TÉCNICO LISBOA

Computer Electronics

Lecture 4: FPGAs

Programmable Hardware

- Designing hardware is challenging, expensive and error prone
- A re-spin of an IC can cost 100s of M$ in modern nodes (e.g. 22 nm)
- Is there a way to “program circuits” and fix them if there are errors or updates?
- YES: electrically programmable digital circuits!

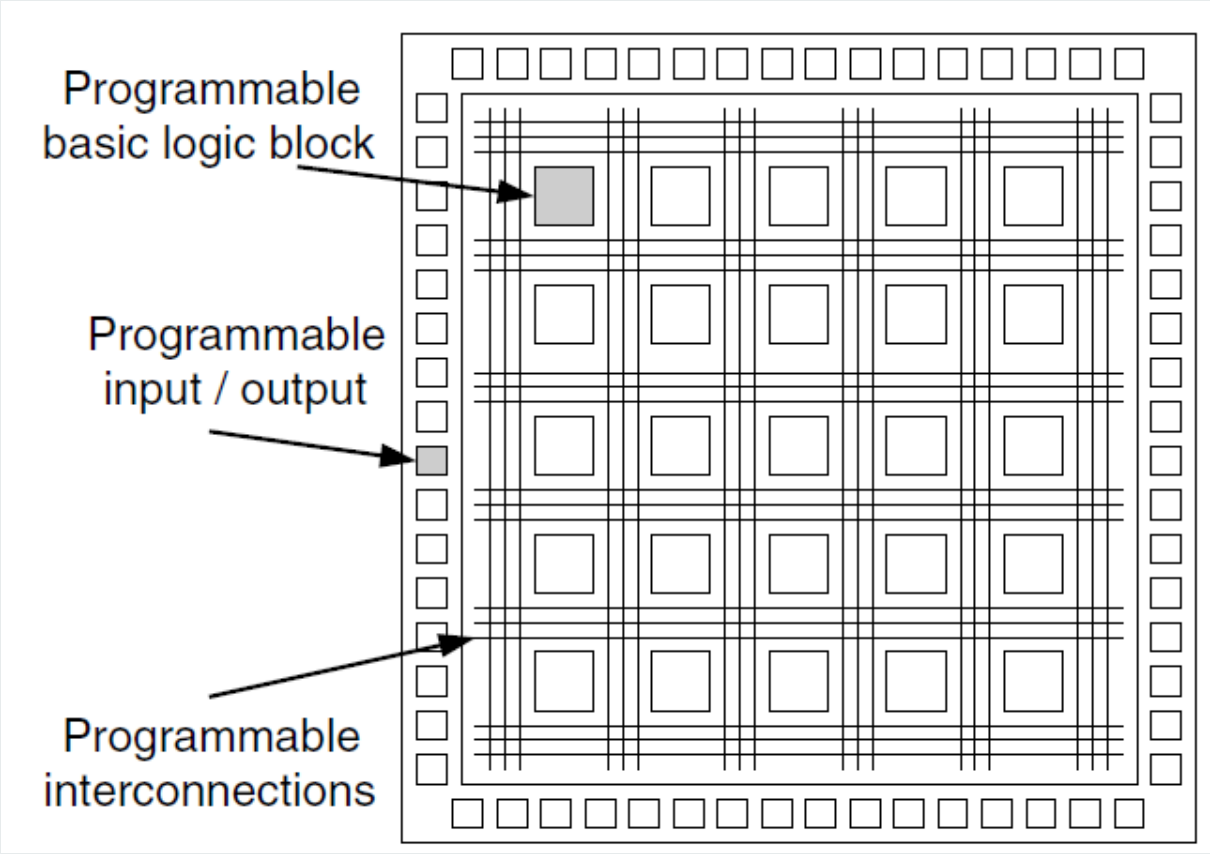
Field Programmable Gate Array (FPGA)

- Can implement any digital circuit, if it fits in the device
- Useful in prototyping chips
- Traditionally useful in low production volume products
  - 100s of K units and below
  - Beats IC unit costs!
- Clock frequencies well below ICs
- Cheap development cost
  - FPGA tools are cheap, design flow is relatively simple
- Nowadays useful as a general purpose processor, with high production volumes, like GPUs!!

