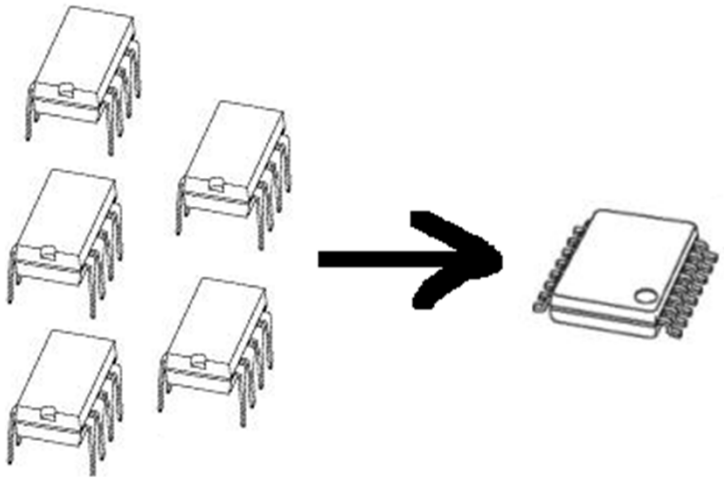
Programmable Logic

- Programmable Array Logic (PAL)
- Read Only Memory (ROM)
- Field Programmable Gate Array (FPGA)

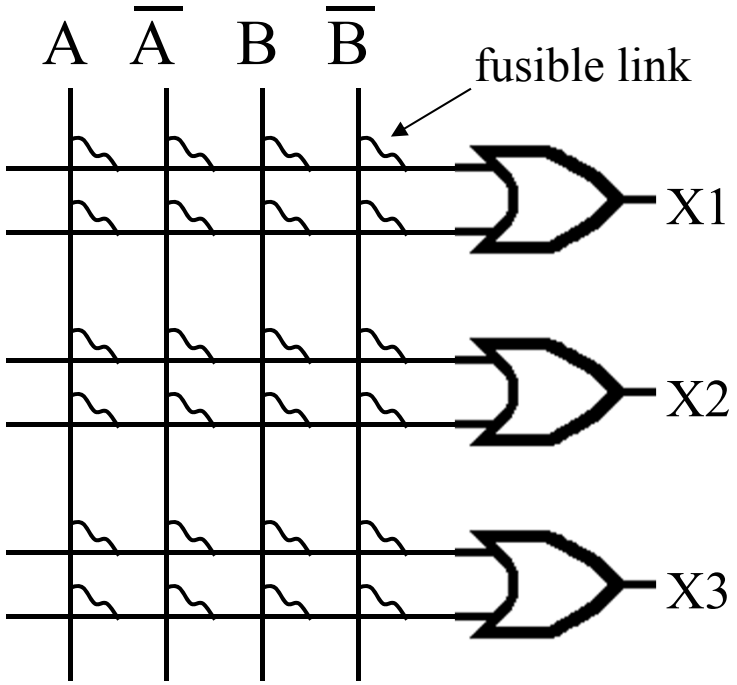
Programmable Logic Devices (PLDs)

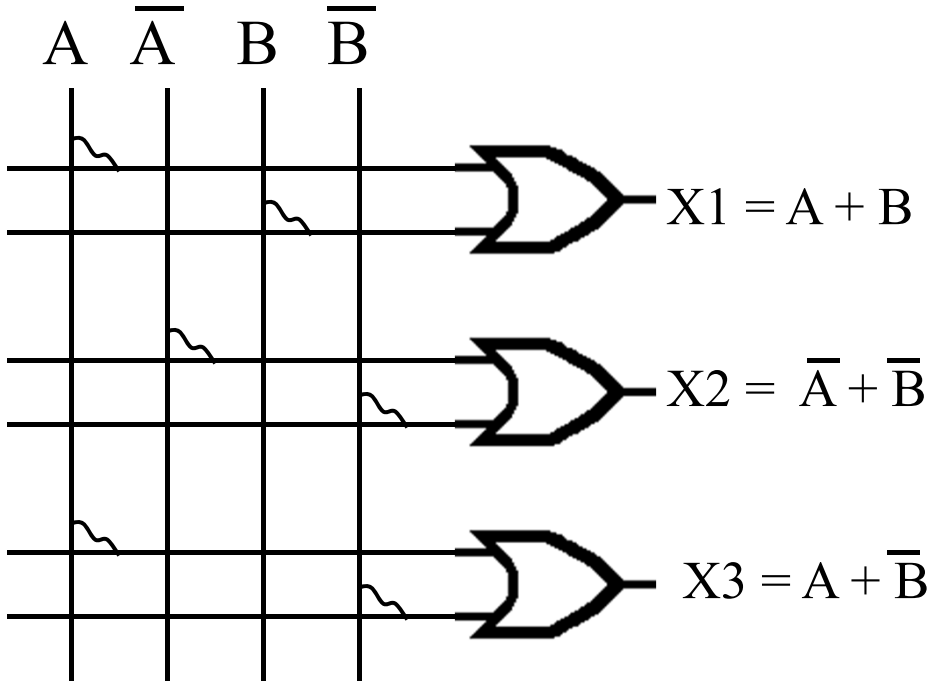
PLDs are used in many applications to replace SSI and MSI circuits. They save space and reduce the component count. Current devices typically contain hundreds to millions of gates.



Programming the device allows it to be configured to a desired logic function.

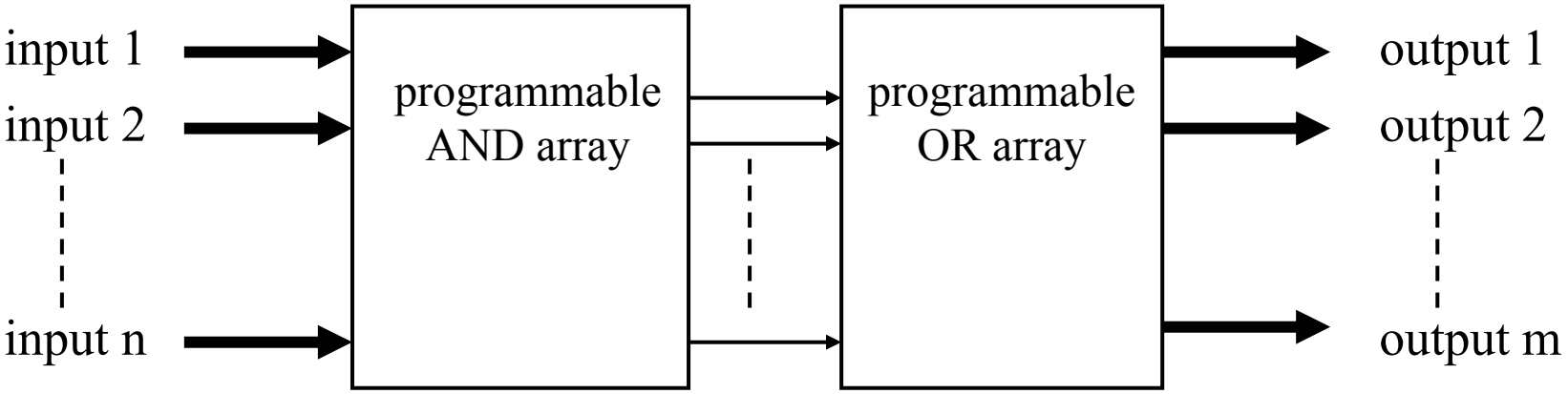
They contain a grid of conductors that form rows and columns, with a fusible link at each cross point. Signal paths are made by blowing the fuses on the connections that are not required.



