

Lecture 3

Combinational units.

Adders

Computing platforms, semester 2

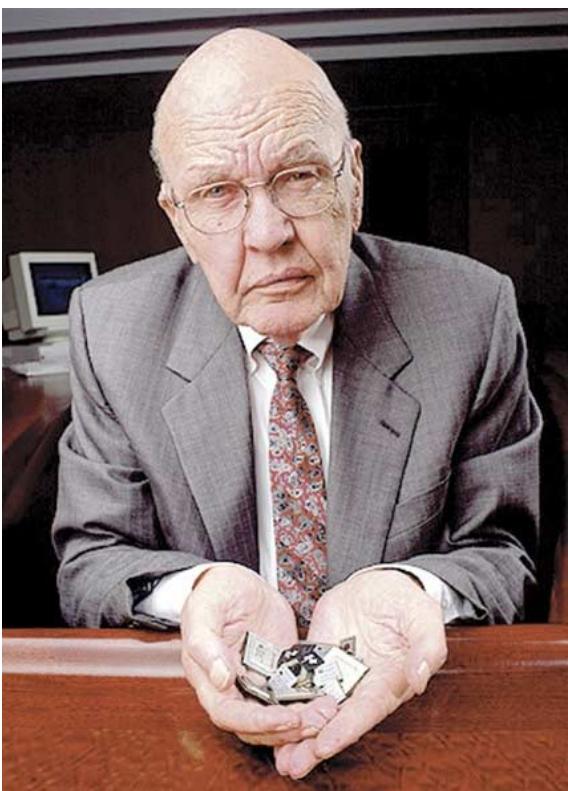
Novosibirsk State University
University of Hertfordshire

D. Irtegov, A.Shafarenko

2019

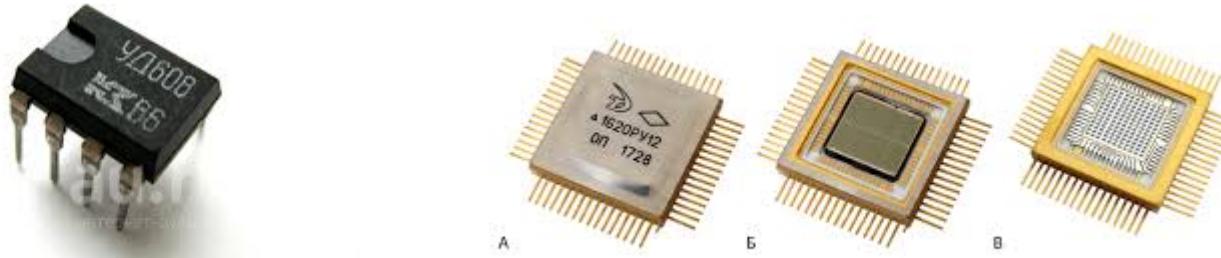
История

12 сентября 1958 года сотрудник фирмы Texas Instruments (TI) Джек Килби продемонстрировал руководству странный прибор - склеенный пчелиным воском на стеклянной подложке устройство из двух кусочков кремния размером 11,1x1,6 мм. Это был объёмный макет – прототип интегральной схемы (ИС) генератора, доказывающий возможность изготовления всех элементов схемы на основе одного полупроводникового материала. Эта дата отмечается в истории электроники как день рождения интегральных схем.

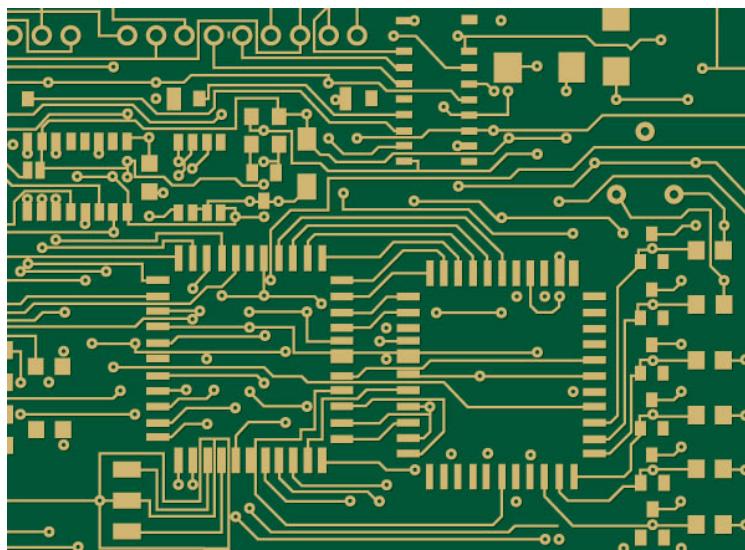


Джек Сент Клер Килби – американский электротехник, один из создателей первой интегральной схемы, нобелевский лауреат по физике. Известен также как создатель первого карманного калькулятора и первого термального принтера.

История



К интегральным схемам (микросхемам, ИС) относятся электронные устройства различной сложности, в которых все однотипные элементы изготавливаются одновременно в едином технологическом цикле, т.е. по интегральной технологии. В отличие от печатных плат (в которых в едином цикле по интегральной технологии одновременно изготавливаются все соединительные проводники) в ИС аналогично формируются и резисторы, и конденсаторы, и диоды и транзисторы.

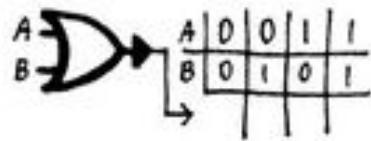


Some terminology

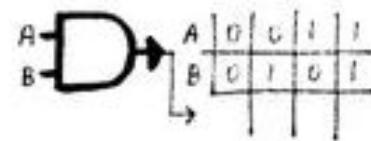
- Combinational units: pure logical functions.
 - Output depends only on input
 - No side effect (except delay)
 - Can be described by single logical expression
- Sequential units: have internal state
 - Output depends not only on the inputs
 - Input can have side-effect (changing internal state)
 - Require logical elements we did not study yet.

Типы комбинационных логических схем

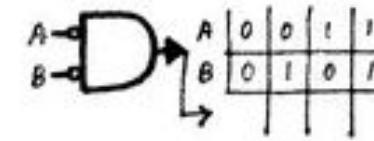
OR:



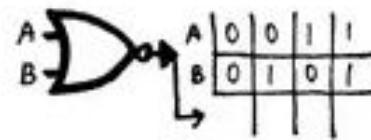
AND:



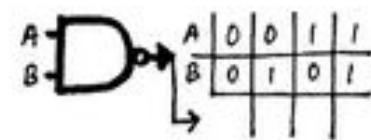
NEG-AND:



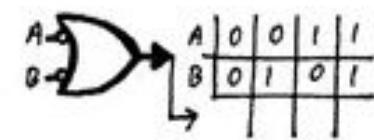
NOR:



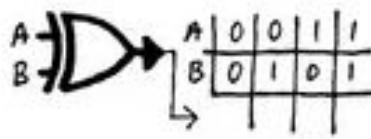
NAND:



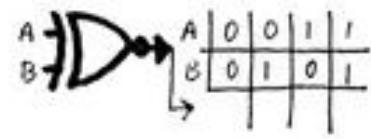
NEG-OR:



XOR:



XNOR:



