Research Project Proposal: Quantum Computing and hardness of computation of combinatorial functions

Marco Venere
marco.venere@mail.polimi.it
Computer Science and Engineering Track

The Starting Question

Electronic Design Automation (EDA) presents a number of NP-hard algorithms whose execution requires a noticeable amount of time.

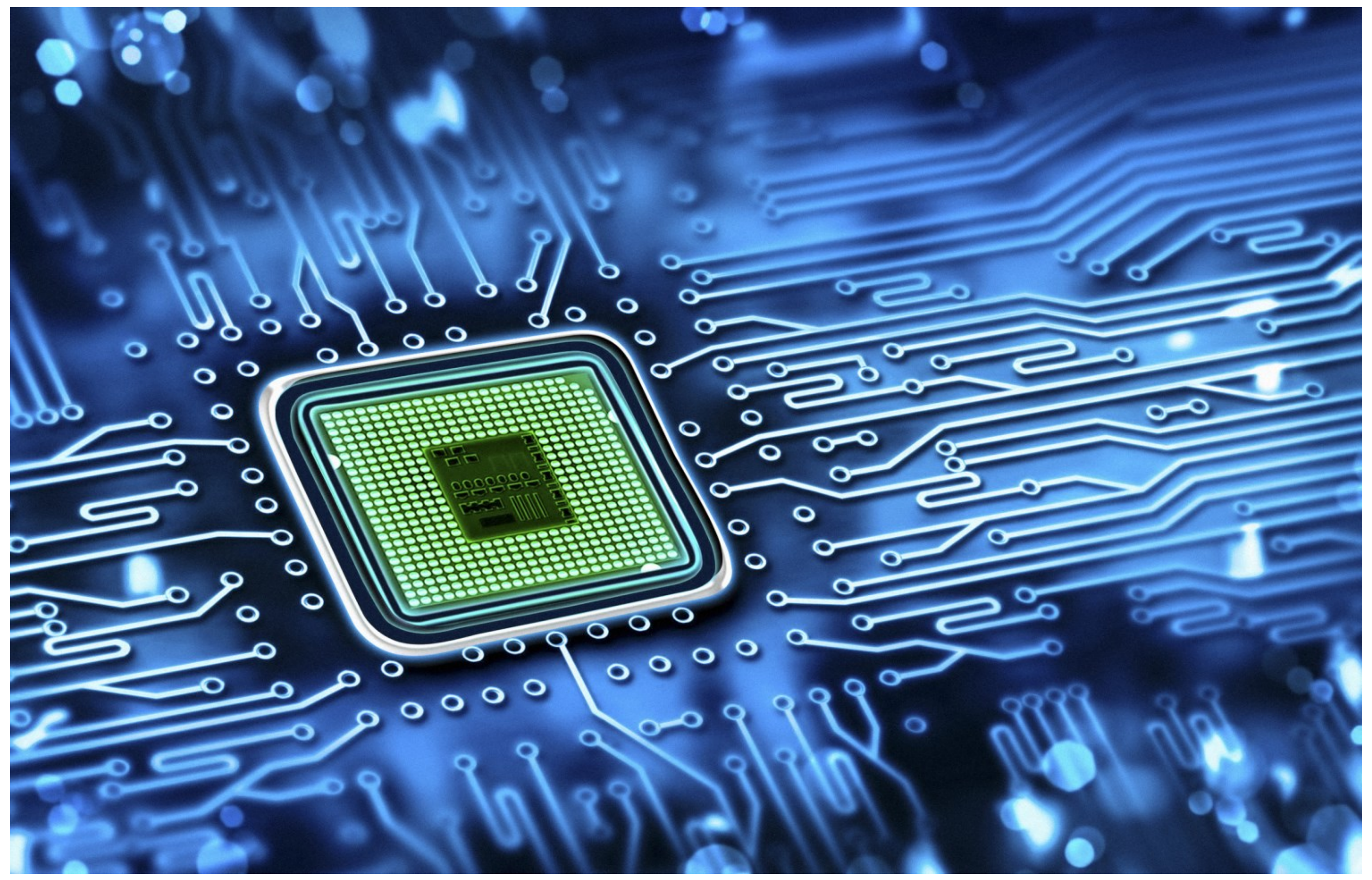
Quantum Computing has the potential to accelerate them, if a proper quantum circuit is designed.

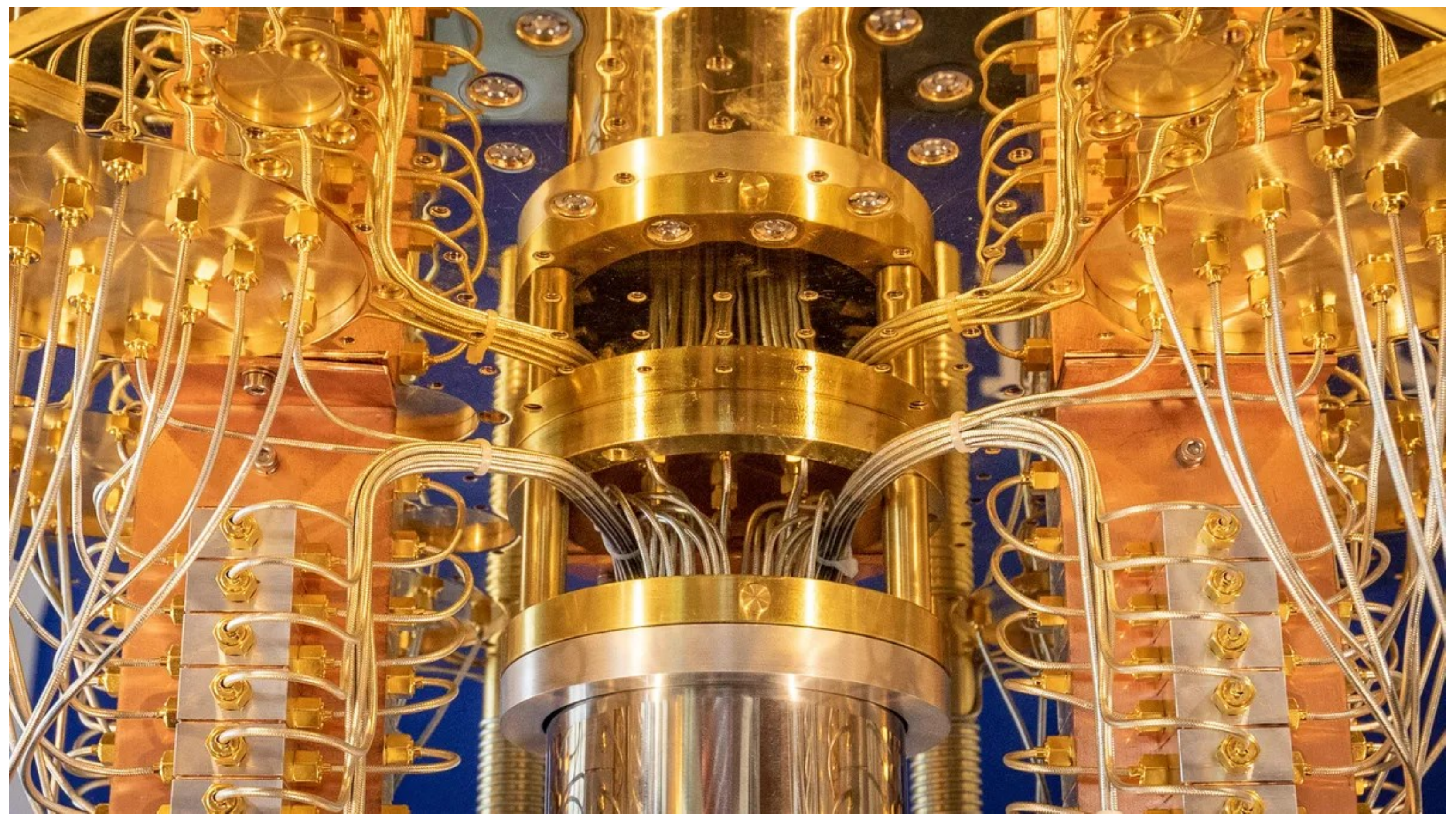
Can we do that?

