Logic Synthesis for Switching Lattices by Decomposition with P-Circuits

Marie Skłodowska-Curie grant agreement No 691178
(European Union's Horizon 2020 research and innovation programme)

Anna Bernasconi[1], Valentina Ciriani[2], *Luca Frontini*[2], Gabriella Trucco[2], Valentino Liberali[3], Tiziano Villa[4]

[1]Dipartimento di Informatica, Università di Pisa, Italy, anna.bernasconi@unipi.it

[2]Dipartimento di Informatica, Università degli Studi di Milano, Italy, {valentina.ciriani, luca.frontini, gabriella.trucco}@unimi.it

[3]Dipartimento di Fisica, Università degli Studi di Milano, Italy, valentino.liberali@unimi.it

[4]Dipartimento di Informatica, Università degli Studi di Verona, Italy, tiziano.villa@univr.it

Introduction

CMOS technology

- Transistor size has shrunk for decades
- The trend reached a critical point

The Moore's Law era is coming to an end

New emerging technologies

- Biotechnologies, molecular-scale self-assembled systems
- Graphene structures
- Switching lattices arrays

These technologies are in an early state

A novel synthesis approach should be focused on the properties of the devices

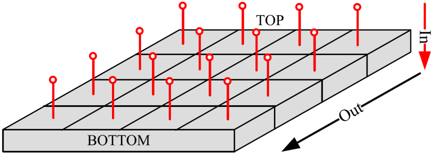
Synthesis efficiency can be an important factor for a technology choice

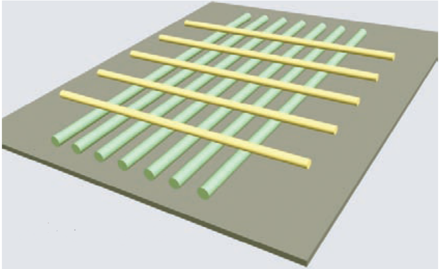
We focus on Synthesis for Switching Lattices

How Switching Lattices are made

Nanowires are one of the most promising technologies

- Nanowire circuits can be made with **self-assembled structures**
- **pn-junctions** are built crossing *n*-type and *p*-type nanowires
- **Low** V_{in} voltage makes *p*-nanowires conductive and *n*-nanowires resistive
- **High** V_{in} voltage makes *n*-nanowires conductive and *p*-nanowires resistive

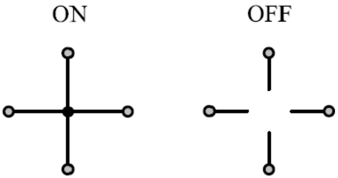


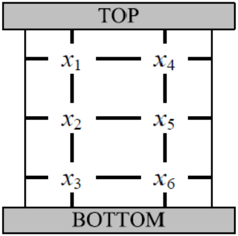


The Switching Lattices

Switching Lattices are **two-dimensional** arrays of **four-terminal** switches

- When switches are **ON** all terminals are connected, when **OFF** all terminals are disconnected
- Each switch is controlled by a boolean literal, **1** or **0**
- The boolean function f is the SOP of the literals along each path from **top** to **bottom**
- The function synthesized by the lattice is:
  $f = x_1x_2x_3 + x_1x_2x_5x_6 +$
  $+x_4x_5x_2x_3 + x_4x_5x_6$

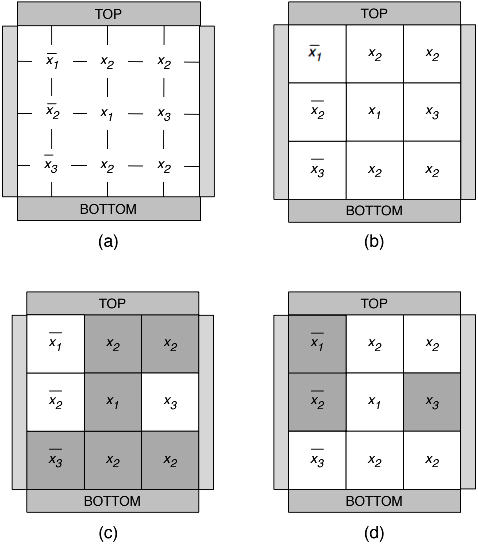




From Crossbars to Lattices

For an easier representation, the **crossbars** are converted to **lattices**:

- A 'checkerboard' notation is used
- Darker and white sites represent **ON** and **OFF**
- a), b): the 4-terminal switching network and the lattice describing $f = \overline{x}_1\overline{x}_2\overline{x}_3 + x_1x_2 + x_2x_3$
- c), d): the lattice evaluated on inputs (1,1,0) and (0,0,1)



The synthesis methods

Altun-Riedel, 2012

- Synthesizes f and f^D from **top to bottom** and **left to right**
- It produces lattices with size growing **linearly** with the SOP
- Time **complexity is polynomial** in the number of products

TOP

LEFT / RIGHT

$\overline{x}_6$	$\overline{x}_8$	$\overline{x}_6$	x_4
$\overline{x}_8$	$\overline{x}_8$	$\overline{x}_5$	x_4
$\overline{x}_6$	$\overline{x}_7$	$\overline{x}_6$	x_4
$\overline{x}_7$	$\overline{x}_7$	$\overline{x}_5$	x_4
$\overline{x}_6$	$\overline{x}_5$	$\overline{x}_6$	x_4
x_1	x_1	x_1	x_1
$\overline{x}_2$	$\overline{x}_2$	$\overline{x}_2$	$\overline{x}_2$
x_3	x_3	x_3	x_3

BOTTOM

Gange-Søndergaard-Stuckey, 2014

- f is synthesized from **top to bottom**
- The synthesis problem is formulated as a **satisfiability problem**, then the problem is solved with a SAT solver
- The synthesis method searches for better implementations starting from an upper bound size
- The synthesis loses the possibility to generate both f and f^D

TOP

x_4	$\overline{x}_6$	$\overline{x}_7$
$\overline{x}_2$	$\overline{x}_5$	$\overline{x}_8$
x_1	$\overline{x}_2$	$\overline{x}_6$
x_3	0	x_6

BOTTOM

In both examples the synthesized function is:

$$f = \overline{x}_8\overline{x}_7\overline{x}_6x_3\overline{x}_2x_1 + \overline{x}_8\overline{x}_7\overline{x}_5x_3\overline{x}_2x_1 + x_4x_3\overline{x}_2x_1$$

Decompositions and P-circuits

Let $f\{x_1, \ldots, x_n\}$ be a completely specified Boolean function

- **Shannon** *decomposition:* $f = \overline{x}_i f|_{\overline{x}_i} + x_i f|_{x_i}$
- **EXOR-based** *decomposition:* $f = (\overline{x}_i \oplus p) f|_{x_i = p} + (x_i \oplus p) f|_{x_i \neq p}$, where p does not depend on x_i

These decompositions are **not oriented to area minimization**: the cubes of f may be split into two smaller sub-cubes when projected onto $f|_{x_i = p}$, and $f|_{x_i \neq p}$.

P-circuits keep unprojected some points of the original function: defining $I = f|_{x_i = p} \cap f|_{x_i \neq p}$, we can keep I unprojected

$$f = (\overline{x}_i \oplus p)(f|_{x_i = p} \setminus I) + (x_i \oplus p)(f|_{x_i \neq p} \setminus I) + I$$

In this way we **avoid to split the cubes of** f

Formal definition of P-circuit

Definition

A *P-circuit* of a completely specified function f is the circuit $P(f)$ denoted by the expression:

$$P(f) = (\overline{x}_i \oplus S(p))\, S(f^{=}) + (x_i \oplus S(p))\, S(f^{\neq}) + S(f^{I})$$

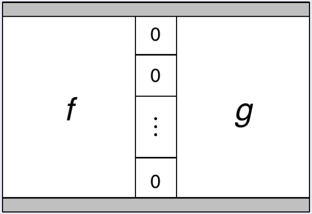
where:

1. $(f|_{x_i=p} \setminus I) \;\subseteq\; f^{=} \;\subseteq\; f|_{x_i=p}$
2. $(f|_{x_i\neq p} \setminus I) \;\subseteq\; f^{\neq} \;\subseteq\; f|_{x_i\neq p}$
3. $\emptyset \;\subseteq\; f^{I} \;\subseteq\; I$
4. $P(f) = f$

Disjunction and conjunction of lattices

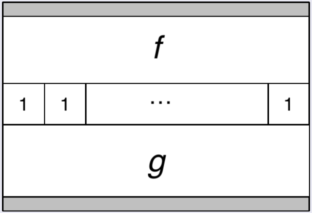
$f + g$

- separate the paths from top to bottom for f and g
- add a column of 0s
- add padding rows of 1s if lattices have different number of rows



