Regularity and Symmetry as a Base for Efficient Realization of Reversible Logic Circuits

Marek Perkowski*, Pawel Kerntopf[1], Alan Coppola[2], Andrzej Buller[3], Malgorzata Chrzanowska-Jeske, Alan Mishchenko, Xiaoyu Song, Anas Al-Rabadi, Lech Jozwiak[4], and Bart Massey

PORTLAND QUANTUM LOGIC GROUP,

Portland State University, Portland, OR 97207

[1]Technical University of Warsaw, Poland, [2] Cypress Semiconductor Corporation,

[3] ATR, Kyoto, Japan, [4]Technical University of Eindhoven,

**mperkows@ece.pdx.edu*

Abstract

A ***Reversible Programmable Gate Array (RPGA)*** is defined in this paper.

It is composed of a collection of ***2*2 Net Structure*** elements arranged in a regular structure, which can represent arbitrary binary functions in reversible logic.

We show that the ***RPGA*** structure enables efficient realization of all symmetric functions and certain non-symmetric functions without variable repetition.

The synthesis technique presented reduces the number of garbage outputs compared to previous methods.

By using standard symmetrization techniques, we show how to map multi-input, multi-output Boolean functions to the ***RPGA***.

Reversible Logic

- As proved by Landauer, using traditional ***irreversible gates*** leads to **energy dissipation**.
- Bennett showed that for power not be dissipated in the circuit it is necessary that arbitrary circuit can be build from ***reversible gates***.

- **Quantum Computing** is a coming revolution
- Quantum computers are *reversible.*
- **Reversible computing** is the step-by-step way of scaling current computer technologies and is the path to future computing technologies, which all happen to use reversible logic.

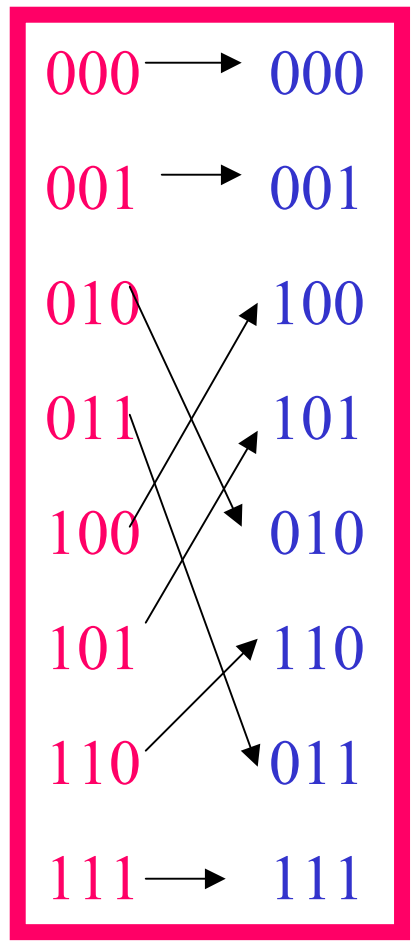
Reversible logic

Reversible are circuits (gates) that have the same number of inputs and outputs and have one-to-one ***mapping*** *between vectors of inputs and outputs; thus the vector of input states can be always reconstructed from the vector of output states.*

Feedback not allowed

Fan-out not allowed

Gate performs a permutation of vectors



Reversible logic

- A gate with ***k*** inputs and ***k*** outputs is called a ***k**k*** gate.
- A ***conservative*** circuit preserves the number of logic values in all combinations.
- In ***balanced binary logic*** the circuit has half of minterms with value 1.
- In ***ternary logic*** one third of minterms have value 0, one third value 1 and one third has value 2 – this is preserved from inputs to outputs.
- A circuit without constants on inputs which includes only reversible gates realizes on all outputs only balanced functions, therefore it can realize non-balanced functions only with ***garbage*** outputs.

Problem to Solve:

For arbitrary multi-output Boolean function, find its realization from reversible gates that minimizes the cost function

$$CF = \alpha c_1 + \beta c_2 + \gamma c_3$$

where c_1 is the gate cost, c_2 is the garbage cost, and c_3 is delay

General characteristic of logic synthesis methods for reversible logic

Very little has been published

Sasao and Kinoshita - Cascade circuits - ***small garbage , high delay***

Picton - binary and multiple-valued PLAs, ***high garbage, high delay, high gate cost***

Toffoli, Fredkin, Margolus - ***examples of good circuits, no systematic methods***

De Vos, Kerntopf - new gates and their properties, ***no systematic methods***

Knight, Frank, De Vieira, Athas, Svenson - circuit design, ***no systematic methods***

Joonho Lim, Dong-Gyu Kim and Soo-Ik Chae
School of Electrical Engineering, Seoul National University- circuit design, ***no systematic methods***

•We introduce regular structures to realize arbitrary functions.

Feynman Gate

- The ***2*2 Feynman gate***, called also ***controled-not*** or ***quantum XOR*** realizes functions ***P = A, Q = A @ B,*** where operator @ denotes EXOR..
- When ***A = 0*** then ***Q = B,*** when ***A = 1*** then ***Q = !B***.
- Every linear reversible function can be built by composing only 2*2 Feynman gates and inverters
- With ***B=0*** Feynman gate is used as a fan-out gate.

