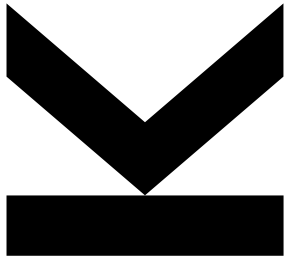
APPLICATION OF SYNTHESIS APPROACHES



Robert Wille

Johannes Kepler University Linz, Austria

robert.wille@jku.at

FURTHER READING

- **Adiabatic Circuits**
  Foundations of Generalized Reversible Computing, RC 2017
  Exploiting Reversible Logic Design for Implementing Adiabatic Circuits, MIXDES 2017
- **Encoder Design**
  Taking One-to-one Mappings for Granted: Advanced Logic Design of Encoder Circuits, DATE 2017
  Synthesis of Approximate Coders for On-chip Interconnects Using Reversible Logic, DATE 2016
  Automatic Design of Low-Power Encoders Using Reversible Circuit Synthesis, DATE 2012
- **Simulation**
  Exploiting Reversibility in the Complete Simulation of Reversible Circuits, AFRICON 2013
- **ATPG**
  A Family of Logical Fault Models for Reversible Circuits, ATS 2015
- **Verification**
  Exploiting Inherent Characteristics of Reversible Circuits for Faster Combinational Equivalence Checking, DATE 2016

APPLICATIONS

- Quantum Computation

APPLICATIONS

- ~~Quantum Computation~~ → Later today

APPLICATIONS

- ~~Quantum Computation~~ → Later today
- Low Power Design

LOW POWER DESIGN #1

- **Rolf Landauer** (1961):

  Deletion of a single bit leads to a power dissipation of $W = kT \ln 2$

  T…Temperature

  k…Boltzmann-Constant

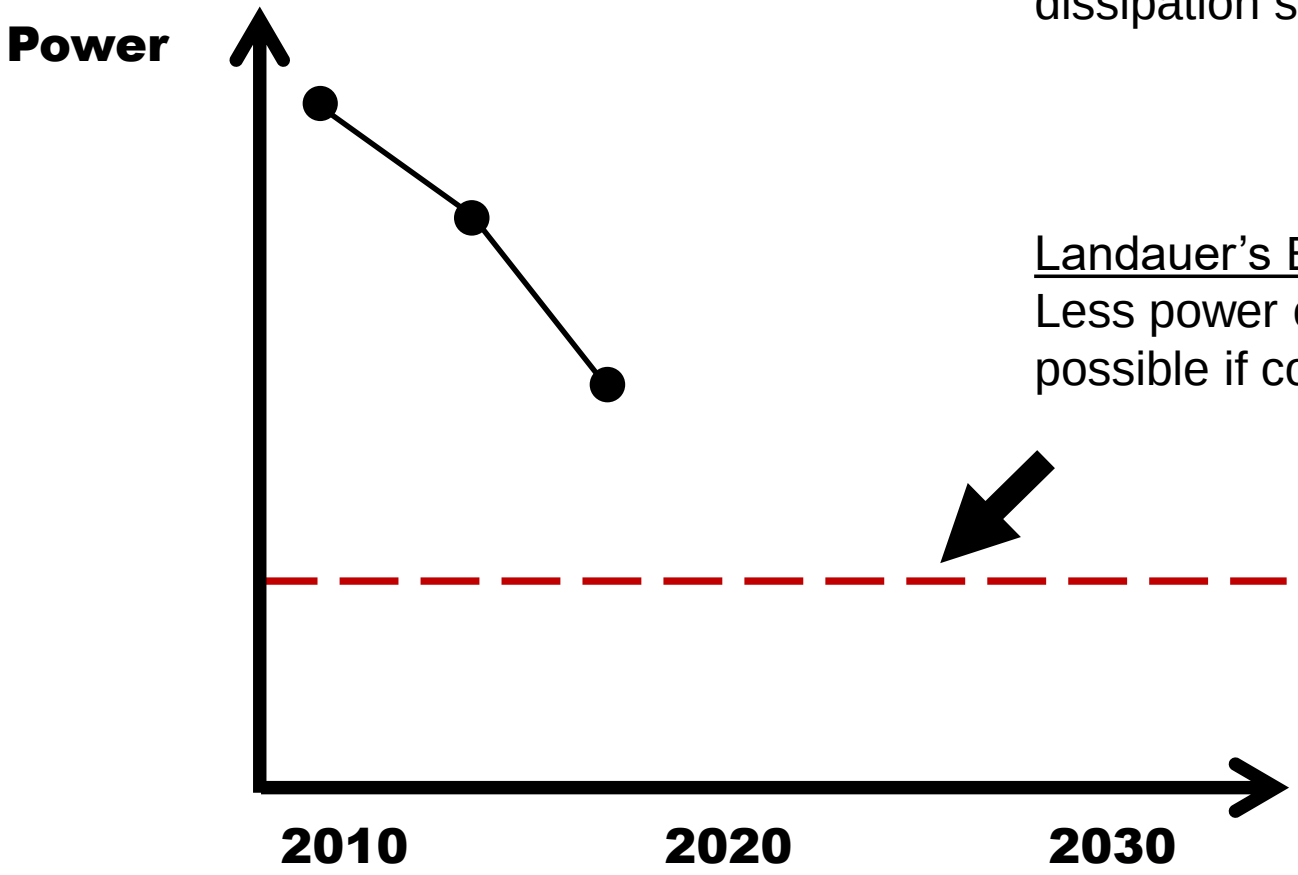
  ($1{,}381 \cdot 10^{-23}$ J/K)

