Building a basic membrane computer

Alejandro Millan, Julian Viejo, Juan Quiros, et al.

14th Brainstorming Week on Membrane Computing, Feb. 1-5, 2016

Grupo ID2 (Investigacion y Desarrollo Digital)

www.dte.us.es/id2

Dept. Tecnologia Electronica - Universidad de Sevilla (Spain)

Introduction

Until now, developments have been focused into improving simulation time of P systems:

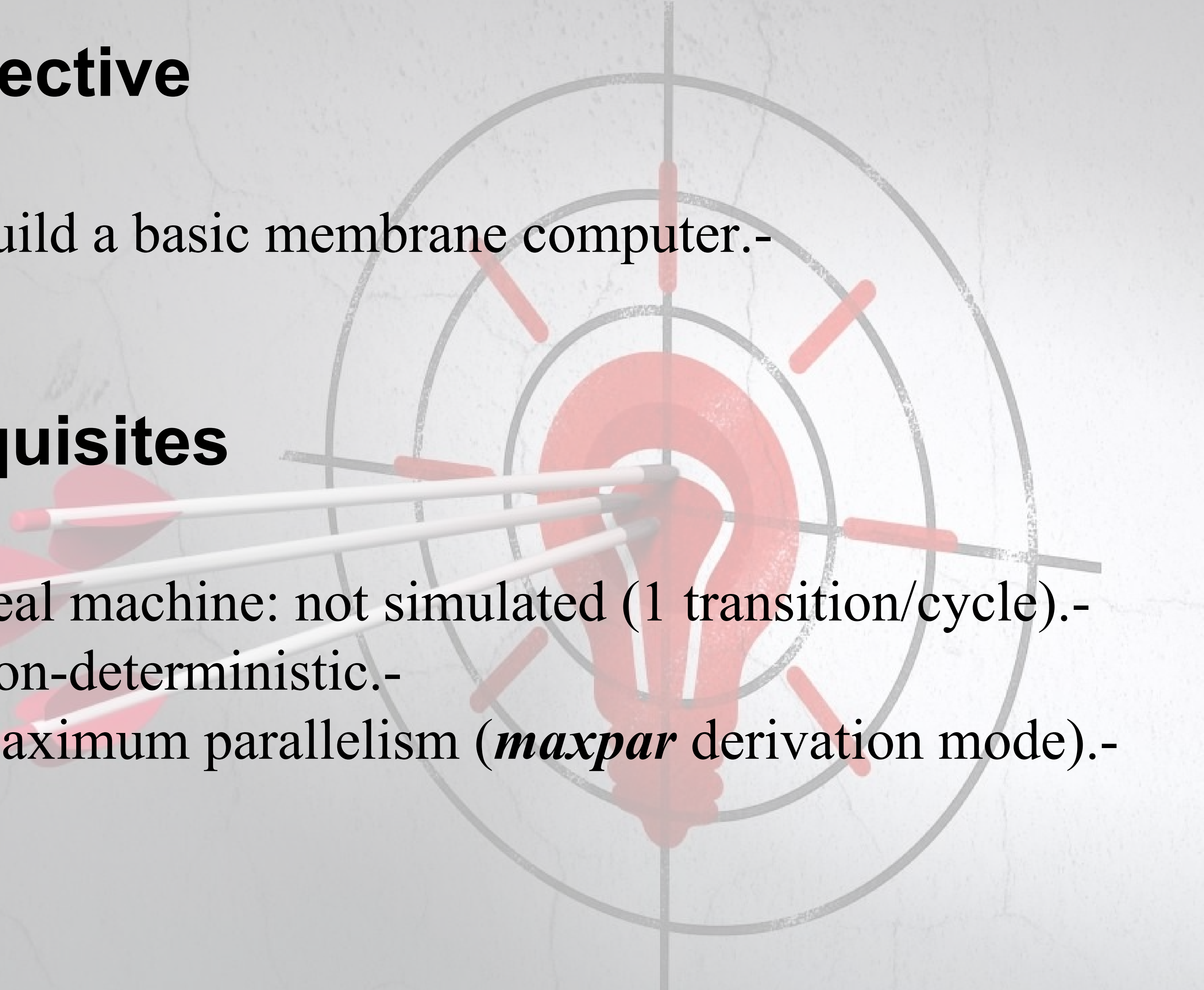
- Software simulation.-
  - RGNC (USe), Ciobanu, UniVr, …
- Hardware simulation.-
  - UPM, Petreska, Nguyen, Quiros, …

Objective

- Build a basic membrane computer.-

Requisites

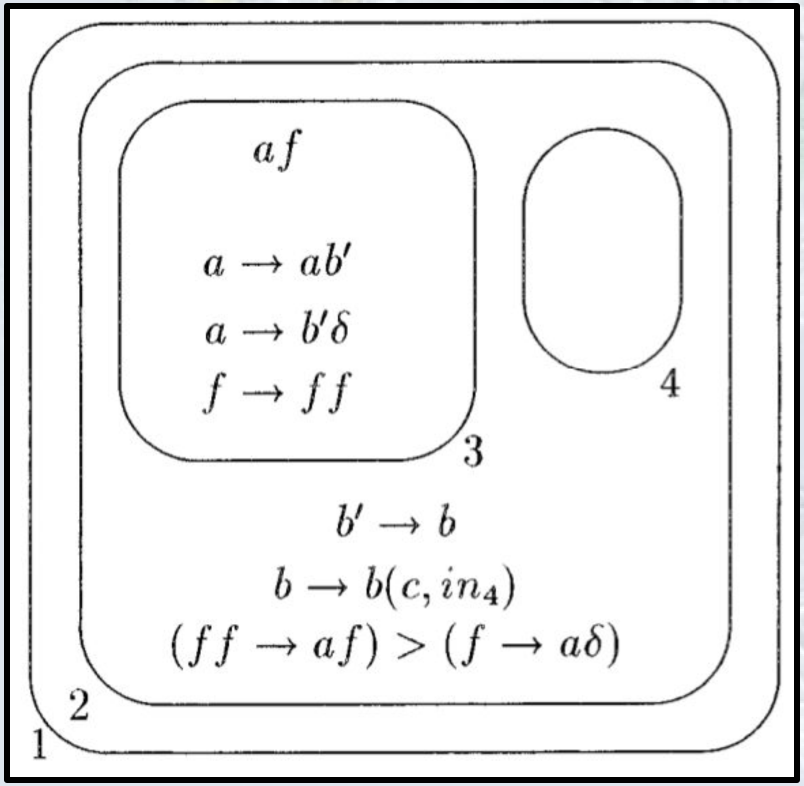
- Real machine: not simulated (1 transition/cycle).-
- Non-deterministic.-
- Maximum parallelism (***maxpar*** derivation mode).-



