Design Automation for Reversible Quantum-Flux-Parametron Logic
-- Towards Energy-Efficient Superconducting Circuits

Tsung-Yi Ho

tyho@cse.cuhk.edu.hk

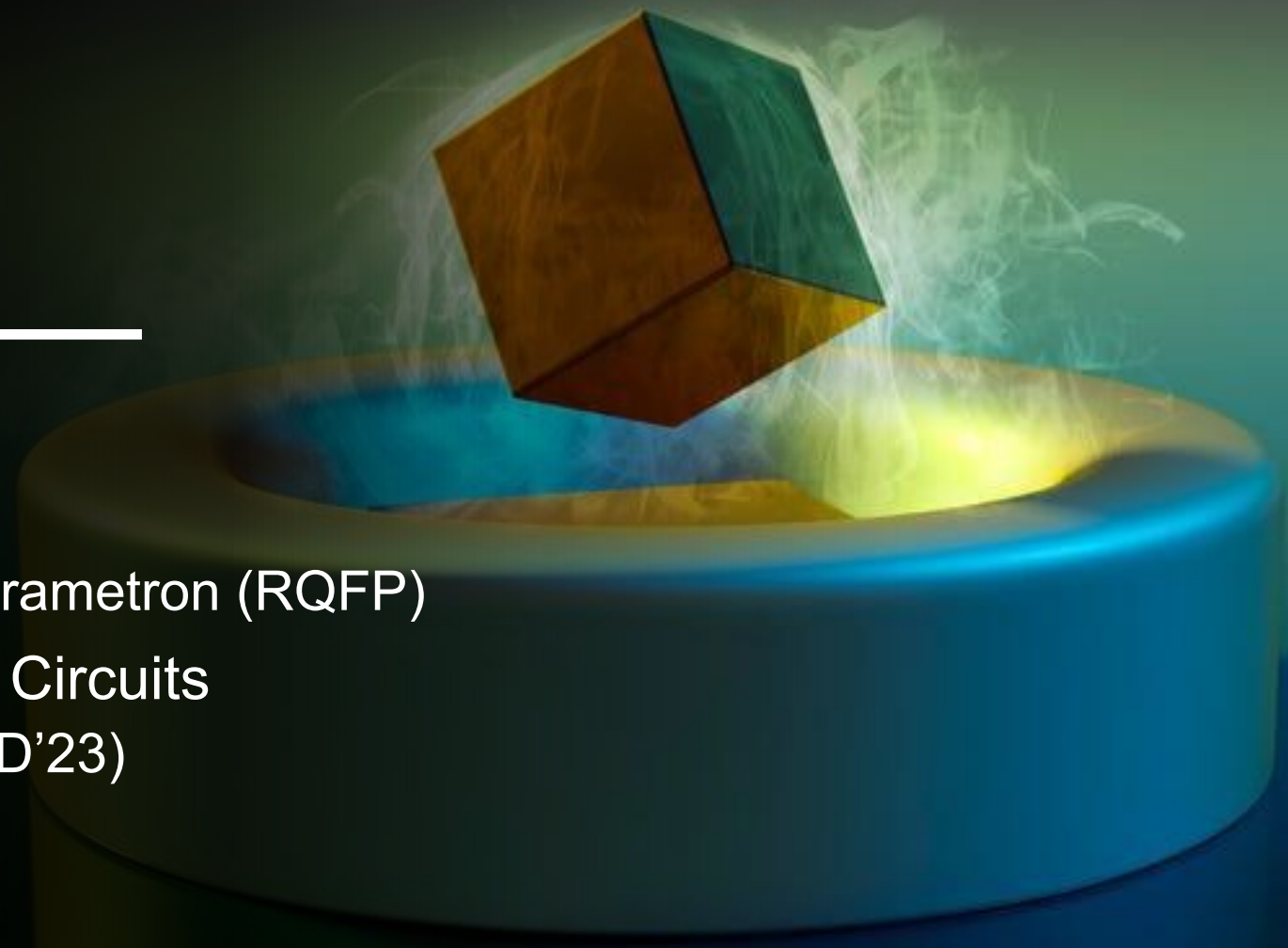
Intelligent DEsign Automation Lab (IDEAL)

Department of Computer Science and Engineering

The Chinese University of Hong Kong, Shatin, Hong Kong

Content

- Introduction
- Extreme Energy Efficiency
  - Reversible Quantum-Flux-Parametron (RQFP)
- Automatic Synthesis of RQFP Circuits
  - Exact Logic Synthesis (ICCAD'23)
  - RCGP (DAC'24, TCAD25)
- Conclusion and Future Work



[Superconductor figure] https://www.popularmechanics.com/science/energy/a44787012/room-temperature-superconductors-explained/

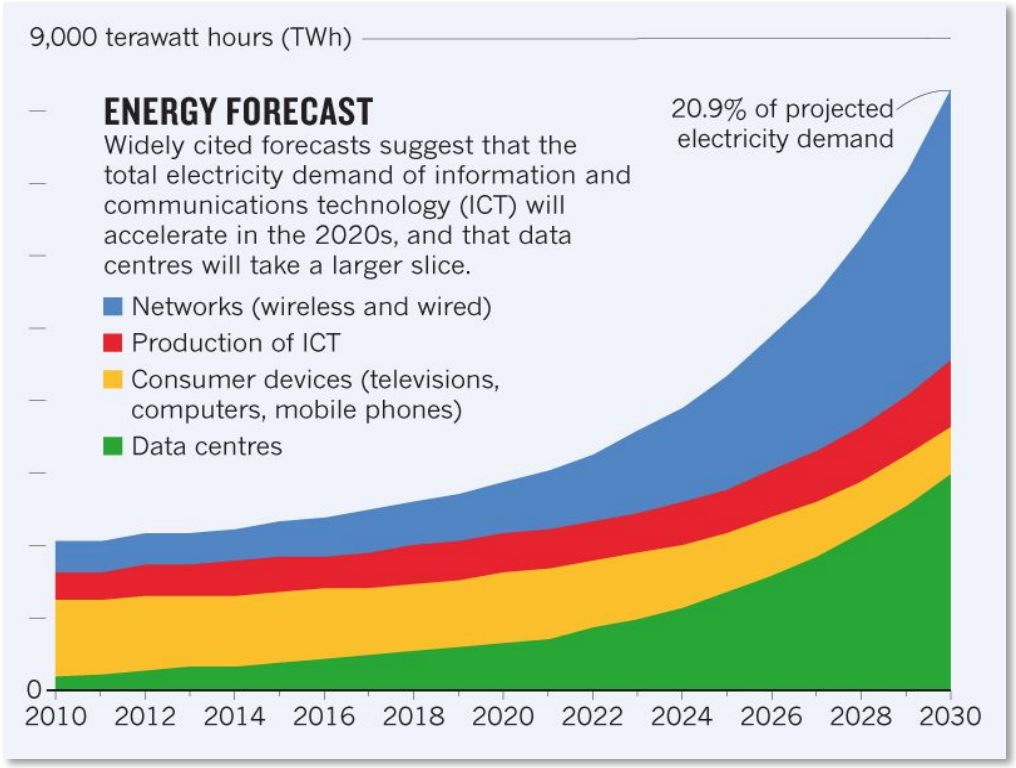
Introduction

Overview and research background

Energy Efficiency Calling

Trend of rising electricity demand of information and communications technology[1]

Currently 10% of the total electric power



- NVIDIA's projected shipment: 1.5 million AI server units annually by 2027, consuming at least 85.4 TWh of electricity annually when running at full capacity. [2-3]
- Bitcoin's electricity consumption is more than what many small countries use in a year. [4]
- Around the globe, data centers currently account for about 1 to 1.5 percent of global electricity use. (460 TWh in 2022) [5]

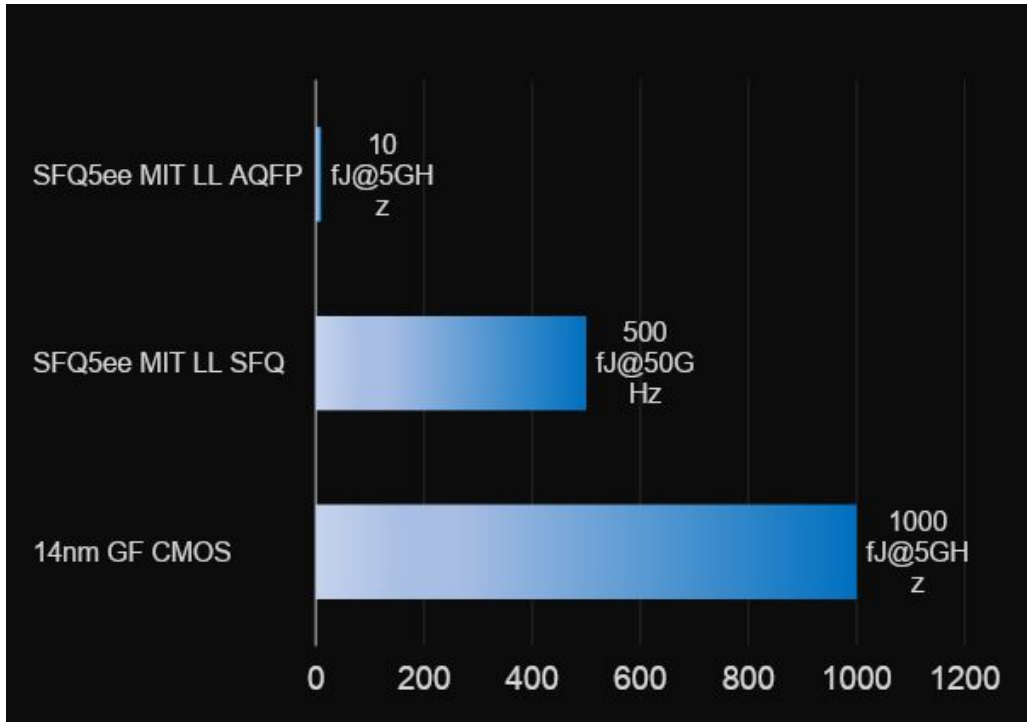
[1] Jones N. How to stop data centres from gobbling up the world's electricity. Nature. vol. 561, no. 7722, pp. 163–166, Sep. 2018.
[2] Bary, E. Nvidia is 'dominating' and could unlock $300 billion in AI revenue by 2027, analyst says. MarketWatch. July 24, 2023.
[3] Alex de Vries, The growing energy footprint of artificial intelligence, vol. 7, no. 10, pp. 2191-2194, 2023.
[4] University of Cambridge Bitcoin Electricity Consumption Index: https://ccaf.io/cbeci/index/comparisons (June 2025).
[5] International Energy Agency, https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks#overview.

Superconductor Electronics

- High speed and low-power consumption.

Technology	Switching Energy	Frequency
CMOS (Charge-transfer based)		
Superconducting Single Flux Quantum Logic (RSFQ)		
Superconducting Adiabatic Quantum-Flux-Parametron Logic (AQFP)		

$K_B T \ln 2$ is the minimum bit energy per switching event according to Landauer's principle[7]

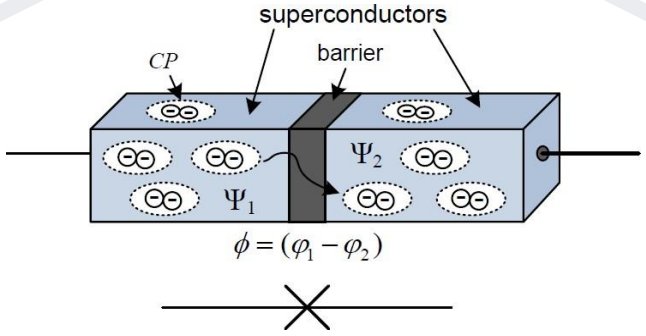


Superconductor electronics become an attractive candidate for future computing systems due to their characteristics of high speed and low-power consumption.

[6] Massoud Pedram, High Performance Computing Using Superconductor Electronics, DAC - Tutorial, 2024.
[7] R. Landauer, "Irreversibility and heat generation in the computing process", IBM J. Res. Develop., vol. 5, no. 3, pp. 183-191, Jul. 1961.

Superconductor Electronics Technology

Josephson junction (JJ)



Switching time: ~1 ps

Characteristic voltage: 0.5 – 1 mV

Characteristic energy: $\sim 10^{-19}$ J

Applications

Digital computing

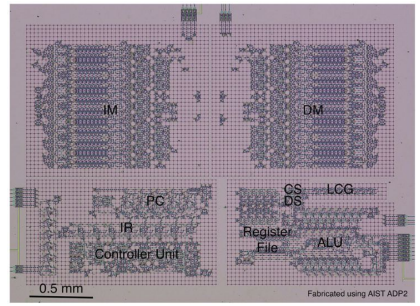
-- Microprocessor [8-9]

Voltage standard [10]

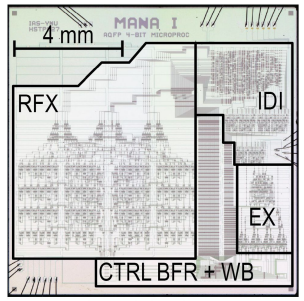
Single photon detection [11]

Quantum computing [12]

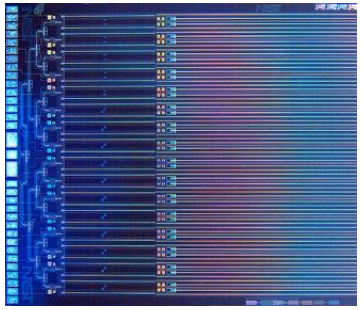
CORE e2h RSFQ Microprocessor [8]



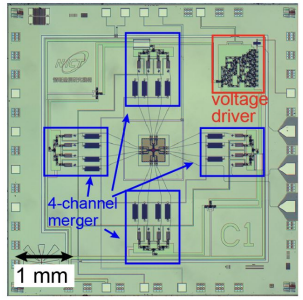
MANA Microprocessor [9]



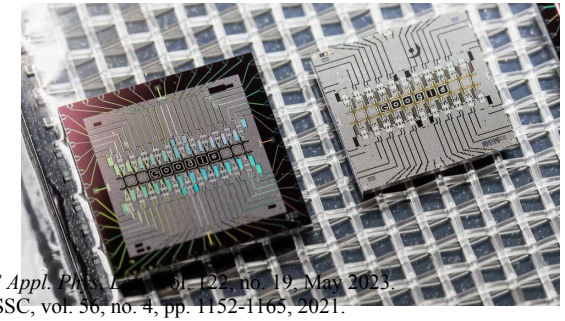
NIST Voltage Standard Chip [10]



SNSPD + SFQ Merger [11]



Google's Quantum Sycamore Chip [12]



[8] Tanaka *et al*., "Execution of stored programs by a rapid single-flux-quantum random-access-memory-embedded bit-serial microprocessor using 50-GHz clock frequency," *Appl. Phys. Lett.*, vol. 122, no. 19, May 2023.
[9] Ayala *et al*., "MANA: A Monolithic Adiabatic iNtegration Architecture Microprocessor Using 1.4-zJ/op Unshunted Superconductor Josephson Junction Devices," IEEE JSSC, vol. 56, no. 4, pp. 1152-1165, 2021.
[10] Benz *et al*., "Application of the Josephson effect to voltage metrology," in Proceedings of the IEEE, vol. 92, no. 10, pp. 1617-1629, Oct. 2004.
[11] Miyajima *et al*., "Single-flux-quantum signal processors monolithically integrated with a superconducting nanostrip single-photon detector array," *Appl. Phys. Lett.*, vol. 122, no. 18, May 2023.
[12] Arute *et al*., "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, Oct. 2019.

