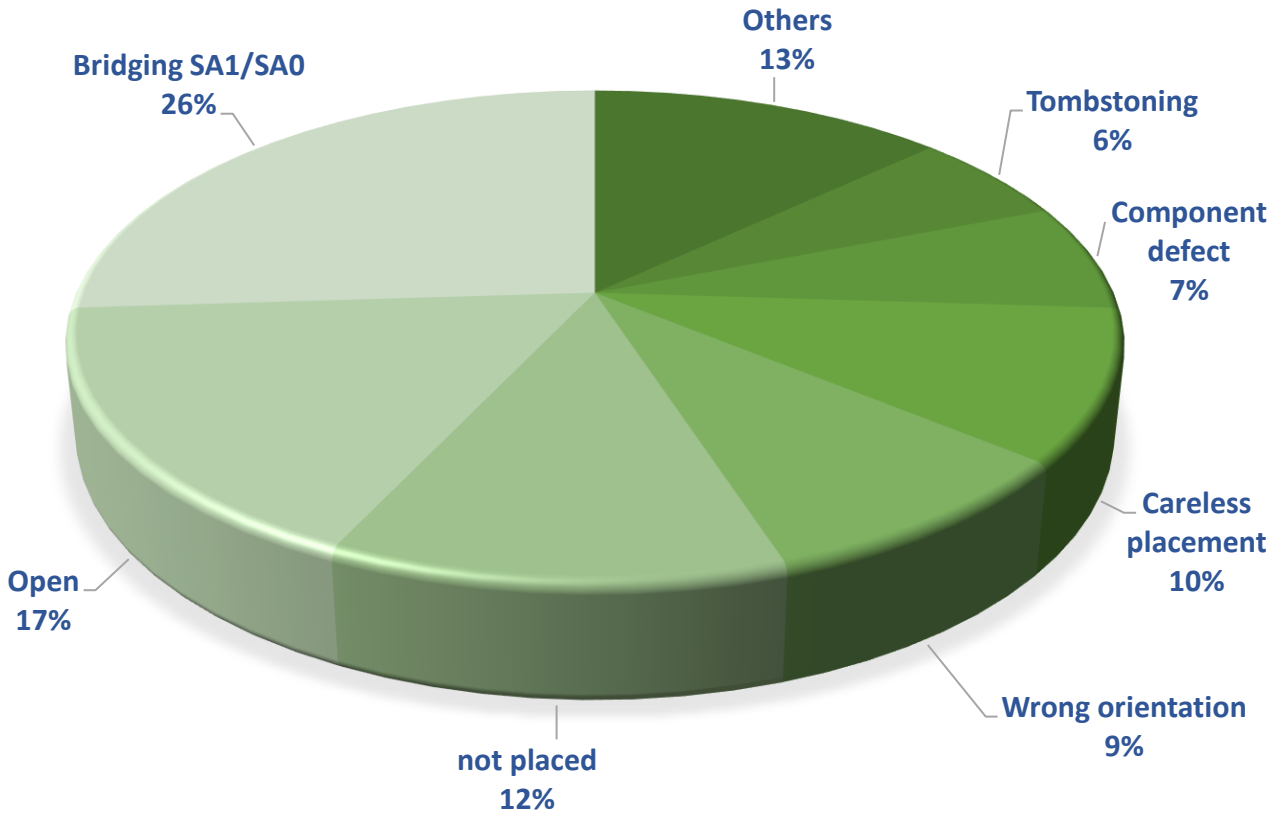
Sneller op de markt door al tijdens de schema fase de HW validatie mogelijkheden vast te stellen

Rik Doorneweert
rik@jtag.com

Typical Production line (SMT)

Goal: zero defects strategy

Typical fault spectrum



Errors due to bad production process or bad design?

Anyhow, find the failures before the end-users of the product do !!!

Which test methods to use for my design?

X-RAY

Rontgen

ICT

In Circuit Test

FP

Flying Probe

Testability Report

Coverage of each Test Method

Nets…%

Pins….%

Components….%

FCT

Functional Test

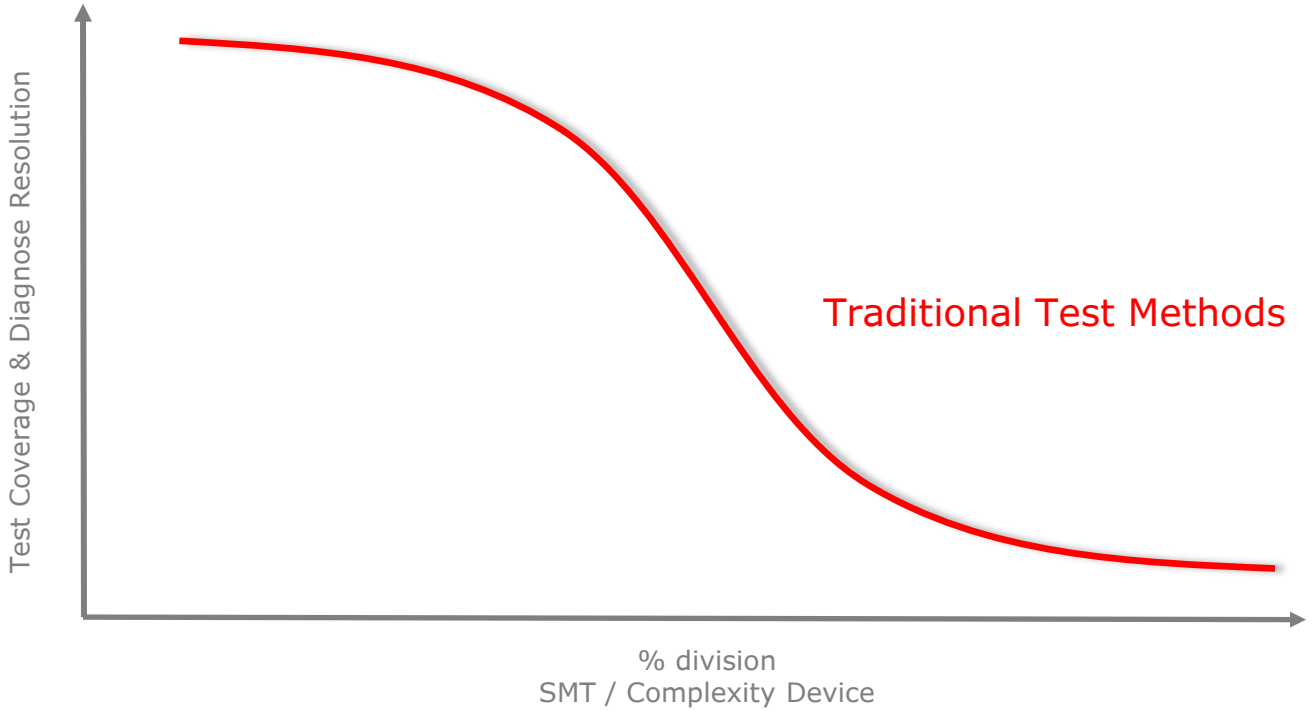
AOI

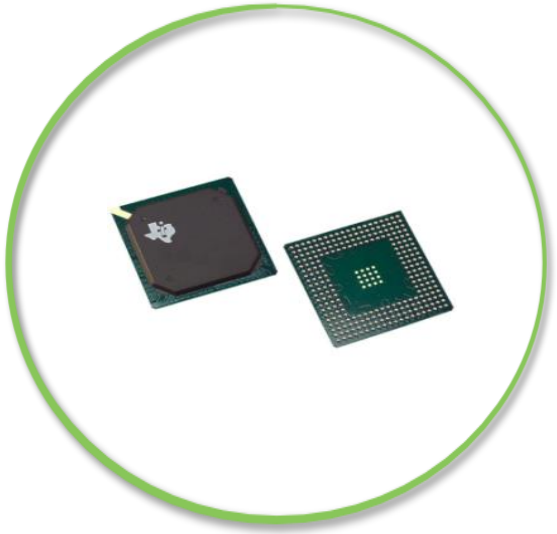
Automatic Optical Inspection

JTAG

Solutions

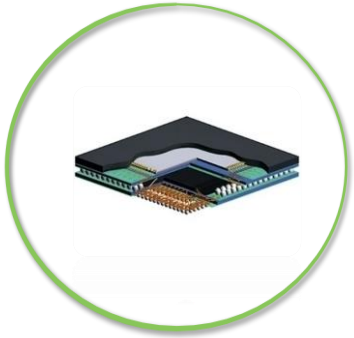
Miniaturization and its consequences….





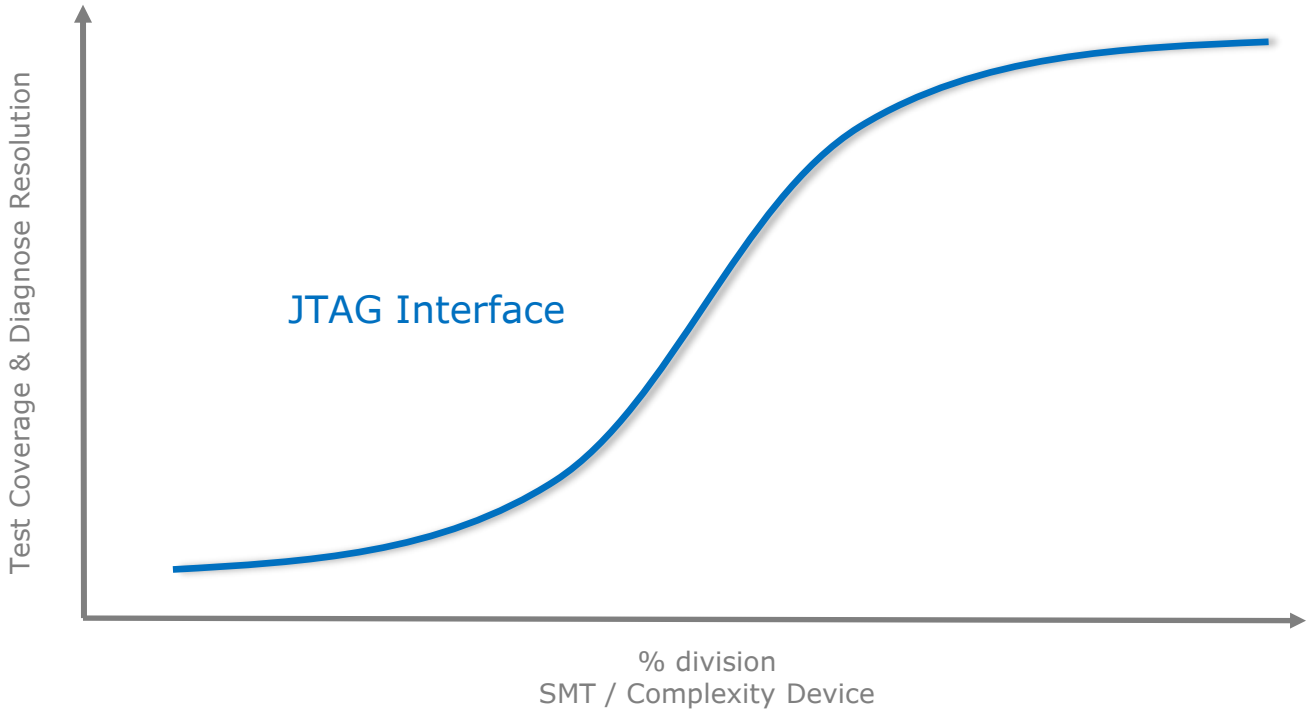
Reduced number of physical test points (ICT)

