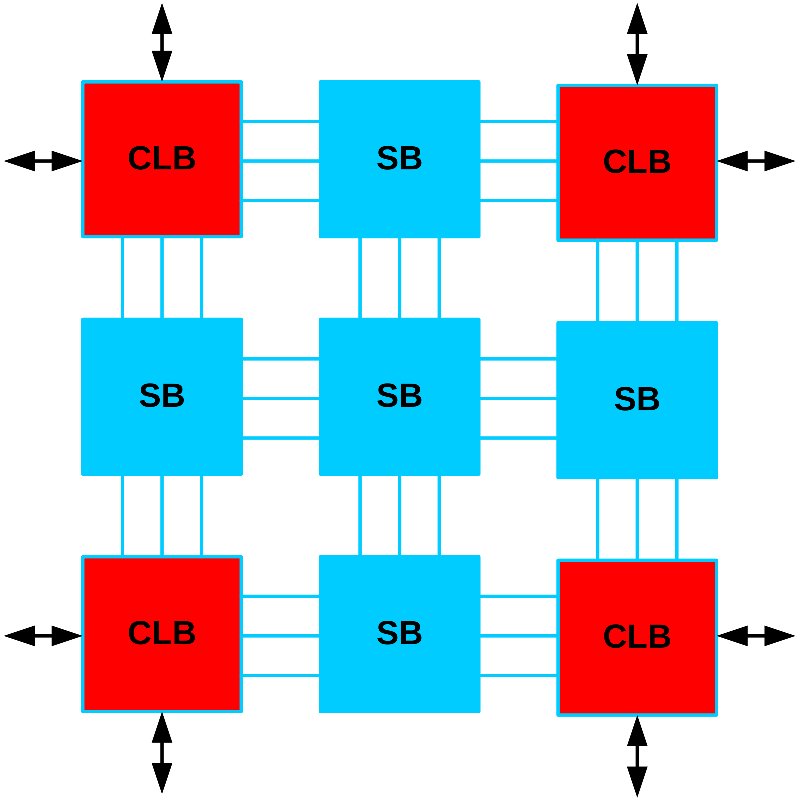
# Manufacturers

- Xilinx
- Altera
- Lattice
- Actel

# FPGA structure

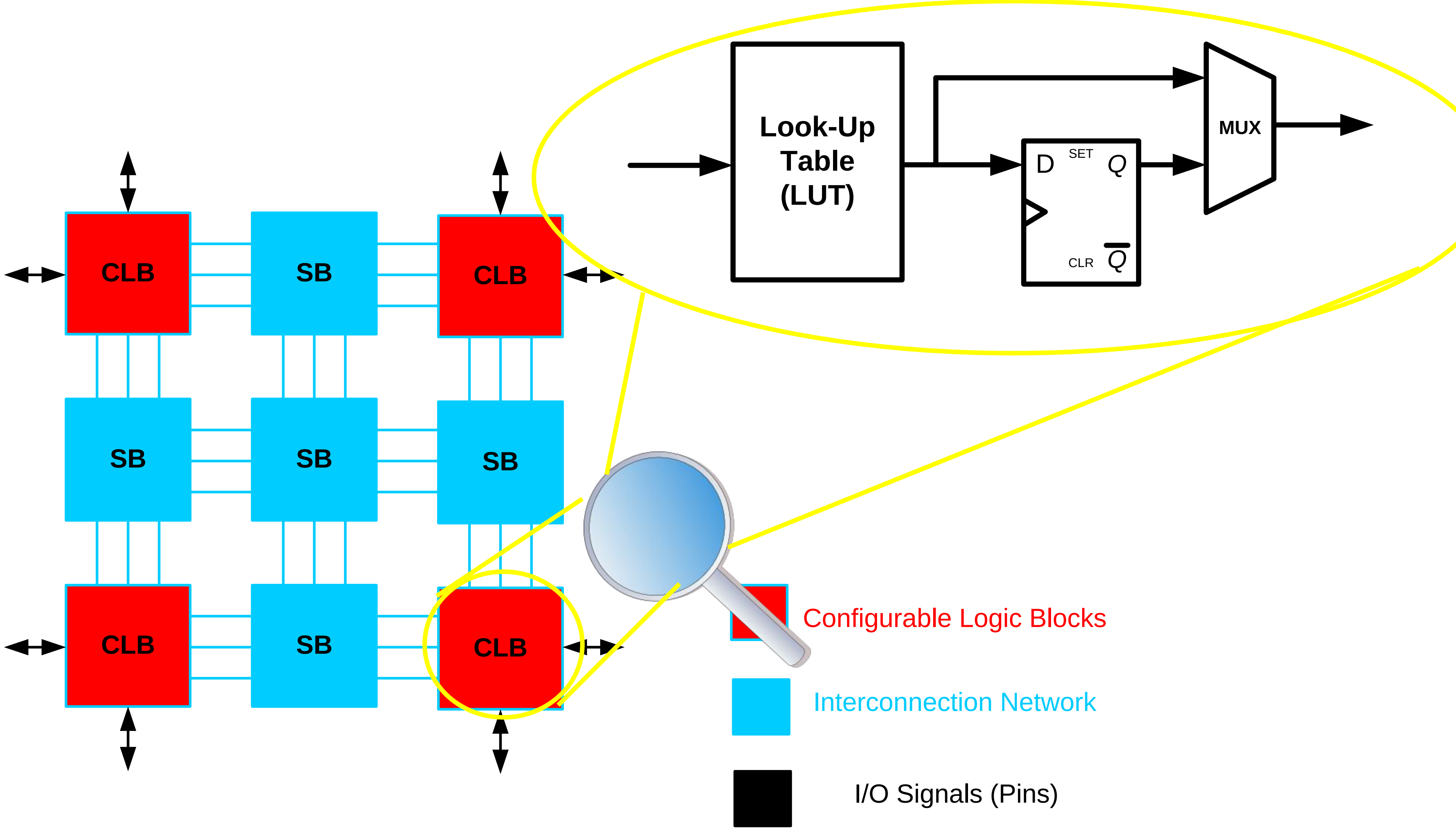


🟥 Configurable Logic Blocks

🟦 Interconnection Network

⬛ I/O Signals (Pins)

# Simplified CLB Structure

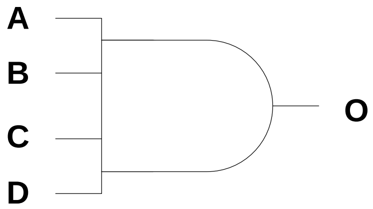


🟥 Configurable Logic Blocks

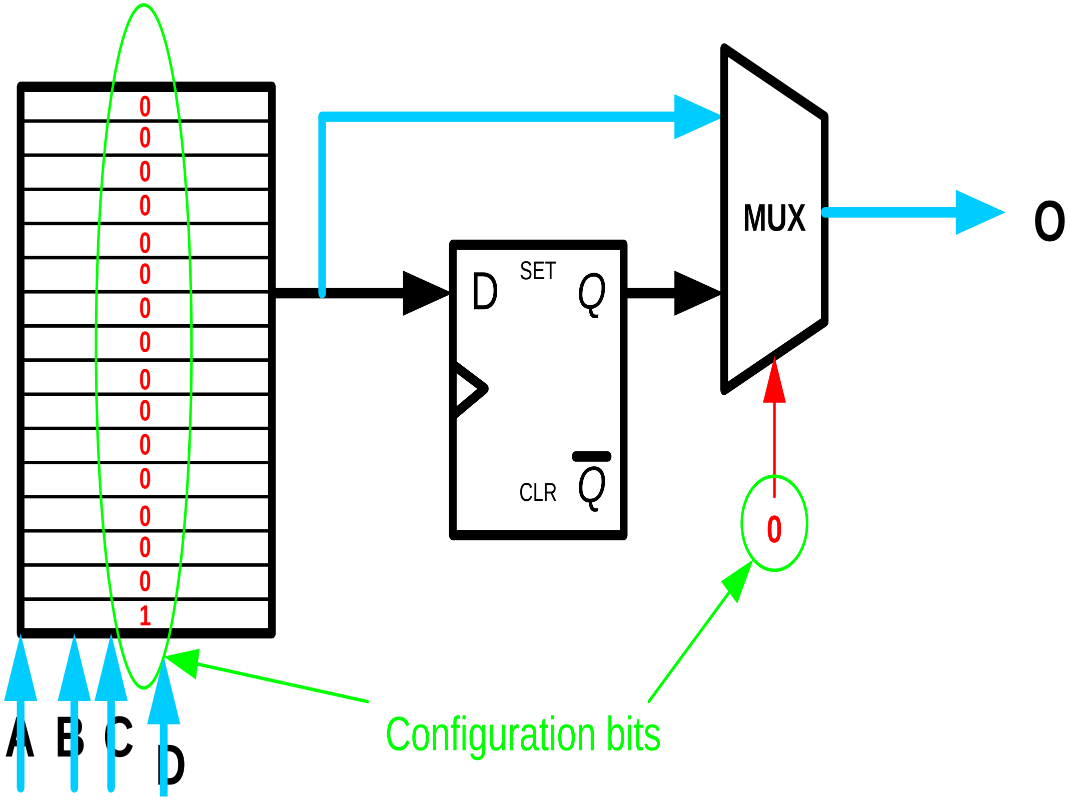
🟦 Interconnection Network

⬛ I/O Signals (Pins)

