Permission to Use Conditions

1. Permission is granted to copy and distribute this slide set for educational purposes only, provided that the complete bibliographic citation and following credit line is included: “Hardware Security Slides by F. Gebali. ©2024. Gebali”.
2. Permission is granted to alter and distribute this material provided that the following credit line is included: “Adapted from Hardware Security Slides by F. Gebali. ©2024. Gebali"
3. This material may not be copied or distributed for commercial purposes without express written permission of the copyright holder.

ECE 448/548 Cyber-System Security

VLSI Technology

F. Gebali

EOW 433

Office Phone: 250-721-6509

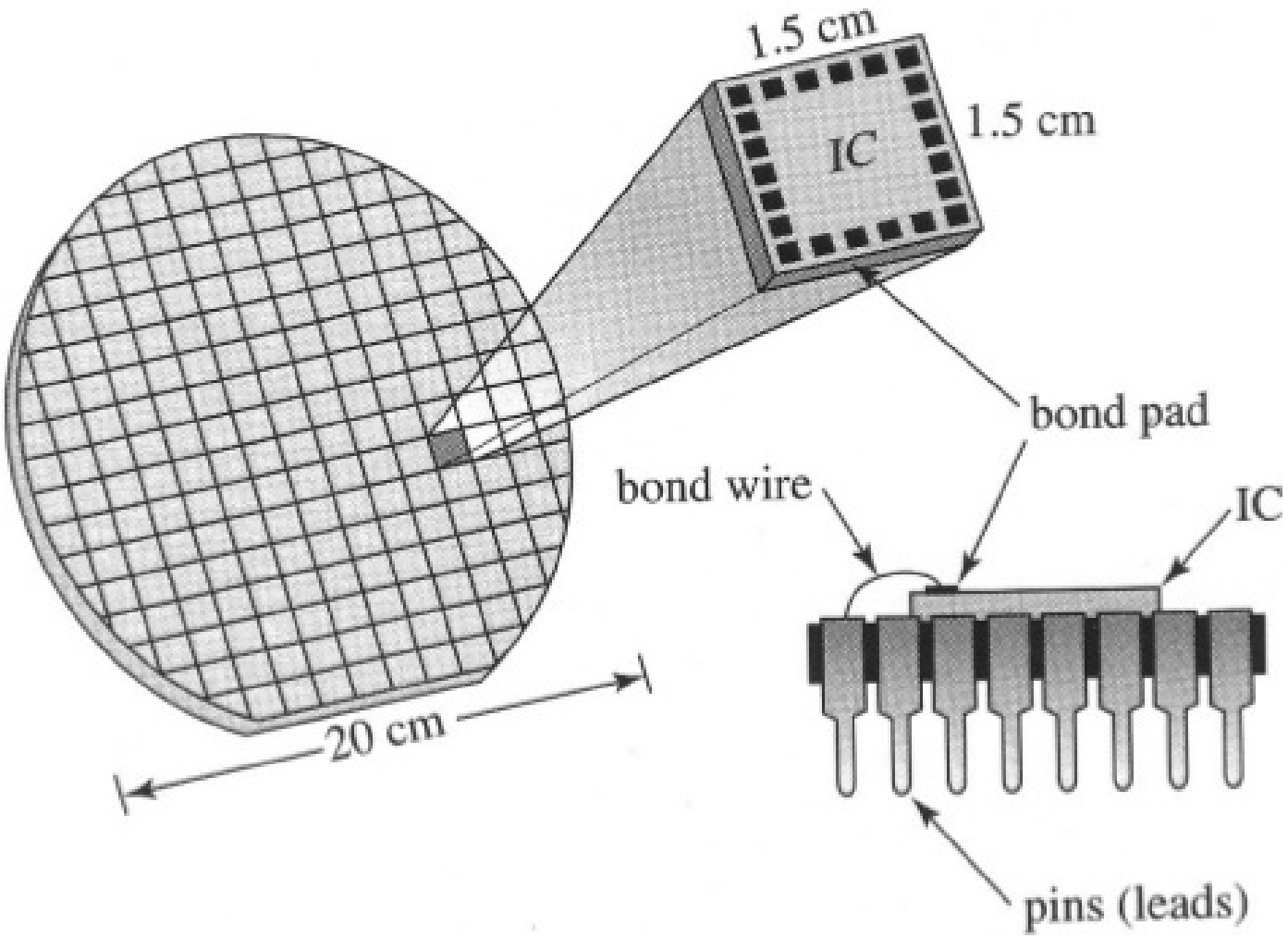
https://ece.engr.uvic.ca/~fayez/

Outline

Integrated Circuit (IC) Fabrication

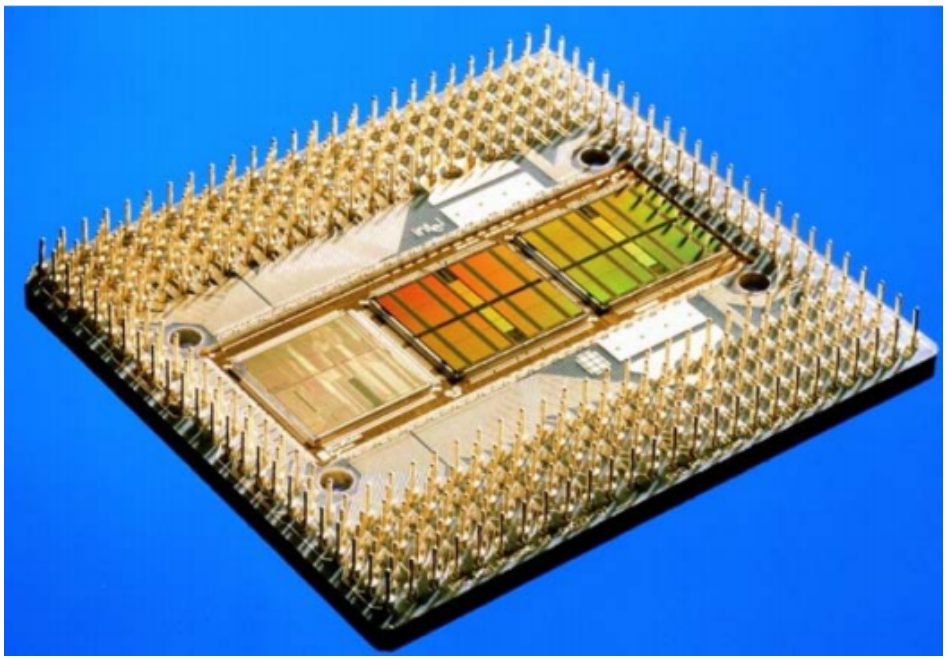
Printed Circuit Board (PCB)

An Integrated Die and Packaging

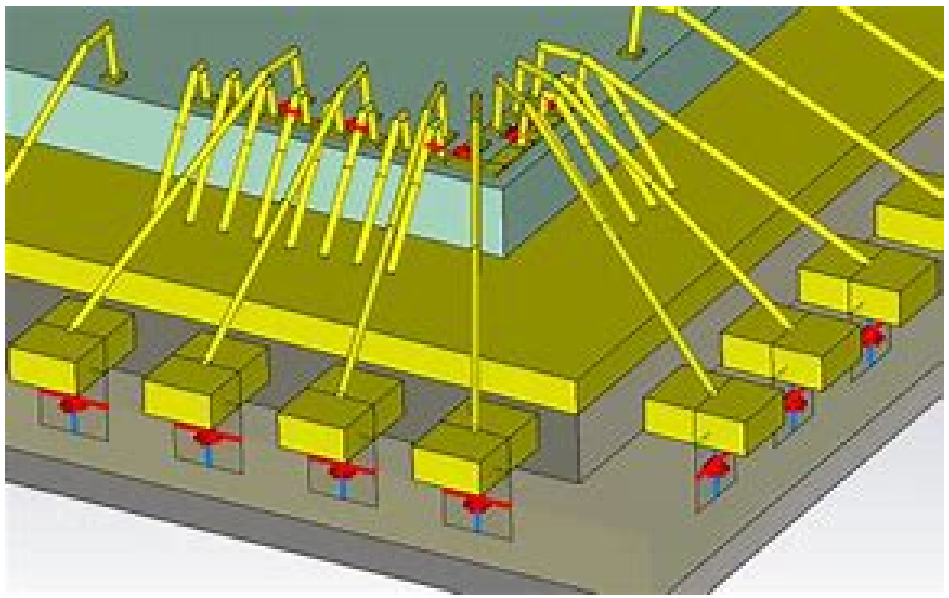


Encapsulated Integrated Circuit

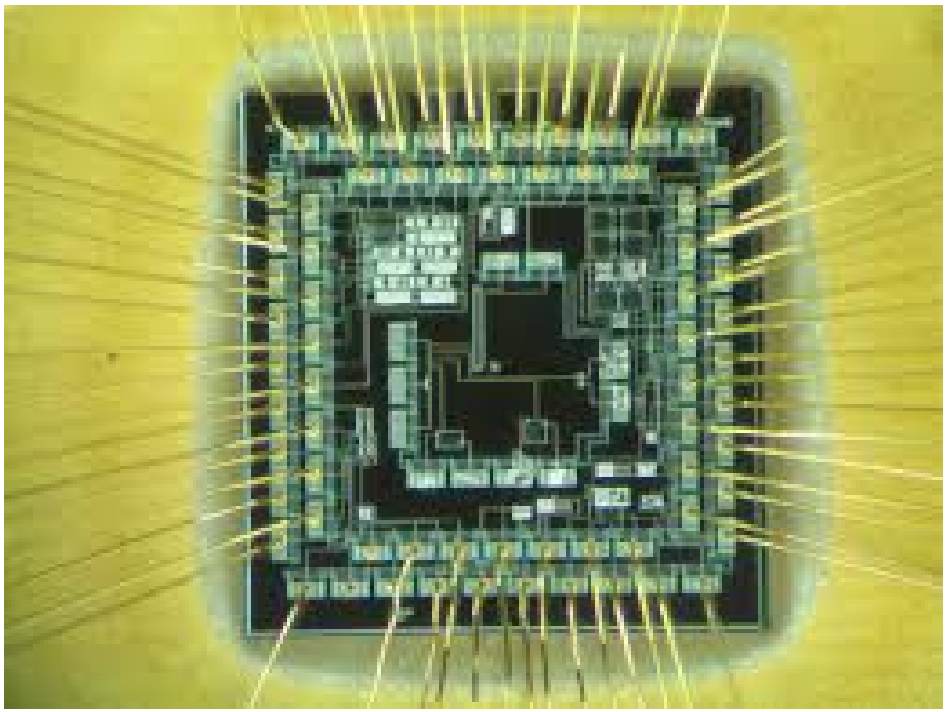
Integrated Circuit I/O: Pin Grid Array (PGA)