- **Charles H. Bennett** (1973):

  Circuits without energy dissipation have to be reversible

LOW POWER DESIGN #2

Important:
Does not mean that each implementation of a reversible circuit is “automatically” power-efficient!
→ How to exploit reversibility for reducing power dissipation still open research problem



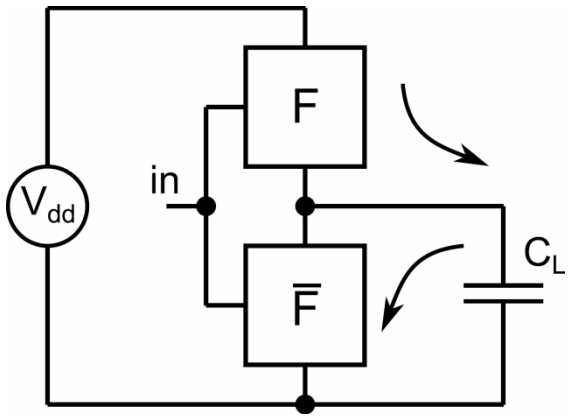
Landauer’s Barrier
Less power consumption is only possible if computation is reversible

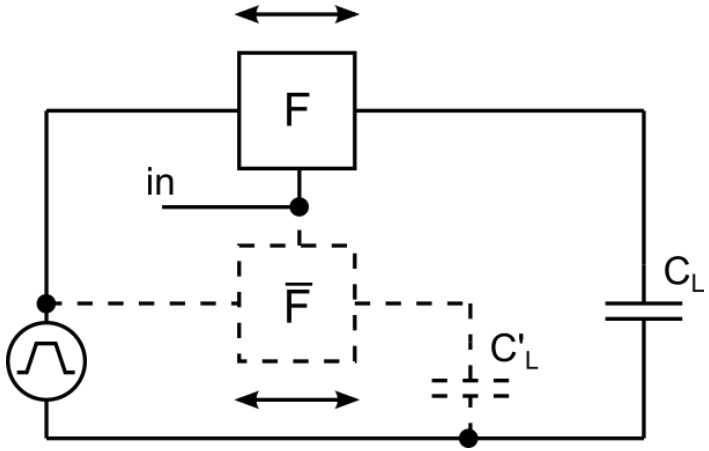
APPLICATIONS

- ~~Quantum Computation~~ → Later today
- Low Power Design
- Adiabatic Circuits

ADIABATIC CIRCUITS

- adiabatic (from Thermodynamics)
  - no transfer of heat or matter into or out of the system
- adiabatic (in electronics)
  - no (or minimal) energy loss





ADIABATIC CHARGING

$$V_C(t) = \frac{1}{C}\int I(t)dt = \frac{1}{C} I_{avg}(t)t$$

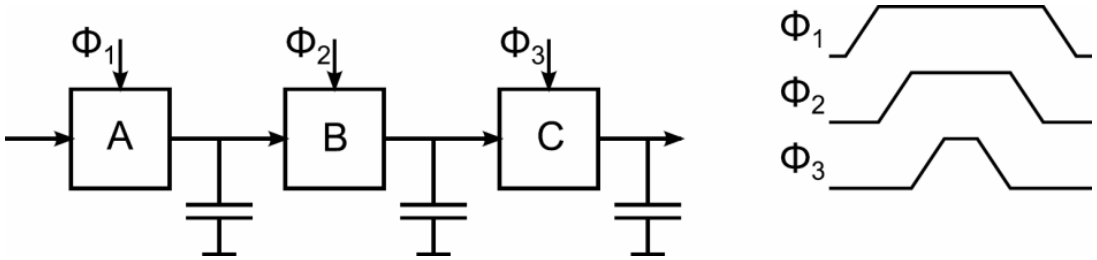
$$I_{avg}(t) = \frac{CV_C(t)}{t}$$

$$E_{diss} = R\int_0^T I(t)^2 dt \geq R\int_0^T I_{avg}(t)^2 dt = RI_{avg}(T)^2 T = \frac{RC}{T} CV_C(T)^2$$