(universal) Binary Fredkin Gate

- Fredkin Gate is a fundamental concept in *reversible and quantum computing*.
- Every Boolean function can be build from 3 * 3 Fredkin gates:

  P = A,

  Q = if A then C else B,

  R = if A then B else C.

Multi-valued Fredkin Gate (Picton Gate)

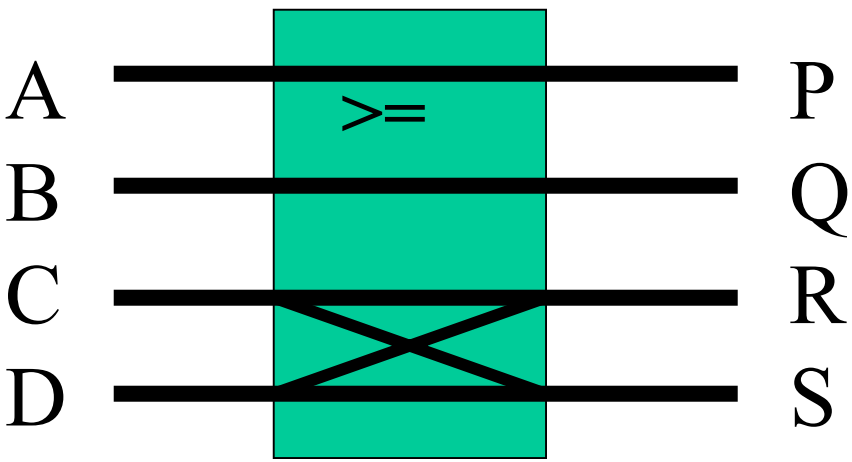
- MVFG is described by equations:

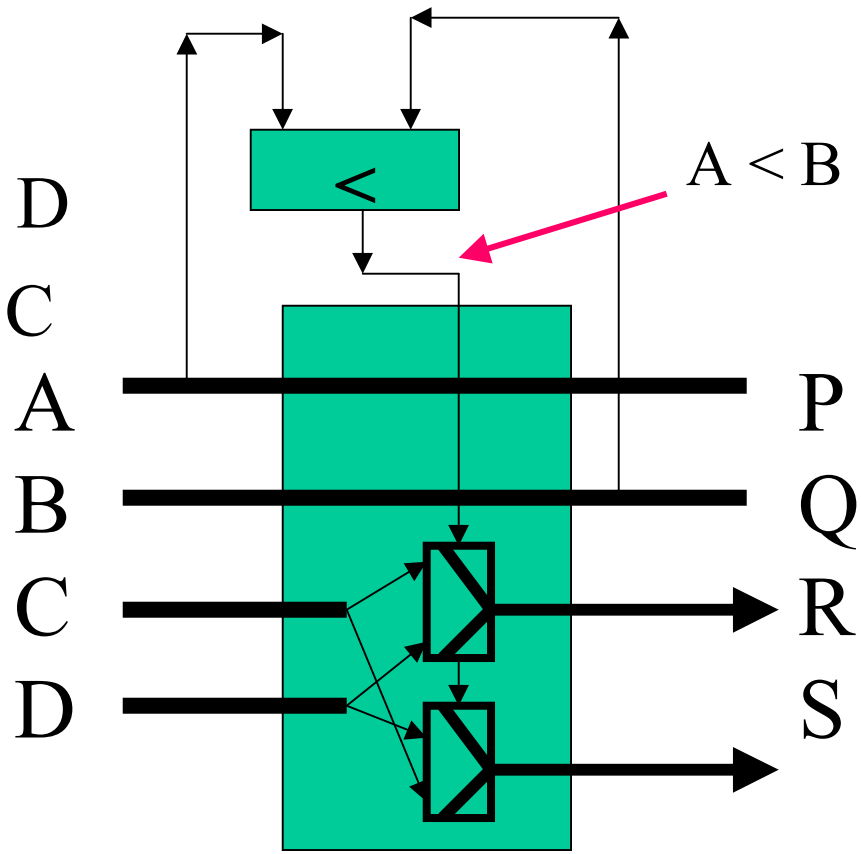
P = A

Q = B

if A < B ***then*** R = C ***else*** R = D

if A < B ***then*** S = D ***else*** S = C





Kerntopf Gate

- The Kerntopf gate is described by equations:

  $P = 1 \oplus A \oplus B \oplus C \oplus AB$,

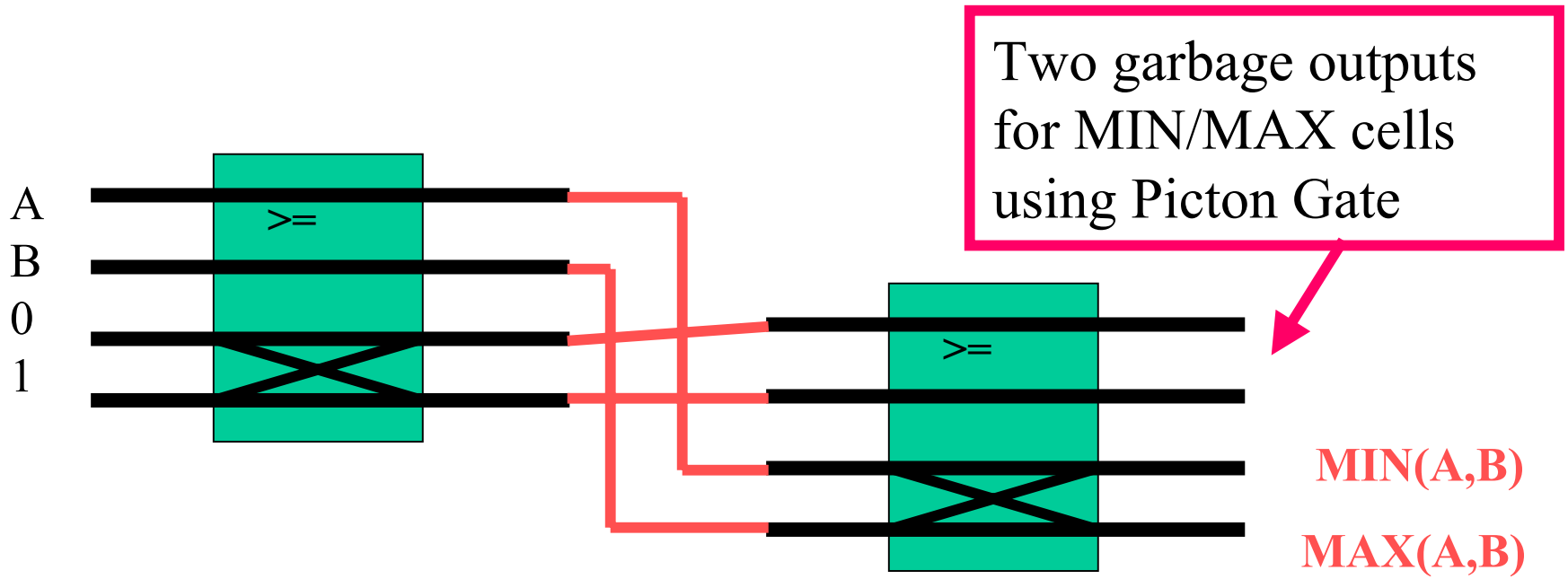
  $Q = 1 \oplus AB \oplus B \oplus C \oplus BC$,

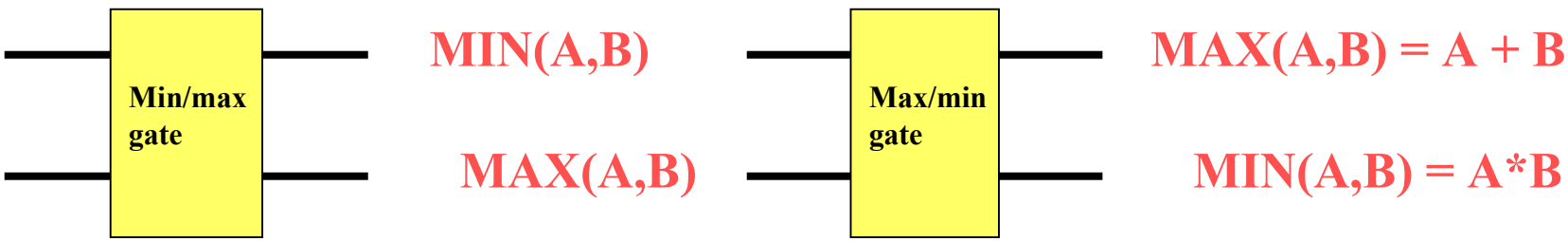
  $R = 1 \oplus A \oplus B \oplus AC$.

- When $C=1$ then $P = A + B$, $Q = A * B$, $R = !B$, so *AND/OR* gate is realized on outputs P and Q with C as the controlling input value.
- When $C = 0$ then $P = !A * ! B$, $Q = A + !B$, $R = A \oplus B$.

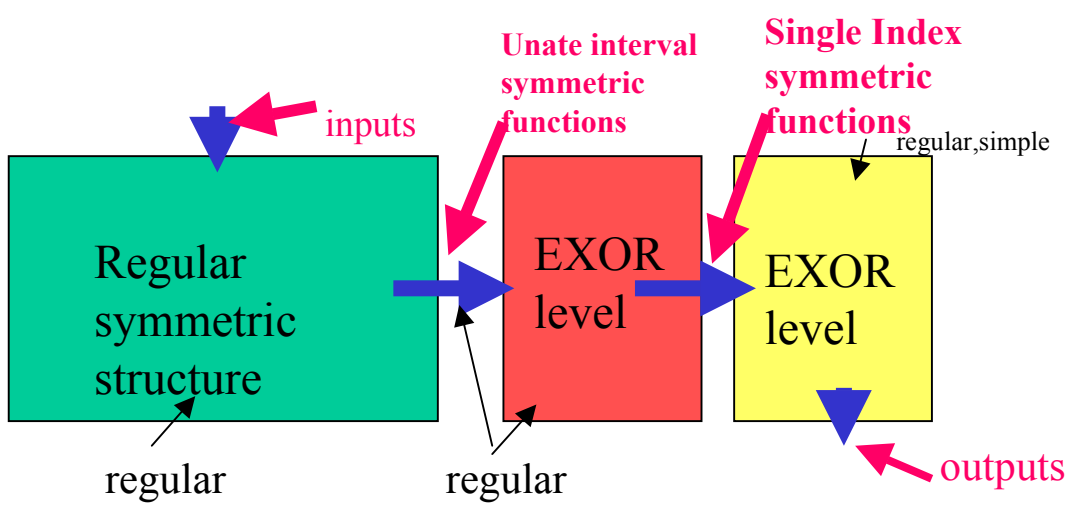
Despite theoretical advantages of Kerntopf gate over classical Fredkin and Toffoli gates, so far there are no published results on optical or CMOS realizations of this gate.