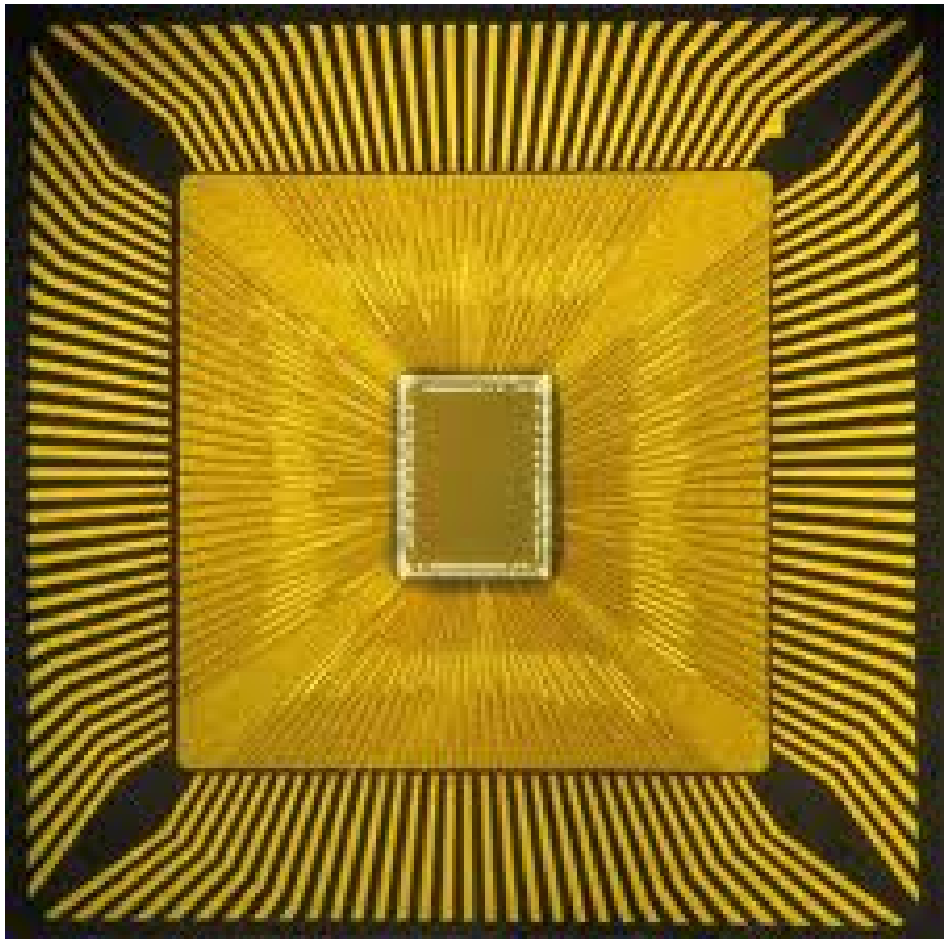
Silicon Core Connection to Padframe



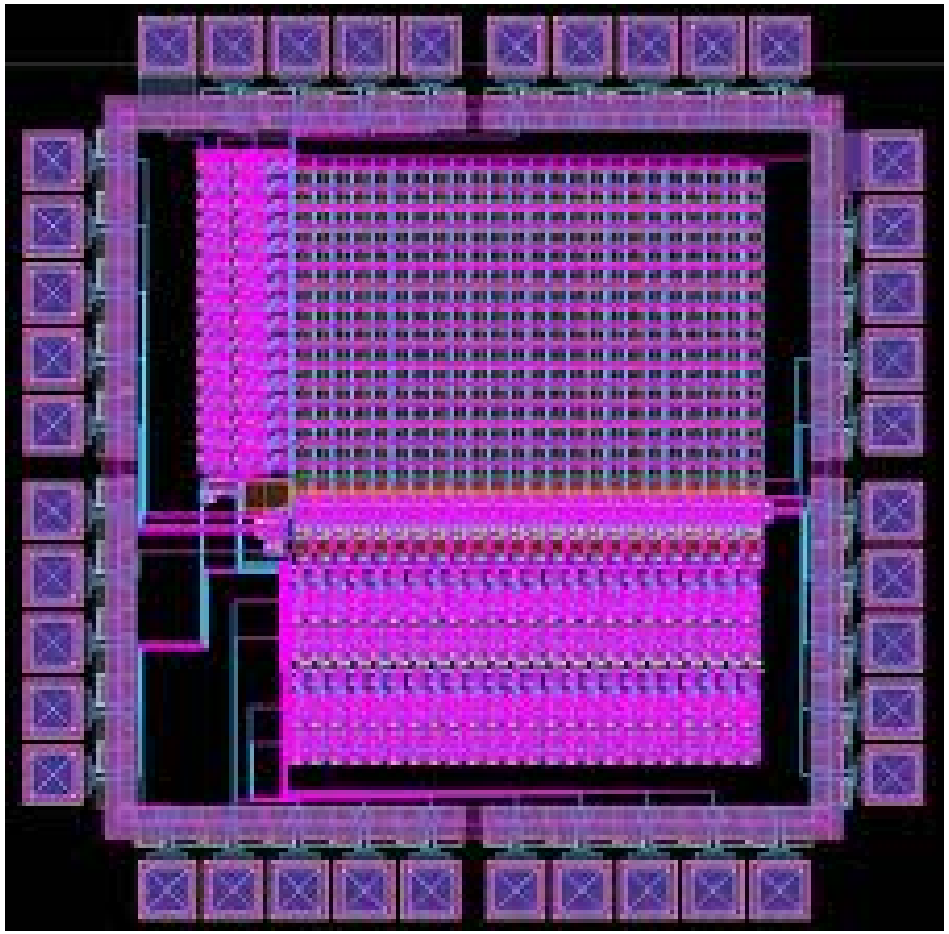
Integrated Circuit Wires Connecting Padframe to I/O Pins



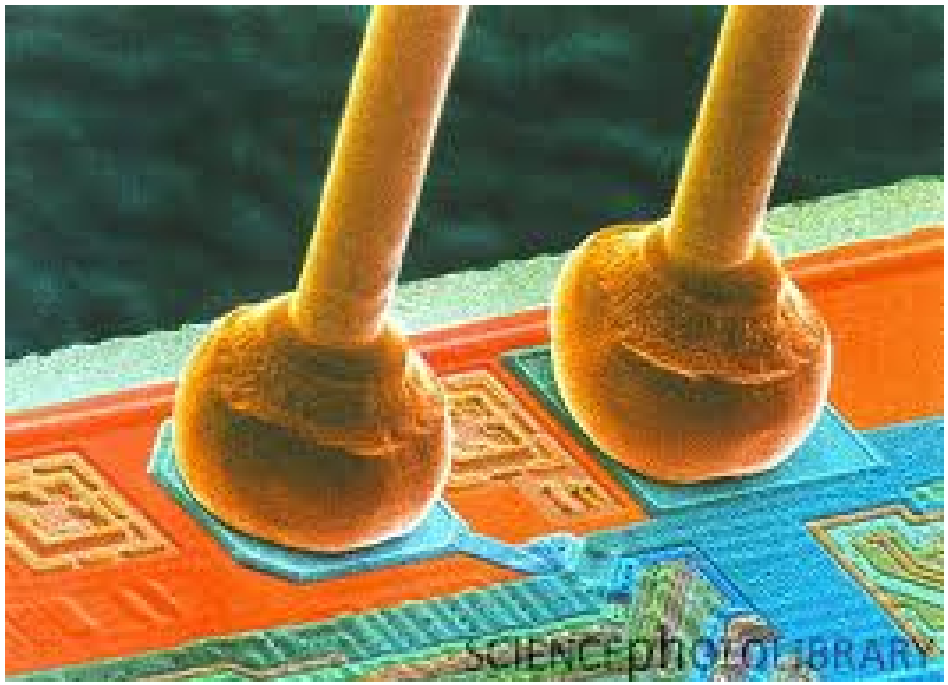
Integrated Circuit Wiring between Padframe & I/O



