



Deliverable – D5

Specification of characterization for the additional asynchronous standard cells for INFINEON 40 nm CMOS process

Grant Agreement No:	214364
Project acronym:	GALAXY
Project title:	GALS InterfAce for CompleX Digital System Integration
Funding Scheme:	STREP
Date of latest version of Annex I against which the assessment will be made:	28.10.2008.
Contractual Date of Delivery to the EC:	31. Jan. 09
Actual Date of Delivery to the EC:	31. Mar. 09 (approved by Project Officer)
Author(s):	Christoph Heer, Jan Dienstuhl (INFINEON), Milos Krstic (IHP), Sean Salisbury (STX)
Participant(s):	INFINEON, IHP, STX
Work Package:	WP3
Security:	Public
Nature:	Report
Version:	1
Total number of pages:	21

Abstract:

This is a report that corresponds to the prototype deliverable - Specification of characterization for the additional asynchronous standard cells for INFINEON 40 nm CMOS process.

It includes a general introduction on the standard cell characterization procedure and explains specifics for the asynchronous extended cell set. In the report the differences between standard state-holding c-element, mutual exclusion elements and toggle elements will be described. Specific rules and guidelines for cell characterization will be explained

Keyword list: asynchronous design, standard cell library, C-element architecture, cell characterization, timing constraints, power characterization



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 31/03/2009
Issue: 1

Function	Responsibility	Date	Signature
Written by:	Christoph Heer, Jan Dienstuhl, Milos Krstic, David Jackson	31.3.2009	
Checked by:			
Approved by:			

Reserved to EC

Approved by:			
---------------------	--	--	--



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA





GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 31/03/2009
Issue: 1

CHANGE RECORDS

<i>ISSUE</i>	<i>DATE</i>	<i>§ : CHANGE RECORD</i>	<i>AUTHOR</i>
1	13-March-09	Initial description of cell characterization	Christoph Heer
2	27-March-09	First version for project internal review	Christoph Heer
3	30-March-09	Included introduction and flow graphic	Christoph Heer
4	31-March-09	Final version for release to Project Officer	Christoph Heer



BIBLIOGRAPHIC RECORD

Project Number:	214364 GALAXY
Project Title:	GALAXY
Deliverable Type:	Report
Deliverable Number:	D5
Contractual Date of Delivery:	31. Jan. 2009
Actual Date of Delivery:	31. Mar. 2009
Title of Deliverable:	Specification of characterization for the additional asynchronous standard cells for INFINEON 45 nm CMOS process
Work package contributing to the Deliverable:	WP3
Authors:	Christoph Heer, Jan Dienstuhl, David Jackson
Abstract	This is a report that corresponds to the prototype deliverable - Specification of characterization for the additional asynchronous standard cells for INFINEON 40 nm CMOS process. It includes a general introduction on the standard cell characterization procedure and explains specifics for the asynchronous extended cell set. In the report the differences between standard state-holding c-element, mutual exclusion elements and toggle elements will be described. Specific rules and guidelines for cell characterization will be explained
Keywords	asynchronous design, standard cell library, C-element architecture, cell characterization, timing constraints, power characterization
Confidentiality Level	Public
Name of Client:	EC
Distribution List:	GALAXY, EC, internet
Authorised by:	Milos Krstic
Issue:	1
Document ID:	D5
Total Number of Pages:	
Contact Details:	krstic@ihp-microelectronics.com



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 31/03/2009
Issue: 1

TABLE OF CONTENTS

1	INTRODUCTION	7
1.1	STANDARD CELLS AND CHARACTERIZATION	7
1.2	STANDARD CELLS AND GALS	9
2	CELL LIBRARY CHARACTERIZATION	10
2.1	COMBINATIONAL CELLS	10
2.2	<i>SEQUENTIAL CELLS</i>	11
3	CHARACTERIZATION OF ASYNCHRONOUS CELLS	13
3.1	C-ELEMENTS	13
3.2	TOGGLE-CELL	16
3.3	MUTEX	19
3.4	DELAY ELEMENTS.....	21



LIST OF FIGURES

Figure 1:	Digital semi-custom design and verification flow	8
Figure 2:	Timing arcs for combinational cells	10
Figure 3:	Timing arc for sequential cells.....	11
Figure 4:	Timing arcs for c-element.....	13
Figure 5:	Layout of initial c-element cell UANCS2X020 (without reset)	13
Figure 6:	Layout of initial c-element cell UANCSR2X020 (c-element with reset)	14
Figure 7:	Timing arcs for complete toggle cell.....	16
Figure 8:	Layout of toggle cell UACTOX020	16
Figure 9:	Simulated timing of toggle cell UACTOX020	17
Figure 10:	Timing arcs for mutex element.....	19
Figure 11:	Layout of initial mutex element UAMUTEX020.....	19
Figure 12:	Layout of initial mutex element UAMUTEX020.....	20



1 INTRODUCTION

1.1 STANDARD CELLS AND CHARACTERIZATION

Standard cells are used in the majority of semi-custom System-on-Chip (SoC) digital design flows. They encapsulate simple digital logic functions that are used repeatedly to build up the higher level, complex functionality of a design. Their functionality and properties are abstracted from the underlying transistor level implementation using logic models for function and simplified models for other properties, such as timing and power.