$f \cdot g$

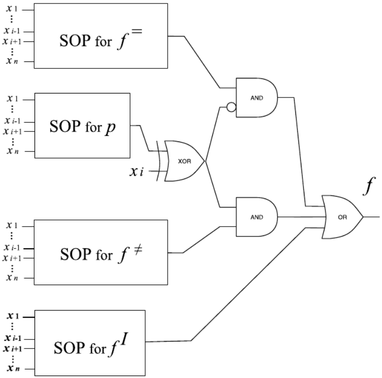
- any top-bottom path of f is joined to any top-bottom path of g
- add a row of 1s
- add padding columns of 0s if lattices have different number of columns

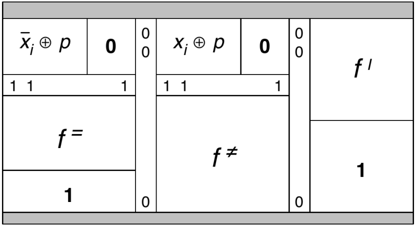


P-circuit lattice implementation

Theorem

Let f be a Boolean function depending on n binary variables, and let $P(f) = (\overline{x}_i \oplus S(p))\, S(f^{=}) + (x_i \oplus S(p))\, S(f^{\neq}) + S(f^{I})$ be a P-circuit representing f. The lattice obtained composing the lattices for the three sets $f^{=}$, $f^{\neq}$, and f^{I} and for the functions $(\overline{x}_i \oplus p)$ and $(x_i \oplus p)$ implements the function f.



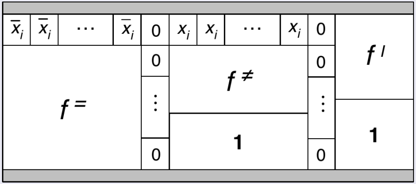


P-circuit projection function

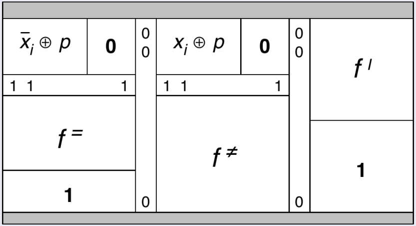
Corollary

The two lattices on the right implement a function f through its P-circuit representations with projection functions $p = 0$ and $p = x_j$, respectively.

$p = 0$



$p = x_i$



Synthesis example

In this example is used the synthesis by Altun-Riedel

$$z4(2) = x_3\overline{x}_4\overline{x}_6\overline{x}_7 + x_1\overline{x}_3x_4\overline{x}_6 + \overline{x}_1x_3\overline{x}_6\overline{x}_7 + \overline{x}_3\overline{x}_4x_6\overline{x}_7 + x_1x_3x_4x_6 + x_1\overline{x}_3\overline{x}_6x_7 + \overline{x}_1x_3\overline{x}_4\overline{x}_6 + \overline{x}_3x_4\overline{x}_6x_7 + \overline{x}_1\overline{x}_3\overline{x}_4x_6 + x_1x_3x_6x_7 + x_3x_4x_6x_7$$

x_3x_4 \ x_6x_7	00	01	11	10
0	0	0	1	1
1	0	1	0	1
2	1	0	1	0
	1	1	0	0

$x_1 = 0$

x_3x_4 \ x_6x_7	00	01	11	10
0	0	1	0	1
1	1	1	0	0
2	0	0	1	1
	1	0	1	0

$x_1 = 1$

The lattice size without decomposition is 12×12

x_3x_4 \ x_6x_7	00	01	11	10
0	0	0	1	1
1	0	0	0	1
2	1	0	0	0
	1	1	0	0

$z^{=}$

x_3x_4 \ x_6x_7	00	01	11	10
0	0	1	0	1
1	1	1	0	0
2	0	0	1	1
	1	0	1	0

$z^{\neq}$

x_3x_4 \ x_6x_7	00	01	11	10
0	0	0	0	0
1	0	1	0	0
2	0	0	1	0
	0	0	0	0

z^{I}

P-circuit representation:

$$P(z) = \overline{x}_1\, S(z^{=}) + x_1\, S(z^{\neq}) + S(z^{I})$$

$$S(z^{=}) = \overline{x}_3\overline{x}_4x_6 + x_3\overline{x}_4\overline{x}_6 + \overline{x}_3x_6\overline{x}_7 + \\ + x_3\overline{x}_6\overline{x}_7$$

