation

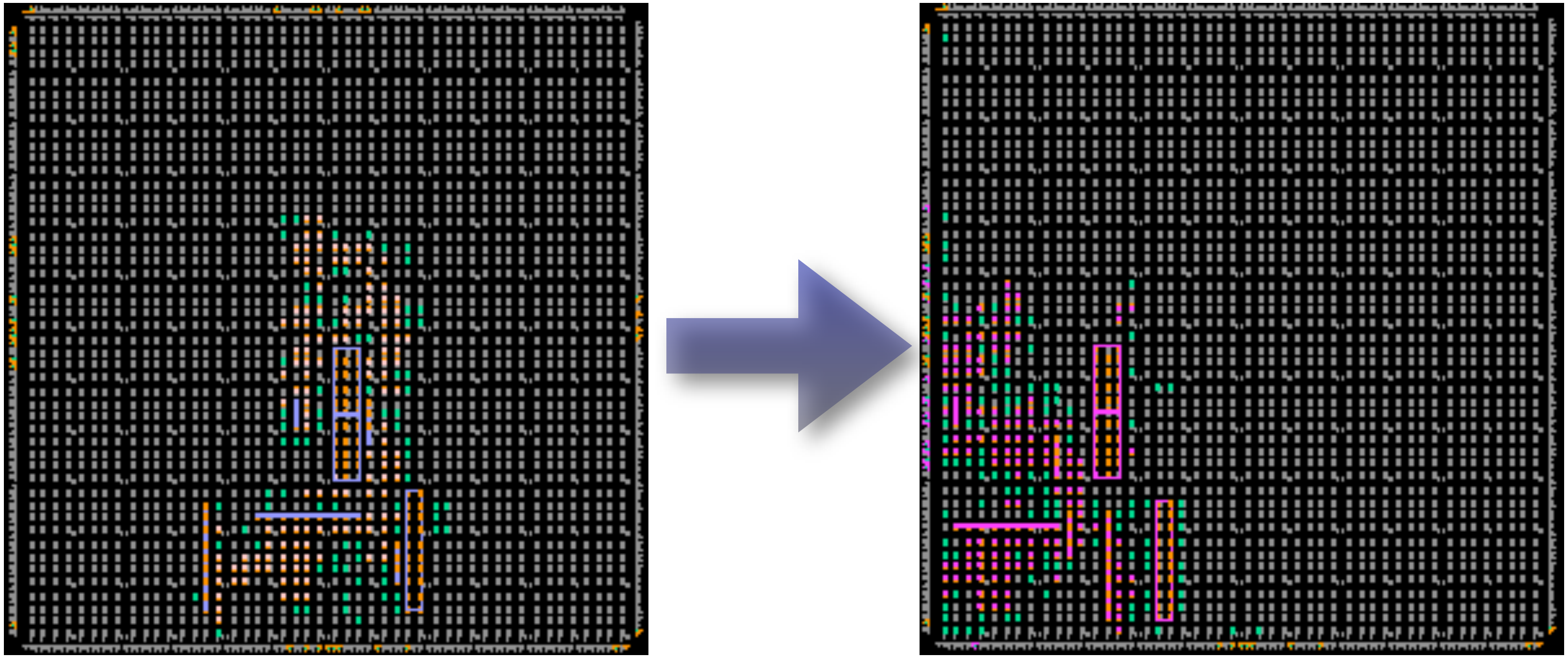
Atmel Figaro tool support manual placement:

- to improve timing on a path
- to squeeze a design that automatic placement can't make it fit in the device
- to resolving contention
- to make easier to route a net
- to reserve a particular area for an instance

"Only optimize placement manually if you are familiar with the device architecture and know how your design should us it."
Figaro Help

Motivation–Manual effort

	Atmel's P&R		Manual Placement		Gain		Time
	Max frequency	Device utilization	Max frequency	Device utilization	Frequency	Device utilization	[h]
Memory elaboration	7.1 MHz	11.4%	10.4 MHz	12%	32%	5%	8h
			10.5 MHz	11.8%			+2h
Scruble string	10.9 MHz	6%	15 MHz	8.5%	27%	29%	6h
1-player game	11.5 MHz	15%	15.3 MHz	20.3%	25%	26%	5h