Research Areas

This research project proposal builds at the intersection of two research areas:

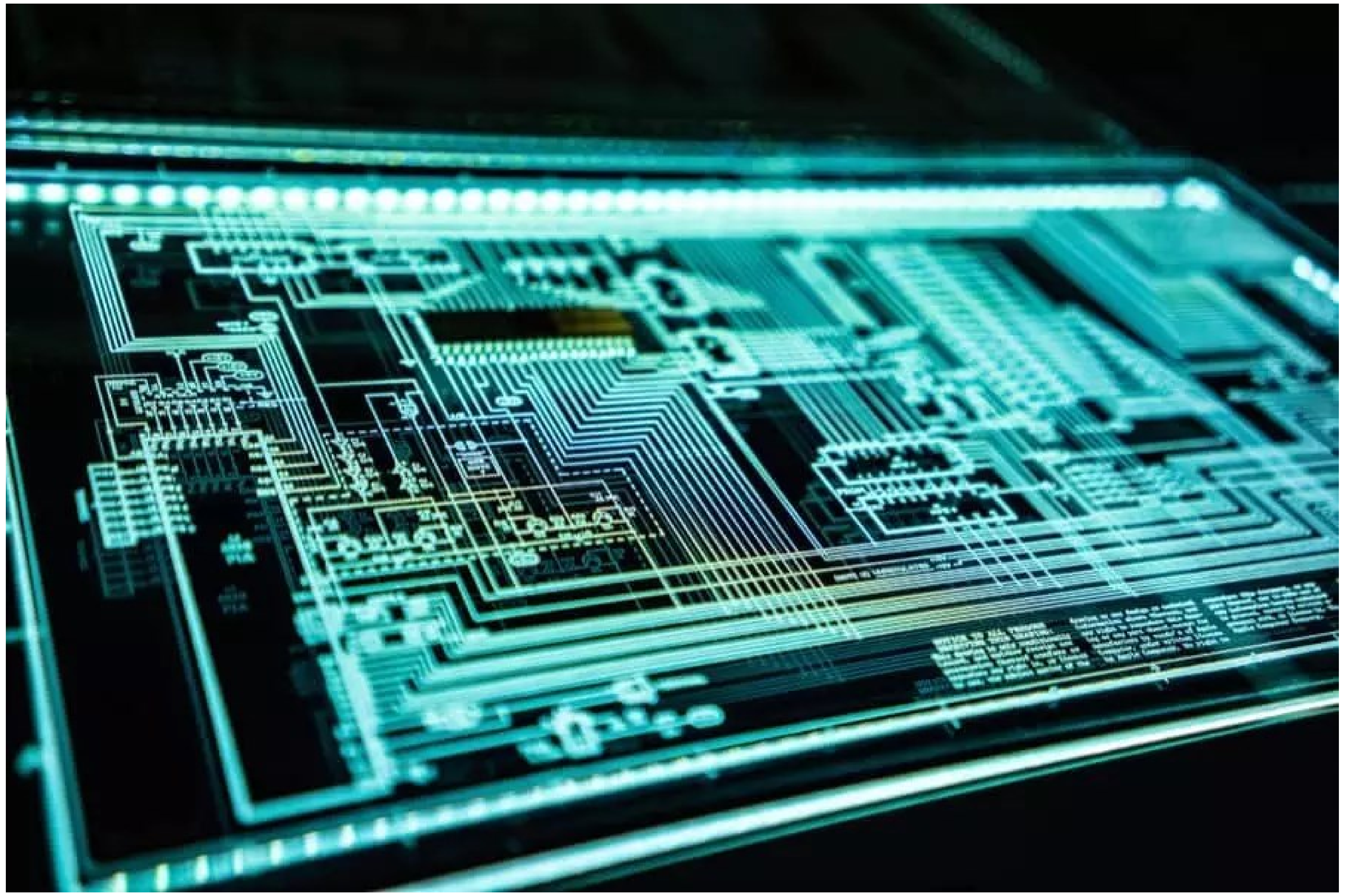
1. Electronic Design Automation (EDA)
2. Quantum Computing





Electronic Design Automation

EDA is a research field dedicated to developing algorithms and software tools for automating tasks related to the design of digital electronic circuits.



Electronic Design Automation

EDA is a research field dedicated to developing algorithms and software tools for automating tasks related to the design of digital electronic circuits.

- Frontend: logic design flow. Given a high-level description of the circuit, it produces a netlist, which is a document defining all the instances of the used blocks and their interconnections.