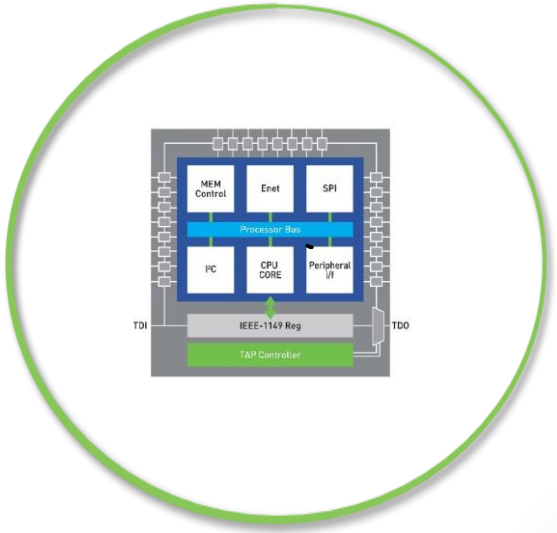
More efforts to develop a test program



More complex failure localization

Contribution of the JTAG Interface



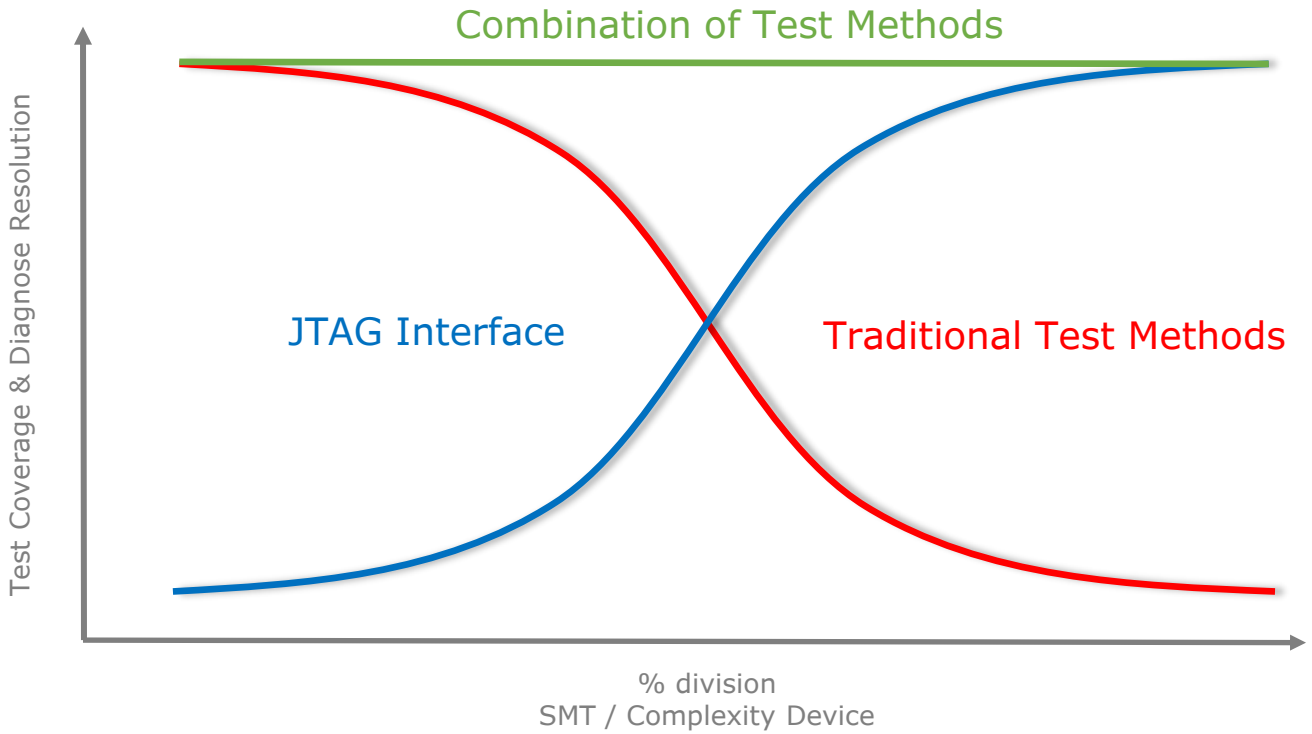


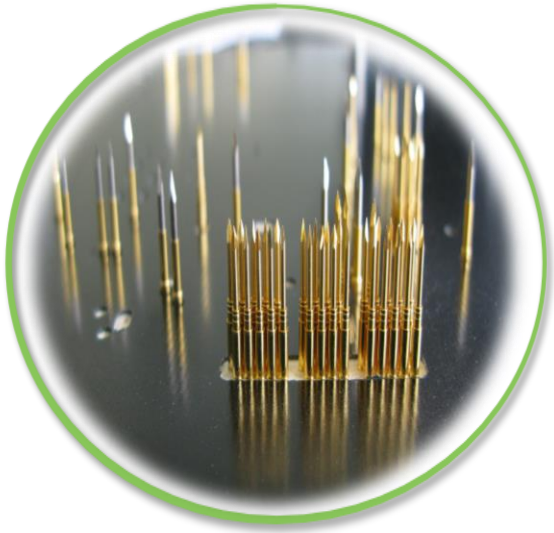
Firmware independent

IEEE 1149.x

Embedded Test

Best solution: Combine test methods



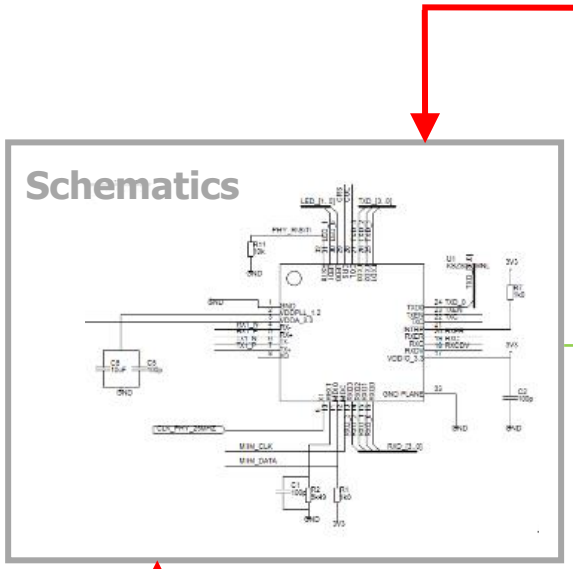


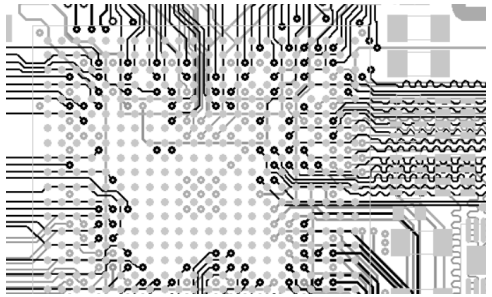
Max. Test coverage

Synergy

Optimal Failure localization

Typical HW engineering process





Prototype validation

Functional Test Creation

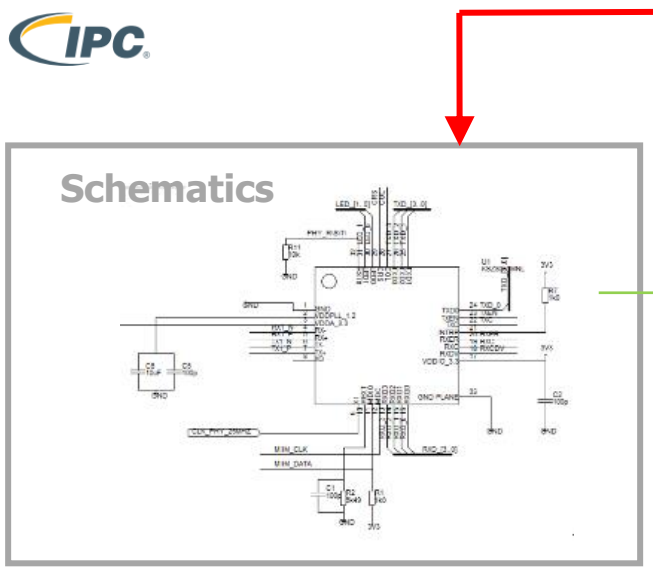
Production (EMS)

Test Methods:

- SPI
- AOI
- ...
- Functional test

- ❑ Coverage?
- ❑ Execution times?

Better HW engineering process



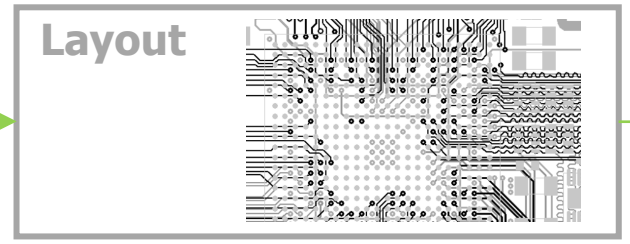
Accessibility

Testability

(Bscan) **Test Generation**

Coverage

Production (EMS)



HW Engineer and Test Engineer – cooperation

(2 sides of the design)

Layout and Test Engineer

Questions to get to best fitting Test Strategy

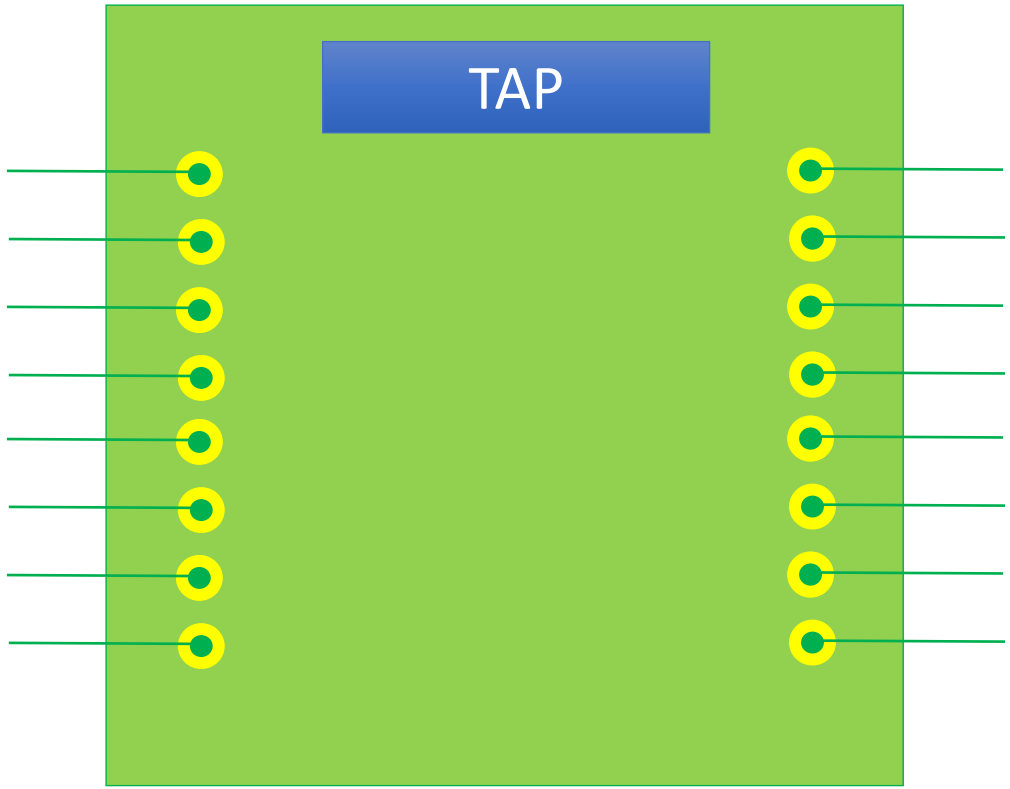
1. What is the expected Production volume?
2. What is the expected lifetime of this board?
3. What is the total cost of the BOM?
4. How many productions errors are acceptable, what is the impact of a failing board?
5. What can you as HW engineer do to help the EMS to achieve zero defects delivery?
6. Which test methods performs the EMS partner by default?
7. What is the expected coverage per applied test method?
8. What is the level of pinpointing diagnostics of each of the test methods we apply?
9. What is the expertise of the person that will repair the boards that fail after each of the test steps?
10. How to program the programmable device(s), production tool or engineering tool?

Basics of Boundary scan

- **Chip level**
- **Test possibilities on board level**
- **Programming via the JTAG interface**

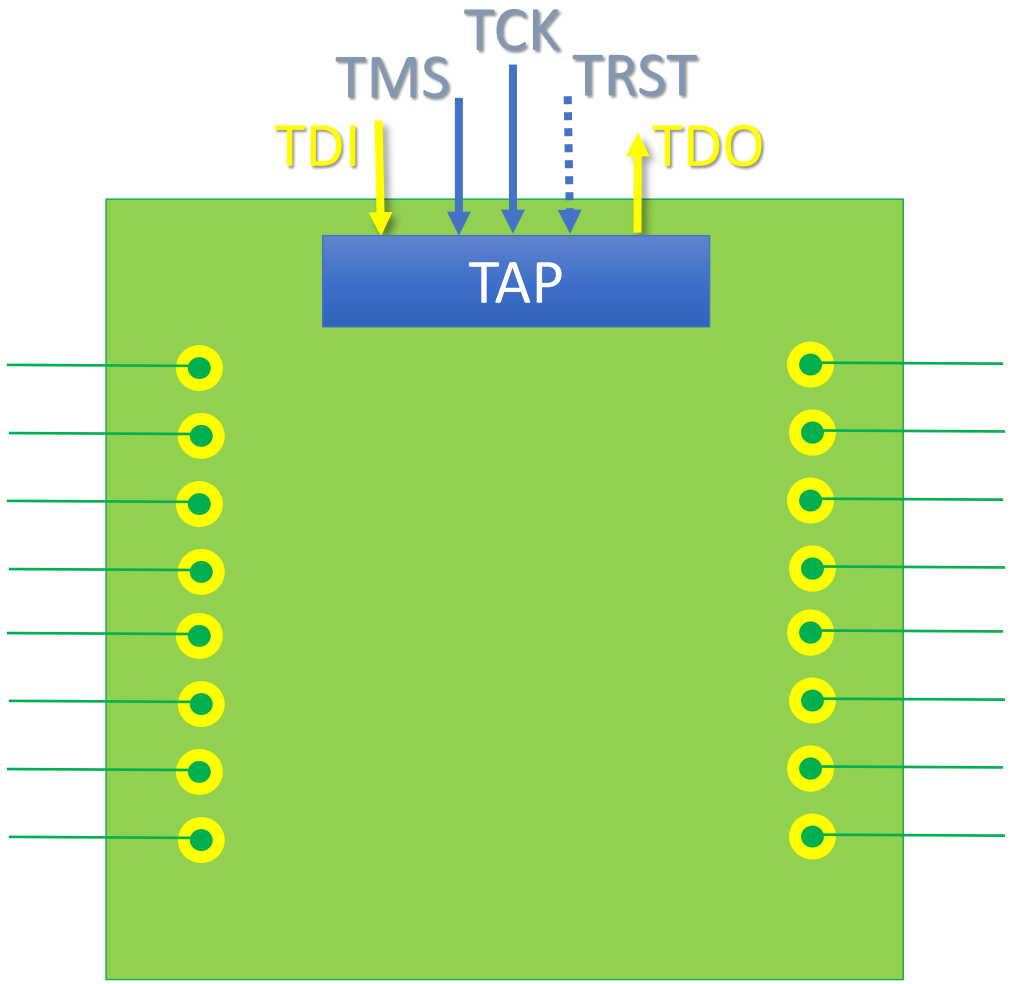
Basics – Chip Level

- Test Access Port (TAP)