The chosen P systems [Pãun, 2000]

Computer 1

Computer 2



af

$a \rightarrow ab'$

$a \rightarrow b'\delta$

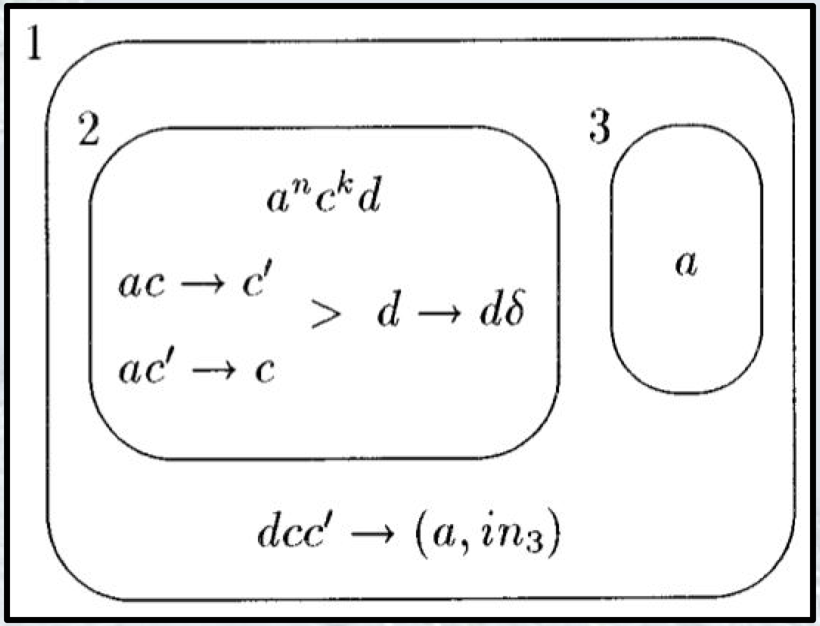
$f \rightarrow ff$

$b' \rightarrow b$

$b \rightarrow b(c, in_4)$

$(ff \rightarrow af) > (f \rightarrow a\delta)$

$\mathbf{n^2}$ generator (**n >= 1**).-



$a^n c^k d$

$ac \rightarrow c'$

$ac' \rightarrow c$

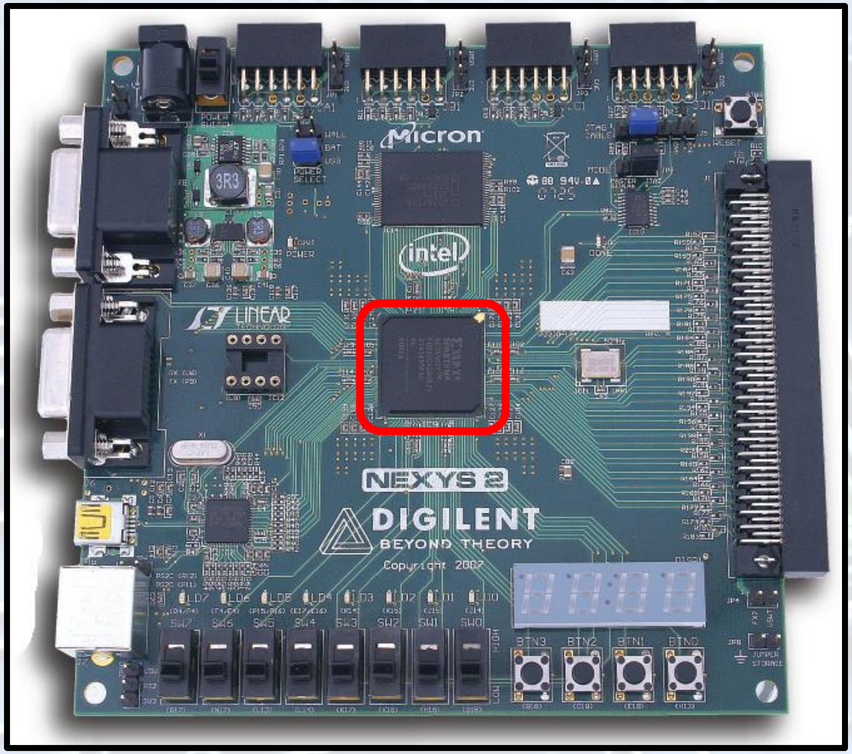
$> \; d \rightarrow d\delta$

a

$dcc' \rightarrow (a, in_3)$

Divisor test (**k** divides **n**?).-

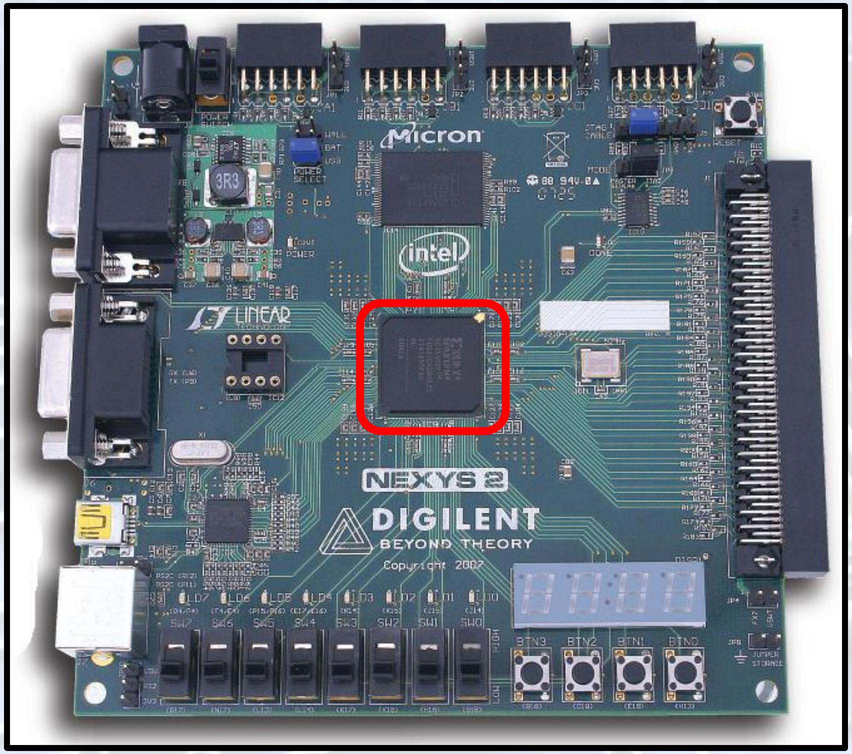
FPGA Technology (I)



Training board

- **FPGA** = Field-Programmable Gate Array.-
- **Resources:**
  - ○ Look-Up-Tables (logic).-
  - ○ Flip-Flops (memory).-