Electronic Design Automation

EDA is a research field dedicated to developing algorithms and software tools for automating tasks related to the design of digital electronic circuits.

- Frontend: logic design flow. Given a high-level description of the circuit, it produces a netlist, which is a document defining all the instances of the used blocks and their interconnections.

- Backend: physical design flow. Given a netlist, it defines the final schematic of the circuits.

Electronic Design Automation

EDA is a research field dedicated to developing algorithms and software tools for automating tasks related to the design of digital electronic circuits.

- Frontend: logic design flow. Given a high-level description of the circuit, it produces a netlist, which is a document defining all the instances of the used blocks and their interconnections.

- Backend: physical design flow. Given a netlist, it defines the final schematic of the circuits.

We will operate at the backend level.

Electronic Design Automation

Backend consists of several stages:

Electronic Design Automation

Backend consists of several stages:

- Technology Mapping: each logic block of the input netlist must be mapped to one or more library components

Electronic Design Automation

Backend consists of several stages:

- Technology Mapping: each logic block of the input netlist must be mapped to one or more library components
- Placement: establish how to place library components in the chip’s core

Electronic Design Automation

Backend consists of several stages:

- Technology Mapping: each logic block of the input netlist must be mapped to one or more library components
- Placement: establish how to place library components in the chip's core
- Routing: establish how to interconnect such components

Electronic Design Automation

Backend consists of several stages:

- Technology Mapping: each logic block of the input netlist must be mapped to one or more library components

- Placement: establish how to place library components in the chip's core

- Routing: establish how to interconnect such components

Main focus: Boolean Matching Problem in Technology Mapping

Electronic Design Automation

- Boolean Matching Problem:

Let f be the Boolean function computed by some element of a technology library. Let g be another Boolean function, called the target function. Boolean matching is the problem of determining whether f can be used to implement g. We assume that f and g are represented by Binary Decision Diagrams (BDDs).

Electronic Design Automation

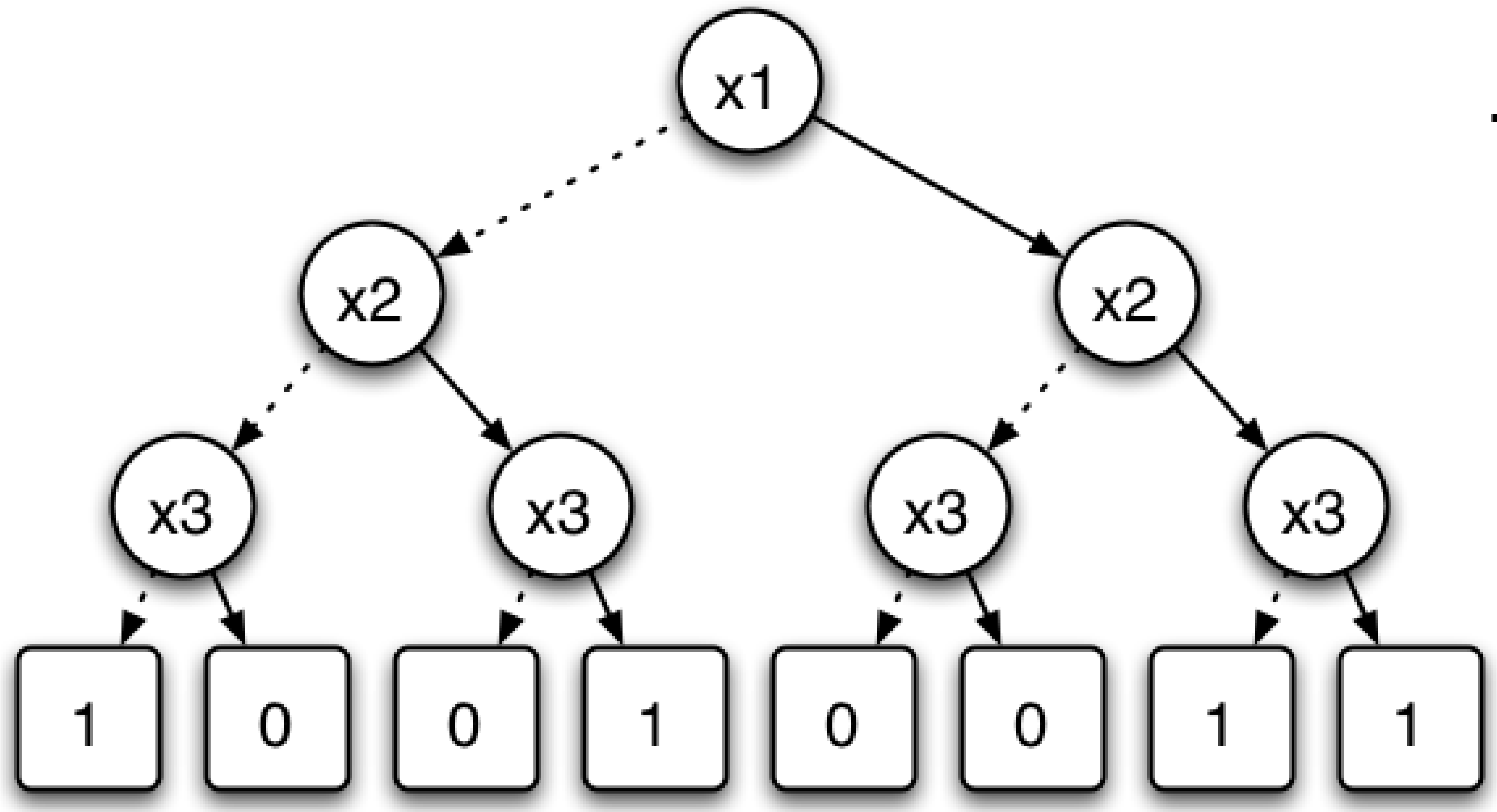
- Boolean Matching Problem:

Let f be the Boolean function computed by some element of a technology library. Let g be another Boolean function, called the target function. Boolean matching is the problem of determining whether f can be used to implement g. We assume that f and g are represented by Binary Decision Diagrams (BDDs).