This abstraction and simplification of functionality and property, is what allows Electronic Design Automation (EDA) tools to produce complex, multi-million gate designs. A particular digital design flow may also employ higher levels of abstraction, but standard cells are seen as leaf components and, as such, usually form the lowest level that digital EDA tools deal with.

The creation of these simplified standard cell property models is called characterization. Once the physical implementation of each standard cell is completed, a lower level more complex model of the underlying components within the standard cell, such as transistors, resistors and capacitors is extracted from the layout. This model is then run through a simulator that models closely how the implementation will behave on the die. These accurate models and simulators operate at what is generally known as the SPICE level and take into account analogue effects and complex interaction of the components and wires that make up the standard cell implementation.

To generate the abstracted, simplified models that digital design tools can use, the lower level SPICE simulations are run at a set of expected operating conditions and the results captured as either lookup tables, equations or a mix of both. The operating conditions include shared attributes such as voltage supply, temperature or process quality to parameters that affect an individual cell in the design such as the output load it is driving or the worst case input slew that will be applied to its inputs. There is a limit to the accuracy of the simplified model and if the standard cell is used in conditions beyond that for which it was characterized, then the model may fail to reach the required level of accuracy to maintain yield.

Once each standard cell has been characterized, the abstracted models are placed in a library, or set of libraries, that the digital EDA tools can read when creating or checking functionality. There are many vendor specific library formats in commercial use, but the "Liberty" format, originally designed by Synopsys, is understood by the majority of digital tools. These library files contain properties and models such as Boolean functionality, area, pin-pin delay, power and noise. They can also define properties such as minimum set-up and hold that can be used by tools that check the design will function correctly after fabrication at the required operating conditions.

The type of digital design and checking tools that use this cell property information are:

- Synthesis tools that can generate low level standard cell connectivity descriptions from a higher abstracted description of the required functionality
- Place and Route tools that need to understand cell timing, drive strength, area etc.
- Timing engines that can statically check timing including cross talk effects or create delays for back-annotation onto dynamic simulation.
- Power analysis tools that can take the cell power models, a design netlist and an activity description to generate accurate leakage and dynamic power reports and thermal graphs.
- Functional equivalence tools that check a higher level functional description matches the functionality of the standard cell connectivity implementation netlist.



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 31/03/2009
Issue: 1

To maintain a high level of manufacturing yield as the minimum feature size shrinks to 65nm, 40nm and below, more complex standard cell models have become necessary to allow existing and new tools to take into account effects that could safely be ignored at larger geometries. Here there is a balance between the accuracy of modelling these effects for yield and the run times of the tools used to create and check the digital circuit. For instance, Composite Current Source (CCS) and Effective Current Source Models are beginning to augment or replace the older Non Linear Models for delay and power.

