Digital Electronics and Microprocessor Systems (ELEC211)

Dave McIntosh and Valerio Selis

dmc@liv.ac.uk

v.selis@liv.ac.uk

Digital 5: CPLDs and FPGAs
Shannon's expansion

UNIVERSITY OF LIVERPOOL

Outline

- Complex Programmable Logic Devices (CPLDs)
- Field Programmable Gate Arrays (FPGAs)
- Shannon's expansion

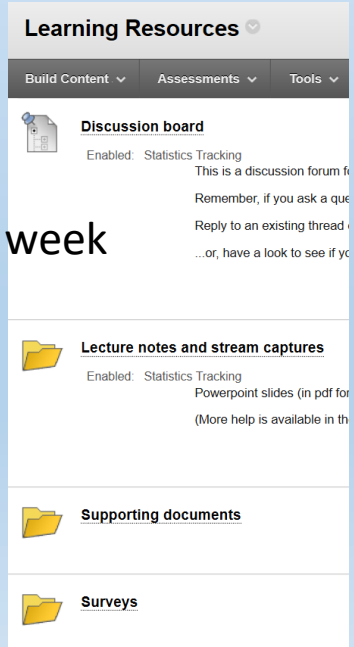
Previous material

MUX ✓
PLA, PAL ✓
Flip-flops ✓
Karnaugh maps ✓

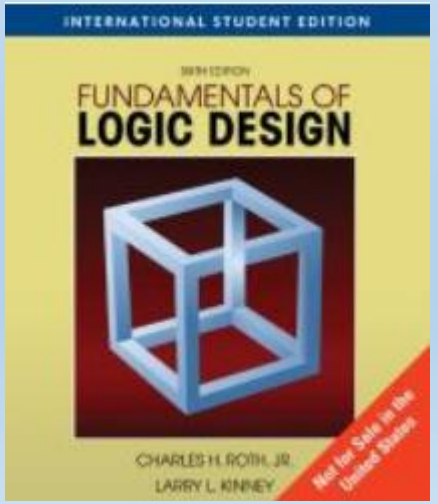
Use VITAL!:

- Stream lectures
- Handouts
- Notes and Q&A each week
- Discussion Board
- Exam resources

www.liv.ac.uk/vital



Course textbook – please borrow and use it!



UNIVERSITY OF LIVERPOOL

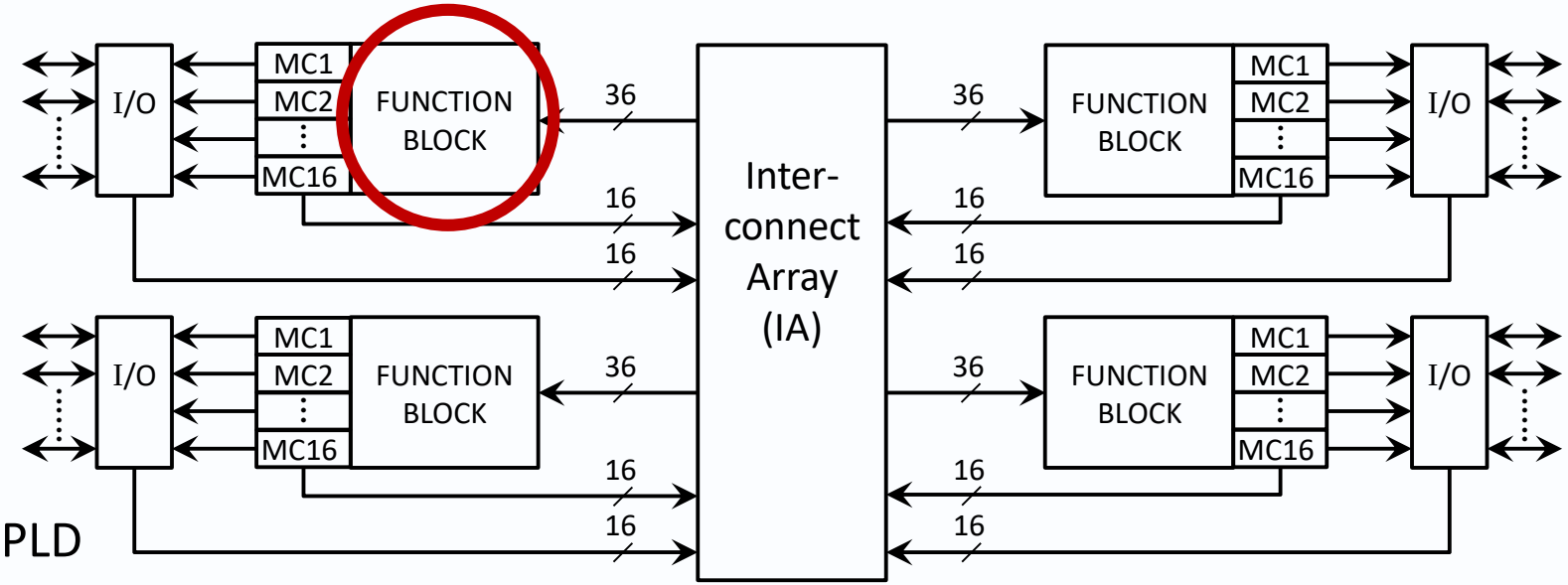
Complex Programmable Logic Devices (CPLD)

Xilinx XCR3064XL

UNIVERSITY OF LIVERPOOL

From OHO Elektronik GOP_XCR3064XL user manual v0.9 at:
http://www.trenz-electronic.de/fileadmin/docs/OHO-Elektronik/TE0251-GOP_XCR3064XL/documents/UM_XCR3064XL.pdf

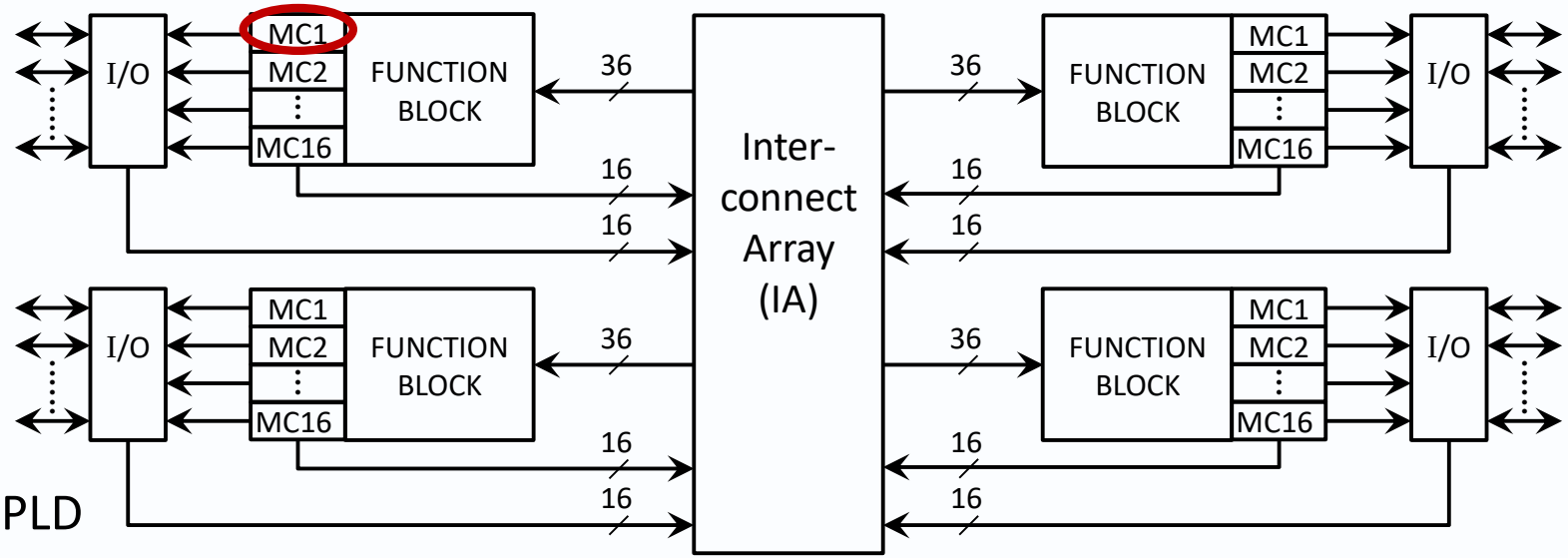
Complex Programmable Logic Devices (CPLDs)



Xilinx XCR3064XL CPLD

Each Function Block is a programmable AND-OR array, configured as a PLA.