  - ○ Arithmetic:
    - ■ Distributed.-
    - ■ Especific (MULT).-

FPGA Technology (II)



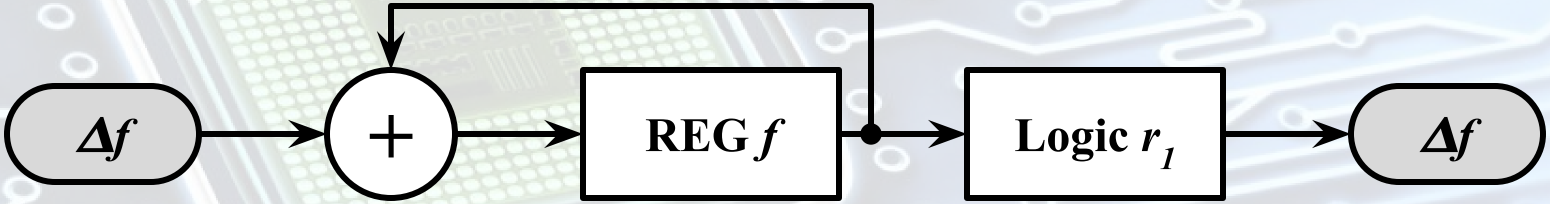
Training board

- **Design:**
  - ○ Hardware Description Language (HDL).-
  - ○ Place & Routing.-
  - ○ Reconfigurable.-

- **Applications:**
  - ○ Prototyping.-
  - ○ Quick deployment.-
  - ○ Low production.-

Object/rule implementation

$$r_1: \quad f \longrightarrow \quad ff$$



Competition - Case 1 (Algorithm)

$$r_1: \quad a \longrightarrow ab'$$

$$r_2: \quad a \longrightarrow b'\delta$$

1. Randomly let:

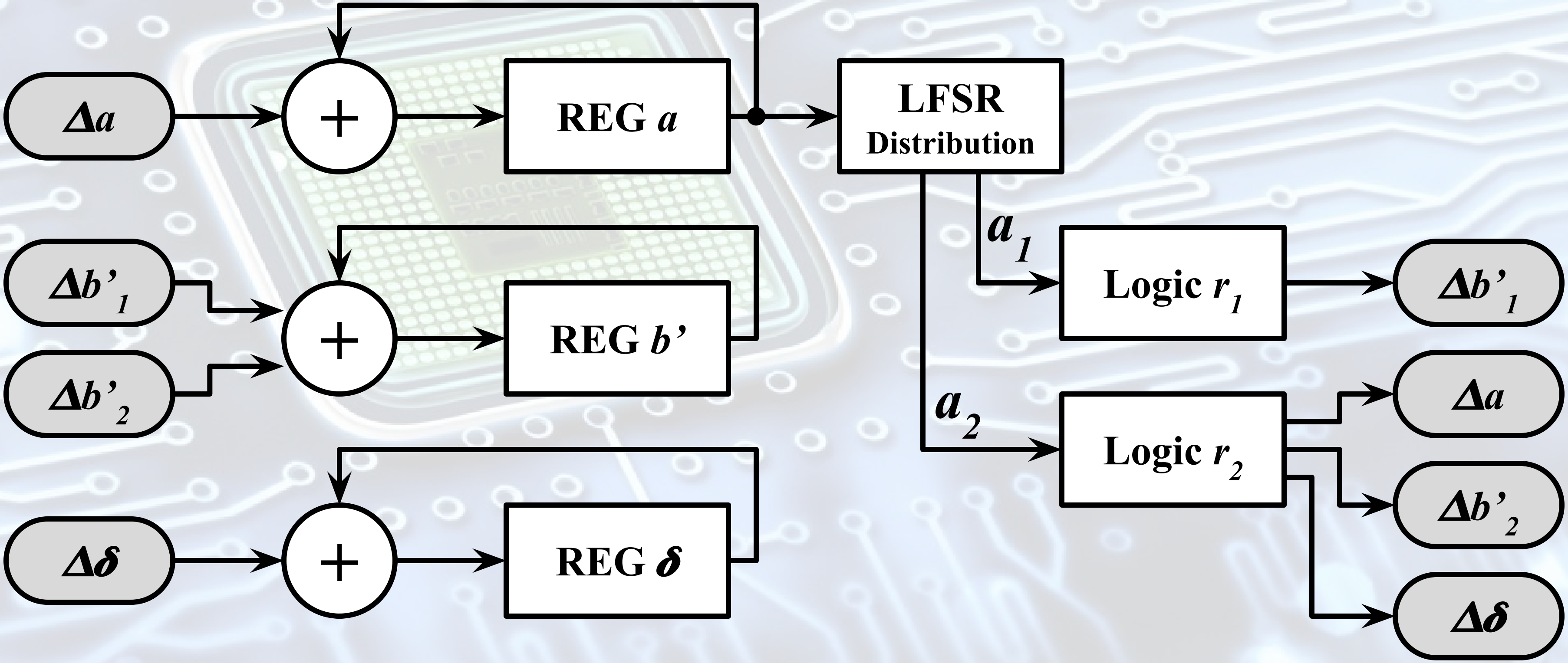
   $$a = a_1 + a_2$$

2. Apply r_1 x a_1 times and r_2 x a_2 times.-

Competition - Case 1 (Design)