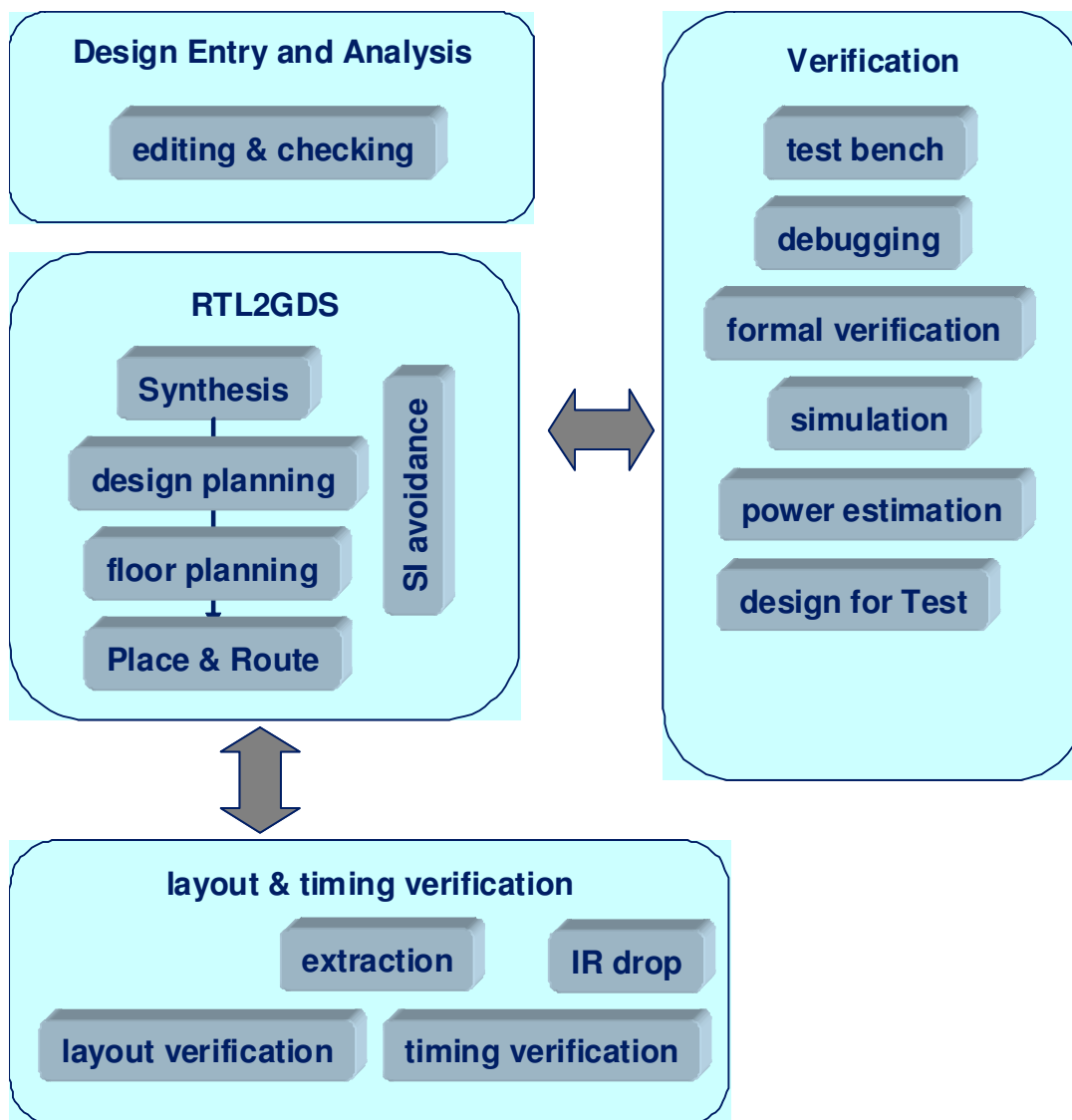


Figure 1: Digital semi-custom design and verification flow



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 31/03/2009
Issue: 1

1.2 STANDARD CELLS AND GALS

Certain GALS methodologies employ functionality that is purely asynchronous in nature so that portions of the design operate without a clock. This functionality can usually be created using the stock set of cells in a generic standard cell library. However, better performance, power, area or reliability can often be achieved with a few additional standard cells that complement the stock library. These additional cells then must be characterized in the same way as the existing set of cells so that the digital EDA tools can employ them and check designs using these new cells in the same way as the existing cells.

As such any GALS designs using these additional cells will require these functional and property models for each new cell. Any tool that can automatically synthesize higher level functionality or place and route the design or check a specific property of the design will have the necessary information for the new cells in the required format.

These additional standard cells have unusual functionality with respect to the stock set of cells found in normal digital libraries. This can create difficulties when attempting to characterize their properties either by hand or using an automatic characterization tool. The nature of the cells functionality must be understood and the properties of interest must be captured in a format that will allow the digital EDA tools to deal with them properly.

This document describes the characterization of these additional cells made available to the GALAXY project partners in the Infineon 40nm process



2 CELL LIBRARY CHARACTERIZATION

Cell characterization is a standard step in cell library development and essential to use the cell in synthesis based semi-custom design flows. Based on an extracted netlist, the physical characterization of the cell is derived with an analog SPICE simulator. There are significant differences for pure combinational logic and sequential, that means state-holding cells. To understand the specific difficulties to characterize the cells of the asynchronous library extension, this chapter will briefly introduce the characterization of a standard cell set.

2.1 COMBINATIONAL CELLS

Combinational cells implement Boolean functions. The output signal is directly following input signal changes according to the implemented logic function. With respect to timing behaviour combinational cells can be easily characterized for their input output behaviour via so called timing arcs. A 3-input gate is sketched below.

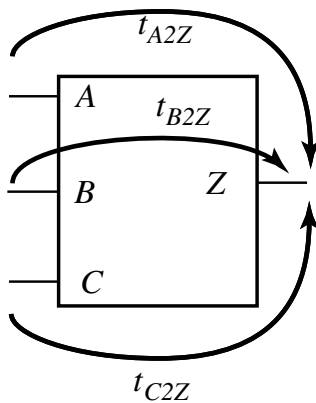


Figure 2: Timing arcs for combinational cells

There are three timing arcs from the inputs A, B and C to the output Z. Timing arcs are characterized with respect to their delay of an input change to the caused and corresponding output change. Assume this cell to be a 3-input NAND gate. Two inputs have to be stable on high level (input conditions) to allow the third input to switch the output. If the third input goes from low to high. The output will go from high to low. If the third input goes low to high, the output will follow with a high to low transition. These transitions are characterized for all three inputs with respect to delay and power consumption.

```
delay
{
  timing_arcs
  {
    A      to      Z      INV  @      B=H, C=H
    B      to      Z      INV  @      A=H, C=H
    C      to      Z      INV  @      A=H, B=H
  }
}
```



As there are no other input transitions causing output changes, this characterizes the cell completely with respect to delay. But as other input transitions may cause power consumption due to cell internal switching even without output changes, there are additional simulations required. This is reflected in the additional power arcs, where power is consumed, but the output stays constant.

```
power
{
  power_arcs
  {
    A      BOTH  nochangeZ=H      @      B=0, C=0
    A      BOTH  nochangeZ=H      @      B=0, C=1
    A      BOTH  nochangeZ=H      @      B=1, C=0
    B      BOTH  nochangeZ=H      @      A=0, C=0
    B      BOTH  nochangeZ=H      @      A=1, C=0
    B      BOTH  nochangeZ=H      @      A=0, C=1
    C      BOTH  nochangeZ=H      @      A=0, B=0
    C      BOTH  nochangeZ=H      @      A=0, B=1
    C      BOTH  nochangeZ=H      @      A=1, B=0
  }
}
```

2.2 SEQUENTIAL CELLS

Sequential cells have a state holding function. Only after the rising edge of the clock signal CP the logic value at input D is transferred to output Q. Therefore only the timing arcs from clock to output have to be characterized.