Superconductor Electronics Technology

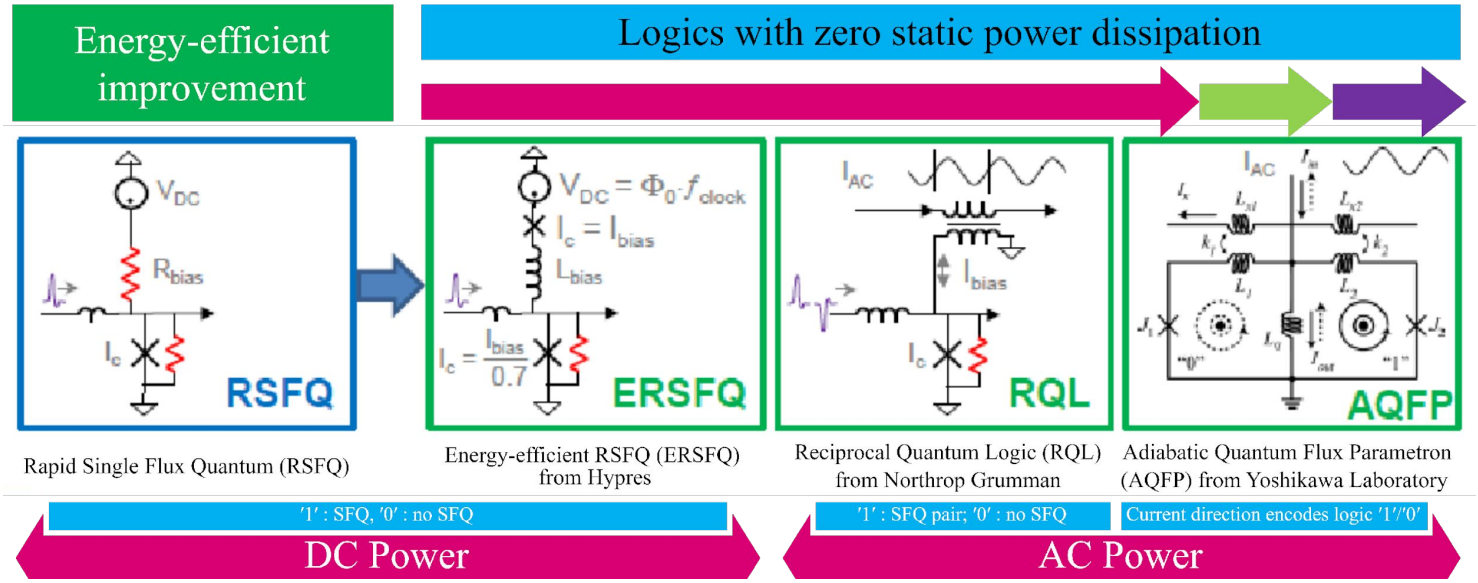
Superconducting logic circuit family:

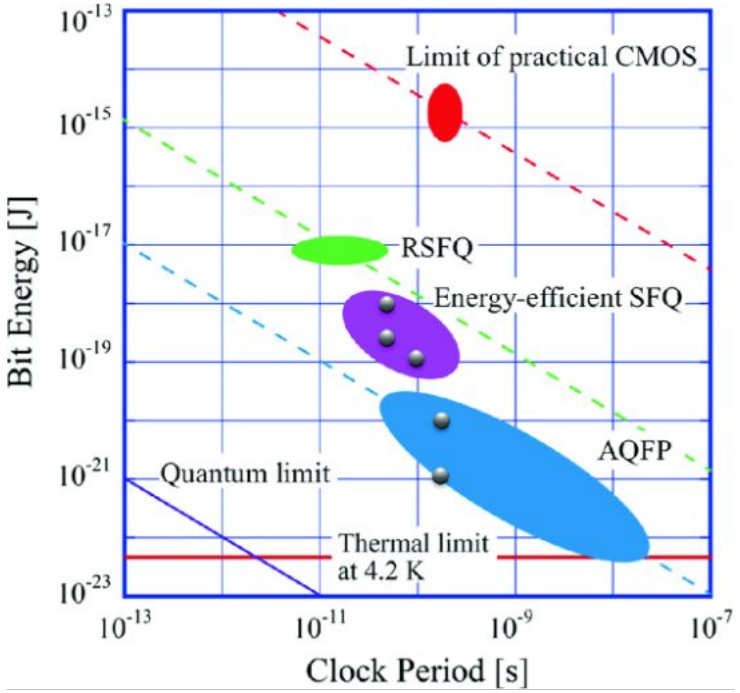
01 Rapid single flux quantum (RSFQ)[13] — **Significant static power** ✖

02 Energy-efficient RSFQ (ERSFQ)[14]

03 Reciprocal quantum logic (RQL)[15]

04 **Adiabatic quantum-flux-parametron (AQFP)[16]** — **Ultra-high energy efficiency** ✔





Comparison of energy-delay products of AQFP, ERSFQ, RSFQ, and CMOS circuits. [17]

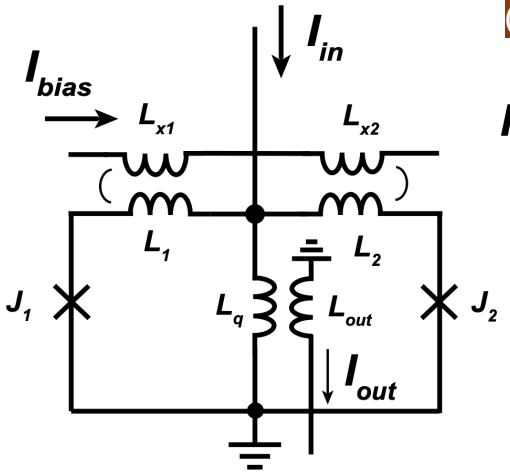
[13] K. Likharev and V. Semenov, "RSFQ logic/memory family: a new Josephson-junction technology for sub-terahertz-clock-frequency digital systems," IEEE TAS, vol. 1, no. 1, pp. 3–28, 1991.
[14] O. A. Mukhanov, "Energy-efficient single flux quantum technology," IEEE Transactions on Applied Superconductivity, vol. 21, no. 3, pp. 760–769, 2011.
[15] Q. P. Herr, A. Y. Herr, O. T. Oberg, and A. G. Ioannidis, "Ultra-low-power superconductor logic," Journal of Applied Physics, vol. 109, no. 10, p. 103903, 2011.
[16] Takeuchi *et al.*, "An adiabatic quantum flux parametron as an ultra-low-power logic device," Superconductor Science and Technology, vol. 26, no. 3, p. 035010, Jan 2013.
[17] O. Mukhanov, N. Yoshikawa, I. P. Nevirkovets, and M. Hidaka, "Josephson Junctions for Digital Applications," Cham: Springer International Publishing, pp. 611–701, 2019.

Adiabatic Quantum-Flux-Parametron (AQFP)

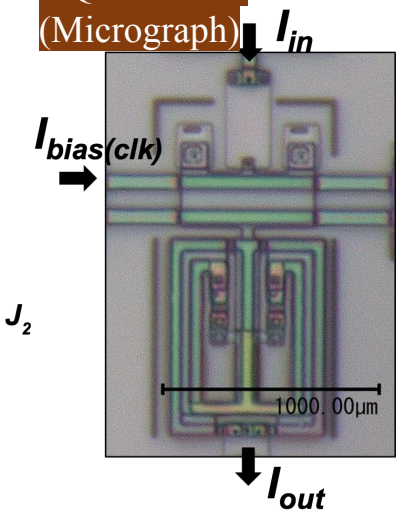
Ultra-high energy efficiency

- Very small switching energy due to adiabatic operation.
  - AQFP gate switching energy ranges from $\mathbf{10^{-20}}$ J to $\mathbf{10^{-21}}$ J at 5 GHz operation.[18]
- Clock speeds on par with state-of-the-art CMOS logic (5-10 GHz)
  - Typically, 5 GHz for adiabatic switching
- A 2.5 GHz prototype AQFP-based processor (MANA) has achieved an energy efficiency that was **80 times** that of 7-nm FinFET equivalent technology, even accounting for the cooling.[9]

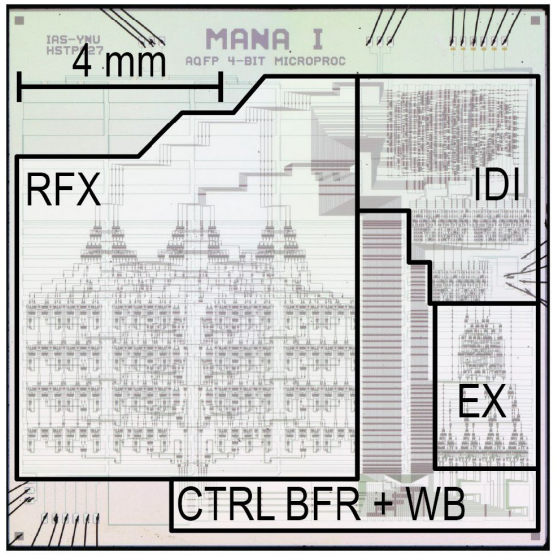
AQFP Device (Schematic)



AQFP Device (Micrograph)



AQFP buffer is the basic component with AC I_{bias} as its excitation current and the clock signal.

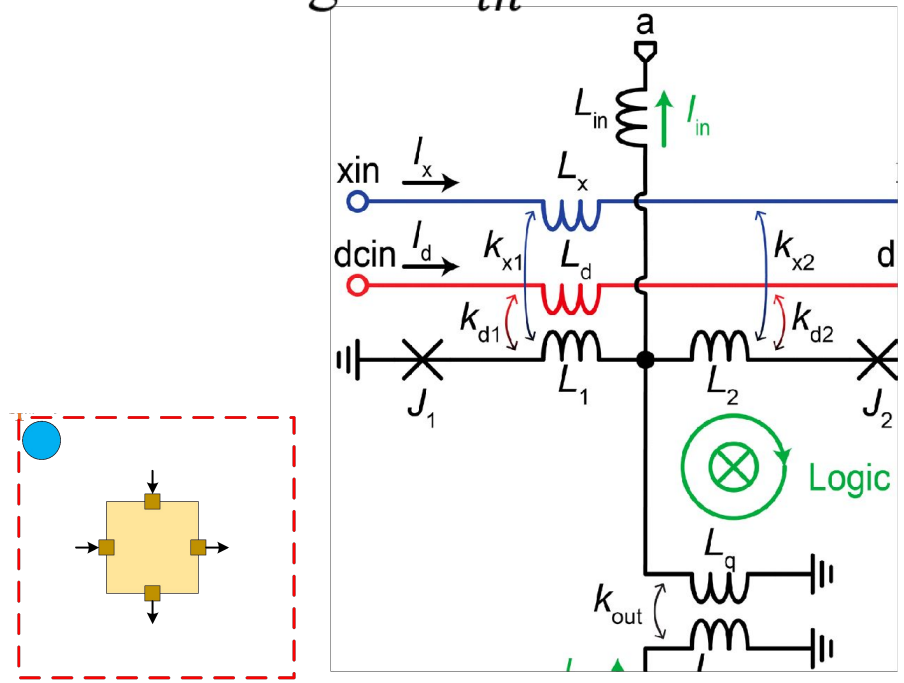


[18] Takeuchi *et al.*, "An adiabatic superconductor 8-bit adder with $24k_BT$ energy dissipation per junction," Applied Physics Letters, vol. 114, no. 4, p. 042602, 2019.

Fundamental Component – AQFP Buffer

AQFP buffer:

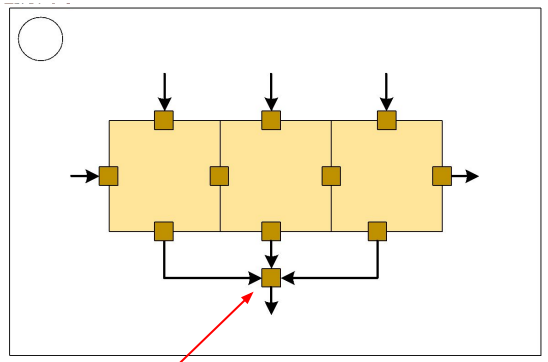
- Data output (q) value depends on the direction of the data input (a) current I_{in}.
- AC I_x serves as the excitation current and the clock signal X_{in}.
- DC I_d provides +/- offset to AC.



$+I_{in}$ → SFQ stored in left loop, logic '1'.
$-I_{in}$ → SFQ stored in right loop, logic '0'.

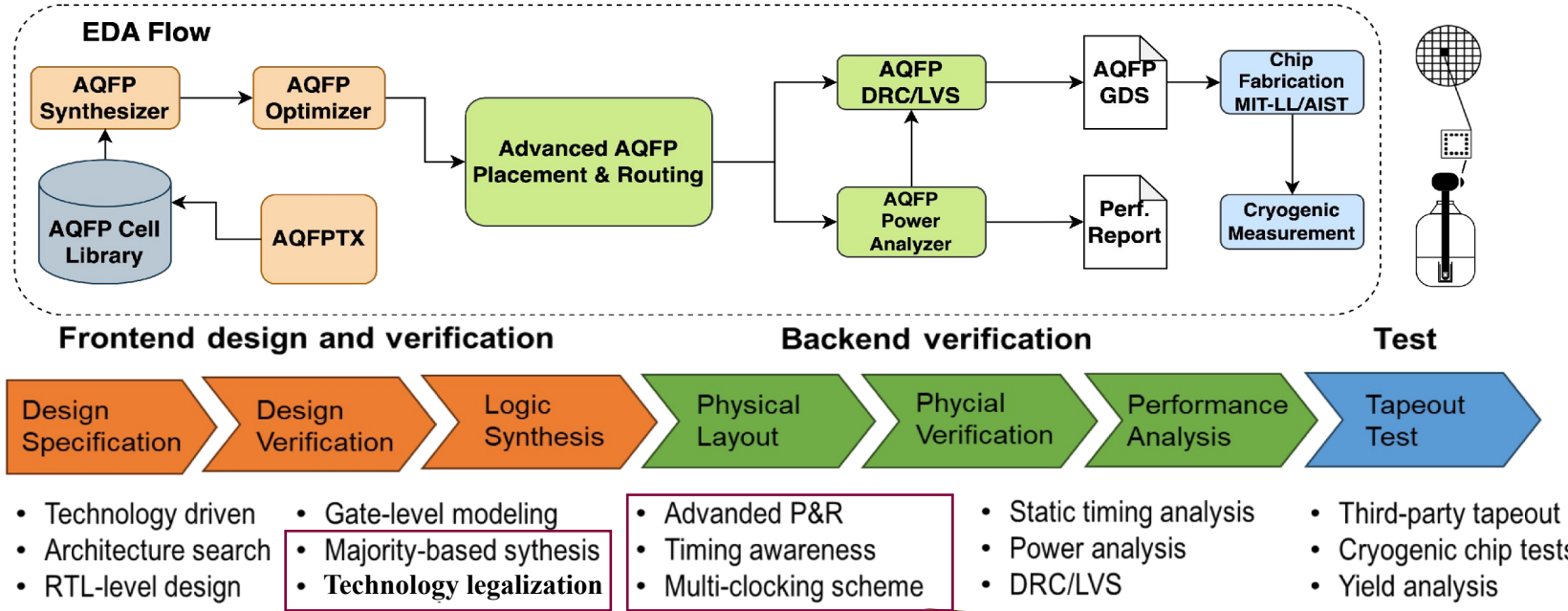
Make up logic cells using AQFP buffers

- 3-input majority gate



3:1-branch $\mathrm{M}(a,b,c) = ab + ac + bc$

AQFP EDA Tool Chain Development



Majority Logic Synthesis

DATE'23 Bayesian optimization combining fan-out optimization and majority-based synthesis

Technology Legalization

ICCAD'21 Fan-out optimization using dynamic programming

ASPDAC'23 and TCAD Fan-out optimization using dynamic programming and logic level assignment using ILP

Placement

ICCAD'20 Analytical global placement and row-wise detailed placement

ICCAD'23 Placement optimization for delay-line clocking scheme

[ICCAD'21] C. -Y. Huang, Y. -C. Chang, M. -J. Tsai and T. -Y. Ho, "An Optimal Algorithm for Splitter and Buffer Insertion in Adiabatic Quantum-Flux-Parametron Circuits," ICCAD, 2021.
[ASPDAC'23] Rongliang Fu, Mengmeng Wang, Yirong Kan, Nobuyuki Yoshikawa, Tsung-Yi Ho, and Olivia Chen, "A Global Optimization Algorithm for Buffer and Splitter Insertion in Adiabatic Quantum-Flux-Parametron Circuits," ASPDAC, 2023.
[DATE'23] Rongliang Fu, Junying Huang, Mengmeng Wang, Yoshikawa Nobuyuki, Bei Yu, Tsung-Yi Ho, and Olivia Chen, "BOMIG: A Majority Logic Synthesis Framework for AQFP Logic," DATE, 2023.
[TCAD] Rongliang Fu, Mengmeng Wang, Yirong Kan, Olivia Chen, Nobuyuki Yoshikawa, Bei Yu, Tsung-Yi Ho, "Buffer and Splitter Insertion for Adiabatic Quantum-Flux-Parametron Circuits," TCAD, 2024.
[ICCAD'20] Y. -C. Chang, H. Li, O. Chen, Y. Wang, N. Yoshikawa and T. -Y. Ho, "ASAP: an analytical strategy for AQFP placement", ICCAD, 2020.
[ICCAD'23] Rongliang Fu, Olivia Chen, Bei Yu, Nobuyuki Yoshikawa and Tsung-Yi Ho, "DLPlace: A Delay-Line Clocking-based Placement Framework for AQFP Circuits," ICCAD, 2023.