$$r_1: \quad a \longrightarrow ab'$$

$$r_2: \quad a \longrightarrow b'\delta$$



Competition - Case 2 (Algorithm)

$$r_1: \quad ac \longrightarrow c'$$

$$r_2: \quad ac' \longrightarrow c$$

1. Randomly let:

$$a = a_1 + a_2$$

2. Let:

$$\lambda_1 = max\{\ 0\ ;\ a_1 - c\ \}$$

$$\lambda_2 = max\{\ 0\ ;\ a_2 - c'\ \}$$

$$\tau_1 = min\{\ c\ ;\ a_1 + \lambda_2\ \}$$

$$\tau_2 = min\{\ c'\ ;\ a_2 + \lambda_1\ \}$$

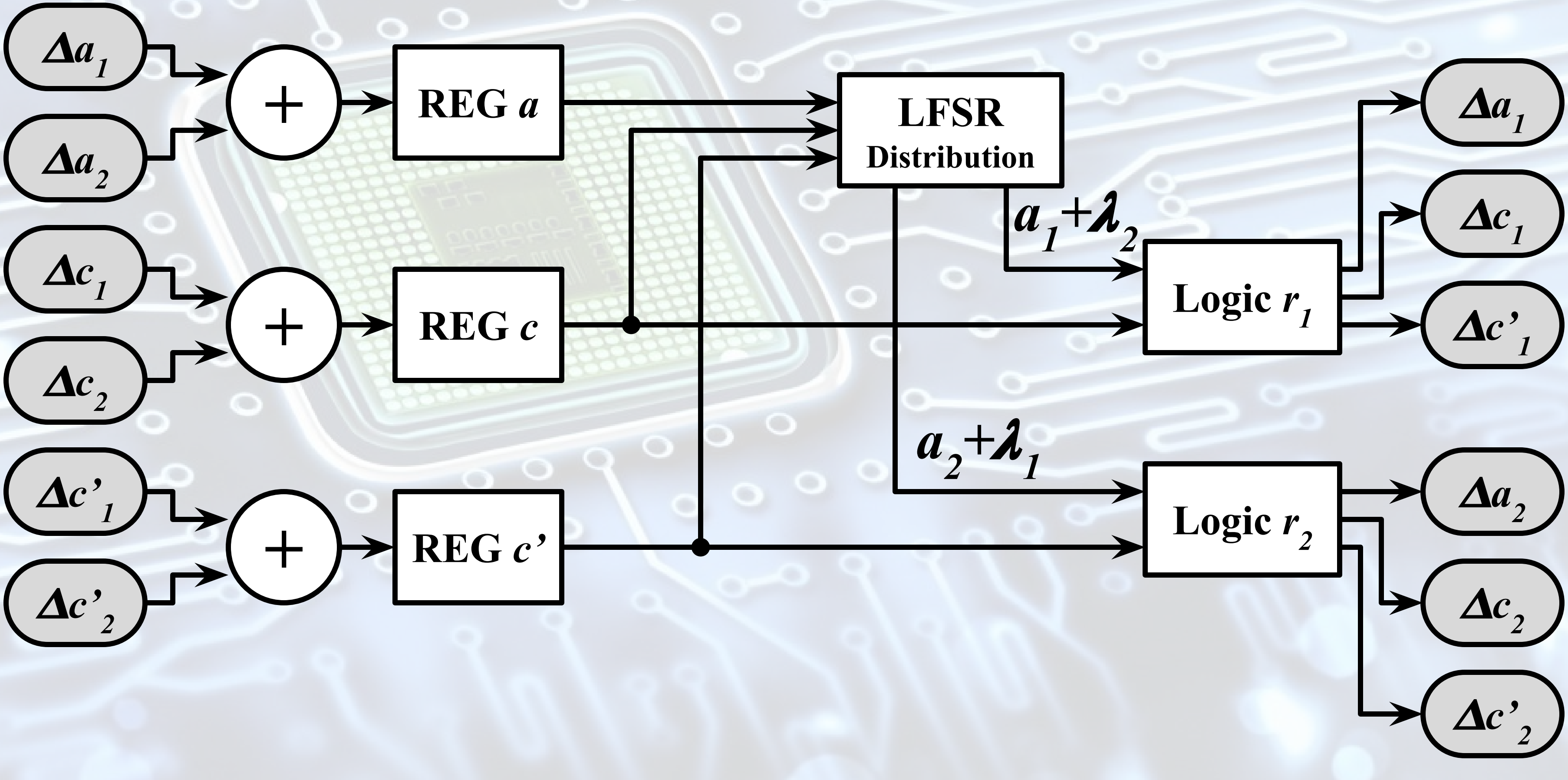
3. Apply r_1 x τ_1 times

   and r_2 x τ_2 times.-

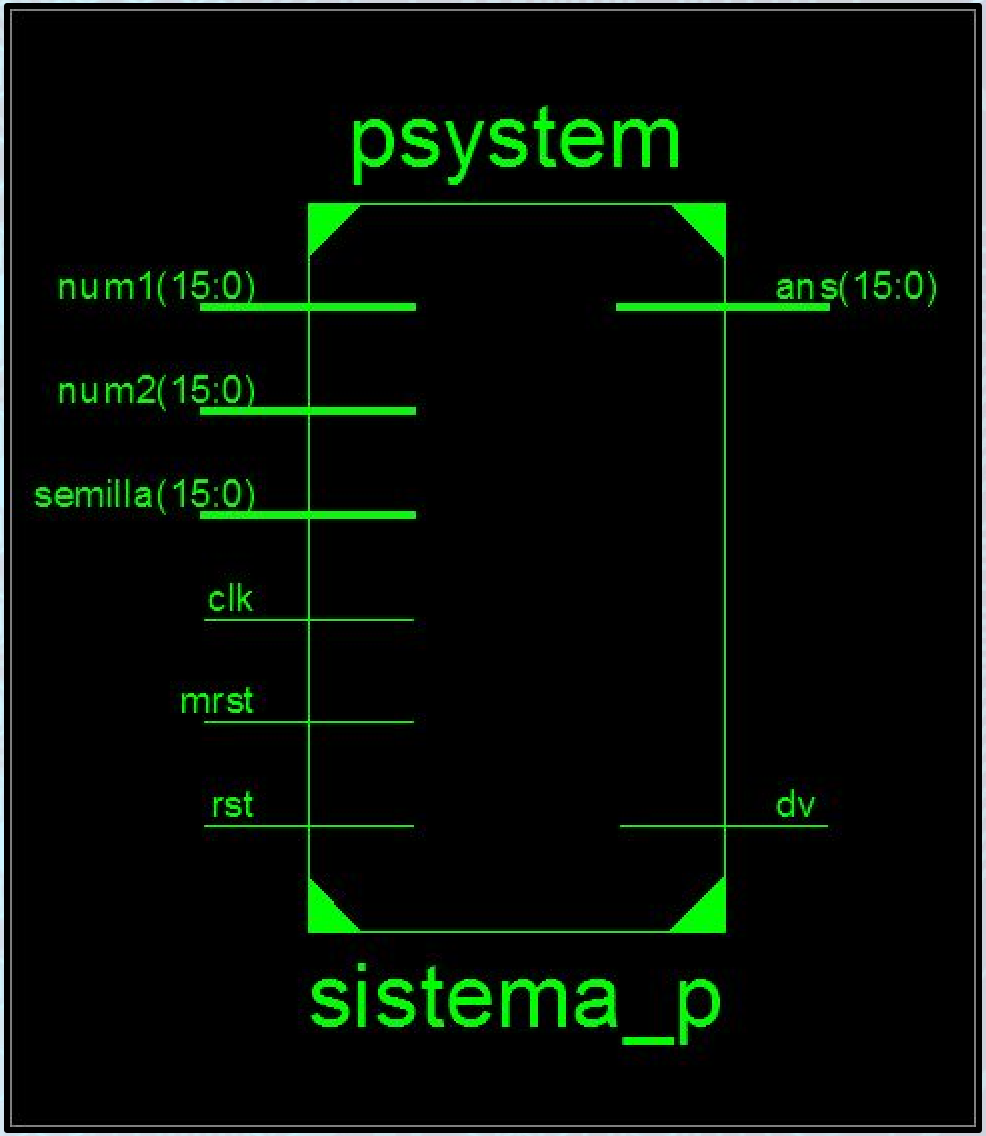
Competition - Case 2 (Design)