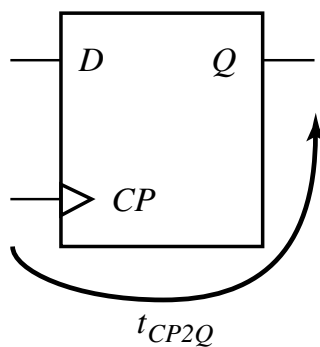


Figure 3: Timing arc for sequential cells

```
delay
{
  timing_arcs
  {
    # CP to Q
    CP    LH      @      D=L
    CP    LH      to    Q    LH    @      D=H
    CP    LH      to    Q    HL    @      D=L
  }
}
```



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 31/03/2009
Issue: 1

But in addition to timing arcs there are additional constraints, which have to be met to have valid timings. In detail the input signal D has to be stable a certain period before the clock signal transition (setup time) and it has to stay stable until a certain time after the clock signal transition (hold time). The clock signal itself has to fulfil certain constraints with respect to the minimum pulse width (mpw). Also these timing constraints have to be measured via SPICE simulation and characterized.

```
constraint
{
    constraint_arcs
    {
        setup D LH vs CP LH MEAS CP to SF2 LH
        hold CP LH vs D HL MEAS CP to SF2 LH
        setup D HL vs CP LH MEAS CP to SF2 HL
        hold CP LH vs D LH MEAS CP to SF2 HL

        mpw CP L MEAS CP LH to SF2 LH @ D=1
        mpw CP H MEAS CP LH to SF2 LH @ D=1
        mpw CP L MEAS CP LH to SF2 HL @ D=0
        mpw CP H MEAS CP LH to SF2 HL @ D=0
    }
}
```

Finally also the input transitions, which do not generate an output transition, but cause internal switching have to be characterized with respect to power consumption.

```
power
{
    power_arcs # smart conditional
    {
        CP LH @ @ D=L
        CP HL nochangeQ=0 @ D=0
        D BOTH nochangeQ=0 @ CP=0
        CP LH nochangeQ=0 @ D=0
        D BOTH nochangeQ=0 @ CP=1
        CP HL nochangeQ=0 @ D=1
        CP LH @ @ D=H
        CP HL nochangeQ=1 @ D=1
        D BOTH nochangeQ=1 @ CP=0
        CP LH nochangeQ=1 @ D=1
        D BOTH nochangeQ=1 @ CP=1
        CP HL nochangeQ=1 @ D=0
    }
}
```



3 CHARACTERIZATION OF ASYNCHRONOUS CELLS

For the characterization of the asynchronous cells we do not consider any combinational cells as they are characterized by the standard methodology as described before. But we have to define specific ways to characterize C-elements as state-holding cells, toggle elements as a special case of state-holding element and Mutex-cells because of their inherently unpredictable timing.

3.1 C-ELEMENTS

C-elements have 5 classes of inputs: symmetric, up, down, set and reset. For the description of the characterization in this document we just focus on the symmetric inputs and the reset. The other inputs (up, down) will be characterized accordingly. The output of a C-element rises (for an active-high gate) when all inputs are high. The output then falls when all inputs are low. Reset inputs force the output low irrespective of the state of other inputs. All inputs are active high unless otherwise indicated (by „b” or “<x>N” terms).

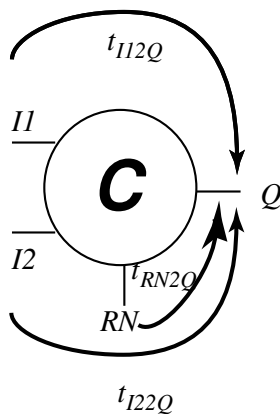


Figure 4: Timing arcs for c-element

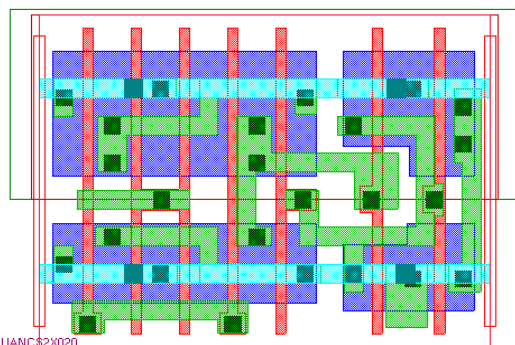


Figure 5: Layout of initial c-element cell UANCS2X020 (without reset)