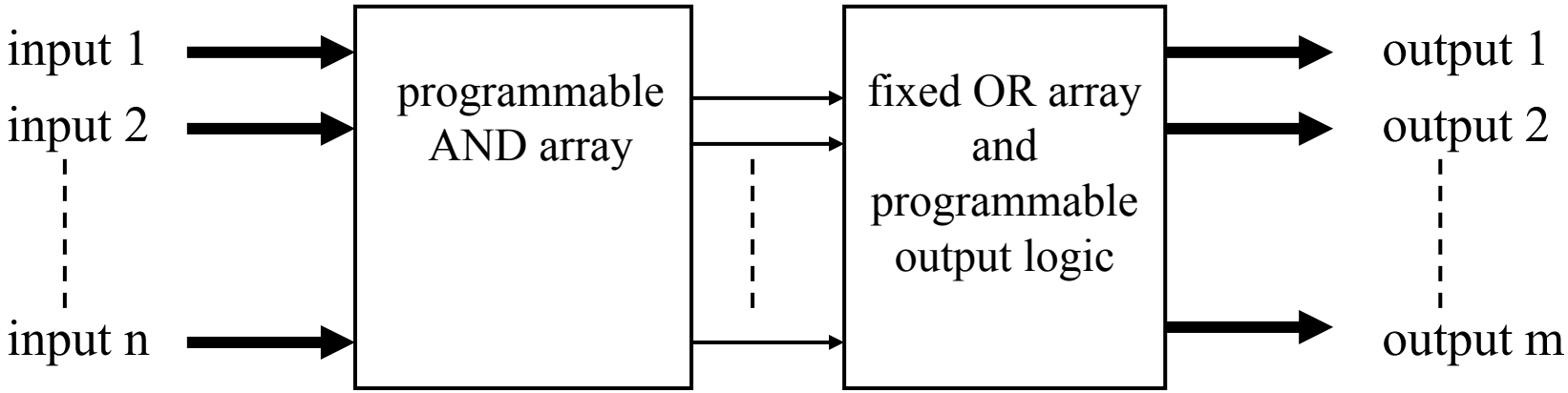


Classification of PLDs

PLA - Programmable Logic Array



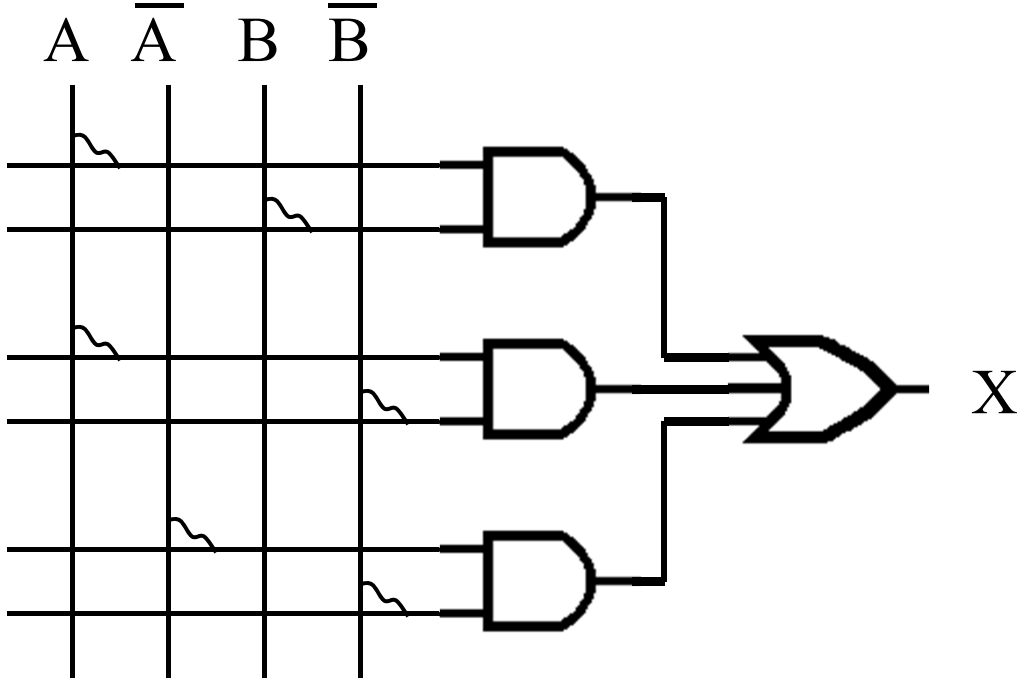
PAL - Programmable Array Logic



Programmable Array Logic (PAL)

The PAL contains a one-time programmable array of AND gates that connects to a fixed array of OR gates.

Each fused cross point is called a **cell** and is the programmable element of a PAL.

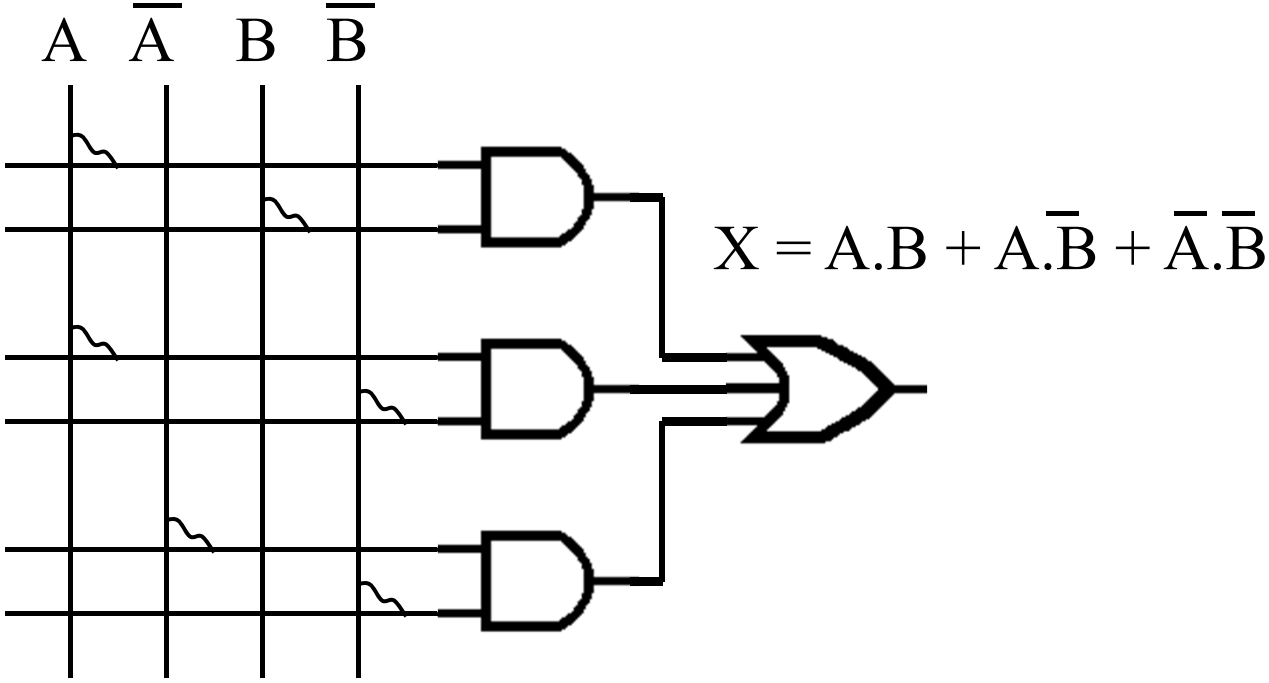


This structure allows any sum-of-products (SOP) expression with a defined number of variables to be implemented.