Extreme Energy Efficiency Calling

- However, the minimum energy dissipation of these irreversible superconducting digital circuits, even AQFP, could still **not be lower than the order of K_BT ln 2**, according to Landauer's principle.

  Rolf Landauer: Whenever using a logically irreversible gate, we dissipate energy into the environment.

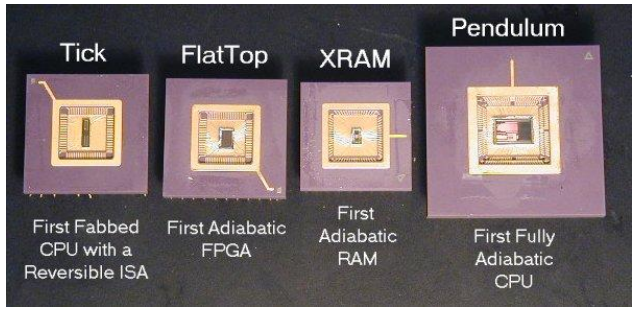
  - Information is physical
  - Information loss = energy loss

- How can we break the minimum energy bound of $K_B T \ln 2$?
  - **Reversible computing:** No energy dissipation for logic operations in reversible computing without the reduction in information entropy, under ideal physical circumstances
    - **Logical reversibility:** A **one-to-one mapping** between vectors of inputs and outputs.
    - **Physical reversibility: no increase in physical entropy** (adiabatic computing)

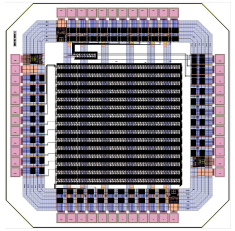
[19] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Thermodynamic Study of Energy Dissipation in Adiabatic Superconductor Logic," Phys. Rev. Appl., vol. 4, no. 3, p. 034007, Sep. 2015.
[20] C. H. Bennett, "Logical reversibility of computation," IBM Journal of Research and Development, vol. 17, no. 6, pp. 525–532, 1973.

Reversible Logic Circuits

- Younis & Knight, 1994 [21]
  - Simplified 3-level adiabatic CMOS design family (SCRL)
- Subsequent works at MIT, 1995-99 [22]
  - Various chips designed using SCRL
- Work at Sandia, 2020 [23]
  - 2LAL (two-level adiabatic logic) shift register
- Work at Prof. Yoshikawa's group, 2014- [24-25]
  - **Reversible Quantum Flux Parametron (RQFP)**
  - 1-bit RQFP full adder (RFA)[26] **dissipate < *kT* ln 2** (even at T = 4K), at speeds on the order of 10 MHz.

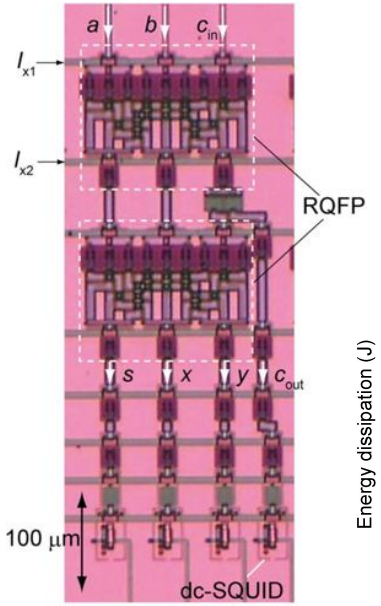


MIT

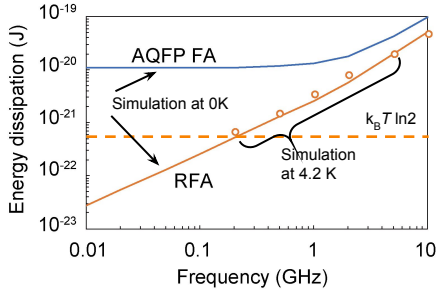


2LAL shift register

Adiabatic CMOS



RQFP full adder



Superconducting

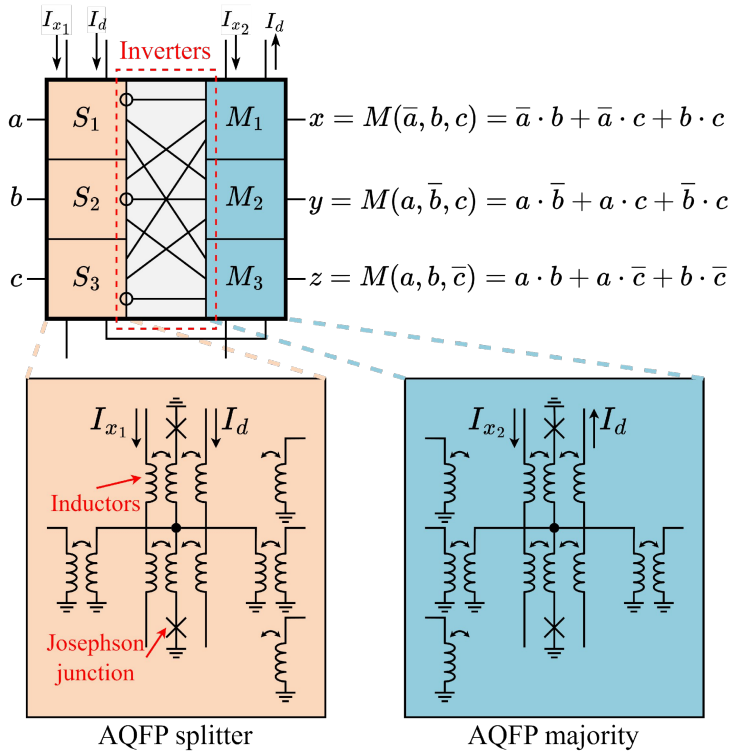
[21] Saed G. Younis, "Asymptotically zero energy computing using split-level charge recovery logic," Ph.D. thesis, MIT, 1994.
[22] Frank et al., "Reversible Computing with Fast, Fully Static, Fully Adiabatic CMOS," ICRC, 2020.
[23] M. P. Frank *et al.*, "Special Session: Exploring the Ultimate Limits of Adiabatic Circuits," *ICCD*, pp. 21-24, 2020.
[24] Takeuchi *et al.*, "Reversible logic gate using adiabatic superconducting devices", Scientific Reports, vol. 4,p. 6354, 2014.
[25] Takeuchi *et al.*, "Reversibility and energy dissipation in adiabatic superconductor logic," Scientific Reports, vol. 7,p. 75, 2017.
[26] Yamae *et al.*, "A reversible full adder using adiabatic superconductor logic," Superconductor Science and Technology, vol. 32, no. 3, p. 035005, 2019.

Extreme Energy Efficiency Reversible Quantum-Flux-Parametron

First practical reversible logic gate

Reversible Quantum-Flux-Parametron (RQFP)

- The **first practical reversible logic gate**[24-25] using AQFP technology.
- Its logical and physical reversibility has been demonstrated experimentally.
- Composed of AQFP-based majority and splitter cells.



physical reversibility

Input			Output		
a	b	c	x	y	z
0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	1	0	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	0	1	0
1	1	0	0	0	1
1	1	1	1	1	1

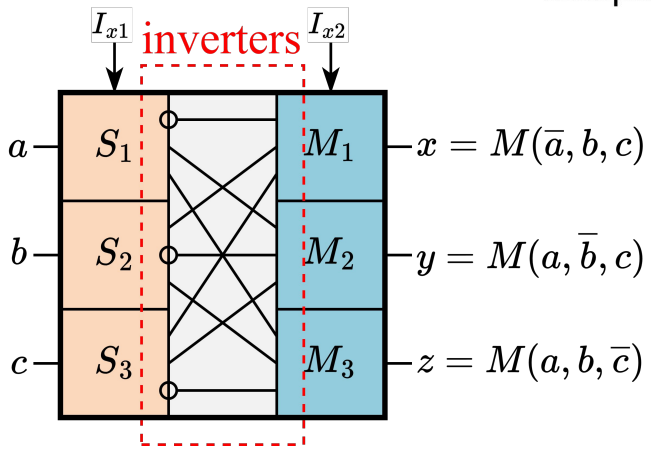
where $M(a,b,c) = ab + ac + bc$

$R(a,b,c) = \{M(\bar{a},b,c), M(a,\bar{b},c), M(a,b,\bar{c})\}$

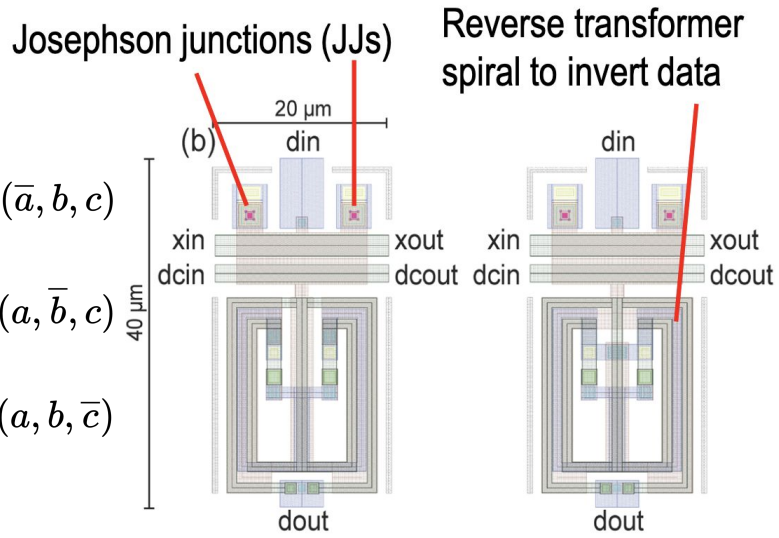
logical reversibility

Characteristics of RQFP Logic − Functionality

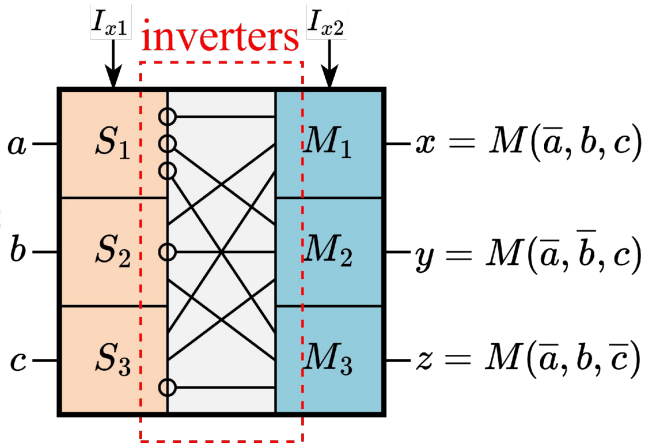
- For a normal RQFP gate, $R(a,b,c) = \{M(\bar{a},b,c), M(a,\bar{b},c), M(a,b,\bar{c})\}$.
- Since inverters can be freely integrated into any input of each AQFP majority, the functionality of RQFP logic gates can be extended for better RQFP circuit designs.
  - each output of an RQFP logic gate can have **eight function choices**: $M(a,b,c)$, $M(a,b,\bar{c})$, $M(a,\bar{b},c)$, $M(a,\bar{b},\bar{c})$, $M(\bar{a},b,c)$, $M(\bar{a},b,\bar{c})$, $M(\bar{a},\bar{b},c)$, and $M(\bar{a},\bar{b},\bar{c})$.



Normal RQFP gate



BFR INV



Input			Output		
a	b	c	x	y	z
0	0	0	0	1	1
0	0	1	1	1	0
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	0	0	1
1	1	1	1	0	0

$R(a,b,c) = \{M(\bar{a},b,c), M(\bar{a},\bar{b},c), M(\bar{a},b,\bar{c})\}$

[ICCAD'23] Rongliang Fu, Olivia Chen, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Exact Logic Synthesis for Reversible Quantum-Flux-Parametron Logic," ICCAD, pp. 1-9, 2023.

Characteristics of RQFP Logic − Functionality

However, **not all inverter configurations** for an RQFP logic gate can make the gate **logically reversible**.

- For example, for $R(a,b,c) = \{M(\mathrm{a},b,c), M(a,\mathrm{b},c), M(a,b,c)\}$, $\boldsymbol{R(1,1,1) = R(1,1,0) = \{1,1,1\}}$

Logical reversibility requires a **one-to-one mapping** between vectors of inputs and outputs.

- Any two outputs in the truth table of an RQFP logic gate with inverter configuration ic must be different.
- Specifically, for any two distinct data input vectors d_1 and d_2 with respective indices $x_1 \in [0,2^3)$ and $x_2 \in [0,2^3)$, $x_1 \neq x_2$, in the truth table.

$$\bigvee_{i\in[0,3)} \mathrm{M}_{j\in[0,3)}(ic[i][j] \oplus d_1[j]) \oplus \mathrm{M}_{j\in[0,3)}(ic[i][j] \oplus d_2[j])$$

where $d_1 = \{x_1 \wedge 1, (x_1 \gg 1) \wedge 1, (x_1 \gg 2) \wedge 1\}$ and $d_2 = \{x_2 \wedge 1, (x_2 \gg 1) \wedge 1, (x_2 \gg 2) \wedge 1\}$.

Overall, there are a total of $2^9 = 512$ possible functions, of which **192** are logically reversible.

Now, how can we design RQFP logic circuits using these reversible gates?

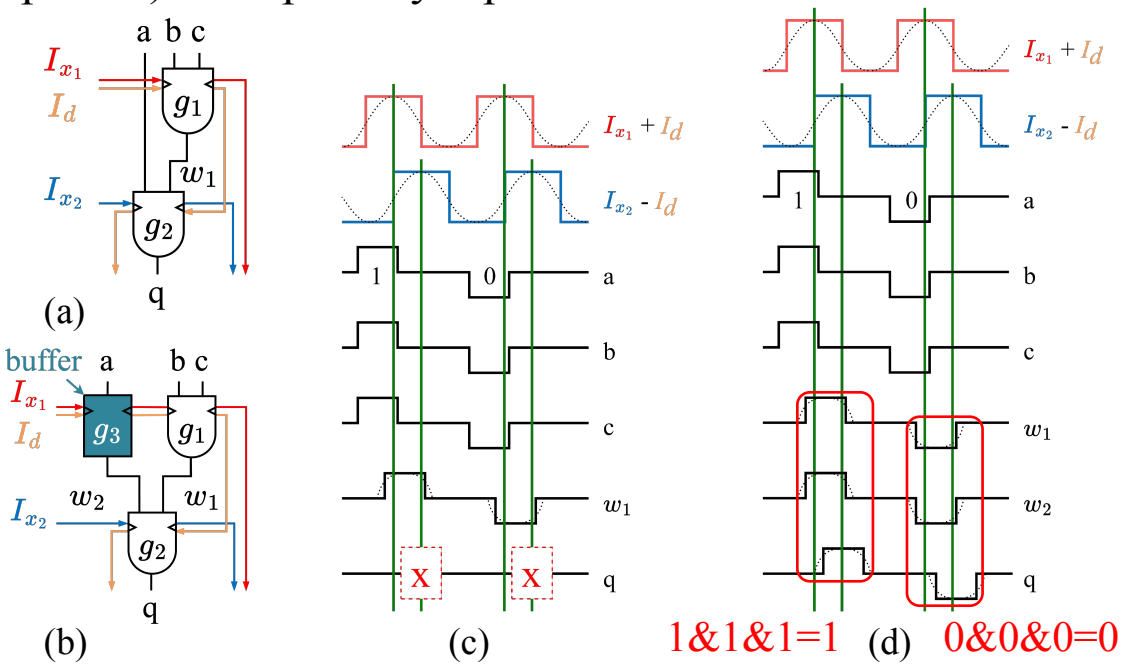
[TCAD25] Rongliang Fu, Robert Wille, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Efficient Cartesian Genetic Programming-based Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits," TCAD, 2025.