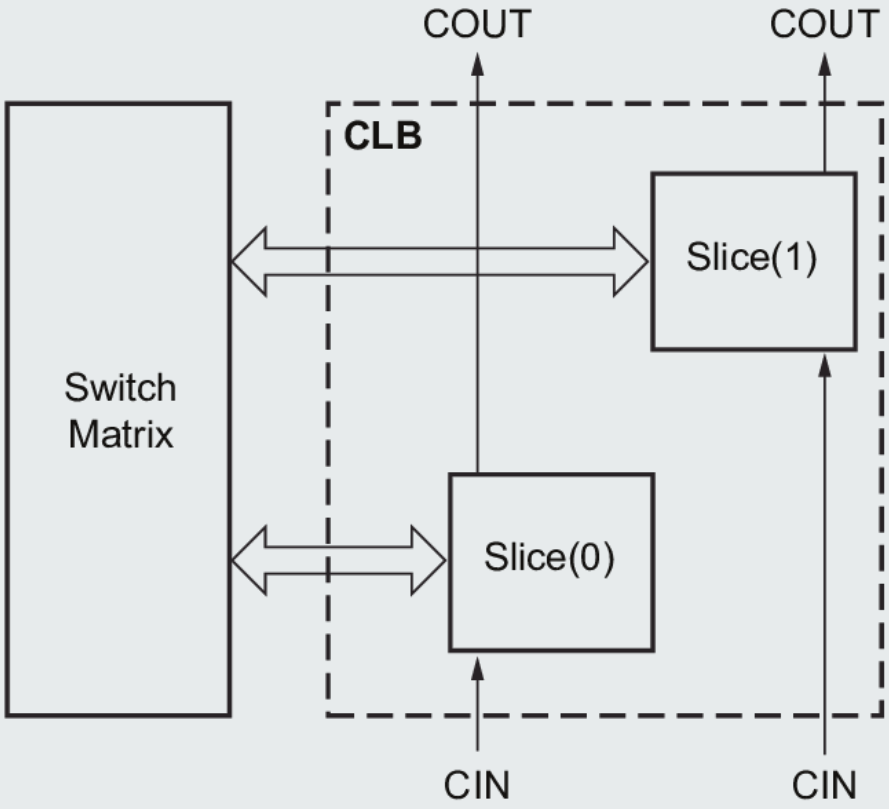
Early FPGA History

- Early 1980
  - Standard IC circuits (microprocessors, bus, I/O controllers, timers, …) had to be interconnected (serial/parallel, multiplexing, …)
  - Custom ICs were used for the interconnection logic but had huge cost and long development.
- 1984
  - Xilinx started selling FPGAs
  - Concept invented in 1967 by Wahlstrom
  - No physical layout, no mask making, no IC manufacturing, lower NRE costs and short time to market.