Complex Programmable Logic Devices (CPLDs)



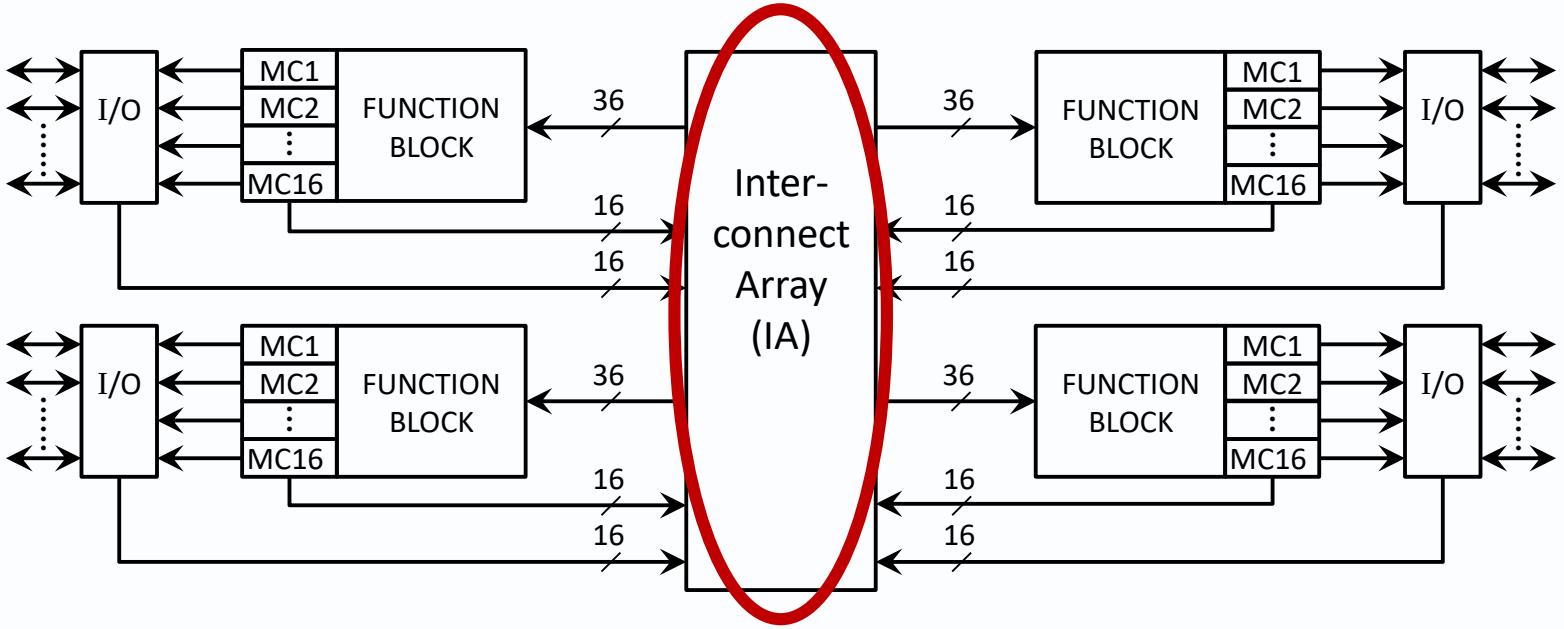
Xilinx XCR3064XL CPLD

Each **Macrocell (MC)** contains a:

- **Flip-Flop** and
- **multiplexers**

…that route signals from the function block to the **Input/Output (I/O)** block or to the **interconnect array (IA)**

Complex Programmable Logic Devices (CPLDs)

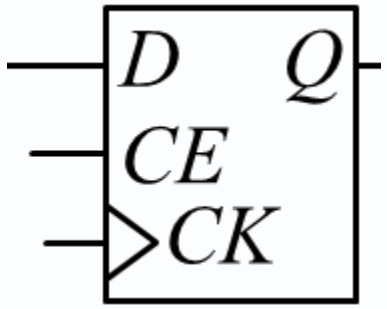


The **IA** selects signals from the **Macrocell outputs** or **I/O blocks** and connects them back to **function block inputs**.

Signal generated in one function block → can input to any other function block

The I/O blocks provide an interface between the bi-directional I/O pins and the interior of the CPLD

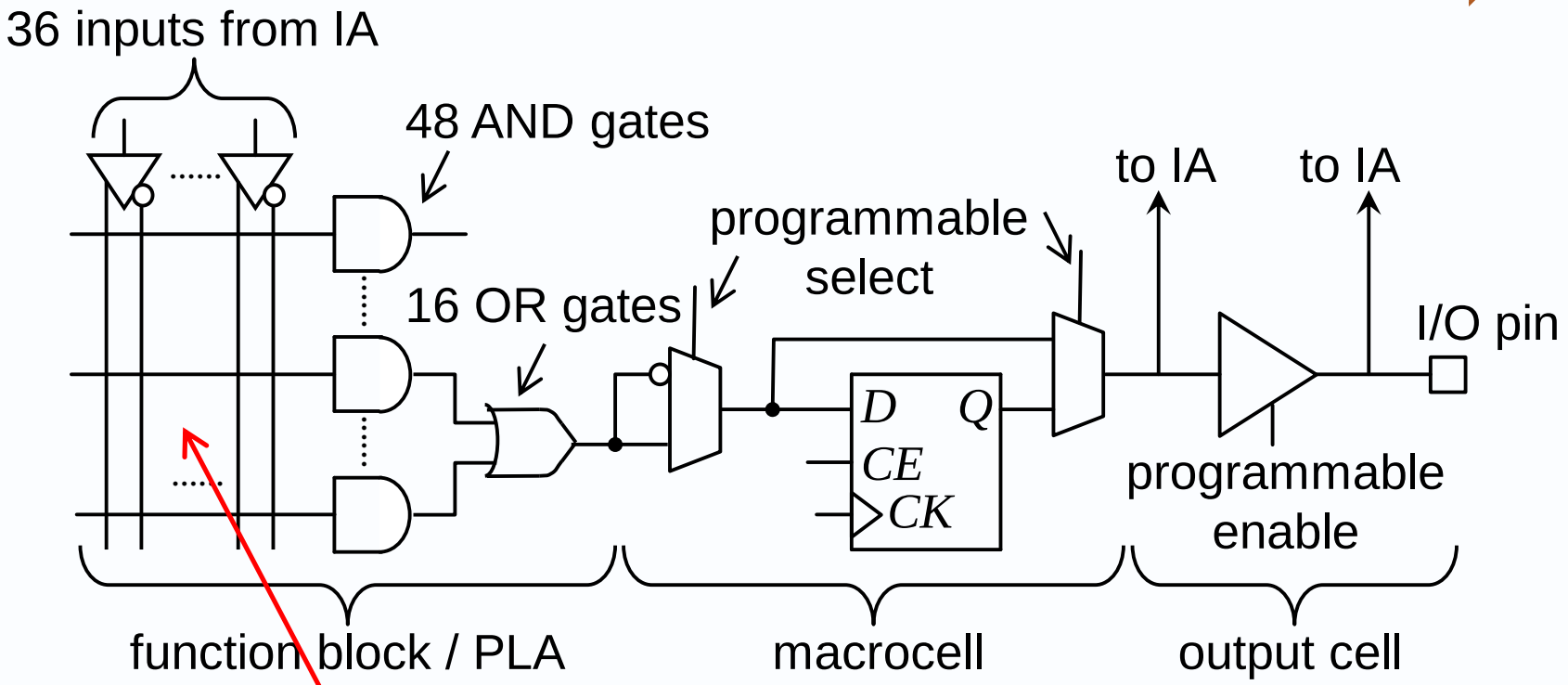
Clock enable (side-note)



Clock enable (CE):