Refer to AQFP Logic Circuit Design

- Since RQFP is a derivative technology of AQFP, we first introduce the characteristics of AQFP logic.

1. Clock-synchronized data propagation

All inputs to each gate have the same delay (clock phases) from primary inputs.

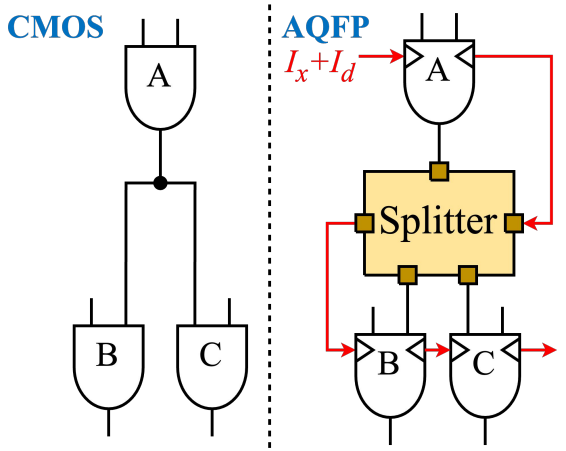


An example shows the necessity of **buffer insertion** for correct operation in an AQFP circuit with the function $q = a\&b\&c$.

2. Single fan-out limitation

Multi-fan-out implementation via the special cell named **splitter**.

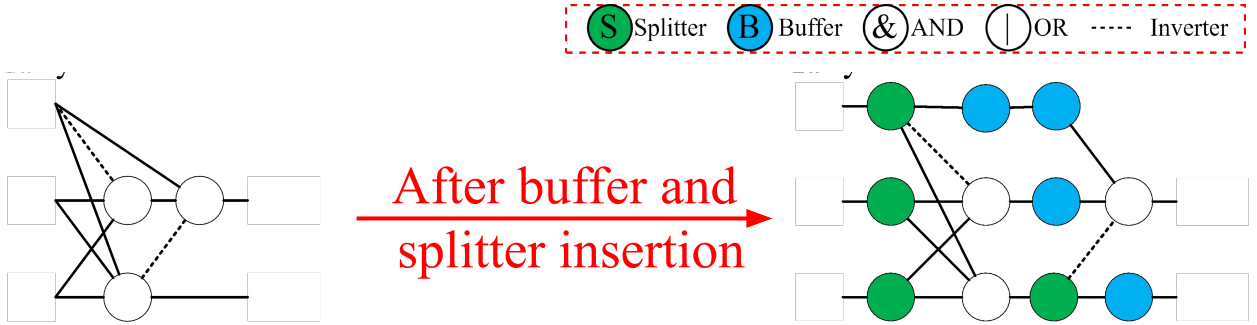
- The splitter requires a clock signal.



AQFP Logic Circuit Design − Technology Legalization

- For a 1-bit full adder,
  - AND-OR-Inverter (AOI) : $\begin{cases} \text{carry} = a \cdot b + (a + b) \cdot c \\ \text{sum} = (a + b + c) \cdot \overline{\text{carry}} + a \cdot b \cdot c \end{cases}$
  - **Majority-Inverter graph (MIG)[27]:** $\begin{cases} \text{carry} = \text{M}(a, b, c) \\ \text{sum} = \text{M}(a, \text{M}(\bar{a}, b, c), \overline{\text{carry}}) \end{cases}$

- Technology legalization
  - Meet clock synchronization and fan-out limitation requirements.

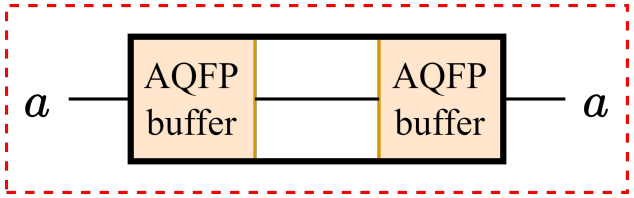


So, RQFP logic circuits also require buffers and splitters for technology legalization.

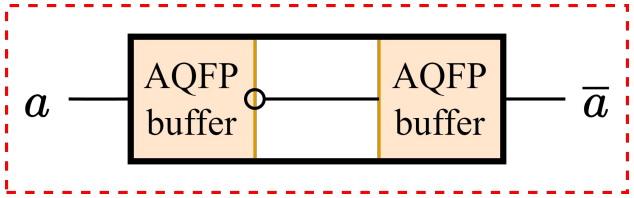
[27] Amaru *et al*., "Majority-Inverter Graph: A New Paradigm for Logic Optimization," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 5, pp. 806–819, 2016.

Characteristics of RQFP Logic − Buffer and Splitter

- For clock-synchronized data propagation,
  - insert RQFP buffers formed by two **cascaded AQFP buffers**.

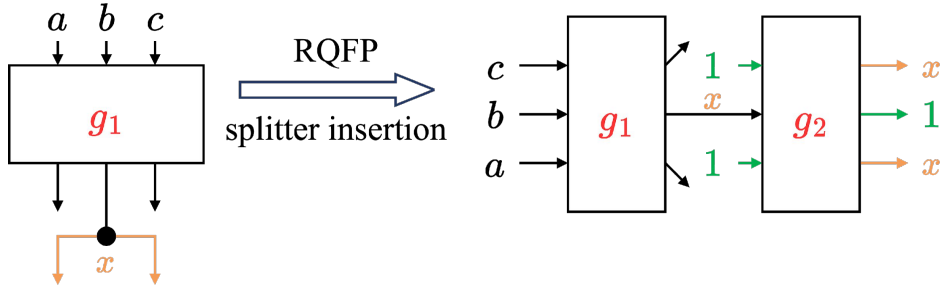


RQFP buffer



RQFP inverter

- For multiple fan-outs,
  - the RQFP splitter can be realized by the **combination of an RQFP gate and constants**.
  - i.e, the **1-to-2** splitter $R(1, x, 1) = \{M(\bar{1}, x, 1), M(1, \bar{x}, 1), M(1, x, \bar{1})\} = \{x, 1, x\}$.



How about the 1-to-3 splitter?

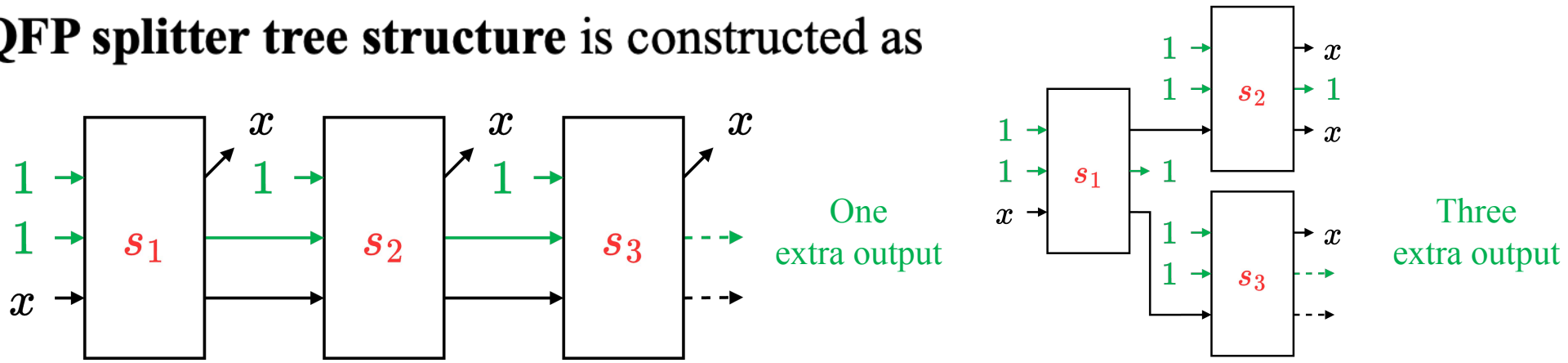
[ICCAD'23] Rongliang Fu, Olivia Chen, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Exact Logic Synthesis for Reversible Quantum-Flux-Parametron Logic," ICCAD, pp. 1-9, 2023.
[DAC'24] Rongliang Fu, Robert Wille and Tsung-Yi Ho, "RCGP: An Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits based on Efficient Cartesian Genetic Programming," DAC, pp. 1-6, 2024.
[TCAD25] Rongliang Fu, Robert Wille, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Efficient Cartesian Genetic Programming-based Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits," TCAD, 2025.

Characteristics of RQFP Logic − Buffer and Splitter

However, we found that no matter what inverter configuration an RQFP logic gate has, it cannot function as a 1-to-3 RQFP splitter with logical reversibility.

- **Only 1-to-2 RQFP splitters** can be used to implement multiple fan-outs, where each splitter introduces one additional garbage output.

To reduce the number of extra outputs caused by the insertion of RQFP splitters, a **chained RQFP splitter tree structure** is constructed as

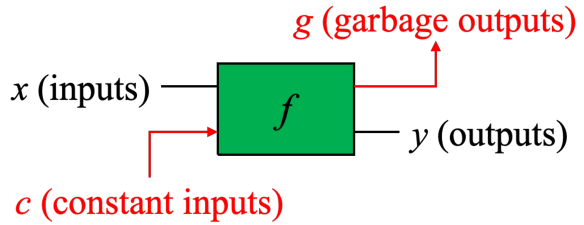


s_1, s_2, and s_3 have the same inverter configuration $R(1,1,x) = \{M(\bar{1},1,x), M(1,1,\bar{x}), M(1,\bar{1},x)\}$.

[TCAD25] Rongliang Fu, Robert Wille, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Efficient Cartesian Genetic Programming-based Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits," TCAD, 2025.

RQFP Logic Circuit Design

- To implement RQFP logic circuit design, we need to consider
  - functional equivalence by RQFP gates, (number of RQFPs, buffers, garbage outputs)
  - fan-out limitation by RQFP splitters, (number of RQFPs, garbage outputs)
  - clock synchronization by RQFP buffers. (number of buffers)



RQFP logic-based realization of the function f with inputs $\boldsymbol{x}$ and outputs $\boldsymbol{y}$.

- **RQFP gates and garbage outputs** significantly impact the energy dissipation of RQFP logic circuits.
- Observe that **the first two steps** impact the number of RQFP logic gates and garbage outputs.
- Therefore, we need to combine the first two steps to achieve a better design of the RQFP logic circuit.

[ICCAD'23] Rongliang Fu, Olivia Chen, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Exact Logic Synthesis for Reversible Quantum-Flux-Parametron Logic," ICCAD, pp. 1-9, 2023.
[DAC'24] Rongliang Fu, Robert Wille and Tsung-Yi Ho, "RCGP: An Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits based on Efficient Cartesian Genetic Programming," DAC, pp. 1-6, 2024.
[TCAD25] Rongliang Fu, Robert Wille, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Efficient Cartesian Genetic Programming-based Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits," TCAD, 2025.

Automatic Synthesis of RQFP Circuits

1) Exact logic synthesis (ICCAD’23)

2) Efficient Cartesian Genetic Programming (DAC’24, TCAD24)

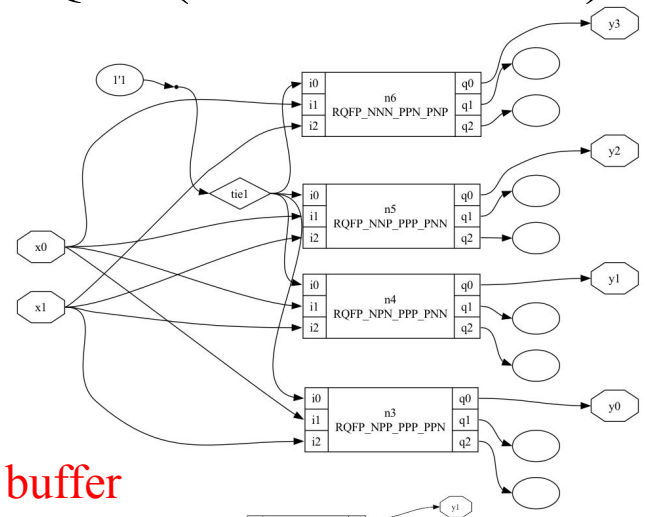
[ICCAD’23] Rongliang Fu, Olivia Chen, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Exact Logic Synthesis for Reversible Quantum-Flux-Parametron Logic," ICCAD, pp. 1-9, 2023.
[DAC’24] Rongliang Fu, Robert Wille and Tsung-Yi Ho, "RCGP: An Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits based on Efficient Cartesian Genetic Programming," DAC, pp. 1-6, 2024.
[TCAD25] Rongliang Fu, Robert Wille, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Efficient Cartesian Genetic Programming-based Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits,” TCAD, 2025.

Simple Logic Synthesis for RQFP Logic

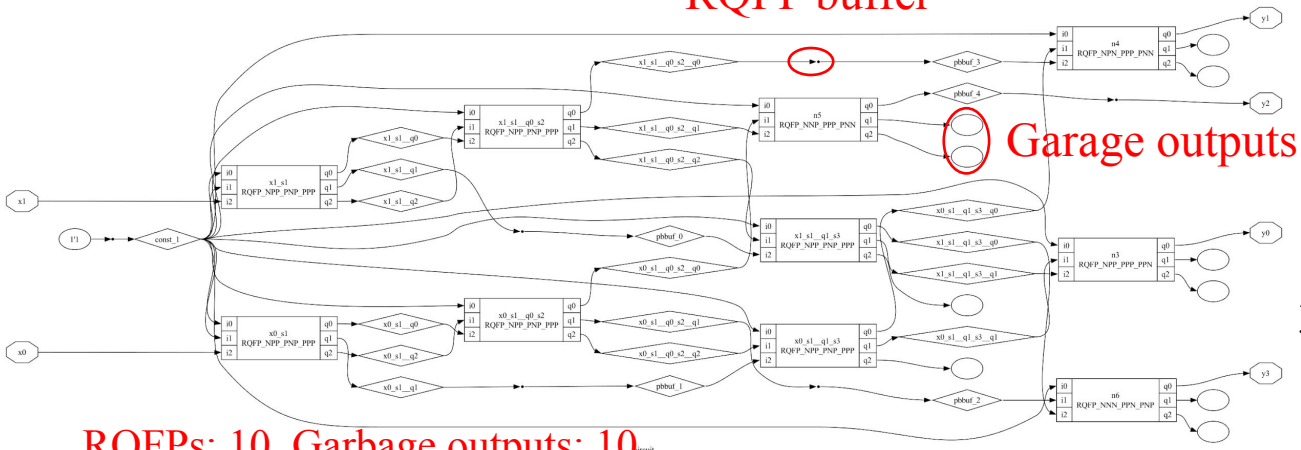
1) AOI/MIG-based logic synthesis
2) Convert AOI/MIG into RQFP (Transformation)
3) RQFP splitter insertion
4) RQFP buffer insertion

```
module decoder_2_4( x0 , x1 , y0 , y1 , y2 , y3 );
  input x0 , x1 ;
  output y0 , y1 , y2 , y3 ;
  wire n3 , n4 , n5 , n6 ;
  assign n3 = x0 & x1 ;
  assign n4 = x0 & ~x1 ;
  assign n5 = ~x0 & x1 ;
  assign n6 = x0 | x1 ;
  assign y0 = n3 ;
  assign y1 = n4 ;
  assign y2 = n5 ;
  assign y3 = ~n6 ;
endmodule
```

1) and 2)