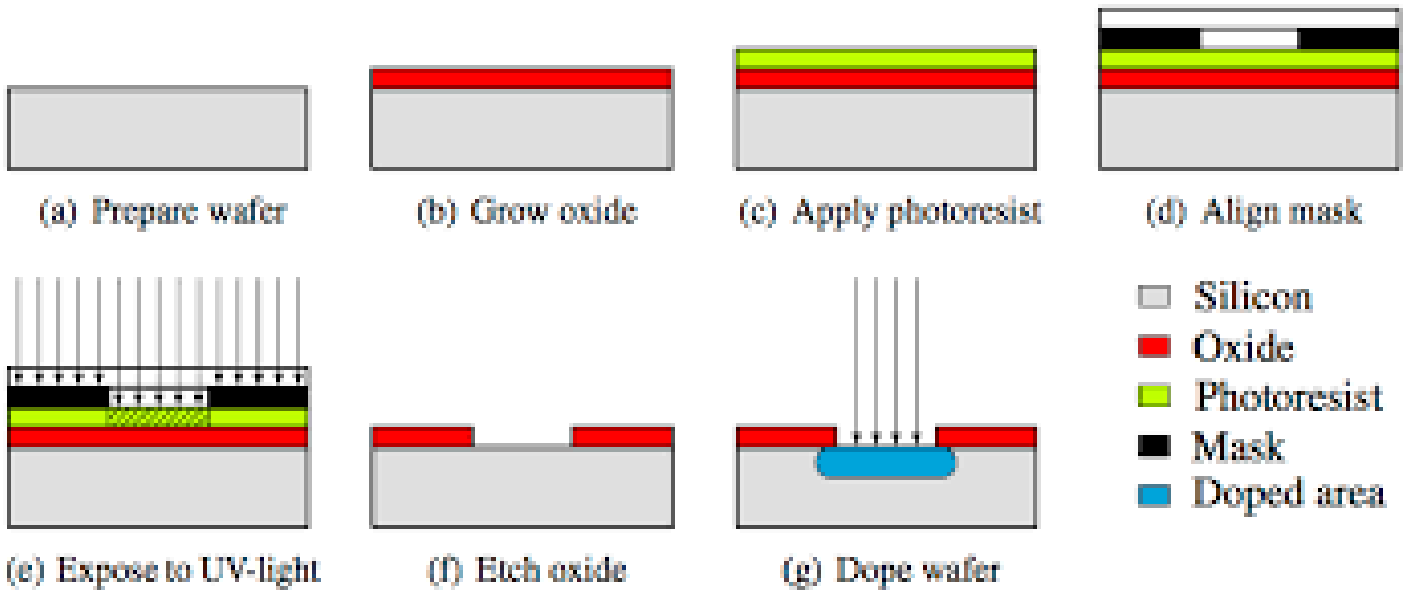
Silicon Core Connection to Padframe



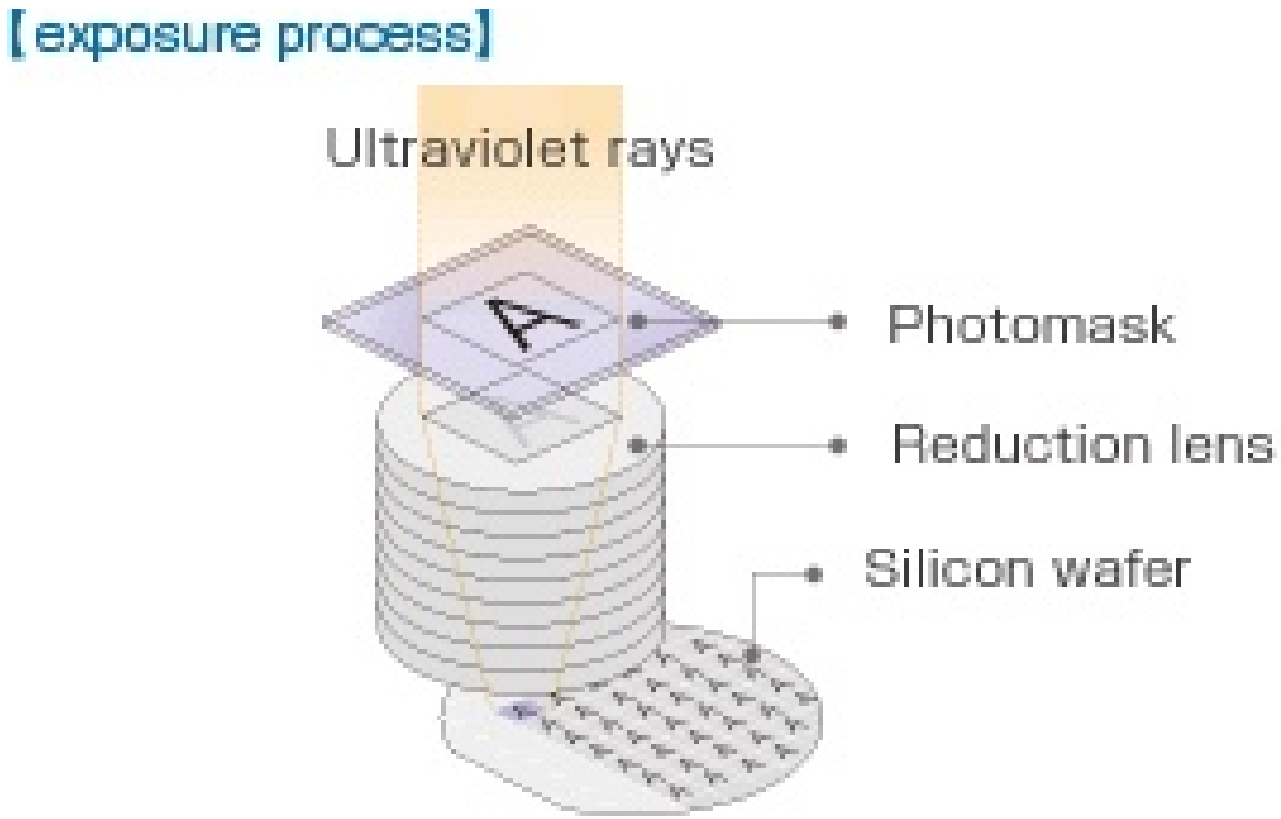
Close-Up View of Padframe to Wires



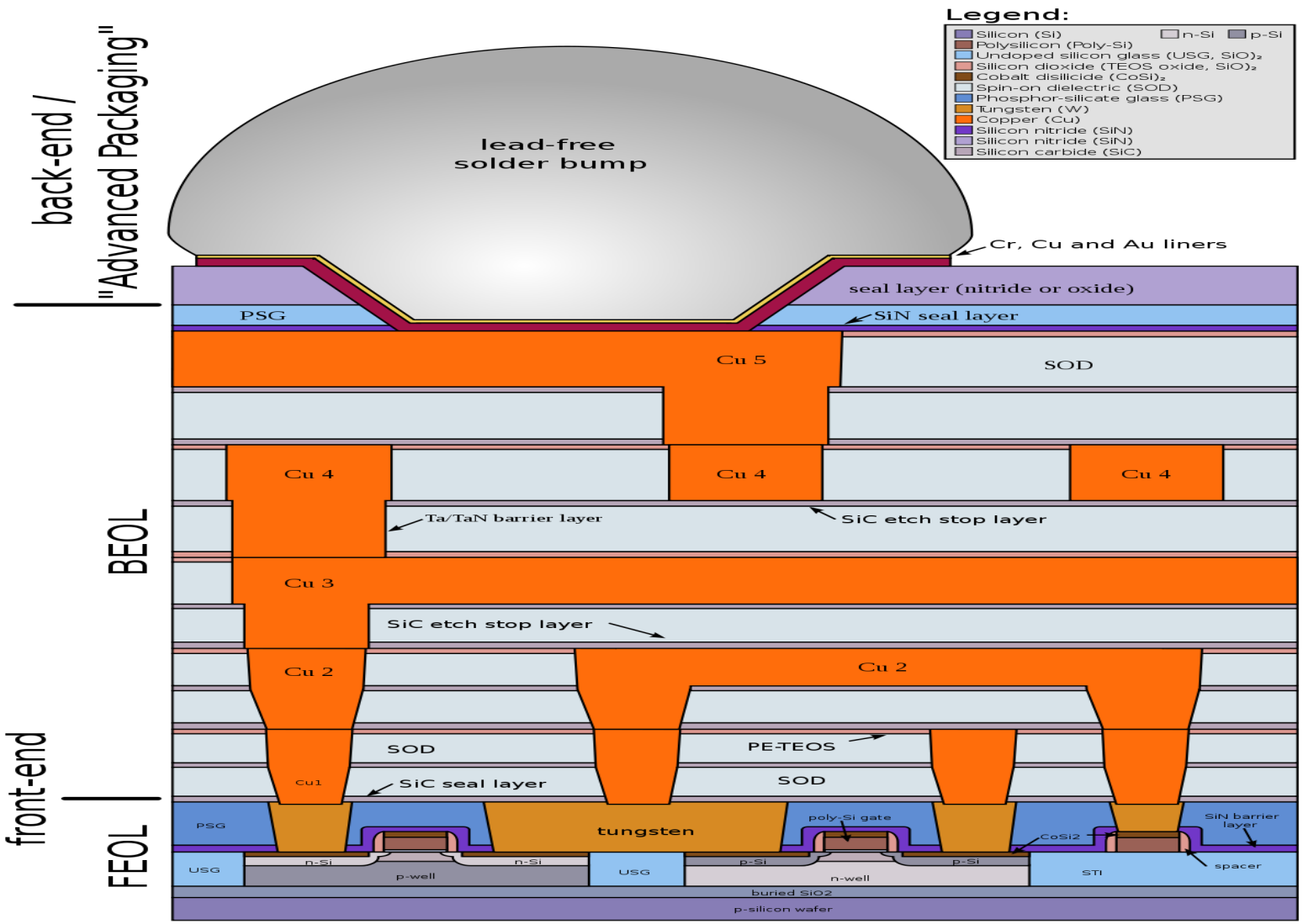
IC Fabrication: Masking



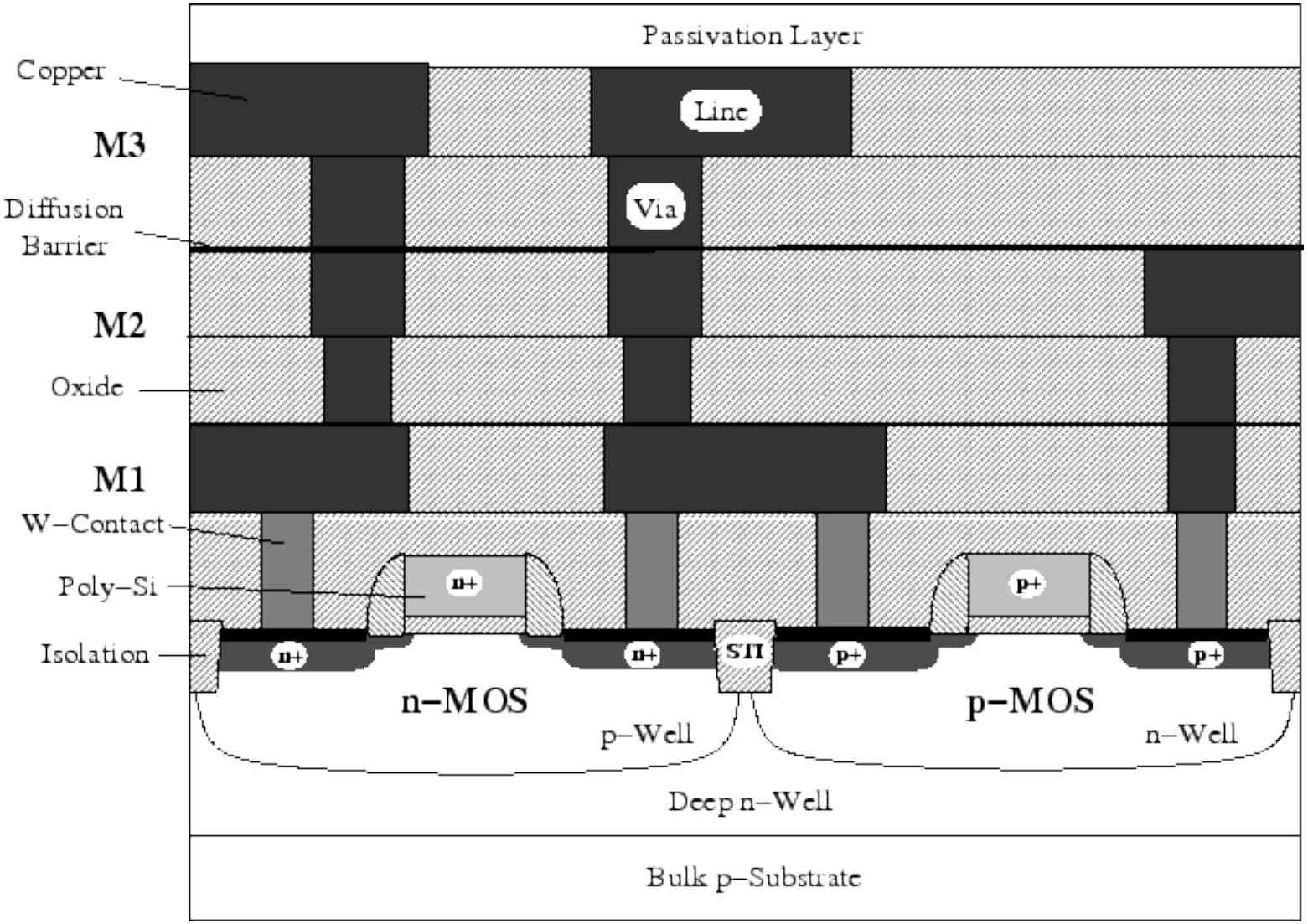
IC Fabrication: Masking



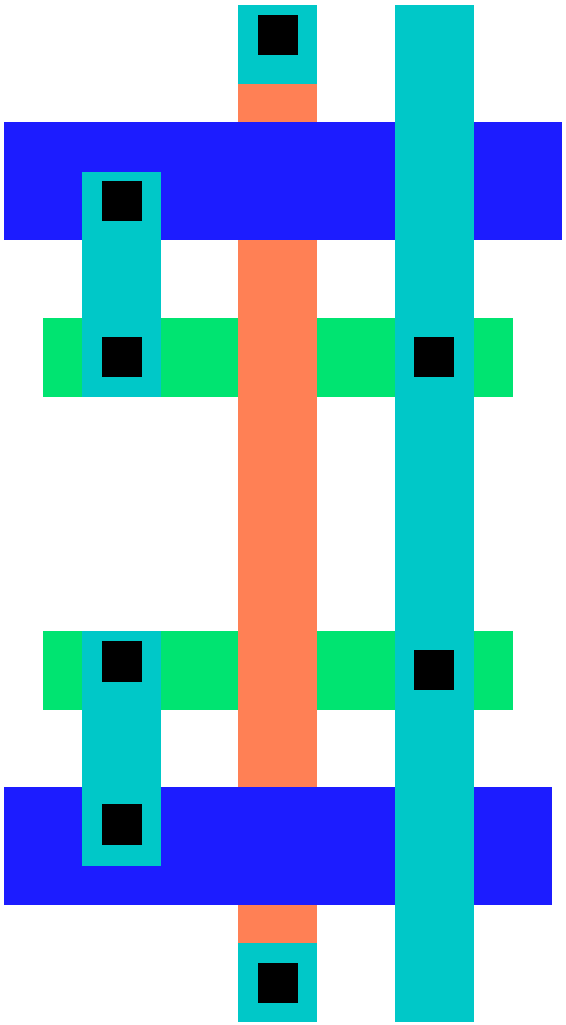
Advanced CMOS Layers

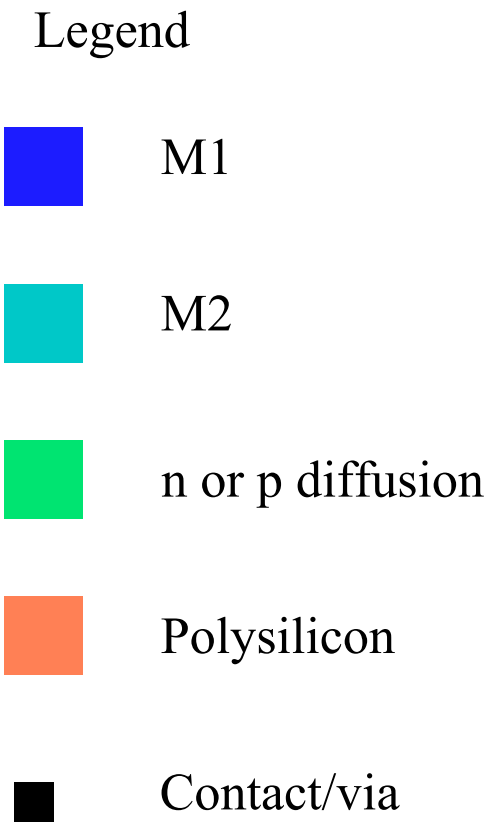


Advanced CMOS Layers

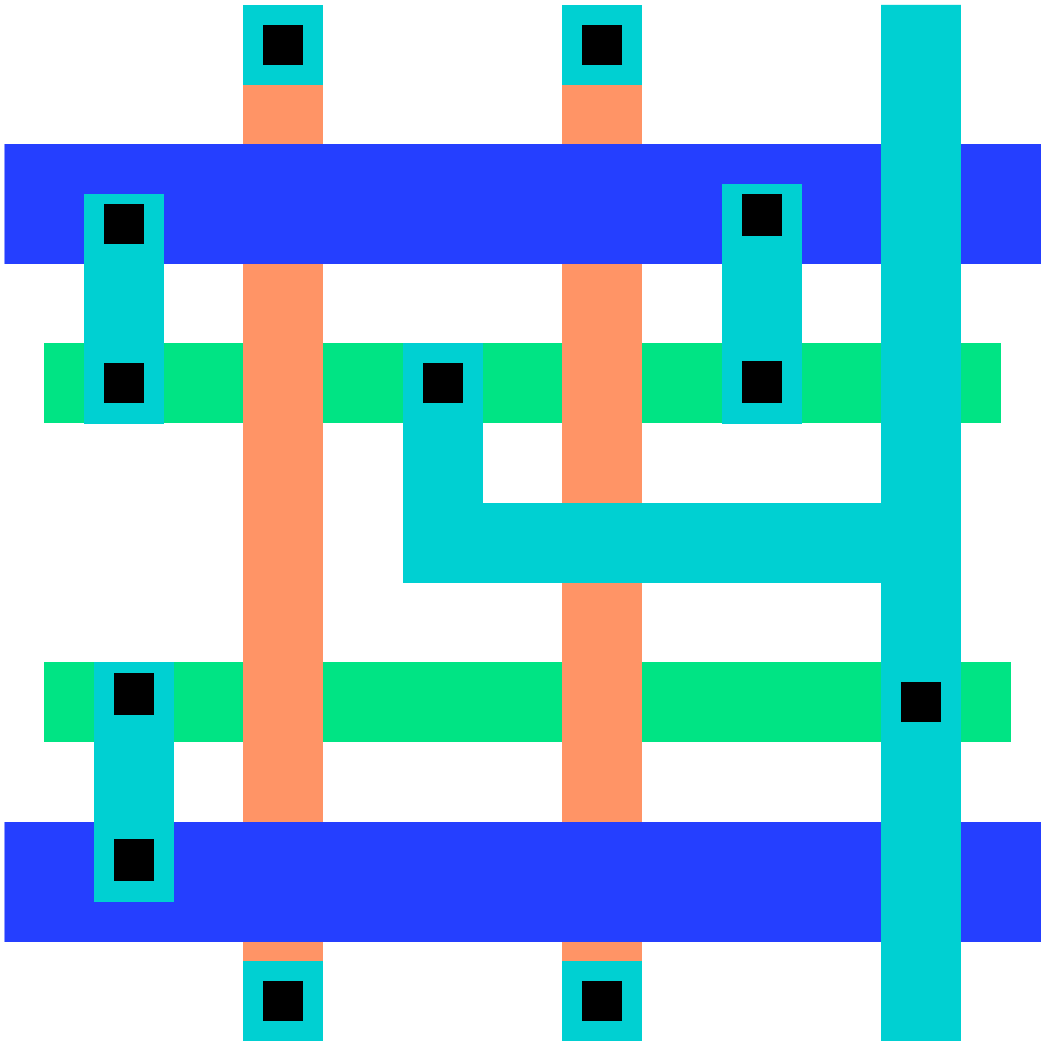


The Layers Defining an IC: Simple Inverter





The Layers Defining an IC: Simple NAND Gate



IC Layout Rules

1. Minimum width of a wire

IC Layout Rules

1. Minimum width of a wire
2. Minimum separation between wires in same layer

IC Layout Rules

1. Minimum width of a wire
2. Minimum separation between wires in same layer
3. Minimum separation between contacts or vias

IC Layout Rules

1. Minimum width of a wire
2. Minimum separation between wires in same layer
3. Minimum separation between contacts or vias
4. Minimum size of a via or contact