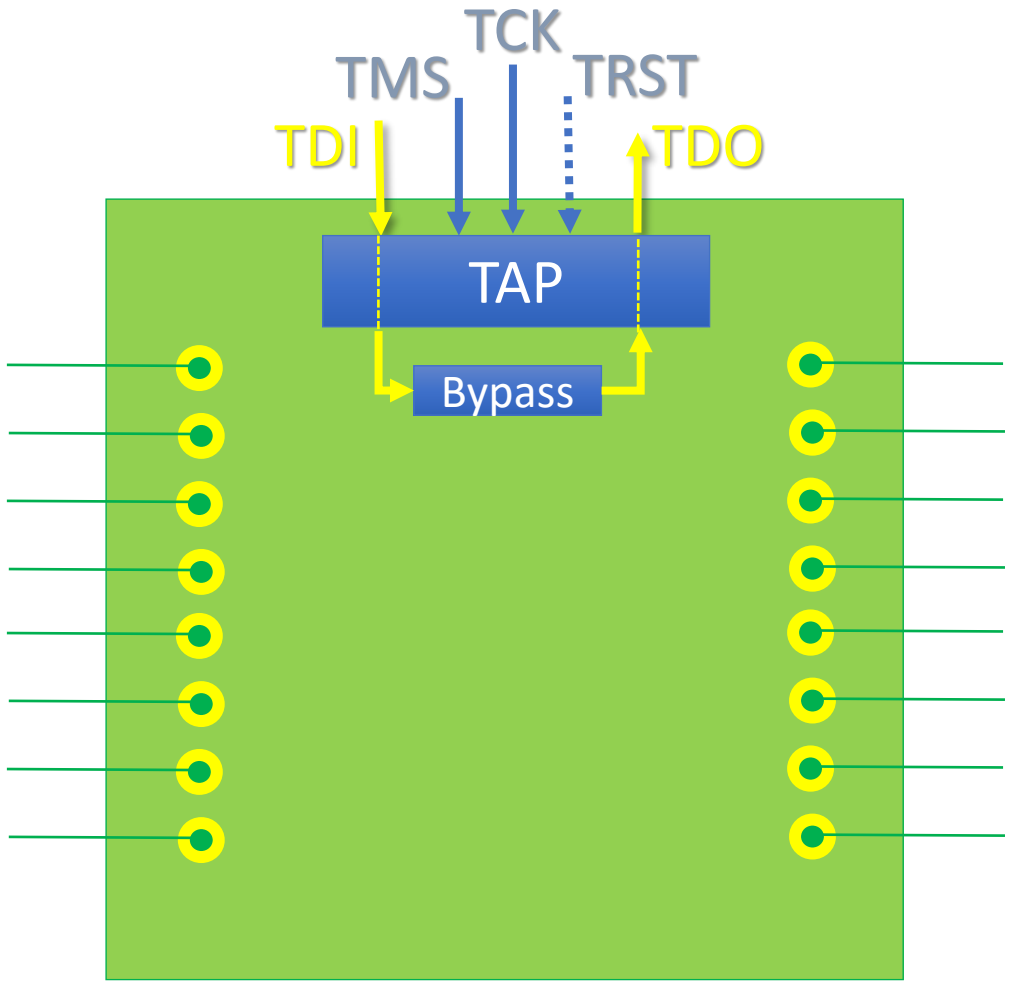
Basics – Chip Level

- Test Access Port (TAP)
- TAP Signals:
  - Test Data Input
  - Test Data Output
  - Test Mode Signal
  - Test Clock
  - Test Reset



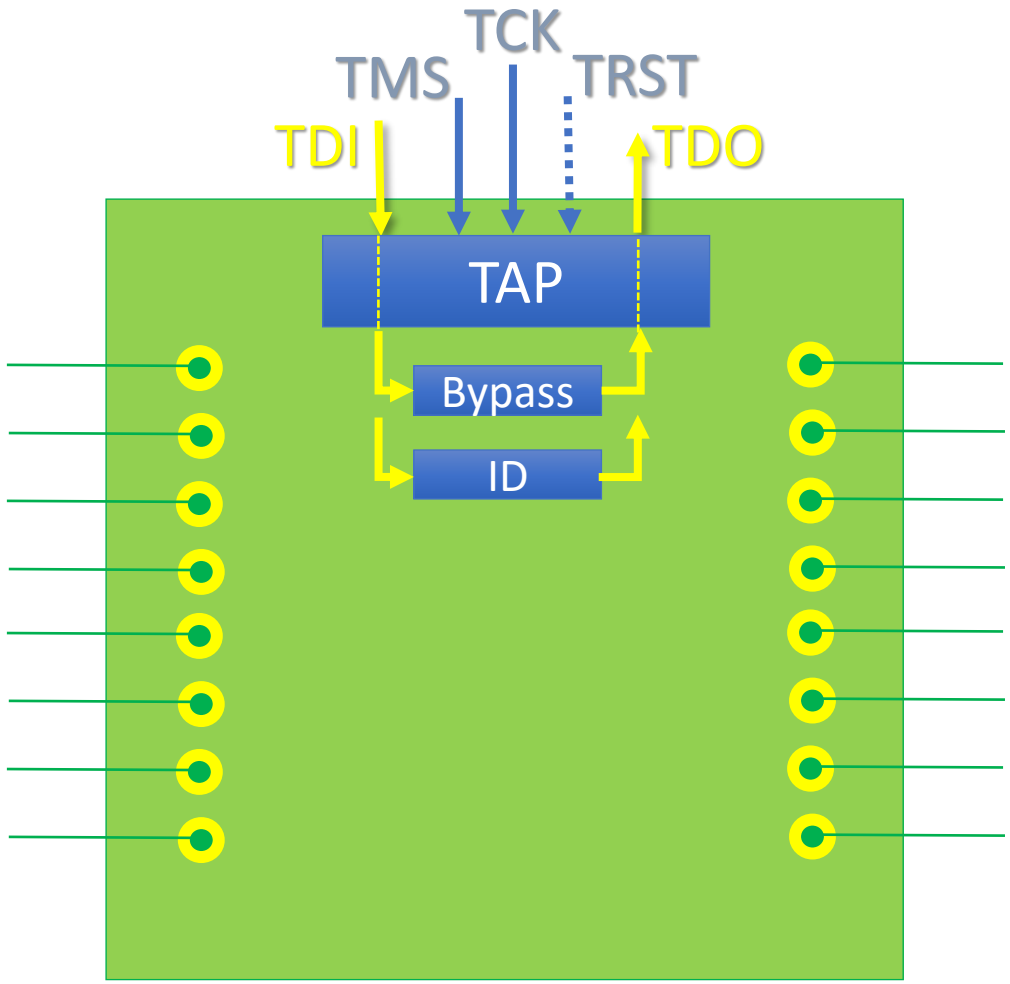
Basics – Chip Level

- Test Access Port (TAP)
- TAP Signals:
  - Test Data Input
  - Test Data Output
  - Test Mode Signal
  - Test Clock
  - Test Reset
- Bypass register



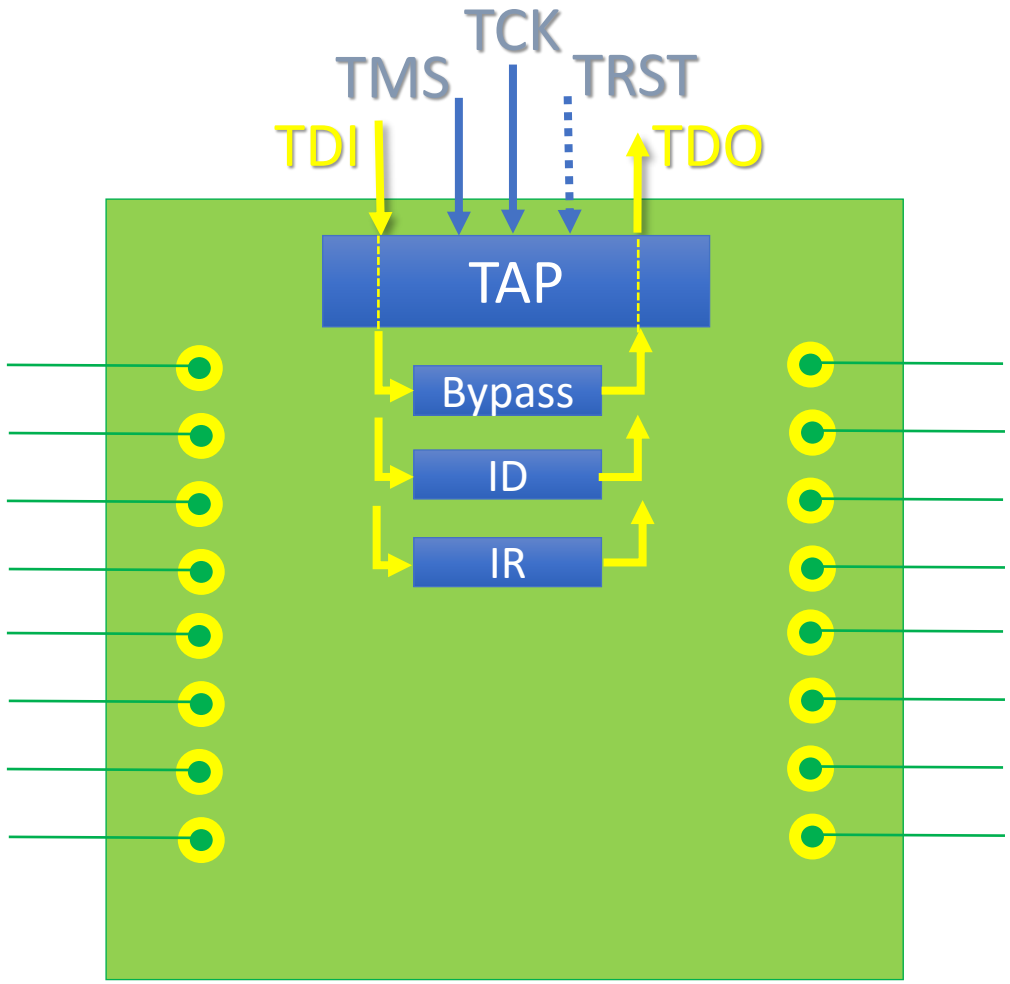
Basics – Chip Level

- Test Access Port (TAP)
- TAP Signals:
  - Test Data Input
  - Test Data Output
  - Test Mode Signal
  - Test Clock
  - Test Reset
- Bypass register
- Identification register