memory elaboration design

PURPLE

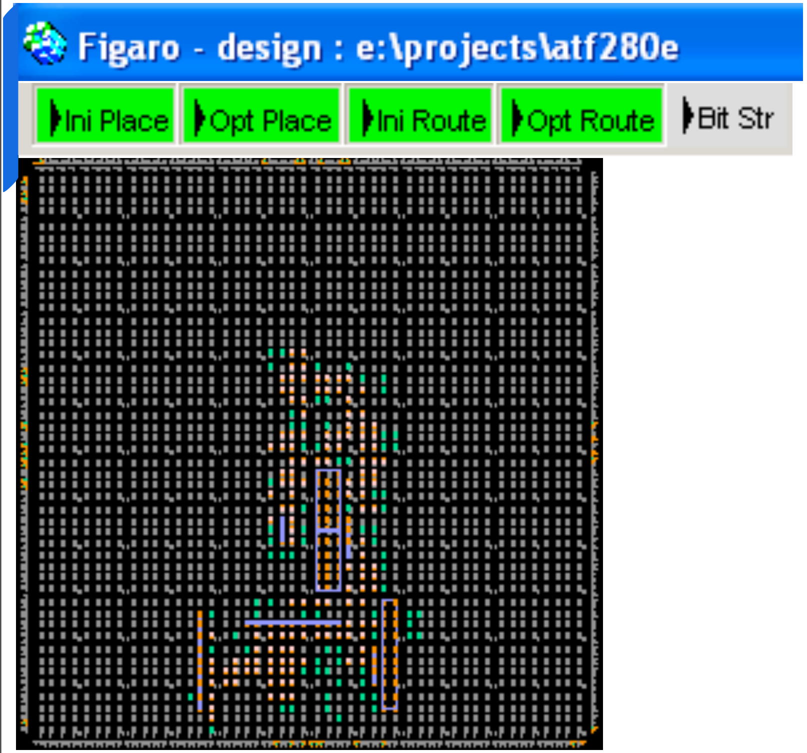
- Platform to apply a new placement algorithm on Atmel AT40K, AT280E and ATFS450 series
- Independent platform
  - use of others algorithms in order to improve particular features of the design
  - use a set of constrains to drive the placement
- No external constrains are supported to drive placement in Atmel Figaro tool

Platform to Upgrade and Redo Placement Leading Efficiency

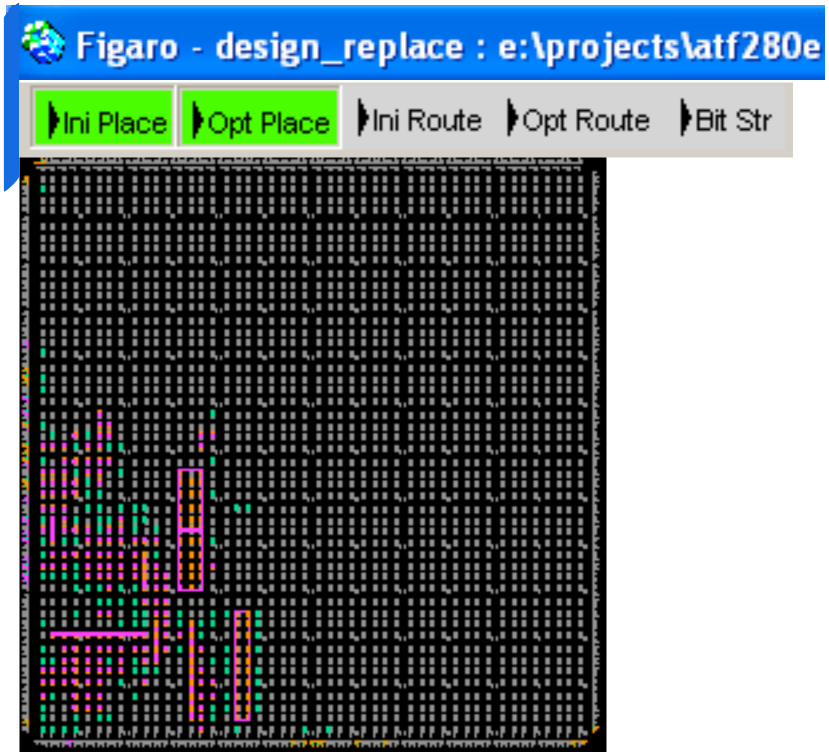
PURPLE

- This platform could work in parallel with Figaro tool in order to
  - use Figaro’s solution as starting point
  - lets Figaro to provide to the routing step and the bitstream generation.
- The platform would be able to:
  - read a project Figaro file and extract the netlist
  - give info about the Figaro placement
  - contain a re-place engine
  - create a project Figaro file implementing the new solution