# Example: 4-input AND gate

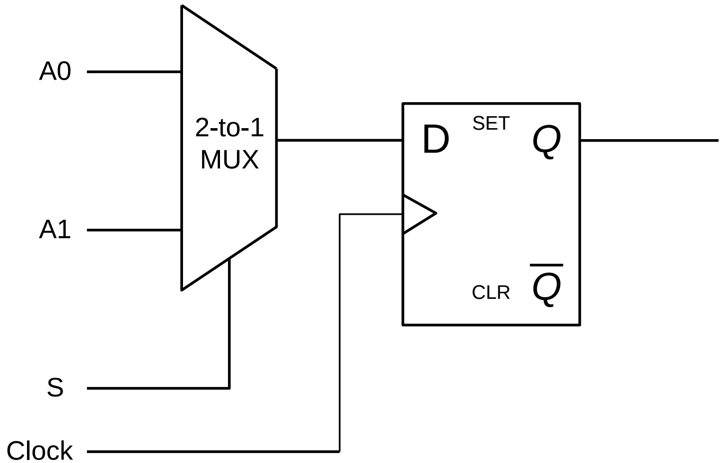


| A | B | C | D | O |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

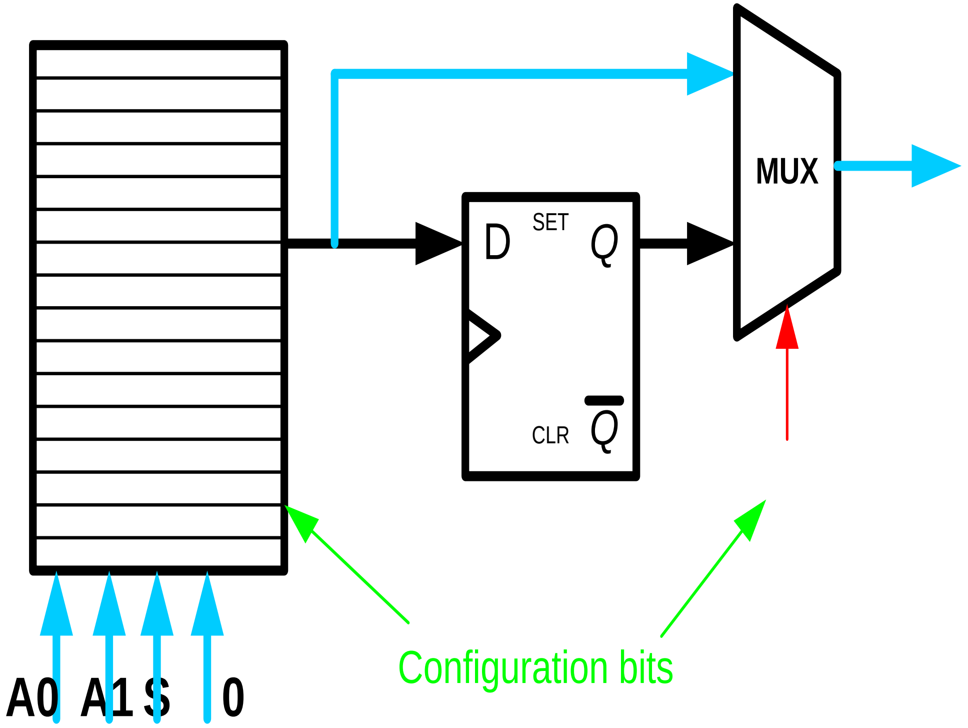


Configuration bits

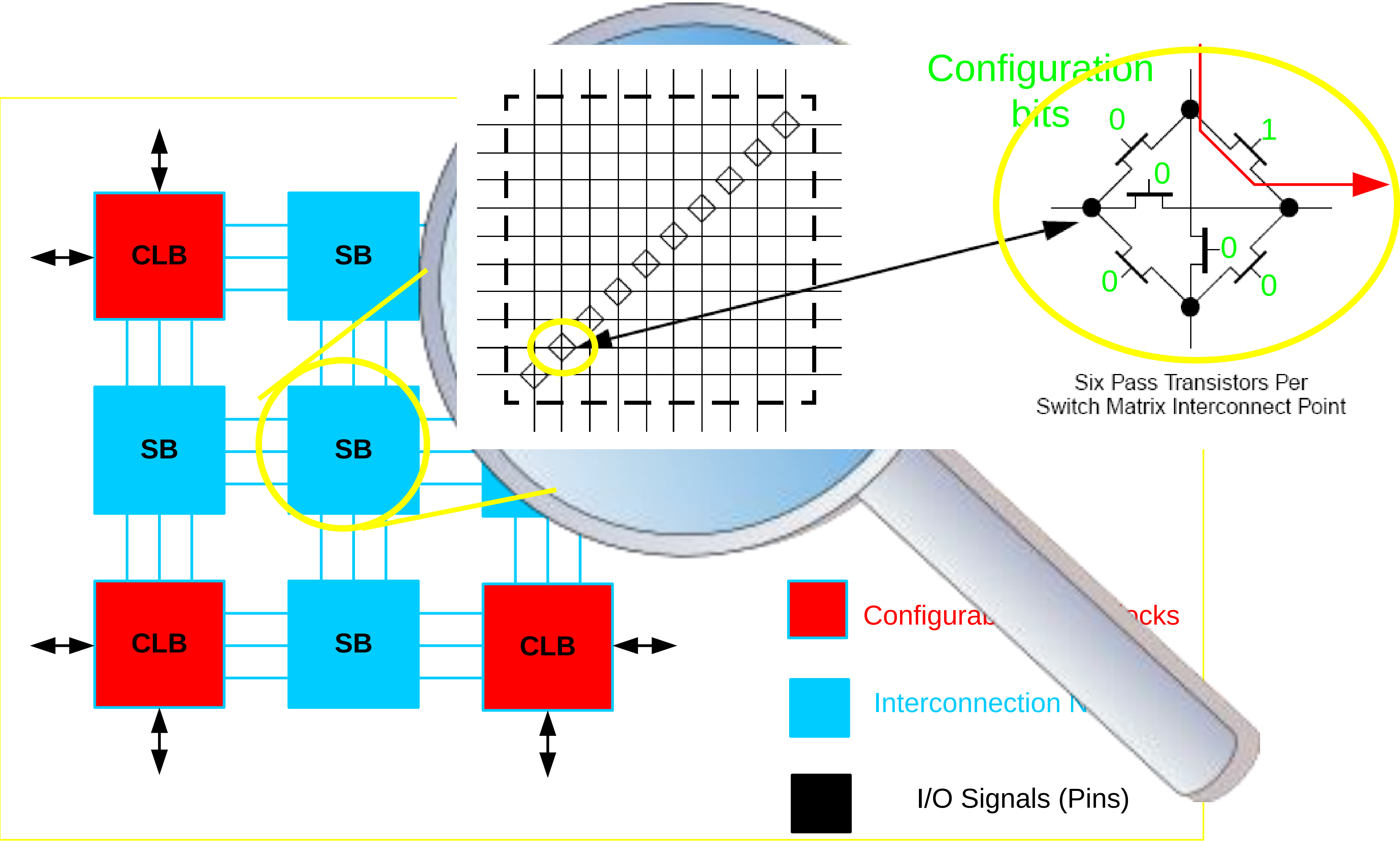
# Example 2: Find the configuration bits for the following circuit



| A0 | A1 | S |  |
|---|---|---|---|
| 0 | 0 | 0 |  |
| 0 | 0 | 1 |  |
| 0 | 1 | 0 |  |
| 0 | 1 | 1 |  |
| 1 | 0 | 0 |  |
| 1 | 0 | 1 |  |
| 1 | 1 | 0 |  |
| 1 | 1 | 1 |  |