Programmable Array Logic (PAL)

The PAL contains a one-time programmable array of AND gates that connects to a fixed array of OR gates.

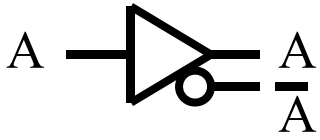
Each fused cross point is called a **cell** and is the programmable element of a PAL.



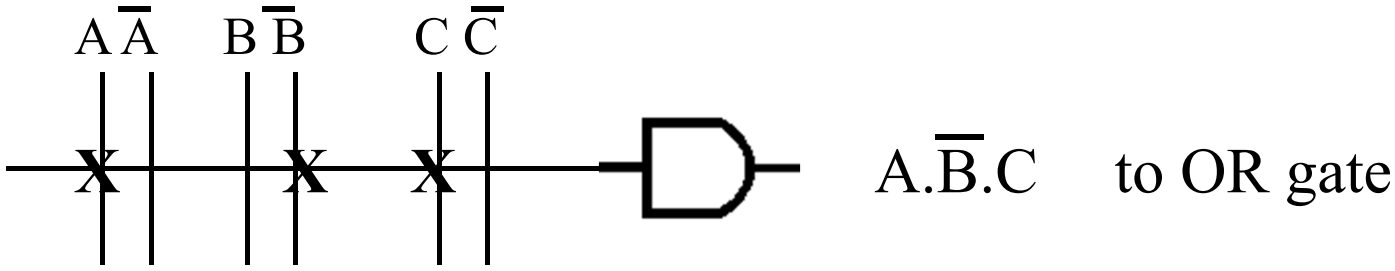
This structure allows any sum-of-products (SOP) expression with a defined number of variables to be implemented.

PAL Operation

Real PALs have many AND gates and OR gates in the array. The inputs are buffered to provide a connection for the variable and its complement.



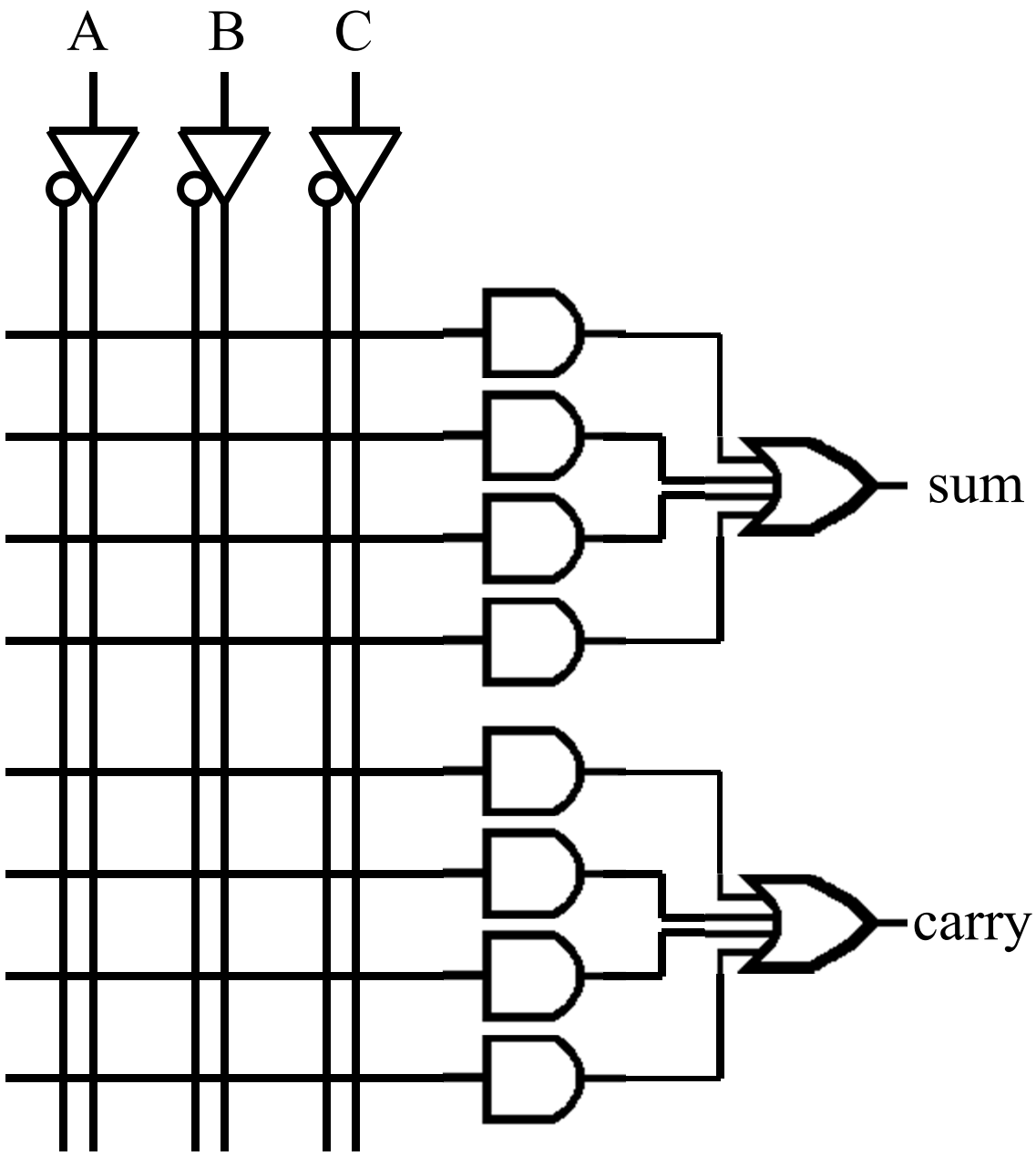
To simplify the logic diagram, a single line is used for multiple input AND gates with a cross to represent a connection.



An XOR gate may be used as a controlled inverter on the output stage to provide both the True and Complement of the output.