Use of two Multi-valued Fredkin (Picton) Gates to create MIN/MAX gate

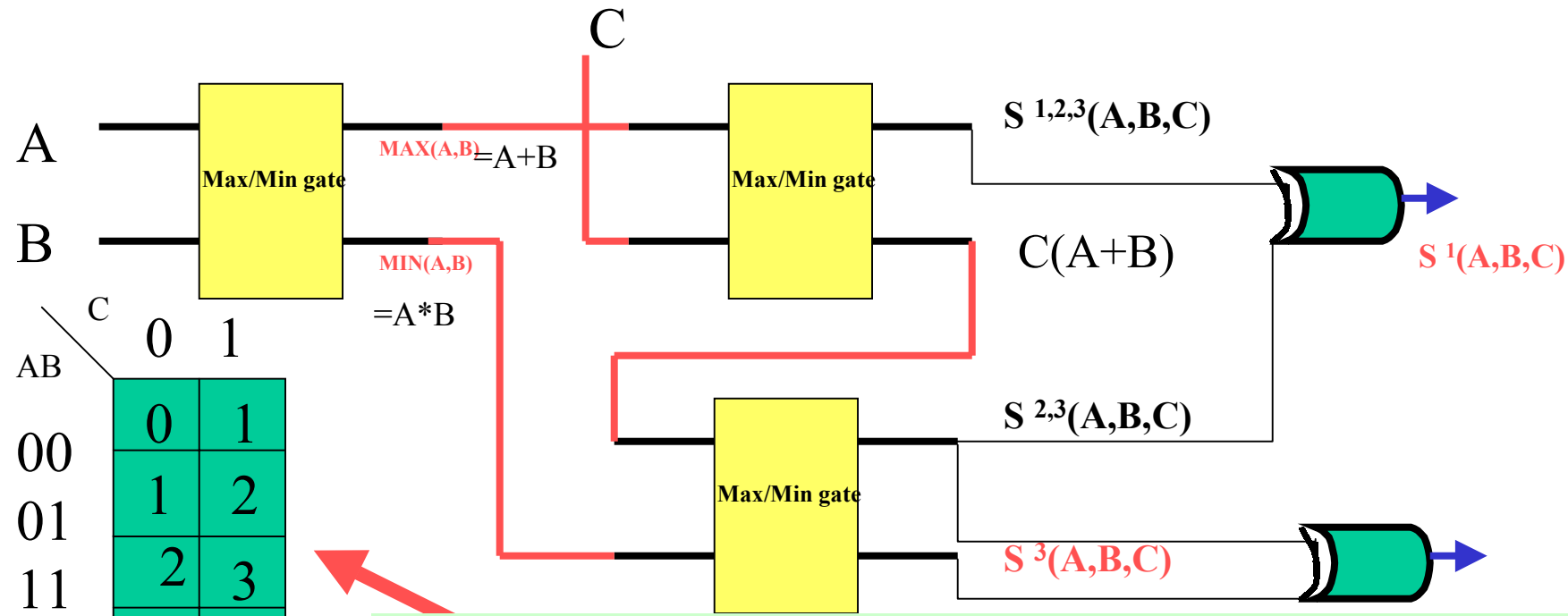




Regular Structure for Symmetric Functions

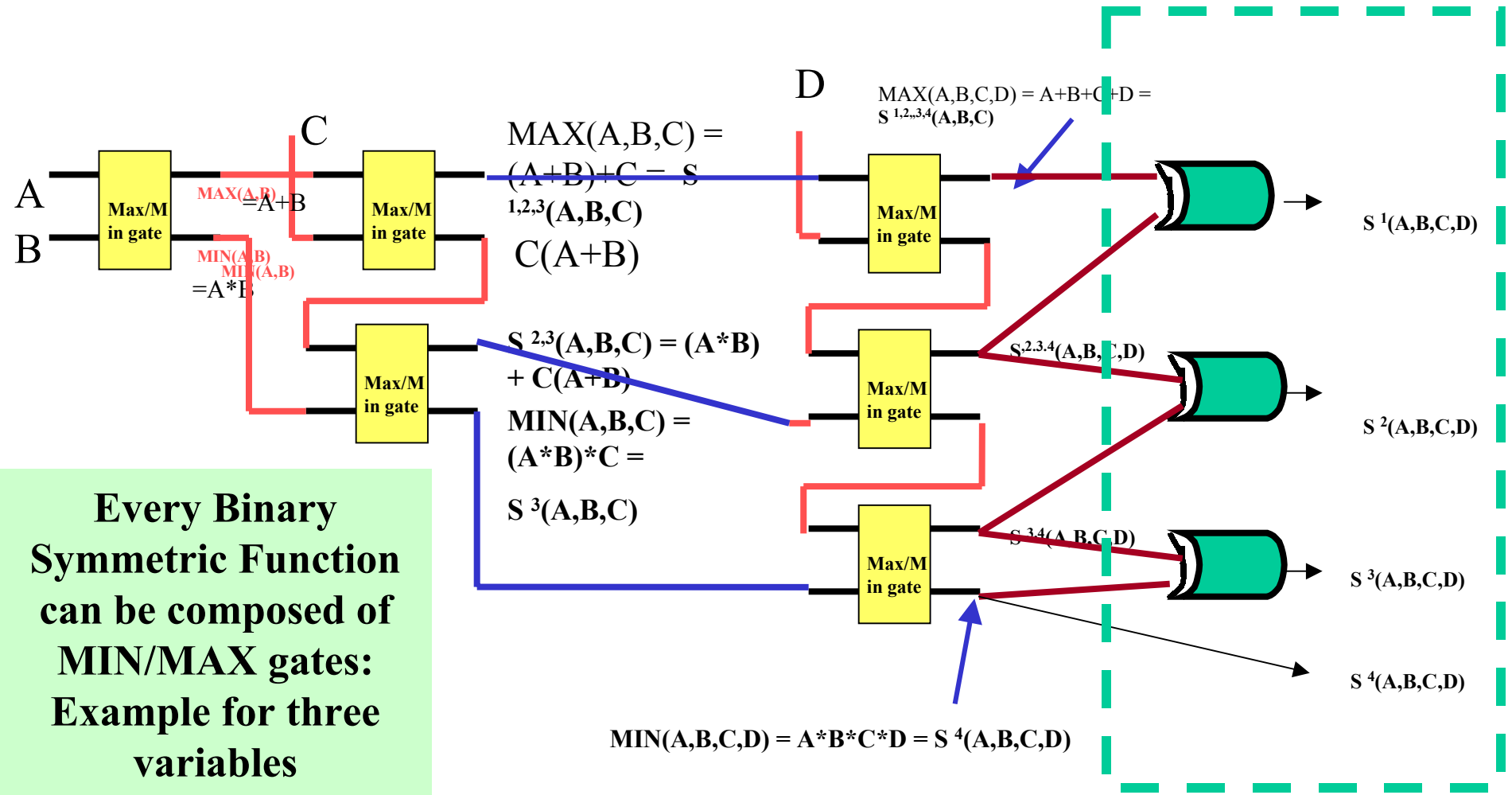