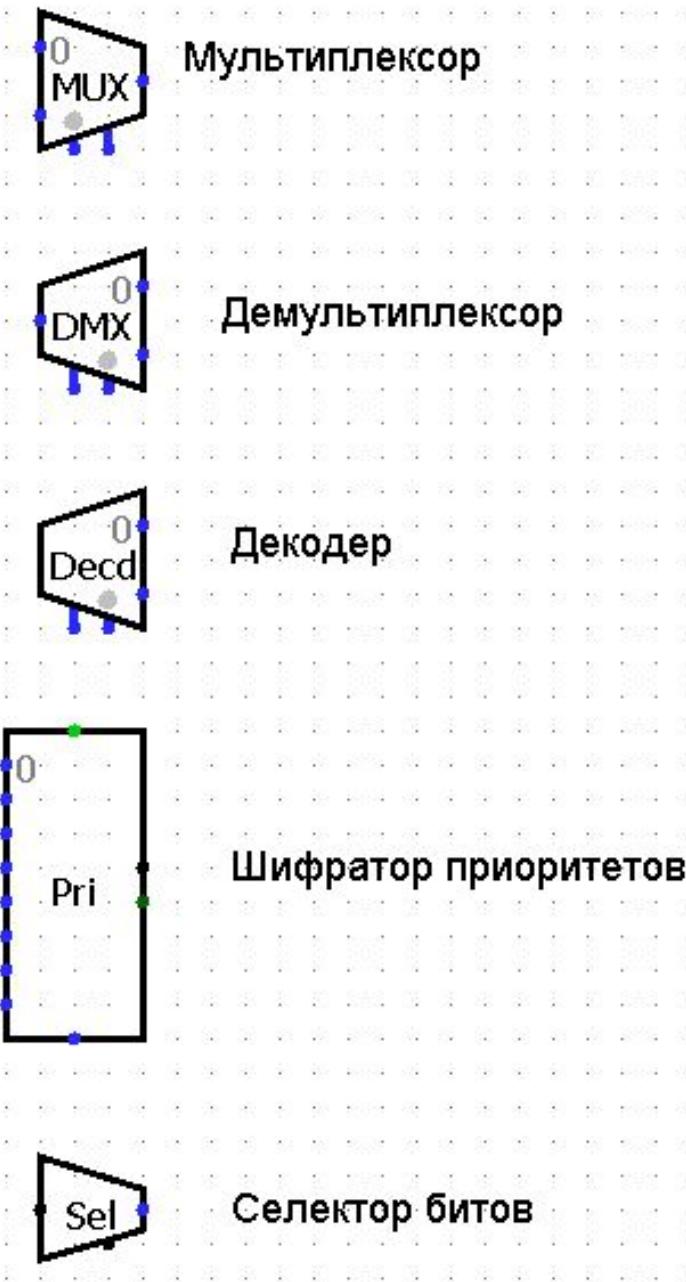
NOT:



Типы комбинационных логических схем

- Мультиплексор
- Демультиплексор
- Компаратор
- Кодер
- декодер
- Сумматор
- Вычитатель
-
- ALU (арифметико-логическая единица)