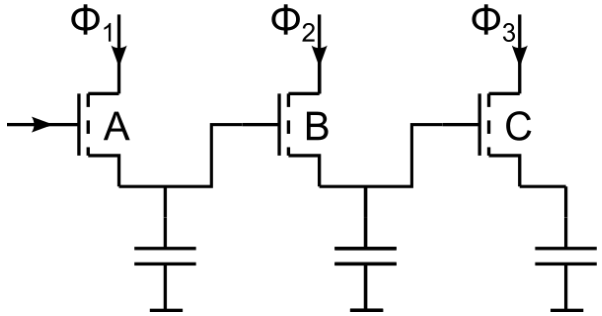
- Supply Voltage
  - ramping up the voltage results in a constant current
- additional Rules
  - Never turn on a transistor if there is a voltage difference between drain and source.
  - Never turn off a transistor if there is a current through it.

COMPLEX ADIABATIC CIRCUITS

- pipeline structure

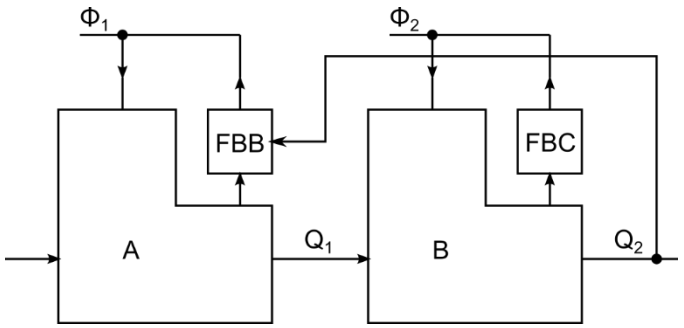


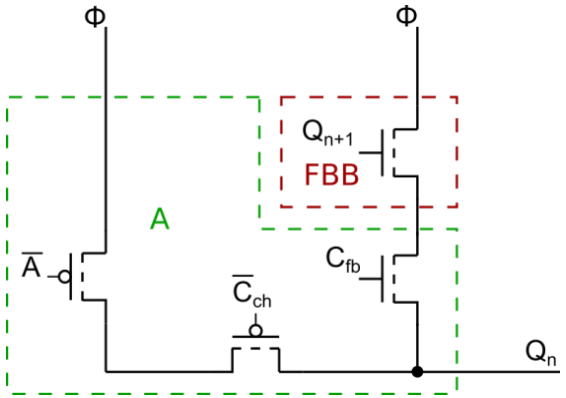
- To respect the rules mentioned
  - A must not change until B has been charged and discharged
  - B must not change until C has been charged and discharged



SEPARATE DISCHARGE PATH

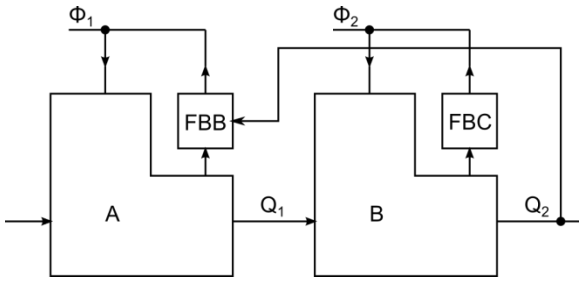
- Input is allowed to change before discharge phase
- Signal has no longer to propagate through whole chain and back