- NPN-equivalence:

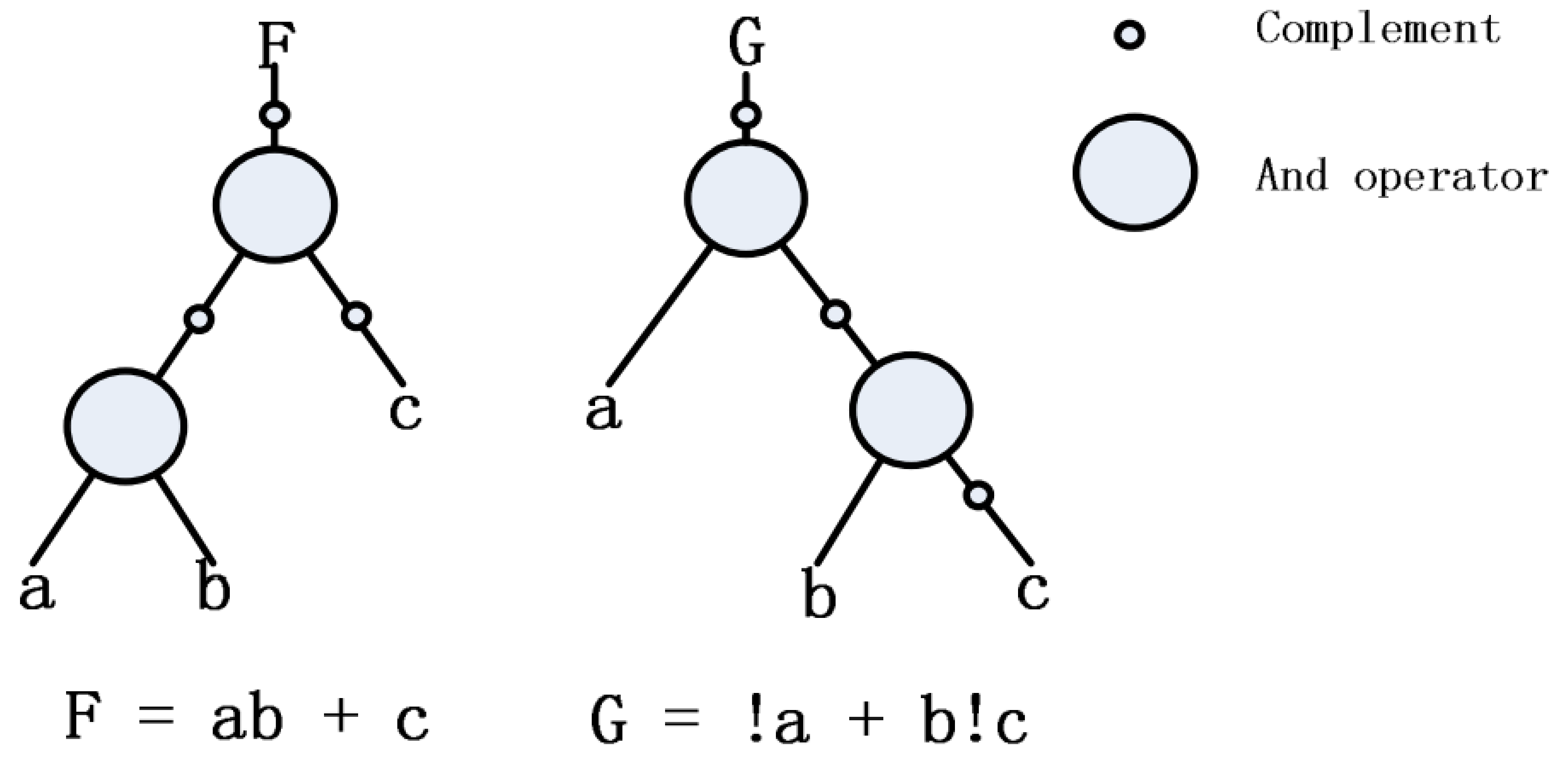
Given two single-output Boolean functions, f and g, with the same number of inputs, then f and g are said to be NPN-equivalent iff f can be formed from g by negating zero or more inputs, permuting inputs or negating the output, or by combinations of these transformations.

Electronic Design Automation



x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Binary Decision Diagram (BDD) and corresponding Truth Table



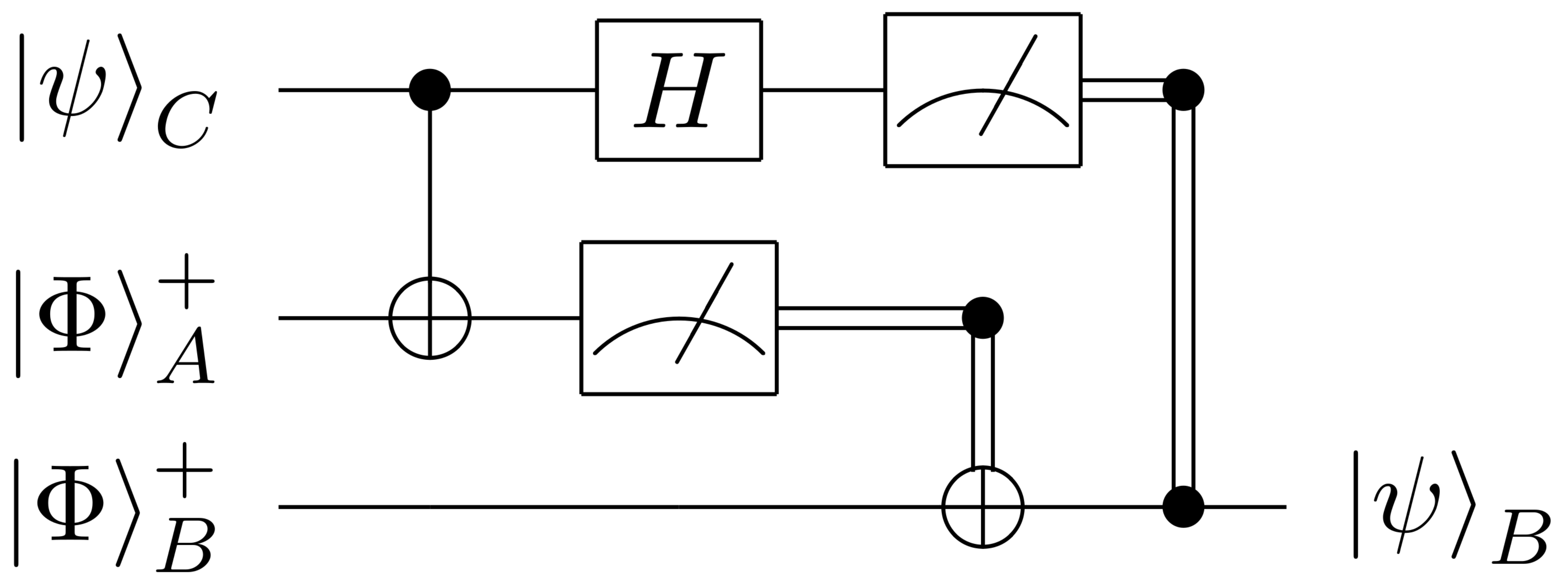
Two NPN-equivalent Boolean functions*

* from Huang, Zheng et al. “Fast Boolean matching based on NPN classification.” 2013 International Conference on Field-Programmable Technology (FPT) (2013): 310-313.