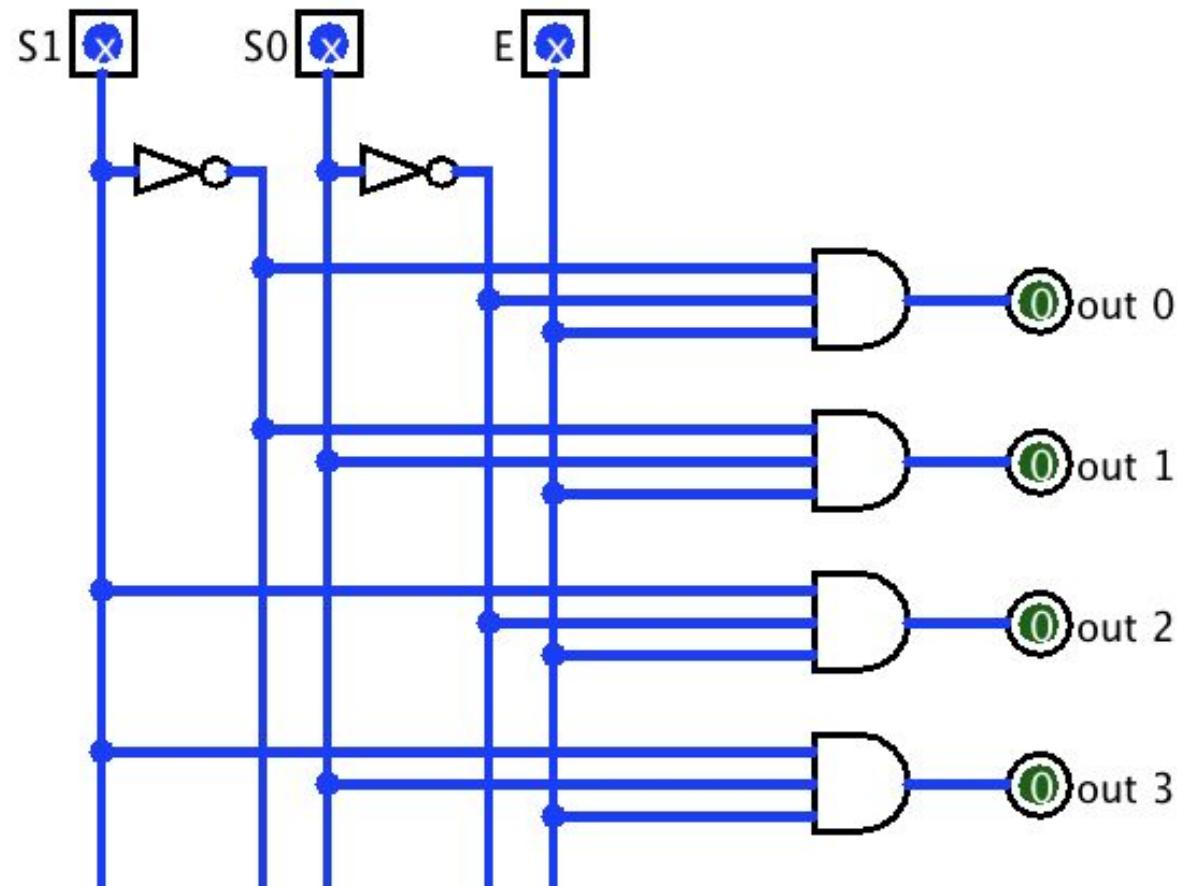
Плексоры Logisim



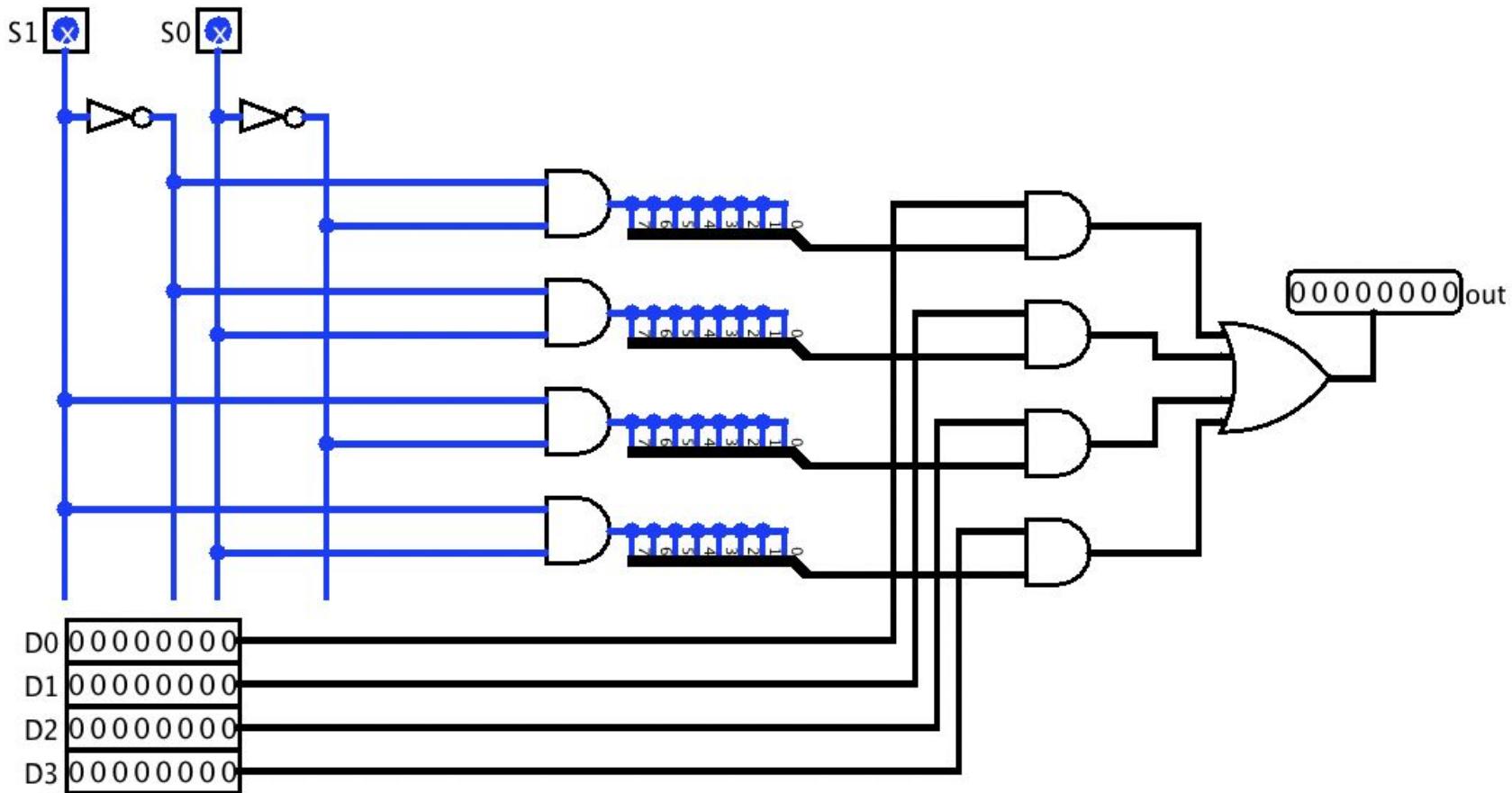
Popular combinatory units

- Decoder
 - k inputs, 2^k outputs.
 - Interprets inputs as k -bit number n , sets n -th output to 1, all others to 0
 - Standard decoders usually have “Enable” input. When Enable is down, all outputs are down
- Multiplexer
 - $2^k m$ -bit inputs and one k -bit selector input
 - Connects n -th input to the output

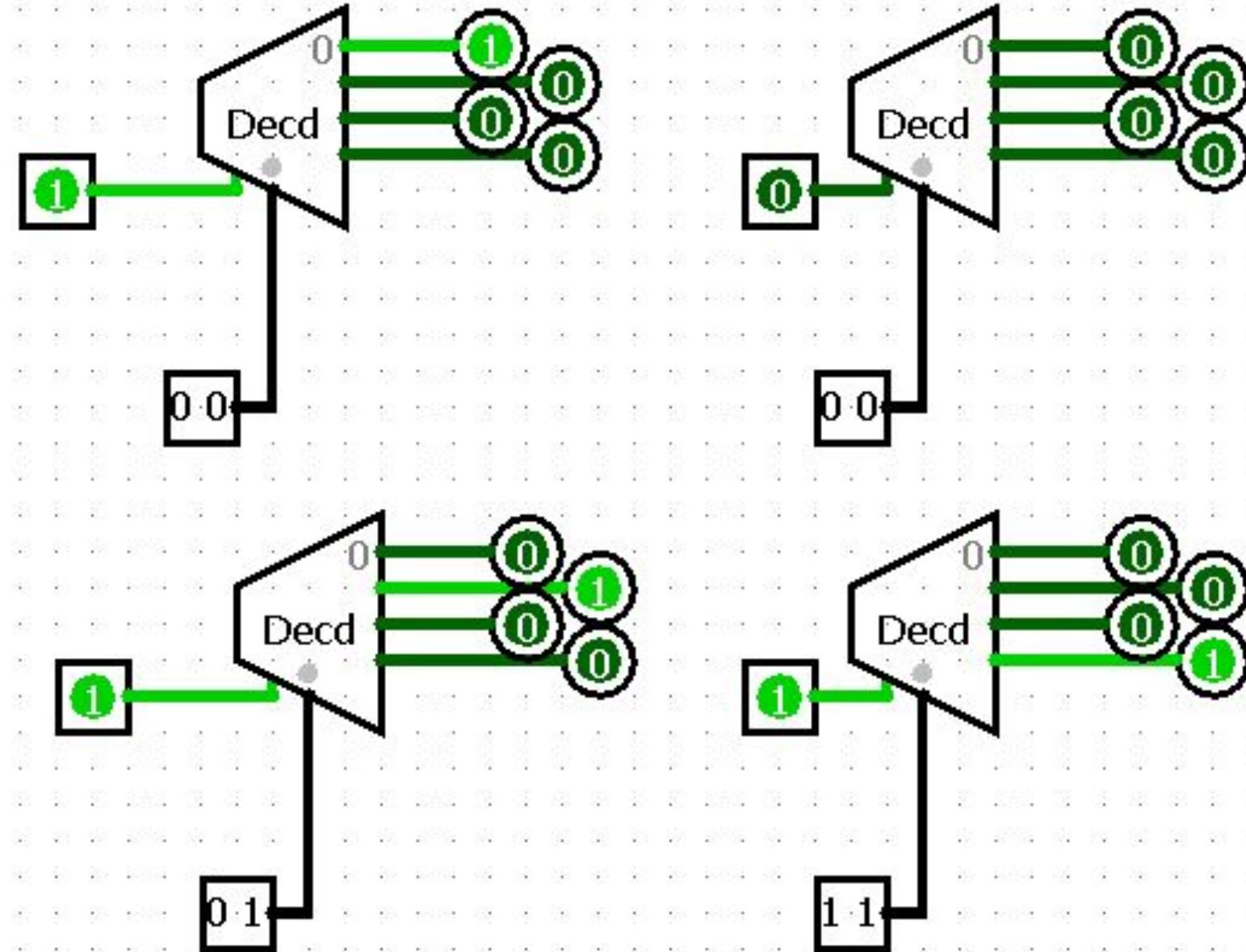
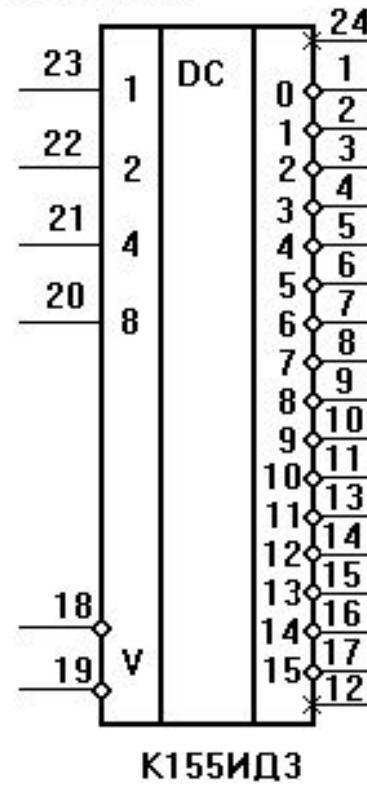
2x4 decoder