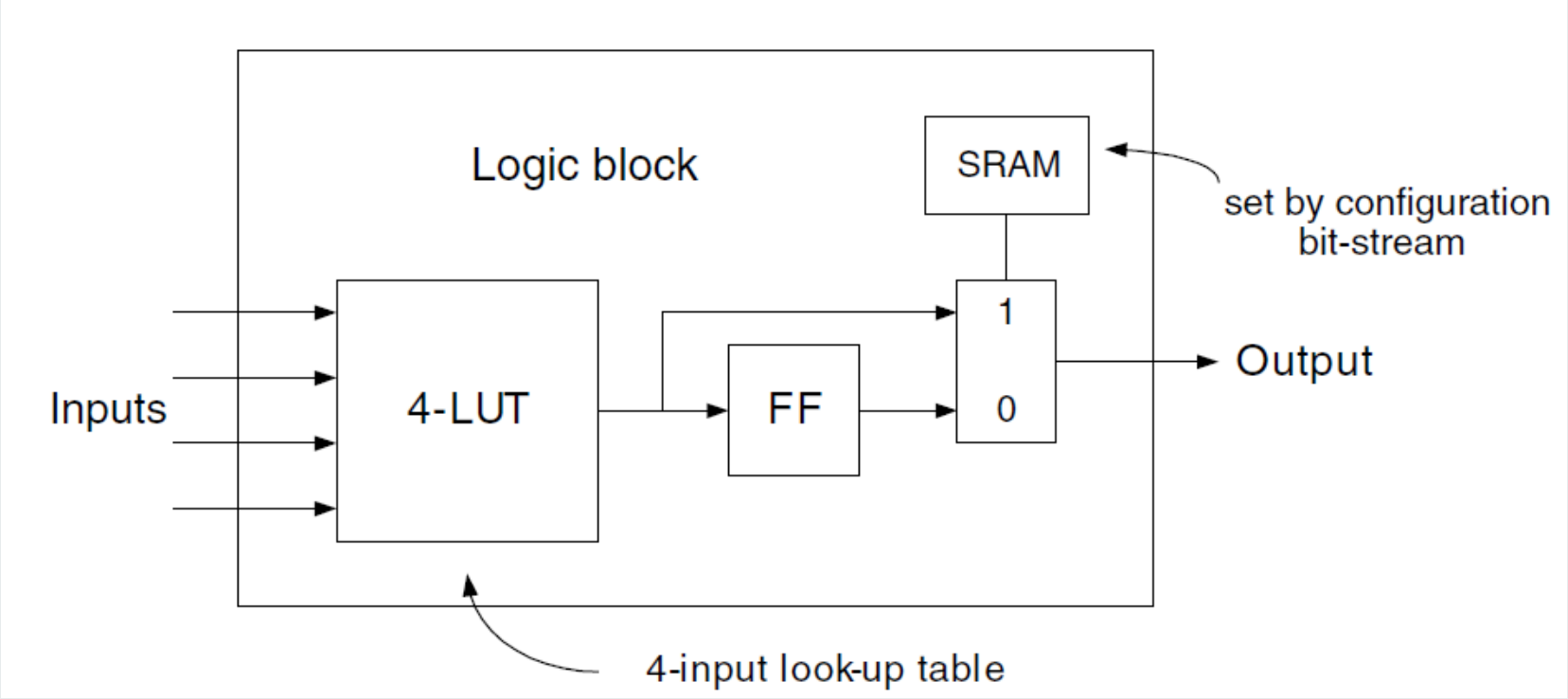
FPGA basic architecture



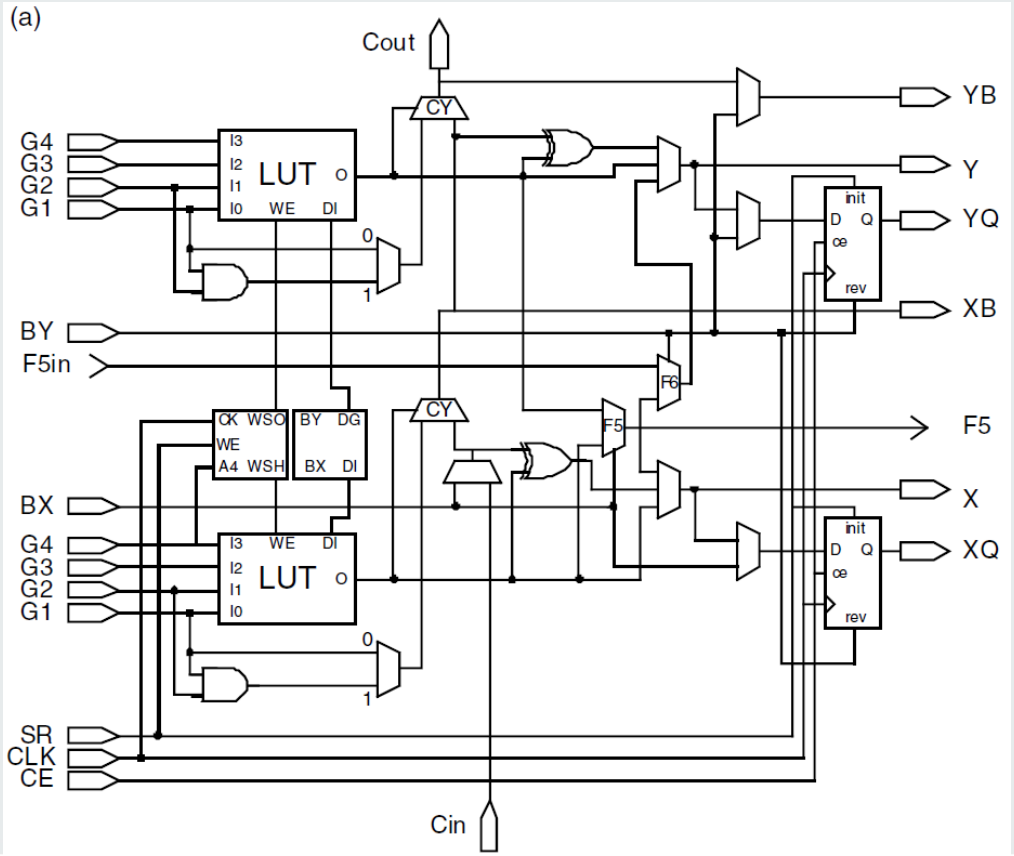
Basic Programmable Logic Block



Basic slice

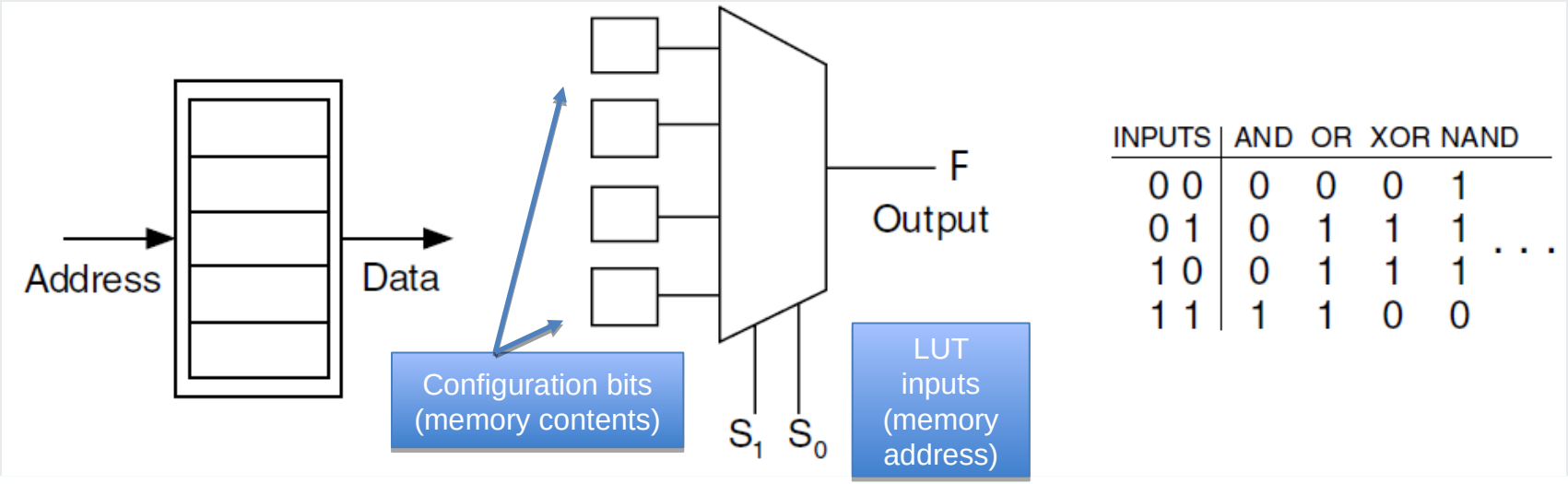

Logic block
SRAM
set by configuration
bit-stream
1
Output