Every single index Symmetric Function can be created by EXOR-ing last level gates of the previous regular expansion structure



Indices of symmetric binary functions of 3 variables

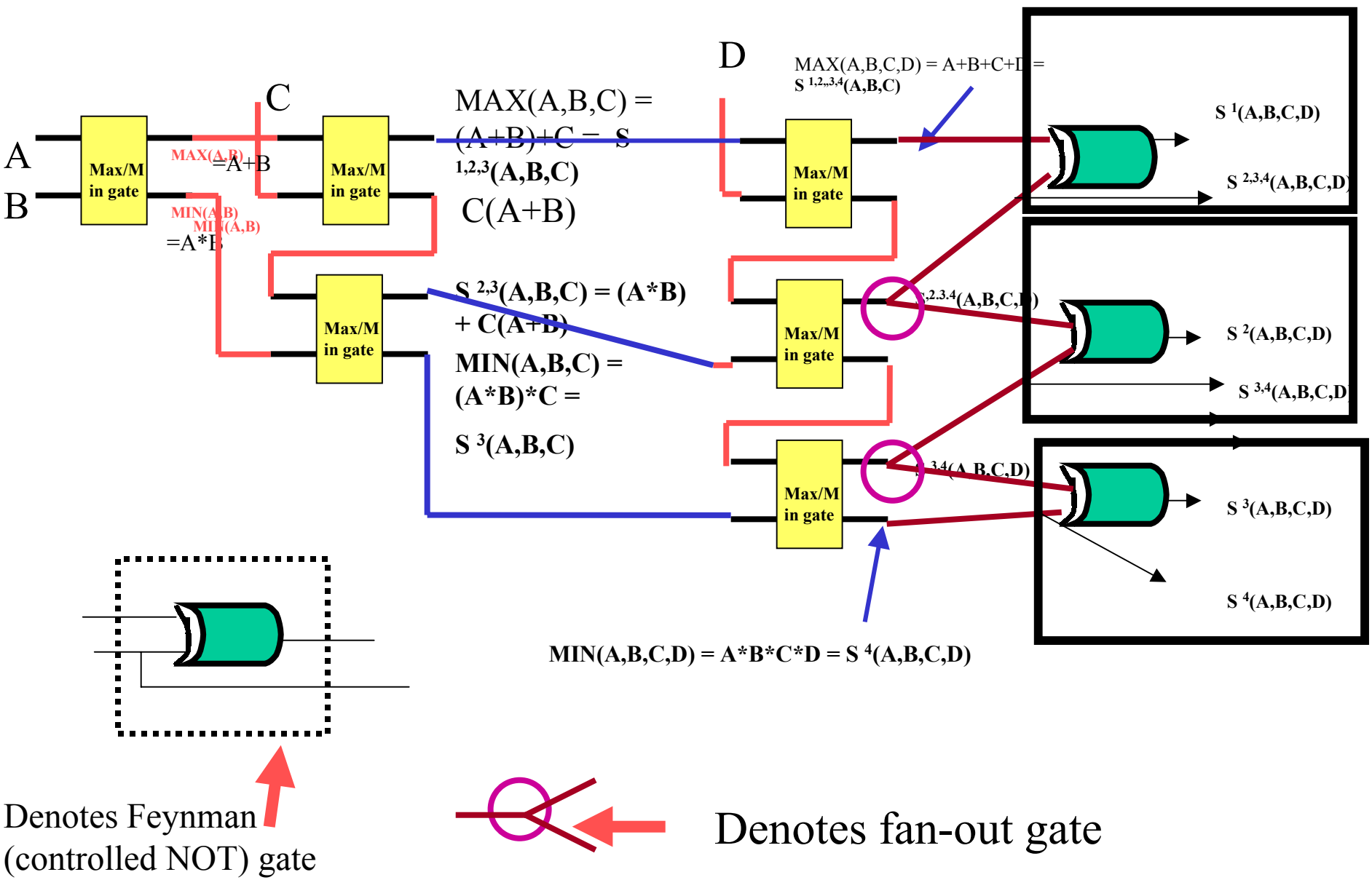
Example for four variables, EXOR level added