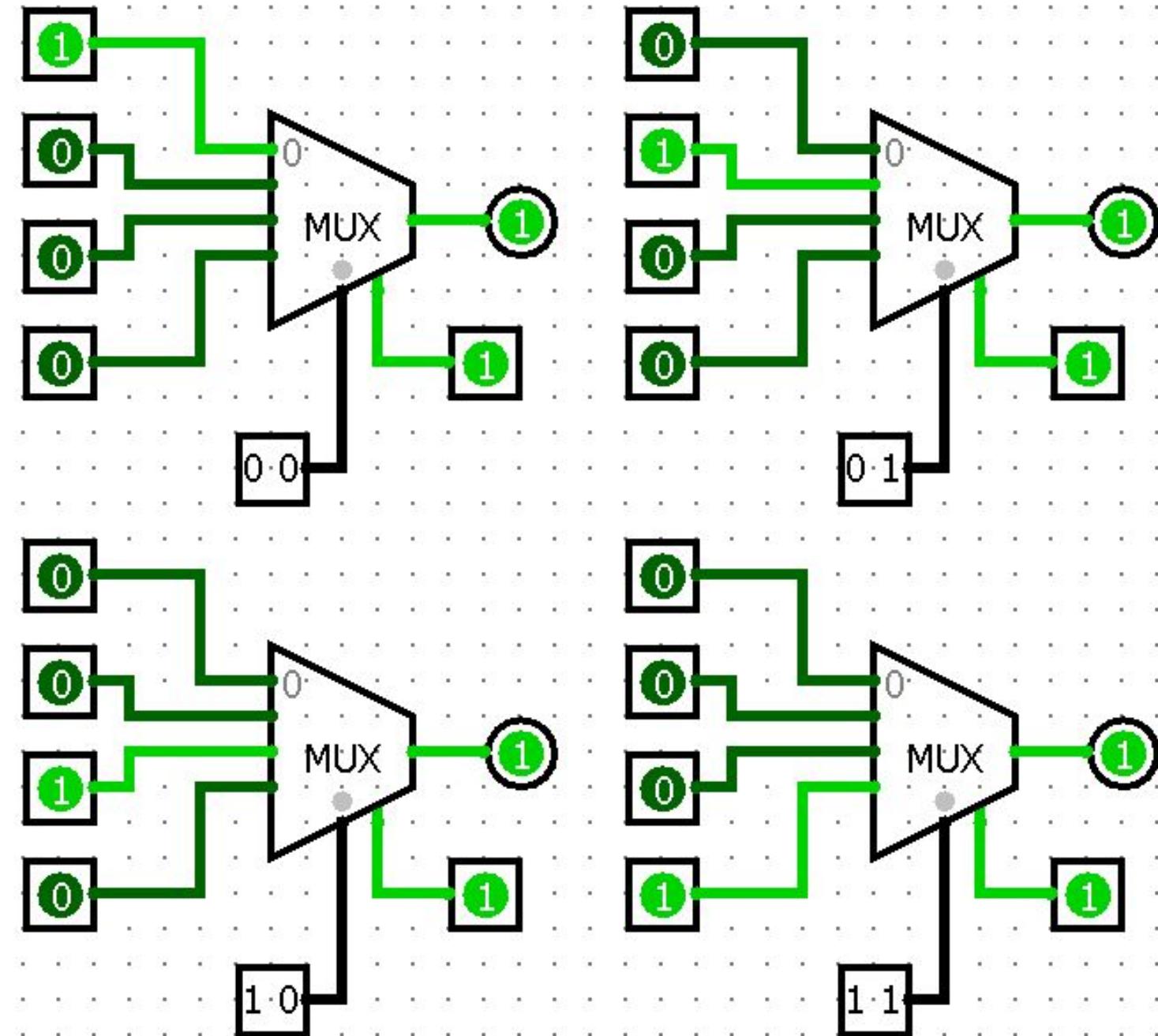
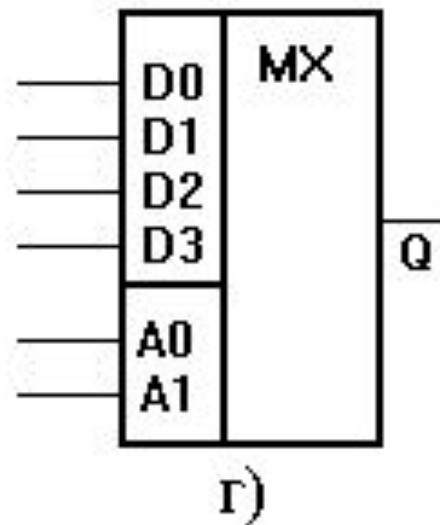
Multiplexer (mux)



Decoder



Multiplexer

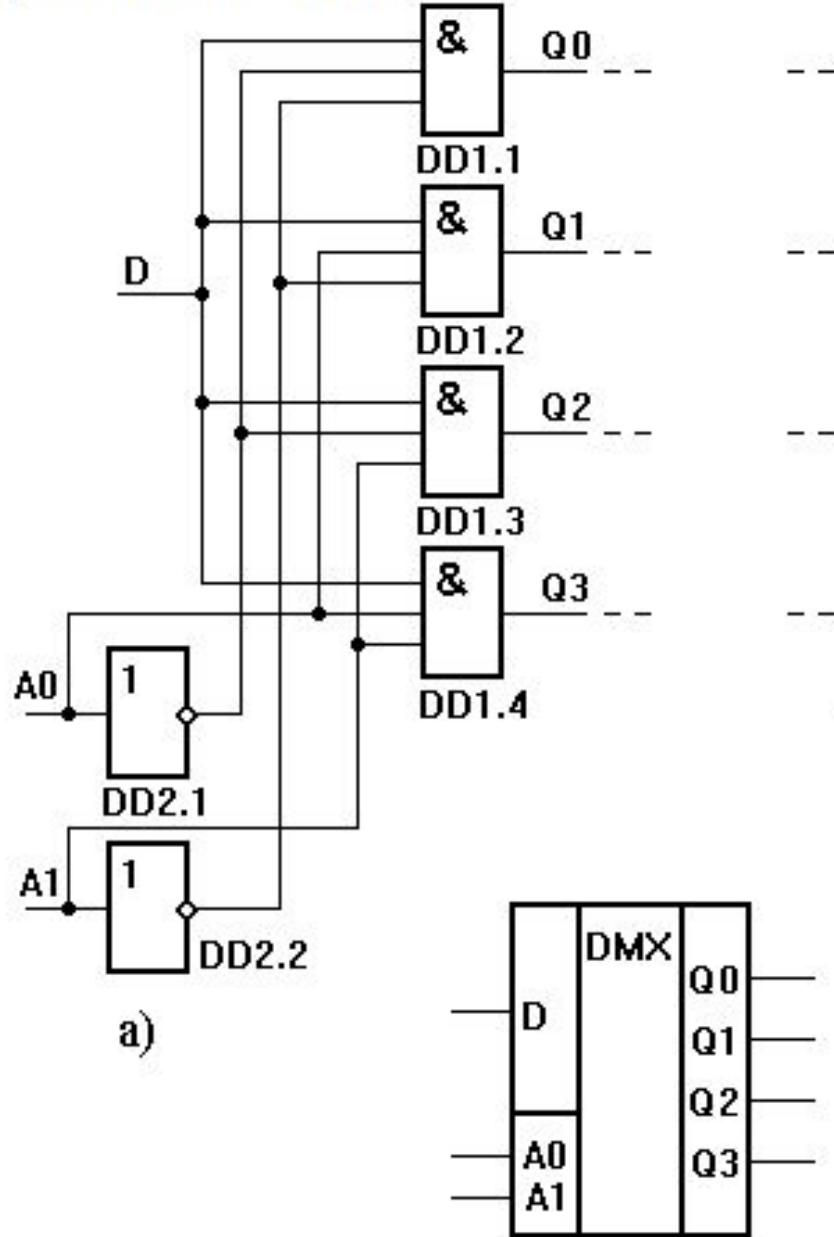


Demultiplexer

это устройство, обеспечивающее соединение одного из информационных выходов с одним входом.

Номер информационного выхода, который соединяется с входом, задается в двоичном коде на адресных входах.

Если демультиплексор имеет n адресных входов, то в нем может быть 2^n информационных выходов.