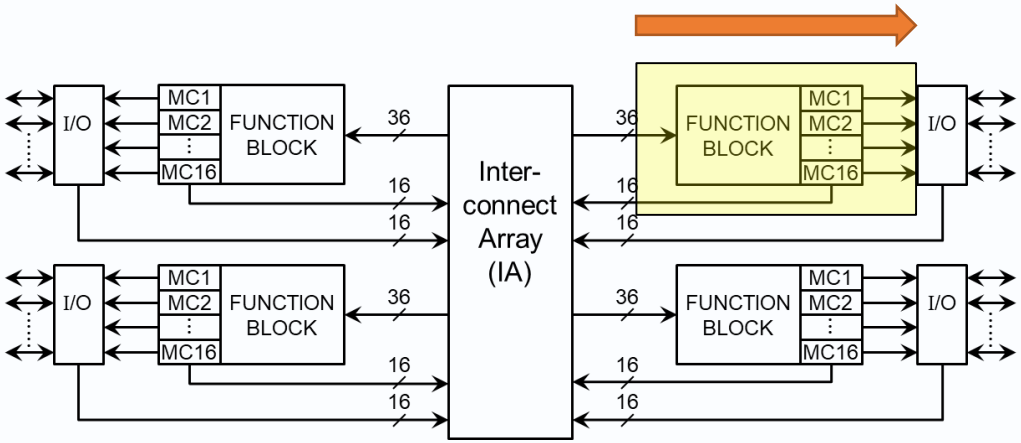
- When CE=1, normal flip-flop operation on active edge
- When CE = 0, clock disabled, no state change ($Q^+=Q$)

CPLD Function Block & Macrocell



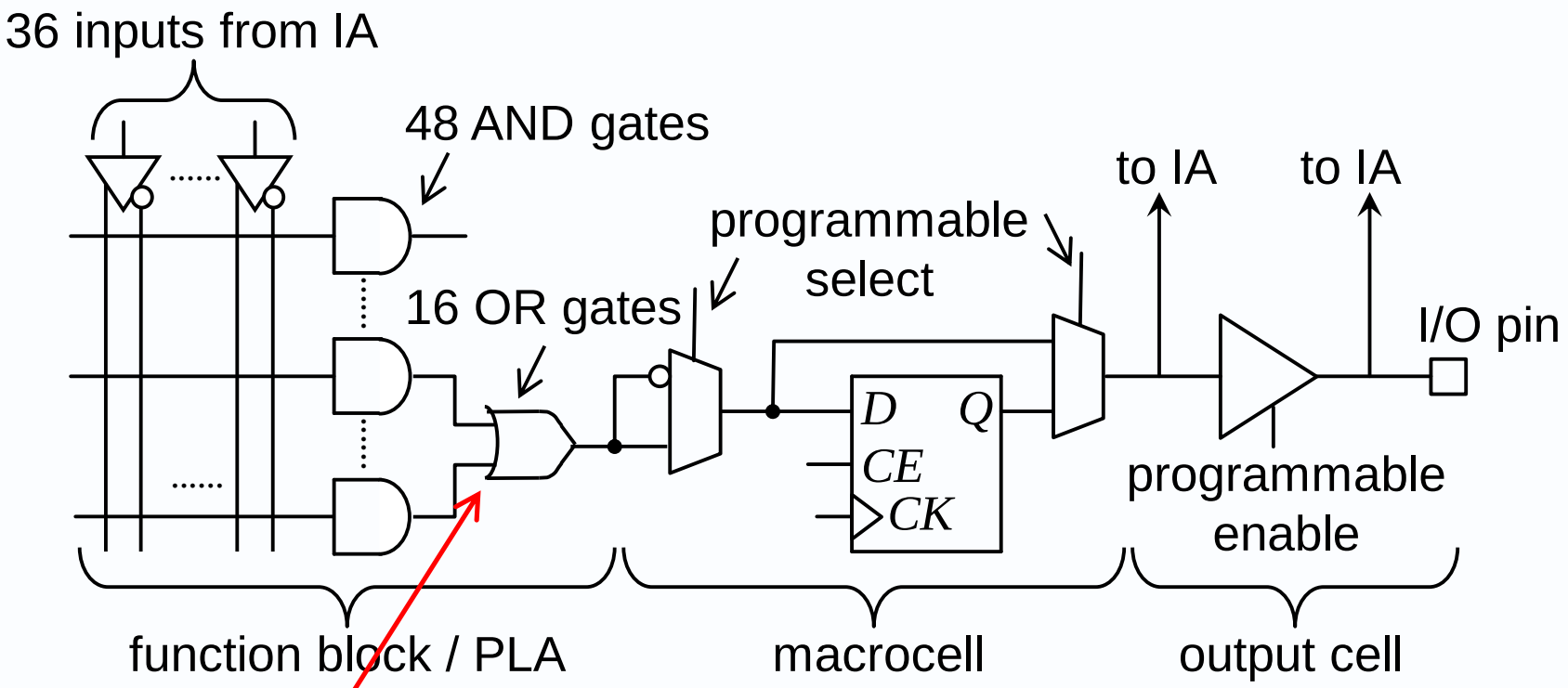
How is a signal (generated in the PLA) routed to an I/O pin via a macrocell?

- Any of the 36 inputs from the IA can be connected to any of the 48 AND gates.