Inputs
4-LUT
FF
0
4-input look-up table

Real slice


(a)
Cout
CY
YB
Y
YQ
XB
G4
G3
G2
G1
I3
I2
I1
I0
LUT
O
WE
DI
0
1
init
D
Q
ce
rev
BY
F5in
F6
F5
CK
WSO
BY
DG
WE
A4
WSH
BX
DI
X
XQ
BX
SR
CLK
CE
Cin

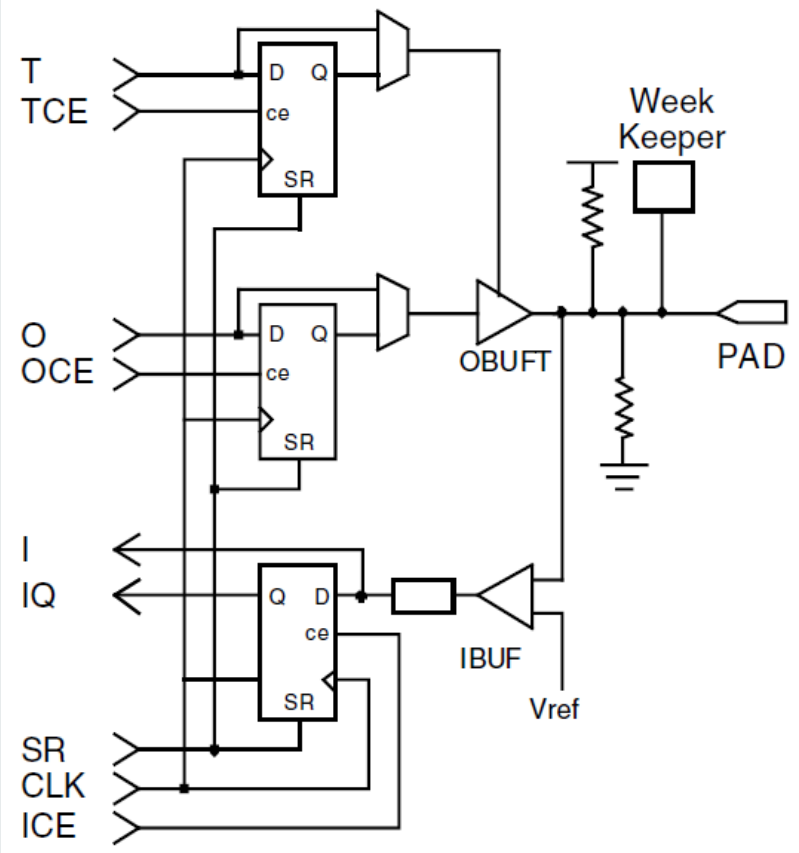
Look-Up Table (LUT)