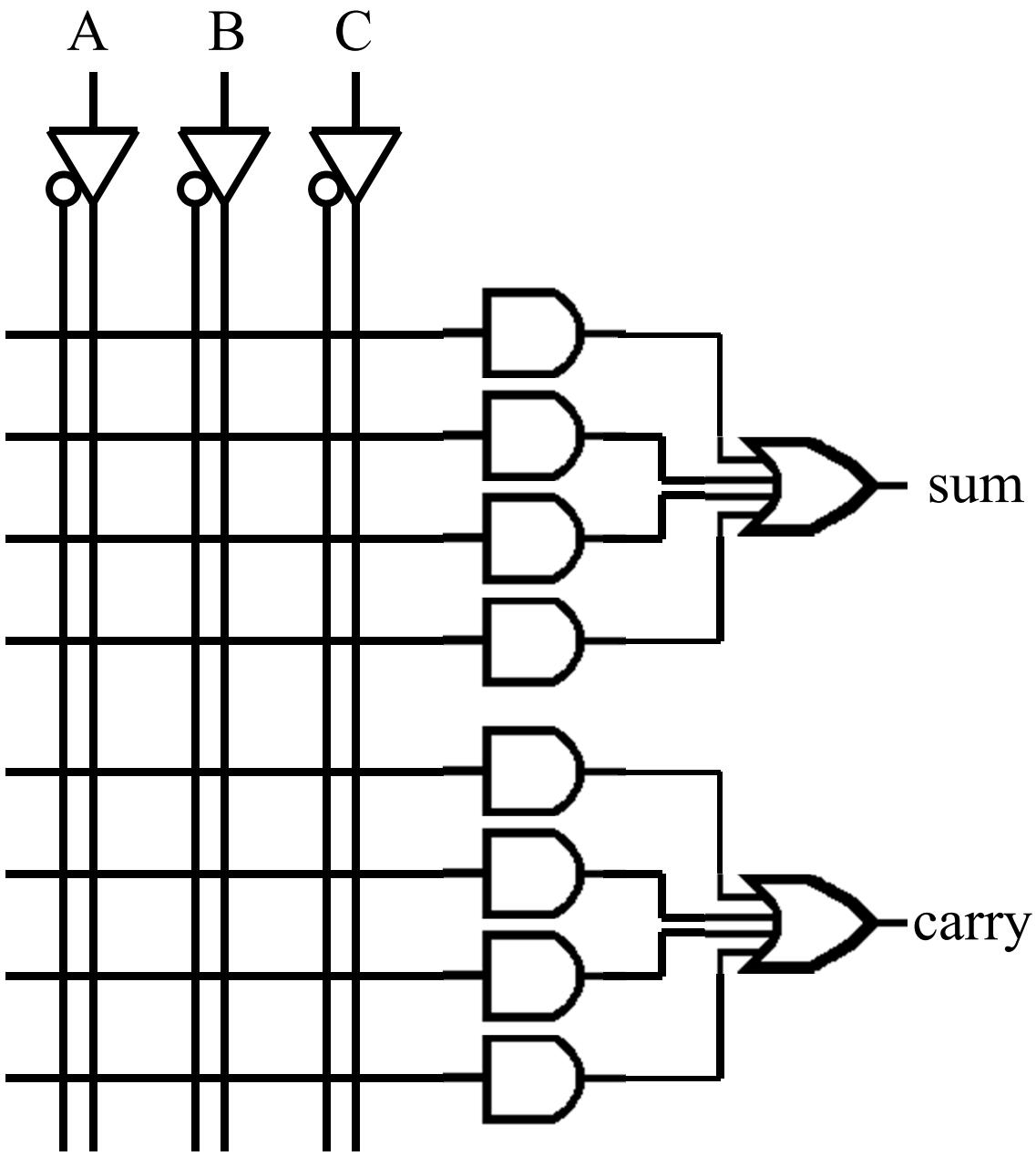
Show the PAL programming connections required to implement the sum and carry outputs for a full adder.

A	B	C	C_o	S_o
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



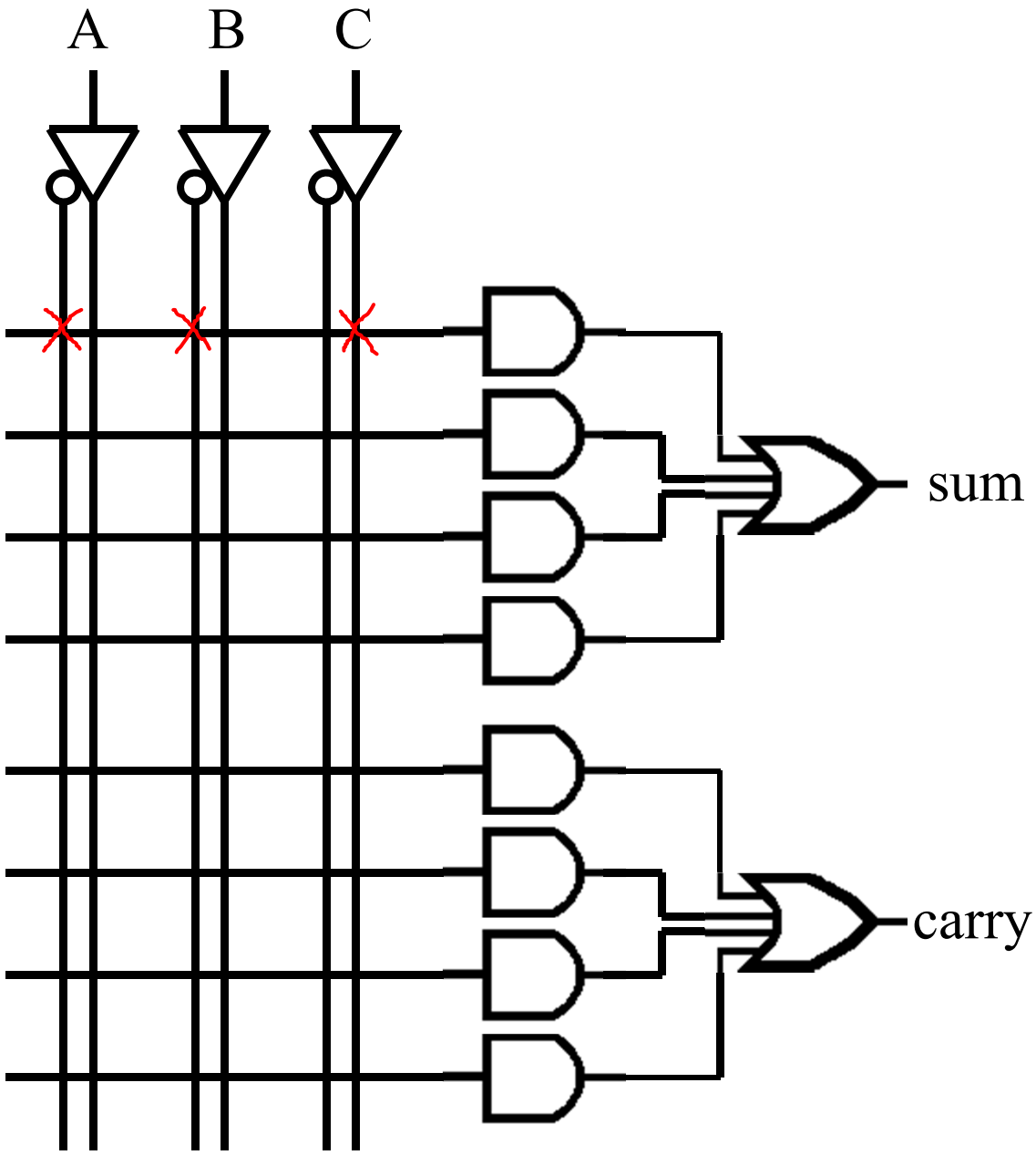
Show the PAL programming connections required to implement the sum and carry outputs for a full adder.