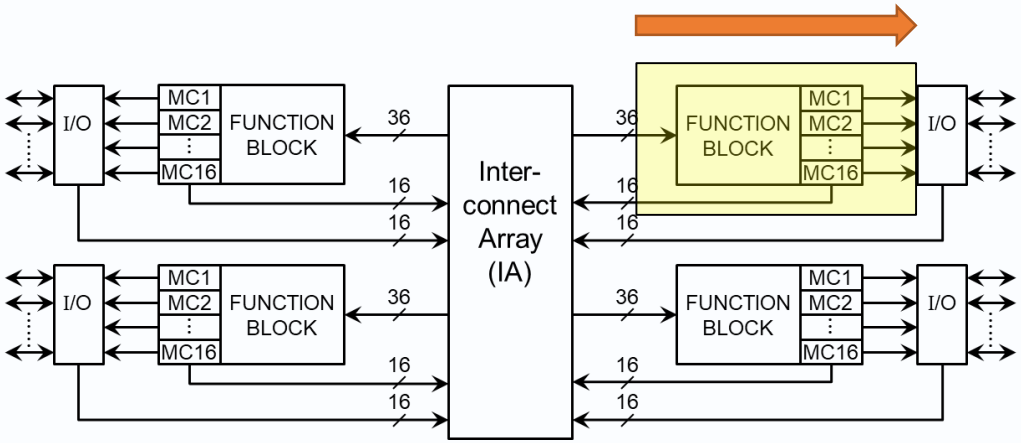
UNIVERSITY OF LIVERPOOL

CPLD Function Block & Macrocell



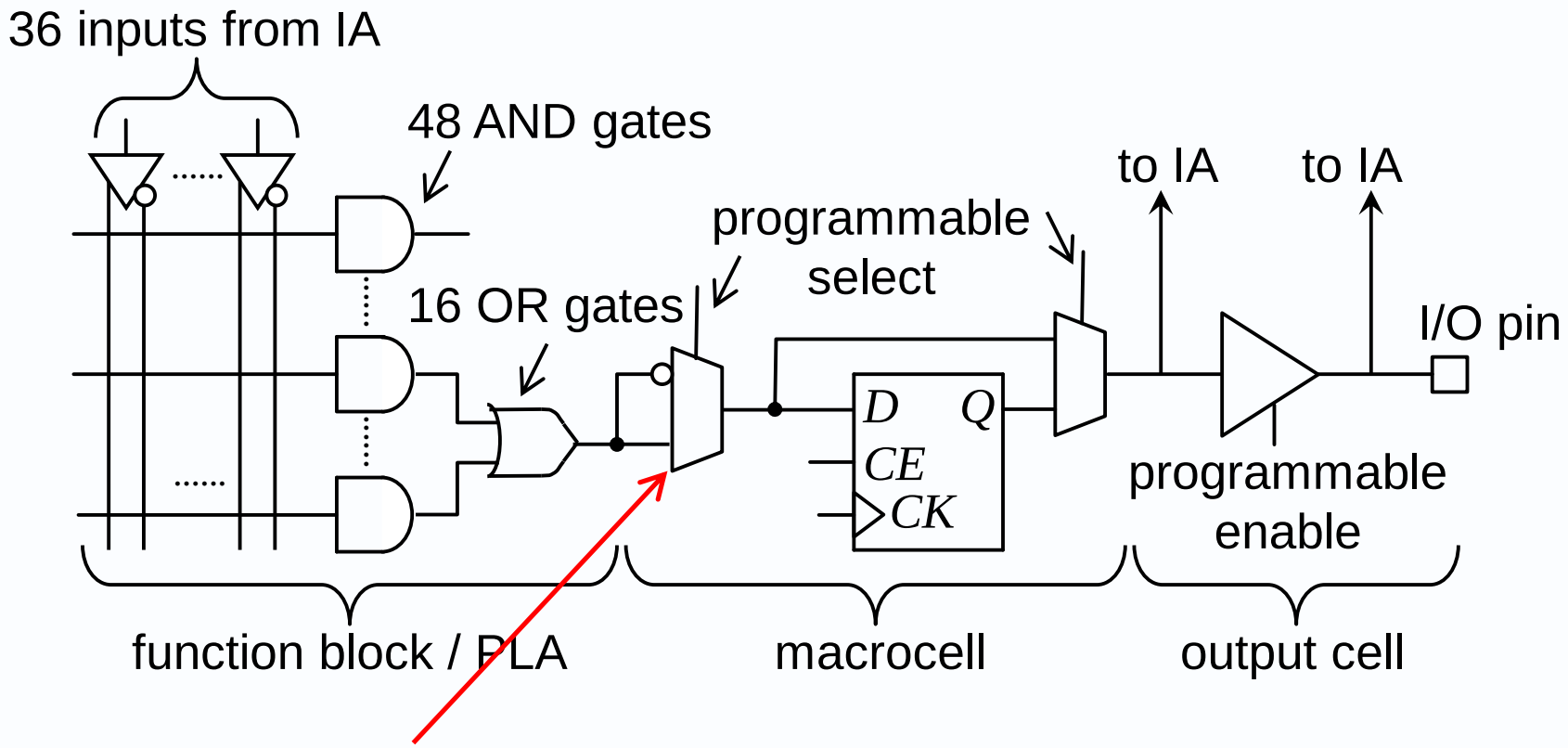
How is a signal generated in the PLA routed to an I/O pin via a macrocell?

- Each OR gate can accept up to 48 product terms from the AND array.



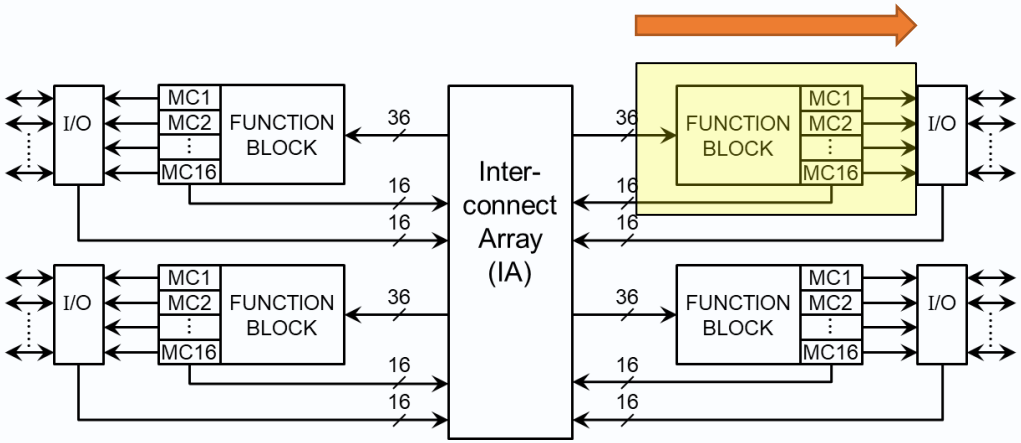
UNIVERSITY OF LIVERPOOL

CPLD Function Block & Macrocell

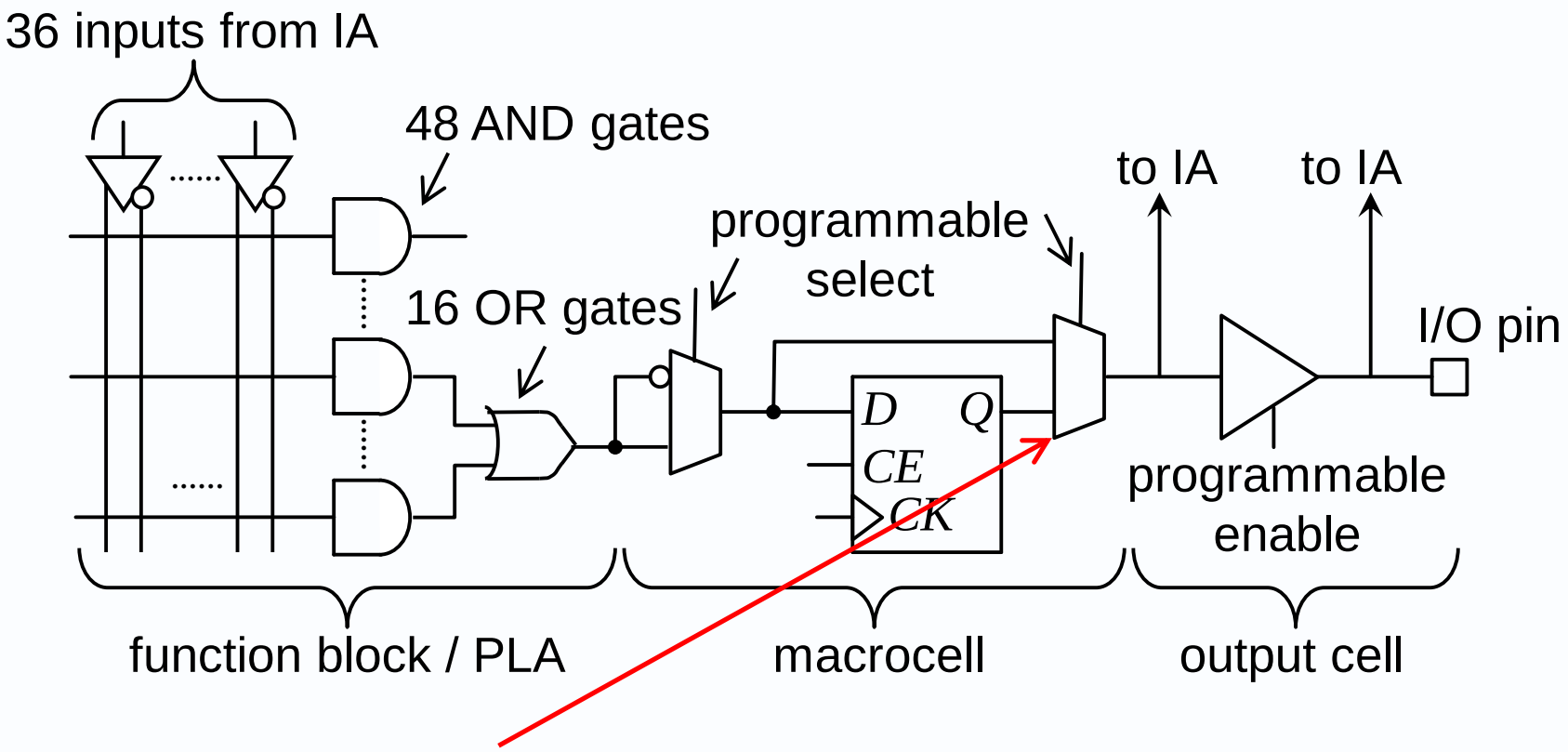


How is a signal generated in the PLA routed to an I/O pin via a macrocell?

- The first Mux can be programmed to select the output of the OR gate or its inverse.