Quantum Computing

It is a model of computation whose operations can harness the phenomena of quantum mechanics, such as superposition, interference, and entanglement.

A quantum circuit, i.e., a network of quantum logic gates, is a complex linear-algebraic generalization of boolean circuits, in which bits are replaced by qubits.



Quantum circuit representing teleportation of a quantum state

Quantum Computing

Quantum algorithms have often proven to offer a noticeable speedup w.r.t. their analogue classical implementation, and are becoming an essential tool to tackle significant problems, whose complexity class prevents classical supercomputers from achieving correct solutions in a reasonable time.

Theoretically speaking, Quantum Computing has been proven to achieve at most a square-root speedup for NP-hard problems, which still results to be a desirable outcome, considering the current limits of classical solvers.

Quantum Computing

Quantum algorithms have often proven to offer a noticeable speedup w.r.t. their analogue classical implementation, and are becoming an essential tool to tackle significant problems, whose complexity class prevents classical supercomputers from achieving correct solutions in a reasonable time.

Theoretically speaking, Quantum Computing has been proven to achieve at most a square-root speedup for NP-hard problems, which still results to be a desirable outcome, considering the current limits of classical solvers.

The Boolean Matching Problem is NP-hard.

EDA - State of the Art

Three main approaches for the solution of the Boolean Matching Problem:

1. Reduction to canonical form

2. Signature Pruning

3. Spectral-based methods

EDA - Reduction to canonical form

Boolean Functions can be associated with their canonical form, and it has been proven that functions belonging to the same equivalence class are associated with the same canonical form.

Therefore, computing such a form for the target Boolean Function and comparing it against the canonical forms of the functions belonging to the given libraries is theoretically proven to solve this task.

EDA - Reduction to canonical form

1. Burch, and Long., *Efficient boolean function matching*. In 1992 IEEE/ACM International Conference on Computer-Aided Design (1992), pp. 408–411.

This paper proposes an algorithm for Boolean matching in which functions are translated to their canonical form and then compared. The algorithm to obtain a canonical form is polynomial in the size of the Binary Decision Diagram (BDD) of the input function.

Phase Matching:

$$(f \circ \varphi)\,(x_1, \ldots, x_n) = f(\varphi_1 \oplus x_1, \ldots, \varphi_n \oplus x_n)$$

All phase-related functions reduce to the same canonical form.

EDA - Reduction to canonical form

2. Agosta, G., Bruschi, F., Pelosi, G., and Sciuto, D., *A transform-parametric approach to boolean matching*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 28, 6 (2009), 805–817.

This paper focuses on P-equivalence Boolean matching.

For what concerns the canonical-form-based methods, they define classes of equivalence of Boolean functions under permutation, and consider the lexicographical maximum of the class as the canonical form of the class itself.

$$P = \left\{\pi : (x_{n-1}, \ldots, x_0) \mapsto \left(x_{\pi(n-1)}, \ldots, x_{\pi(0)}\right)\right\}$$

$$\mathcal{F}(f) = \max_{\pi \in P}\{f \circ \pi\}$$

EDA – Signature Pruning

Such a method performs a search in the space of the potentially NPN-equivalent Boolean Functions after reducing it by pruning: indeed, there exist specific signatures, i.e. alternative representations for functions, which can be proven to be invariant for NPN-transformations. Therefore, pruning can be done on the search tree, based on comparison of signatures.