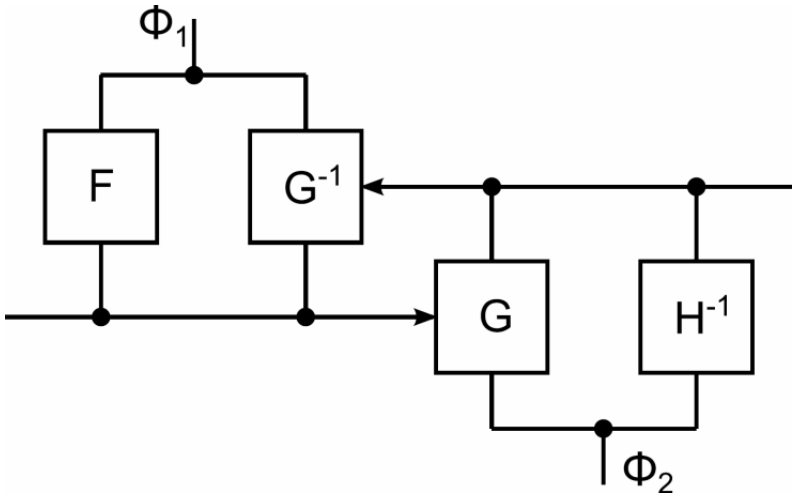




DISCHARGE PATH

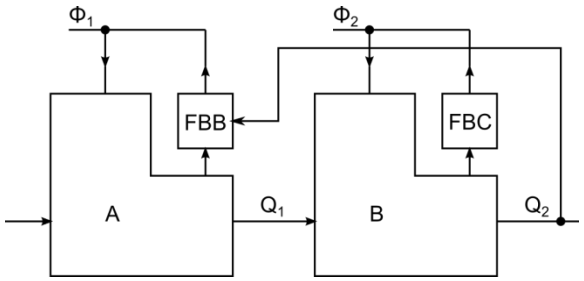
- Never turn on a transistor if there is a voltage difference between drain and source
- Discharge ONLY when charged!
  - State (charged/uncharged) of cell F can be determined by output of following cell G via function G^{-1}
  - Output of following cell G controlls discharge path of F
  - G must be reversible to realize G^{-1}

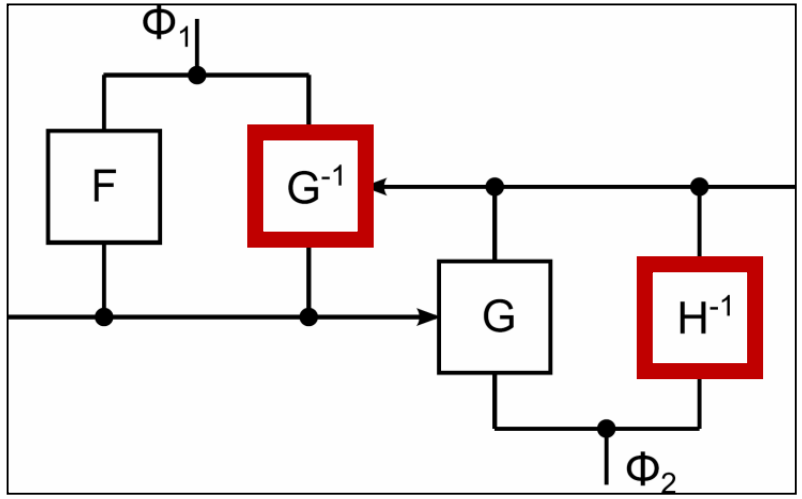




DISCHARGE PATH

- Never turn on a transistor if there is a voltage difference between drain and source
- Discharge ONLY when charged!
  - State (charged/uncharged) of cell F can be determined by output of following cell G via function G^{-1}
  - Output of following cell G controlls discharge path of F
  - **G must be reversible to realize G^{-1}**





APPLICATIONS

- ~~Quantum Computation~~ → Later today
- Low Power Design
- Adiabatic Circuits
- Encoders

ENCODERS

■ Complete

$x_3x_2x_1$	$x_3x_2x_1$
000	000
001	101
010	011
011	111
100	010
101	100
110	110
111	001

■ Incomplete

$x_3x_2x_1$	$x_3x_2x_1$
000	000
001	–
010	010
011	–
100	–
101	001
110	–
111	–

■ Application-specific

$x_3x_2x_1$	Hw		$x_3x_2x_1$
000	1		001
001	0		000
010	2		011
011	3	⇒	111
100	2		110
101	1		010
110	2		101
111	1		100