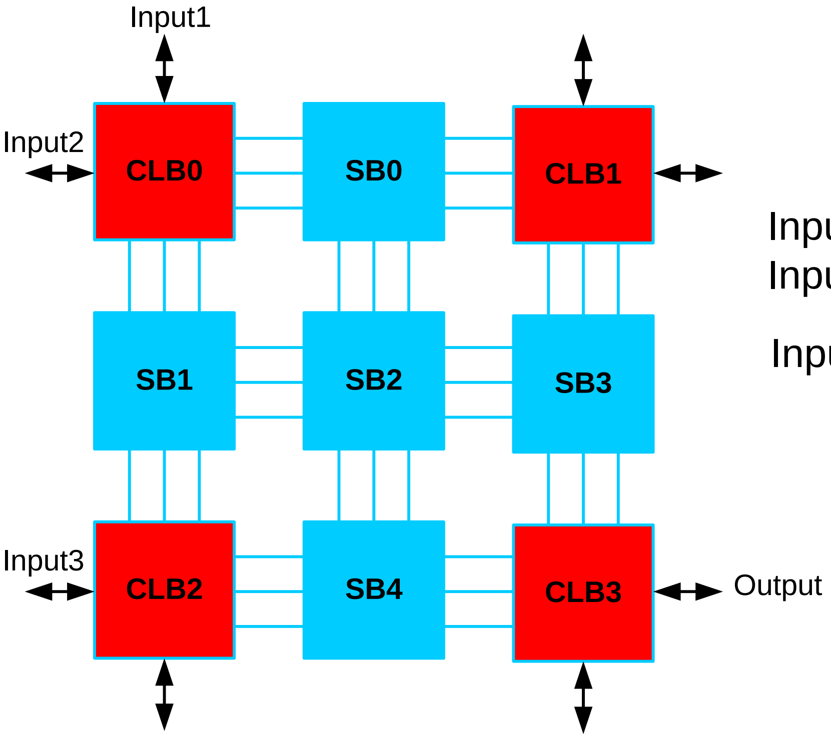


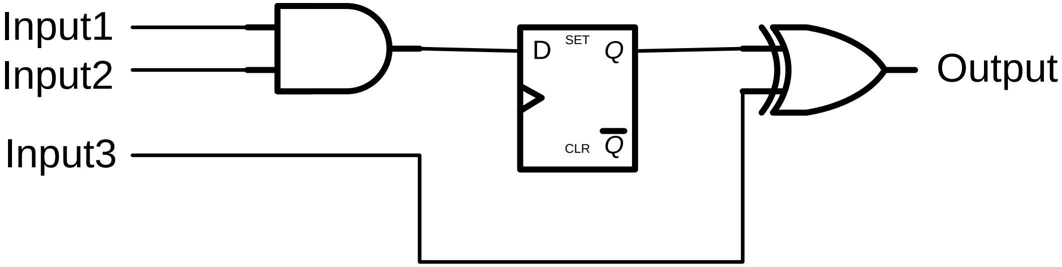
# Interconnection Network



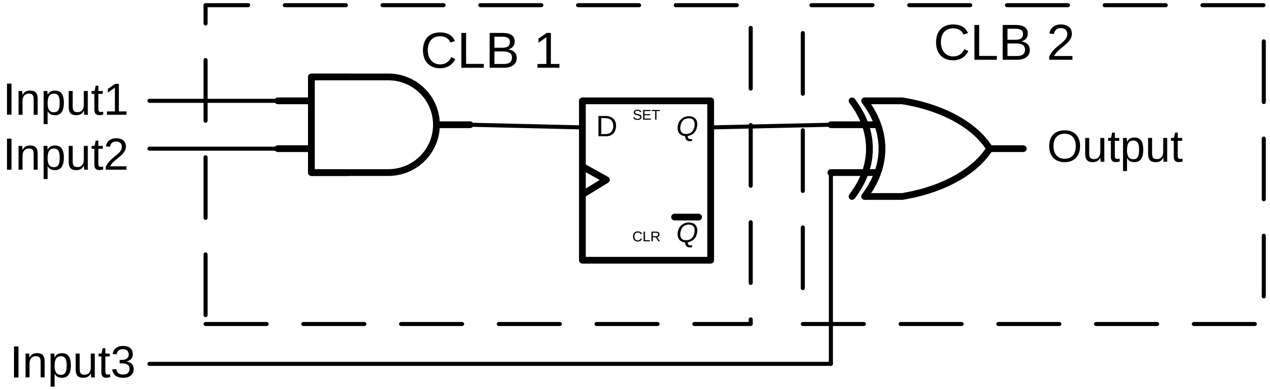
# Example 3

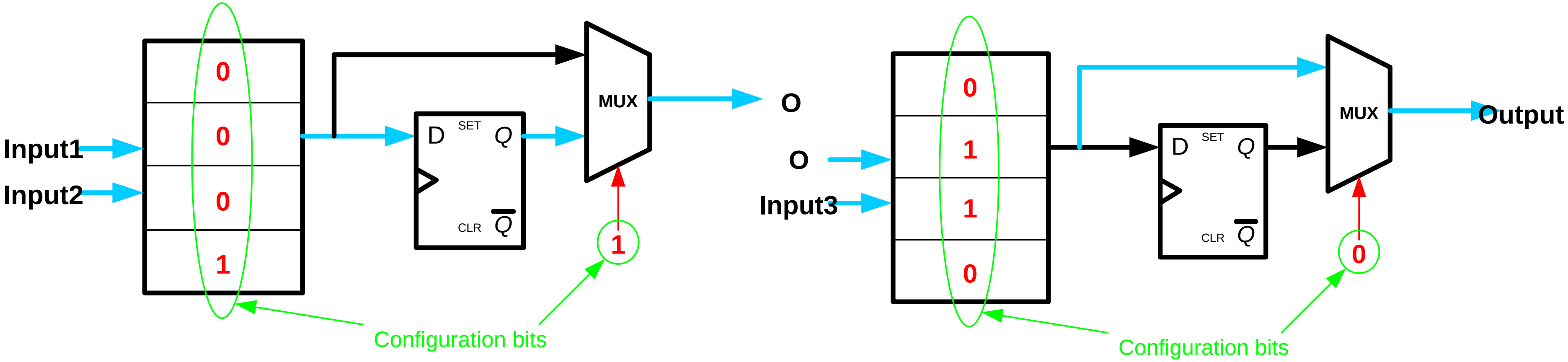
- Determine the configuration bits for the following circuit implementation in a 2x2 FPGA, with I/O constraints as shown in the following figure. Assume 2-input LUTs in each CLB.