EDA – Signature Pruning

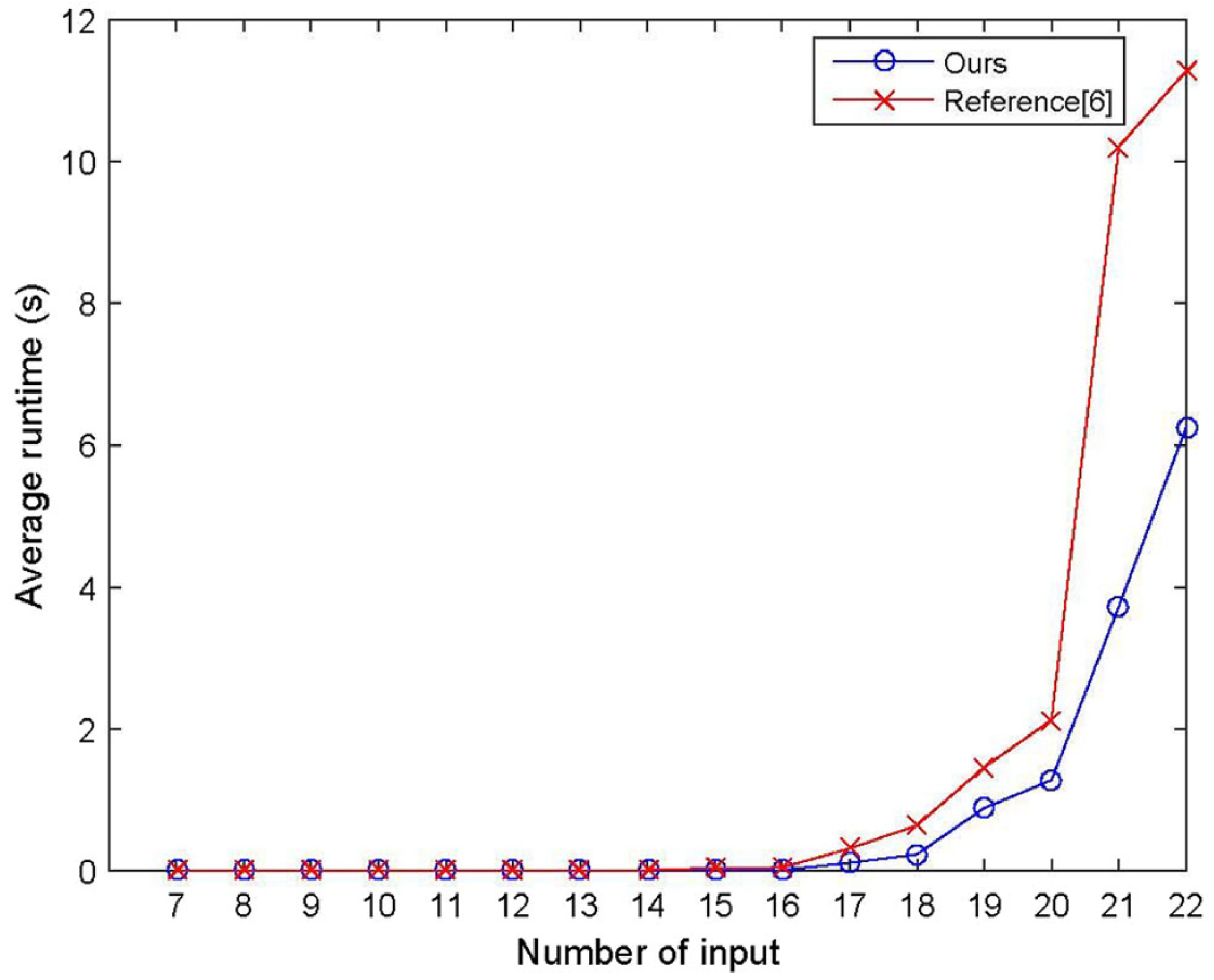
1. Zhang, J., Yang, G., Hung, W. N., Zhang, Y., and Wu, J., *An efficient npn boolean matching algorithm based on structural signature and shannon expansion*. Cluster Computing 22, 3 (2019), 7491–7506.

This paper presents a signature-based Boolean matching algorithm that takes advantage of the structural signature (SS) vector, which comprises a first-order signature value, two symmetry marks, and a group mark.

Definition 18 (Structural signature vector). *An n-variable Boolean function f has a structural signature (SS) vector $V_f = \{V_0, V_1, ..., V_{n-1}\}$ and $V_i = (|f_{x_i}|, |f_{\overline{x_i}}|, |C_i|, C_i, G_i)$ is the structural signature value of x_i.*

EDA – Signature Pruning

1. Zhang, J., Yang, G., Hung, W. N., Zhang, Y., and Wu, J., *An efficient npn boolean matching algorithm based on structural signature and shannon expansion*. Cluster Computing 22, 3 (2019), 7491–7506.

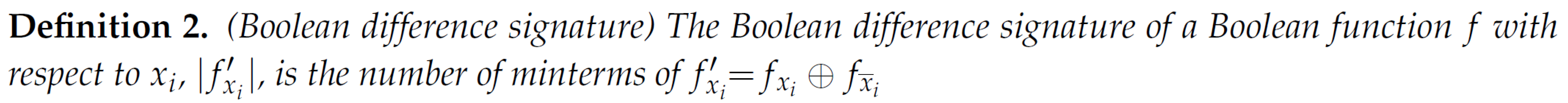


Comparison between state of the art and authors' work on average runtime

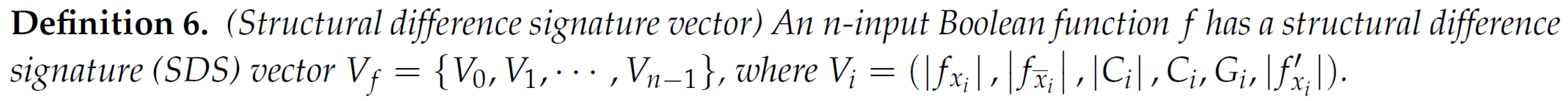
EDA – Signature Pruning

2. Zhang, J., Yang, G., Hung, W., Wu, J., and Zhu, Y., *A new pairwise npn boolean matching algorithm based on structural difference signature*. Symmetry 11 (12 2018), 27.

This paper considers the usage of the structural difference signature (SDS) of a Boolean function as a tool to reduce the search space in the Boolean matching process. Based on the previous paper, it reimplements the structural signature (SS) vector, by introducing the concept of Boolean difference, and the consequent Boolean difference signature. This signature can be used to distinguish variables.



Definition 2. *(Boolean difference signature) The Boolean difference signature of a Boolean function f with respect to x_i, $|f'_{x_i}|$, is the number of minterms of $f'_{x_i} = f_{x_i} \oplus f_{\overline{x}_i}$*



Definition 6. *(Structural difference signature vector) An n-input Boolean function f has a structural difference signature (SDS) vector $V_f = \{V_0, V_1, \cdots, V_{n-1}\}$, where $V_i = (|f_{x_i}|, |f_{\overline{x}_i}|, |C_i|, C_i, G_i, |f'_{x_i}|)$.*

EDA – Signature Pruning

2. Zhang, J., Yang, G., Hung, W., Wu, J., and Zhu, Y., *A new pairwise npn boolean matching algorithm based on structural difference signature*. Symmetry 11 (12 2018), 27.

Procedure 1 NPN Boolean Matching.

```
Input: $f$ and $g$
Output: 0 or 1
  function MATCHING($f, g$)
      Create BDD of $f$ and $g$
      $sp\_f = bddtrue, sp\_g = bddtrue, trans\_list = NULL$
      Compute $|f|$ and $|g|$
      if $|f| = |g|$ then
          if $|f| \neq |\overline{g}|$ then
              Return Handle_SDS($f, g$)
          else
              if Handle_SDS($f, g$)=1 then
                  Return 1
              else
                  Return Handle_SDS($f$,$\overline{g}$)
              end if
          end if
      else
          if $|f| = |\overline{g}|$ then
              Return Handle_SDS($f$,$\overline{g}$)
          else
              Return 0
          end if
      end if
  end function
```

EDA – Spectral-based methods

These methods exploit the property that NPN-equivalence of Boolean Functions translates into equivalence under coefficient permutation in the sequence domain of the Walsh Transform. Therefore, an exploration of the permutation tree of the Walsh coefficients is performed.