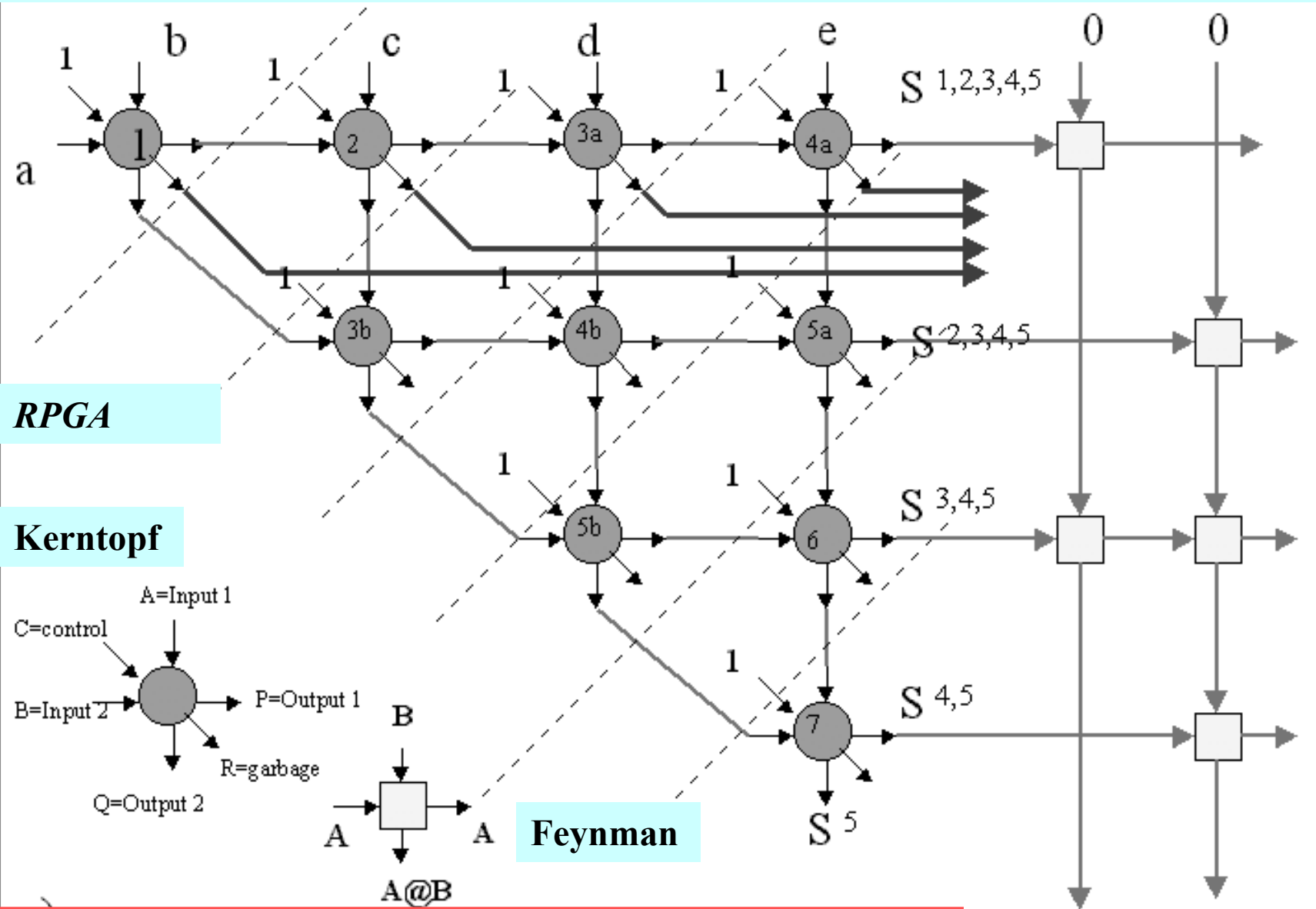
Every Binary Symmetric Function can be composed of MIN/MAX gates: Example for three variables

It is obvious that any multi-output function can be created by OR-ing the outputs of **EXOR level**