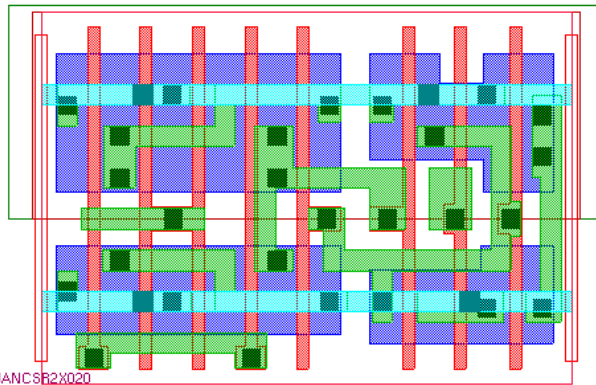


Figure 6: Layout of initial c-element cell UANCSR2X020 (c-element with reset)

The timing of the c-element is easily described by an input transition, while the other input is stable and the reset is inactive. Therefore we have four timing arcs to describe for a 2-input c-element. In addition, we characterize the timing from reset active to output signal as given below in a characterization template file.

```
delay
{
  timing_arcs
  {
    R      to      Q      NIVN      @      I1=L, I2=L
    I1     LH      to      Q      HL      @      I2=H, R=H
    I1     HL      to      Q      LH      @      I2=L, R=H
    I2     LH      to      Q      HL      @      I1=H, R=H
    I2     HL      to      Q      LH      @      I1=L, R=H
  }
}
```

As the c-element is a state-holding cell like a sequential flip-flop or register, we may also characterize the setup time and a minimum pulse width. For many asynchronous design styles this may seem unnecessary as it should be guaranteed, that the circuit works monotonously (no input change without output change of the cell) and that no spikes or glitches (small unintended pulses) will occur. Nevertheless, we want to provide this characterization as it might help to verify the final asynchronous circuits and state machines. Static timing analysis may then be used to verify whether there's real monotonous behaviour and if the circuit is really glitch-free.



GALAXY

GALS InterfACE for CompleX Digital SYstem Integration

Confid. Level: Public
Date : 31/03/2009
Issue: 1

```
constraint {
  constraint_arcs {
    setup I2 LH vs I1 HL MEAS I2 to Q HL @ R=H
    setup I1 LH vs I2 HL MEAS I1 to Q HL @ R=H
    setup I2 HL vs I1 LH MEAS I2 to Q LH @ R=H
    setup I1 HL vs I2 LH MEAS I1 to Q LH @ R=H

    mpw I1 L MEAS I1 HL to Q LH @ I2=L, R=H
    mpw I1 H MEAS I1 LH to Q HL @ I2=H, R=H
    mpw I2 L MEAS I2 HL to Q LH @ I1=L, R=H
    mpw I2 H MEAS I2 LH to Q HL @ I1=H, R=H
    mpw R L MEAS R HL to Q HL @ I1=H, I2=L
    mpw R L MEAS R HL to Q HL @ I1=L, I2=H
  }
}
```

Finally even for C-elements we have to characterize power consumption in case of input changes without any output change.

```
power
{
  power_arcs # smart conditional
  {
    R BOTH nochangeQ=0 @ I1=0, I2=1
    R BOTH nochangeQ=0 @ I1=1, I2=0
    R BOTH nochangeQ=0 @ I1=1, I2=1
    I1 BOTH nochangeQ=0 @ R=0, I2=0
    I1 BOTH nochangeQ=0 @ R=0, I2=1
    I1 BOTH nochangeQ=0 @ R=1, I2=1
    I2 BOTH nochangeQ=0 @ R=0, I1=0
    I2 BOTH nochangeQ=0 @ R=0, I1=1
    I2 BOTH nochangeQ=0 @ R=1, I1=1
    I1 LH nochangeQ=0 @ R=1, I2=0
    I1 BOTH nochangeQ=1 @ R=1, I2=0
    I2 BOTH nochangeQ=1 @ R=1, I1=0
  }
}
```



3.2 TOGGLE-CELL

For the toggle-cell we will only characterize the complete toggle as it is the finally used cell. Even as consisting of two half-toggle elements, the timing is much easier to characterize for the complete toggle. Otherwise the timing verification tools would require additional runtime to compute the final timing based on rather complex half-toggle modules and the inherent local feedback of the toggle cell. Feedback structures are often the root cause for poor or slow convergence of circuit simulators and timing analysis.