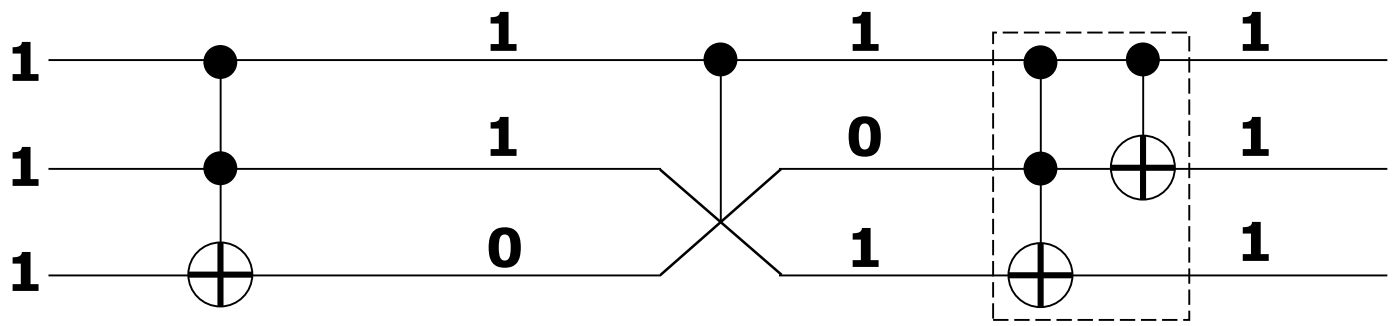
■ Challenges

- ☐ Exponential Complexity
- ☐ Guaranteeing one-to-one mappings
- ☐ Exploiting the full degree of freedom

$$\sum_{i=0}^{m} \binom{m}{i}$$

USING REVERSIBLE LOGIC

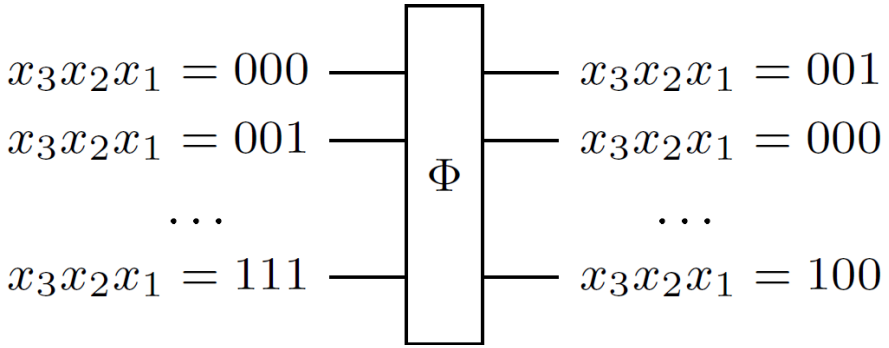
- Reversible circuits: Get a 1:1 mapping “for free”!



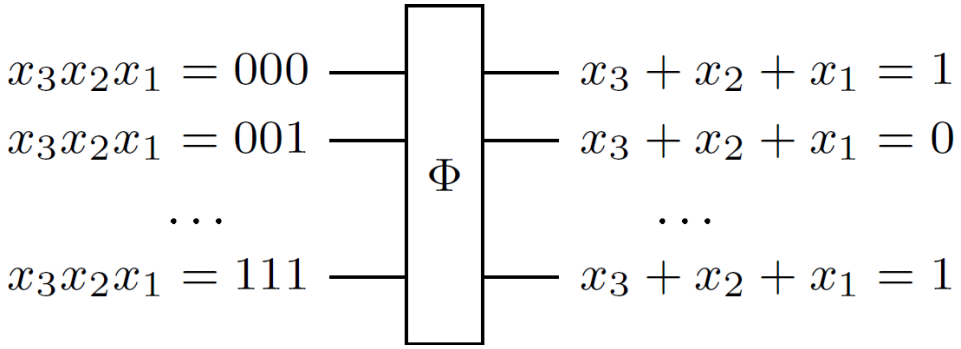
- Challenges
  - ☐ Exponential Complexity
  - ☑ Guaranteeing one-to-one mappings
  - ☐ Exploiting the full degree of freedom

USING REVERSIBLE LOGIC

■ State-of-the-art:



■ Using Reversible Logic:



■ Application-specific

$x_3x_2x_1$	Hw		$x_3x_2x_1$
000	1		001
001	0		000
010	2		011
011	3		111
100	2	$\Rightarrow$	110
101	1		010
110	2		101
111	1		100