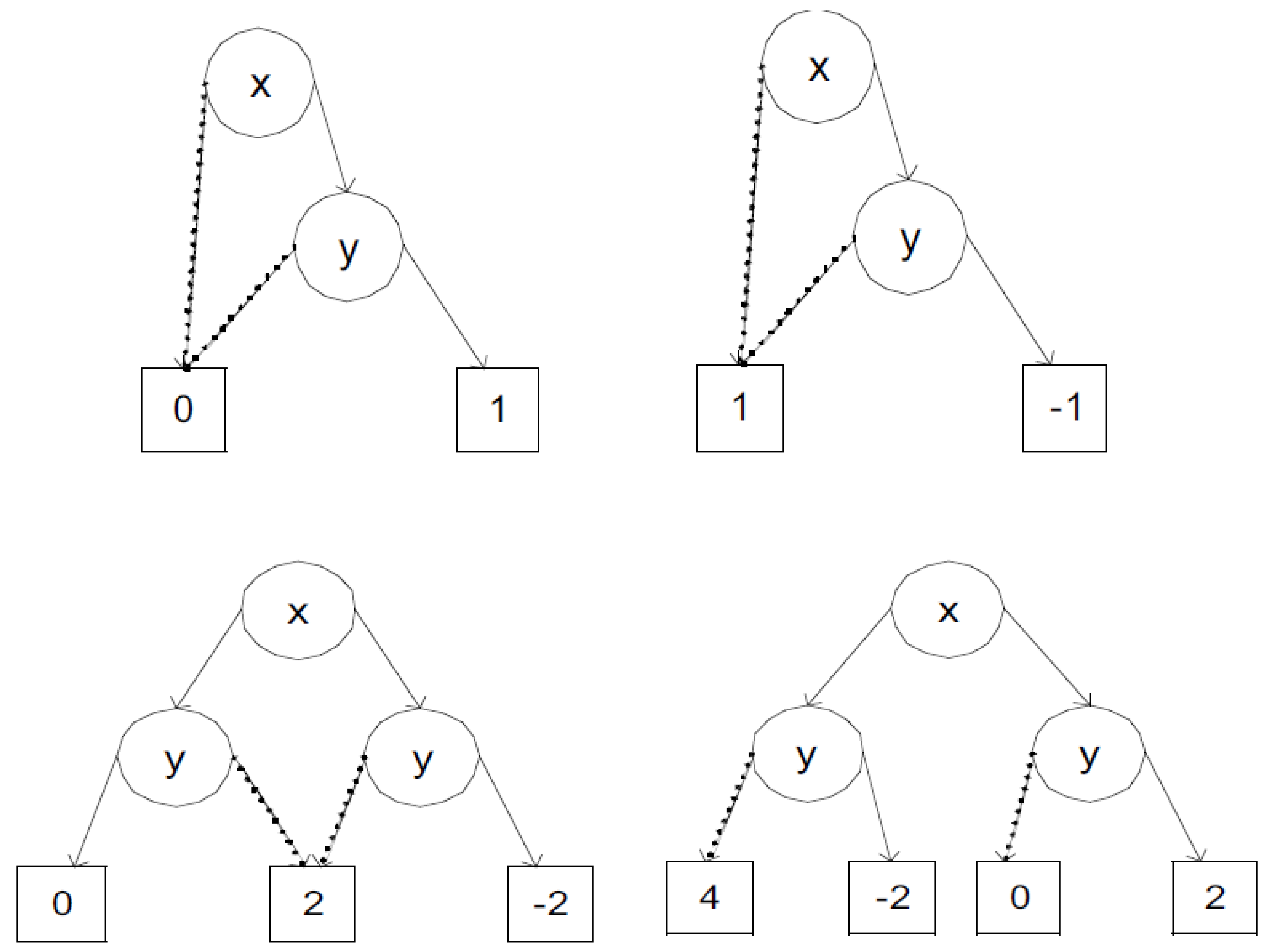
The Walsh spectrum is a Fourier representation of a Boolean function expanded about the orthogonal Walsh basis functions.

$$w_1 \equiv \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad w_n \equiv \begin{bmatrix} w_{n-1} & w_{n-1} \\ w_{n-1} & -w_{n-1} \end{bmatrix}$$

EDA – Spectral-based methods

1. Moore, J., Fazel, K., Thornton, M. A., and Miller, D. M., *Boolean function matching using walsh spectral decision diagrams*. In 2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software (2006), pp. 127–130.

This paper considers two approaches for Boolean matching. They are the usage of Luks' hypergraph method and the Walsh Spectral Decision Diagram (SDD). The latter is described and implemented by the authors, who show its speedup w.r.t. the former solution.



Computation of the Spectral Decision Diagram starting from the Binary Decision Diagram

EDA – Spectral-based methods

2. Agosta, G., Bruschi, F., Pelosi, G., and Sciuto, D., *A transform-parametric approach to boolean matching*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 28, 6 (2009), 805–817.

This paper focuses on P-equivalence Boolean matching.

For what concerns spectral-based methods, it considers the usage of the Walsh coefficients. The Walsh Transform is defined by means of the Hadamard Matrix. The cofactor representation uses, in particular, a C_n matrix, which appears to be very similar to the Hadamard Matrix.