$$r_1: \quad ac \longrightarrow c'$$

$$r_2: \quad ac' \longrightarrow c$$



System interface - Computer 2



n Number

k Number

Seed

Clock

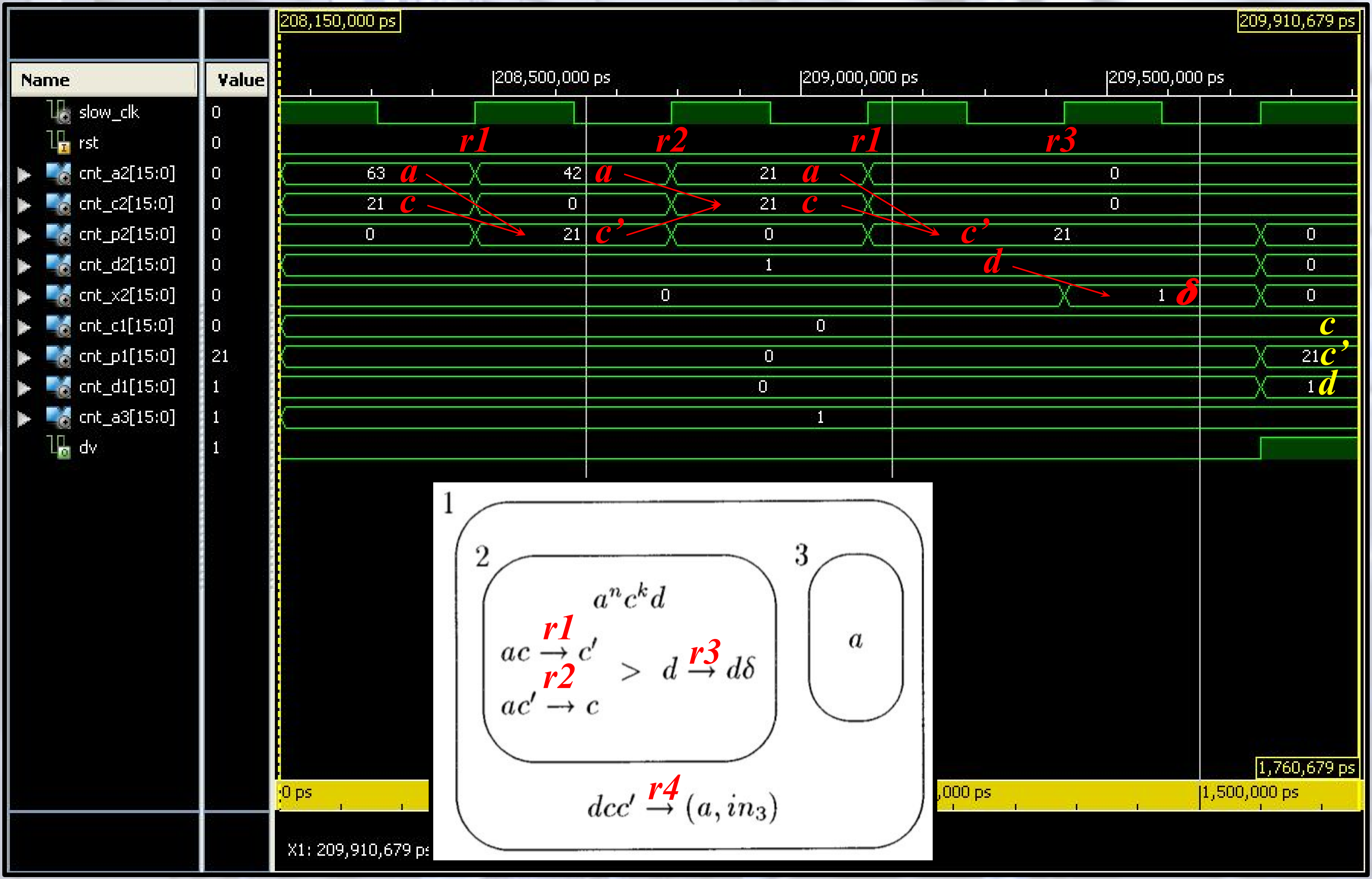
Master reset

Reset

Answer

Data Valid

Operation detail - Computer 2



208,150,000 ps

209,910,679 ps

Name	Value
slow_clk	0
rst	0
cnt_a2[15:0]	0
cnt_c2[15:0]	0
cnt_p2[15:0]	0
cnt_d2[15:0]	0
cnt_x2[15:0]	0
cnt_c1[15:0]	0
cnt_p1[15:0]	21
cnt_d1[15:0]	1
cnt_a3[15:0]	1
dv	1