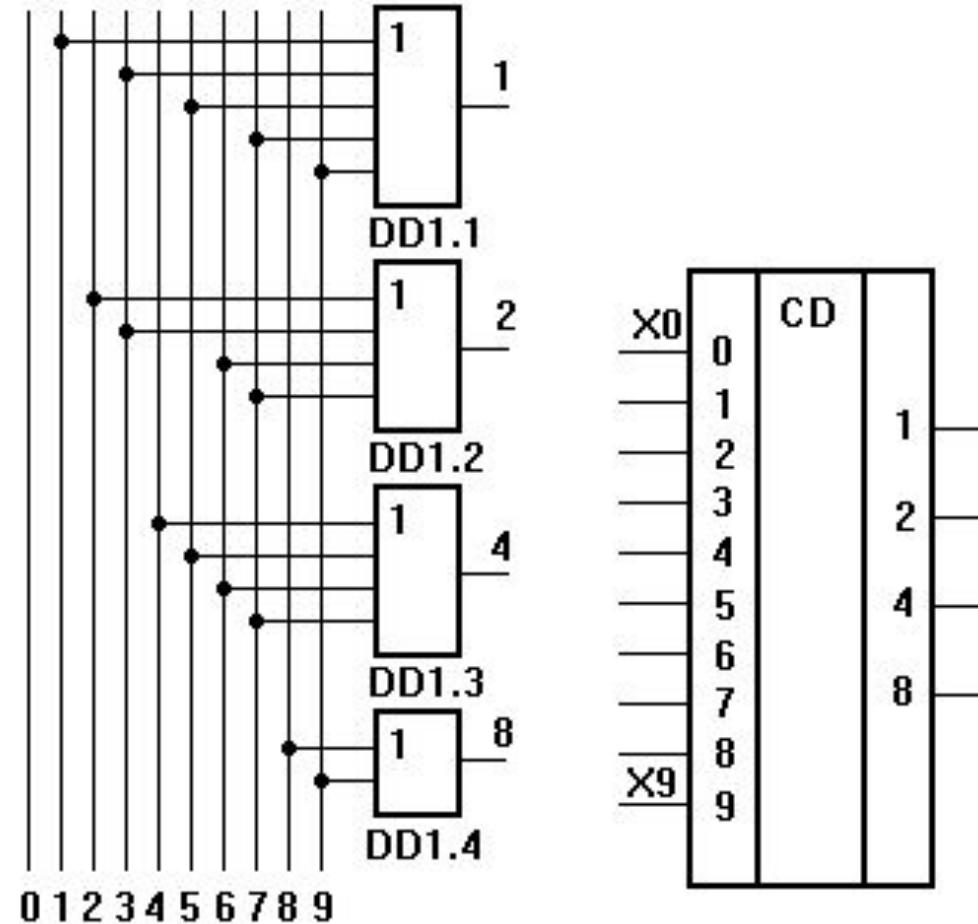


coder

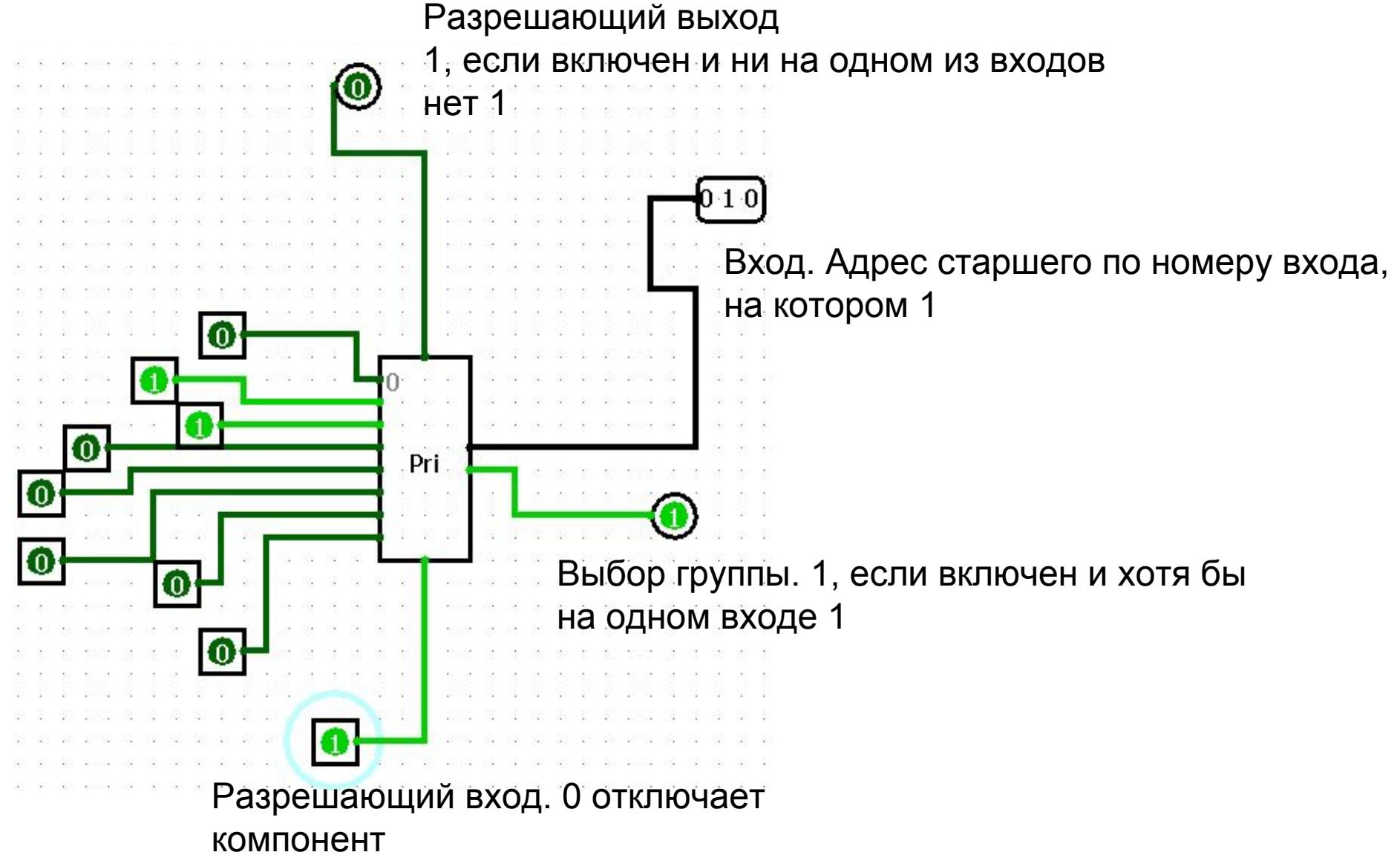
Шифратор (кодер) преобразует сигнал на одном из входов в n -разрядное двоичное число.

Функциональная схема шифратора, преобразующего десятичные цифры в 4-разрядное двоичное число, приведена на рисунке

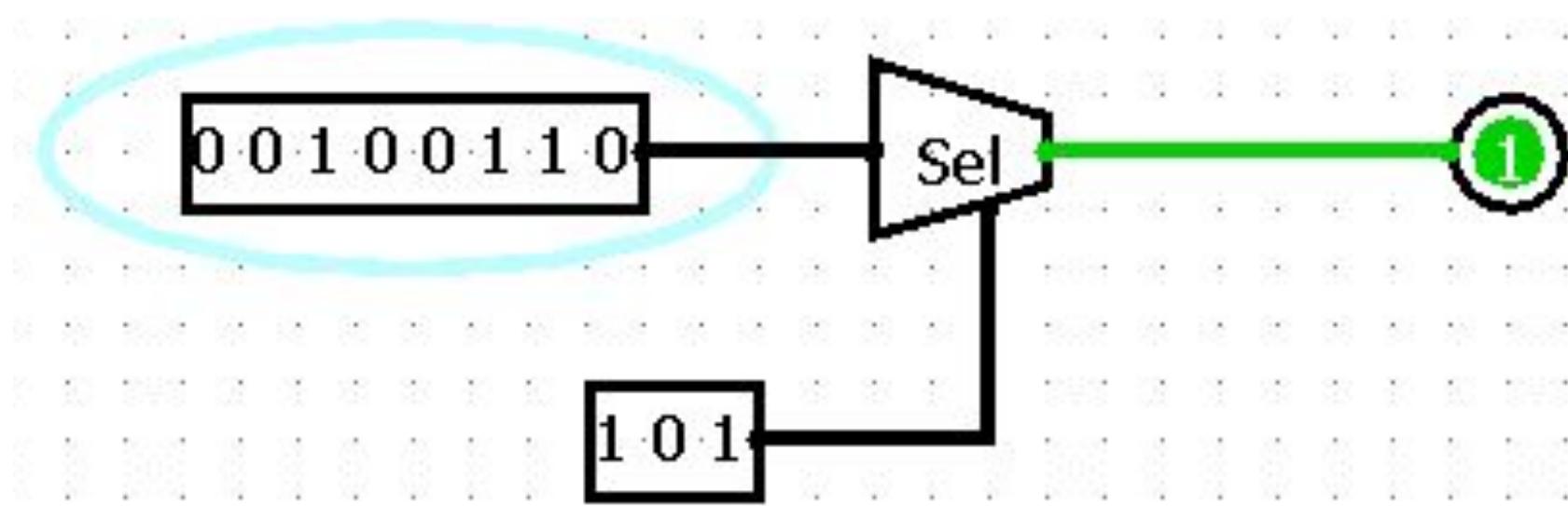
При появлении сигнала логической единицы на одном из десяти входов на четырех выходах шифратора будет присутствовать соответствующее двоичное число.