Now we extend to Reversible Logic



Using Kerntopf and Feynman Gates in **Reversible *Programmable Gate Array***



Arbitrary symmetric function can be created by exoring single indices

Generalizations and Current Work

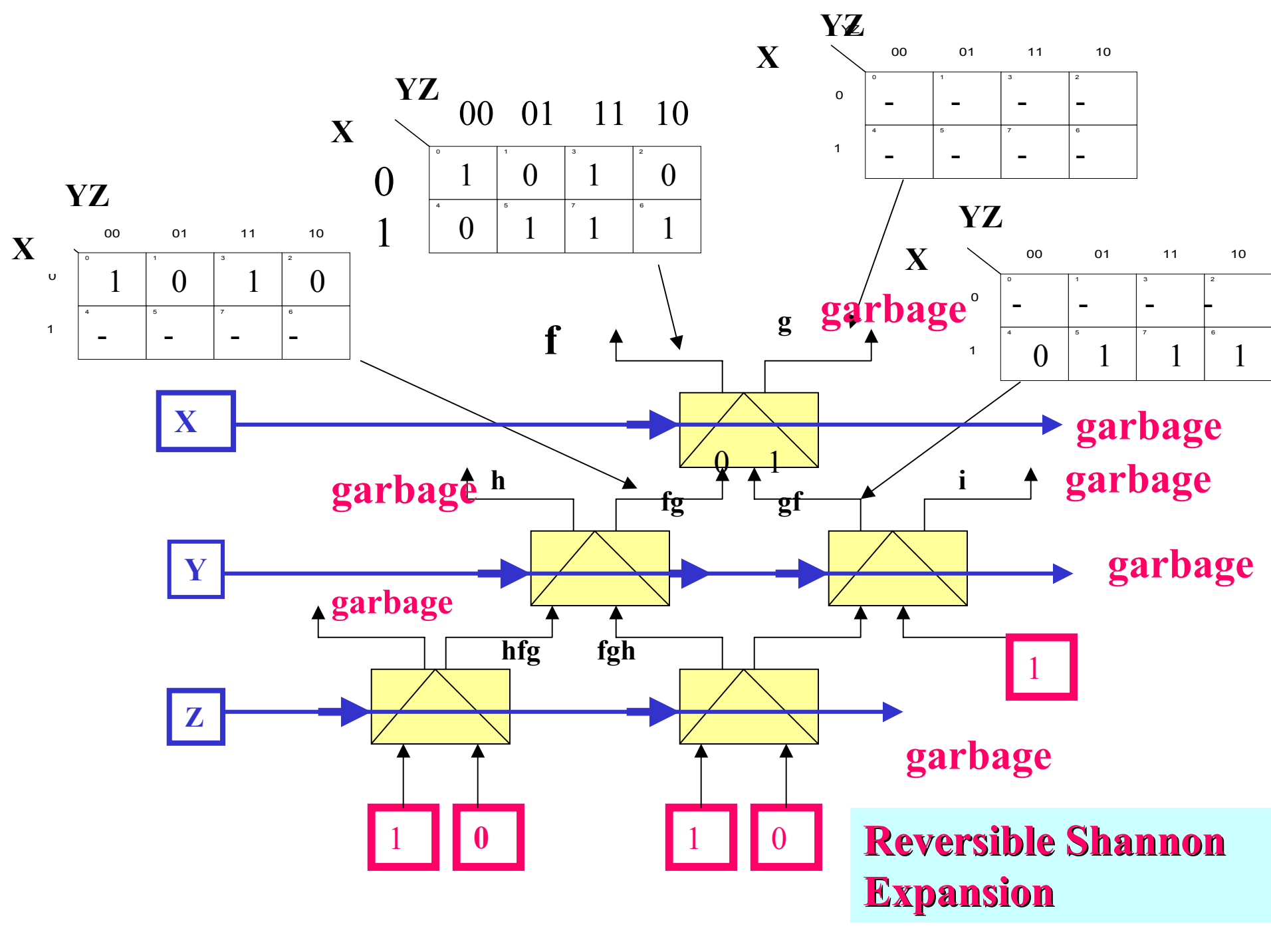
- Arbitrary symmetric function can be realized in a net without repeated variables.
- Arbitrary (non-symmetric) function can be realized in a net with ***repeated variables*** (so-called ***symmetrization***).
- Many non-symmetric functions can be realized in a net without ***repeated variables***.
- We work on the characterization of the functions realizable in these structures without repetitions and respective synthesis algorithms.

Very many new circuit types, which are reversible and multi-valued generalizations of Shannon Lattices, Kronecker Lattices, and other regular structures introduced in the past.

- *Layout-driven synthesis* to regular structures
- **CMOS**, Optical, **Quantum dot** technologies.
- Software

Approaches developed by PQLG

1. **ESOP-like PLAs** - Toffoli and Feynman gates
2. **Lattice diagrams** - Fredkin gates
3. **Nets** - Multi-valued Fredkin (or Kerntopf) and Feynman gates
4. **Cascades** - *Sasao,* all kinds of gates
5. **Compositional methods** - *Jozwiak,* all kinds of gates
6. **Decompositional methods** - *Ashenhurst-Curtis, Bi-Decomposition,* all kinds of gates
7. **Levelized decompositions** - k-through gates
8. **Mapping from decision diagrams** - k-through gates
9. **Permutation-based and group-theory-based** - all kinds of gates


YZ
X
00 01 11 10
0 1 0 1 0
1 0 1 1 1
f
g
garbage
X
Y
Z
h
fg
gf
i
hfg
fgh
0 1
1
1 0
1 0
garbage
garbage
garbage
garbage
garbage
garbage
Reversible Shannon Expansion

Regular Structures and Decompositions of Reversible Logic

- New concepts:
  - ***(1)*** Reversible Shannon Expansion for ***k*k*** binary Fredkin Gates ***(k>2),***
  - ***(2)*** planar and 3-D Reversible Fredkin Lattice structures for logic based on binary Fredkin gates,
  - ***(3)*** other regular structures,
  - ***(4)*** use of symmetry, unateness, threshold logic,
  - ***(5)*** new universal reversible gates,
  - ***(6)*** adaptation of BDD-based, decomposition-based and technology mapping methods from standard binary logic
  - **(7)** generalizations to multiple-valued logic.