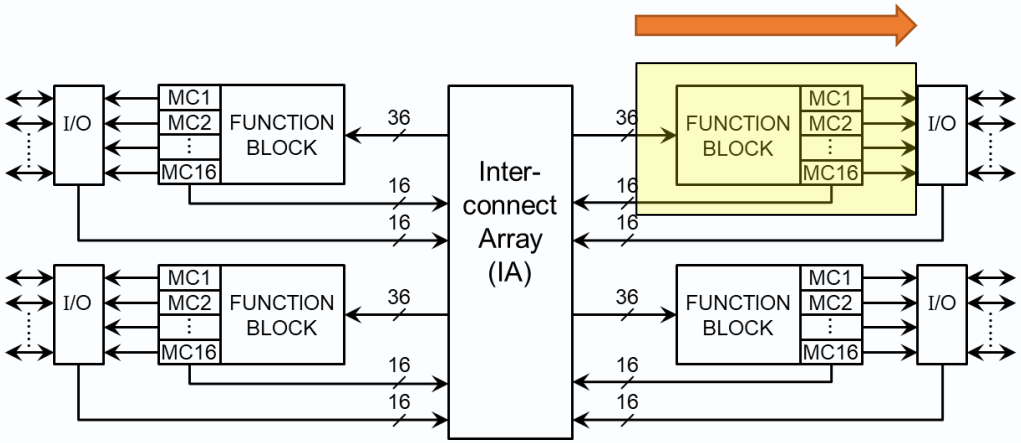


CPLD Function Block & Macrocell

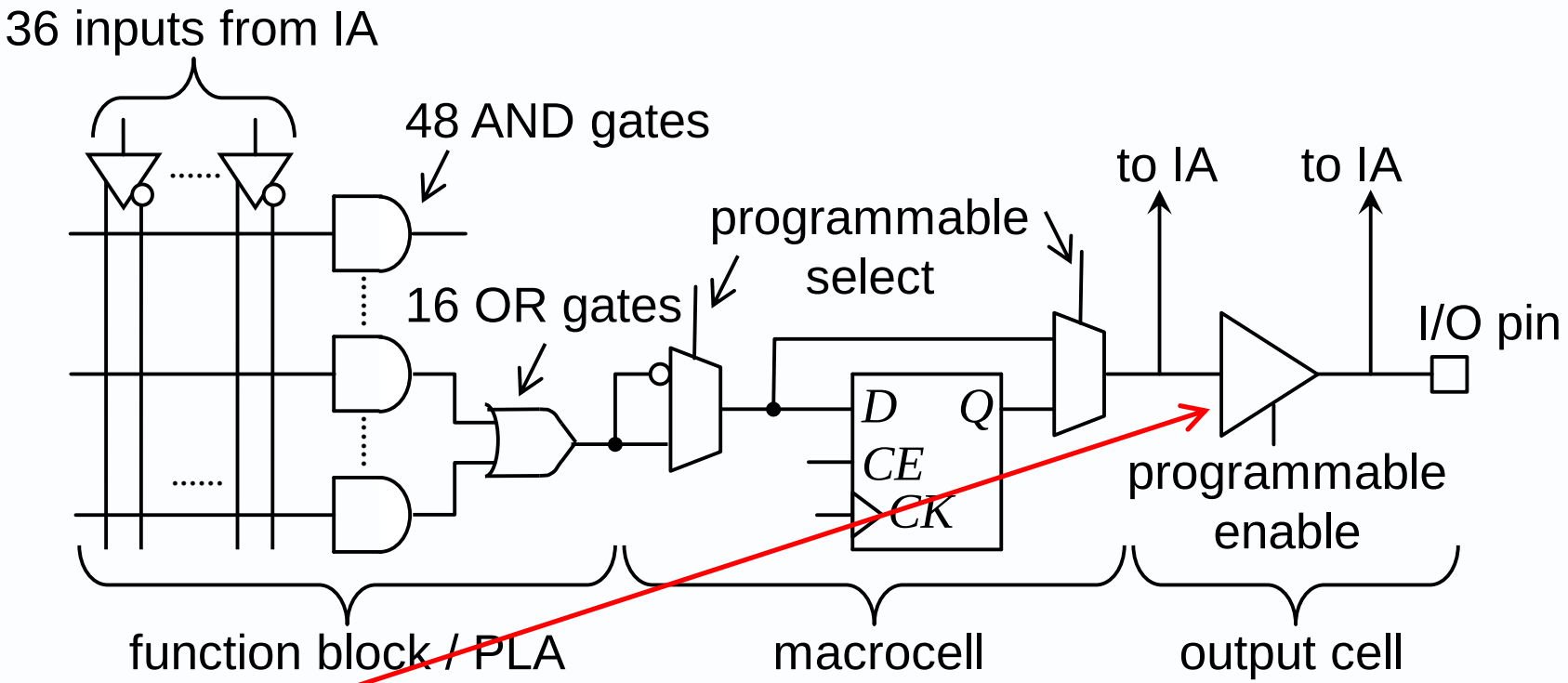


How is a signal generated in the PLA routed to an I/O pin via a macrocell?

- The second Mux can be programmed to select either the combinational output or the flip-flop output (Q).

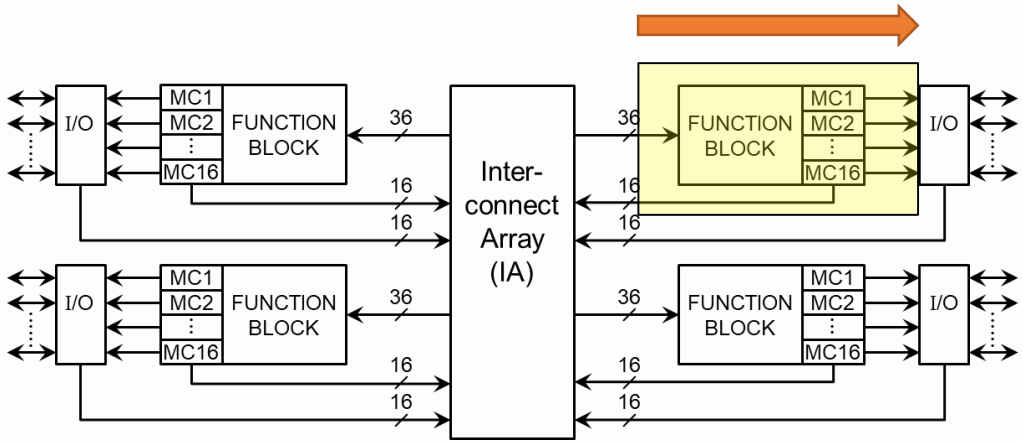


CPLD Function Block & Macrocell



How is a signal generated in the PLA routed to an I/O pin via a macrocell?

- The tri-state buffer defines the I/O pin as input (buffer disabled) or output
- Sophisticated CAD software is available for fitting logic circuits into a CPLD



UNIVERSITY OF LIVERPOOL

Field Programmable Gate Arrays (FPGA)

This material will take you through a Altera® DE1 Development and Education board (Figure 1). Featuring an Altera Cyclone® II 2C20 FPGA, the DE1 board is designed for university and college laboratory use. It is suitable for a wide range of exercises in courses on digital logic and computer organization, from simple tasks that illustrate fundamental concepts to advanced designs.