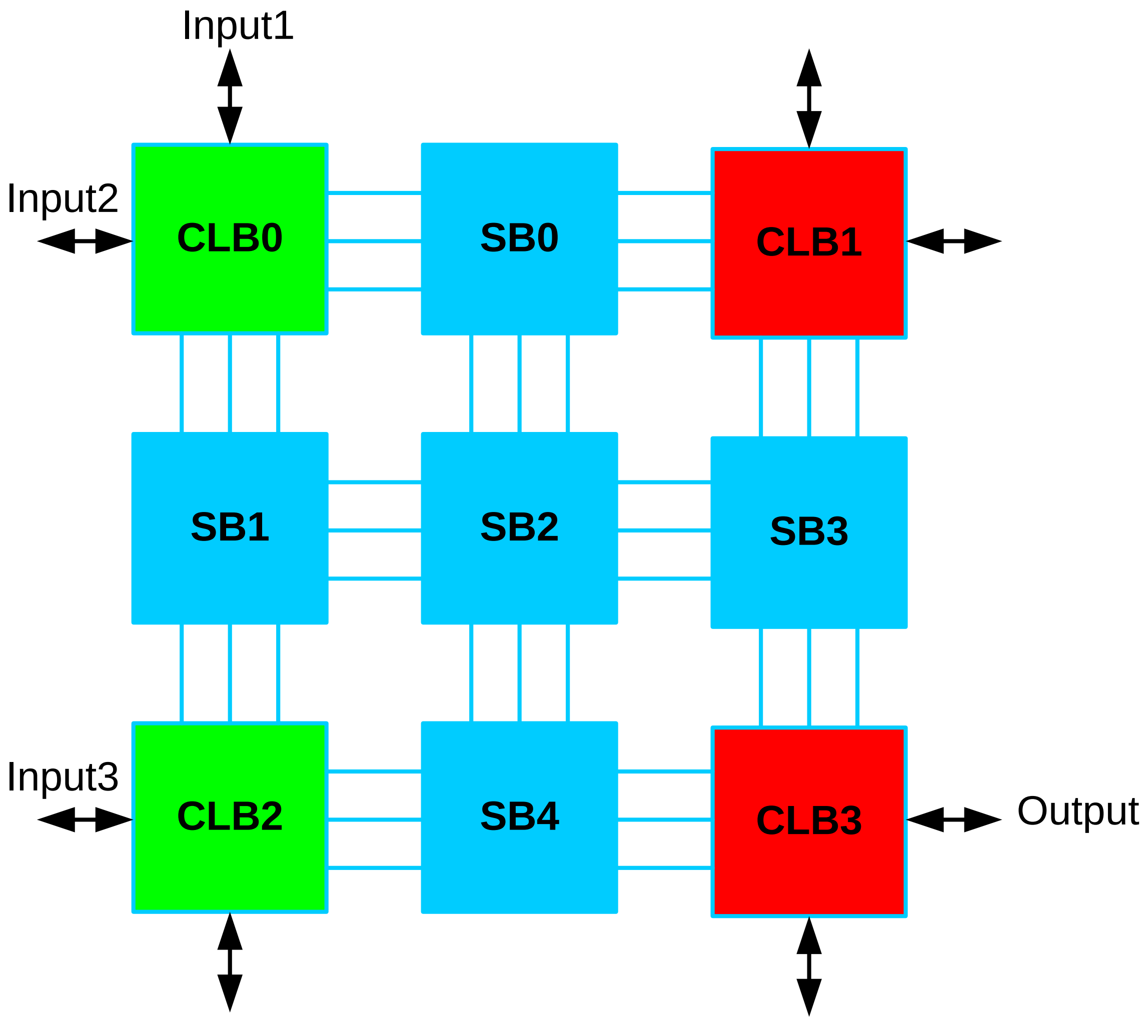


# CLBs required

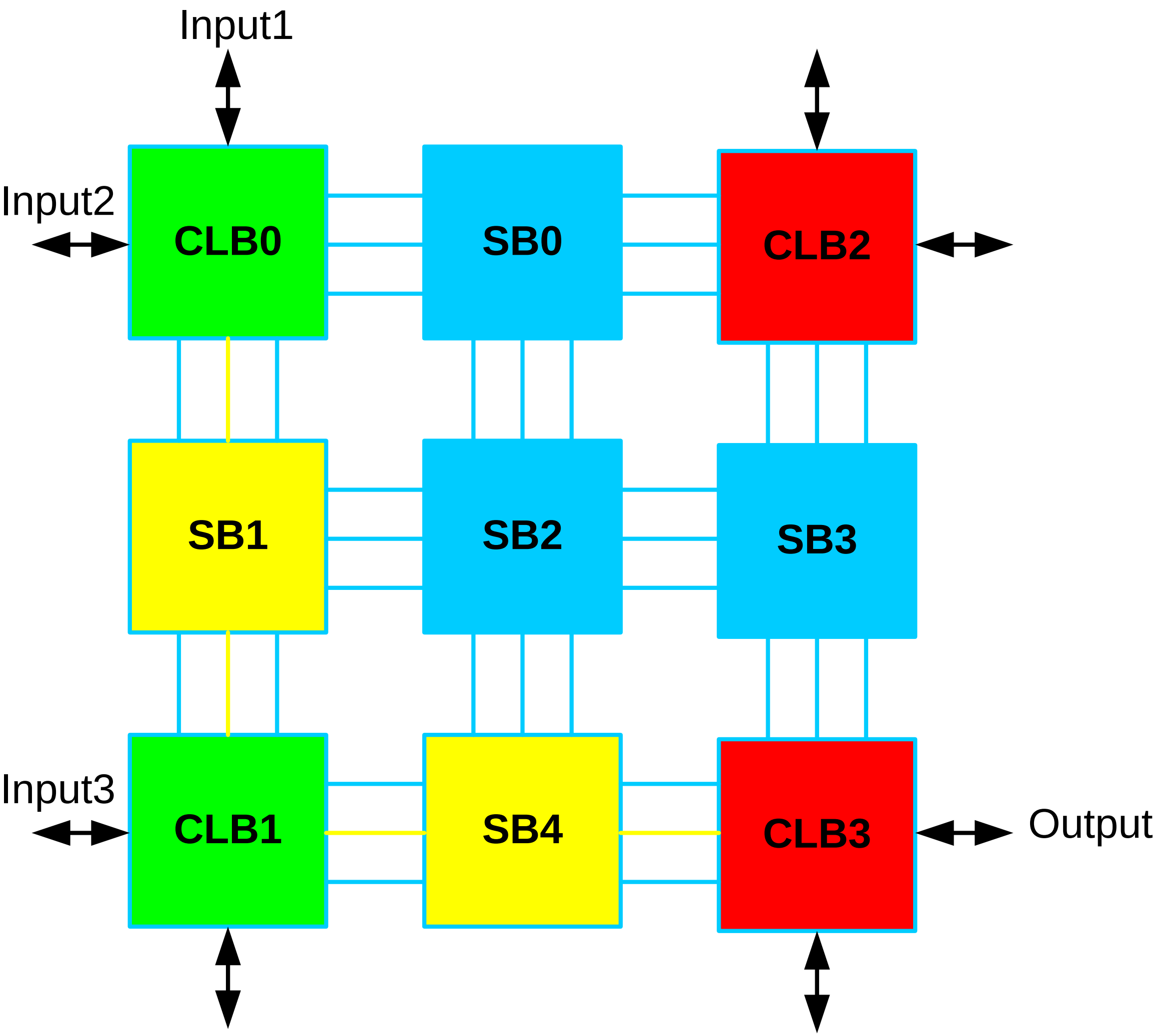




# Placement: Select CLBs

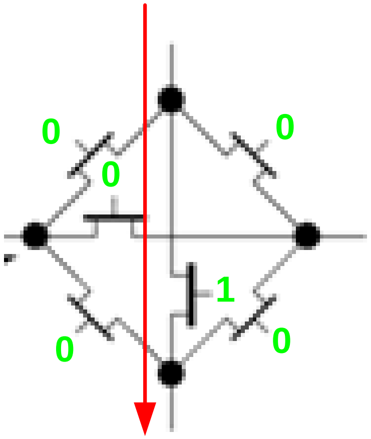


# Routing: Select path



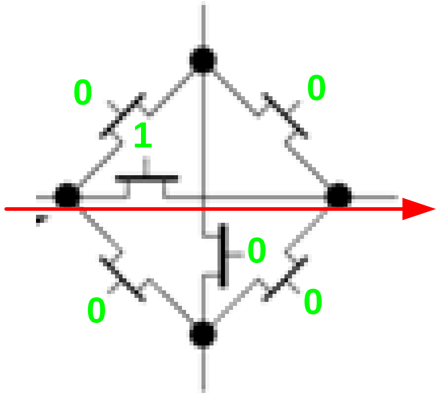
**SB1**

Configuration bits



**SB4**

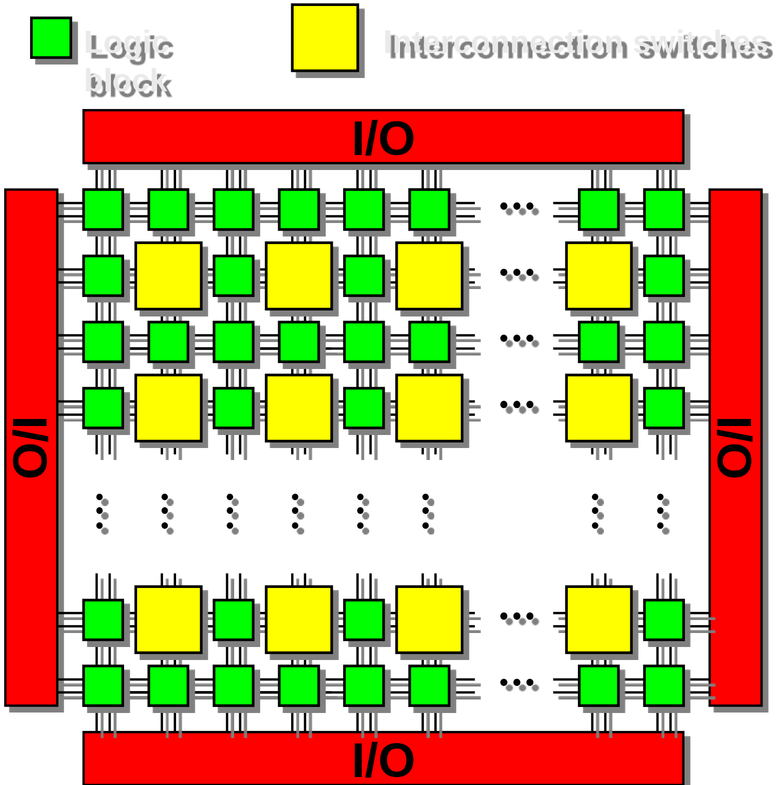
Configuration bits

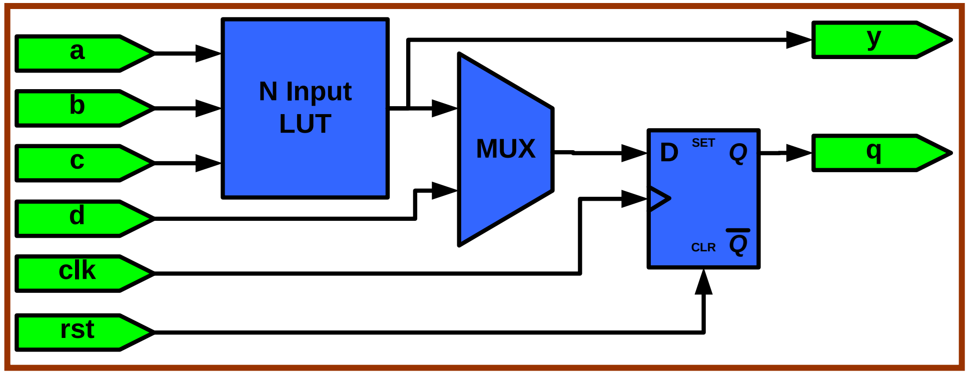


# Configuration Bitstream

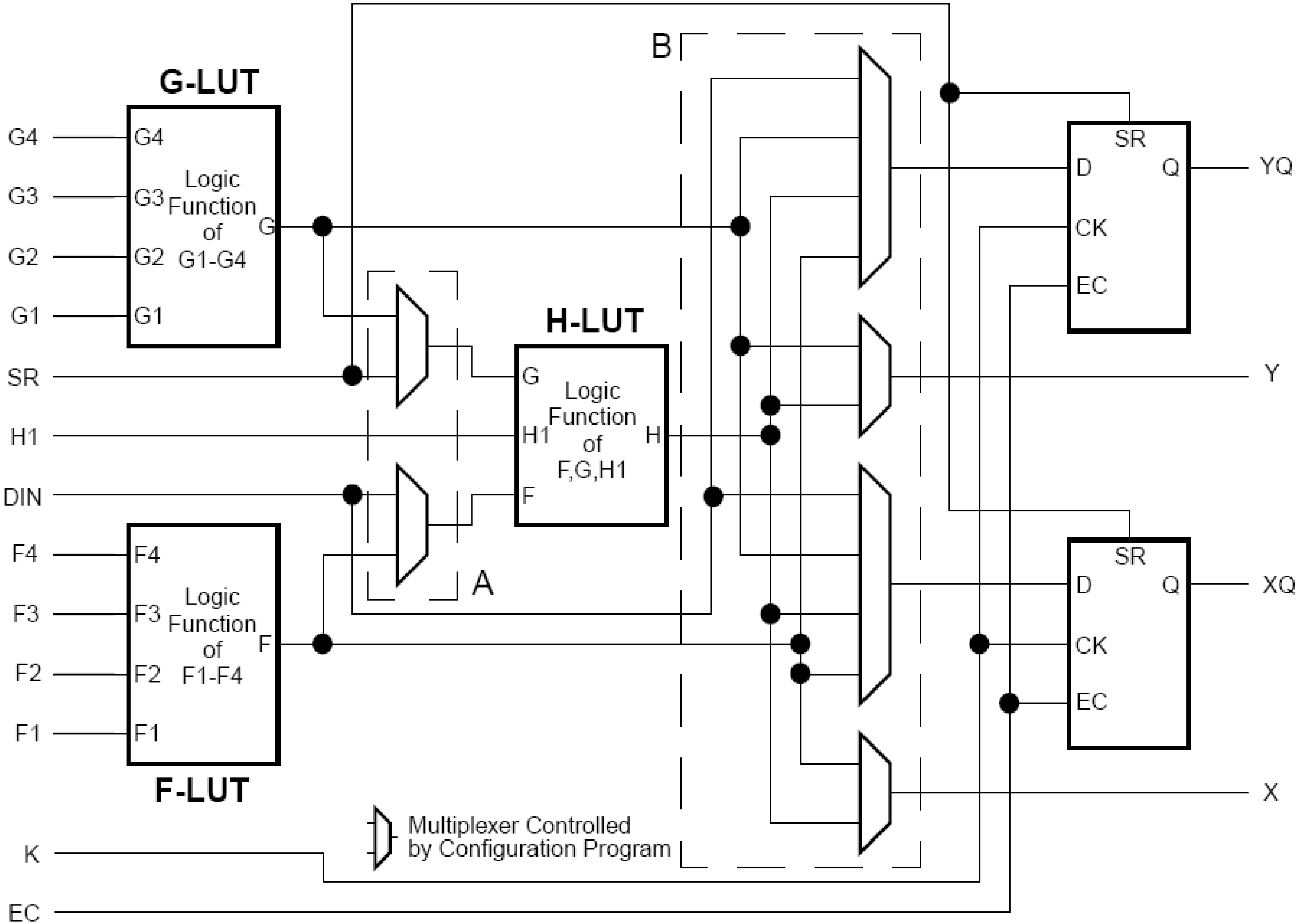
- The configuration bitstream must include ALL CLBs and SBs, even unused ones
- CLB0: 00011
- CLB1: 01100
- CLB2: XXXXX
- CLB3: XXXXX
- SB0: 000000
- SB1: 000010
- SB2: 000000
- SB3: 000000
- SB4: 000001

- Programmable logic blocks (Logic Element “LE”)
  Implement combinatorial and sequential logic. Based on LUT and DFF.
- Programmable I/O blocks
  Configurable I/Os for external connections supports various voltages and tri-states.
- Programmable interconnect
  Wires to connect inputs , outputs and logic blocks.
  - clocks
  - short distance local connections
  - long distance connections across chip