IC Layout Rules

1. Minimum width of a wire
2. Minimum separation between wires in same layer
3. Minimum separation between contacts or vias
4. Minimum size of a via or contact
5. Clock tree and PLL for clock synchronization

IC Layout Rules

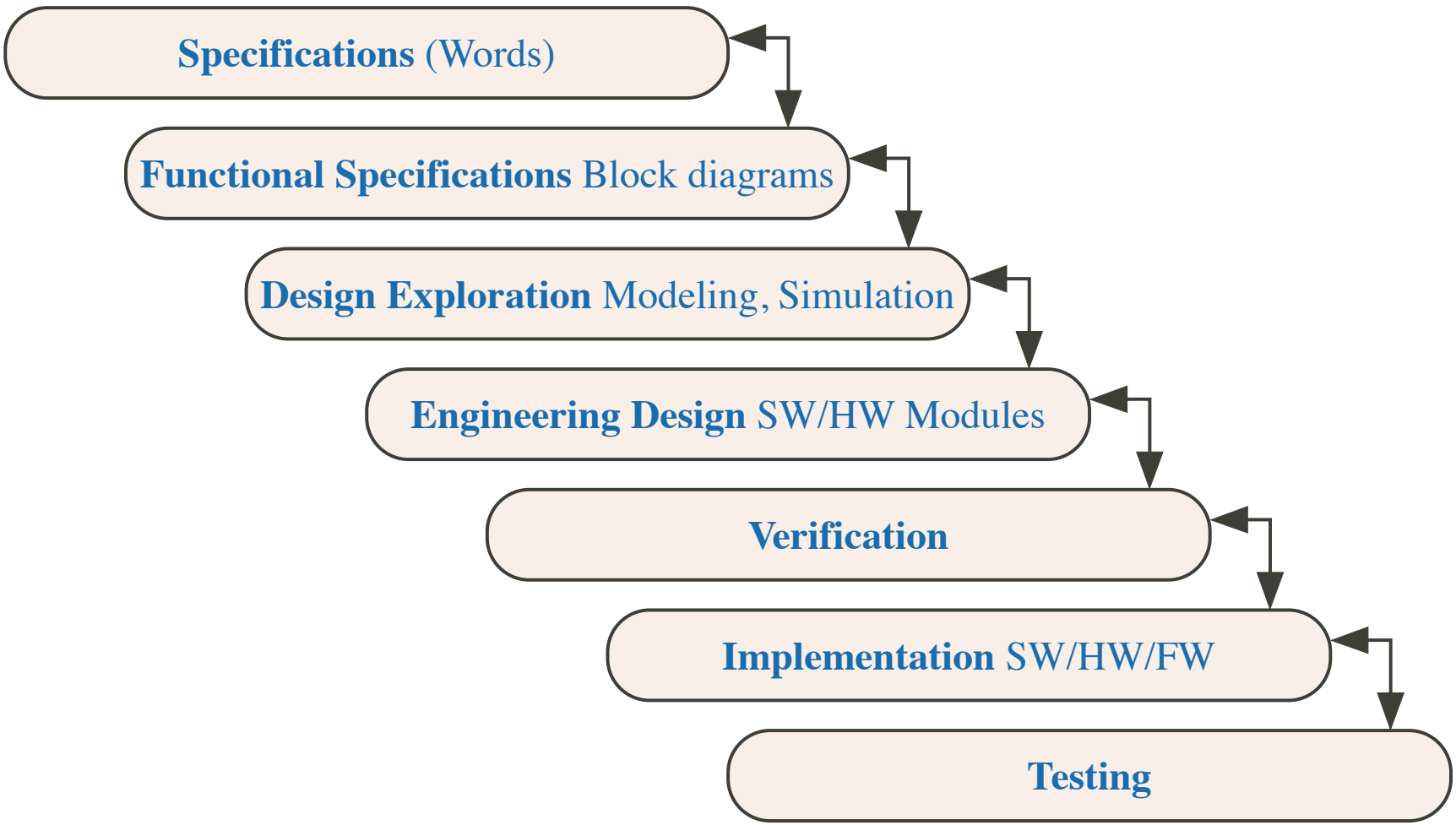
1. Minimum width of a wire
2. Minimum separation between wires in same layer
3. Minimum separation between contacts or vias
4. Minimum size of a via or contact
5. Clock tree and PLL for clock synchronization
6. Power supply wire width (reduce ground bounce)

IC Layout Rules

1. Minimum width of a wire
2. Minimum separation between wires in same layer
3. Minimum separation between contacts or vias
4. Minimum size of a via or contact
5. Clock tree and PLL for clock synchronization
6. Power supply wire width (reduce ground bounce)
7. Minimum gates rise/fall times (reduce overlap power)

IC Design Flow

Integrated Circuit Digital Design Process: Waterfall Model

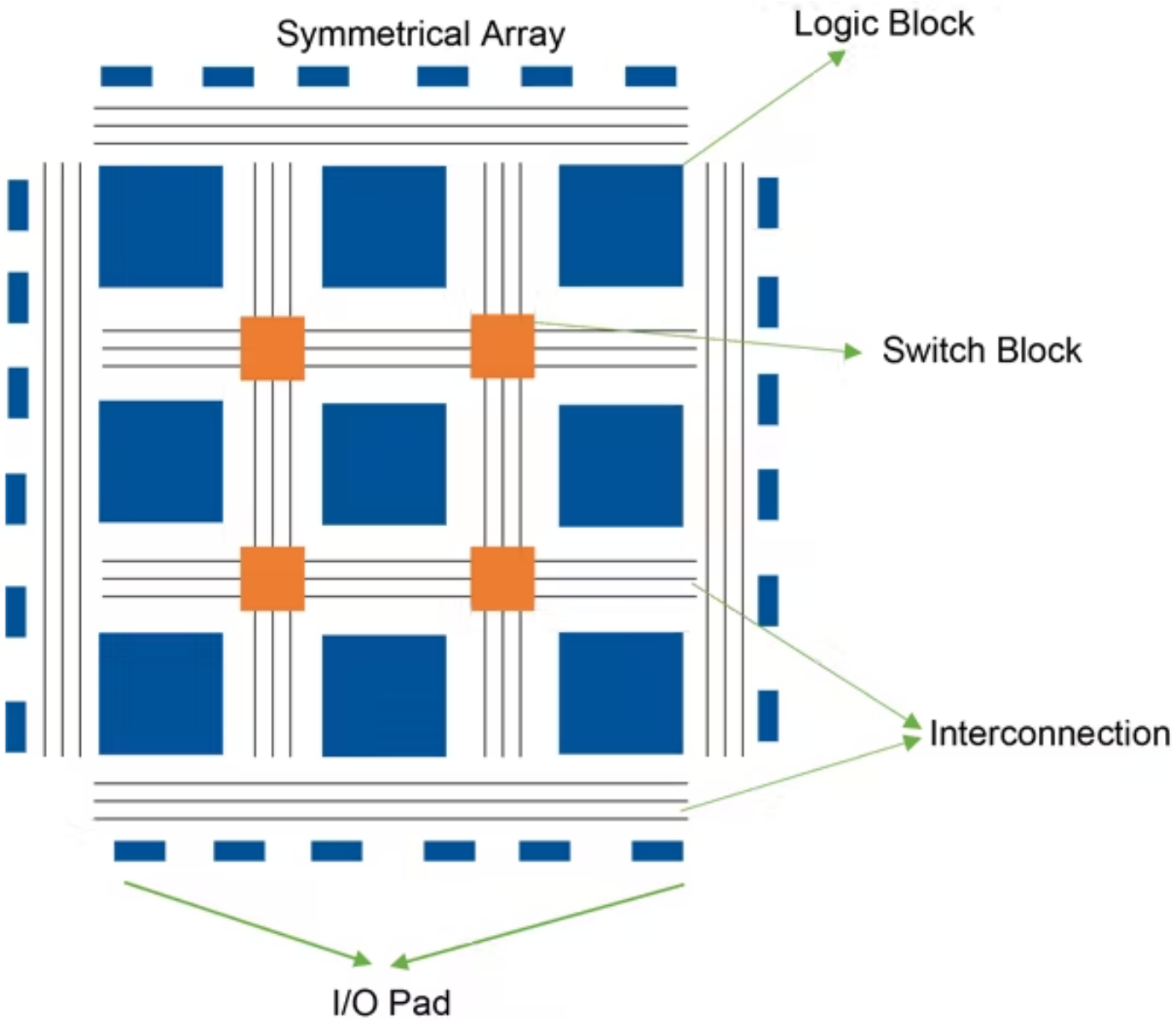


FPGA Design

Why use FPGAs?

1. Short product release time
2. Performance is starting to match ASICs
3. Allows for inexpensive design changes
4. Recovery from design errors is simple re-configuration
5. Easily update design or add new functionality
6. Can integrate IP functional modules (μP, DSP, etc.)

FPGA Basic Architecture



FPGA Types

1. SRAM (volatile, reprogrammable)
2. Antifuse (non-volatile, one-time programmable)
3. EPROM (non-volatile, reprogrammable)

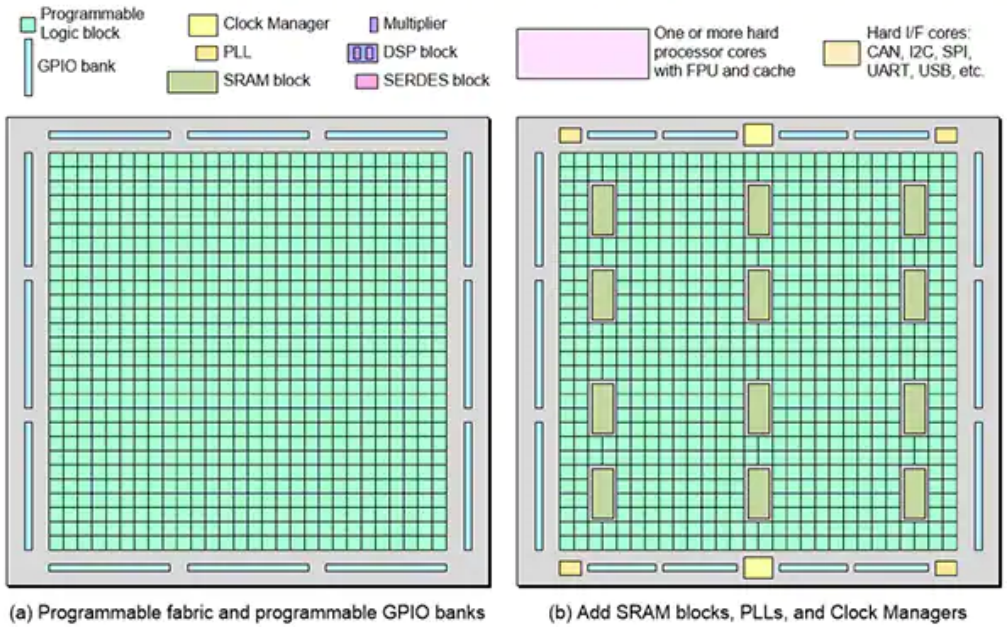
SRAM-Based FPGA

1. Advantages:
   1. Easily reprogrammable
   2. Requires standard IC processing
2. Disadvantages
   1. Volatile
   2. Large area
   3. External configuration from memory or file

FPGA Main Components: The Fabric

1. Built-in CLB blocks
2. Built-in Block RAM
3. DSP block IP
4. Processor IP

FPGA Capabilities



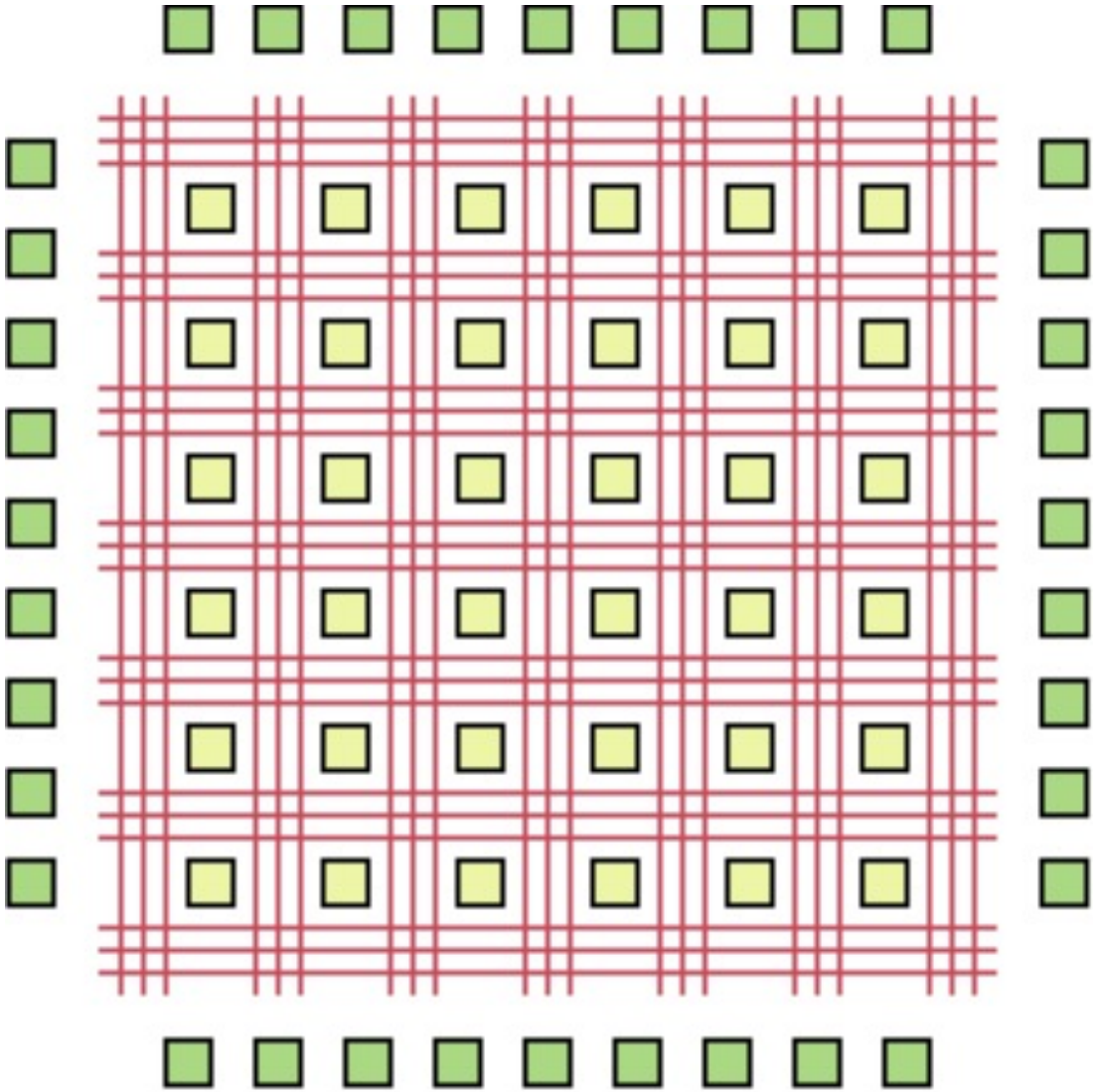
(a) Programmable fabric and programmable GPIO banks