шифратор приоритетов

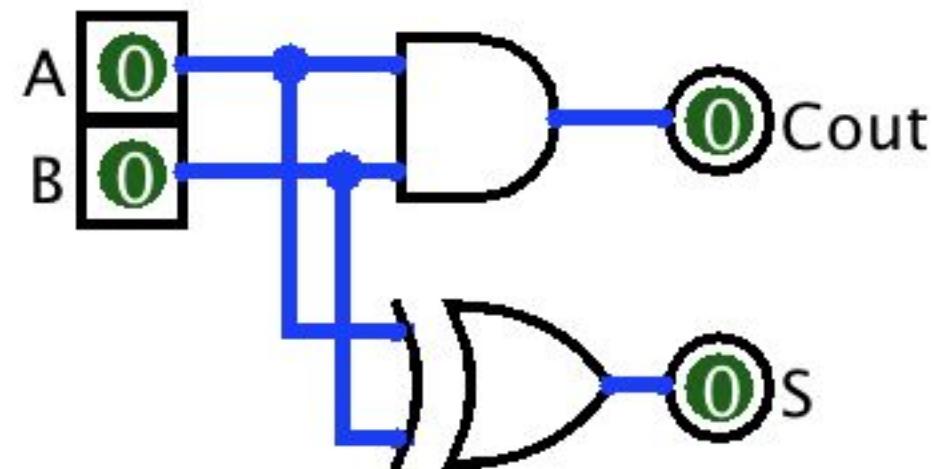


Селектор битов



So, the adder

- Two bit half adder
- Adds two bits and produces result and carry bit
- $S=AB$, $Cout=A \wedge B$



The full adder

- Has three inputs: A, B, Cin
- Also has two outputs: R and Cout
- Can be constructed from two half adders

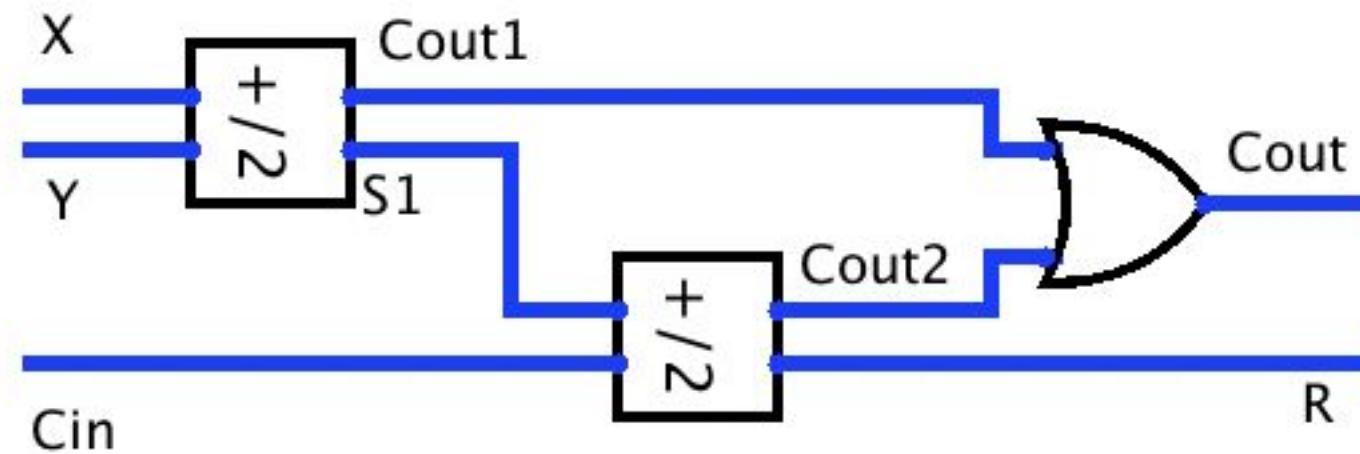
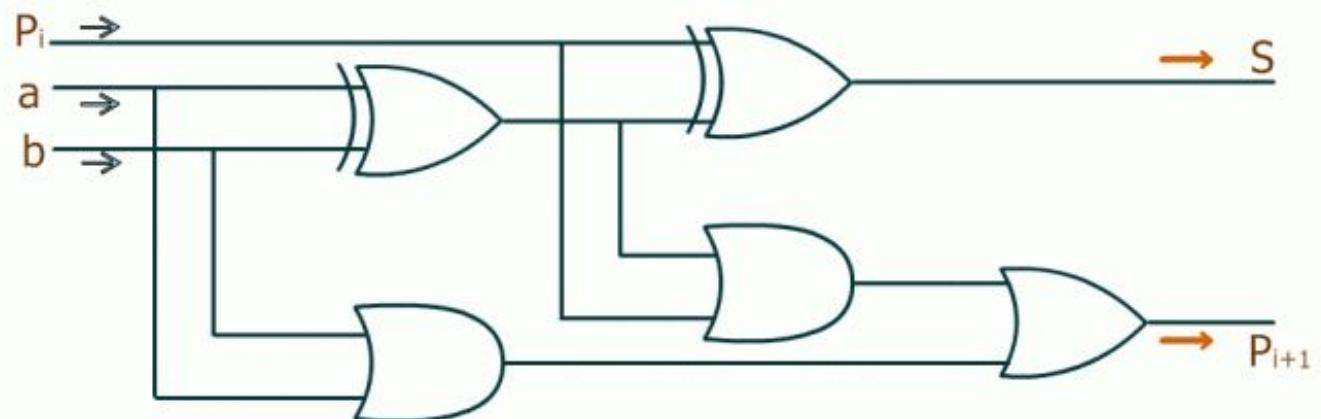


Таблица истинности для сумматора

Входы			Выходы	
a	b	P_i	S	P_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
0	1	1	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

a - первое слагаемое
 b - второе слагаемое
 S - сумма разряда
 P_i - перенос из младшего разряда
 P_{i+1} - перенос в старший разряд

Схема сумматора



a - первое слагаемое
 b - второе слагаемое
 S - сумма разряда
 P_i - перенос из младшего разряда
 P_{i+1} - перенос в старший разряд

Сумматор с переносом пульсации

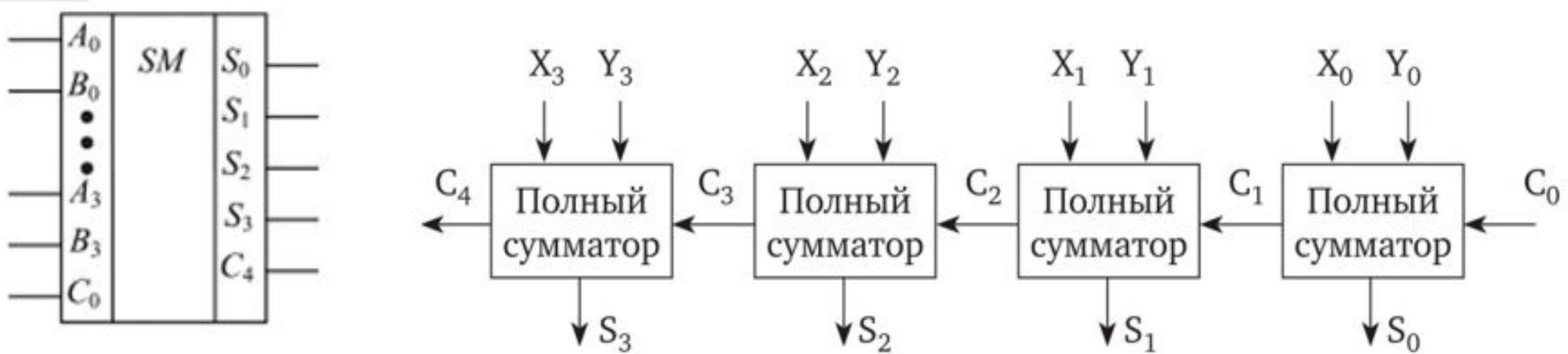
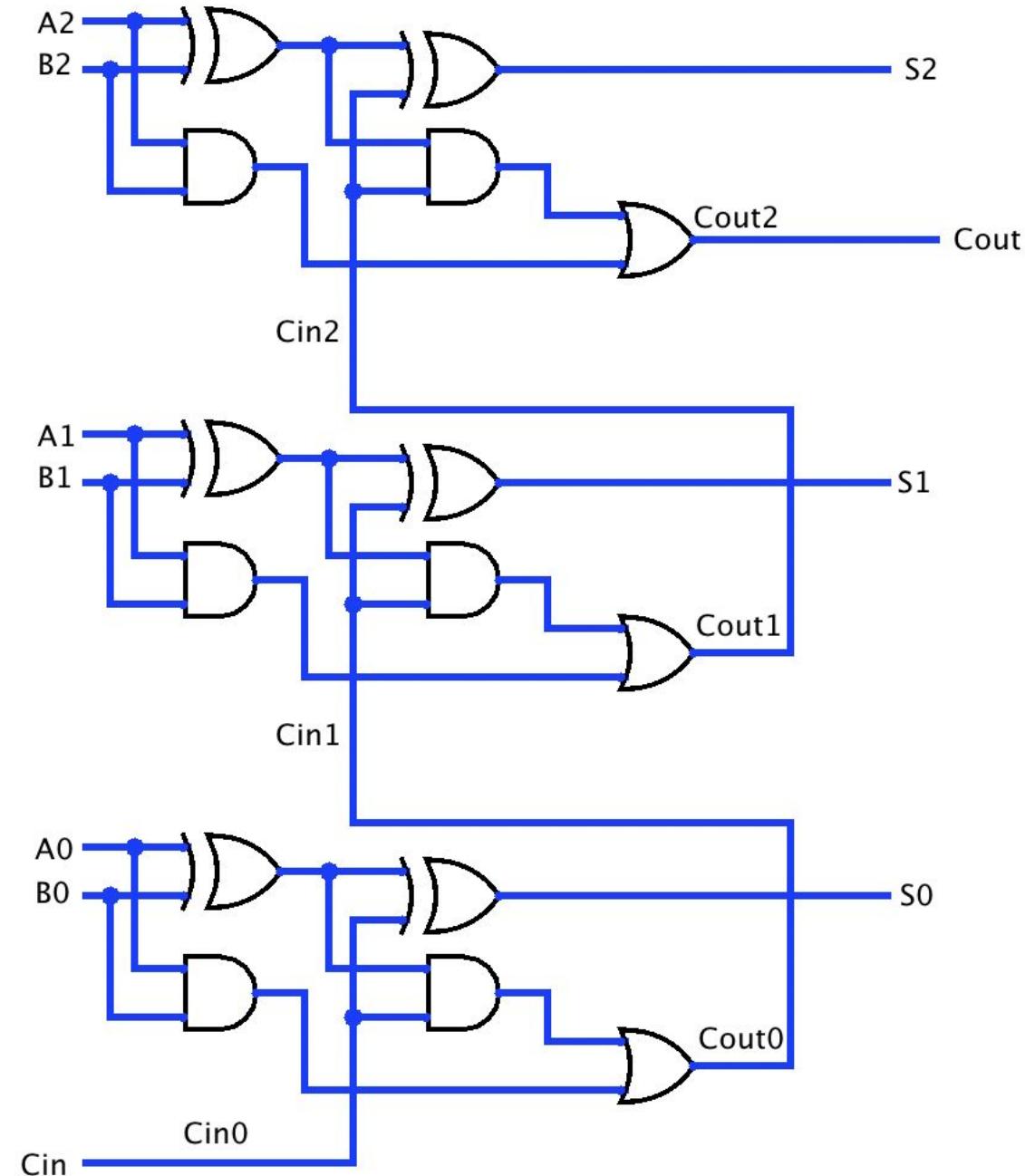