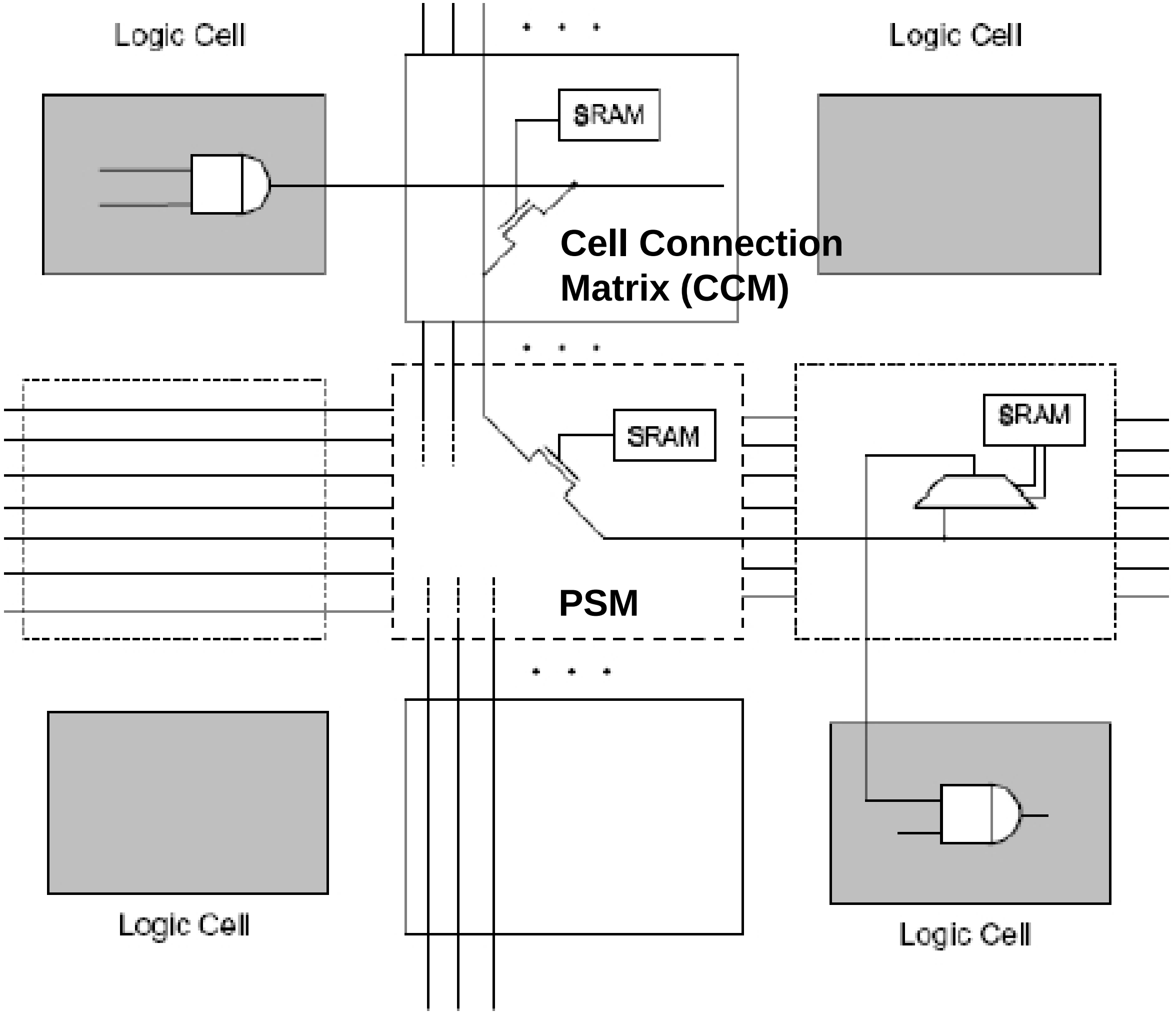




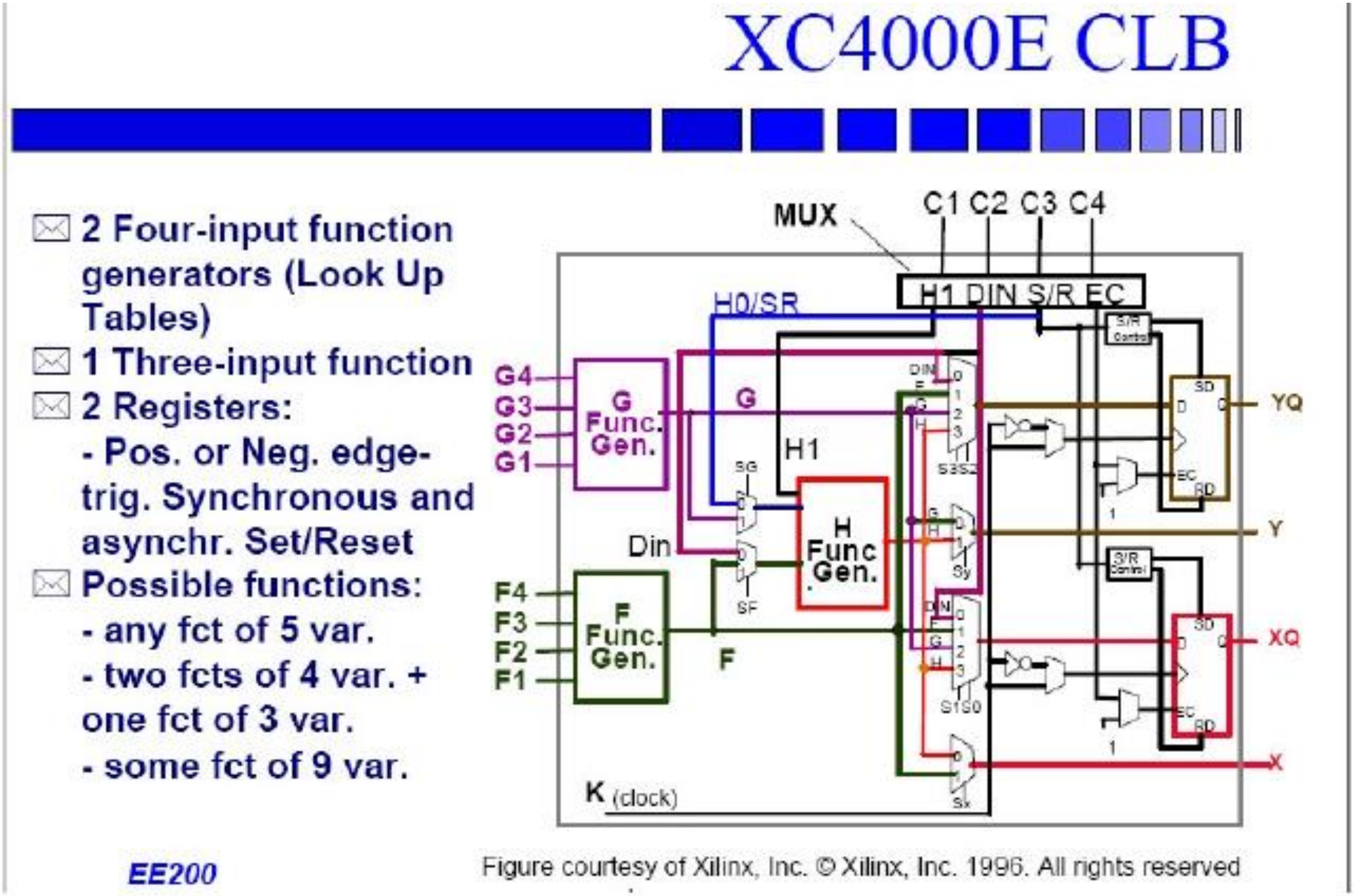
# Realistic FPGA CLB: Xilinx



# SRAM-type FPGA Interconnect Architecture (contd)



# Configuration Logic Block (CLB)



## XC4000E CLB

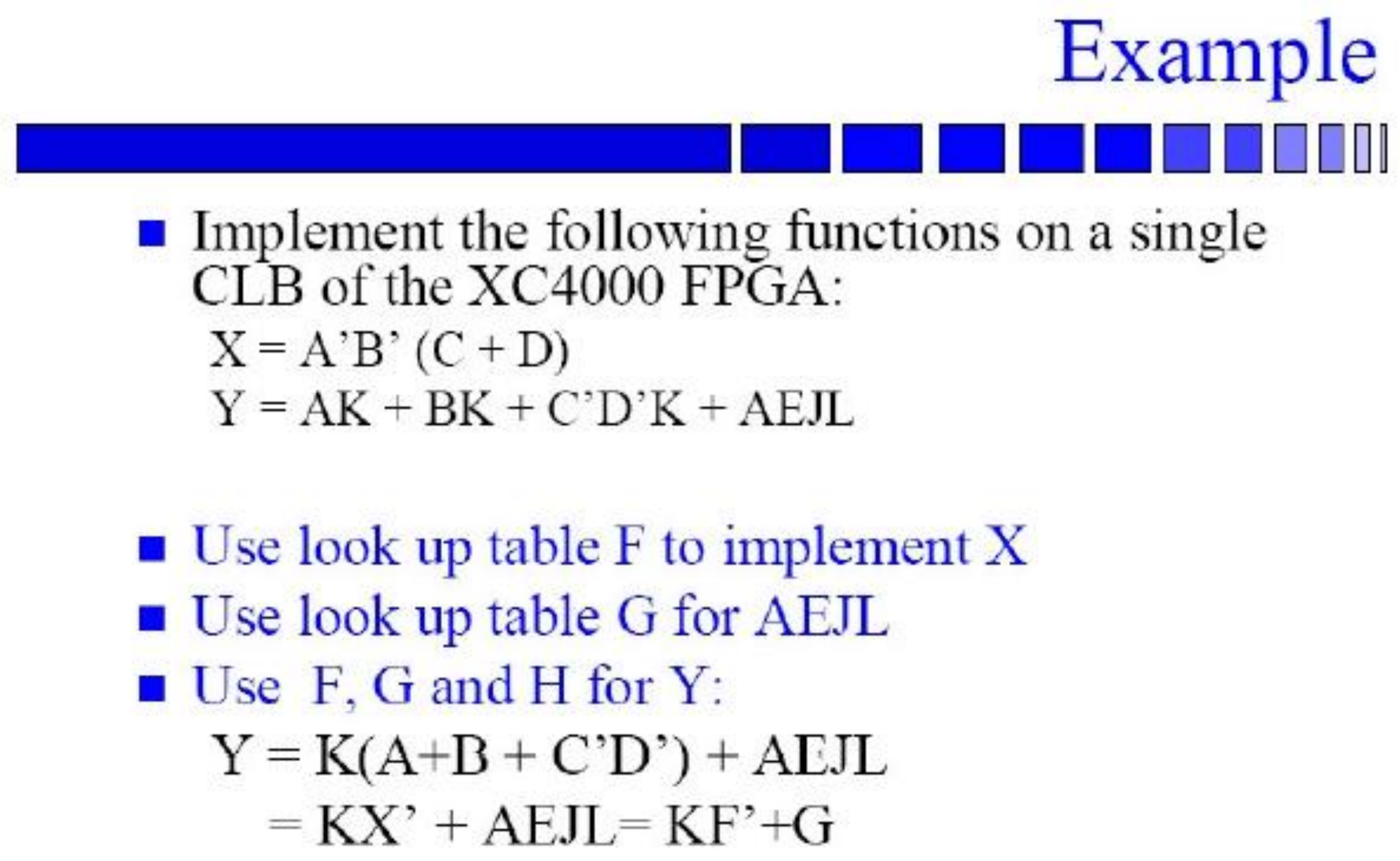
- 2 Four-input function generators (Look Up Tables)
- 1 Three-input function
- 2 Registers:
  - Pos. or Neg. edge-trig. Synchronous and asynchr. Set/Reset
- Possible functions:
  - any fct of 5 var.
  - two fcts of 4 var. + one fct of 3 var.
  - some fct of 9 var.

EE200

Figure courtesy of Xilinx, Inc. © Xilinx, Inc. 1996. All rights reserved

- 5-i/p function implemented using G, F and H LUTs (Look Up Tables) using *Shannon's Expansion*: *p(a,b,c,d,e) = a p(1, b, c, d, e) + a' p(0, b, c, d, e) = a q(b,c,d,e) + a'r(b,c,d,e).* q( ) impl. using LUT G, r impl. using LUT F and *p=ag + a'h* impl. using LUT H
- The LUT o/ps can go through a FF (for seq. ckt design) or bypass it for a combinational o/p