$$S(z^{\neq}) = x_3x_4x_6 + \overline{x}_3x_4\overline{x}_6 + x_3x_6x_7 + \\ + \overline{x}_3\overline{x}_6x_7 + \overline{x}_3\overline{x}_4x_6\overline{x}_7 + x_3\overline{x}_4\overline{x}_6\overline{x}_7$$

$$S(z^{I}) = x_3x_4x_6x_7 + \overline{x}_3x_4\overline{x}_6x_7$$

The final lattice decomposed with P-circuits is 7×14:

$\overline{x}_1$	$\overline{x}_1$	$\overline{x}_1$	$\overline{x}_1$	0	x_1	x_1	x_1	x_1	x_1	x_1	0	x_4	x_4
x_6	x_3	x_6	x_3	0	x_6	$\overline{x}_3$	x_6	$\overline{x}_3$	$\overline{x}_3$	$\overline{x}_4$	0	x_7	x_7
$\overline{x}_3$	$\overline{x}_6$	$\overline{x}_3$	$\overline{x}_6$	0	x_3	$\overline{x}_6$	x_3	$\overline{x}_6$	$\overline{x}_4$	$\overline{x}_4$	0	x_6	$\overline{x}_3$
$\overline{x}_4$	$\overline{x}_4$	$\overline{x}_7$	$\overline{x}_7$	0	x_6	$\overline{x}_3$	x_6	$\overline{x}_3$	$\overline{x}_3$	$\overline{x}_7$	0	x_3	$\overline{x}_6$
1	1	1	1	0	x_3	$\overline{x}_6$	x_3	$\overline{x}_6$	$\overline{x}_7$	$\overline{x}_7$	0	1	1
1	1	1	1	0	x_3	x_4	x_3	x_7	x_6	x_3	0	1	1
1	1	1	1	0	x_4	x_4	x_7	x_7	$\overline{x}_3$	$\overline{x}_6$	0	1	1

Experiments

- Benchmarks are taken from LGSynth93
- Each benchmark output is considered as a separate boolean function
- A total of 1886 functions
- We evaluate the results for $p = 0$ using $x_i = x_1$
- We compare P-circuits decomposition with synthesis by Altun-Riedel, 2012 and Gange-Søndergaard-Stuckey, 2014
- We use a collection of Python scripts and a SAT solver to perform the Gange-Søndergaard-Stuckey synthesis.

- The algorithm has been implemented in C
- The experiments have been run on a machine with 16 CPU @2.5 GHz, running Centos 6.6

The Experiments

F-Name	Altun-Riedel		Gange-Søndergaard-Stuckey	
	standard Row×Col	P-circuit Row×Col	standard Row×Col	P-circuit Row×Col
bcc(35)	24×38	**25×24**	⋆	⋆
alu2(6)	14 × 13	17×10	⋆	**16×10**
bench(3)	4×6	7×4	**4×3**	6×4
ex5(46)	6×3	6×3	⋆	⋆
ex5(57)	10×8	14×8	**3×7**	13×3
max128(9)	10×9	14×7	**4×6**	13×4
max128(10)	16 × 17	17×10	⋆	**15×10**

⋆: does not end in 10 min. for each SAT problem

Results of the Experiments

To evalutate our approach we compare our results with the ones obtained in Altun-Riedel and in Gange-Søndergaard-Stuckey

Decomposing the function with P-circuits we obtain:

Altun-Riedel

- More compact area in 36% of cases
- Average area reduction of about 25%
- Very limited increase in time

Gange-Søndergaard-Stuckey

- More compact area in 33% of cases
- Average area reduction of about 24%
- In overall we save area and time

- In many cases the method Gange-Søndergaard-Stuckey fails in computing a result in a reasonable time
- We set a maximum of ten minutes for each SAT execution
- If synthesis is stopped we use the synthesis method by Altun-Riedel

Conclusions

- **A new method** for the synthesis of lattices with reduced size
- Based on **P-circuit decomposition**
- The lattice synthesis benefits from this decomposition:
  - **smaller lattices**: at least 24% of area reduction in 33% of functions
  - **affordable computing time**, in some cases even less time than without decomposition

In future works we will apply more complex types of decomposition

- within the P-circuit class
- other decomposition methods