Quantum Computing Revolution is coming.

What can be done by researchers who have no access to well-funded laboratories?

- Portland Quantum Logic Group is an attempt to **give an answer** to the above question.
- We are a group of researchers interested in ***collaboration*** on reversible logic, quantum logic, reversible and quantum computing and nano-technologies.
- We share information about new developments, we collaborate on developing new methodologies, algorithms and circuits.
- We discuss theoretical problems and realization issues.

PQLG current work includes:

- **1.** Designing circuits in reversible technologies
- **2.** Designing gates in new reversible technologies.
- **3.** Developing logic, system and layout methodologies for reversible and quantum logic (every quantum logic is reversible).
- **4.** Designing systems (such as microprocessors) using reversible logic.
- **5.** Developing highly efficient algorithms and computer programs for logic synthesis in reversible logic.

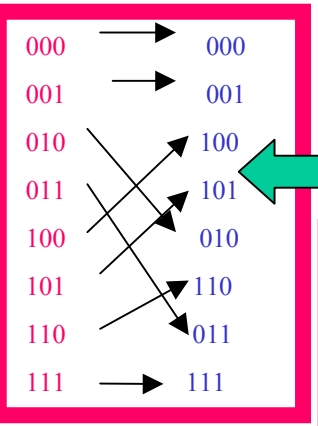
- **6.** Developing test algorithms and design for test methods for reversible technologies.
- **7.** Developing formal verification and validation methods for reversible systems
- **8.** Developing concepts of multiple-valued, fuzzy and continuous reversible logic.
- **9.** Developing highly efficient data representations for reversible logic.

Regularity and Symmetry as a Base for Efficient Realization of Reversible Logic Circuits

PQLG

- Bennett showed that for power not be dissipated in the circuit it is necessary that arbitrary circuit can be build from ***reversible gates***.

INPUTS OUTPUTS



Feedback not allowed

Fan-out not allowed

Gate performs a permutation of vectors

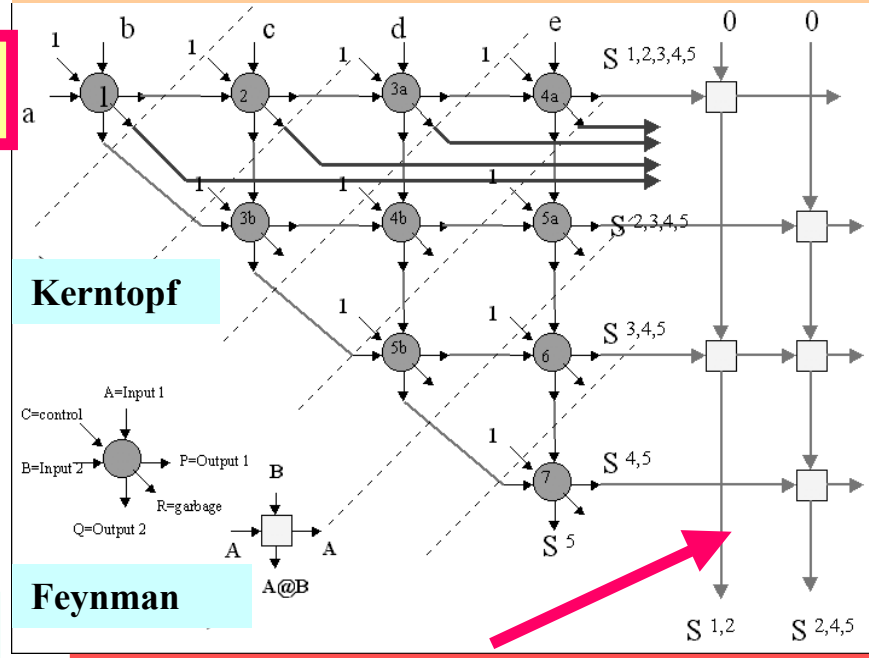
•The *2*2* ***Feynman gate***, called also *controled-not* or *quantum XOR* realizes functions *P = A, Q = A @ B,* where operator @ denotes EXOR..
•Every linear reversible function can be built by composing only 2*2 Feynman gates and inverters
•With *B=0* Feynman gate is used as a fan-out gate.

The **Kerntopf gate** is described by equations:

$P = 1 \oplus A \oplus B \oplus C \oplus AB$,
$Q = 1 \oplus AB \oplus B \oplus C \oplus BC$,
$R = 1 \oplus A \oplus B \oplus AC$.

When *C=1* then $P = A + B$, $Q = A * B$, $R = !B$, so *AND/OR* gate is realized on outputs *P* and *Q* with *C* as the controlling input value.

Using Kerntopf and Feynman Gates in Reversible *Programmable Gate Array*



Arbitrary symmetric function can be created by exoring single indices

Problem to Solve:

For arbitrary multi-output Boolean function, find its realization from reversible gates that minimizes the cost function

$$CF = \alpha c_1 + \beta c_2 + \gamma c_3$$

where c_1 is the gate cost, c_2 is the garbage cost, and c_3 is delay