A	B	C	C_o	S_o
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



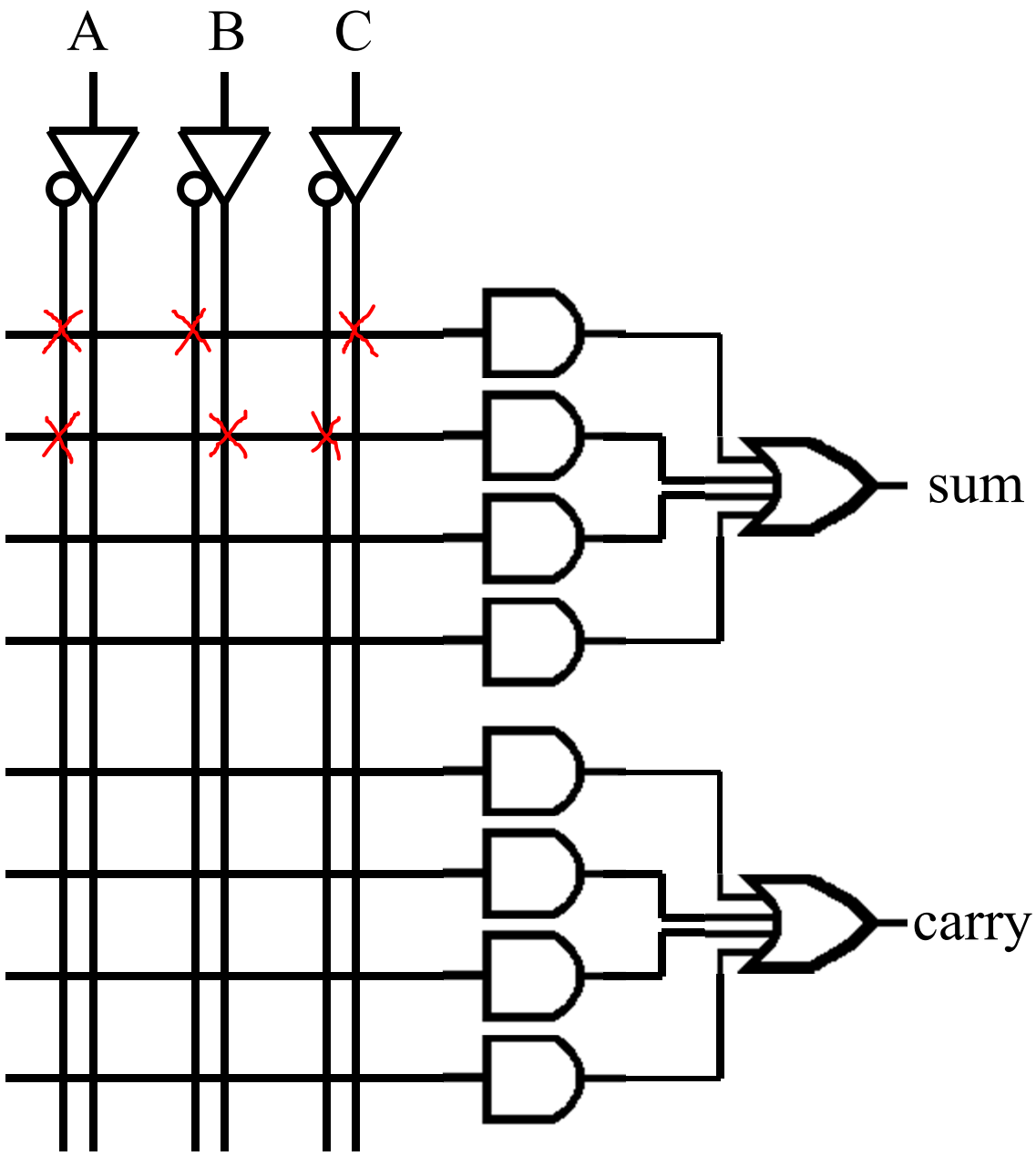
Show the PAL programming connections required to implement the sum and carry outputs for a full adder.

A	B	C	C_o	S_o
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



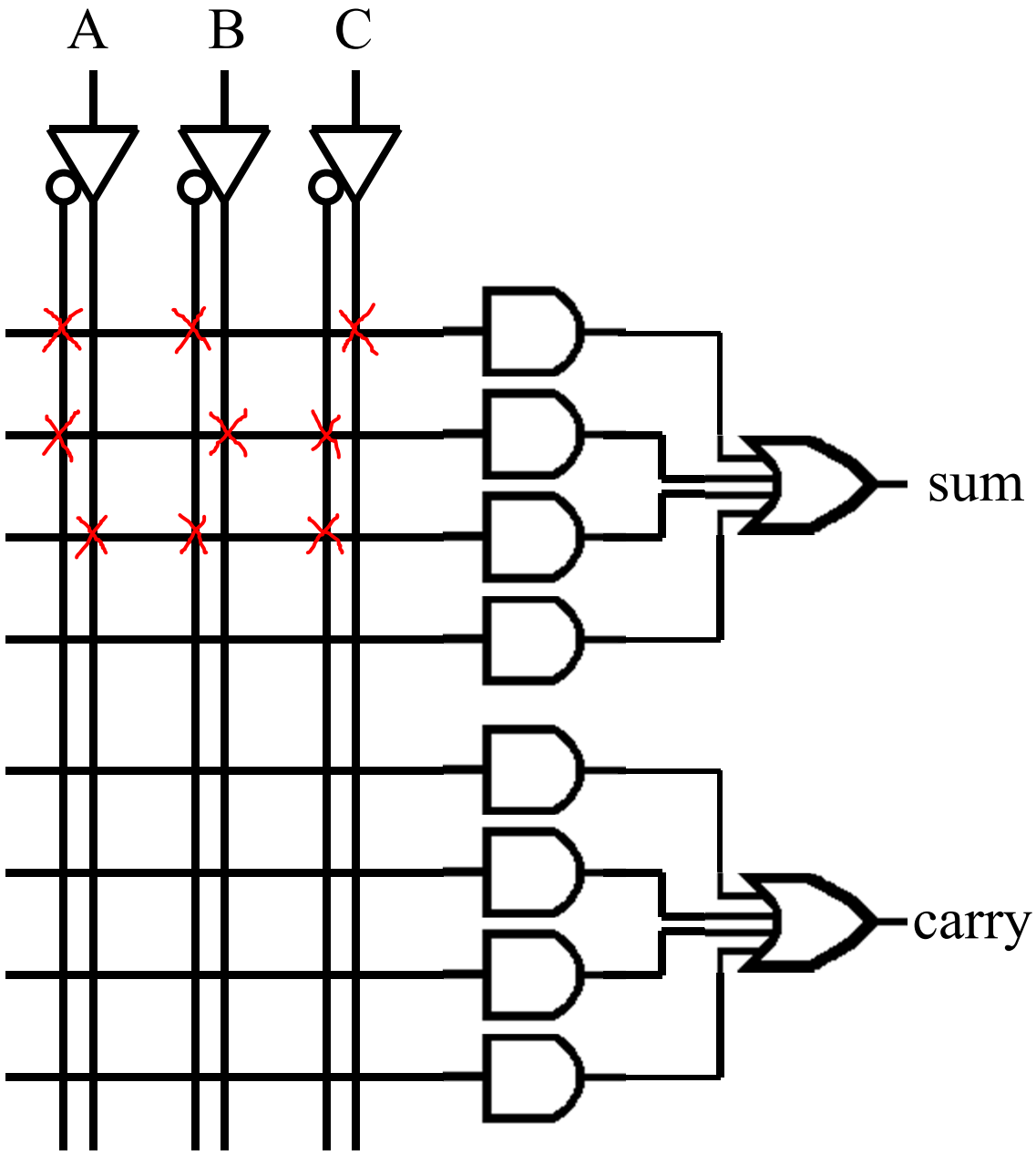
Show the PAL programming connections required to implement the sum and carry outputs for a full adder.