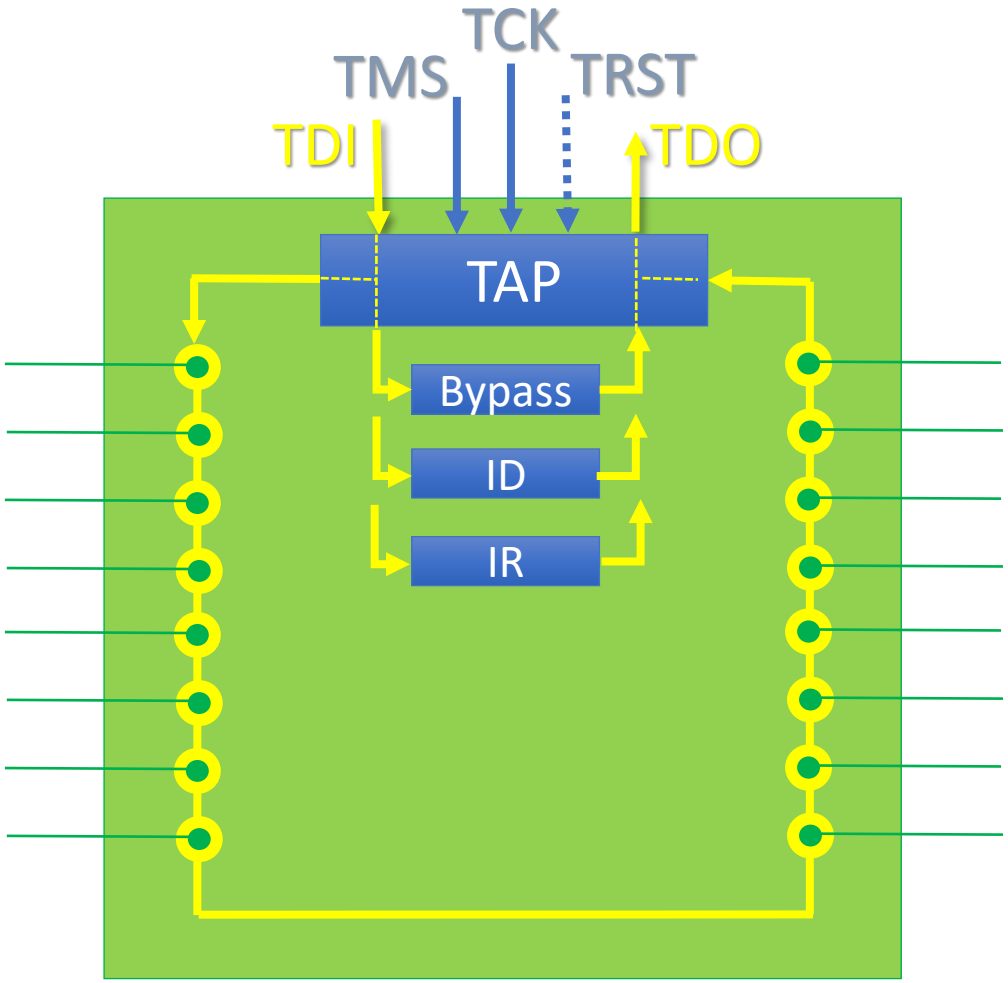
Basics – Chip Level

- Test Access Port (TAP)
- TAP Signals:
  - Test Data Input
  - Test Data Output
  - Test Mode Signal
  - Test Clock
  - Test Reset
- Bypass register
- Identification register
- Instruction register



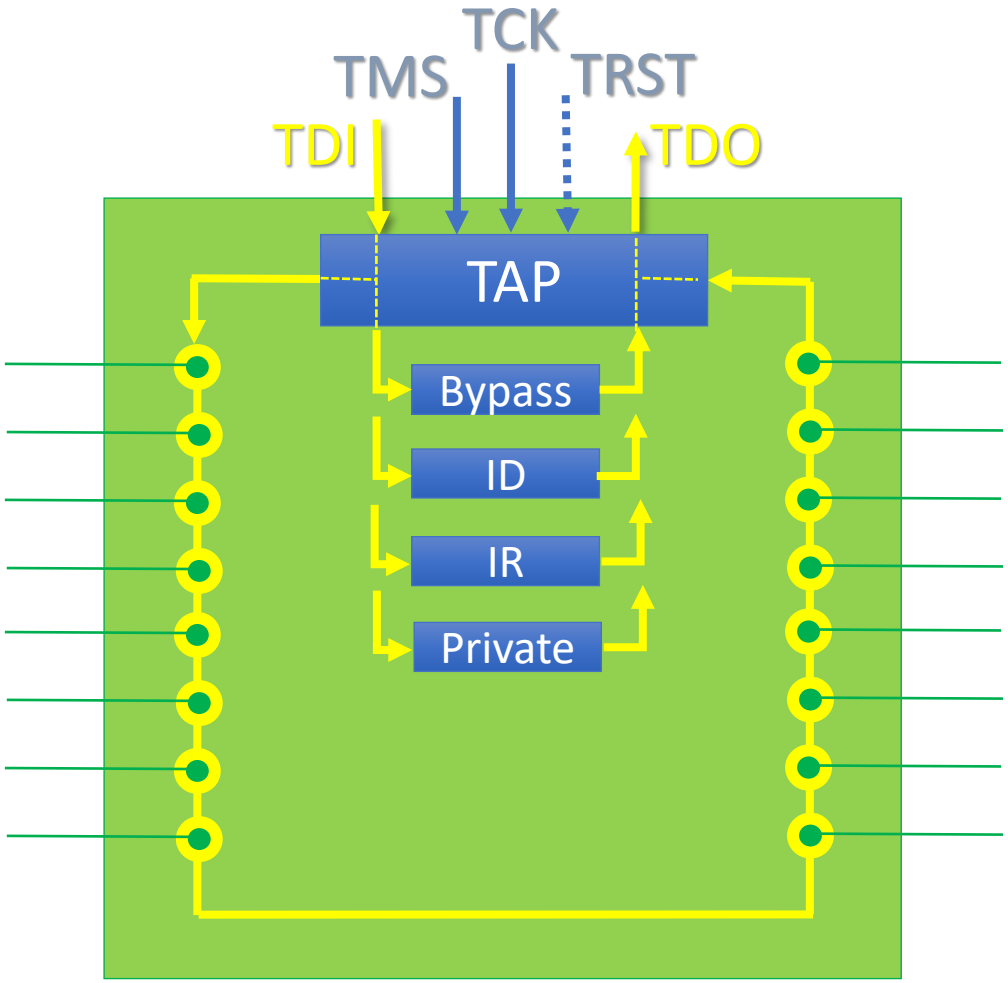
Basics – Chip Level

- Test Access Port (TAP)
- TAP Signals:
  - Test Data Input
  - Test Data Output
  - Test Mode Signal
  - Test Clock
  - Test Reset
- Bypass register
- Identification register
- Instruction register
- Boundary Scan register



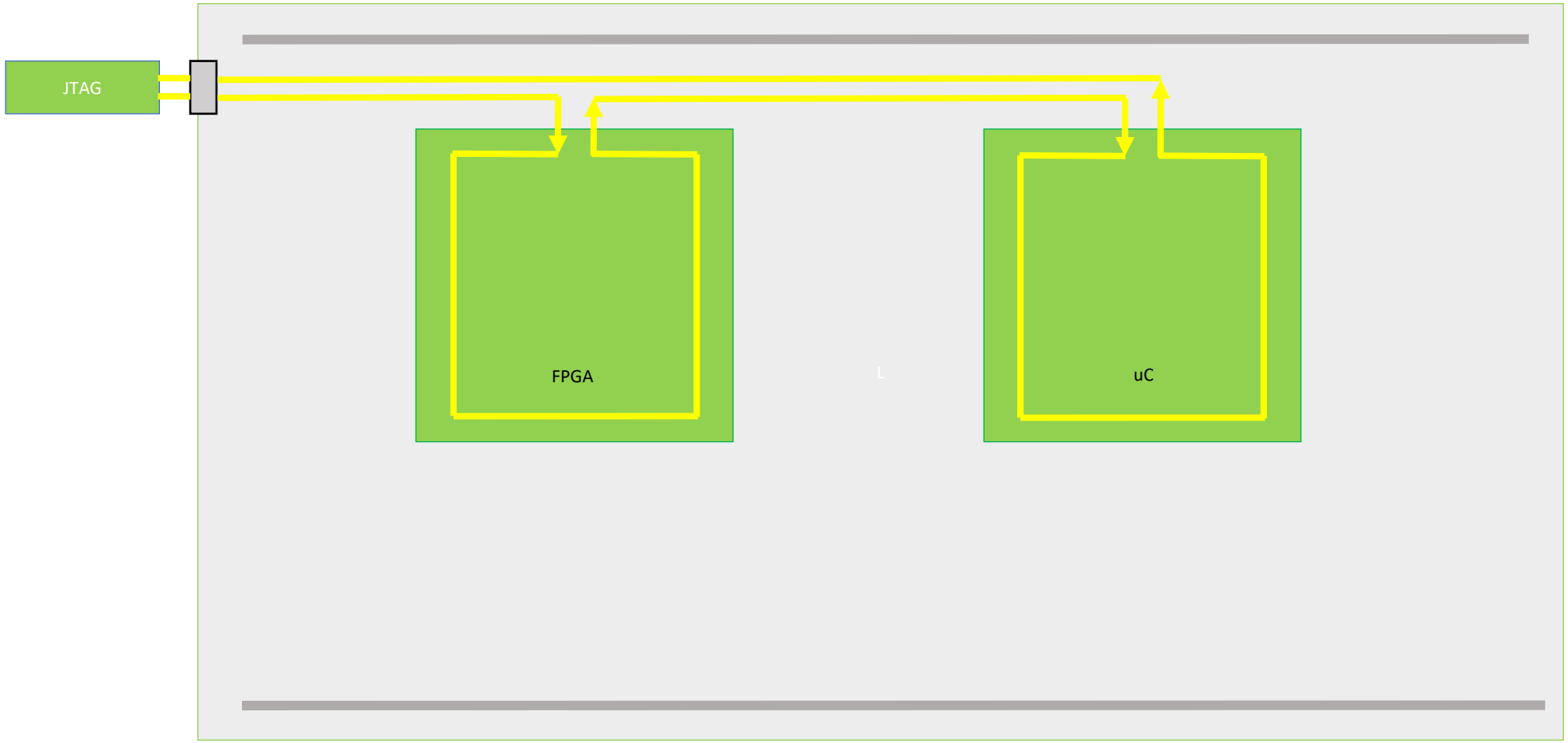
Basics – Chip Level

- Test Access Port (TAP)
- TAP Signals:
  - Test Data Input
  - Test Data Output
  - Test Mode Signal
  - Test Clock
  - Test Reset
- Bypass register
- Identification register
- Instruction register
- Boundary Scan register
- Private



Basics - Test Applications

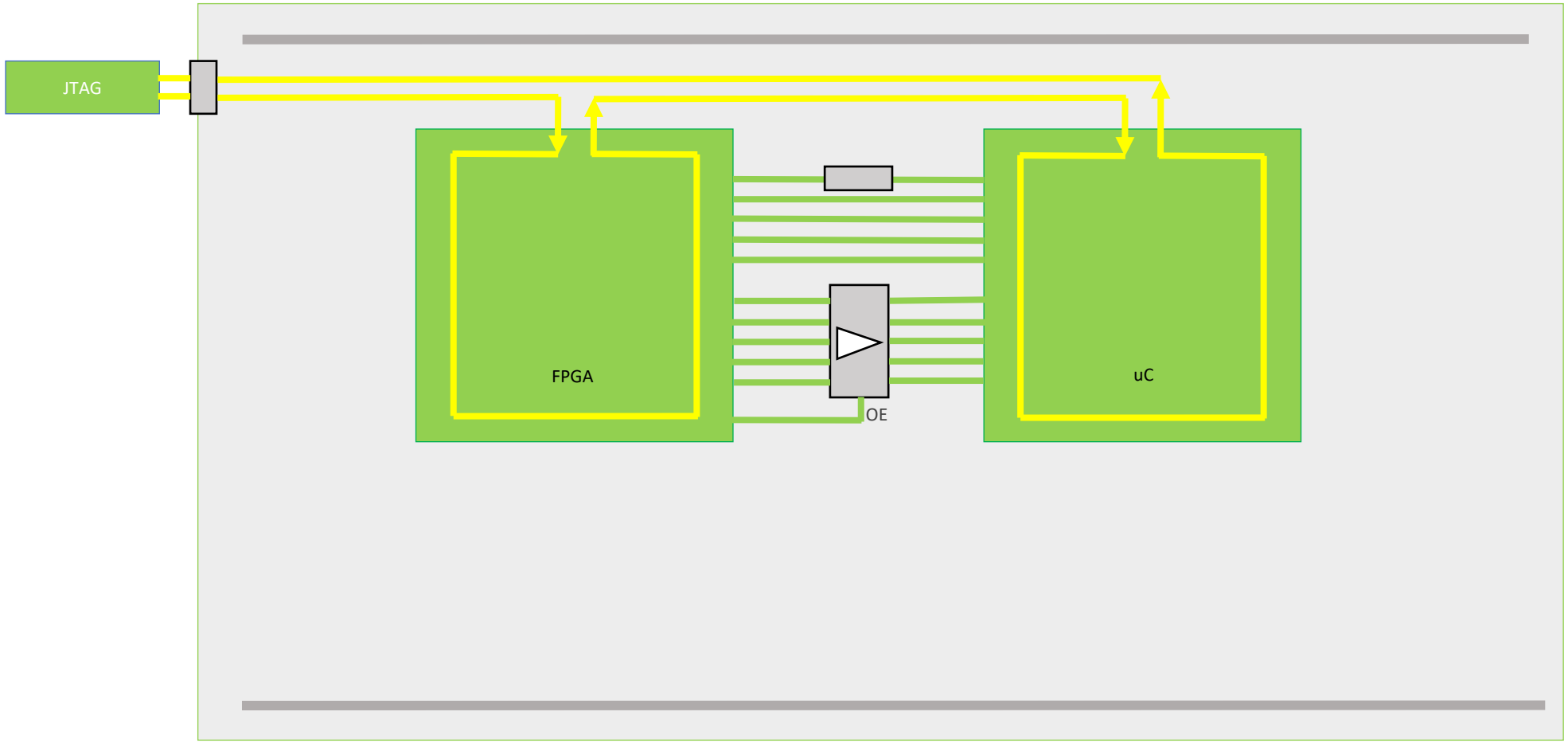
- Chain



Capture value | Identification value | TRST connection | Boundary Scan Register length