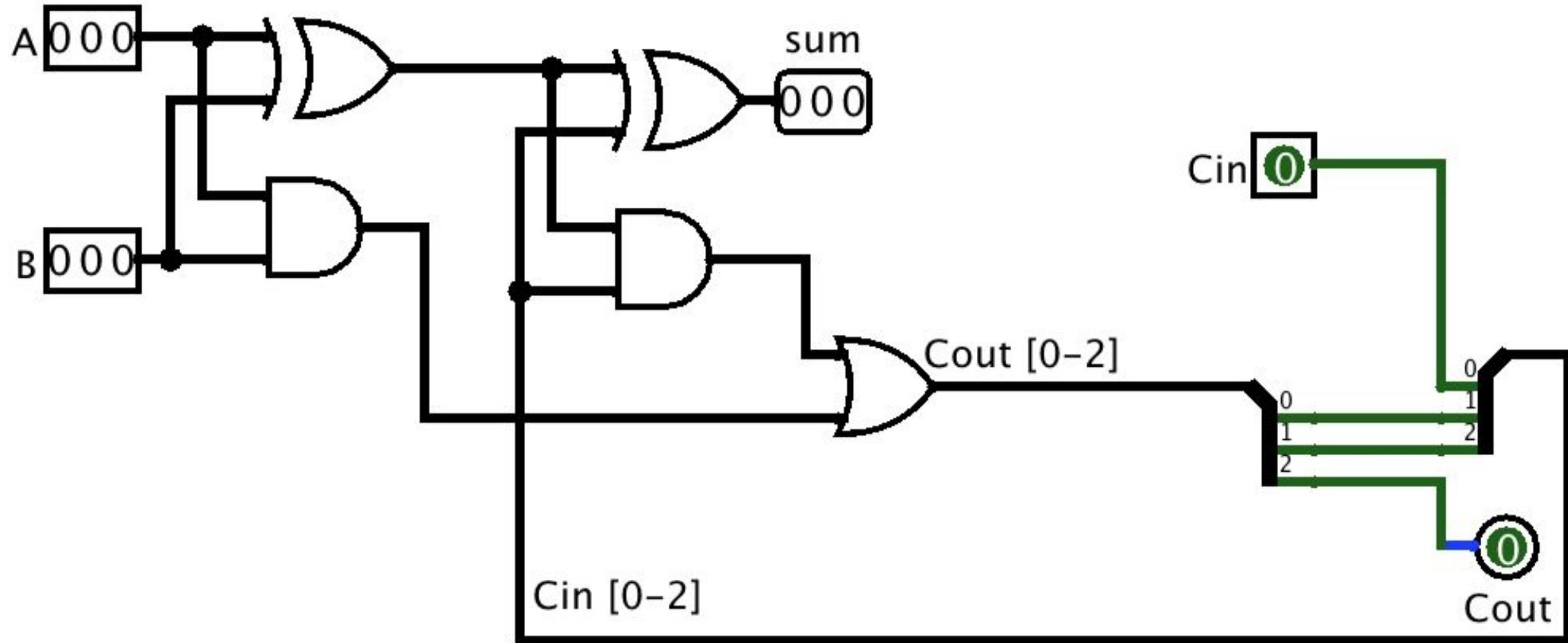
Рис. 4.7. Четырехразрядный двоичный сумматор

Three-bit adder

- Place three full adders in a row
- Connect Cout0 to Cin1, etc
- Sounds simple and good enough for Logisim
- For real devices, carry must propagate from bit 0 to N
- This design is known as ripple carry adder
- Delay $\approx O(N)$
- We discuss this later

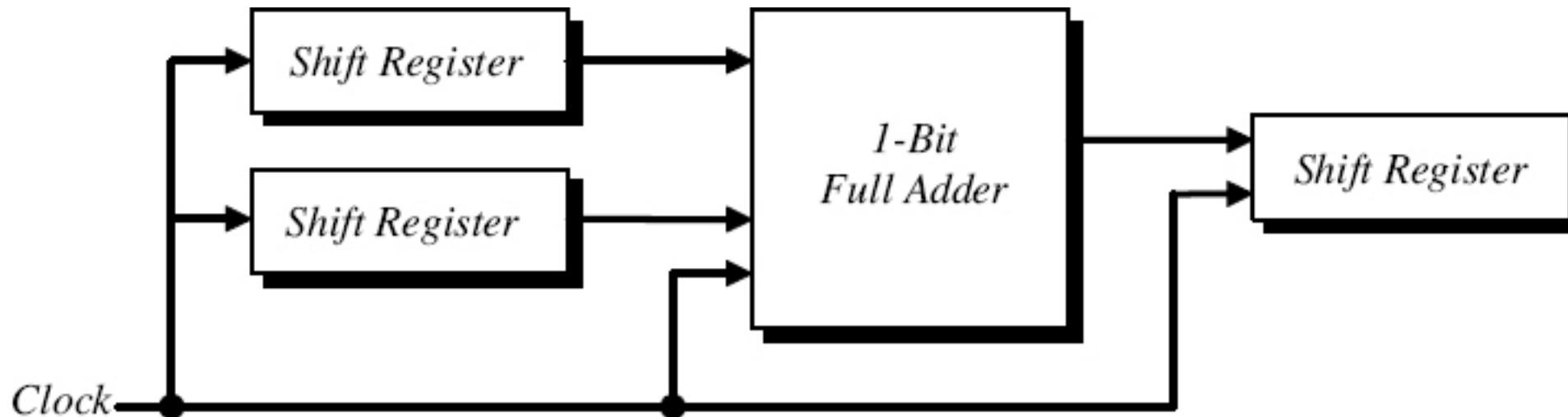


Logisim notation for repeating circuits



How to deal with carry propagation delay?

- A sequential adder.
 - Carry propagation time for N-bit adder $\approx N$ (delay of single adder)
 - Why bother building N adders if they cannot actually work in parallel?