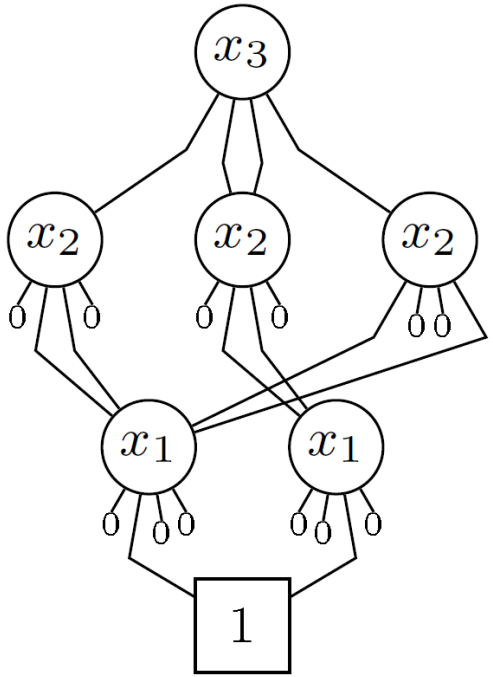
■ Challenges

- ☐ Exponential Complexity
- ☑ Guaranteeing one-to-one mappings
- ☑ Exploiting the full degree of freedom

JKU

USING REVERSIBLE LOGIC

- QMDDs



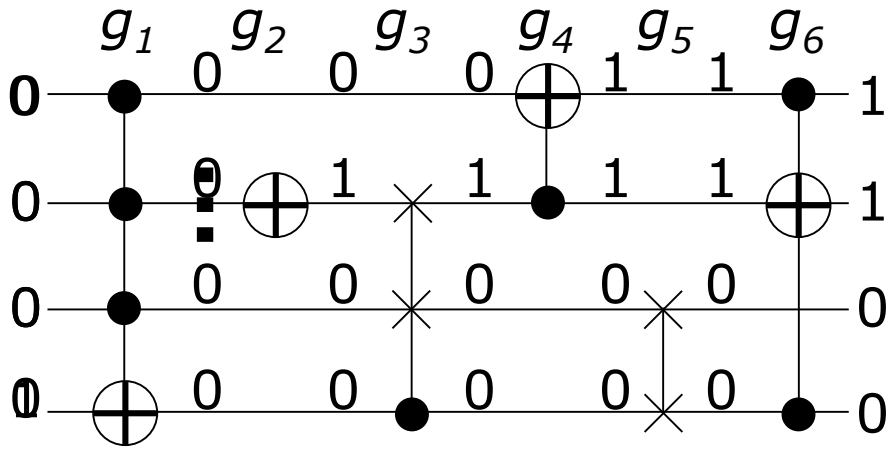
Outputs \ Inputs (x_1 x_2 x_3)	000	001	010	011	100	101	110	111
000	0	0	0	**1**	0	0	0	0
001	0	0	0	0	0	0	**1**	0
010	0	**1**	0	0	0	0	0	0
011	0	0	0	0	**1**	0	0	0
100	0	0	0	0	0	**1**	0	0
101	0	0	**1**	0	0	0	0	0
110	0	0	0	0	0	0	0	**1**
111	**1**	0	0	0	0	0	0	0

- Challenges
  - ✓ Exponential Complexity
  - ✓ Guaranteeing one-to-one mappings
  - ✓ Exploiting the full degree of freedom

APPLICATIONS

- ~~Quantum Computation~~ → Later today
- Low Power Design
- Adiabatic Circuits
- Encoders
- Further
  - ☐ Complete Simulation

COMPLETE SIMULATION



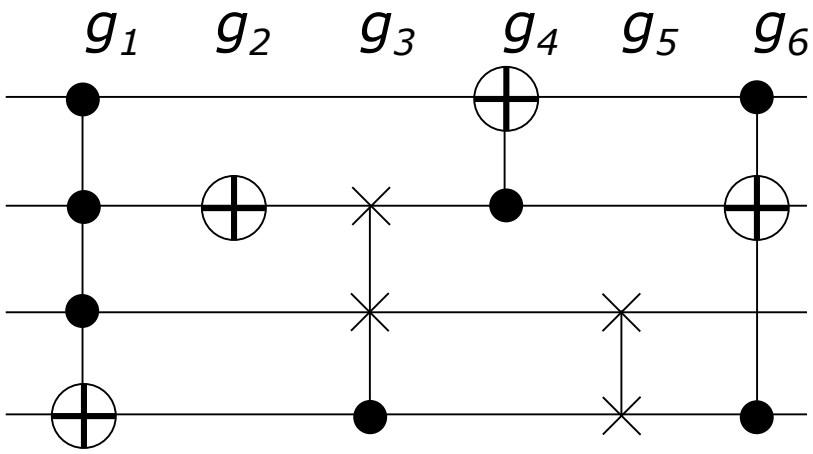
➔ Run-time complexity: $2^n d$
(n: number of line; d: number of gates)