- This is called *technology mapping*: mapping the logic to CLB logic components

# Mapping



## Example

- Implement the following functions on a single CLB of the XC4000 FPGA:

  $X = A'B'(C + D)$

  $Y = AK + BK + C'D'K + AEJL$

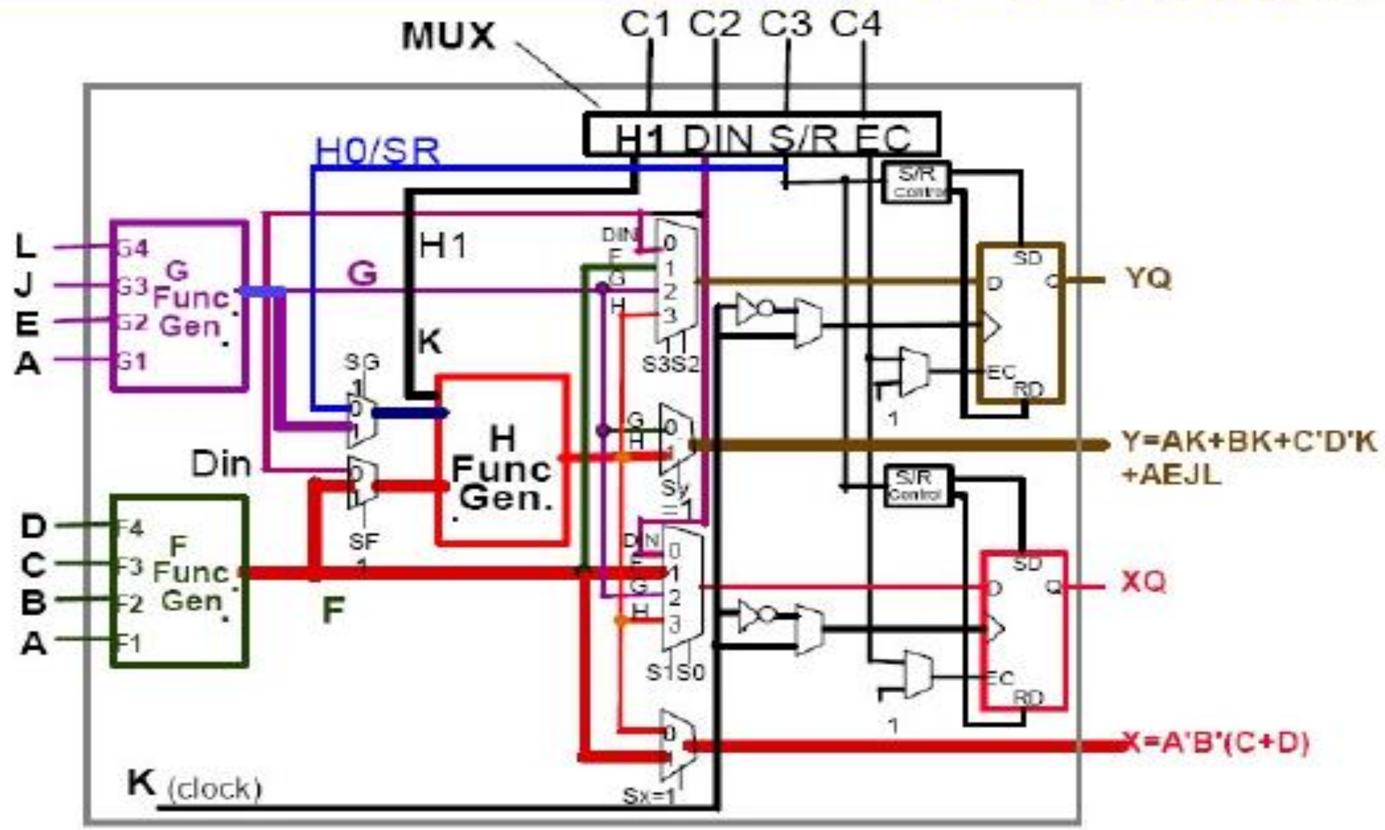
- Use look up table F to implement X
- Use look up table G for AEJL
- Use F, G and H for Y:

  $Y = K(A+B + C'D') + AEJL$

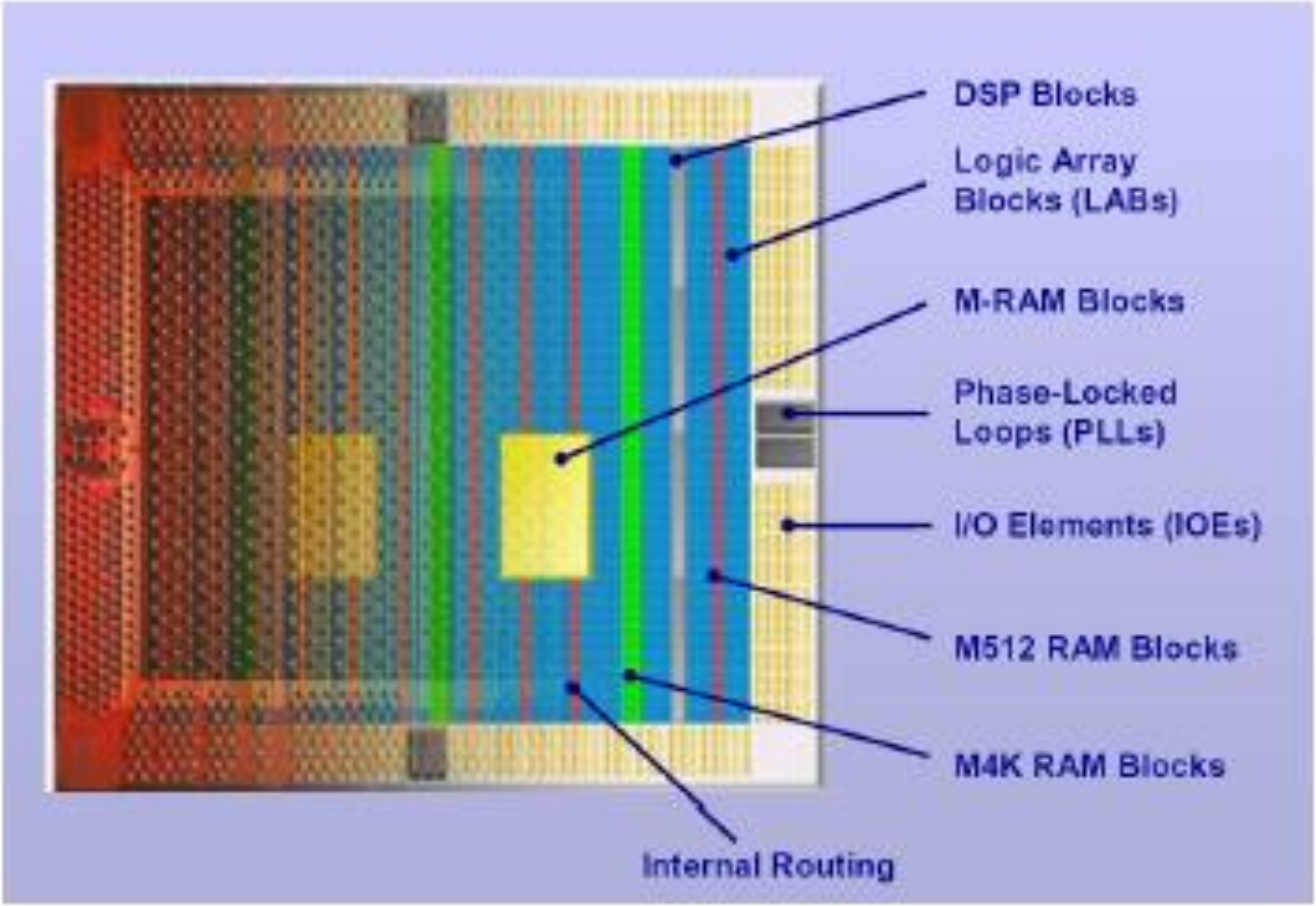
  $= KX' + AEJL = KF'+G$

# Programming a CLB (contd)

## Example



# Components of Modern FPGAs



# FPGA EDA Tools

- Must provide a design environment based on digital design concepts and components (gates, flip-flops, MUXs, etc.)
- Must hide the complexities of placement, routing and bitstream generation from the user. Manual placement, routing and bitstream generation is infeasible for practical FPGA array sizes and circuit complexities.