A	B	C	C_o	S_o
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



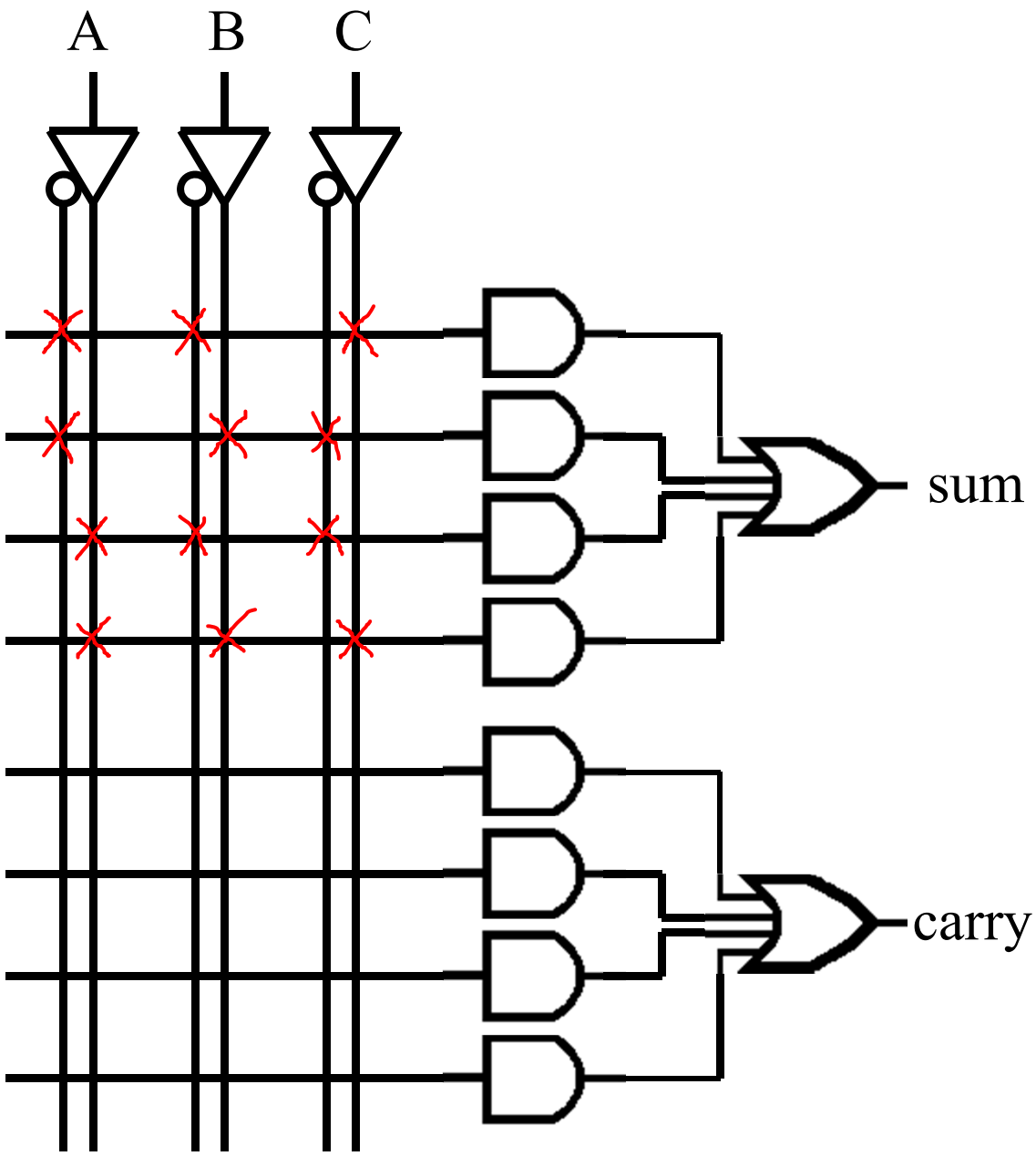
Show the PAL programming connections required to implement the sum and carry outputs for a full adder.

A	B	C	C_o	S_o
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



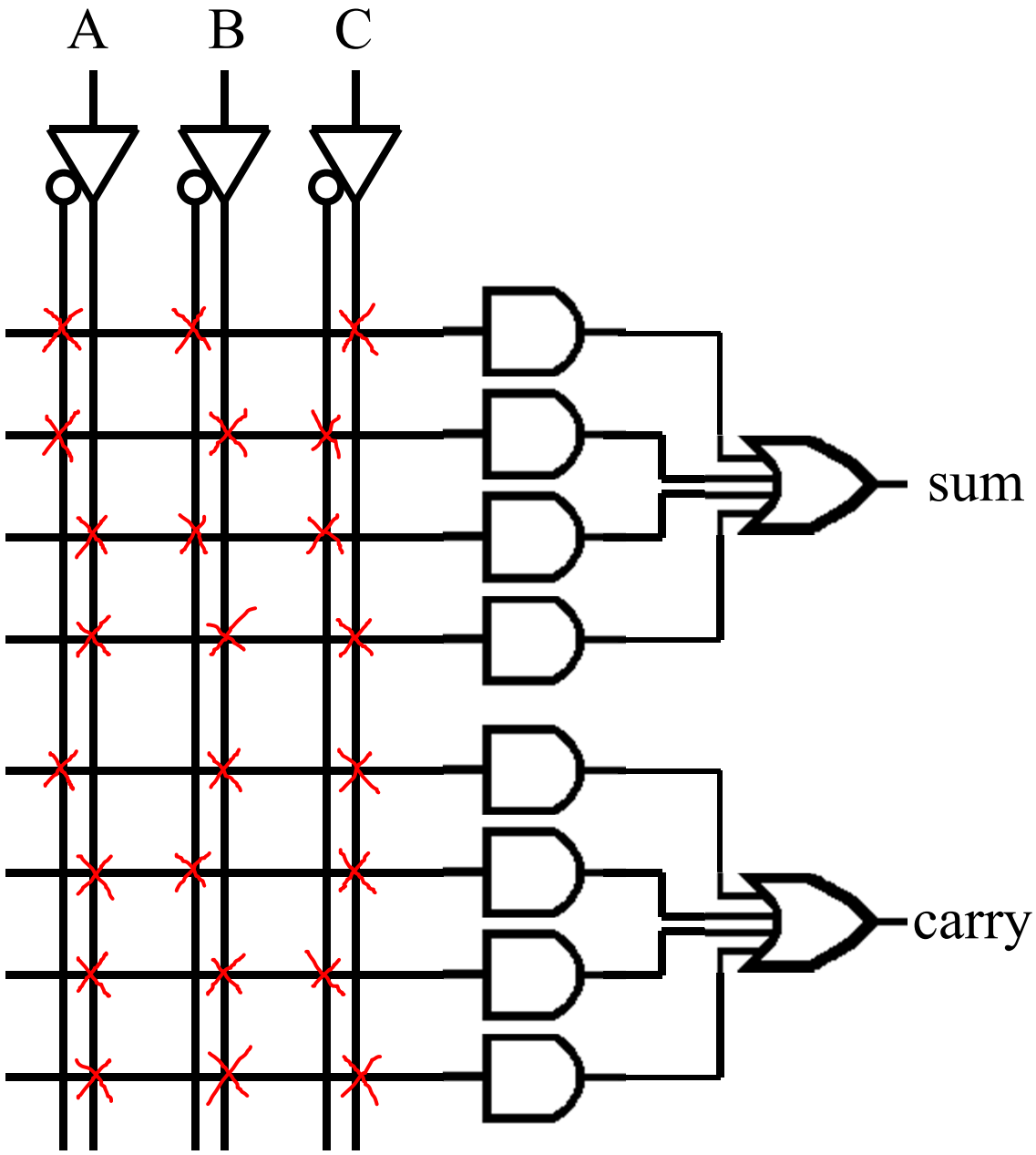
Show the PAL programming connections required to implement the sum and carry outputs for a full adder.