Basics - Test Applications

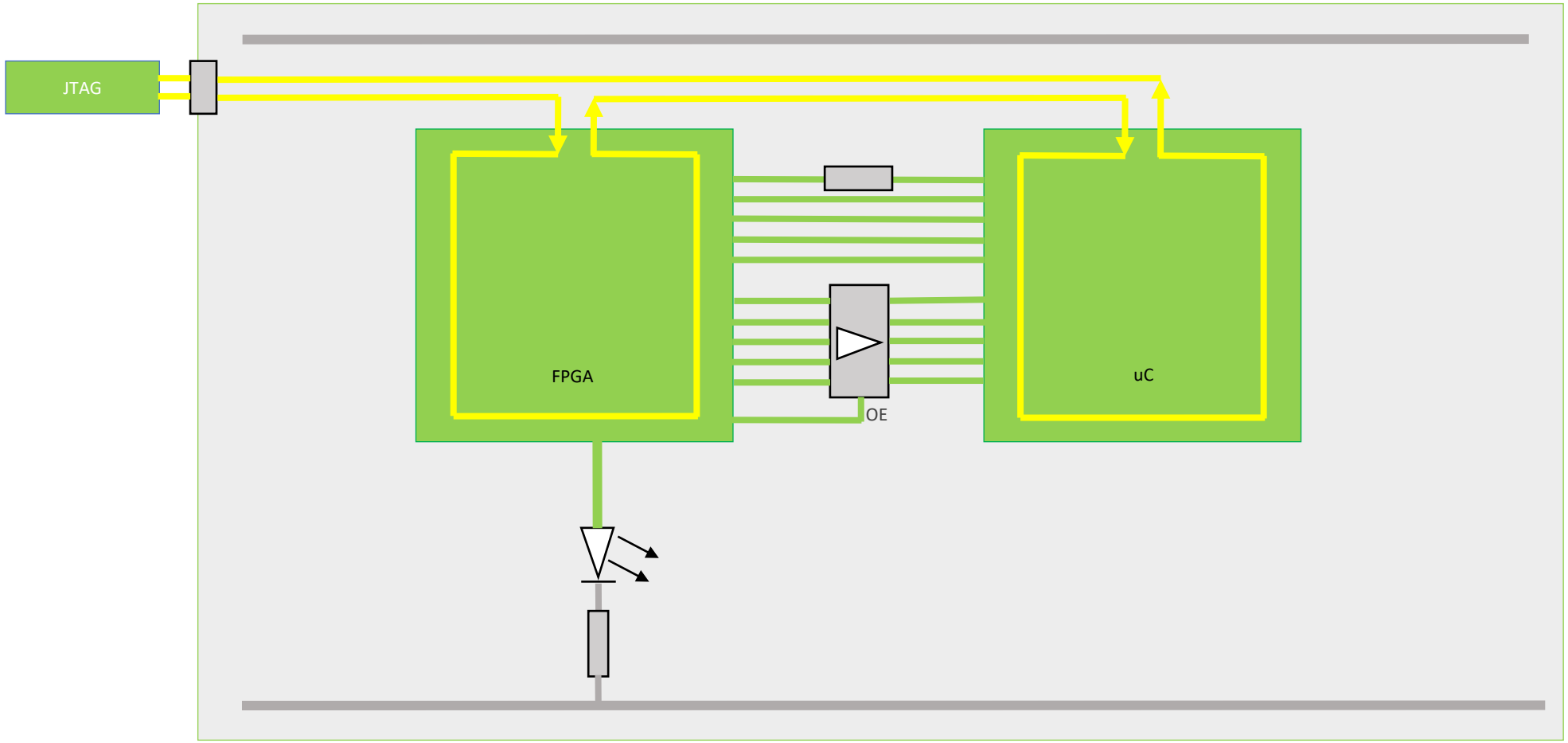
- Chain
- Interconnect



Testing on: Opens, Shorts , Stuck at 1 (SA1), Stuck at 0 (SAO)

Basics - Test Applications

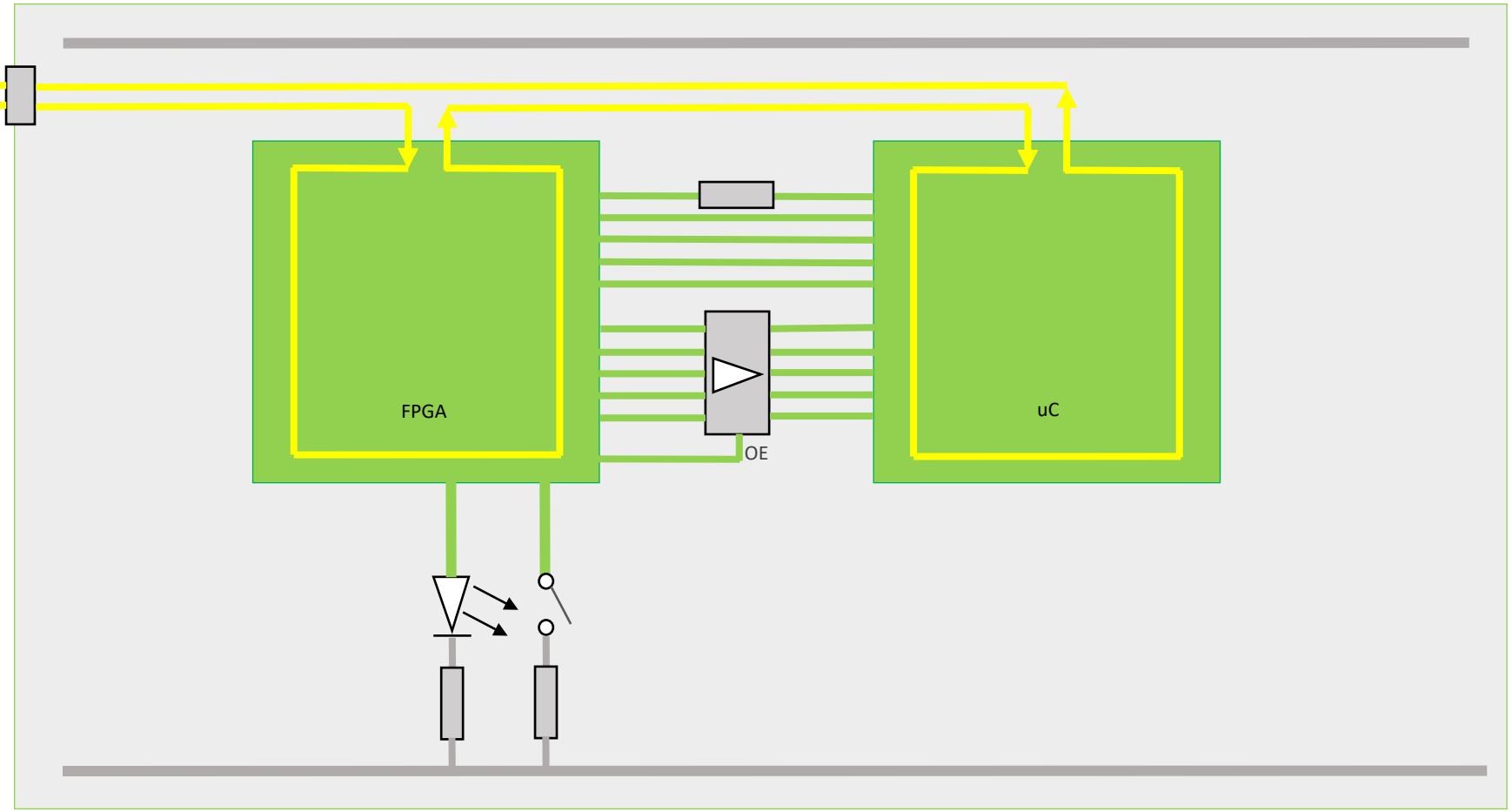
- Chain
- Interconnect
- LEDs



Drive LEDs – observe LEDs manually or by LED analyzer

Basics - Test Applications

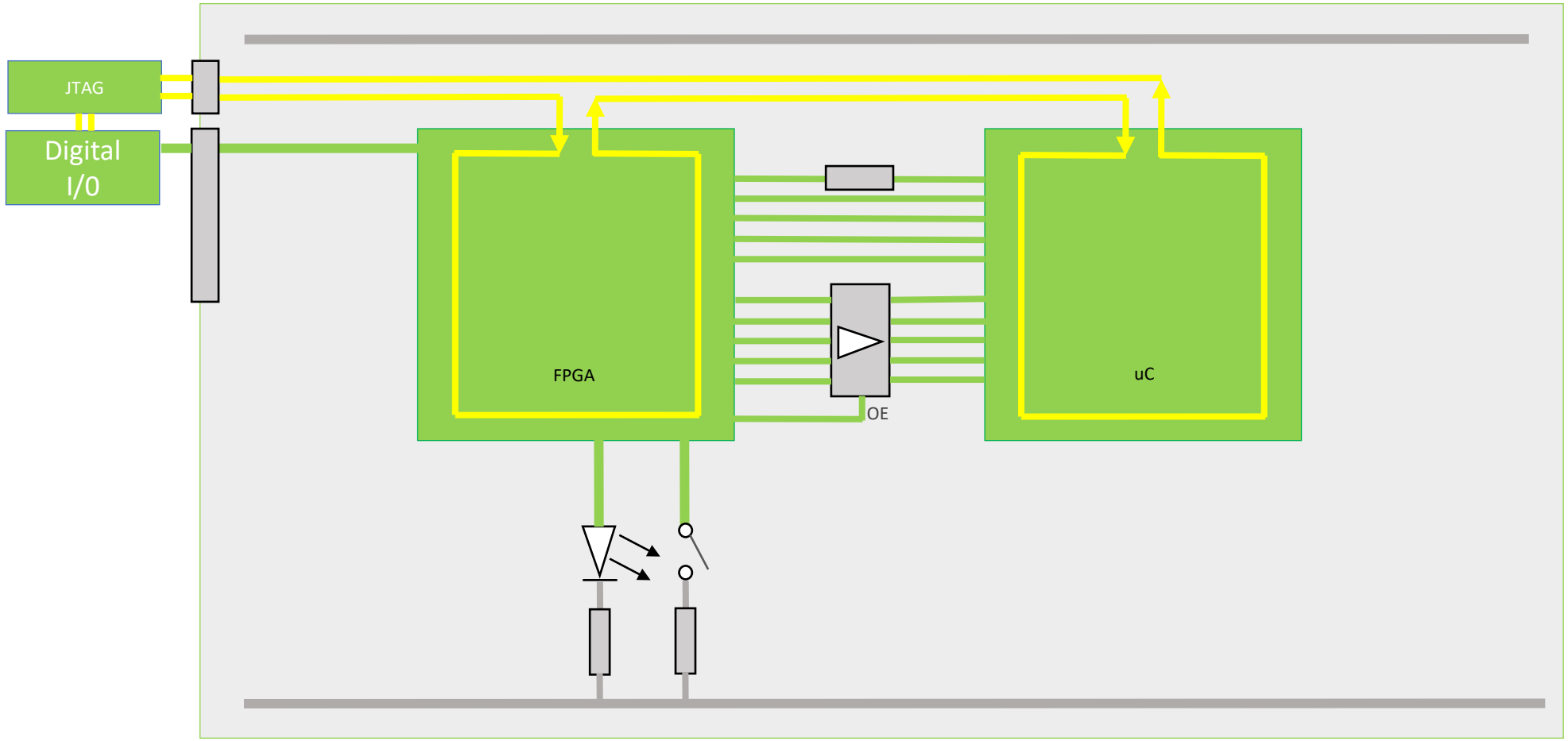
- Chain
- Interconnect
- LEDs
- Switches