EXPLOITING REVERSIBILITY

- Functional effect of single gates
- For many input pattern, single gates have no effect
  (only patterns with control lines set to 1)

- Reversible gates realize permutations
- If an input pattern 0000 maps to the output 0001, then 0001 also maps to 0000
  □ Each gate eventually leads to an exchange of patterns

EXPLOITING REVERSIBILITY



Input patterns with effect:

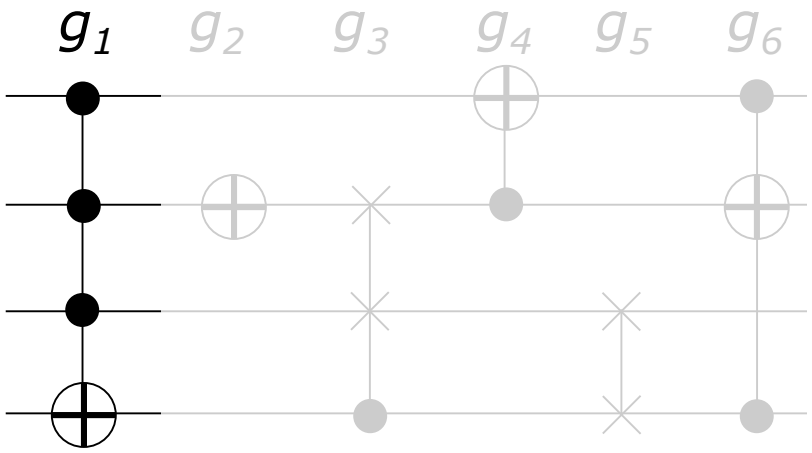
g_1: 2 g_4: 8

g_2: 16 g_5: 16

g_3: 8 g_6: 4

- Reversible gates realize permutations
  - g_1: Swaps 1110 and 1111
  - …
  - g_6: Swaps all pattern of the form 11-1 with corresponding pattern of the form 10-1

EXPLOITING REVERSIBILITY



Input patterns with effect:

g_1: 2

- Reversible gates realize permutations
  - g_1: Swaps 1110 and 1111

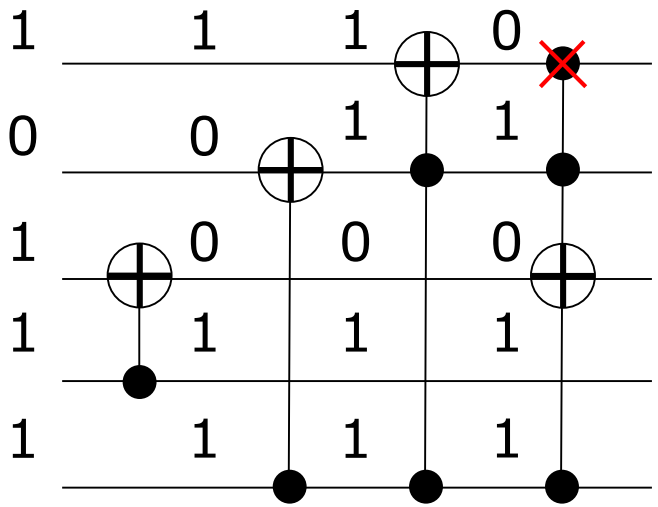
➔ Simulation of g_1 requires a single swap operation (compared to 16 simulation steps)

APPLICATIONS

- ~~Quantum Computation~~ → Later today
- Low Power Design
- Adiabatic Circuits
- Encoders
- Further
  - Complete Simulation
  - ATPG