- Structure: a very small memory
- Implementation: Latch + Multiplexer



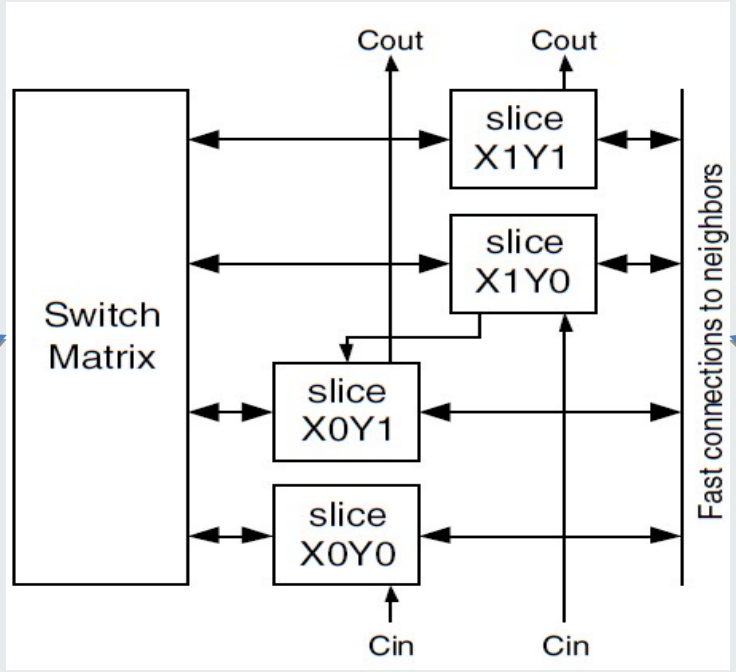
INPUTS	AND	OR	XOR	NAND
0 0	0	0	0	1
0 1	0	1	1	1
1 0	0	1	1	1
1 1	1	1	0	0