Testing switch positions

Basics - Test Applications

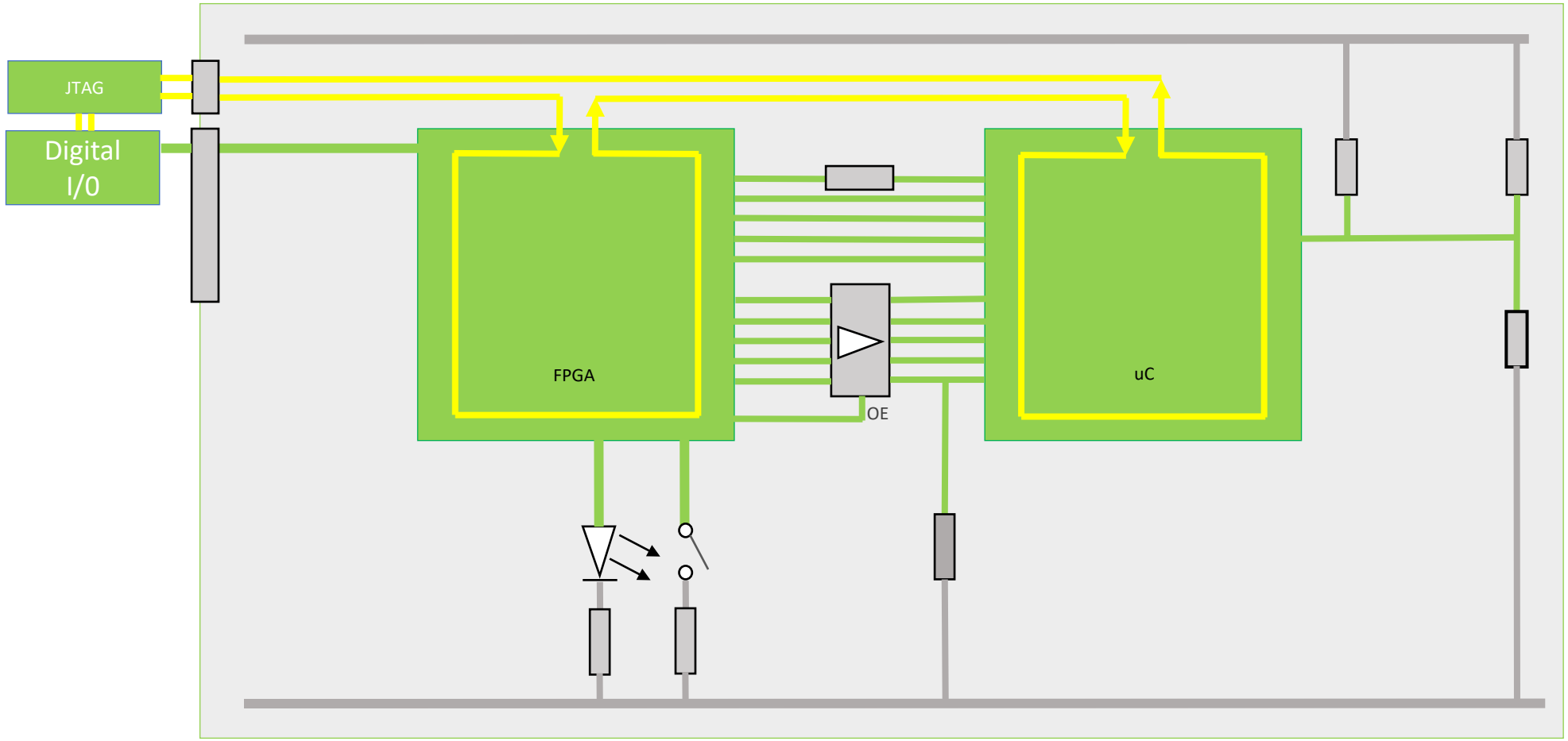
- Chain
- Interconnect
- LEDs
- Switches
- Connectors



Testing connections through connectors and via physical test points

Basics - Test Applications

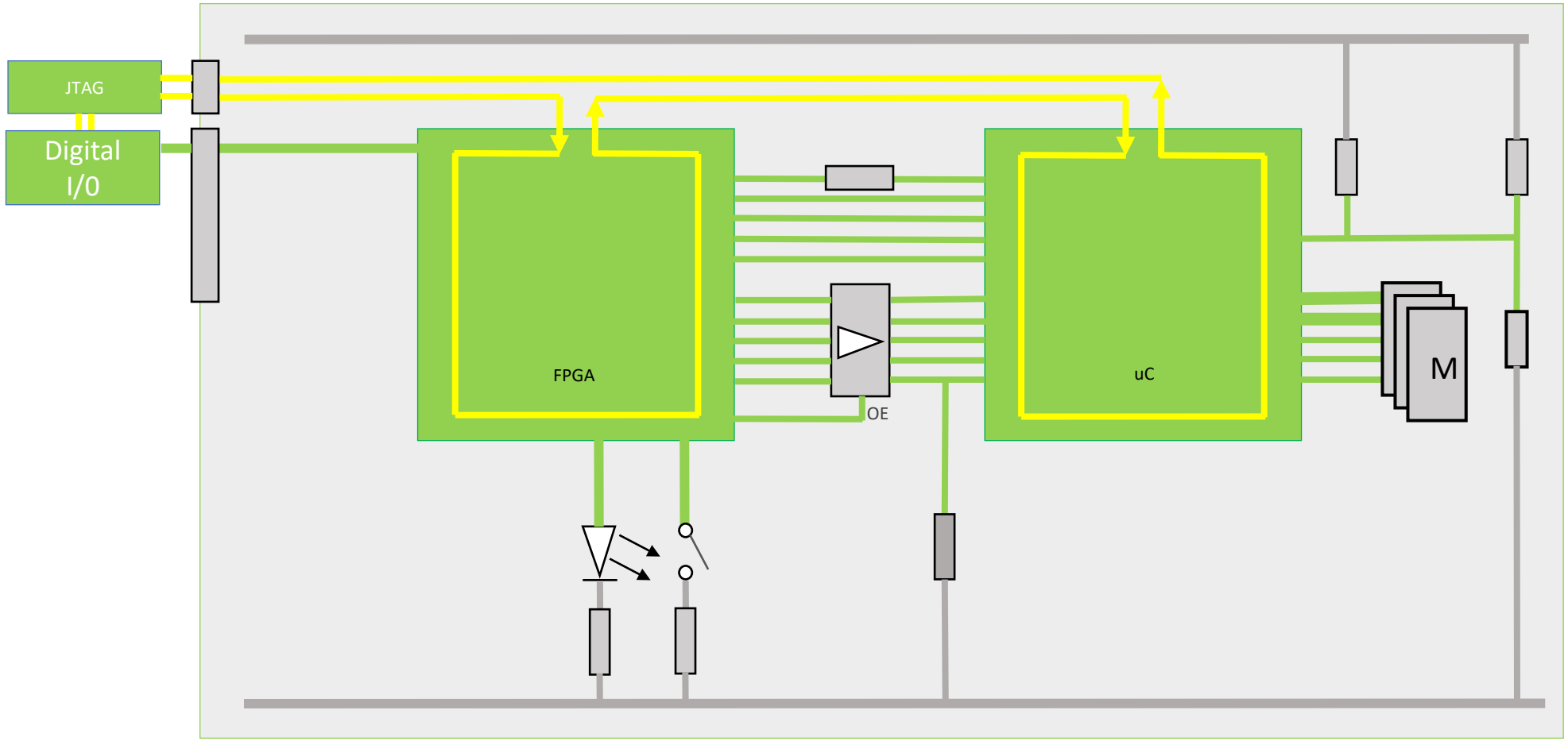
- Chain
- Interconnect
- LEDs
- Switches
- Connectors
- Resistors presence



Check presence of Pull up / Pull down resistors

Basics - Test Applications

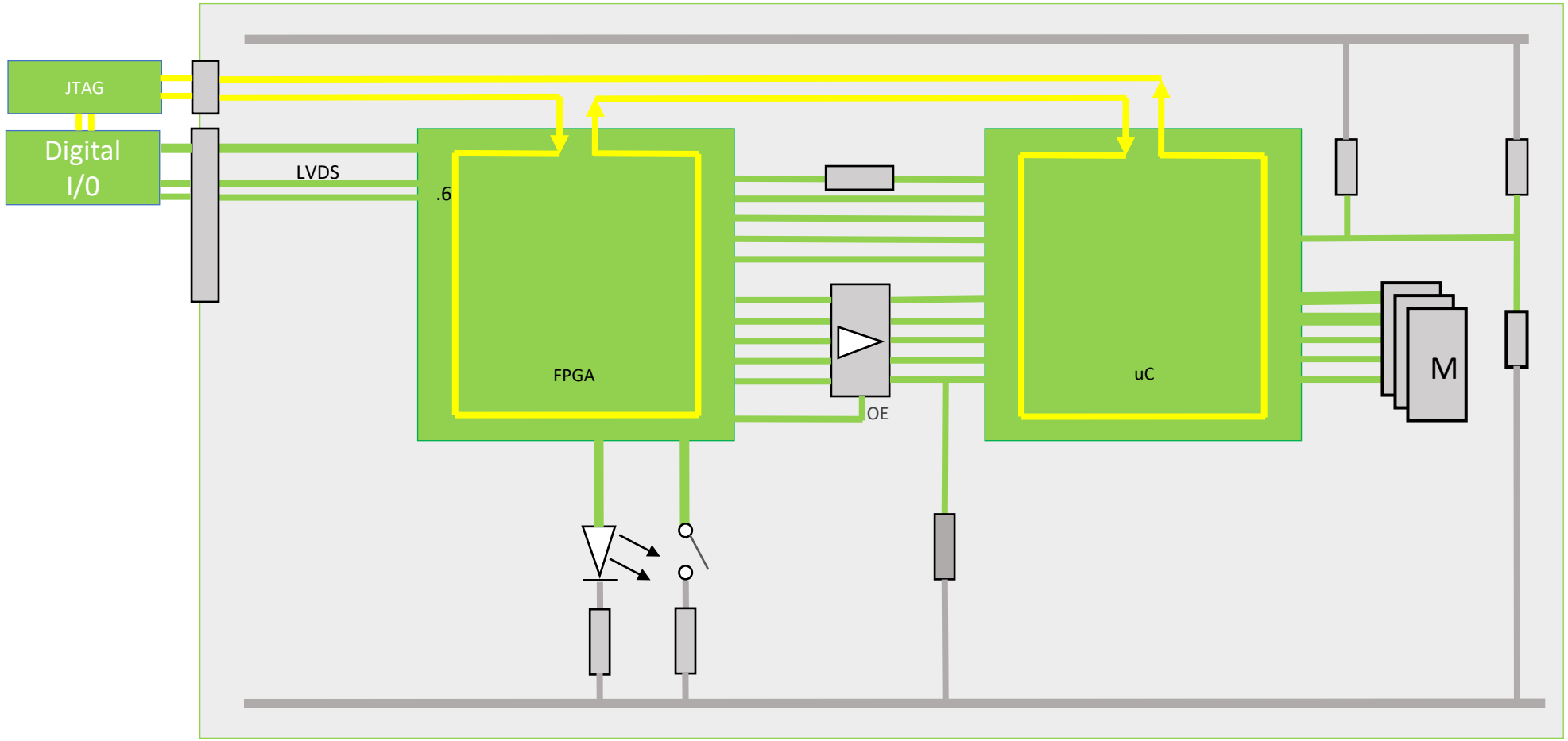
- Chain
- Interconnect
- LEDs
- Switches
- Connectors
- Resistors presence
- Memory connections



Test address - , data – and control lines of SRAM, DRAM, SDRAM, DDR2/3/4 etc.

Basics - Test Applications

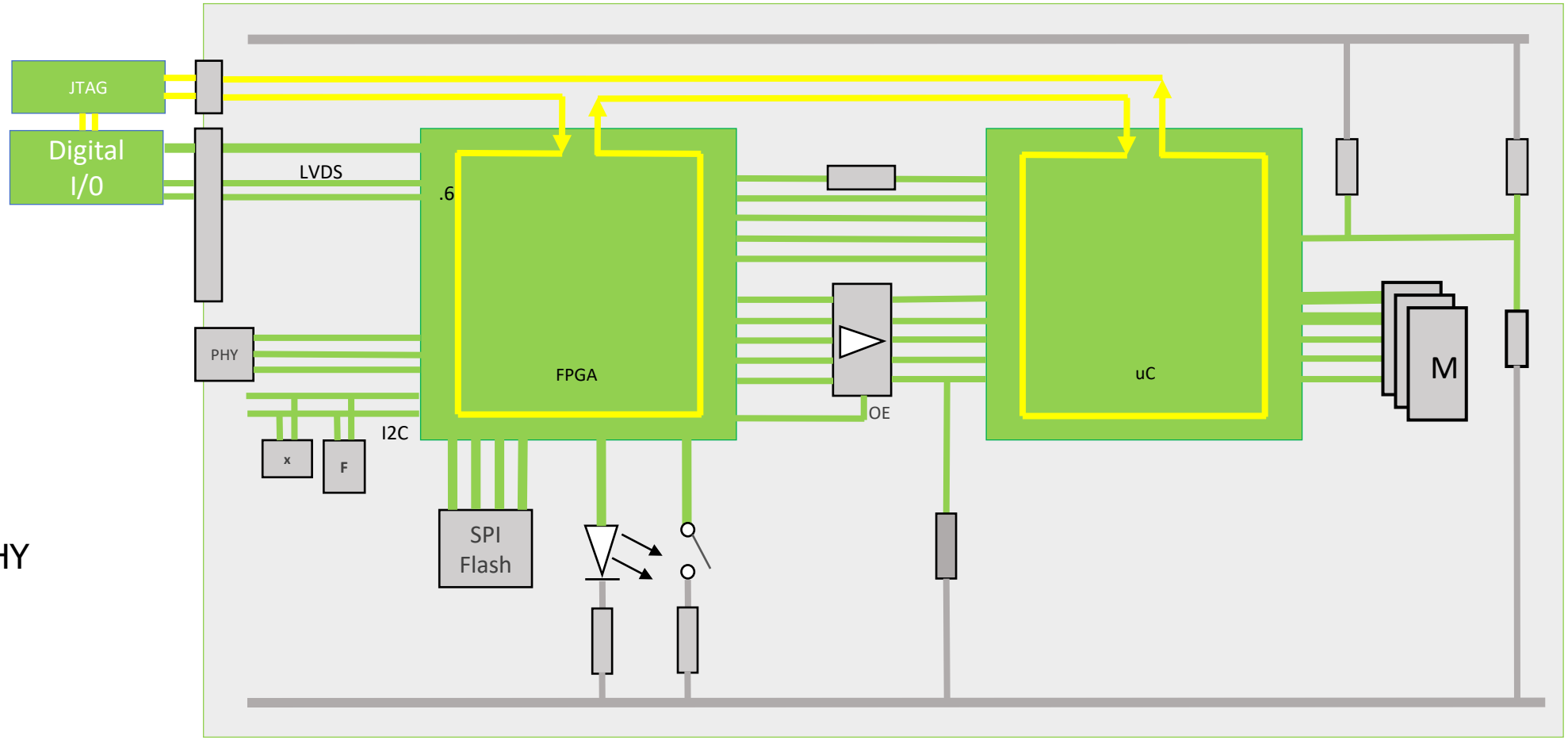
- Chain
- Interconnect
- LEDs
- Switches
- Connectors
- Resistors presence
- Memory connections
- LVDS