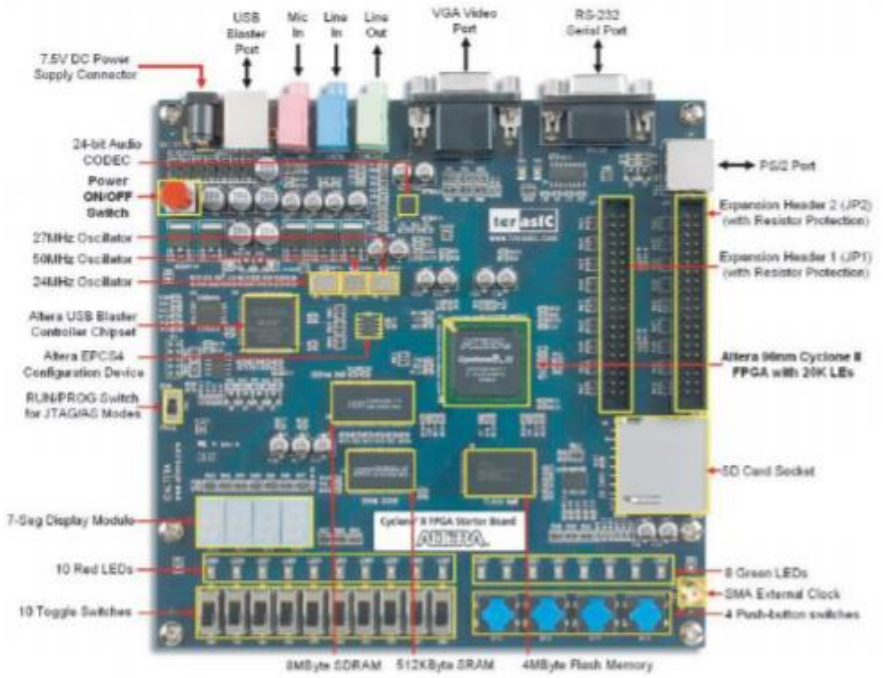


Figure 1 Altera DE1 Board

The following hardware is provided on the DE1 board:

- Altera Cyclone® II 2C20 FPGA device
- Altera Serial Configuration device – EPCS4
- USB Blaster (on board) for programming and user API control; both JTAG and Active Serial (AS) programming modes are supported
- 512-Kbyte SRAM
- 8-Mbyte SDRAM
- 4-Mbyte Flash memory

Altera DE1 board

Cyclone II 2C20 FPGA

From supporting material for Experiment 28, 2018/19

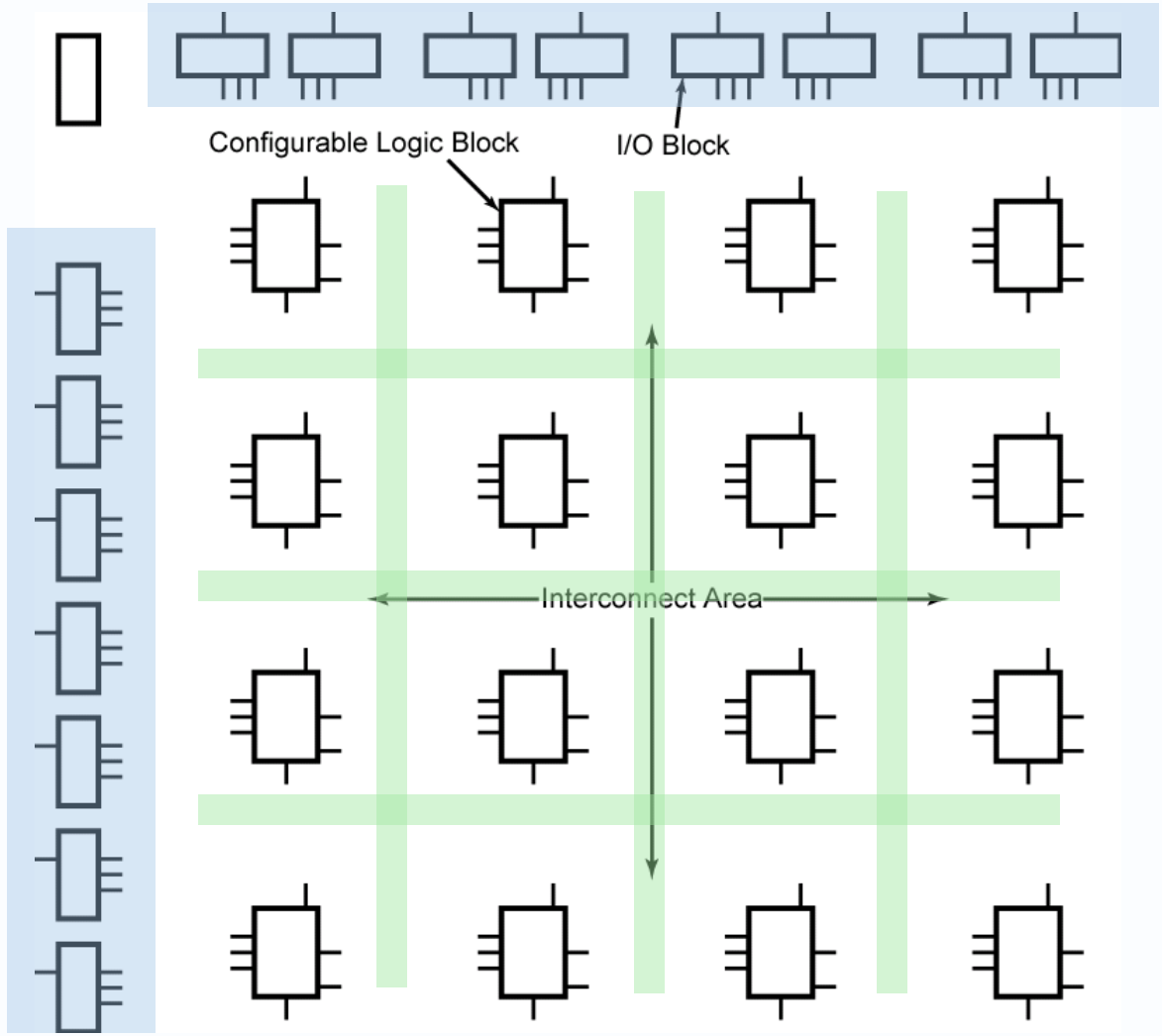
UNIVERSITY OF LIVERPOOL

Field Programmable Gate Arrays (FPGA)

An FPGA is an IC that contains an array of identical logic cells.

The I/O blocks connect the Configurable Logic Blocks (CLBs) to IC pins

The I/O blocks, the CLBs and the interconnection between CLBs are programmable



Simplified CLB (Configurable Logic Block)

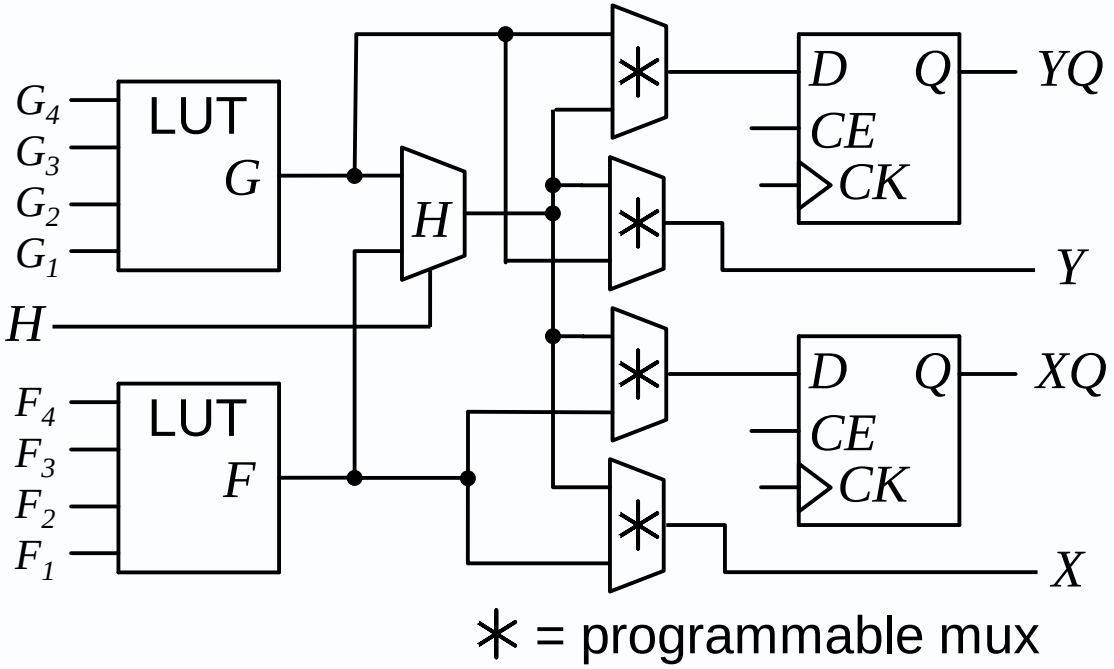
This (simplified) CLB contains 2 function generators, 2 flip-flops, and multiplexers for routing signals within the CLB

The function generators are implemented as lookup tables (LUT).