(b) Add SRAM blocks, PLLs, and Clock Managers

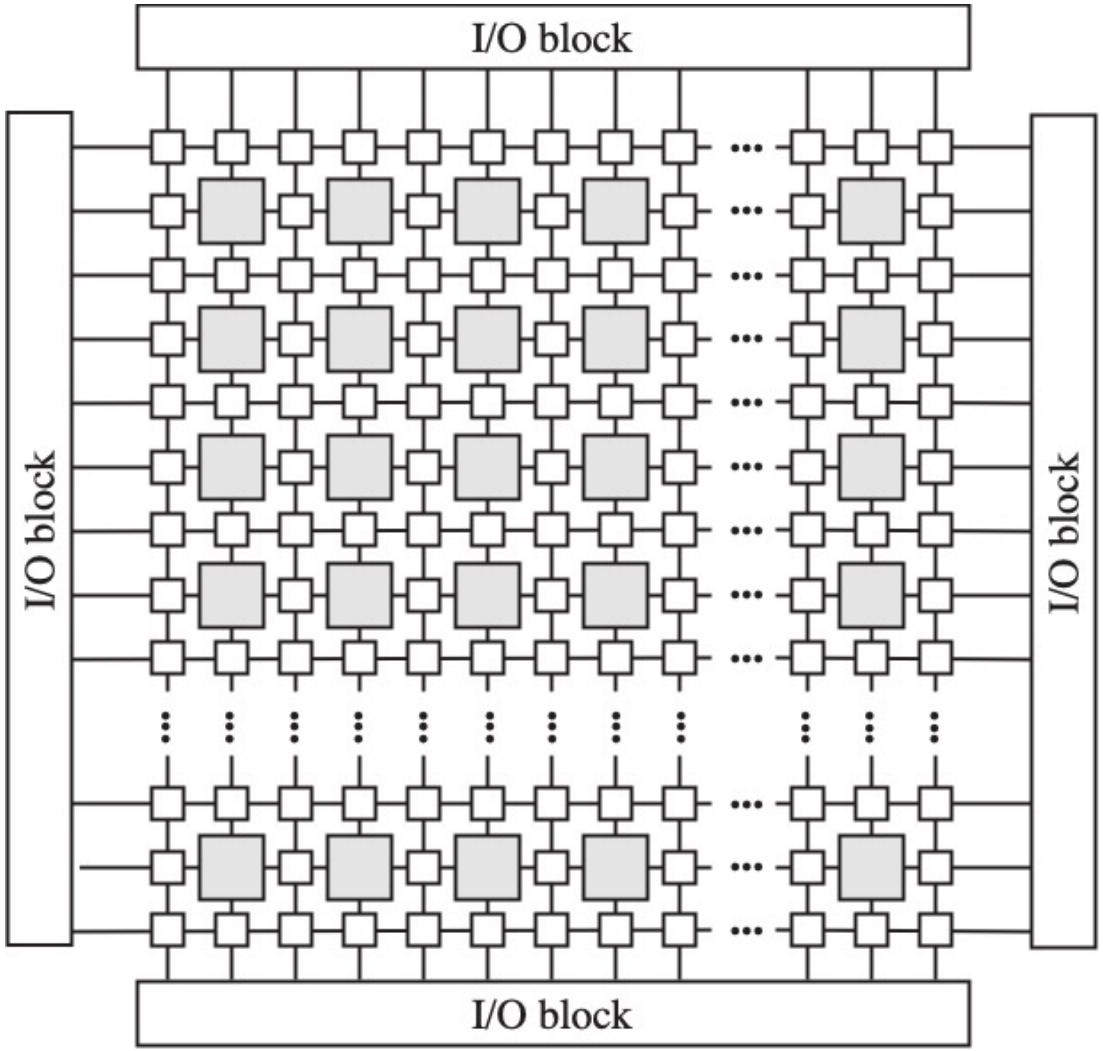
(c) Add DSP slices and SERDES blocks

(d) Add hard processor core(s) and peripherals

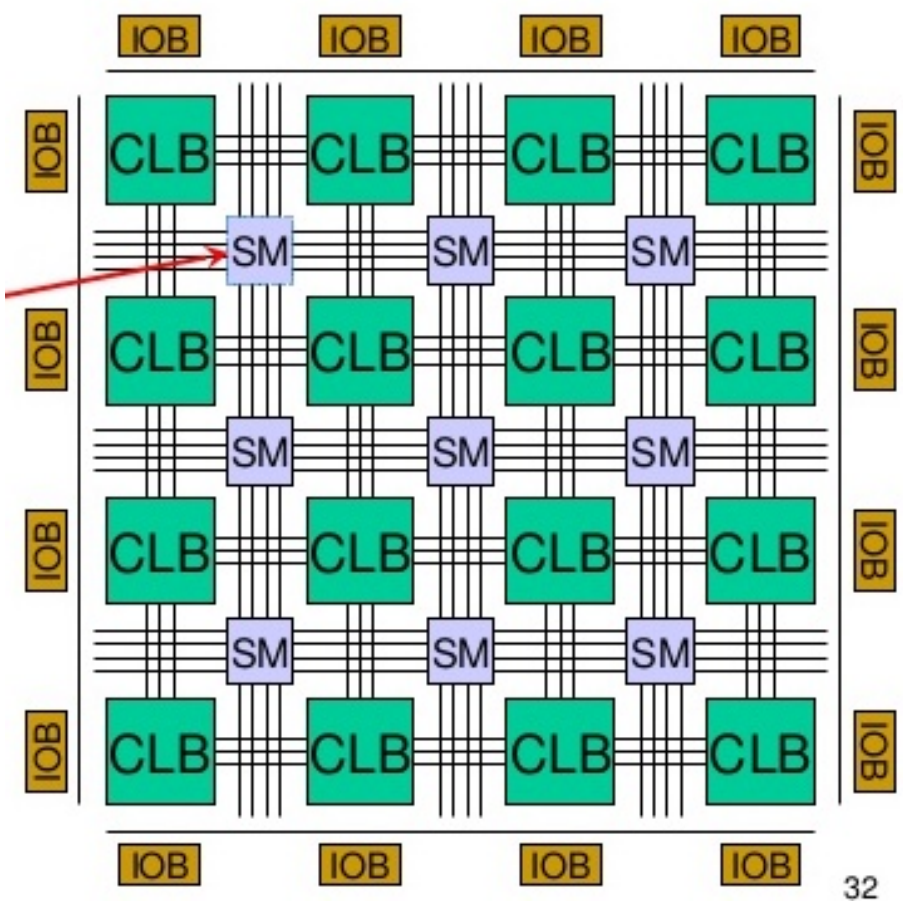
FPGA Architecture: IOB, CLB, Switch Matrix



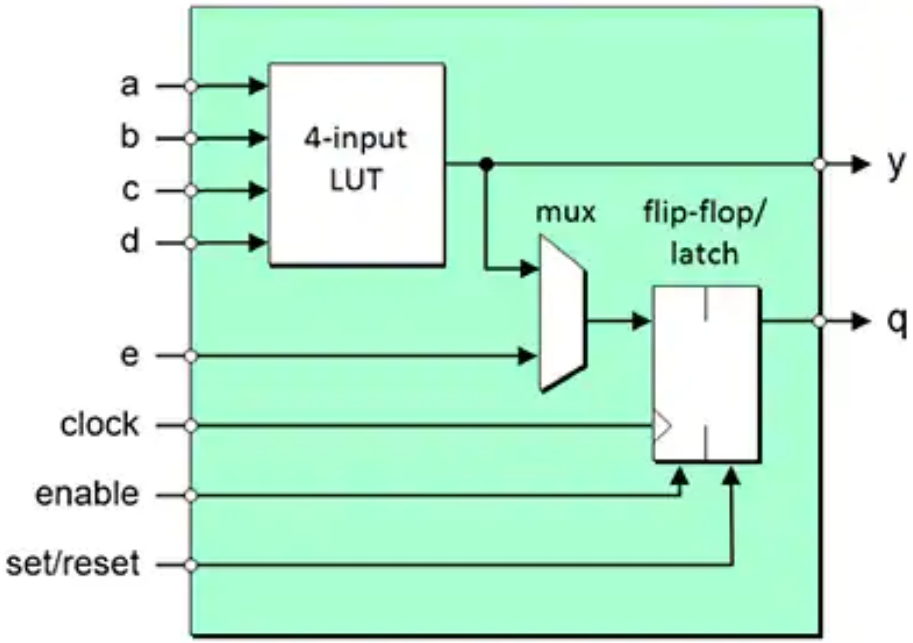
FPGA Architecture: IOB, CLB, Switch Matrix



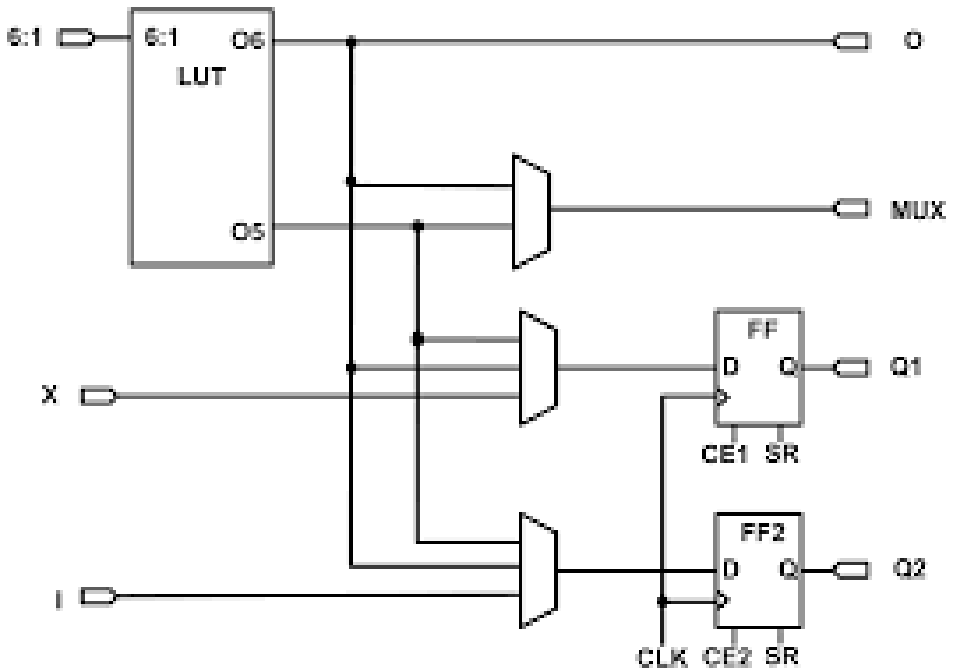
FPGA Architecture: IOB, CLB, Switch Matrix