A	B	C	C_o	S_o
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Show the PAL programming connections required to implement the sum and carry outputs for a full adder.

A	B	C	C_o	S_o
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Read Only Memories (ROM)

ROM is a type of memory in which permanent binary information is stored. This information is specified by the designer and is then programmed into the device. ROM retains data when the power is turned off. This is known as nonvolatile memory.

Mask ROM

Permanently programmed during manufacture. Used to store tables of data or standard functions. Once programmed, cannot be changed.

PROM

Uses fusible links which are burned open or left intact to represent a ‘0’ or ‘1’. Programming is achieved by passing sufficient current through the link and is irreversible.

EPROM

Data can be erased by UV light in the case of an EPROM, electrically in the case of an EEPROM. The device can then be reprogrammed.

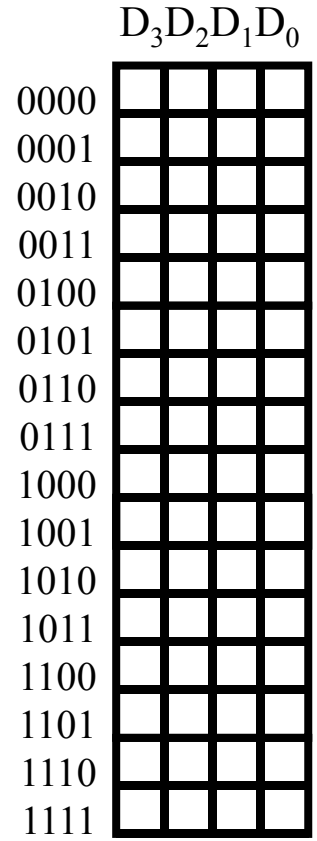

M27C256B
-12F1
B88BV
9514B
SINGAPORE

Look-Up Tables

A ROM can be used to implement combinational logic. For example, the 16 x 4 bit wide ROM shown can be considered as a combinational circuit with four outputs.

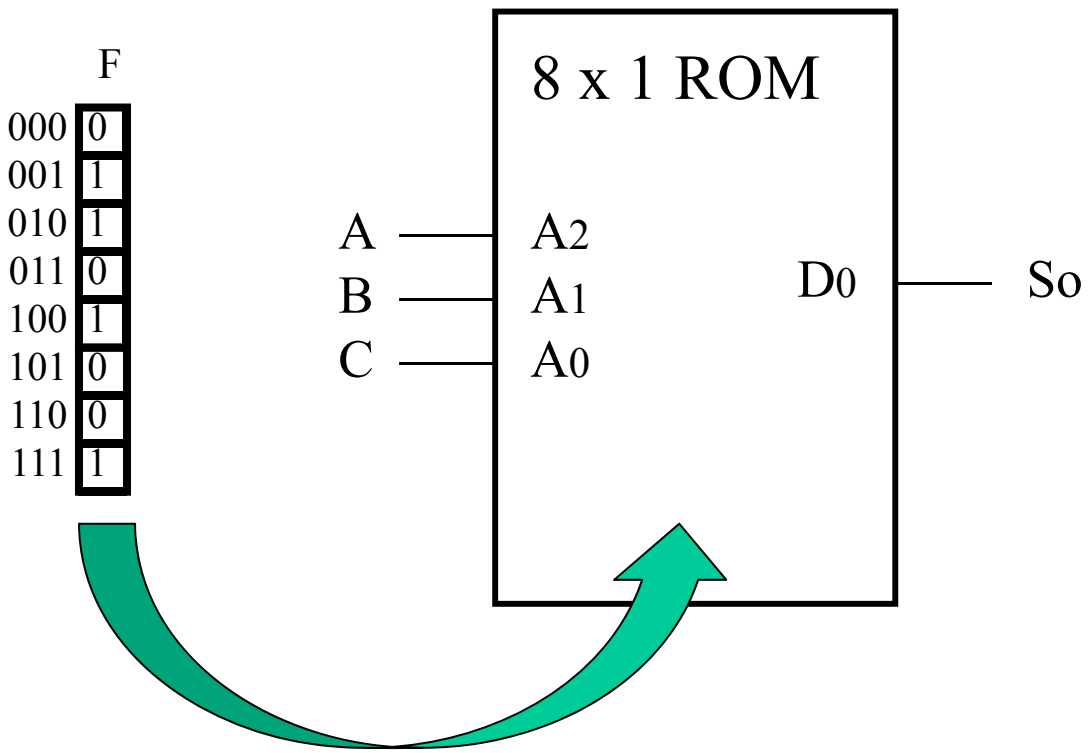
This ROM will have four address lines to uniquely locate each data word. The four bits can be considered as Boolean variables. The data stored in the memory gives the output for each of the input variable combinations.

Each memory location represents a minterm. This is known as a Look Up Table (LUT).



Look-Up Tables – Full Adder SUM output

A	B	C	S_o
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



LUT Example

Design a combinational circuit using a ROM which accepts a 3-bit number and generates an output binary number equal to the square of the input.