Programmable IO



Programmable interconnections

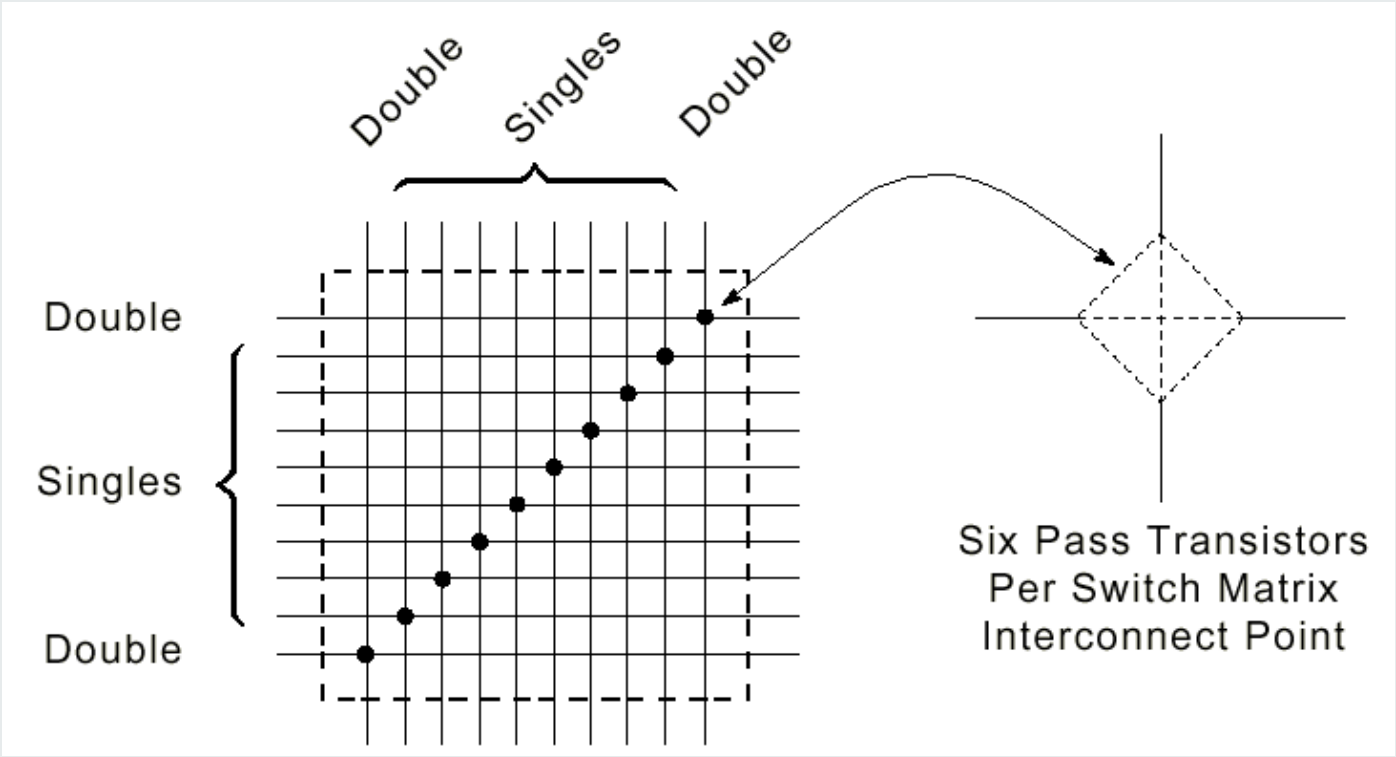
- Local CLB interconnection

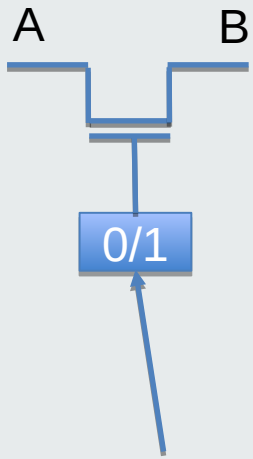


Connection to distant CLBs

Connection to neighbor CLBs

Switch matrices





Configuration bit:

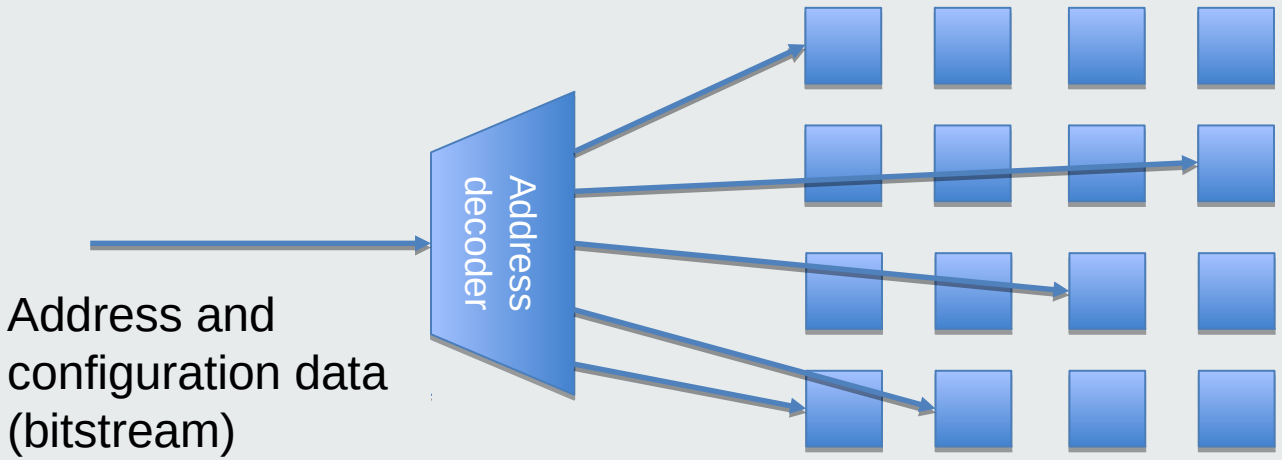
- 0: A not connected to B
- 1: A connected to B

Interconnect line types

- Long Lines: bidirectional wires that distribute signals across the device. Vertical and horizontal long lines span the full height and width of the device.
- Hex Lines: route signals to every third or sixth block away in all four directions.
- Double Lines: route signals to every first or second block away in all four directions.
- Direct Lines: route signals to neighboring blocks—vertically, horizontally, and diagonally.
- Fast Lines: internal CLB local interconnections from LUT outputs to LUT inputs.