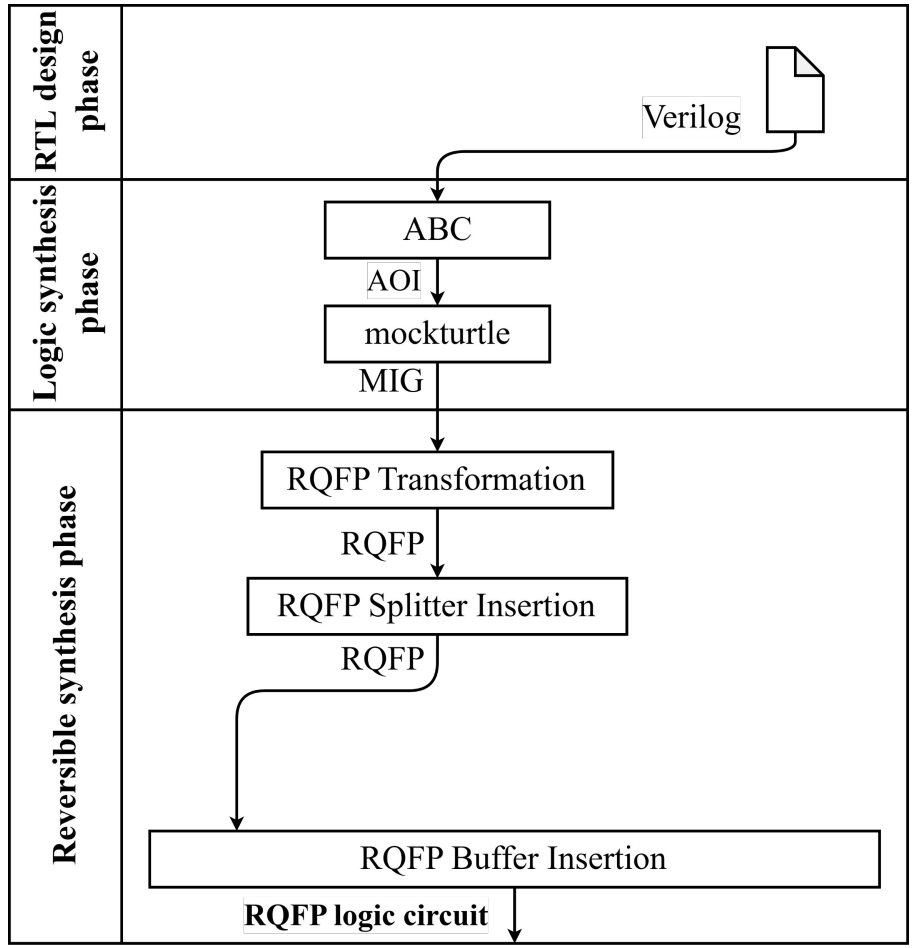
RQFP buffer



3) and 4)

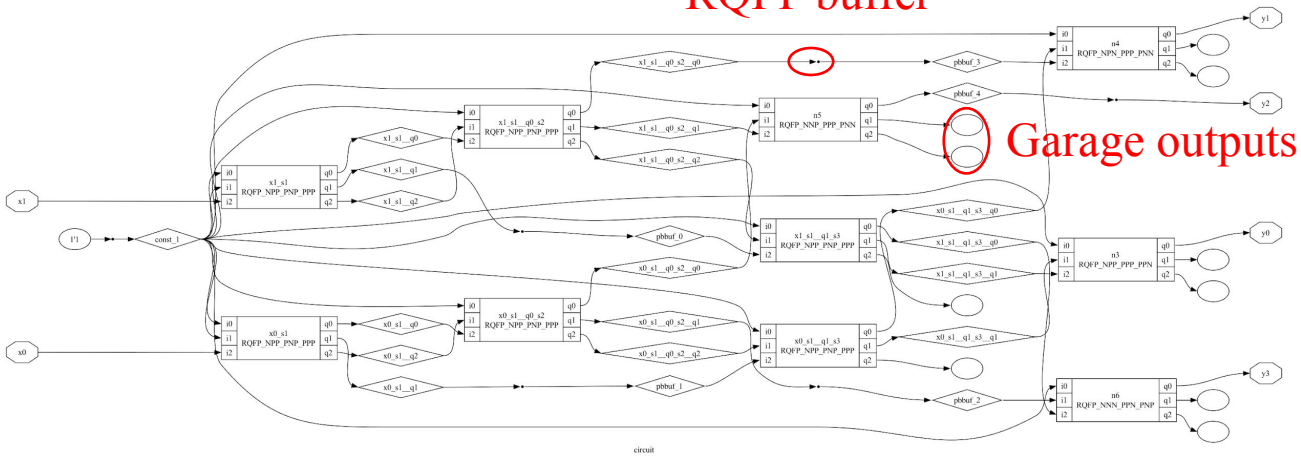
RQFPs: 10, Garbage outputs: 10

'P': Positive (0), 'N': Negative (1). For example, $M(\bar{a}, b, c)$ => NPP



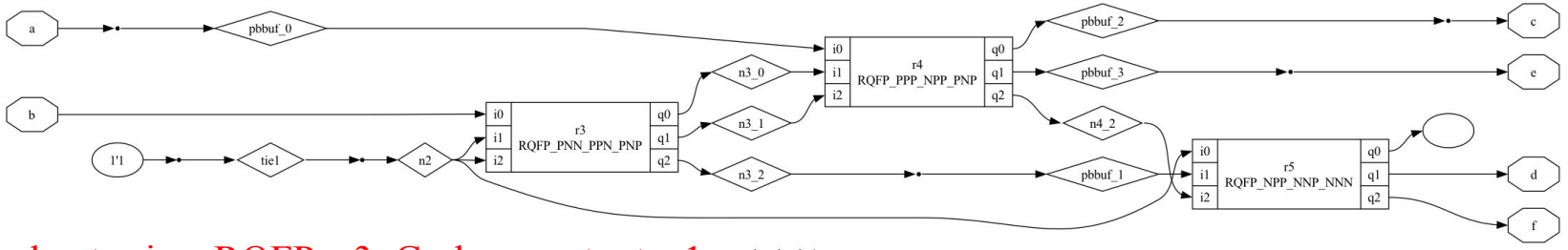
Exact Logic Synthesis for RQFP Logic (ICCAD'23)

RQFP buffer

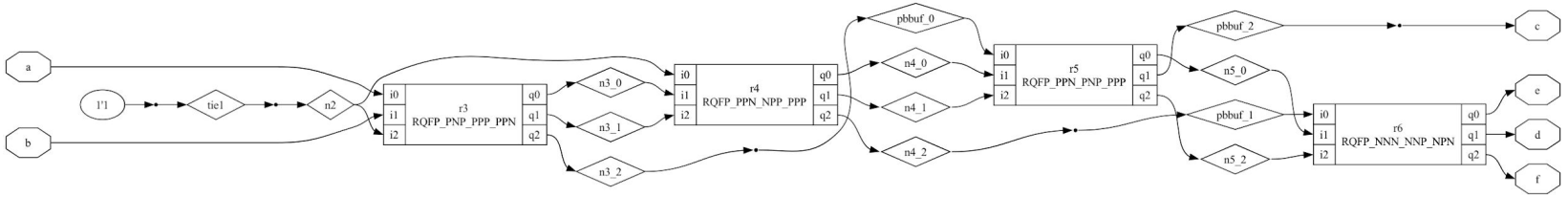


Garage outputs

RQFPs: 10, Garbage outputs: 10



Optimal gate size: RQFPs: 3, Garbage outputs: 1



Optimal garbage outputs: RQFPs: 4, Garbage outputs: 0

'P': Positive (0), 'N': Negative (1). For example, $M(\bar{a}, b, c)$ => NPP

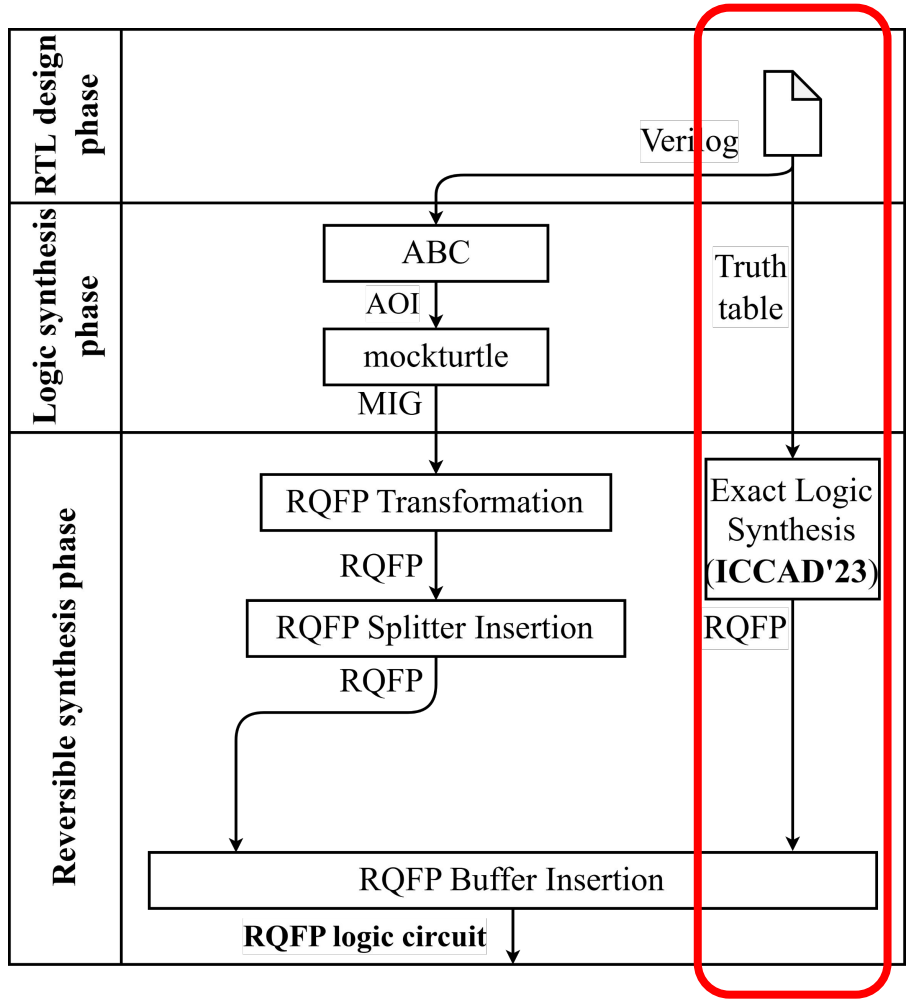
Exact Logic Synthesis for RQFP Logic (ICCAD'23)

Exact logic synthesis is to explore the detailed realization of f_i, i.e.,

$$\forall\{x_1, \dots, x_n\}, f_i(x_1, \dots, x_n) \equiv o_i(x_1, \dots, x_n) \equiv circuit_i(x_1, \dots, x_n),$$

where $circuit_i$ is defined by solving the constructed **SAT model** to choose gate types and connections **under given conditions**, such as r gates and g garbage outputs.

Hence, the key to exact logic synthesis becomes how to effectively and efficiently formulate the **types** of gates and the **connections** between gates, primary inputs (PIs), and primary outputs (POs).

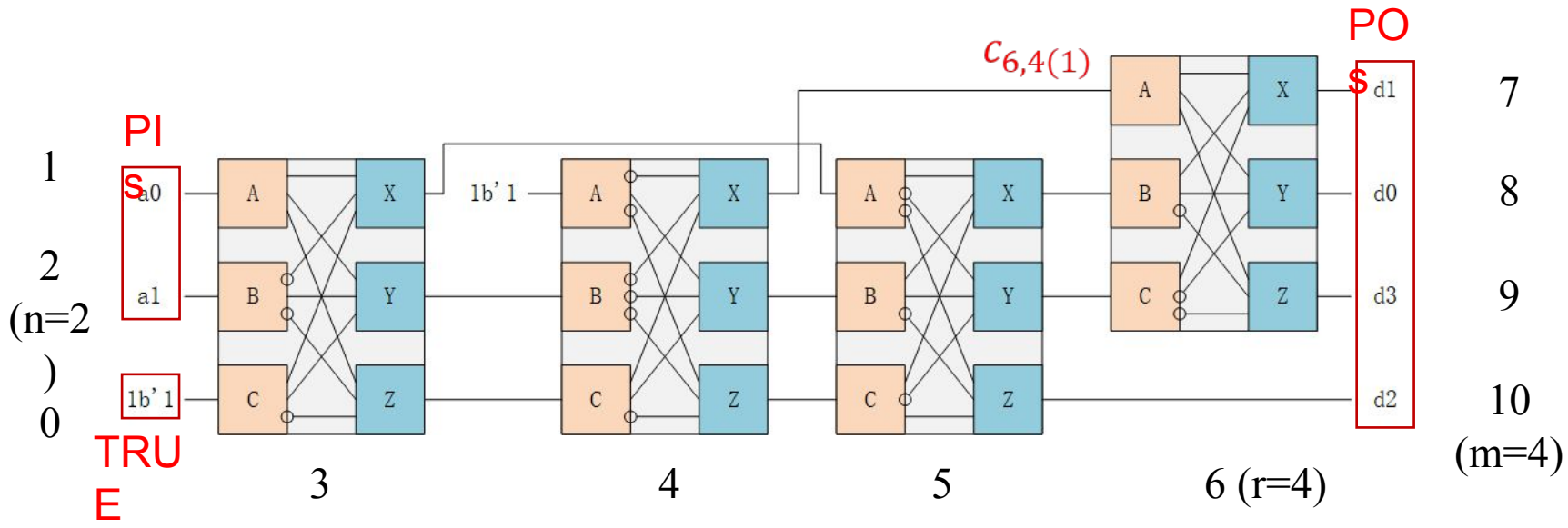


[ICCAD'23] Rongliang Fu, Olivia Chen, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Exact Logic Synthesis for Reversible Quantum-Flux-Parametron Logic," ICCAD, pp. 1-9, 2023.