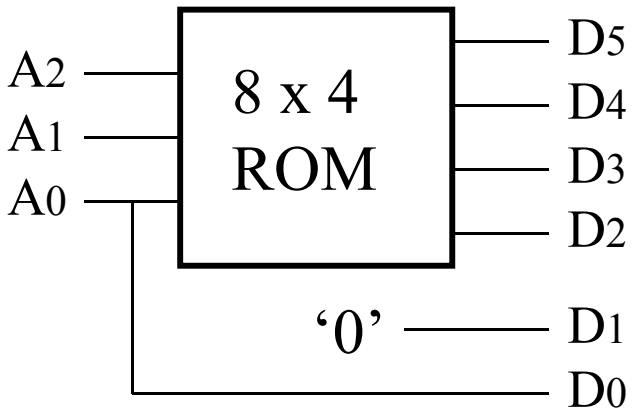
A_2	A_1	A_0	D_5	D_4	D_3	D_2	D_1	D_0	decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

By observation, D_1 is always ‘0’ and D_0 is equal to A_0. This leaves four outputs to be implemented, hence a four bit wide ROM.

ROM truth table

A_2	A_1	A_0	D_5	D_4	D_3	D_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

Circuit Design



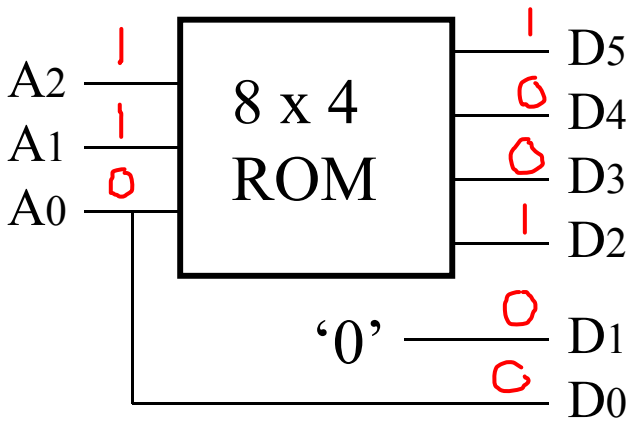
E.g. Calculate 6 squared.

6 in binary is 110

ROM truth table

A2	A1	A0	D5	D4	D3	D2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

Circuit Design

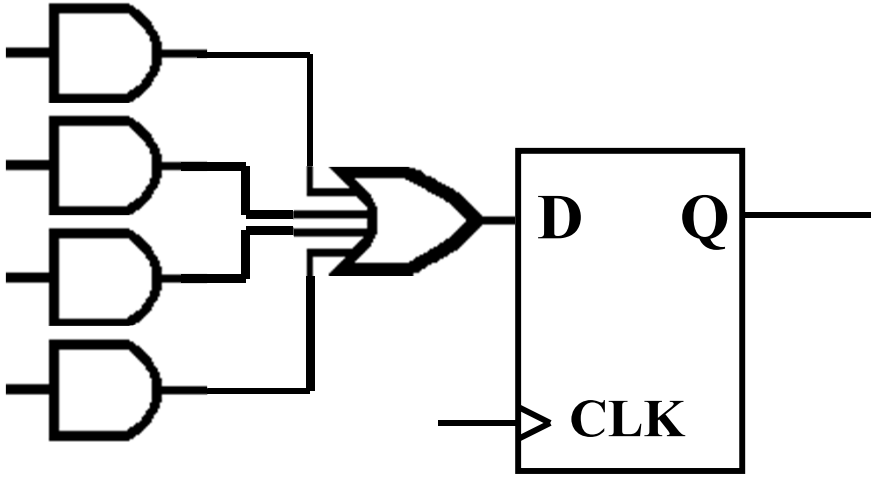


Ans. = 100100 = 36

Sequential Programmable Devices

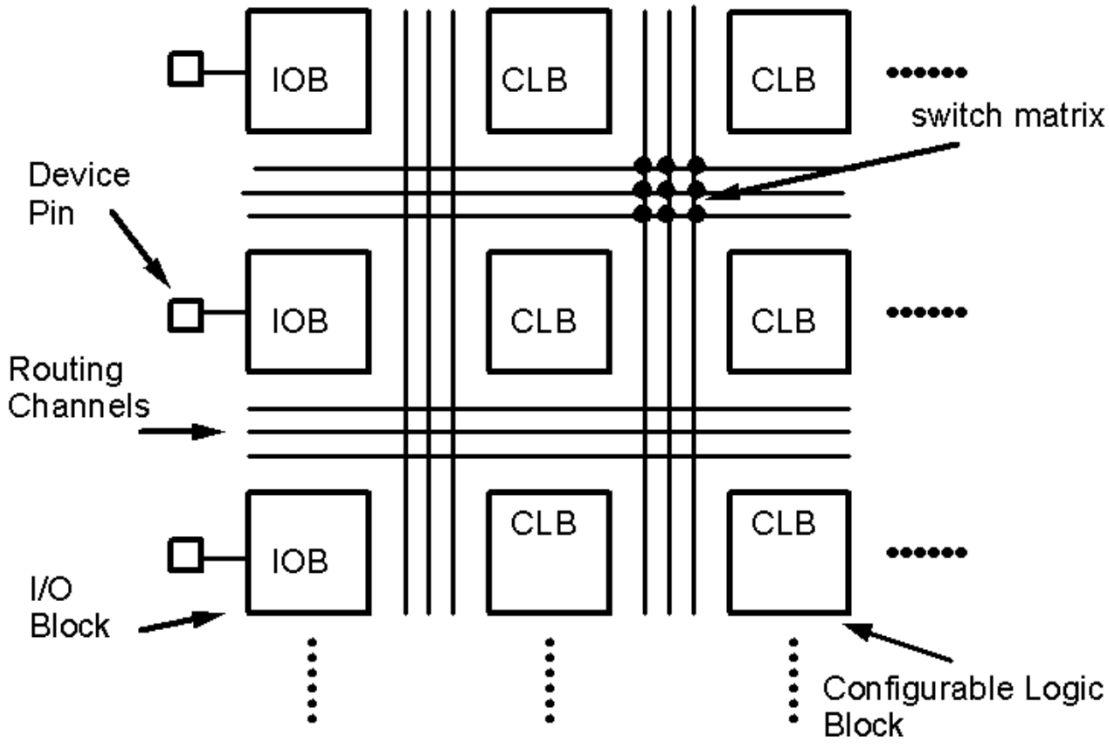
Combinational PLDs consist only of logic gates. Sequential PLDs include both gates and flip-flops. They fall into three general types.

1. Sequential (or simple) programmable logic device (SPLD)
2. Complex programmable logic device (CPLD)
3. Field programmable gate array (FPGA)



Field Programmable Gate Arrays

FPGAs consist of a large array of programmable logic blocks surrounded by I/O blocks. A network of programmable interconnects runs between the logic blocks to provide connectivity.



Resources are available to ensure correct clock operation throughout the chip at the high speeds available.

Xilinx FPGAs

- RAM based - reprogrammable, chip memory
- Tens to hundreds of thousands of gates available
- Rapid prototyping on the desktop
- Cost effective for small volume
- PC design tools available ‘free’ for design capture, simulation, synthesis and download

http://www.xilinx.com

Summary

- Digital logic functions can be implemented on devices programmed by the user
- Combinational logic can be implemented using memory look up tables
- FPGAs provide millions of gates, memory and arithmetic functions on a single device.