Configuration architecture

- RAM organization



Memory and arithmetic

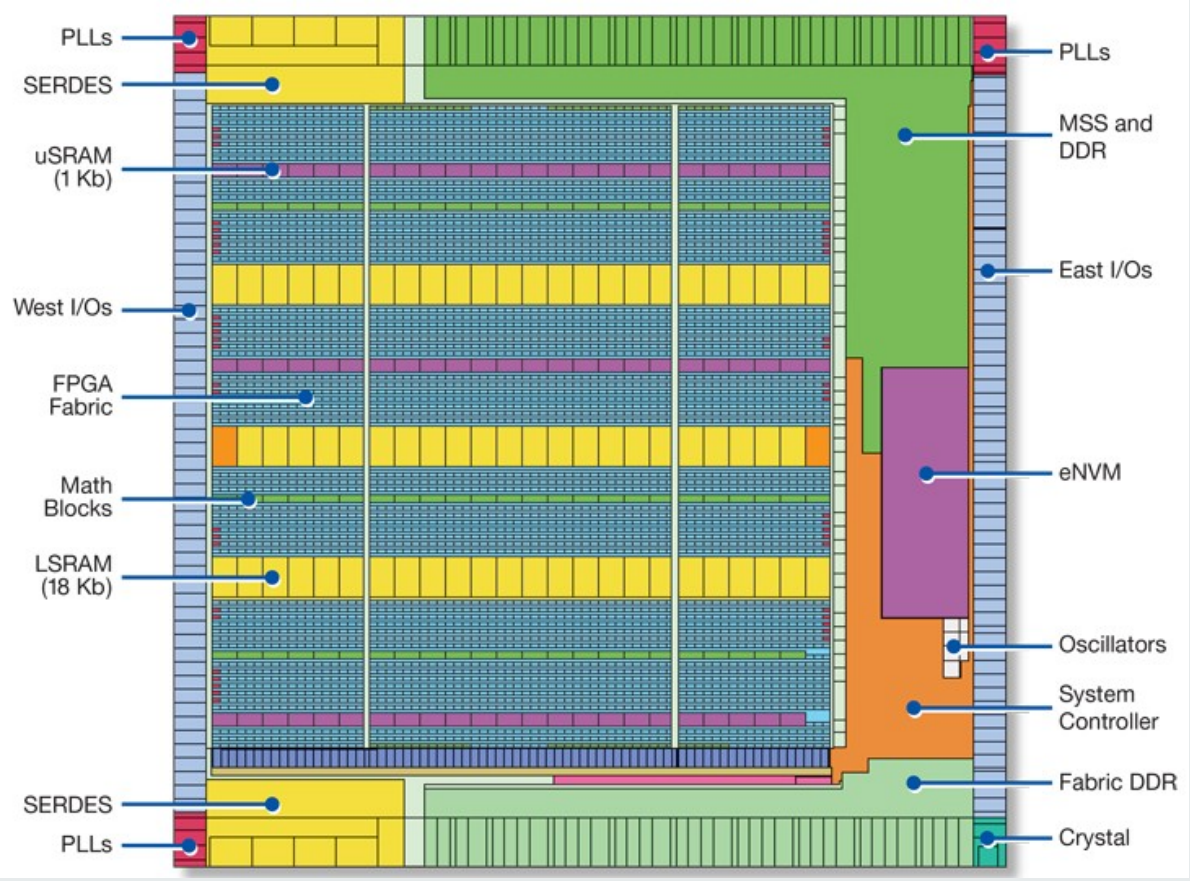
- FPGAs can be programmed to implement registers, register files or larger memories
- The LUTs can be used to implement larger register files or small memories
- FPGAs can be programmed to implement complex arithmetic units such as multipliers
- But, this consumes a lot of LUTs

TÉCNICO LISBOA

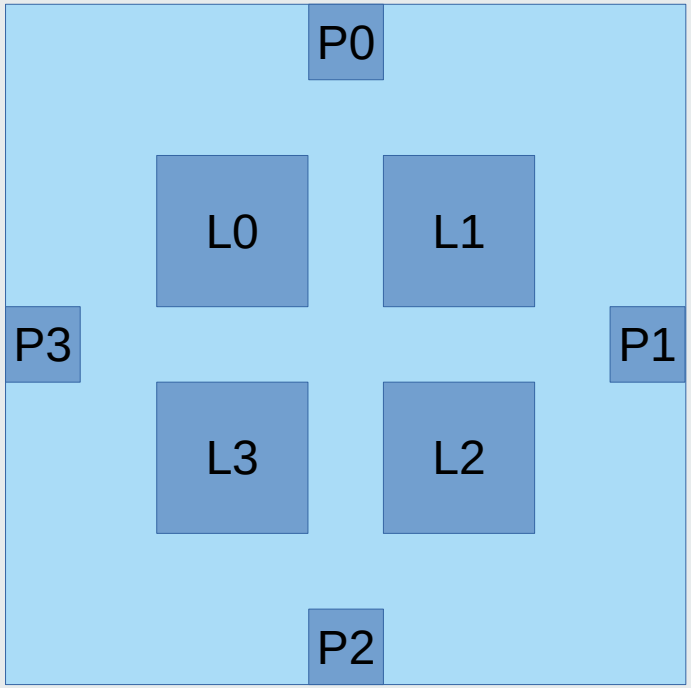
Heterogenous FPGAs

- Contain embedded RAMs
- Contain embedded DSP blocks
  - Multiply Accumulate units (MACC) and variations
    - Multiply only
    - Add only
    - Multiply an accumulate: mul + add
- Result: better silicon utilization leads to lower price products