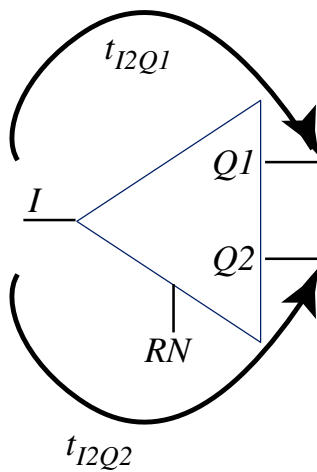


Figure 7: Timing arcs for complete toggle cell

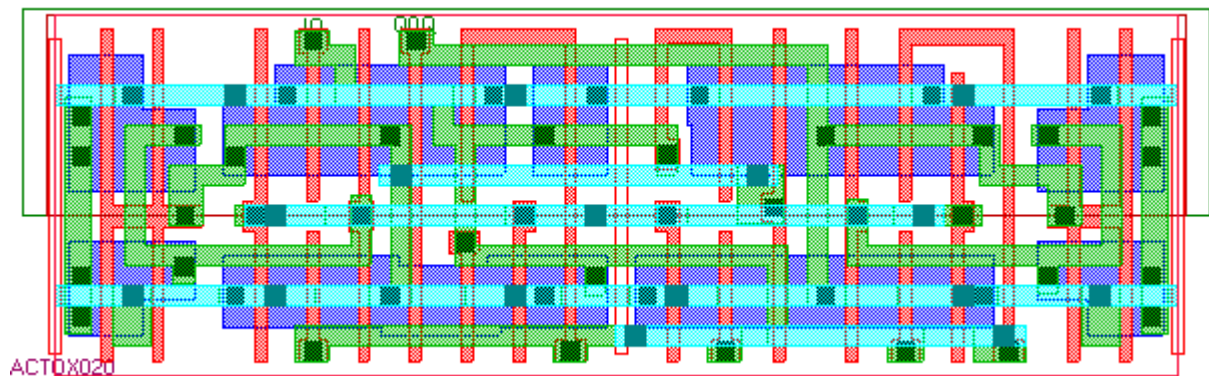


Figure 8: Layout of toggle cell UACTOX020



The timing of the toggle element is sketched in the simulated waveform below. After initial reset (low active RN signal, brown wave form) both outputs Q1 (blue) and Q2 (pink) are reset to zero.

Raising the input signal D from low to high, output Q2 will raise. Lowering input signal D again, output Q1 will raise. The next pulse at input D will then drive Q2 low again on the rising edge of the pulse and drive Q1 low again on the falling edge of the pulse. All transitions will be characterized.

In addition the timing of the reset of the toggle is characterized for the activation of the reset as well as for the removal of the reset.