FPGA Architecture: CLB

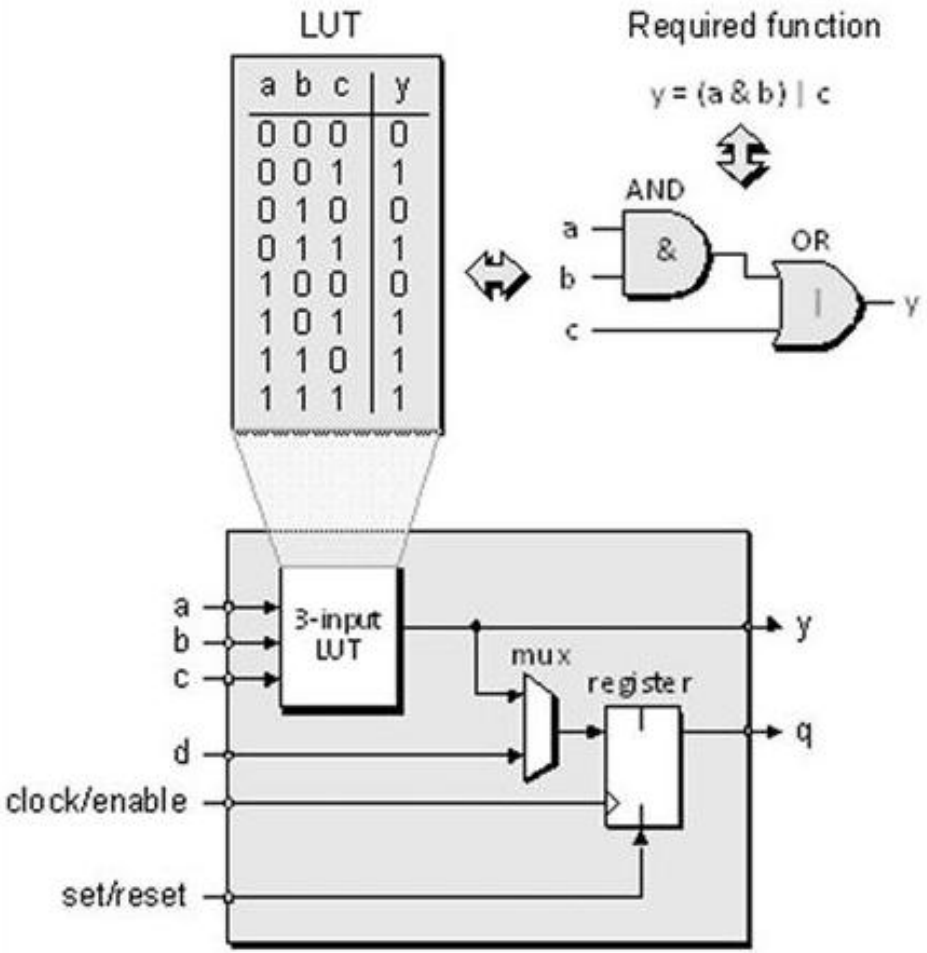


FPGA Configurable Logic Block (Simple CLB)

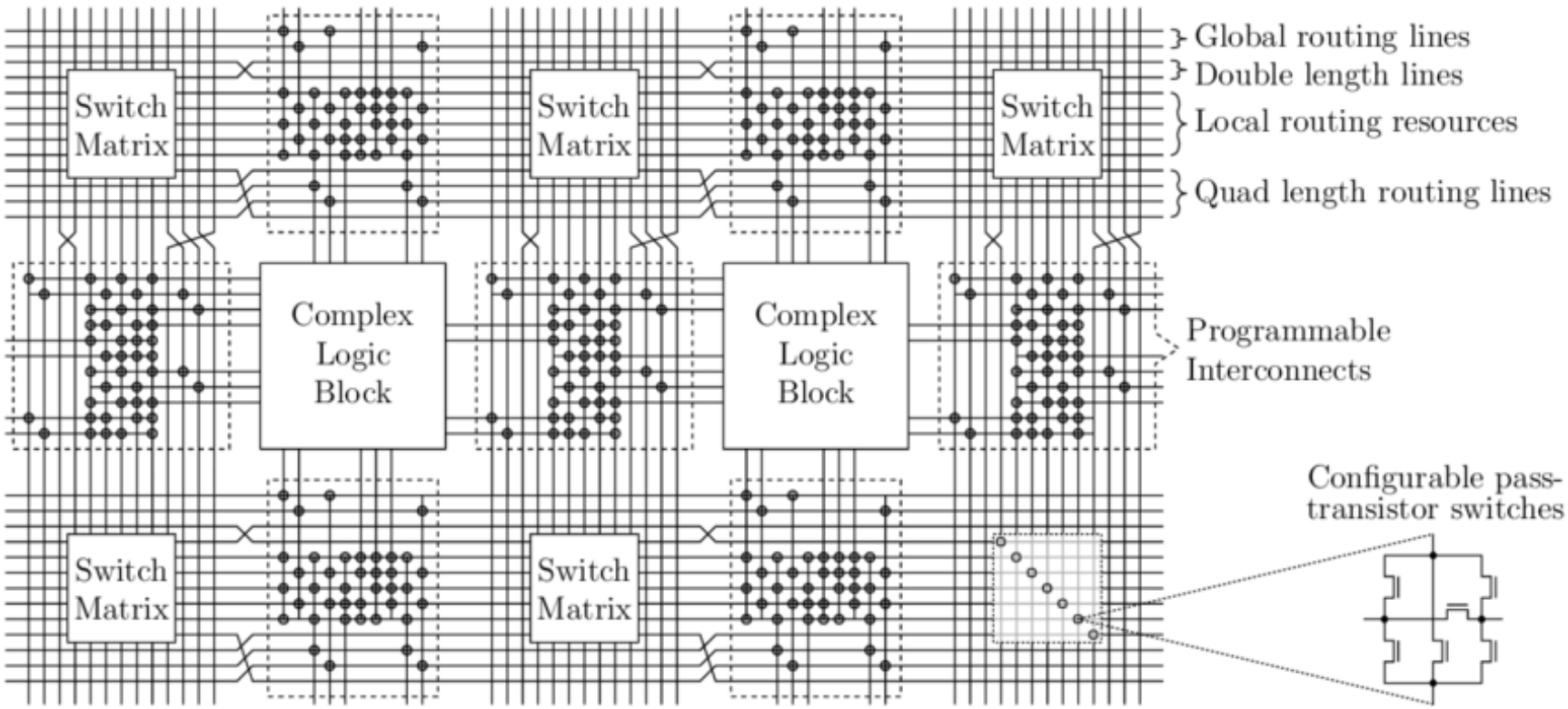


1. LUT stores logic function to be implemented
2. LUT output could be direct or registered

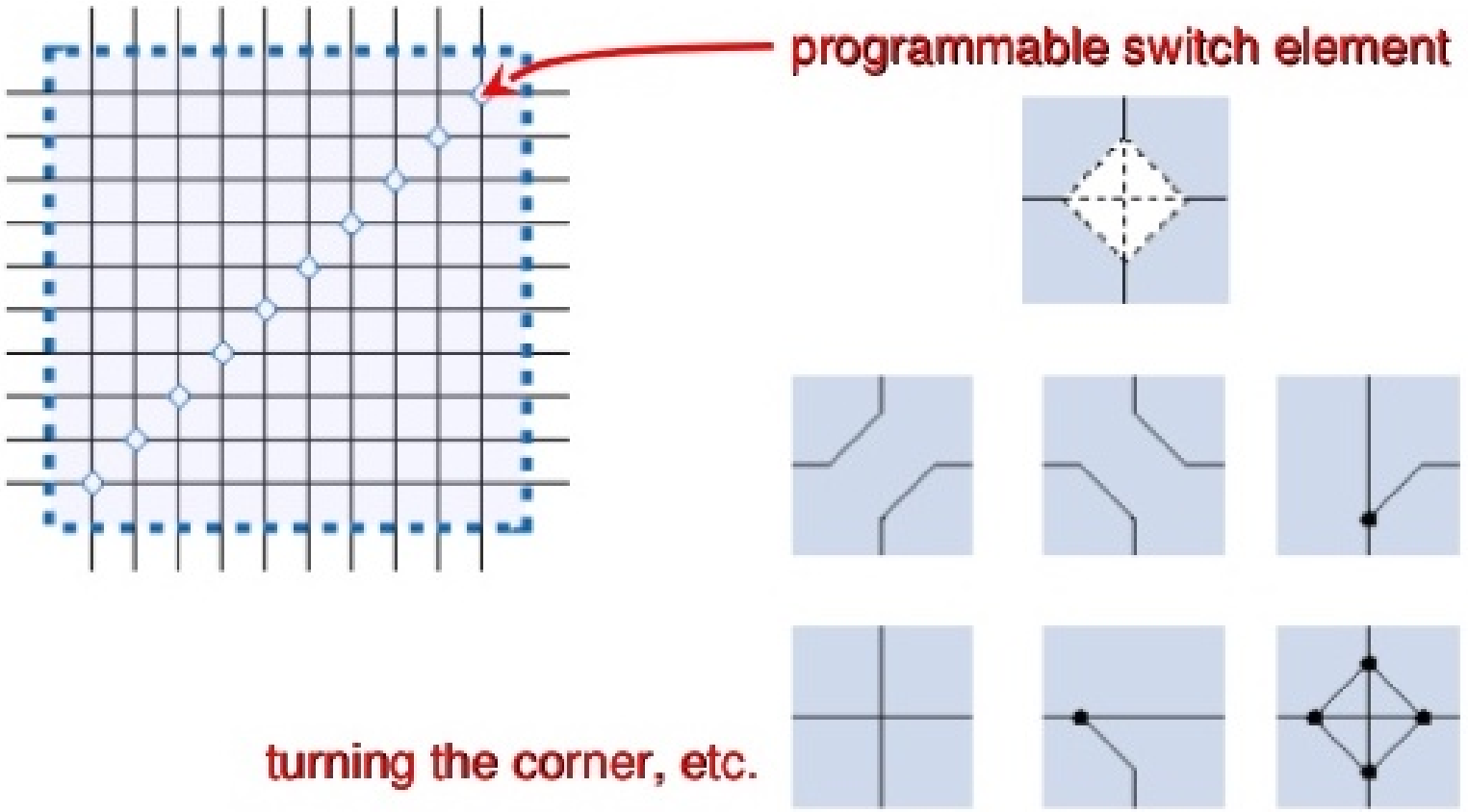
LUT


LUT
a b c y
0 0 0 0
0 0 1 1
0 1 0 0
0 1 1 1
1 0 0 0
1 0 1 1
1 1 0 1
1 1 1 1
Required function
y = (a & b) | c
AND
OR
a
b
c
&
|
y
3-input LUT
mux
register
a
b
c
d
clock/enable
set/reset
y
q