1,760,679 ps

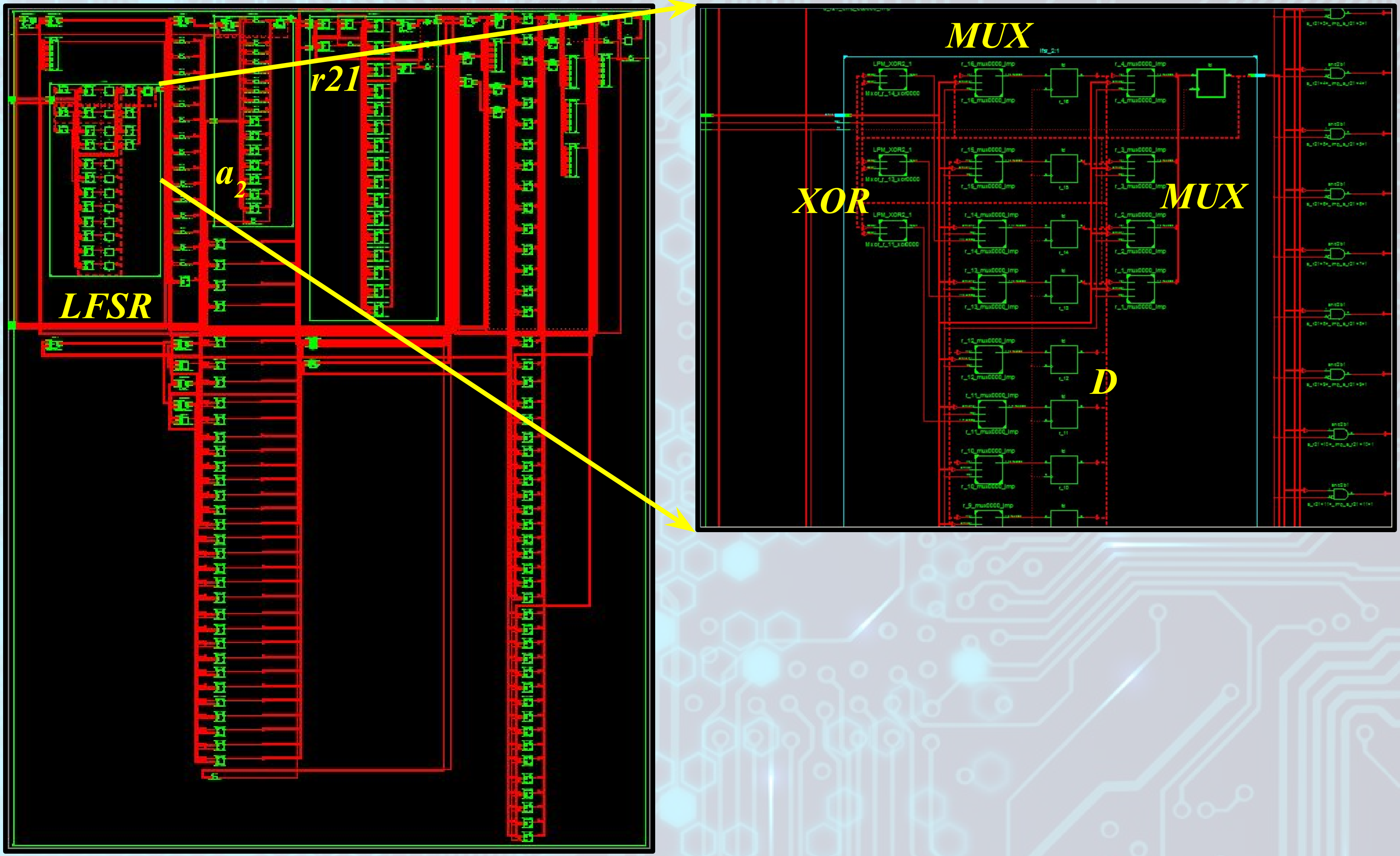
X1: 209,910,679 ps

$$a^n c^k d$$

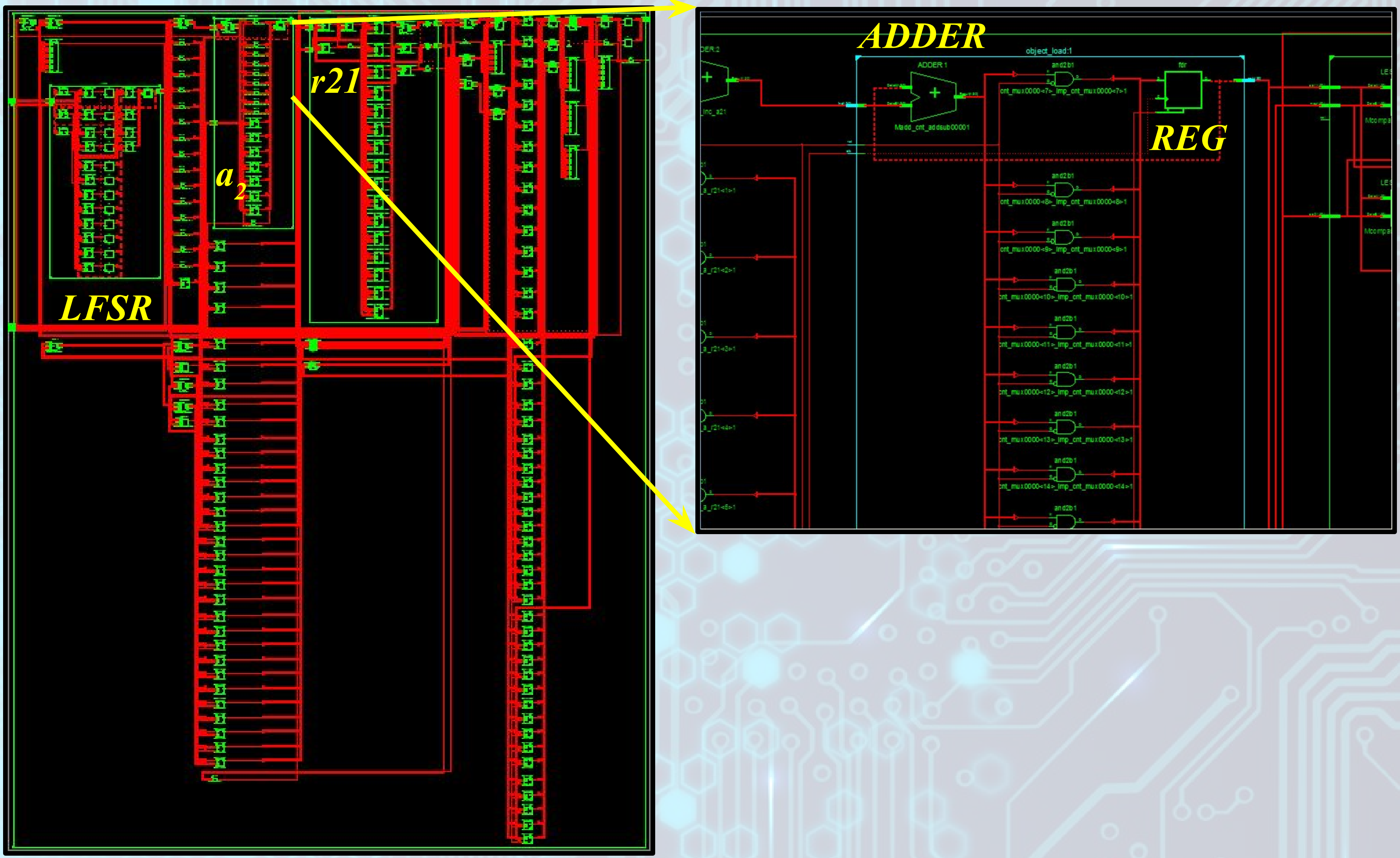
$$ac \xrightarrow{r1} c' \qquad