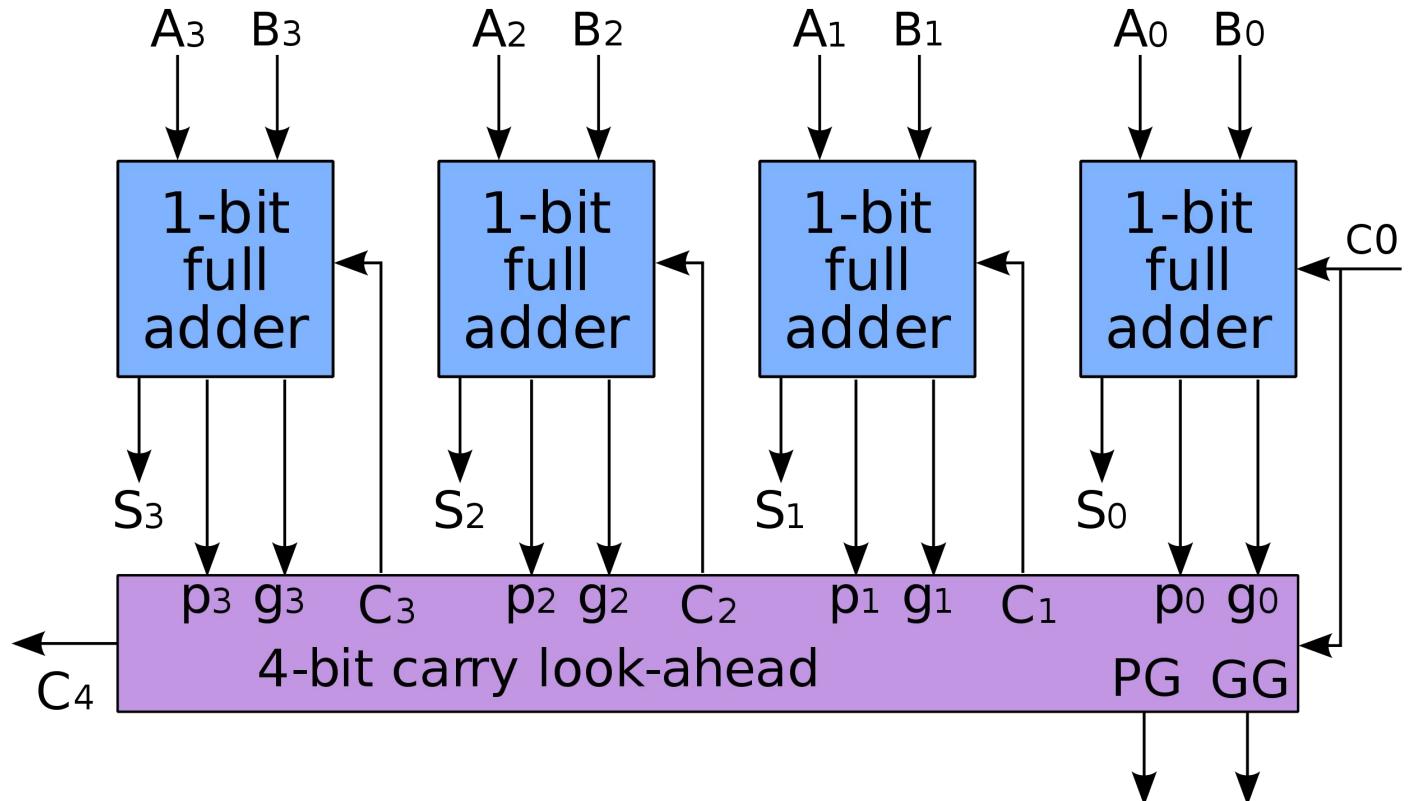


Carry-save adder

- Result is provided as two bits (carry not propagated)
- $1011101010101101111000000001101 + 1101111010101101101111011101111 = 211221202022022122111011102212$
- To produce binary number, we still need to propagate carry
- But we can add these saved-carry numbers without converting to binary
- And propagate only on last stage
- Carry-save adders are used in multiplicators, in cryptography, etc

Carry-lookahead (Сумматор с упреждающим переносом)

- Instead of Cout, every adder provides two signals, P (propagate) and G (generate)
- $P(A,B)=AvB$
- $G(A,B)=A^B$
- $C_{i+1}=G_i \vee (P_i \wedge C_i)$
- Actually, a recursive formula
- $C_4 = G_3 \vee (P_3 \wedge G_2) \vee (G_2 \wedge P_2 \wedge P_3) \vee (G_0 \wedge P_1 \wedge P_2 \wedge P_3) \vee (C_0 \wedge P_0 \wedge P_1 \wedge P_2 \wedge P_3)$
- Gate count $O(N^2)$, but delay close to constant

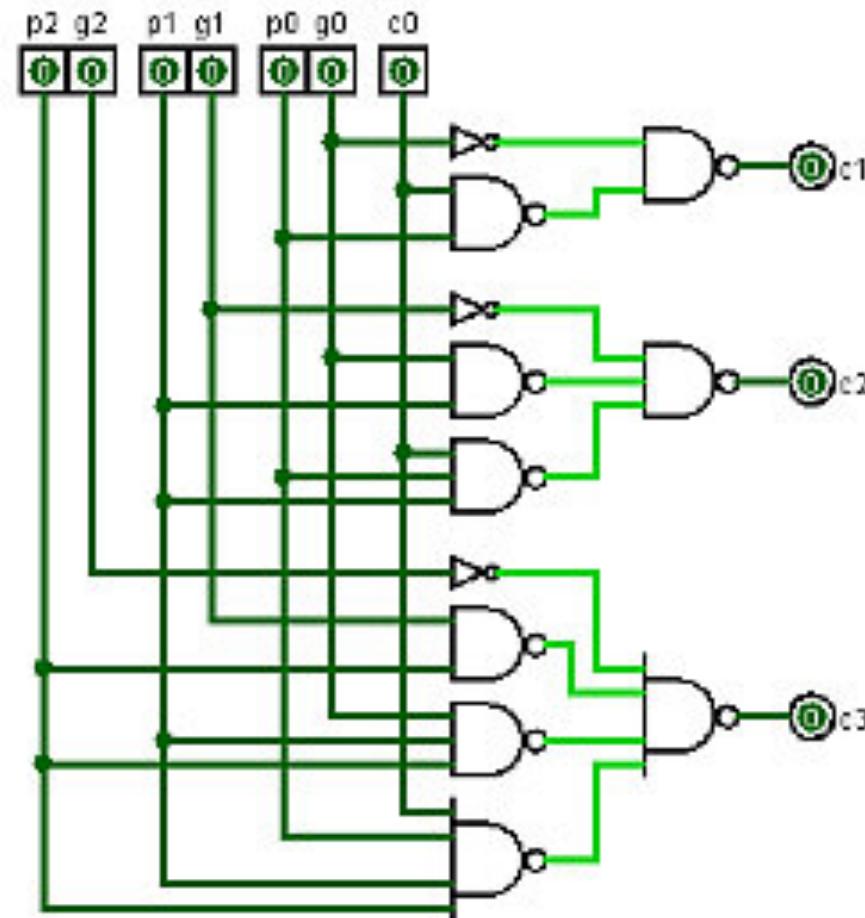


Функция генерации принимает единичное значение, если перенос на выходе данного разряда появляется независимо от наличия или отсутствия входного переноса. Очевидно, что эта функция $G(A,B)=A \wedge B$

Функция прозрачности (транзита) принимает единичное значение, если перенос на выходе данного разряда появляется только при наличии входного переноса. Эта функция $P(A,B)=AvB$

4-bit carry lookahead in Logisim

- Delay depends on number of gates the signal passes between input and output
- *NOT* on total number of gates
- In this design, C3 produced from 7 input signals, but each of these signals pass only two gates
- So, we can improve speed by *increasing* number of gates
- In computing, simpler is not always better



Ways to produce multibit adder

- Often, a combination of ripple carry and carry lookahead is used
- 16-bit adder can be implemented from 4-bit units as:
 - ripple-carry connected via carry-lookahead
 - carry lookaheads connected via ripple carry
 - carry lookaheads connected via second level carry lookahead
- Other tricks, like carry-skip (aka carry bypass) adder
 - Read the Wikipedia...