# FPGA EDA Tools

- High level Description of Logic Design
  - Hardware Description Language (Textual)

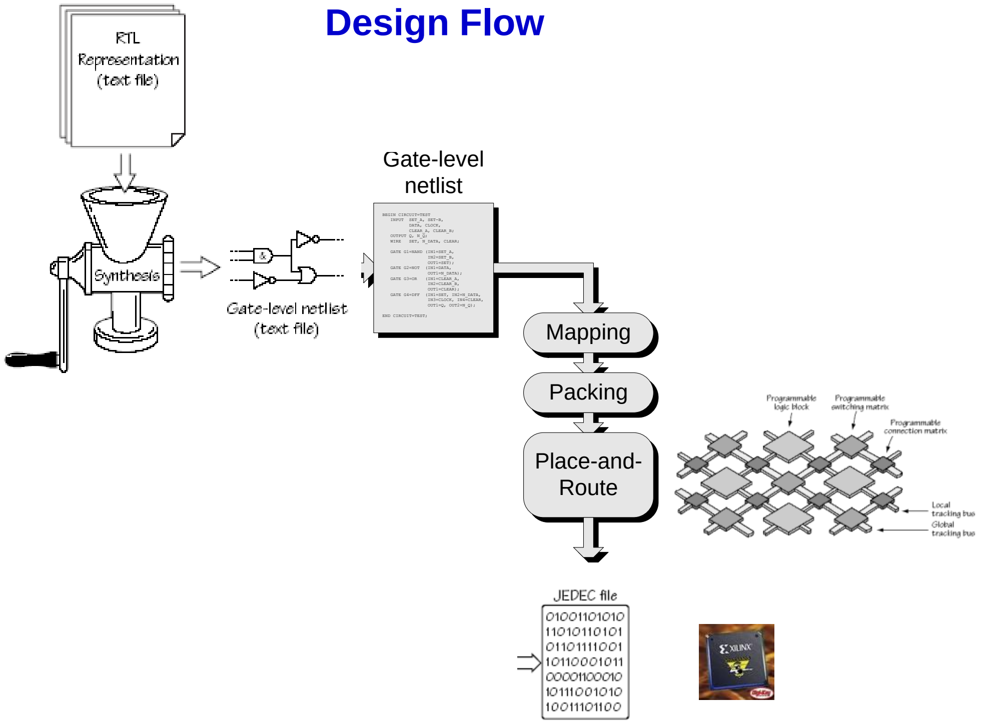
- Compile (Synthesise) into Netlist.
- Boolean Logic Gates.

- Target FPGA Fabric
  - Mapping
  - Routing
- Bit File for FPGA

- Commercial CAE Tools (Complex & Expensive)

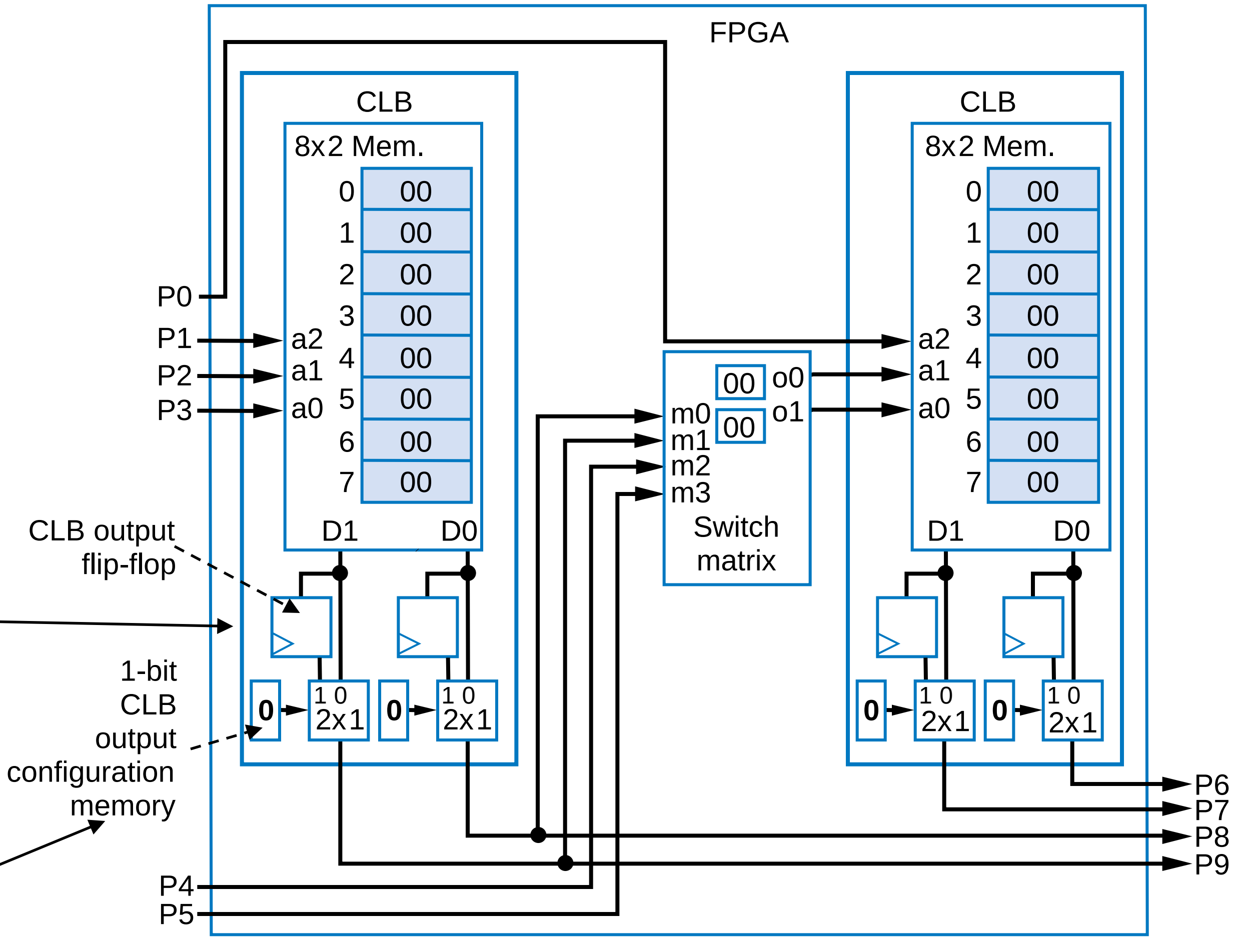
- Logic Simulation

**Design Flow**



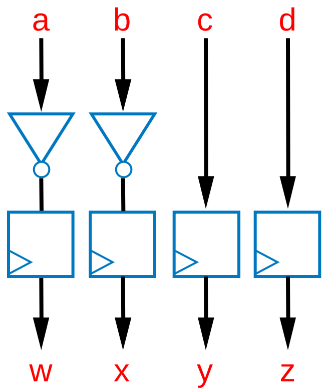
# FPGA Internals: Configurable Logic Blocks (CLBs)

- LUTs can only implement combinational logic
- Need flip-flops to implement sequential logic
- Add flip-flop to each LUT output
  - Configurable Logic Block (CLB)
    - LUT + flip-flops
  - Can program CLB outputs to come from flip-flops or from LUTs directly