$$C_1 = [1] \qquad C_{2n} = \begin{bmatrix} C_n & 0 \\ C_n & C_n \end{bmatrix}$$

QC – Max-Cut Problem

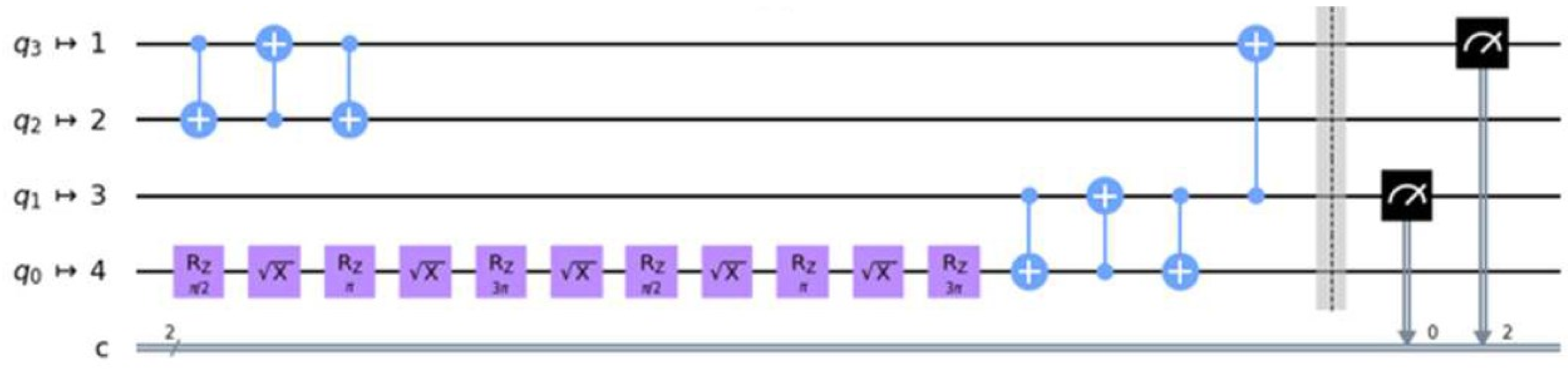
1. Ayaan Verghese, David Byron, Andreas Amann, and Emanuel Popovici, *Max-cut problem implementation and analysis on a quantum computer*. In 2022 33rd Irish Signals and Systems Conference (ISSC), pages 1-6. IEEE, 2022.

This paper presents an approach to the quantum acceleration of the Max-Cut Problem, which is a combinatorial optimization problem consisting in partitioning two sets of graph nodes such that the number of edges between the sets is maximum.

The Quantum Approximate Optimization Algorithm (QAOA) has been used to achieve the final quantum circuit, since it is a quantum algorithm specialixed in solving combinatorial problems. A noise analysis due to qubits' physical properties has been performed too.

QC – Max-Cut Problem

1. Ayaan Verghese, David Byron, Andreas Amann, and Emanuel Popovici, *Max-cut problem implementation and analysis on a quantum computer*. In 2022 33rd Irish Signals and Systems Conference (ISSC), pages 1-6. IEEE, 2022.



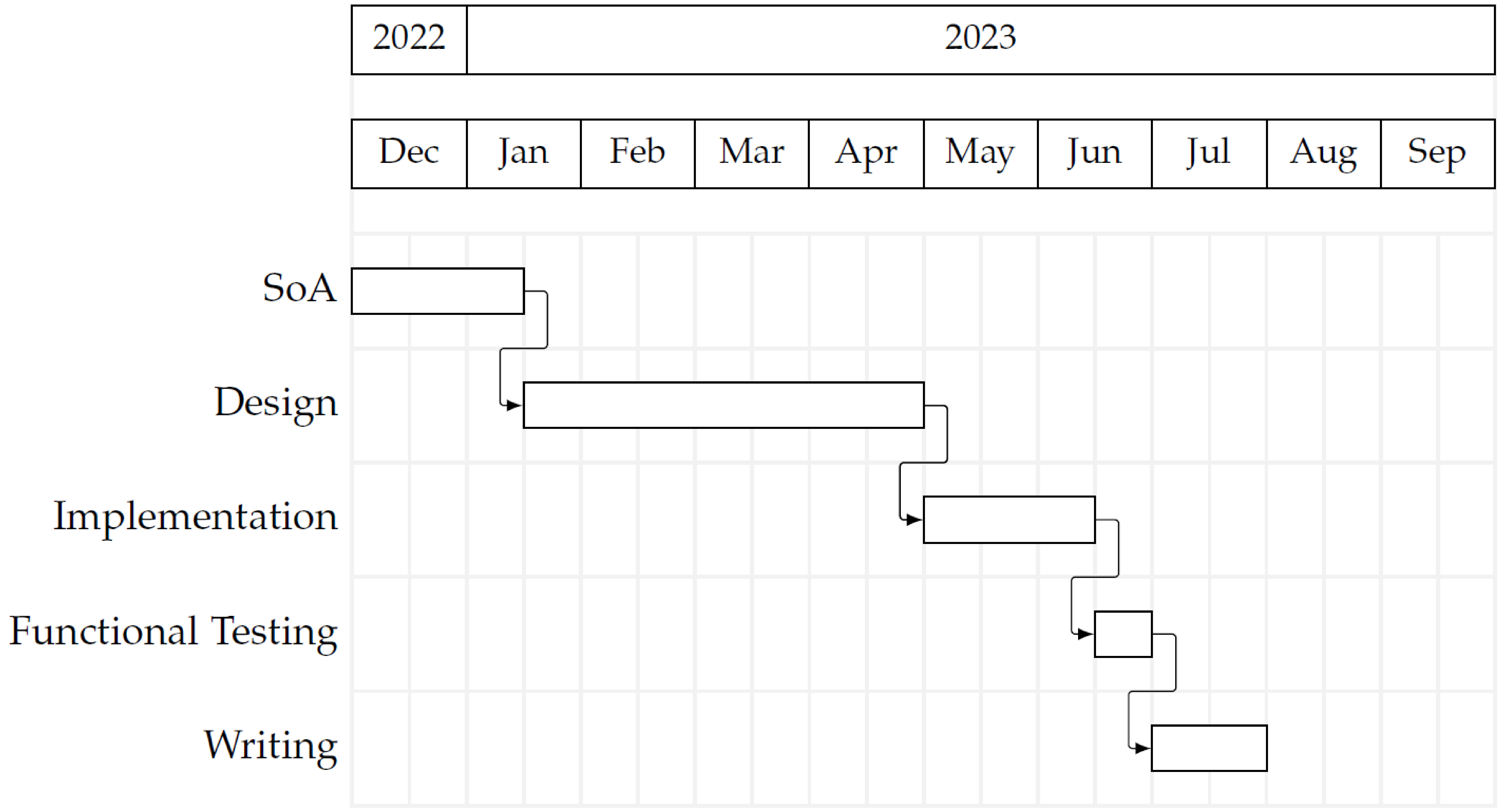
Quantum circuit implementing the max-cut solver

Research plan

The research may be split into the following stages:

1. completion of the study of the State of the Art
2. design of the solution
3. implementation
4. functional testing
5. writing

Research plan



Gantt diagram of the tasks of the research proposal.
Waterfall development model

Thanks for your attention