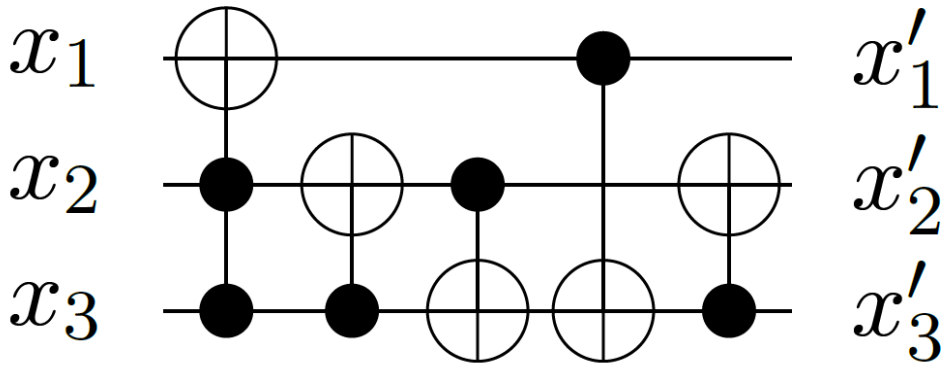
TEST OF REVERSIBLE CIRCUITS

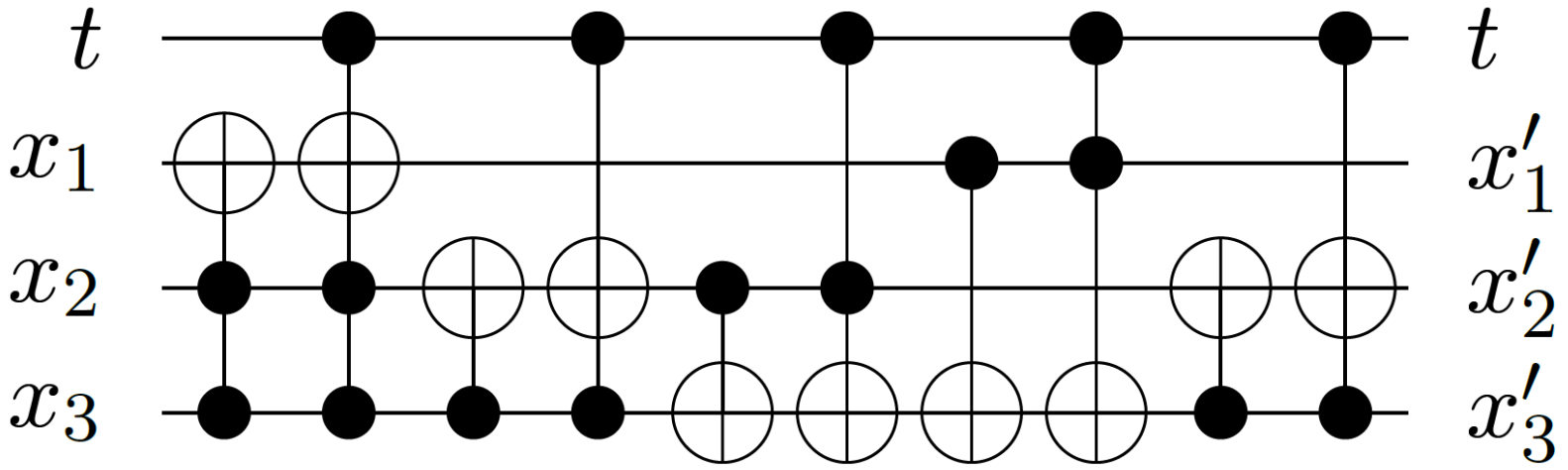
- Generate an input assignment that
  - ☐ triggers the faulty behavior (controlability)
  - ☐ shows the faulty behavior (observability)



- In general easy for reversible circuits
  (except additional restrictions like constant inputs need to be considered)

DESIGN FOR TESTABILITY





APPLICATIONS

- ~~Quantum Computation~~ → Later today
- Low Power Design
- Adiabatic Circuits
- Encoders
- Further
  - ☐ Complete Simulation
  - ☐ ATPG
  - ☐ Verification

VERIFICATION/ EQUIVALENCE CHECKING

- Full connectivity
  (verification by simulation is more promising)

Circuit	No. lines	No. gates	Runtime (conv.)	Runtime (rand. sim.)	No. pattern
alu	101	4472	>500.00	0.03	7
avg16	274	1601	>500.00	0.01	4
avg16	546	3297	>500.00	0.04	3
avg8	147	860	>500.00	0.00	2
avg16	138	753	44.54	0.00	5
apex5	240	3307	42.10	0.06	7
cps	31	2668	26.40	0.03	9
ex1010	196	2981	23.93	0.03	2
table3	109	1987	16.66	0.08	572
pdc	178	2079	14.88	0.03	83
avg8	291	1772	12.97	0.01	2

VERIFICATION/ EQUIVALENCE CHECKING

- Full connectivity
  (verification by simulation is more promising)
- Search space is smaller
- Reversible Miter (ff^{-1})

VERIFICATION/ EQUIVALENCE CHECKING

- Full connectivity
  (verification by simulation is more promising)
- Search space is smaller
- Reversible Miter (ff^{-1})
- XOR-richness
- Undo computations when an error occurred
- …

APPLICATIONS

- ~~Quantum Computation~~ → Later today
- Low Power Design
- Adiabatic Circuits
- Encoders
- Further
  - ☐ Complete Simulation
  - ☐ ATPG
  - ☐ Verification