LVDS connections (IEEE 1149.6)

Basics - Test Applications

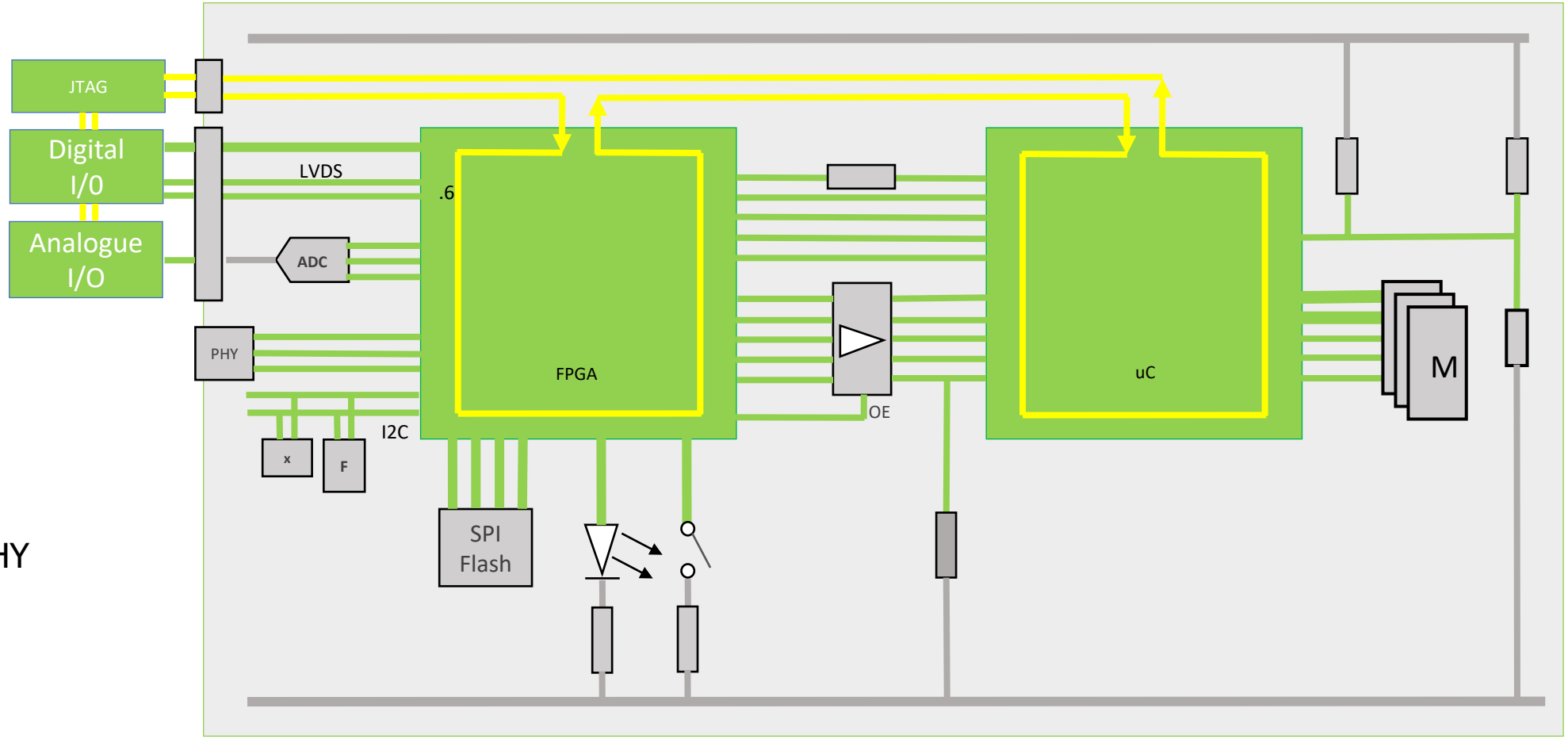
- Chain
- Interconnect
- LEDs
- Switches
- Connectors
- Resistors presence
- Memory connections
- LVDS
- Presence I²C, SPI, PHY



Serial bus tests (I²C, SPI ect.)

Basics - Test Applications

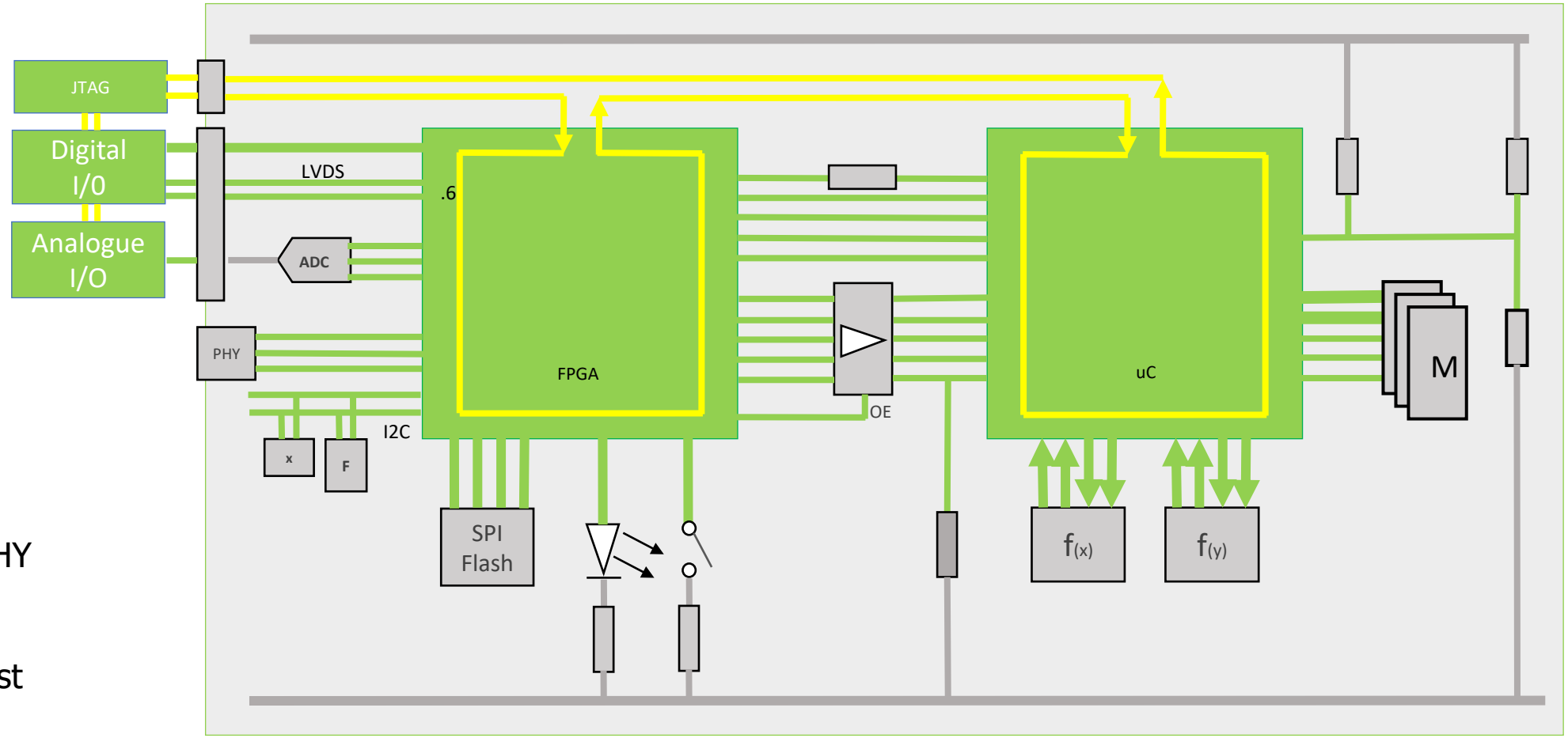
- Chain
- Interconnect
- LEDs
- Switches
- Connectors
- Resistors presence
- Memory connections
- LVDS
- Presence I²C, SPI, PHY
- Voltage, Freq, PWM



Increase Test coverage with external analogue driving and sense capabilities (ADC/DAC, PWM ect.)

Basics - Test Applications

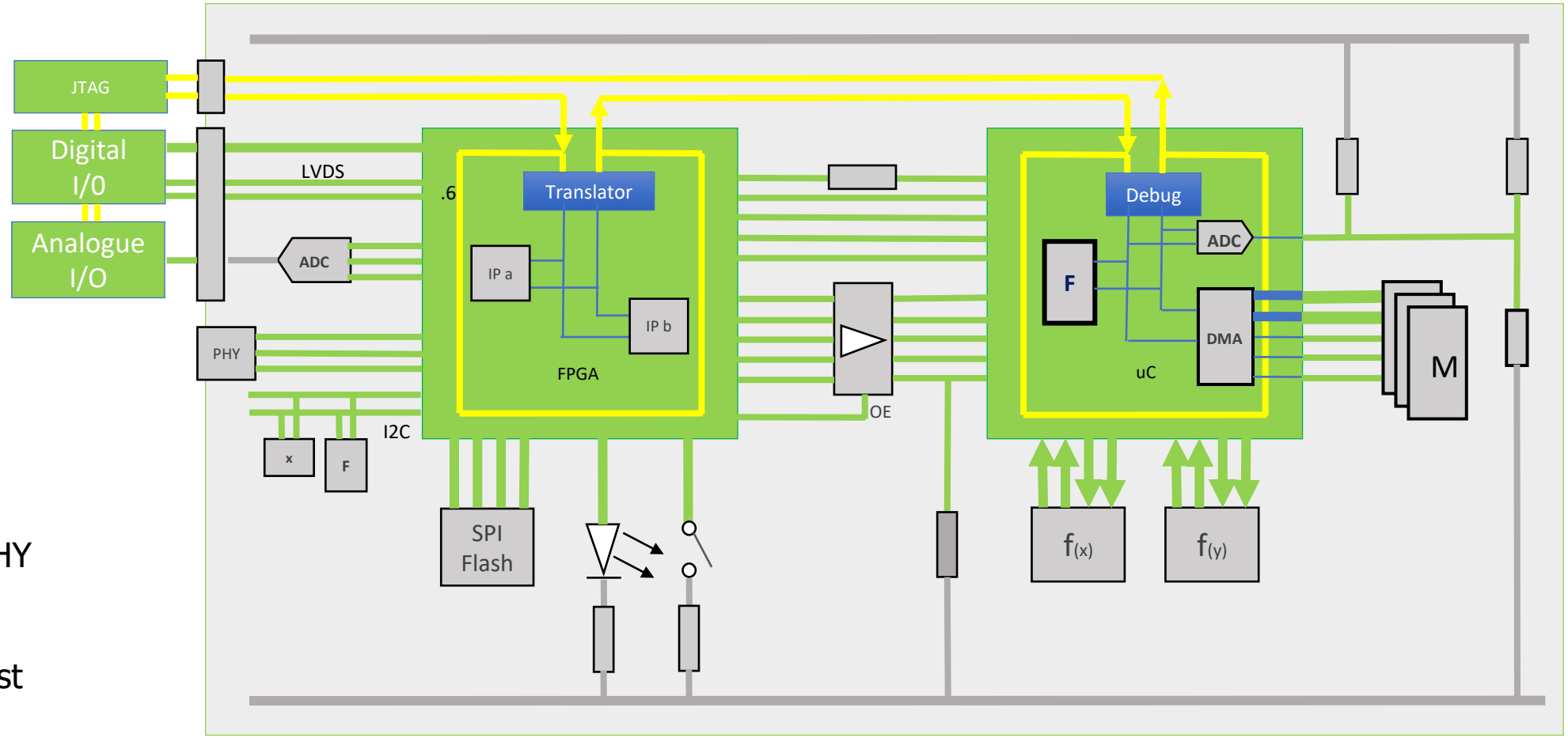
- Chain
- Interconnect
- LEDs
- Switches
- Connectors
- Resistors presence
- Memory connections
- LVDS
- Presence I²C, SPI, PHY
- Voltage, Freq, PWM
- Python Functional Test