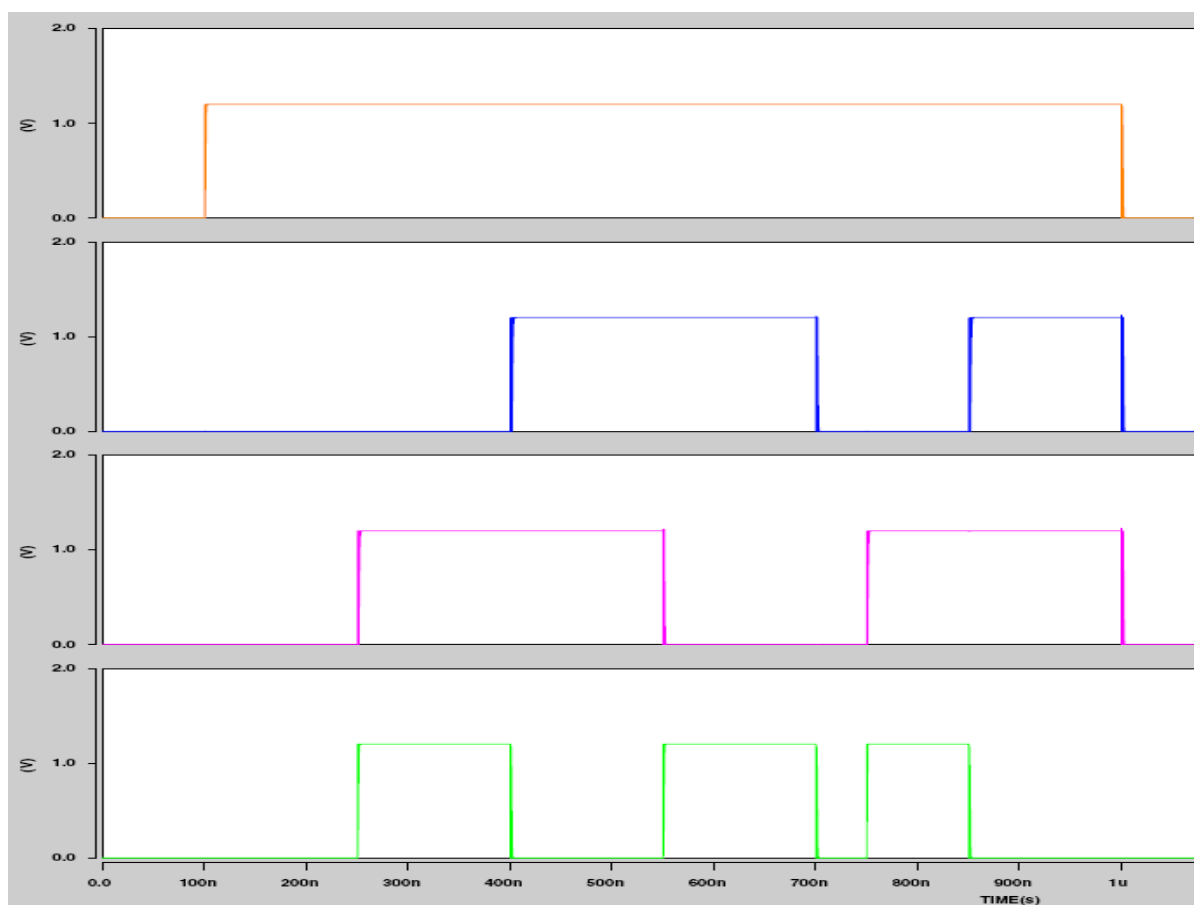


Figure 9: Simulated timing of toggle cell UACTOX020



GALAXY

GALS InterfACE for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 31/03/2009
Issue: 1

```
delay {
  timing_arcs {
    #Init Q0=L, Q1=L @ D=L
    RN    LH                @    D=L
    D    LH    to Q0 LH    @    RN=H
    D    HL    to Q1 LH    @    RN=H
    D    LH    to Q0 HL    @    RN=H
    D    HL    to Q1 HL    @    RN=H
    #Init Q0=H, Q1=H
    D    HL                @    RN=H
    (RN HL, RN HL)  to (Q0 HL, Q1 HL) @    D=L
    #INIT Q0=L, Q1=L @ D=H
    D    LH                @    RN=L
    RN    LH    to Q0 LH    @    D=H
  }
}
```

Important figure for the toggle is the minimum pulse width at input D to cause the toggle events at the outputs and the minimum pulse width of the reset to drive the toggle cell into the initial state. While the ladder is rather uncritical, the minimum input pulse width is of importance for asynchronous and self-timed behaviour.

In general asynchronous control logic is enforced to work with causal and monotonic signal behaviour, which means, that no gate input change may occur without the previous output change having happened. Nevertheless a feedback from a toggle output via very fast and simple gates into the input of the respective gate, might lead to very short input pulses, violating the correct functionality of the toggle cell. Characterization of this mpw constraint will therefore help engineers to use standard tools to verify functionality of the final circuit design.

```
constraint {
  constraint_arcs {
    mpw  D L MEAS D HL to Q1 LH    @ RN=H
    mpw  D L MEAS D HL to Q1 HL    @ RN=H
    mpw  D L MEAS D LH to Q0 LH    @ RN=H
    mpw  D L MEAS D LH to Q0 HL    @ RN=H
    mpw  D H MEAS D LH to Q0 LH    @ RN=H
    mpw  D H MEAS D HL to Q1 LH    @ RN=H
    mpw  D H MEAS D HL to Q1 HL    @ RN=H

    mpw  RN L MEAS RN HL to Q0 HL @ D=0
    mpw  RN L MEAS RN HL to Q1 HL @ D=0
    mpw  RN L MEAS RN HL to Q1 HL @ D=1
  }
}
```

Power consumption due to input signal changes is only characterized for input D changes while reset is active and for reset pin changes, while input D is stable.

```
power {
  power_arcs {
    D    BOTH  nochangeQ0=0, Q1=0    @    RN=0
    RN   BOTH  nochangeQ0=0, Q1=0    @    D=0
  }
}
```



3.3 MUTEX

The Mutex element includes two cross-coupled NAND circuits and an inverter based output filter. The resolution time of this Mutex circuit for the case when both inputs are activated in the same moment has to be characterized. As this is a function of the difference in arrival time of the two signals no final timing can be characterized. Therefore the timing will be given under certain constraints, which are then specified in the constraints section

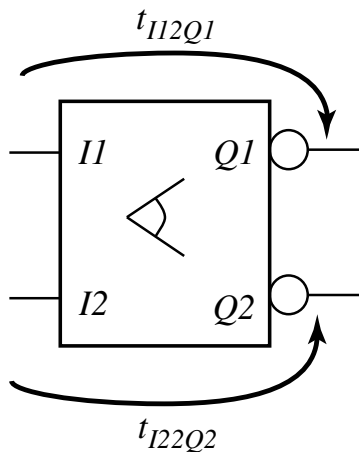


Figure 10: Timing arcs for mutex element

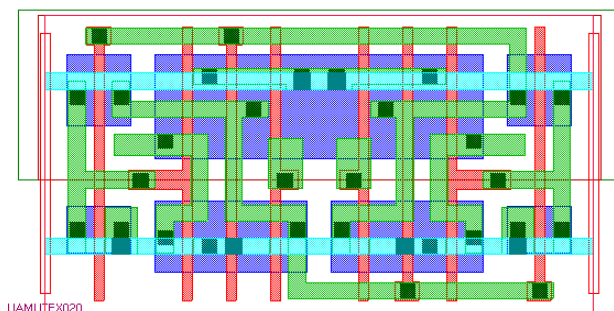


Figure 11: Layout of initial mutex element UAMUTEX020

The behaviour of the mutex is sketched in the attached simulated waveform diagram. As this is an inverting mutex, any raising input edge is followed by a falling output edge of the respective output. Taking that into account, the Output Q1 (yellow) is following input I1 (light blue) while output Q2 (dark blue) is following input I2 (red). If input I1 arrives while I2 is still active, the output transition on Q2 is delayed until output Q1 has finished the cycle.