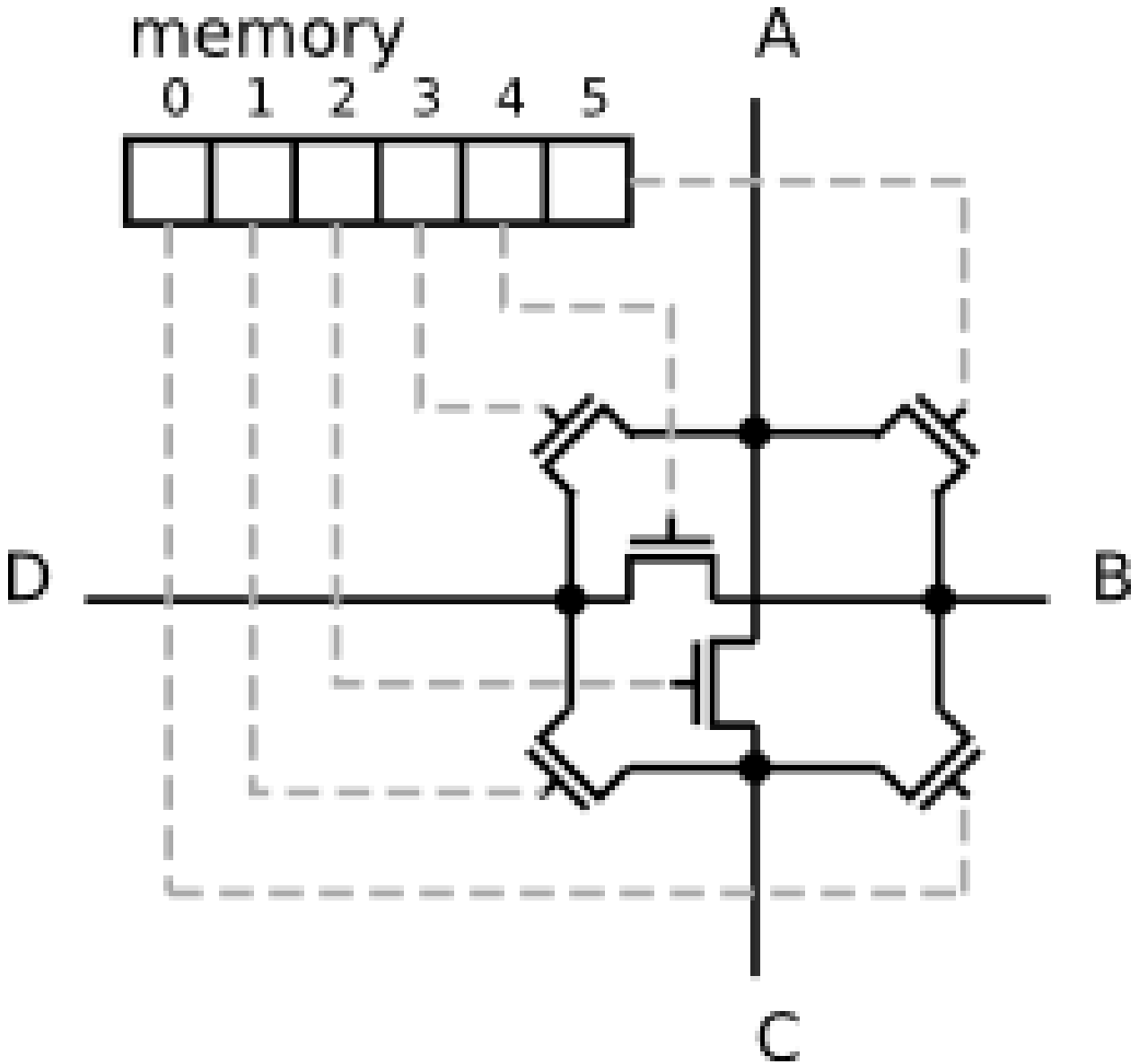
FPGA Programmable Switch Matrix



FPGA Switch Matrix Programming



FPGA Switch Matrix Programming



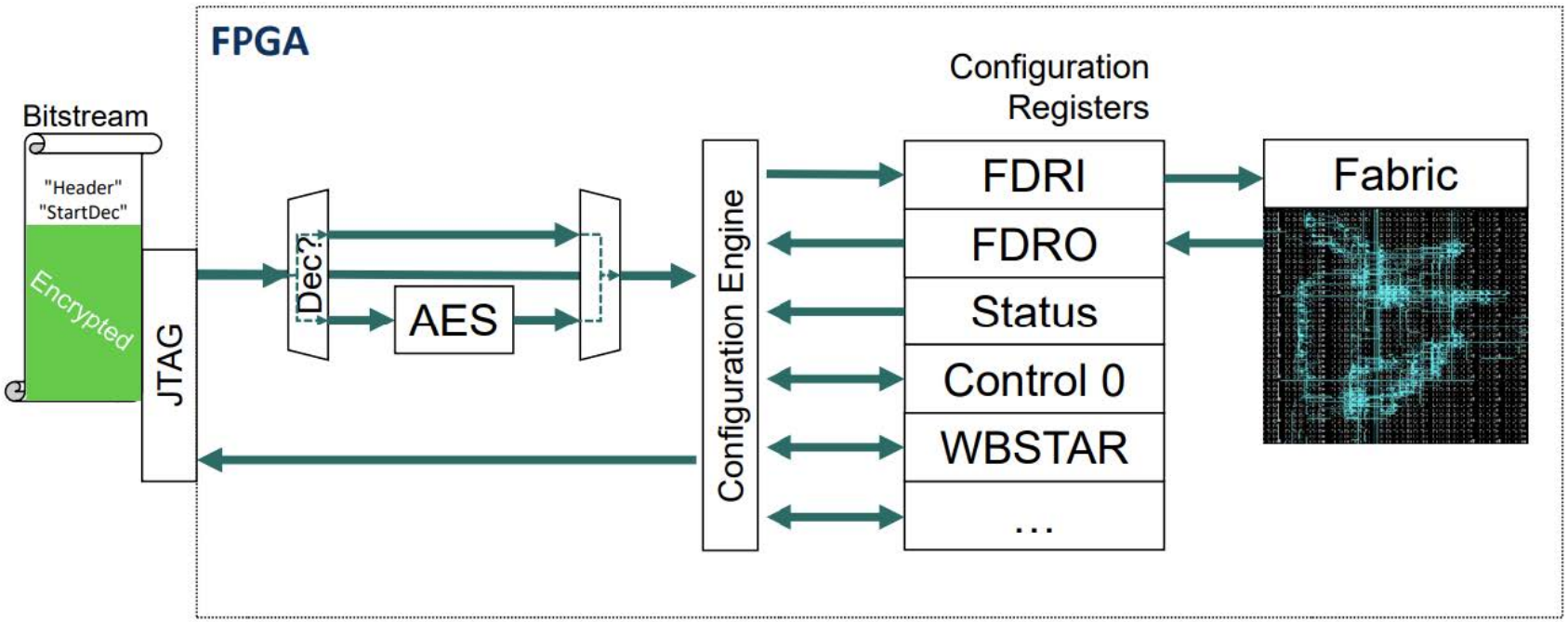
FPGA Design Flow

1. Import 3PIP
2. Design integration at design house
3. Generate HDL code
4. Compile code into FPGA bitstream file
5. Encode bitstream
6. Load coded bitstream in device NVRAM

FPGA Configuration

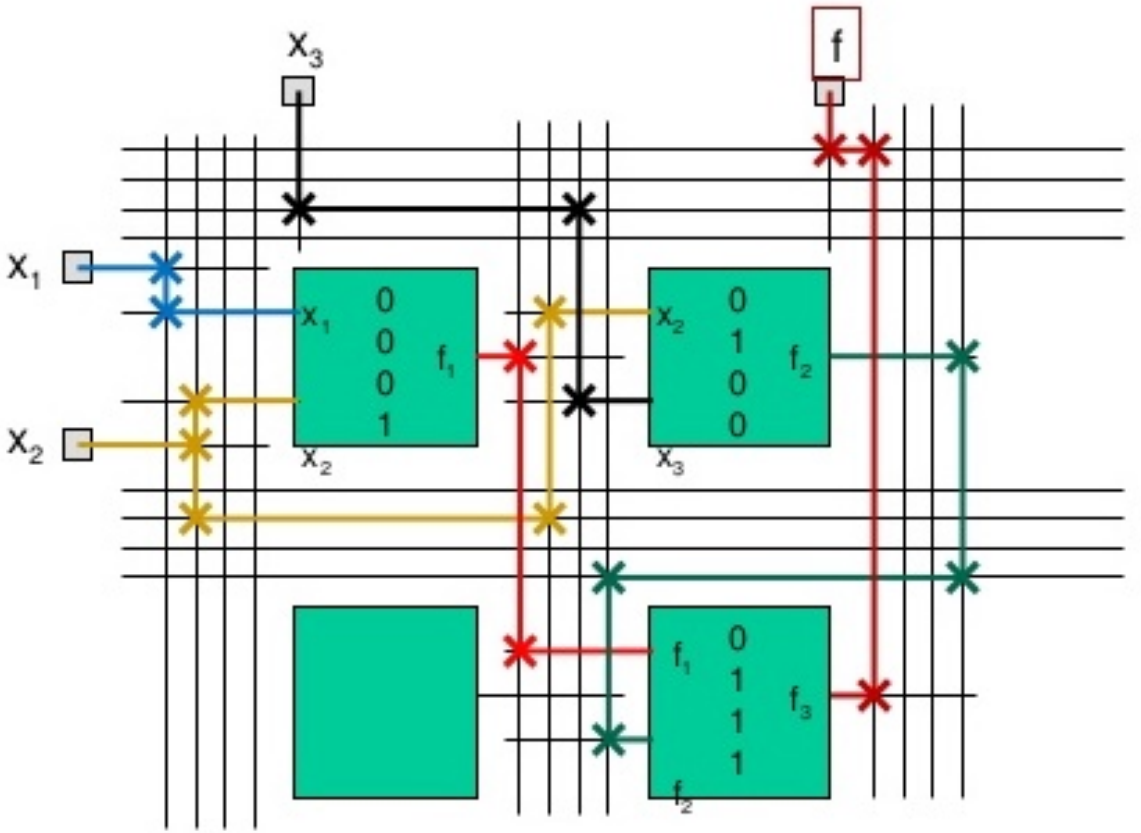
1. All configuration bits are present in bitstream
2. Bitstream is stored on external nonvolatile memory (NVRAM)
3. Configure on power up
4. Configure on demand
5. Configuration data is stored in PROM or external data source