SAT Encoding for RQFP Logic

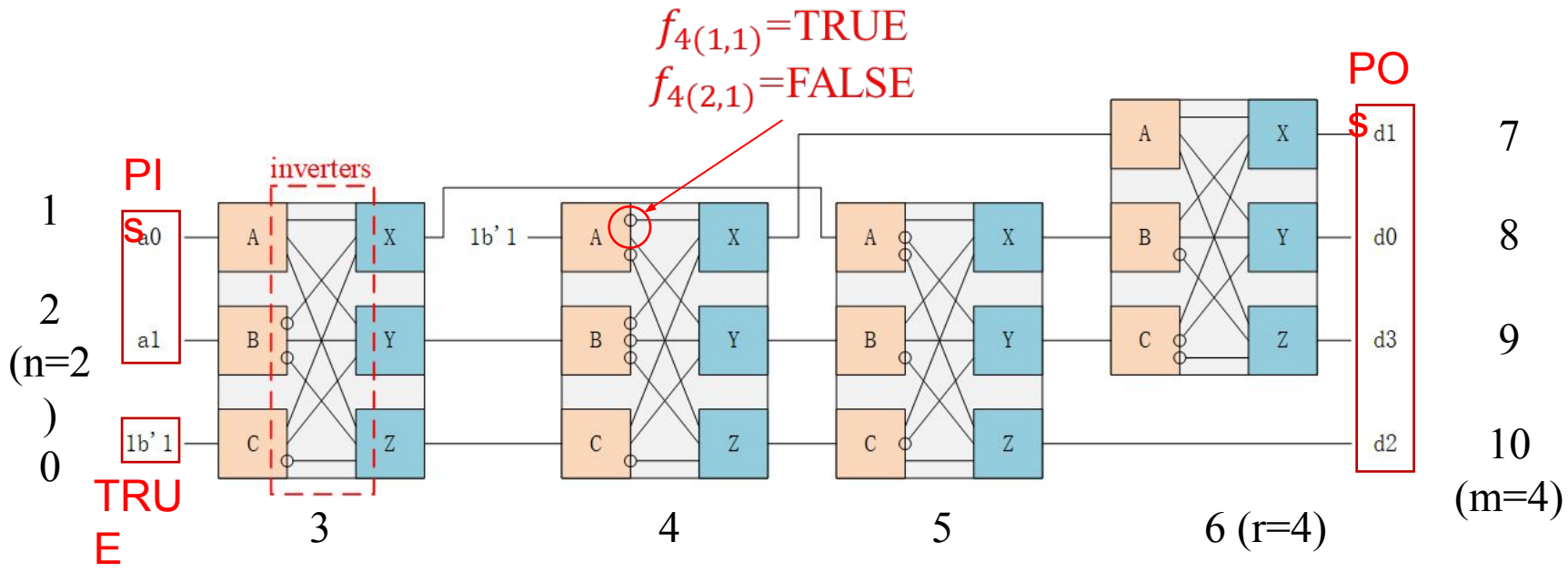
To realize the RQFP logic circuit with m functions $f_i(x_1, \ldots, x_n), i \in [1, m]$ and r RQFP logic gates,

- Encoding for Variables
  - POs: $o_{i[t]}, i \in [1, m], t \in [1,2^n]$
  - Constant TRUE: $x_{0(0)[t]}, t \in [1,2^n]$. PIs: $x_{i(0)[t]}, i \in [1, n], t \in [1,2^n]$
  - Gates' outputs: $x_{i(p)[t]}, i \in [n+1, n+r], p \in [1,3], t \in [1,2^n]$
  - Connections: $c_{i,j(p)}, i \in [n+1, n+r+m], j \in [0, i), p \in [1,3]$



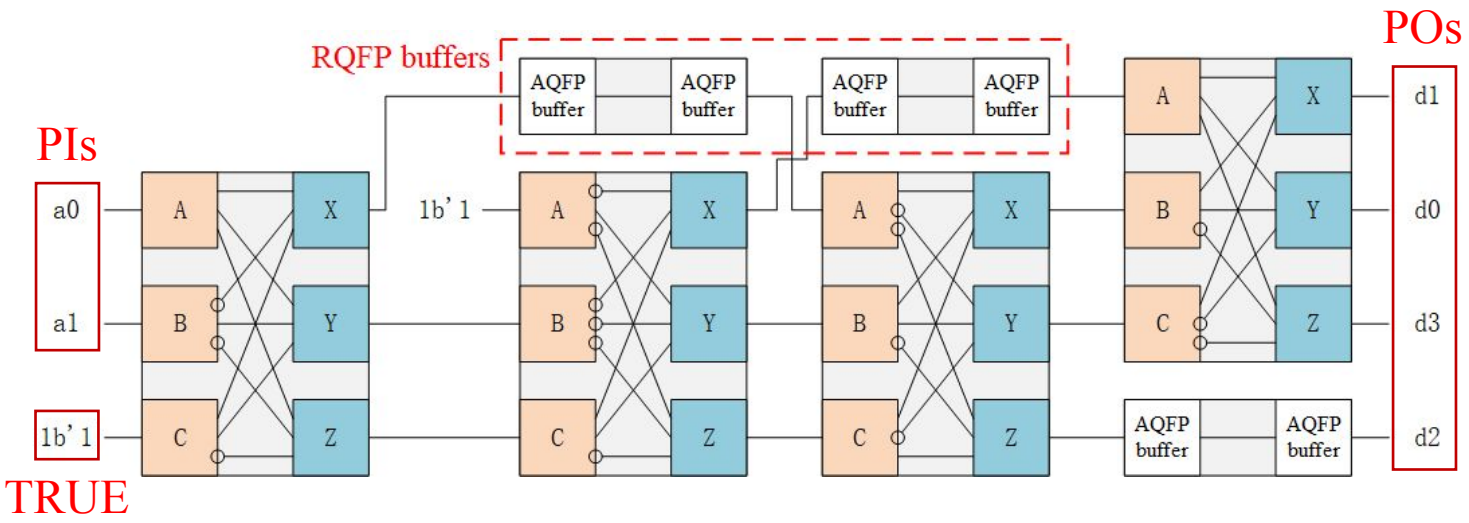
SAT Encoding for RQFP Logic

- Encoding for Gate Functions
  - $f_{i(p,j)}, p \in [1,3], j \in [1,3]$ indicates whether an inverter does not exist in front of the j^{th} input of the p^{th} majority in gate x_i.



Exact Logic Synthesis for RQFP Logic (ICCAD'23)

- Iteratively and incrementally constructs the SAT model.
  - First, increase gates to find a feasible solution.
    - Initial gate number $r = \left\lceil \frac{\max(n,m)}{3} \right\rceil$
  - Then, limit the number of garbage outputs.
    - Lower boundary $g \geq \max(0, n-m)$
  - Finally, insert RQFP buffers.

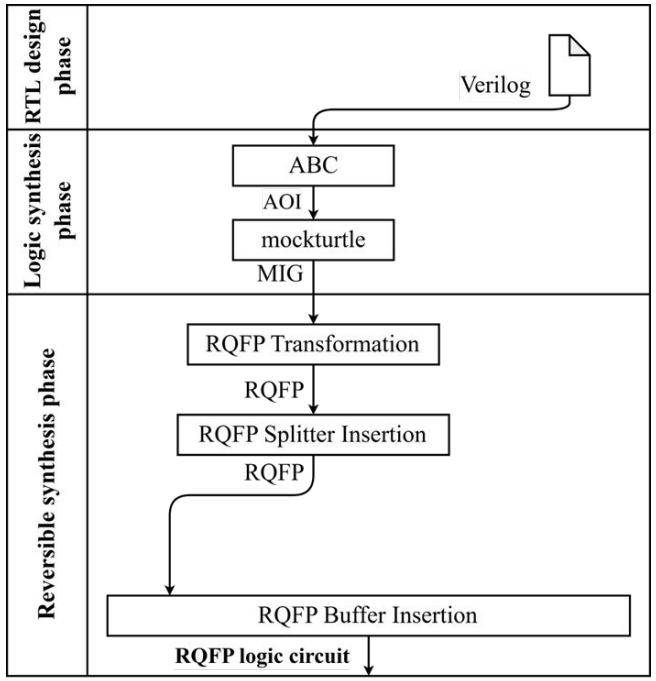


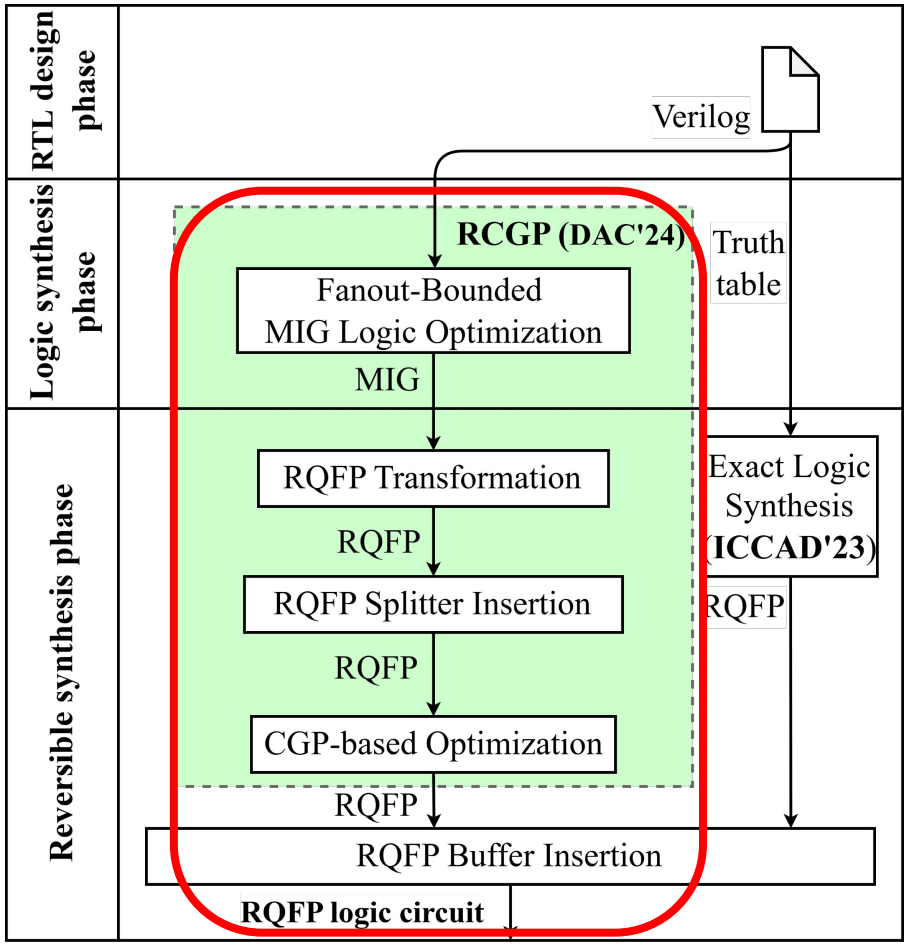

```
3  g = −1
4  sol ← Solver()
5  create variables for PIs, POs, constant true, r gates, and connections.
6  create clauses for function constraints.
7  sol.push() // create a snapshot
8  while true do
9      if g < 0 then
10         create clauses for fan-out constraints and POs' fan-in constraints.
11         sol.push() // create a snapshot
12     else
13         create clauses for garbage output constraints.
14     if sol.check() == SAT then
15         create RQFP logic circuit cir by sol.model().
16         calculate the garbage number g_c of cir.
17         if g_c ≤ g_lb then
18             break
19         else
20             g = g_c − 1
21             sol.pop() // remove all clauses added after the last snapshot
22     else
23         if g ≥ 0 ∧ g ≤ g_lb then
24             break
25         else if g > 0 then
26             g −= 1
27             sol.pop() // remove all clauses added after the last snapshot
28         else
29             sol.pop(2) // remove all clauses added after the penultimate snapshot
30             r += 1 // new a RQFP logic gate
31             create variables for new gate x_{n+r}.
32             create clauses for constraints of x_{n+r}.
33             sol.push() // create a snapshot
34 insert RQFP buffers for cir
```

Cartesian Genetic Programming-based Synthesis for RQFP Logic (DAC'24, TCAD25)

Since **exact logic synthesis is time consuming**,

we propose **RCGP**, an efficient Cartesian Genetic Programming (CGP) [31]-based flow, to generate large RQFP logic circuits.



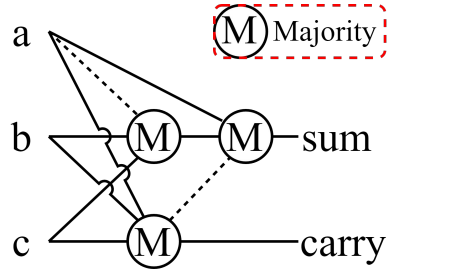


[31] Miller, J.F. "Cartesian genetic programming: its status and future," Genet Program Evolvable, pp. 129–168, 2020.
[32] G. Meuli *et al*., "Majority-based Design Flow for AQFP Superconducting Family," DATE, 34–39, 2022.
[DAC'24] Rongliang Fu, Robert Wille and Tsung-Yi Ho, "RCGP: An Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits based on Efficient Cartesian Genetic Programming," DAC, pp. 1-6, 2024.
[TCAD25] Rongliang Fu, Robert Wille, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Efficient Cartesian Genetic Programming-based Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits," TCAD, 2025.

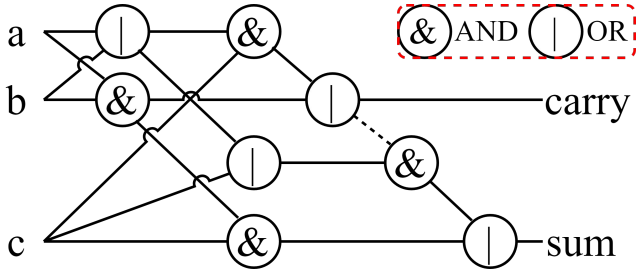
Why use CGP?

- Its encoding features on directed graphs can present the topology of RQFP circuits.
  - RQFPs have three inputs and three distinct outputs
  - Current circuit representation structures, such as AOI and MIG, cannot be applied in RQFP circuit design.

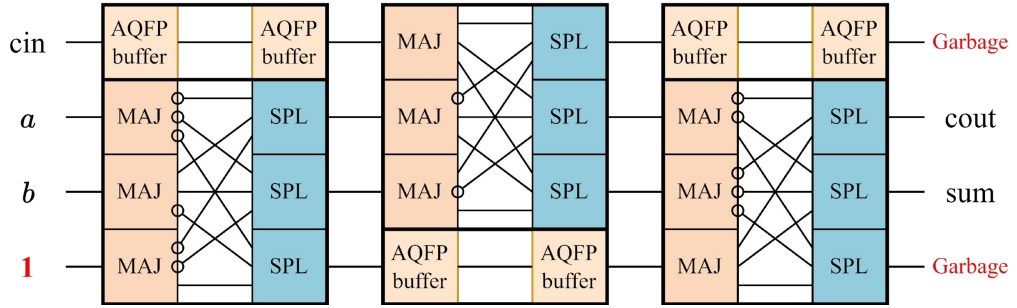
- Its evolutionary process can handle the fan-out limitation problem.
  - The introduction of RQFP splitters for the fan-out limitation in an RQFP circuit causes many garbage outputs.
  - Current logic optimization methods, such as rewriting and resubstitution, cannot handle the fan-out limitation well.



MIG (three inputs and one output)



AOI (two inputs and one output)

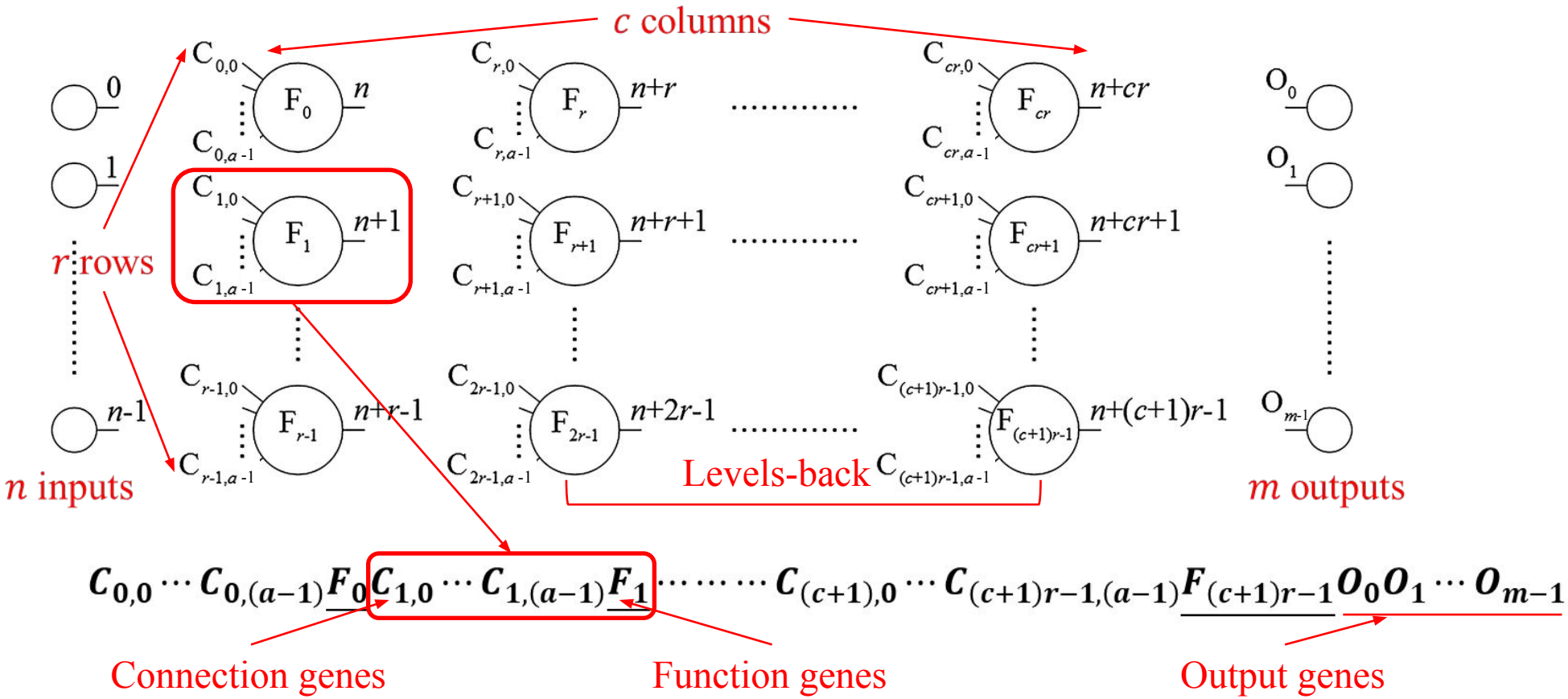


RQFP (three inputs and three outputs)