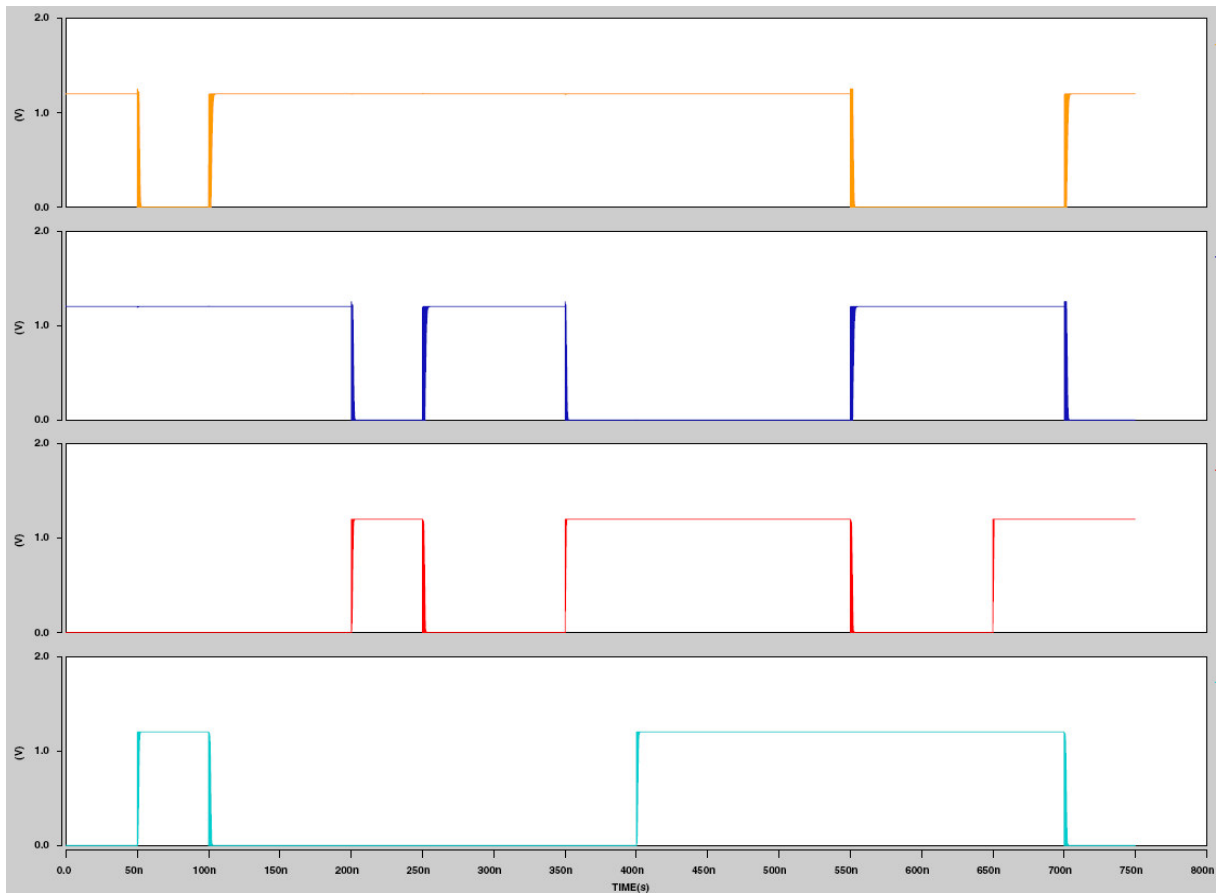


Figure 12: Layout of initial mutex element UAMUTEX020

Timing of the Mutex is given as delay from input to output in case the other input is in-active and the respective output stable on zero. The delay is characterized for both inputs.

```

delay {
  timing_arcs {
    I1          to   Q1    INV   @   I2=0
    I2          to   Q2    INV   @   I1=0
    #Init Q2=H, Q1=L
    I1          LH          @   I2=H
    (I2 HL, I2 HL) to (Q2 LH, Q1 HL) @   I1=1
    (I1 HL, I1 HL) to (Q2 HL, Q1 LH) @   I2=1
  }
}

```



In the constraints section we specify the minimum pulse width, required to generate an output transition.

```
constraint {
  constraint_arcs {
    mpw B L MEAS B HL to Z0 LH @ A=H
    mpw A L MEAS A HL to Z1 LH @ B=H
  }
}
```

Additional power arcs are only characterized for input transitions, where no output change is caused, as the other input had got a granted output.

```
power {
  power_arcs {
    B BOTH nochangeZ0=1, Z1=0 @ A=1
    A BOTH nochangeZ0=0, Z1=1 @ B=1
  }
}
```

Resolution time of the mutex will be investigated in further detail. But as this requires a dedicated simulation testbench and can not be run with standard characterization tools, it is not part of this report. The optimum way of handling the potential metastability and the input dependent resolution time with standard tools and industry reference design flows is part of the ongoing work.

3.4 DELAY ELEMENTS

Delay elements will only be characterized for their specific delay from input to output. The only constraint, which will be characterized, is the minimum pulse width, generating a visible pulse/peak at the output.