FPGA Decrypt & Configuration



Note access via JTAG port

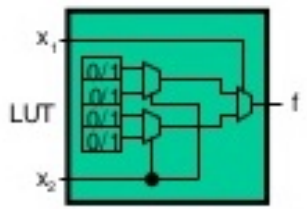
FPGA Configuration Example



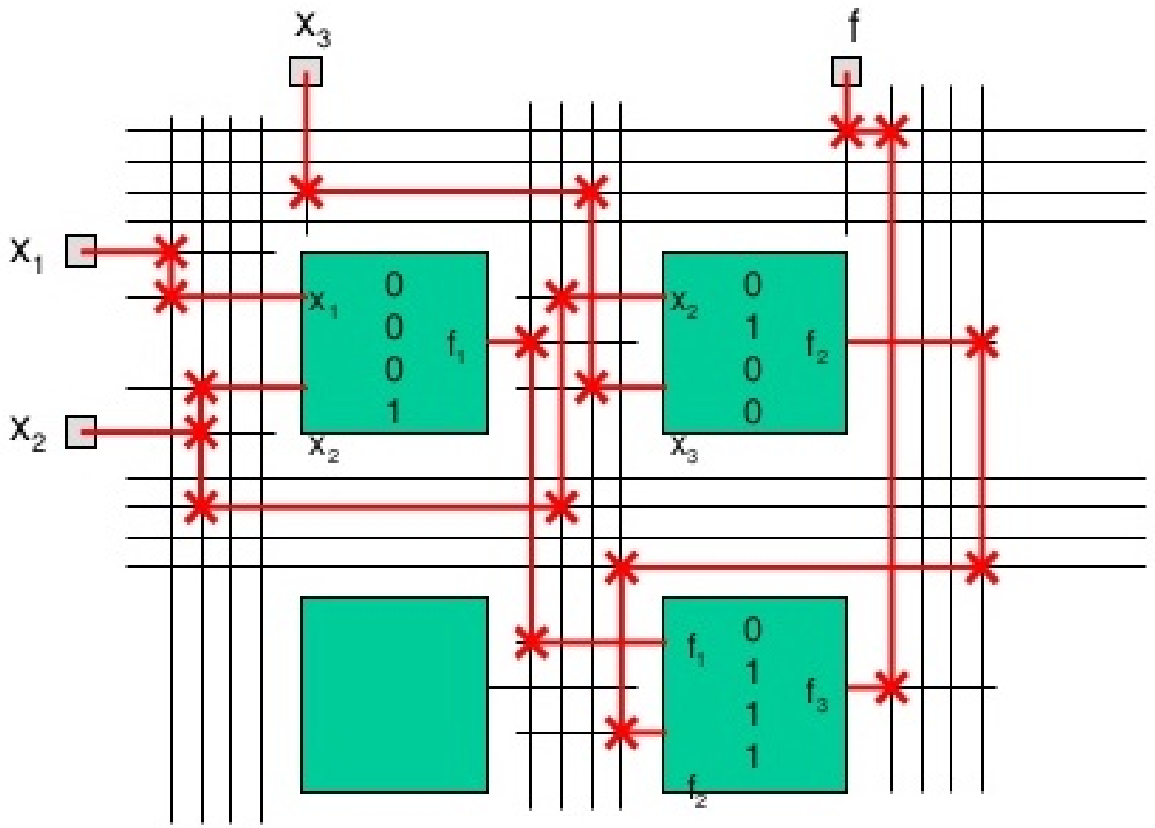
$$f_1 = x_1 x_2$$

$$f_2 = \overline{x_2} x_3$$

$$f = x_1 x_2 + \overline{x_2} x_3$$



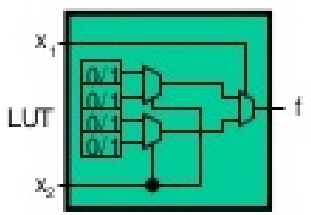
FPGA Configuration Example



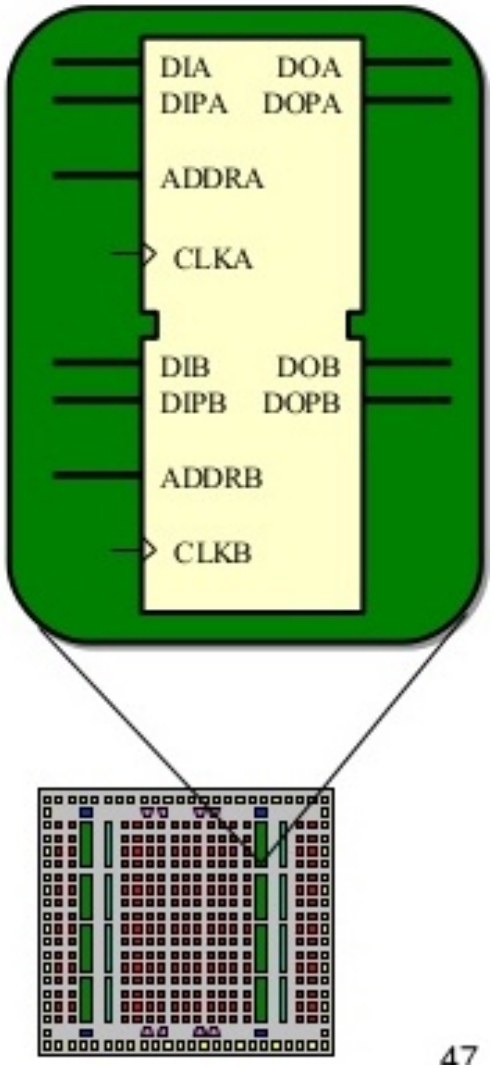
$$f_1 = x_1 x_2$$

$$f_2 = \overline{x_2} x_3$$

$$f = x_1 x_2 + \overline{x_2} x_3$$



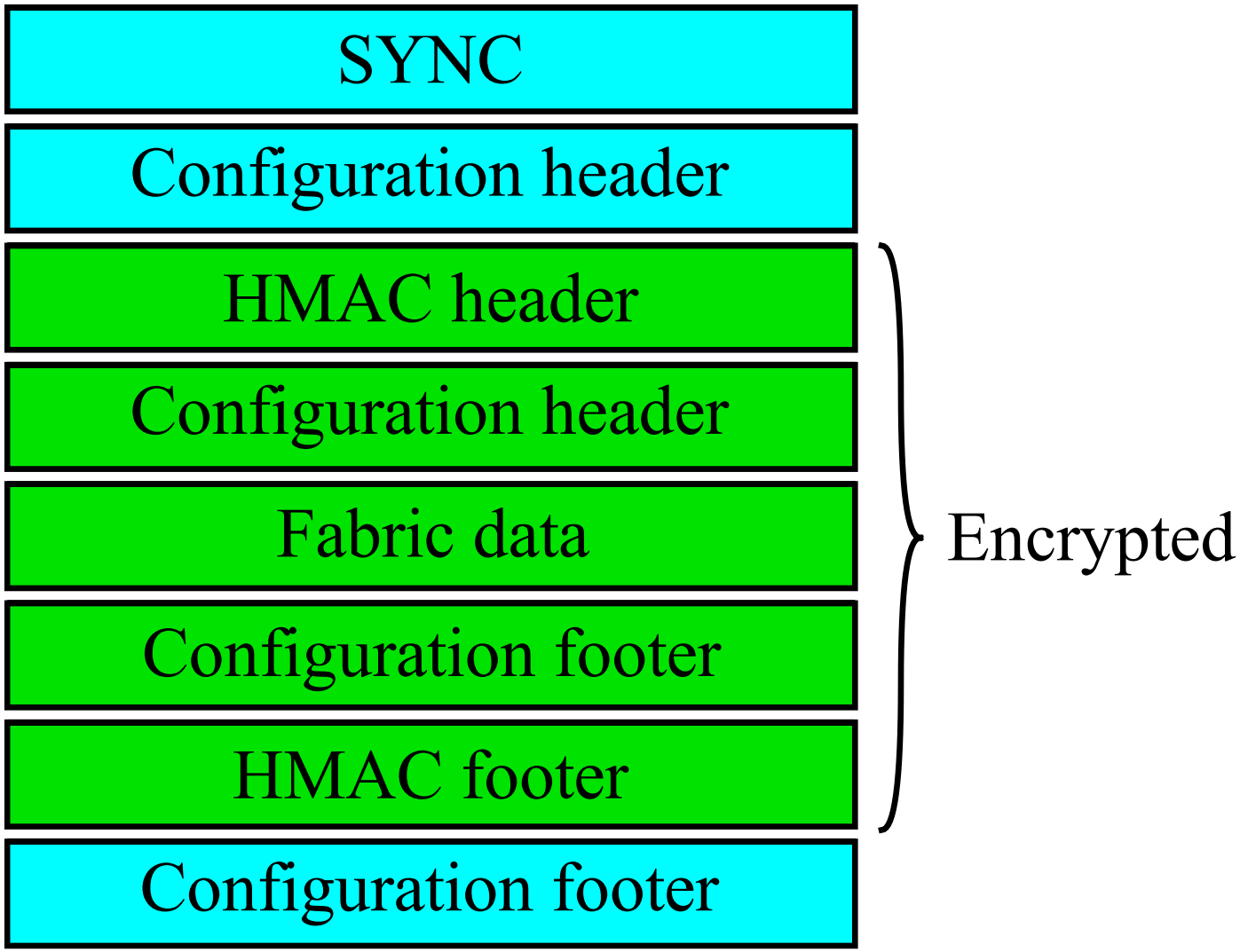
FPGA Dual-Port Block SRAM Block



FPGA Configuration Lifecycle

1. Read encoded configuration file from NVRAM
2. Decode the configuration file
3. Load the file contents into the proper FPGA components

FPGA Configuration Bitstream Structure [1]



FPGA Bitstream Security

1. Configuration bitstream is stored in external non-volatile memory
2. Bitstream is loaded into FPGA on power-up
3. Bitstream is a binary description of the FPGA design. Hence difficult to figure out the design

FPGA Security Attacks

1. Cloning: replicate bitstream for similar FPGA
2. Overproduction: Gaining bitstream allows for privately producing more devices
3. Hardware Trojans: modify bitstream to add a Trojan
4. Side-Channel: Information leaks during FPGA bitstream decryption exposes the secret key
5. Side-Channel: Fault injection introduces errors to test system response

FPGA Security Attacks Continued

1. Replay: where attacker replaces bitstream with a vulnerable one
2. Reverse Engineering: Understand functionality and steal IP
3. Spoofing: Attacker bitstream replaces original one for control of FPGA
4. Tampering: Altering design to leak information, functionality or DoS
5. Bitstream Interception: Allows for parsing the design and steal IP and maybe alter design

FPGA Security Solutions

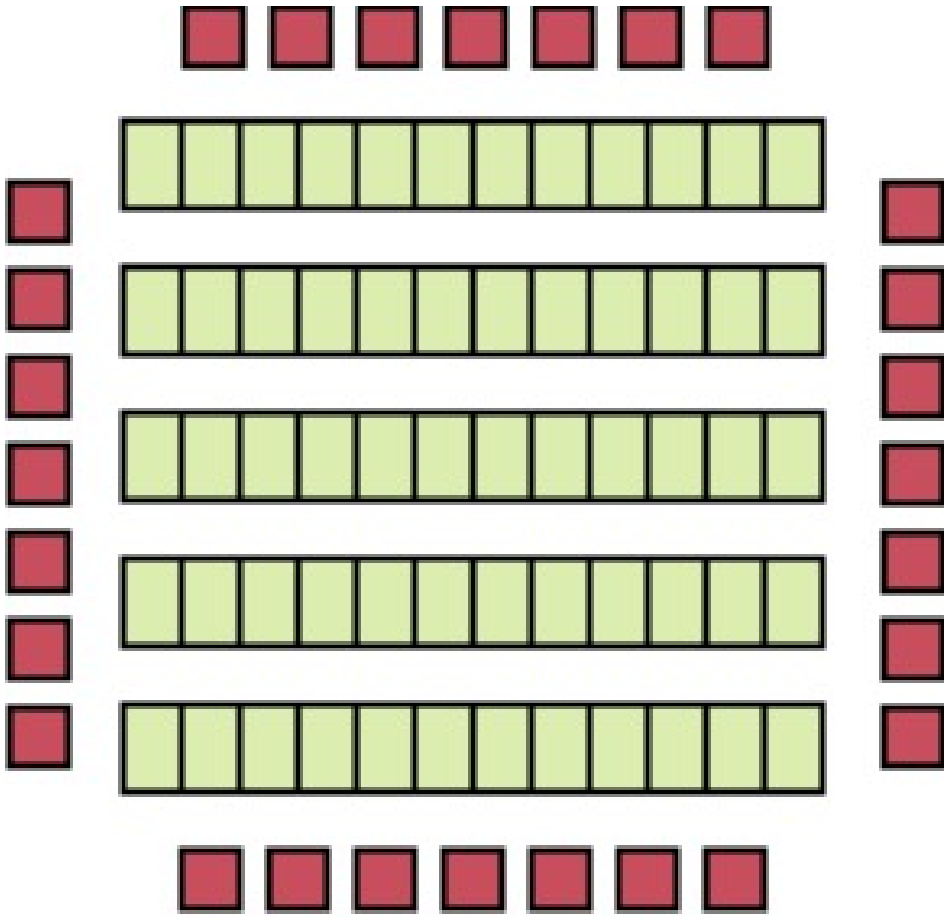
1. Bitstream Authentication: This implies encrypting bitstream and authenticating source and destination
2. Isolate Configuration Process: dedicated bitstream processing not the regular/vulnerable, data processing
3. Using Trusted Platform Module: to store secret keys and perform authentication
4. Watermarks: unique ID to prevent overproduction, cloning and forgeries
5. Bitstream Obfuscation: to prevent reverse engineering by introducing random circuits and functions

ASIC Design

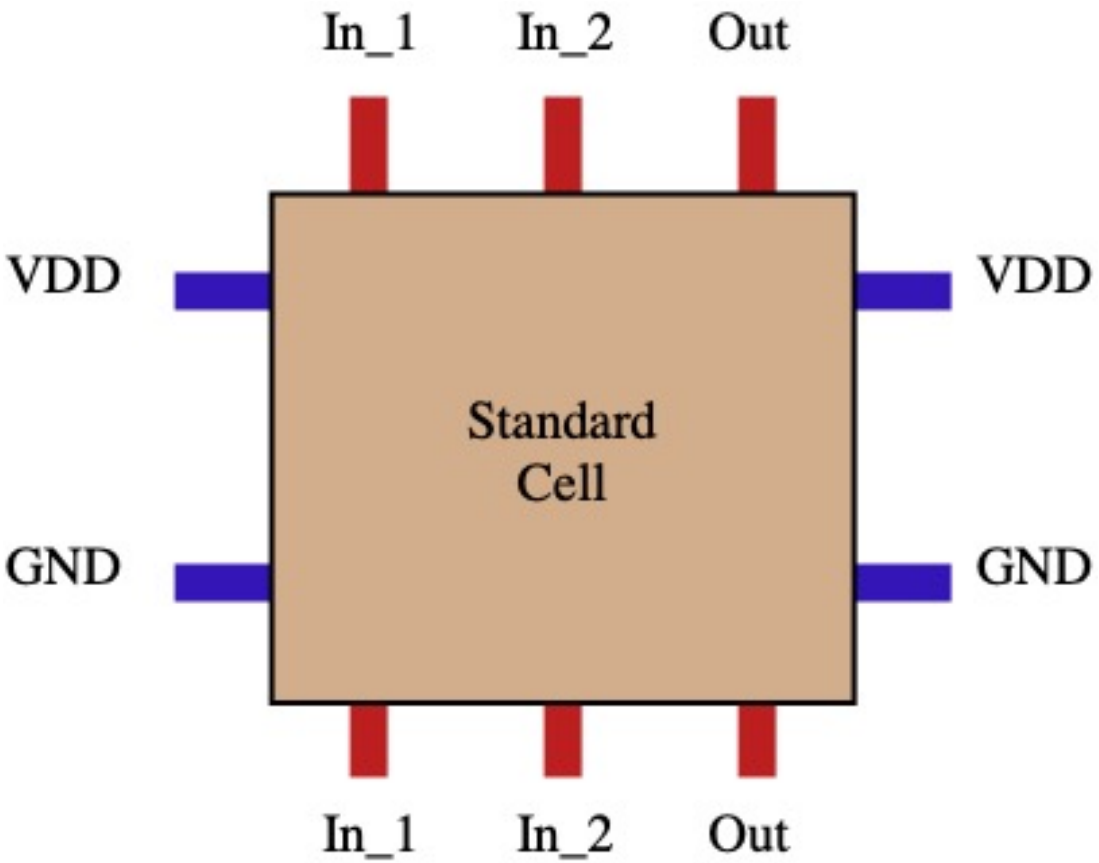
ASIC Design Features

1. Based on standard cell design methodologies
2. Control placement and routing of modules
3. Optimize performance
4. Very time consuming
5. Error prone

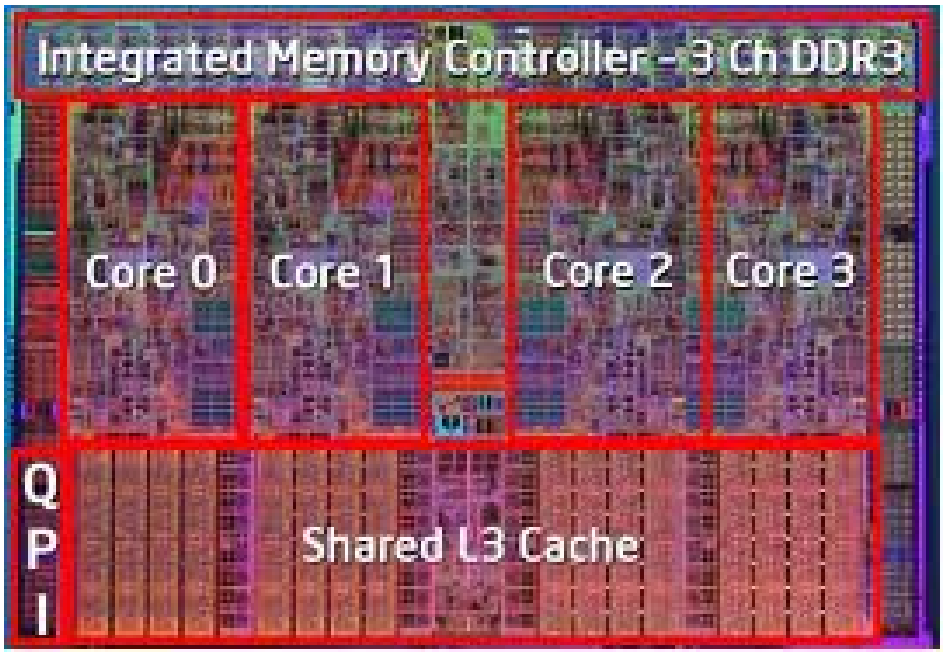
Standard Cells



Standard Cells



What is an Integrated Circuit



[1] M. Ender, A. Moradi, and C. Paar, “The unpatchable silicon: A full break of the bitstream encryption of xilinx 7-series FPGAs,” in *USENIX Security Symposium*, 2020, pp. 1803–1891.