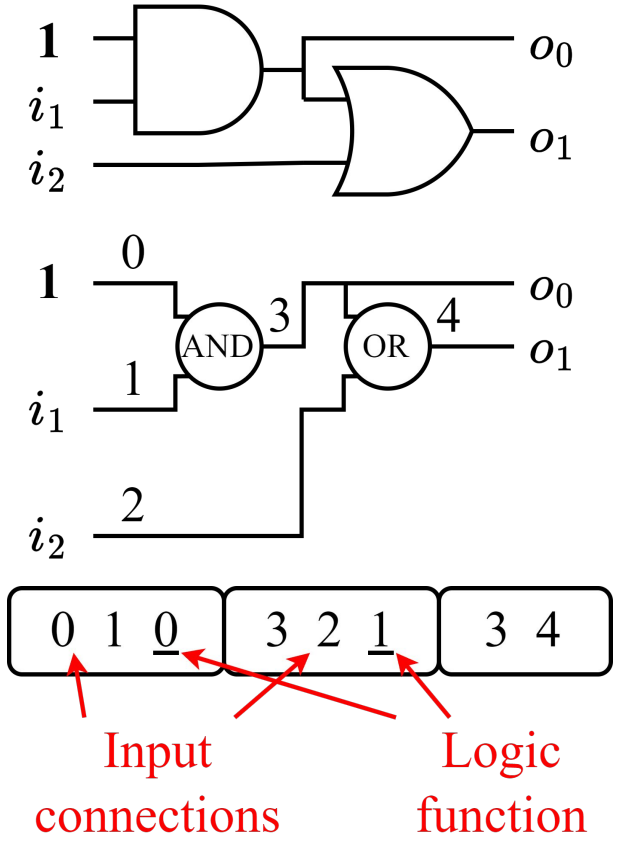
1-bit full adder

Efficient Cartesian Genetic Programming (CGP)

- CGP is an increasingly popular and efficient form of Genetic Programming.
  - Suitable for a program in the form of a directed graph.
  - The genotype in CGP is represented as a string of fixed-length integers, as follows:



Efficient Cartesian Genetic Programming (CGP)



Circuit

$n = 3, m = 2$

Phenotype

$r = 1, c = 2, a = 2, \mathrm{F} = \{0: \mathrm{AND}, 1: \mathrm{OR}\}$

Genotype

$\big((a+1) * r * c + m\big)$ integers

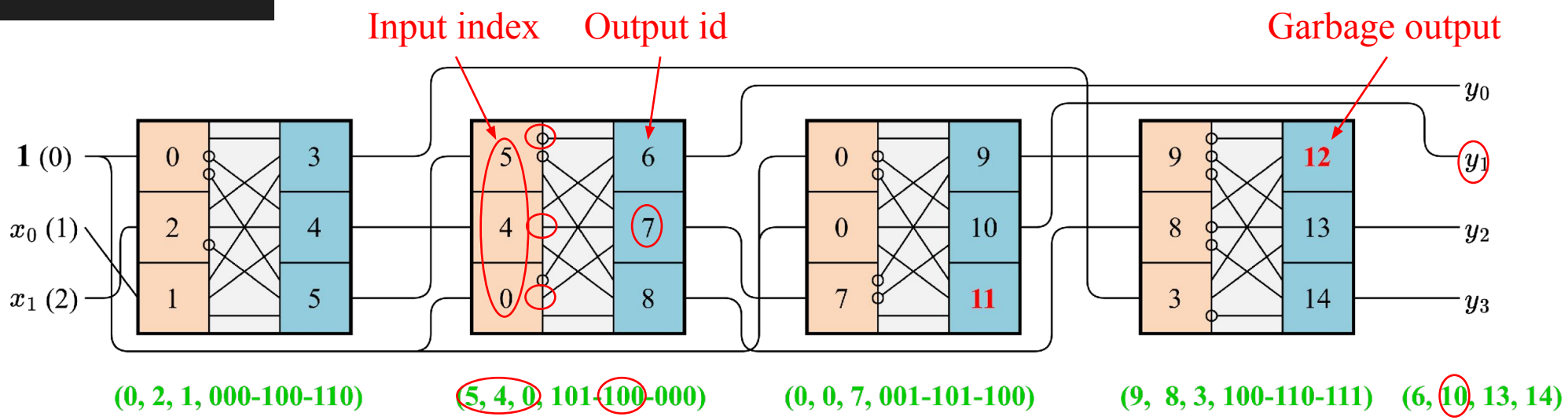
This CGP individual presents a circuit with the function $\begin{cases} o_0 = i_1, \\ o_1 = i_1 + i_2 \end{cases}$, where the identifiers **1**, i_1, and i_2 refer to the three primary inputs with indices 0, 1, and 2, and the identifier o_0 and o_1 stand for the primary outputs of the circuit.

RCGP

- Initialization

```
module decoder_2_4( x0 , x1 , y0 , y1 , y2 , y3 );
  input x0 , x1 ;
  output y0 , y1 , y2 , y3 ;
  wire n3 , n4 , n5 , n6 ;
  assign n3 = x0 & x1 ;
  assign n4 = x0 & ~x1 ;
  assign n5 = ~x0 & x1 ;
  assign n6 = x0 | x1 ;
  assign y0 = n3 ;
  assign y1 = n4 ;
  assign y2 = n5 ;
  assign y3 = ~n6 ;
endmodule
```

Initialization +
RQFP Splitter Insertion

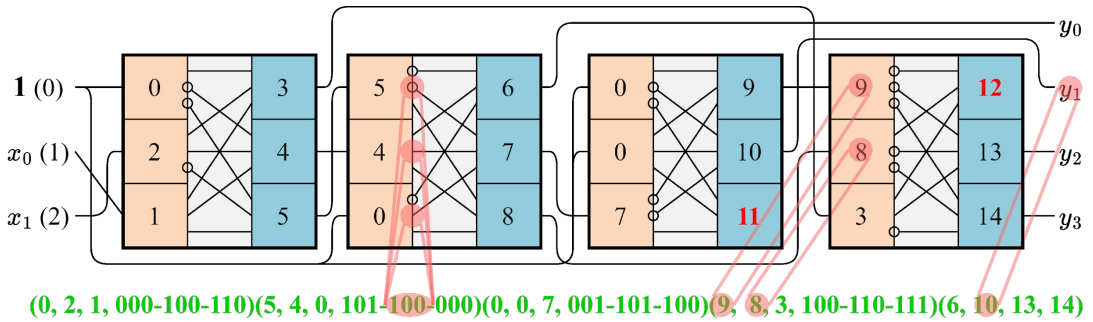


RCGP

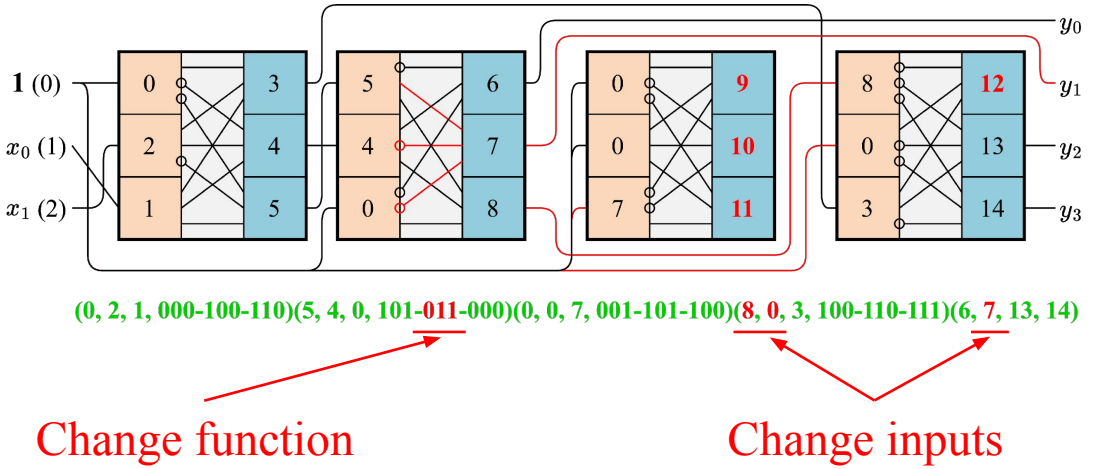
- Mutation

```
module decoder_2_4( x0 , x1 , y0 , y1 , y2 , y3 );
  input x0 , x1 ;
  output y0 , y1 , y2 , y3 ;
  wire n3 , n4 , n5 , n6 ;
  assign n3 = x0 & x1 ;
  assign n4 = x0 & ~x1 ;
  assign n5 = ~x0 & x1 ;
  assign n6 = x0 | x1 ;
  assign y0 = n3 ;
  assign y1 = n4 ;
  assign y2 = n5 ;
  assign y3 = ~n6 ;
endmodule
```

Initialization +
RQFP Splitter Insertion



1) A CGP individual

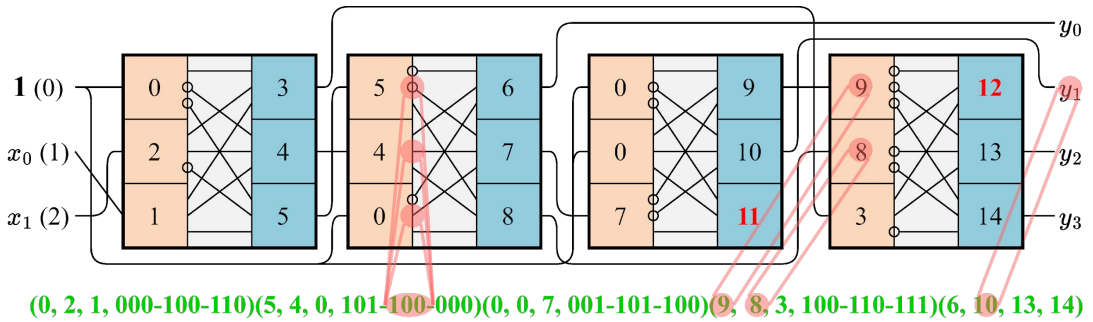


RCGP

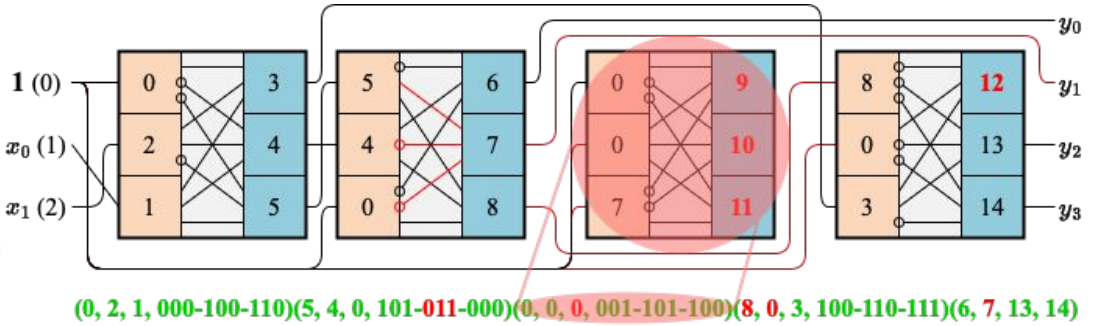
- Gate shrinking

```
module decoder_2_4( x0 , x1 , y0 , y1 , y2 , y3 );
  input x0 , x1 ;
  output y0 , y1 , y2 , y3 ;
  wire n3 , n4 , n5 , n6 ;
  assign n3 = x0 & x1 ;
  assign n4 = x0 & ~x1 ;
  assign n5 = ~x0 & x1 ;
  assign n6 = x0 | x1 ;
  assign y0 = n3 ;
  assign y1 = n4 ;
  assign y2 = n5 ;
  assign y3 = ~n6 ;
endmodule
```

Initialization + RQFP Splitter Insertion

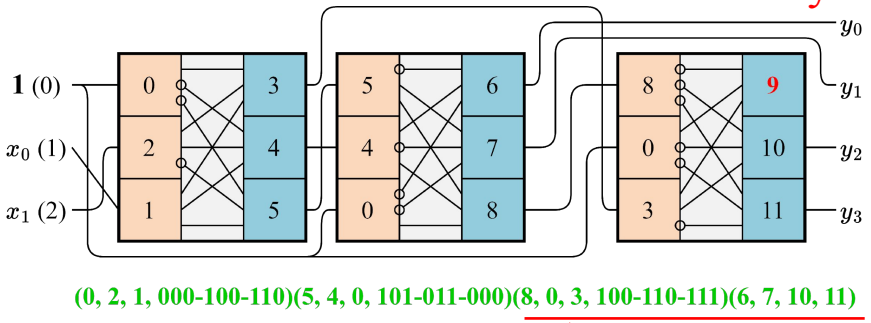


1) A CGP individual

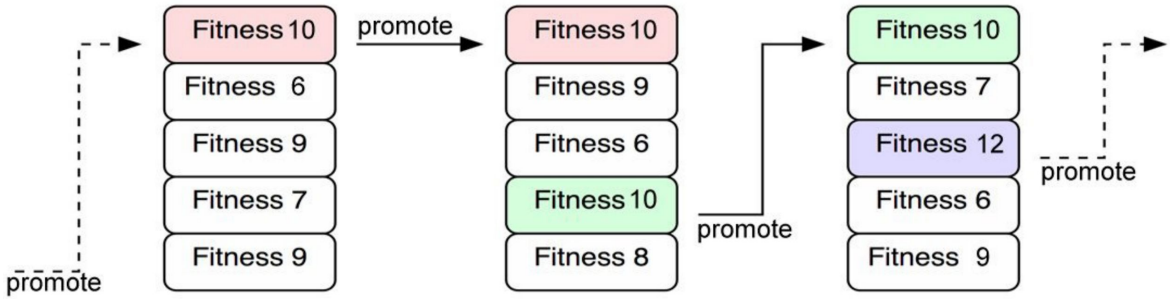


2) Mutation

No any fan-out, remove it



Update CGP encoding



$(1 + \lambda)$ Evolutionary Strategy:

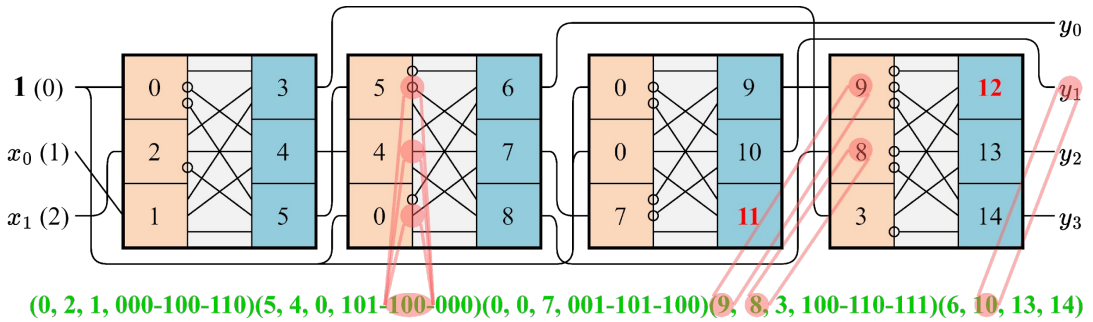
- An offspring is always chosen if it is equally as fit or has better fitness than the parent