A 4-input LUT is essentially a reprogrammable ROM with 16 1-bit words.

The CLB has two combinational outputs (X,Y) & two flip-flop outputs (XQ, YQ).

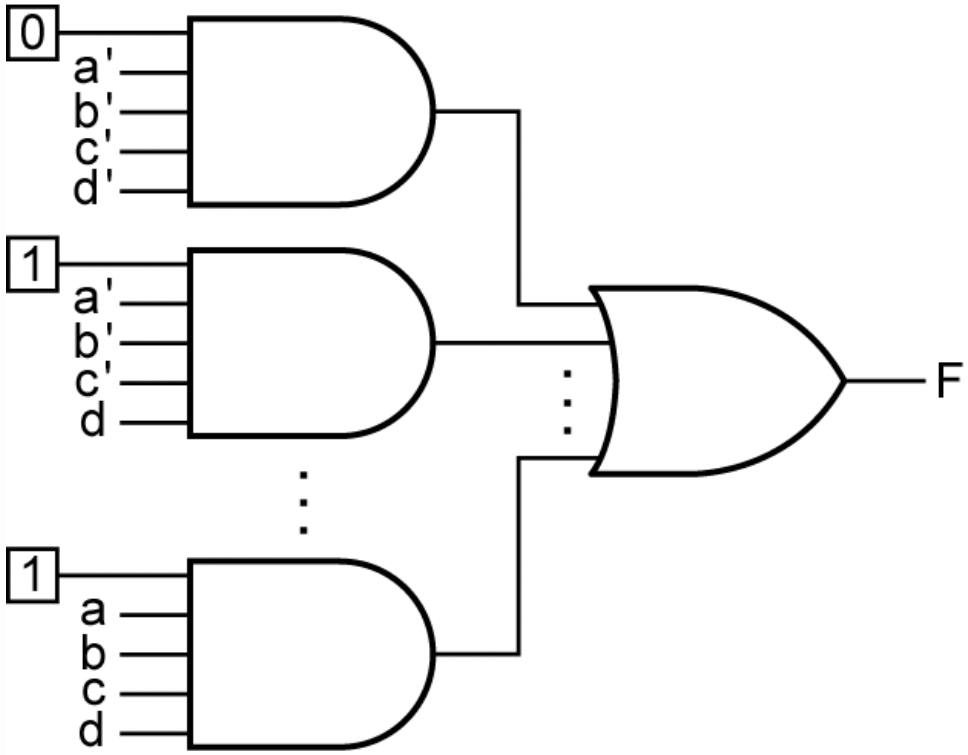
X and Y outputs & the flip-flop inputs are selected by the programmable Multiplexers (the control inputs are programmed when the FPGA is configured)



Implementing a (LUT) function generator

0 or 1 in the squares represents the bits stored in the LUT. These bits enable particular minterms.

A function with only one minterm or with as many as 15 minterms requires a single function generator



e.g.

$$F = abcd \quad \text{or}$$

$$F = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}b\bar{c}d + a\bar{b}\bar{c}d + \bar{a}bc\bar{d} + a\bar{b}c + abc + ab\bar{c} + \bar{a}\bar{b}cd$$

Claude E. Shannon

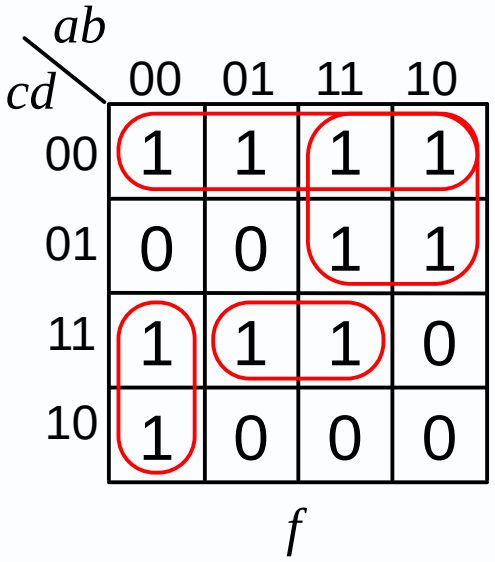
C. Shannon, 1916 -2001, American mathematician & electronic engineer, known as the ‘Father of Information Theory’.

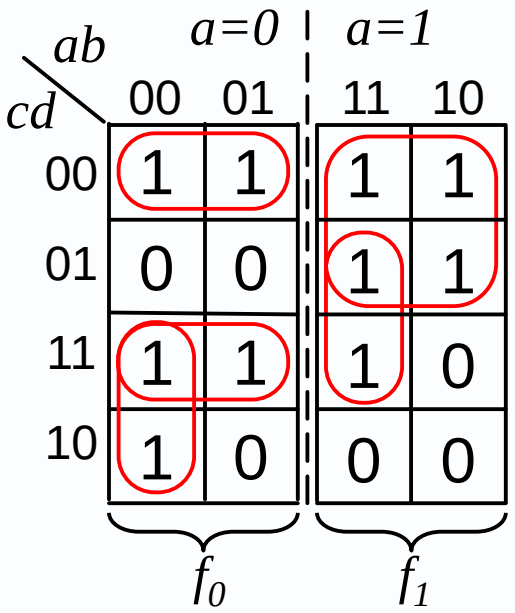
Also, in 1937 he developed digital circuit design theory as an MSc student at MIT; described as "possibly the most important, & also the most famous, MSc of the century."

Shannon's theorem

$$f(a,b,c,d) = c'd' + a'b'c + bcd + ac'$$
$$= a'f_0 + af_1 \ (our\ goal)$$

Here's how we do it:





$$f = a'(c'd' + b'c + cd) + a(c' + bd)$$

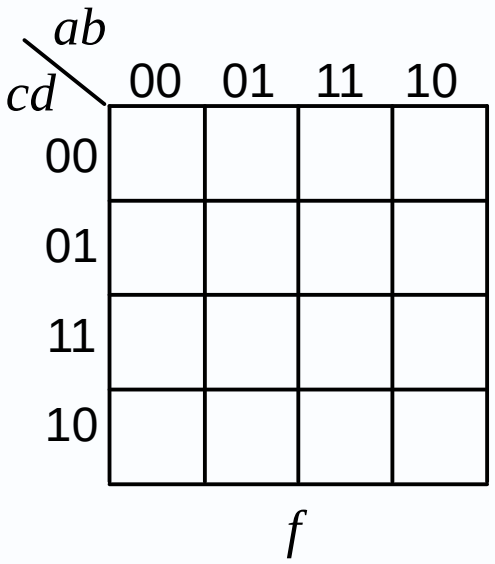
Example: Shannon's expansion (decomposition)

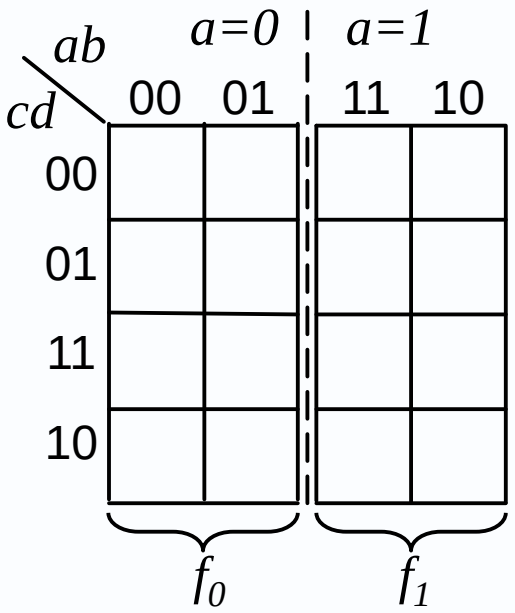
Decompose the following function into 2 functions, one for a' and the other for a.