ac' \xrightarrow{r2} c \quad > \quad d \xrightarrow{r3} d\delta$$

$$dcc' \xrightarrow{r4} (a, in_3)$$

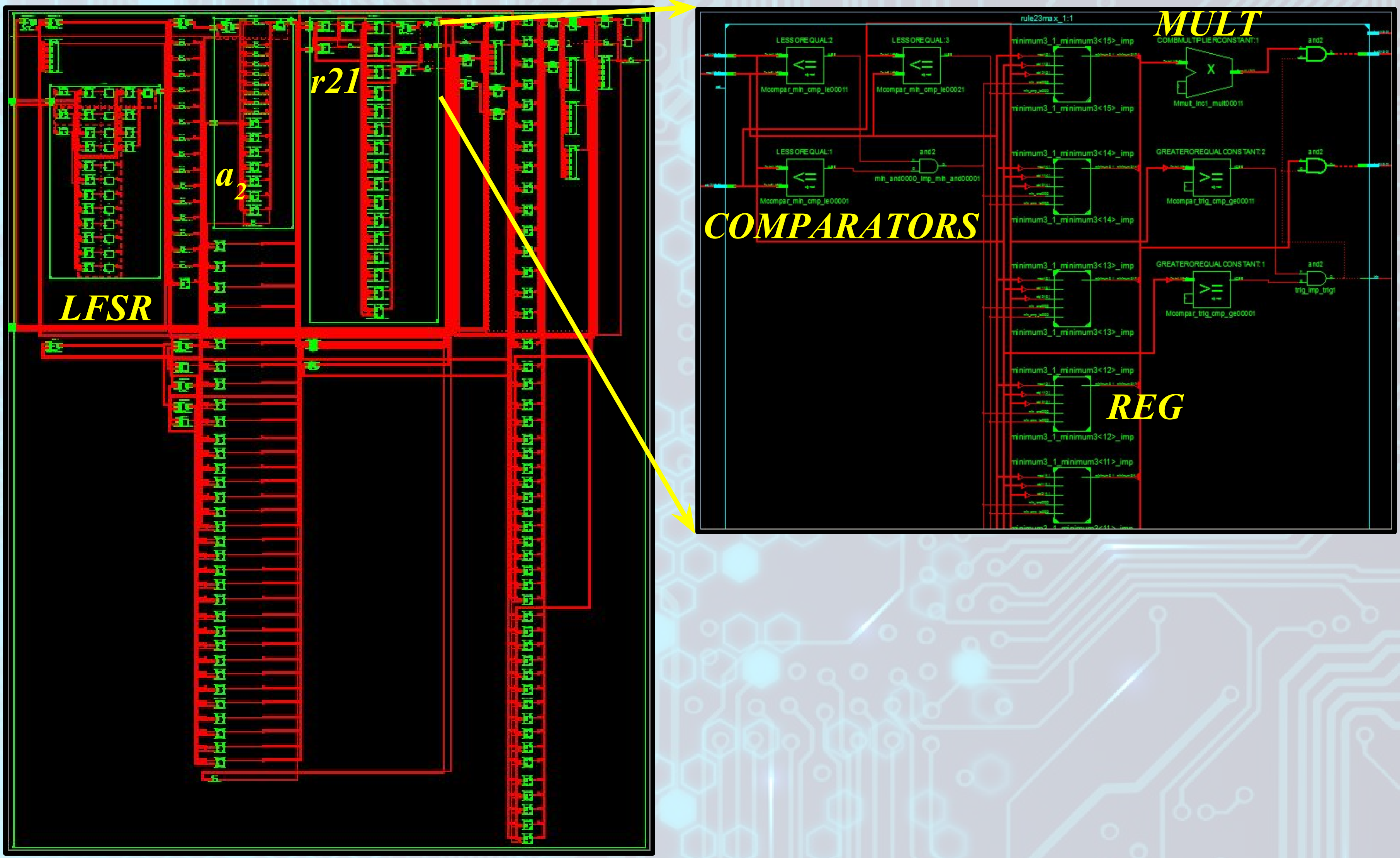
Schematic - Computer 2 (LFSR)