FPGA layout

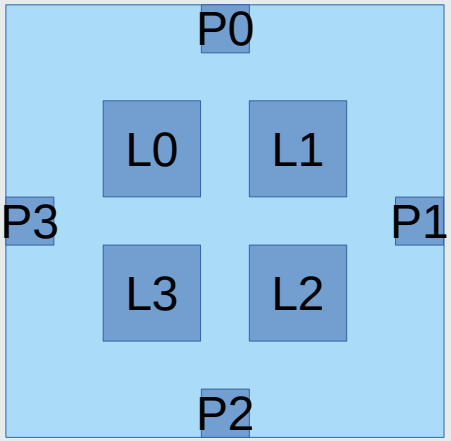

PLLs
SERDES
uSRAM
(1 Kb)
West I/Os
FPGA
Fabric
Math
Blocks
LSRAM
(18 Kb)
SERDES
PLLs
PLLs
MSS and
DDR
East I/Os
eNVM
Oscillators
System
Controller
Fabric DDR
Crystal

Example FPGA

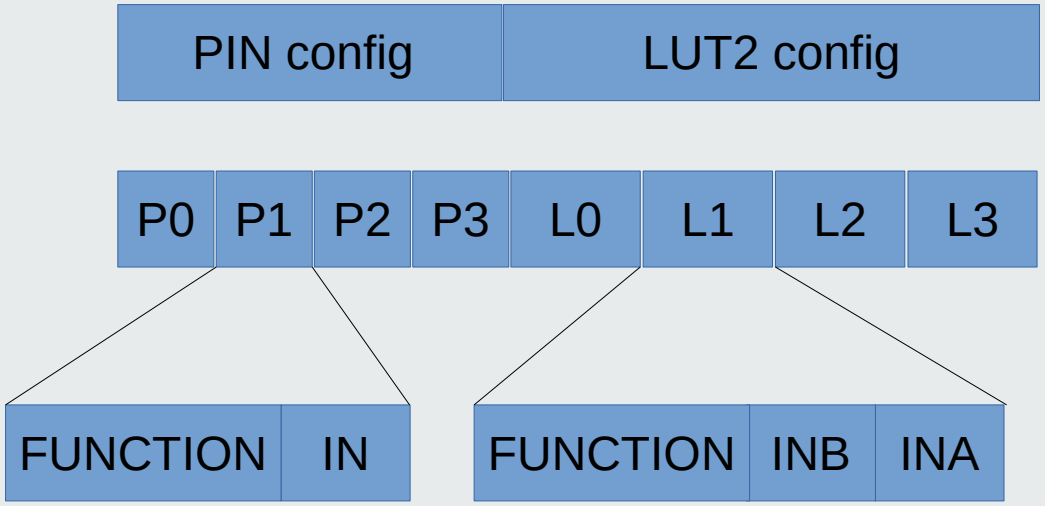


- P0..3: pins, programmable with the following word:
  - Bit 3 – function select bit: input (0) or output (1)
  - Bits 2:1 – select which pin or LUT output
  - Bit 0 – selects FPGA pins (0) or LUT outputs (1);
- L0..3: 2-input (A and B) LUTs, programmable with the following word:
  - Bits 9:6 – function select bits F(A,B)
  - Bits 5:3 – input B selection bits;
  - Bits 2:0 – input A selection bits;

Example FPGA



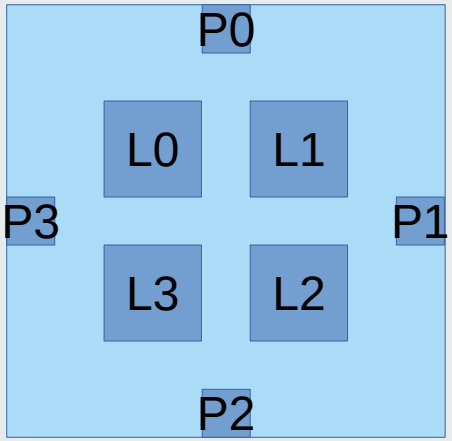
FPGA configuration (bitstream)



1 + 3 = 4 4 + 3 + 3 = 10

Total configuration bits: 4*4 + 4*10 = 56 bits

Example FPGA



Configure logic function Y = X1 X2 X3

P0..2 are inputs
Config: 0 - - - (0 selects input function, since it is not an output the selection bits are don't care)

P3 is output:
Config: 1 10 1 (1 selects output function, 10 selects L2, 1 selects LUTs)

L0 and L1 are AND2:
L0 config: 1000 01 0 00 0 (1000 selects function AND, 01 0 selects P1 and 00 0 select P0)
L1 config: 1000 10 0 00 1 (1000 selects function AND, 10 0 selects P2 and 00 1 selects L0)