$$f(a,b,c,d) = a'c'd' + abd + bcd + b'cd' + acd'$$
$$= a'f_0 + af_1$$

Method

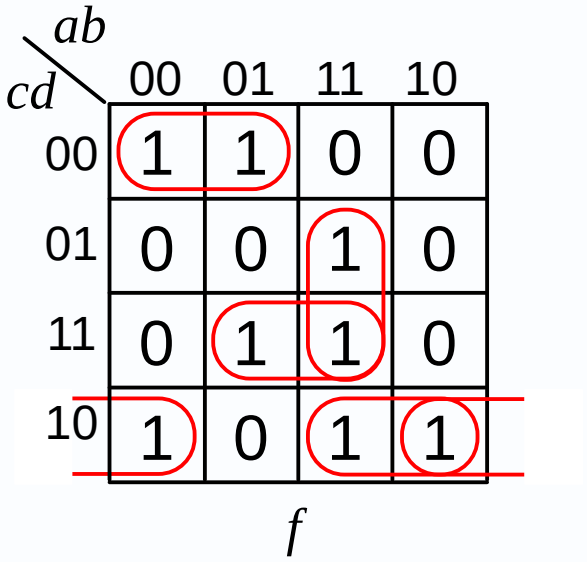
$$f(a,b,c,d) = a'c'd' + abd + bcd + b'cd' + acd'$$
$$= a'(\ldots) + a(\ldots)$$

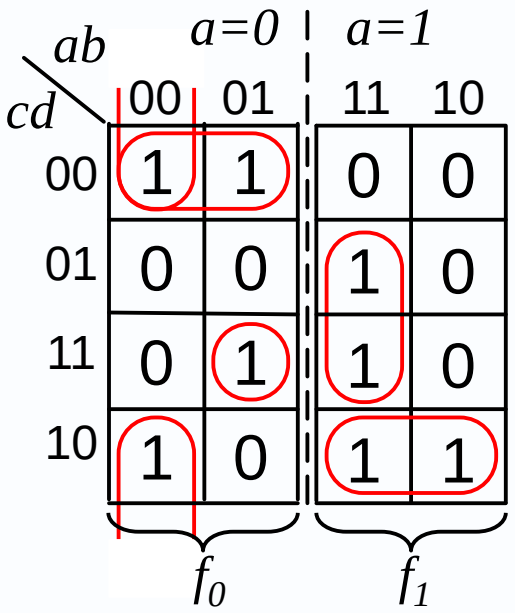




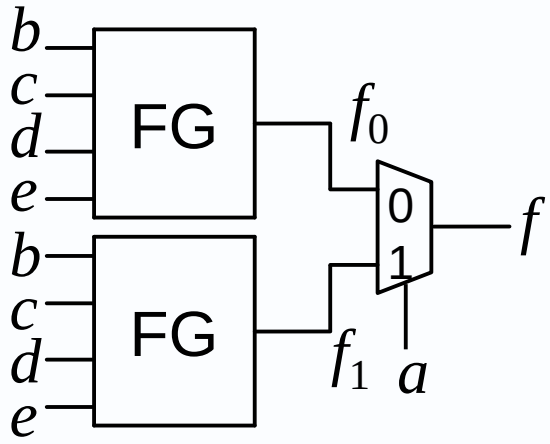
Answer

$$f(a,b,c,d) = a'c'd' + abd + bcd + b'cd' + acd'$$
$$= a'(c'd' + b'd' + bcd) + a(cd' + bd)$$





Realization of 5-variable functions with Function Generators



- Any 5-variable function can be realized using two 4-variable function generators and a 2-to-1 mux.

$$f(a,b,c,d,e) = a'f(0,b,c,d,e) + af(1,b,c,d,e)$$
$$= a'f_0 + af_1$$

Example

Decompose the following function into 2 functions, one for a' and the other for a.

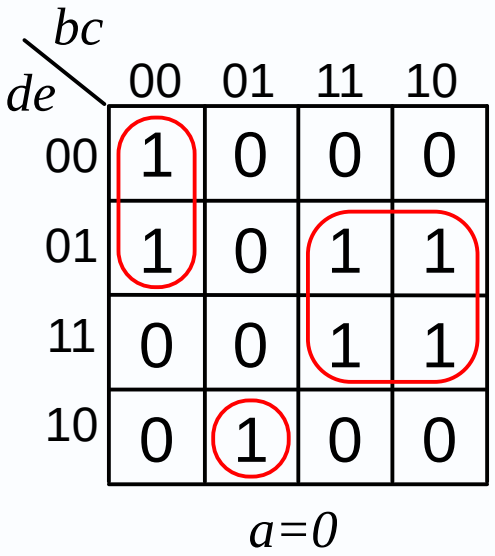
$$f(a,b,c,d,e) = b'c'd' + a'be + b'cde' + ab'd' + bcde$$
$$= a'f_0 + af_1$$

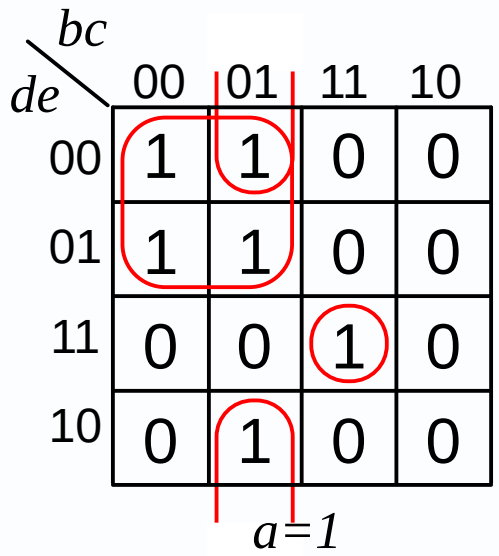
Answer

$$f = (a + a')b'c'd' + a'be' + (a + a')cde' + ab'd' + (a + a')bcde$$

$$f(a,b,c,d,e) = b'c'd' + a'be + b'cde' + ab'd' + bcde$$

$$= a'(b'c'd' + be + b'cde') + a(b'd' + b'ce' + bcde)$$

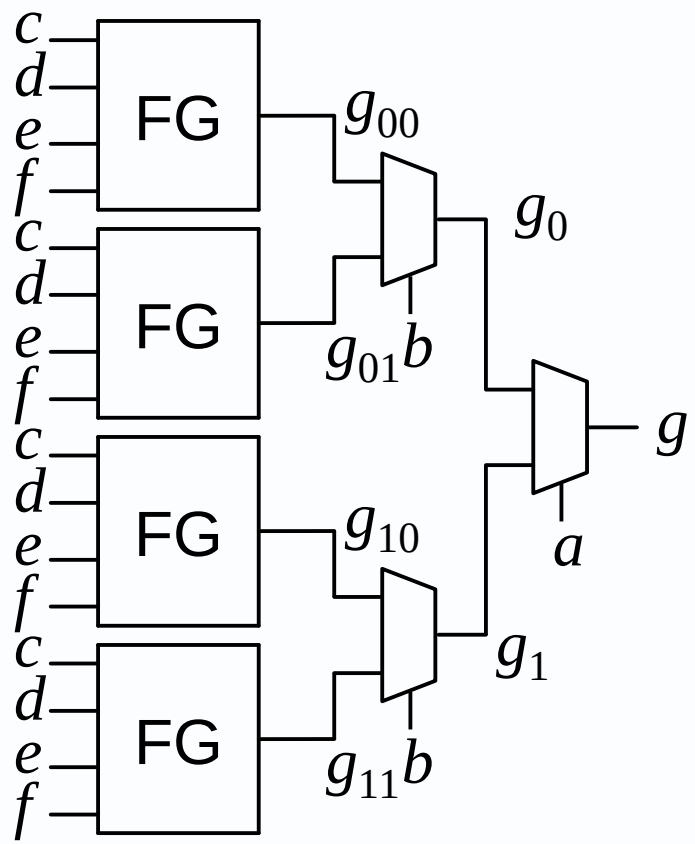




UNIVERSITY OF LIVERPOOL

Realization of 6-variable functions with Function Generators

Any 6-variable function can be realized using four 4-variable function generators and three 2-to-1 mux.



$$g(a,b,c,d,e,f) = a'b'g(0,0,c,d,e,f) + a'bg(0,1,c,d,e,f)$$
$$+ ab'g(1,0,c,d,e,f) + abg(1,1,c,d,e,f)$$

Summary and suggested reading

- CPLDs (Section 9.7)
- FPGAs (Section 9.8)
- Shannon's expansion

Roth and Kinney *Fundamentals of Logic Design*