Schematic - Computer 2 (object *a*)



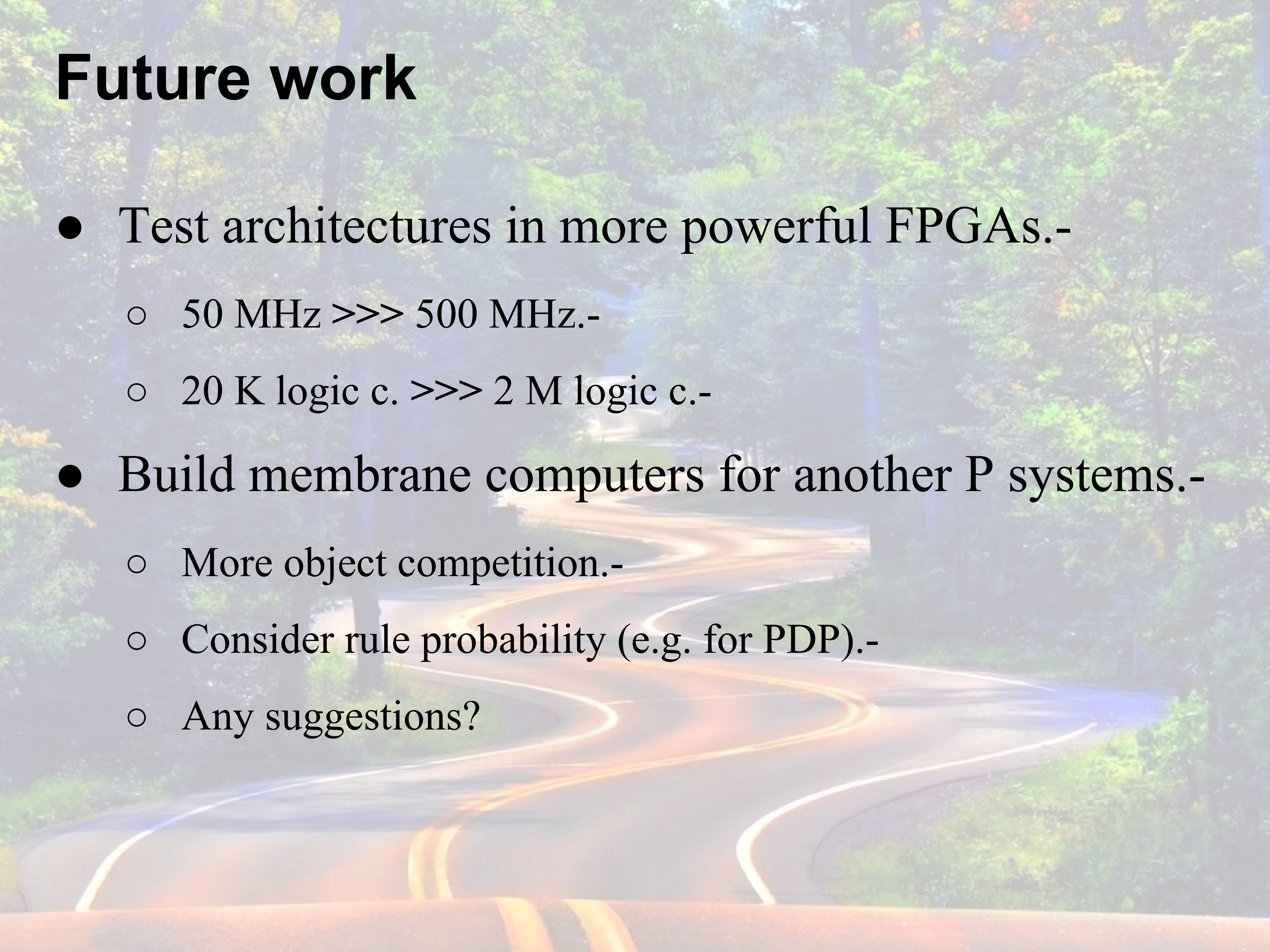
Schematic - Computer 2 (rule *r21*)



Implementation information

FPGA Model	Spartan 3E-1200
Clock frequency	50 MHz
Performance	50 Mtransition/s
Register width	16-bit
FPGA Occupation	2% / 5% slices

Future work



- Test architectures in more powerful FPGAs.-
  - 50 MHz >>> 500 MHz.-
  - 20 K logic c. >>> 2 M logic c.-
- Build membrane computers for another P systems.-
  - More object competition.-
  - Consider rule probability (e.g. for PDP).-
  - Any suggestions?