RCGP

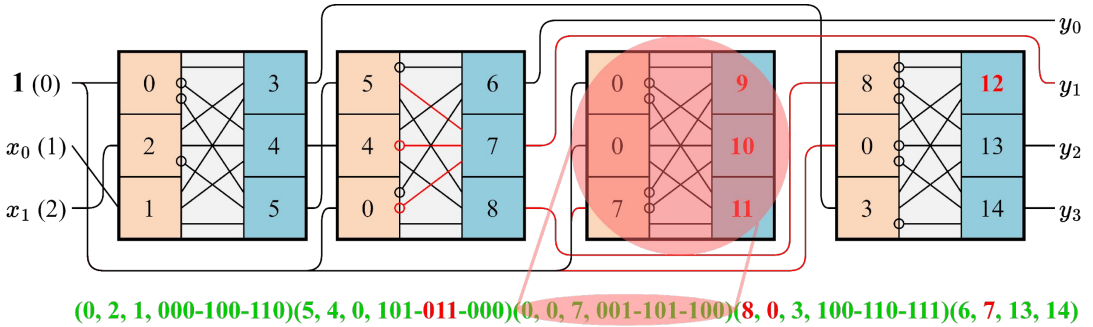
- RQFP buffer insertion

```
module decoder_2_4( x0 , x1 , y0 , y1 , y2 , y3 );
  input x0 , x1 ;
  output y0 , y1 , y2 , y3 ;
  wire n3 , n4 , n5 , n6 ;
  assign n3 = x0 & x1 ;
  assign n4 = x0 & ~x1 ;
  assign n5 = ~x0 & x1 ;
  assign n6 = x0 | x1 ;
  assign y0 = n3 ;
  assign y1 = n4 ;
  assign y2 = n5 ;
  assign y3 = ~n6 ;
endmodule
```

Initialization + RQFP Splitter Insertion

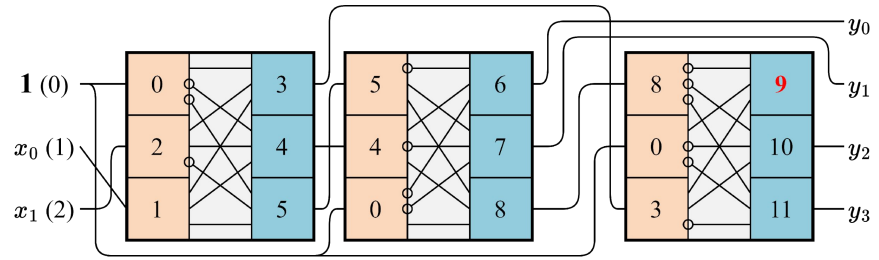


1) A CGP individual

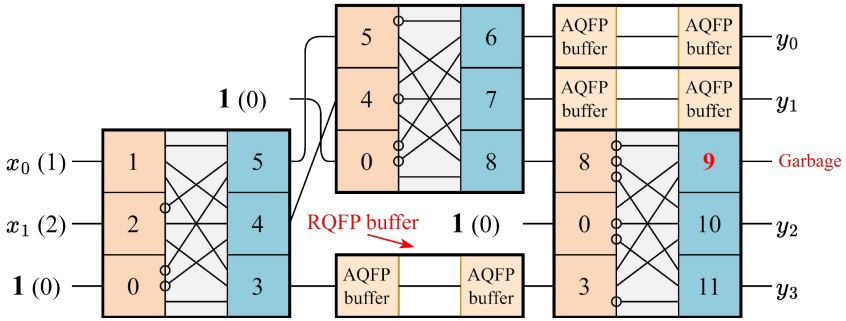


2) Mutation

Up to threshold



3) Gate shrinking



4) RQFP buffer insertion

RCGP

- $(1 + \lambda)$ - CGP search strategy

① Initialize the parent.

② Select the best from the λ offspring generated in each generation as the next parent.

Input: Initial individual p, total number N of generations, mutation rate μ, number λ of offspring.
Output: Optimized individual.

```
1  Calculate the functional fitness f of c.
2  f_n ← f, p_n ← p, n_r ← ∞, n_g ← ∞, n_b ← ∞.
3  if f = n_po × 2^{n_pi} then
4  |  Remove useless nodes to shrink p_n.
5  |  Calculate the number (n_r, n_g, n_b) of RQFP logic
   |    gates, garbage outputs, and RQFP buffers in p_n.
6  for i ∈ [1, N] do
7  |  p ← p_n.
8  |  for j ∈ [1, λ] do
9  |  |  Perform the mutation on p to generate an
   |  |    offspring p'.
10 |  |  Calculate the functional fitness f' of p'.
11 |  |  if f' = 1 then
12 |  |  |  Remove useless nodes to shrink p'.
13 |  |  |  Calculate the number (n'_r, n'_g, n'_b) of gates,
   |  |  |    garbage outputs, and buffers in p'.
14 |  |  |  if (n_r, n_g, n_b) > (n'_r, n'_g, n'_b) then
15 |  |  |  |  (f_n, p_n, n_r, n_g, n_b) ←
   |  |  |  |    (f', p', n'_r, n'_g, n'_b).
16 |  |  else if f' > f_n then
17 |  |  |  (f_n, p_n) = (f', p').
18 return p_n.
```

Experimental results

- Device: Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz and 256.0 GB memory.
- Baseline: Exact logic synthesis with Z3[33]. Initialization: MIG-based logic optimization[32].
- Benchmark: RevLib benchmark circuits[34] and reversible reciprocal circuits[35].

Testcase	Original			Simple					Exact logic synthesis (ICCAD'23)						RCGP (DAC'24, TCAD)					
	PI	PO		ROFB	Buffer	JJ	Depth	Garbage	ROFB	Buffer	JJ	Depth	Garbage	runtime	ROFB	Buffer	JJ	Depth	Garbage	runtime
1-bit full adder	3	2	1	8	7	220	5	10	3	3	84	3	3	14.51	3	3	84	3	3	113.04
4gt10	4	1	3	3	3	84	3	6	3	3	84	3	6	18.44	3	3	84	3	6	98.36
alu	5	1	4	14	17	404	6	17	4	6	120	4	5	96.16	4	6	120	4	5	134.47
c17	5	2	3	13	12	360	5	17				\			7	20	248	6	6	221.84
decoder_2_4	2	4	0	10	5	260	4	10	3	5	92	3	1	10.22	3	3	84	3	1	128.46
decoder_3_8	3	8	0	29	33	828	8	31	7	25	268	7	1	93237.87	7	16	232	5	1	264.68
graycode4	4	4	0	18	13	484	5	22				\			6	4	160	4	2	216.86
ham3	3	3	0	23	23	644	7	24	5	3	132	5	1	192.7	5	3	132	5	1	233.13
mux4	6	1	5	13	14	368	6	16				\			6	3	156	4	7	200.55
4_49_7	4	4	0	48	103	1564	11	42				\			24	45	756	9	13	707.07
graycode6_11	6	6	0	30	17	788	5	36				\			11	17	332	6	3	333.25
mod5adder_66	6	6	0	202	1242	9816	34	195				\			101	288	3576	16	61	3267.38
hwb8_64	8	8	0	2129	40254	212112	128	2037				\			2027	10602	91056	39	1818	96009.00
intdiv4	4	4	0	36	35	1044	9	37				\			14	18	408	6	9	685.99
intdiv5	5	5	0	71	140	2264	13	73				\			37	79	1204	10	22	1071.14
intdiv6	6	6	0	154	538	5848	21	152				\			73	184	2488	14	45	2354.22
intdiv7	7	7	0	293	1947	14820	37	282				\			140	511	5404	18	86	4945.98
intdiv8	8	8	0	562	5701	36292	59	554				\			293	1459	12868	25	187	10554.15
intdiv9	9	9	0	1054	13670	79976	78	1024				\			507	1920	19848	24	380	22717.25
intdiv10	10	10	0	1815	32275	172660	115	1784				\			1119	5972	50744	33	960	100127.44

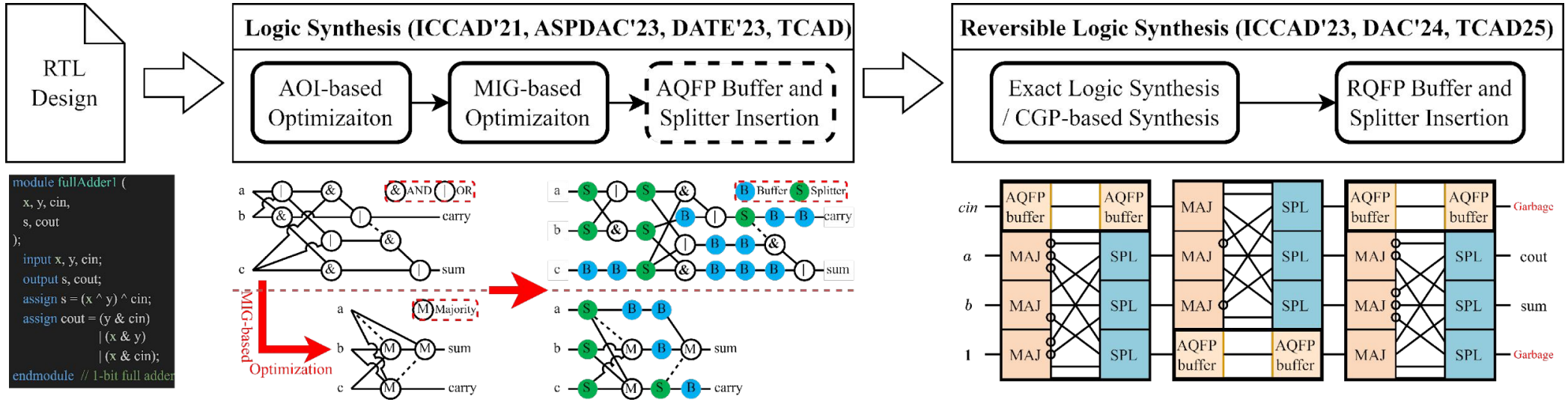
\' represents that the exact logic synthesis method cannot find a feasible solution within 240000 seconds.

[33] L. De Moura and N. Bjørner, "Z3: An efficient SMT solver," in Proceedings of the Theory and Practice of Software, pp. 337–340, 2008.
[34] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler, "RevLib: An online resource for reversible functions and reversible circuits," ISMVL, pp. 220–225, 2008.
[35] Soeken *et al*., "Design automation and design space exploration for quantum computers", 2017.

Conclusion and Future Work

Conclusions

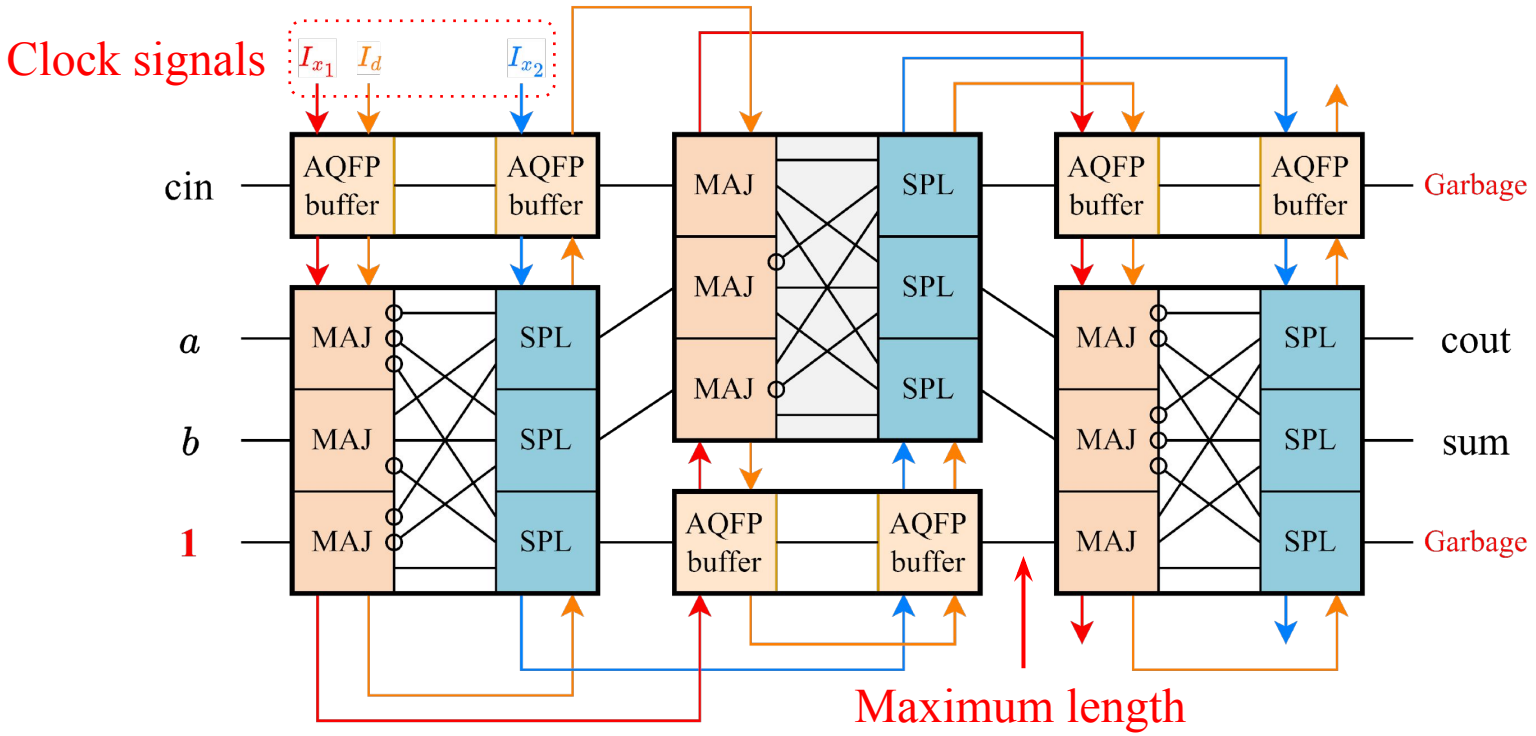
- Introduced energy-efficient superconducting circuits.
- Presented our automation design flow of AQFP circuits.
- Presented our automatic synthesis methods for RQFP circuits.



[ICCAD'21] C. -Y. Huang, Y. -C. Chang, M. -J. Tsai and T. -Y. Ho, "An Optimal Algorithm for Splitter and Buffer Insertion in Adiabatic Quantum-Flux-Parametron Circuits," ICCAD, 2021.
[ASPDAC'23] Rongliang Fu, Mengmeng Wang, Yirong Kan, Nobuyuki Yoshikawa, Tsung-Yi Ho, and Olivia Chen, "A Global Optimization Algorithm for Buffer and Splitter Insertion in Adiabatic Quantum-Flux-Parametron Circuits," ASPDAC, 2023.
[DATE'23] Rongliang Fu, Junying Huang, Mengmeng Wang, Yoshikawa Nobuyuki, Bei Yu, Tsung-Yi Ho, and Olivia Chen, "BOMIG: A Majority Logic Synthesis Framework for AQFP Logic," DATE, 2023.
[TCAD] Rongliang Fu, Mengmeng Wang, Yirong Kan, Olivia Chen, Nobuyuki Yoshikawa, Bei Yu, Tsung-Yi Ho, "Buffer and Splitter Insertion for Adiabatic Quantum-Flux-Parametron Circuits," TCAD, 2024.
[ICCAD'23] Rongliang Fu, Olivia Chen, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Exact Logic Synthesis for Reversible Quantum-Flux-Parametron Logic," ICCAD, pp. 1-9, 2023.
[DAC'24] Rongliang Fu, Robert Wille and Tsung-Yi Ho, "RCGP: An Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits based on Efficient Cartesian Genetic Programming," DAC, pp. 1-6, 2024.
[TCAD25] Rongliang Fu, Robert Wille, Nobuyuki Yoshikawa and Tsung-Yi Ho, "Efficient Cartesian Genetic Programming-based Automatic Synthesis Framework for Reversible Quantum-Flux-Parametron Logic Circuits," TCAD, 2025.

Challenges in RQFP Logic – Physical Design

- Lack of physical design tools specifically for RQFP circuits.
  - Complex clock network.
  - Strict timing and wirelength constraints.



Thank you for your attention!
Q&A