PURPLE



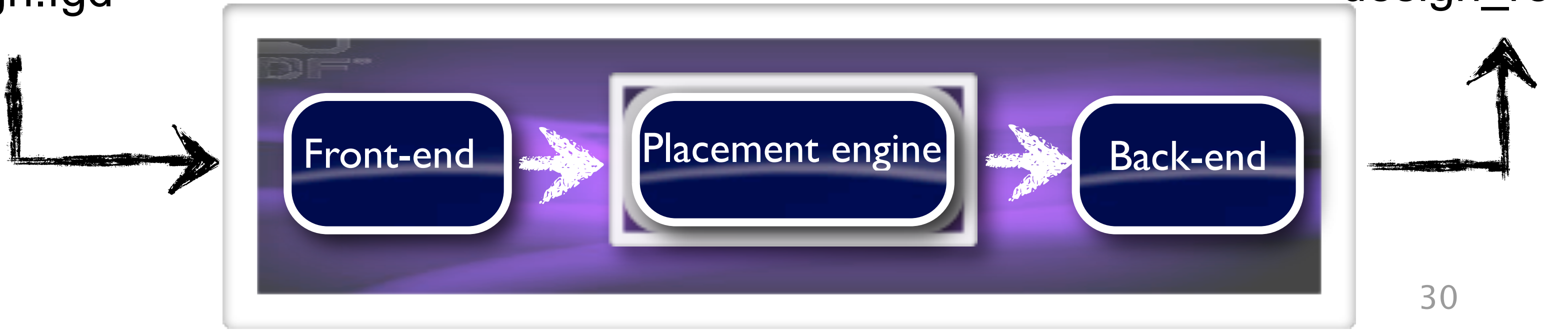
Figaro - design : e:\projects\atf280e



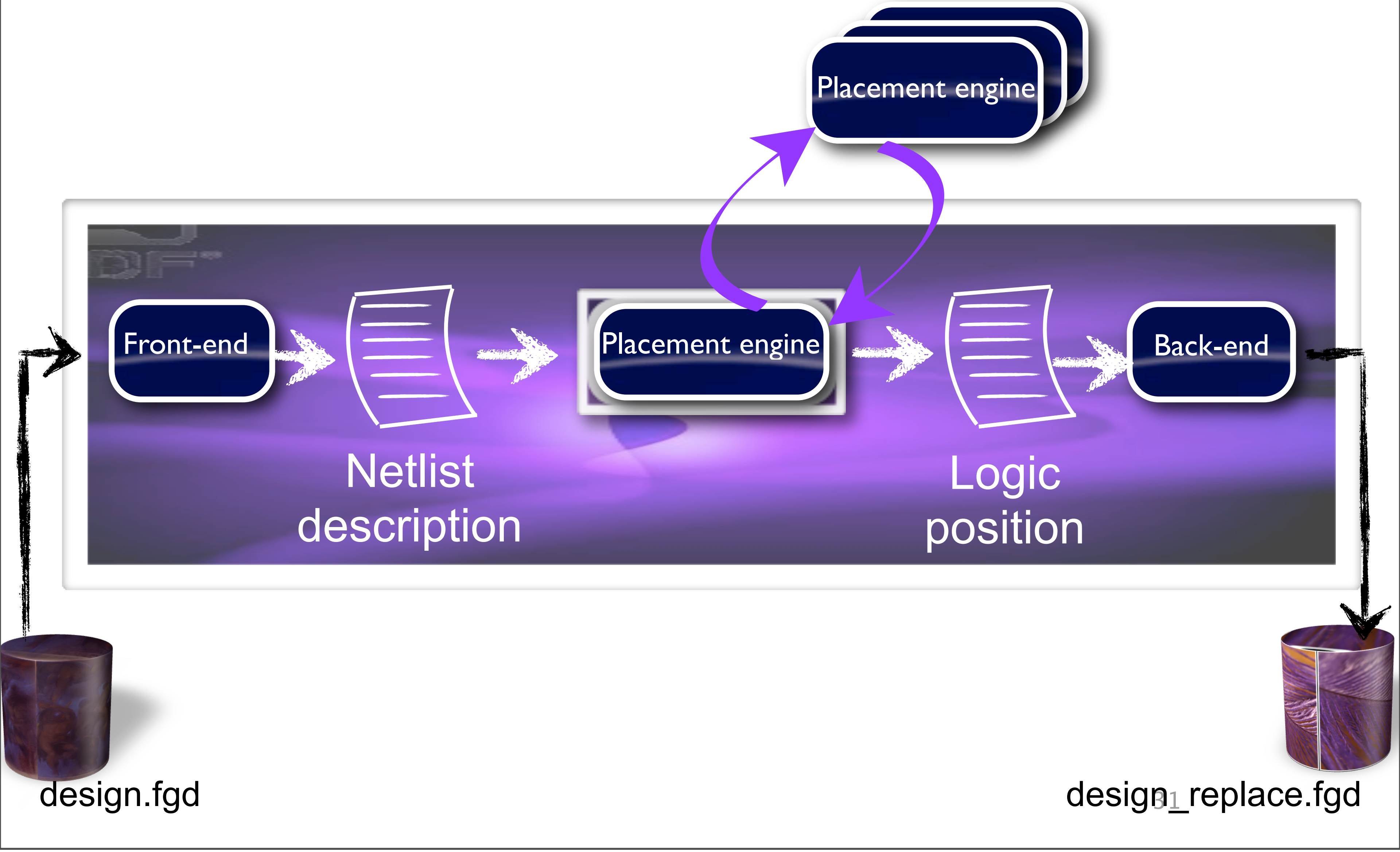
Figaro - design_replace : e:\projects\atf280e

design.fgd

design_replace.fgd



PURPLE


Placement engine
Front-end
Placement engine
Back-end
Netlist description
Logic position
design.fgd
design_replace.fgd

Future Works

- Implement a timing-driven placement algorithm
- Add placement constrains
- Test the new algorithm vs Figaro’s one

Conclusion

- Spread of circuit race the sensitiveness
  - Macro study : more logic in a single resource
    - shared of input pad -> less routing
    - less occupied area -> less surrounding influential resources (that can interfere)
  - Placement study
    - connected resource placed near could use direct connection -> less routing
    - short path between I/O and resource -> less routing

Thanks for the attention