a

# FPGA Internals: Sequential Circuit Example using CLBs



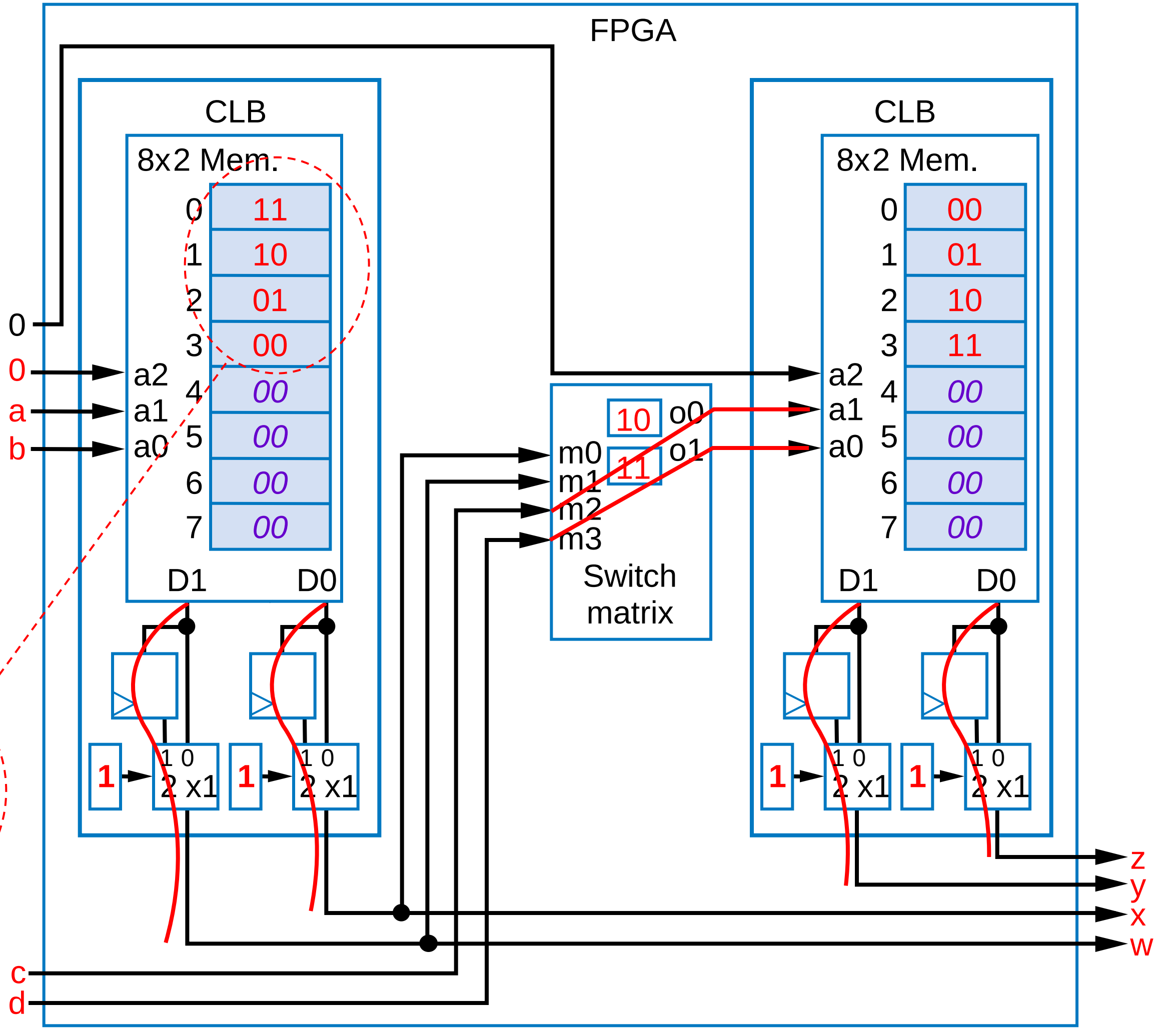
(**a**)

Left lookup table

| a2 | a1 | a0 | D1 | D0 |
|---|---|---|---|---|
| 0 | a | b | w=a' | x=b' |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 |

below unused

(**b**)

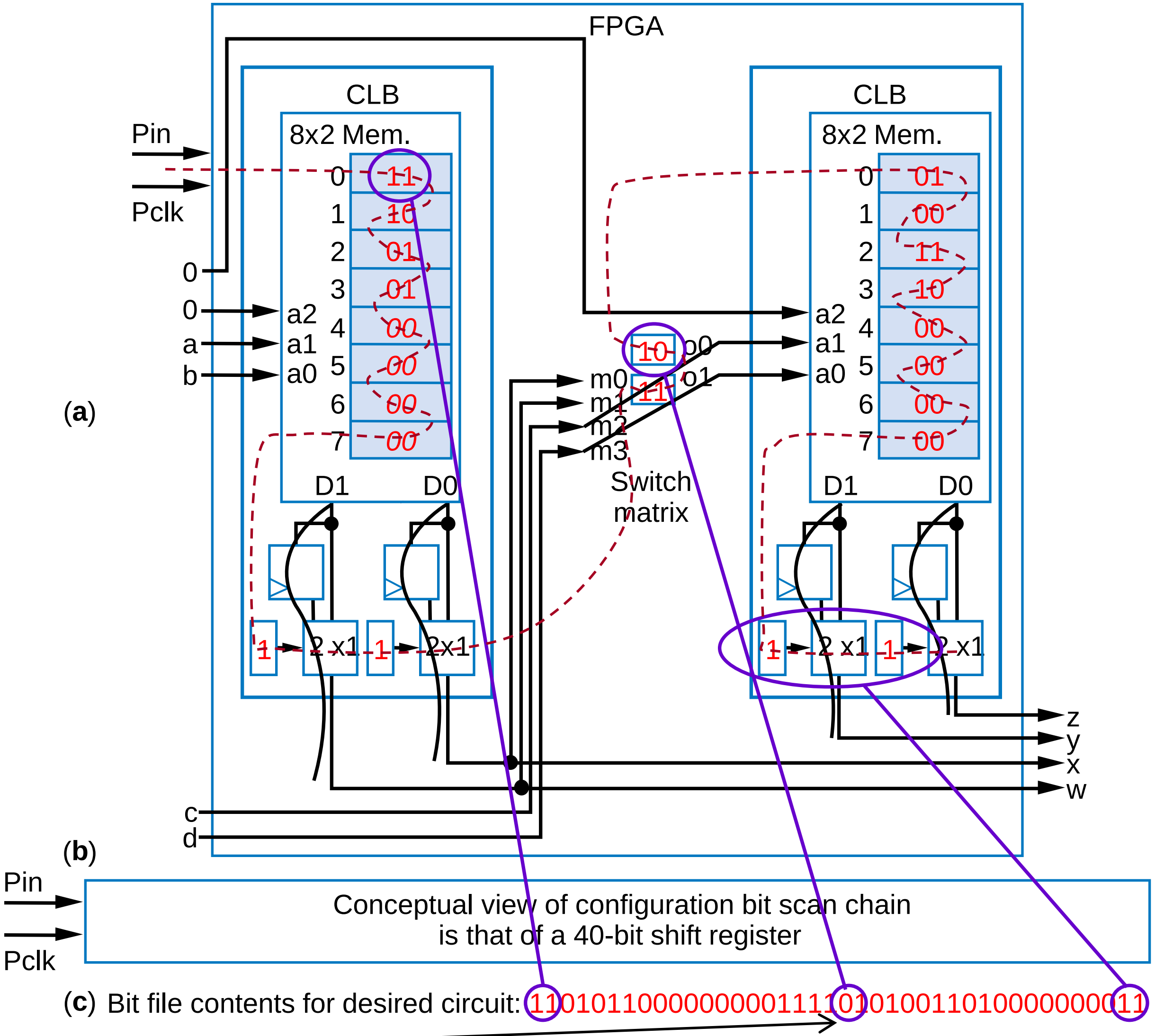


(**c**)

a

# FPGA Internals: Programming an FPGA

- All configuration memory bits are connected as one big shift register
  - Known as scan chain
- Shift in "bit file" of desired circuit



Conceptual view of configuration bit scan chain is that of a 40-bit shift register

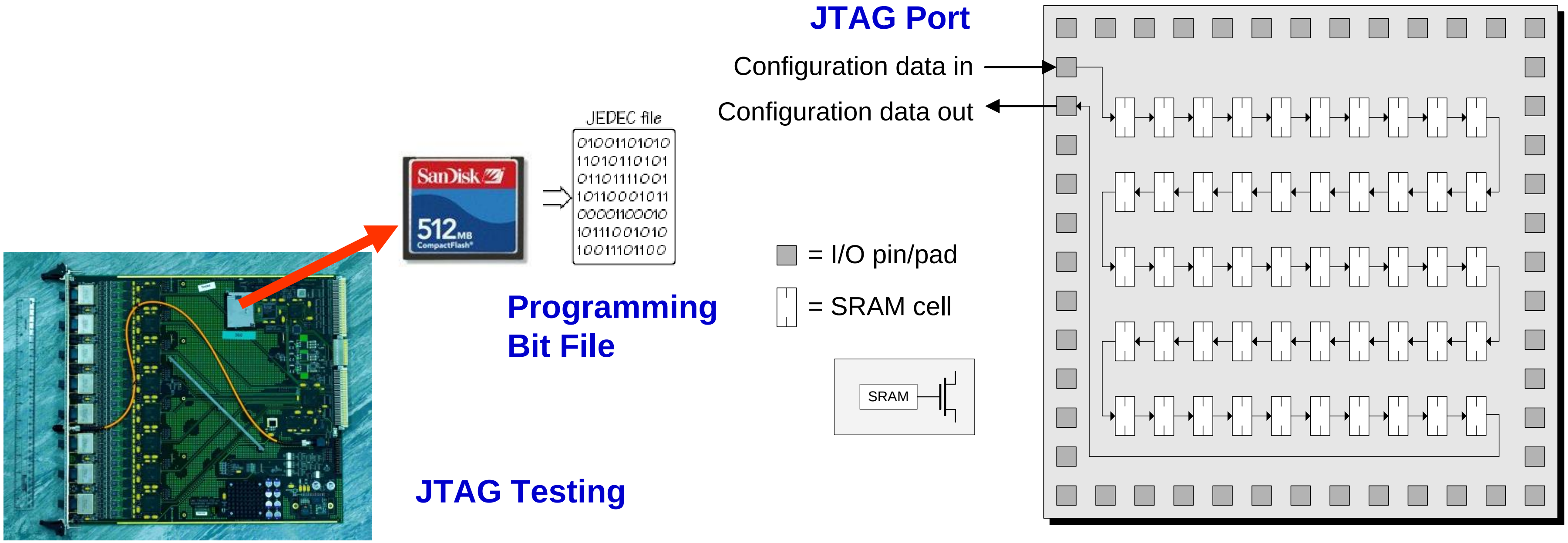
(**c**) Bit file contents for desired circuit: 1101011000000000111101010011010000000011

a

*This isn't wrong. Although the bits appear as "10" above, note that the scan chain passes through those bits from right to left – so "01" is correct here.*

# Configuring an FPGA

- Millions of SRAM cells holding LUTs and Interconnect Routing
- Volatile Memory. Loses configuration when board power is turned off.
- Keep Bit Pattern describing the SRAM cells in non-Volatile Memory e.g. ROM or Digital Camera card
- Configuration takes ~ secs



JTAG Port

Configuration data in

Configuration data out

Programming Bit File

= I/O pin/pad

= SRAM cell

JTAG Testing

# Summary

- Programmable Logic Devices
  - Basics
- Field Programmable Gate Arrays (FPGAs)
  - Architecture

- Design Flow
  - Hardware Description Languages
  - Design Tools