Testing functions with Python based scripts, using Bscan accessible nodes as variable

Basics - Test Applications

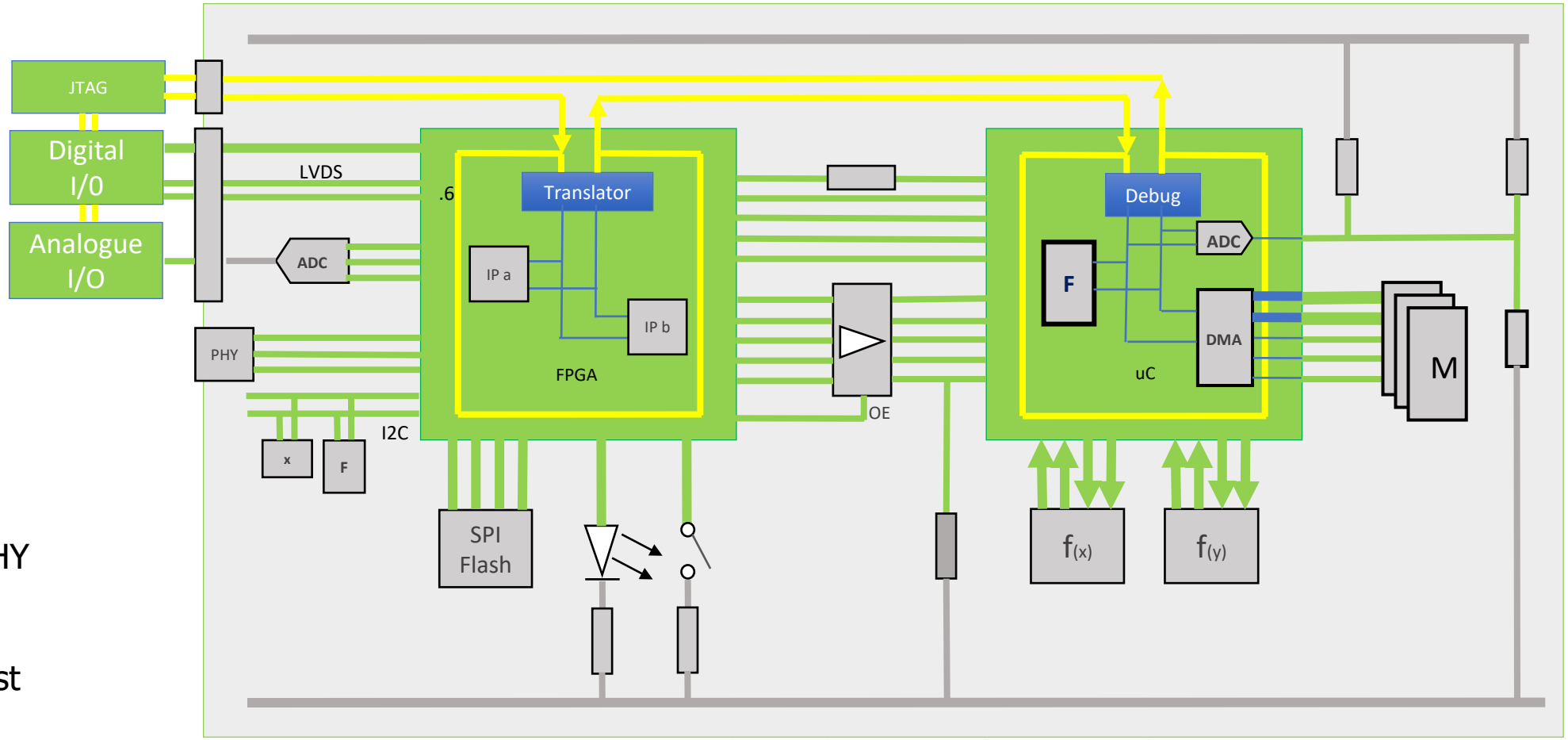
- Chain
- Interconnect
- LEDs
- Switches
- Connectors
- Resistors presence
- Memory connections
- LVDS
- Presence I²C, SPI, PHY
- Voltage, Freq, PWM
- Python Functional Test
- Emulative



Emulative Test for firmware independent @speed Test

Basics - Test Applications

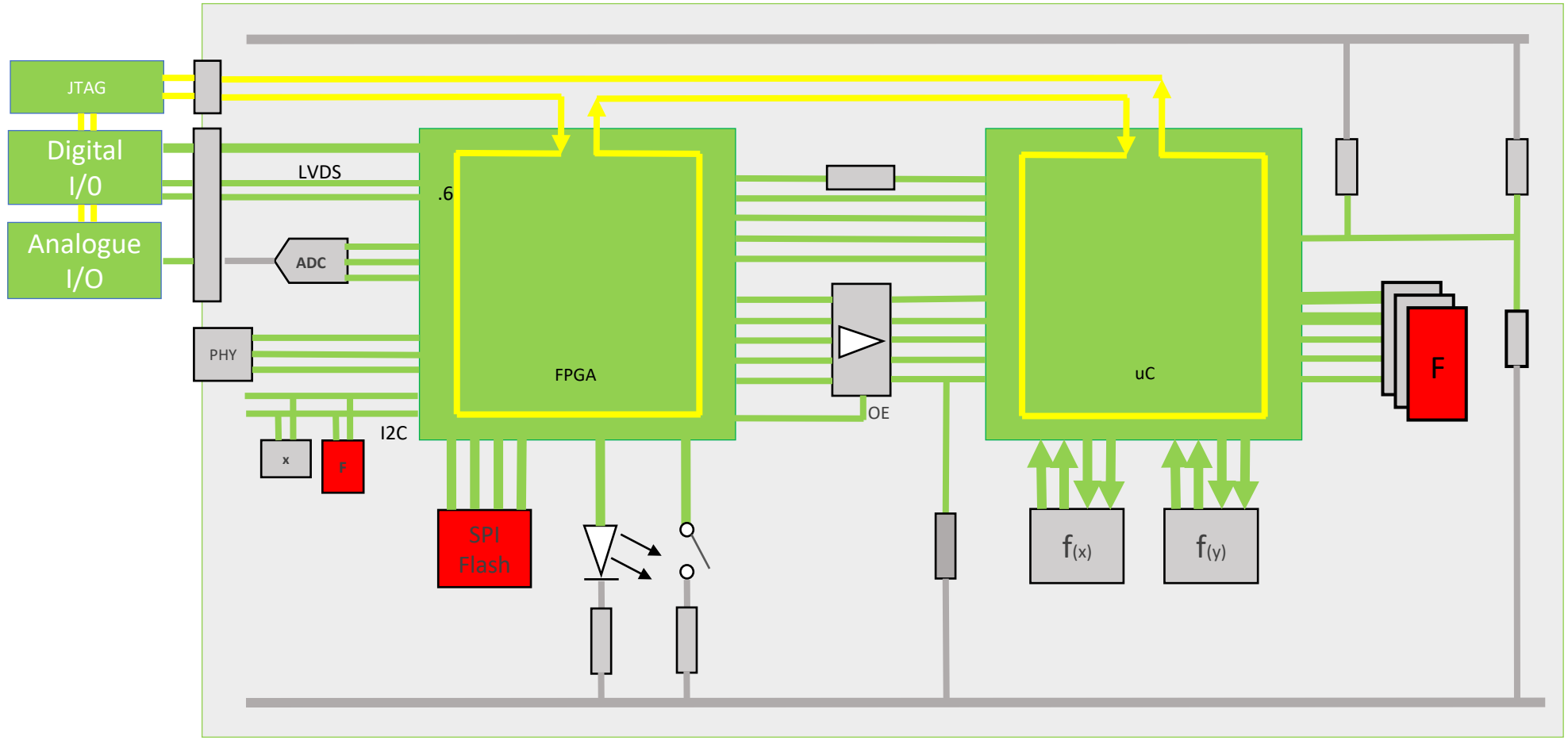
- Chain
- Interconnect
- LEDs
- Switches
- Connectors
- Resistors presence
- Memory connections
- LVDS
- Presence I²C, SPI, PHY
- Voltage, Freq, PWM
- Python Functional Test
- Emulative



ARM Cortex ANALOG DEVICES infineon intel MICROCHIP freescale TEXAS INSTRUMENTS Actel XILINX

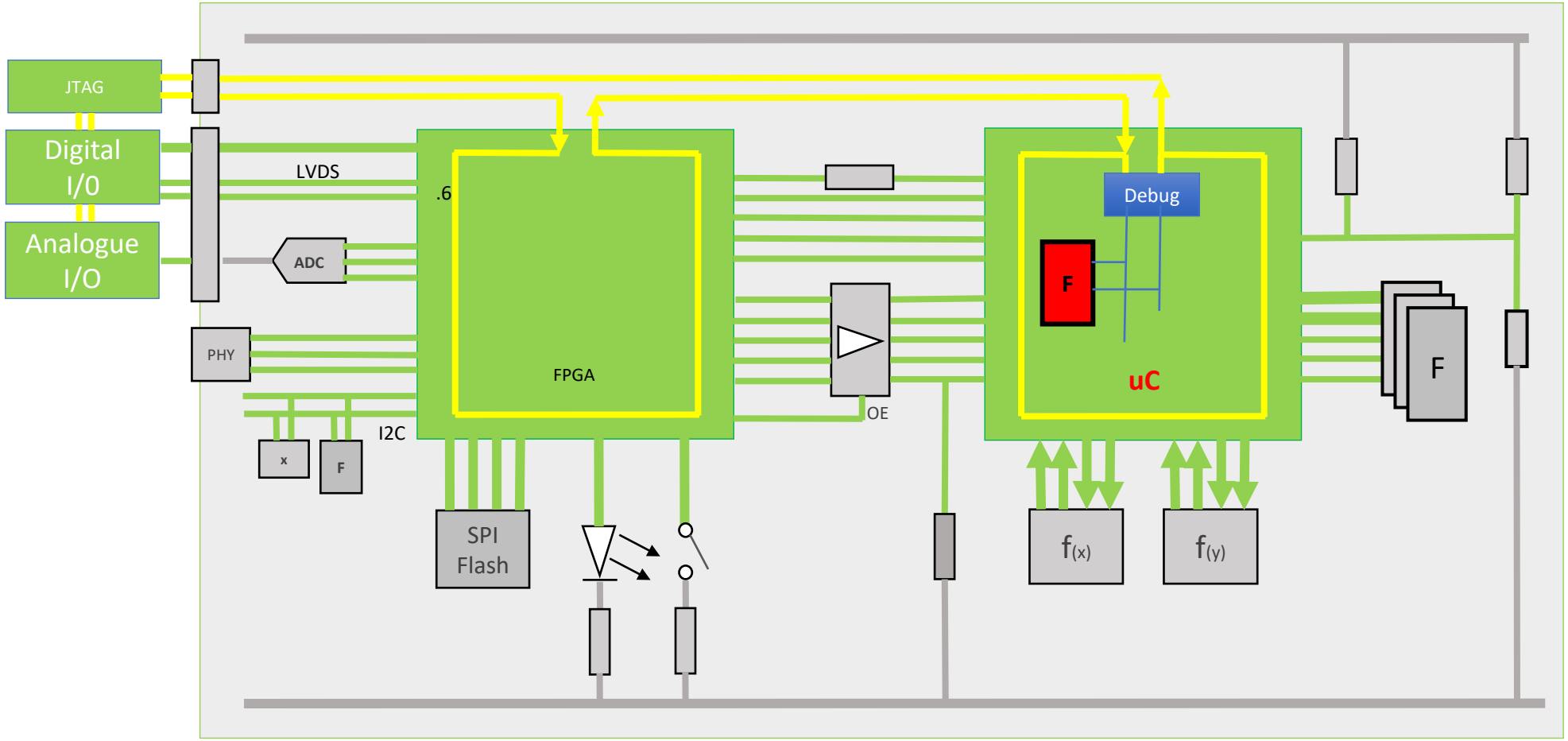
In-System Programming – via Bscan register

- Flash
- I²C Flash
- SPI Flash



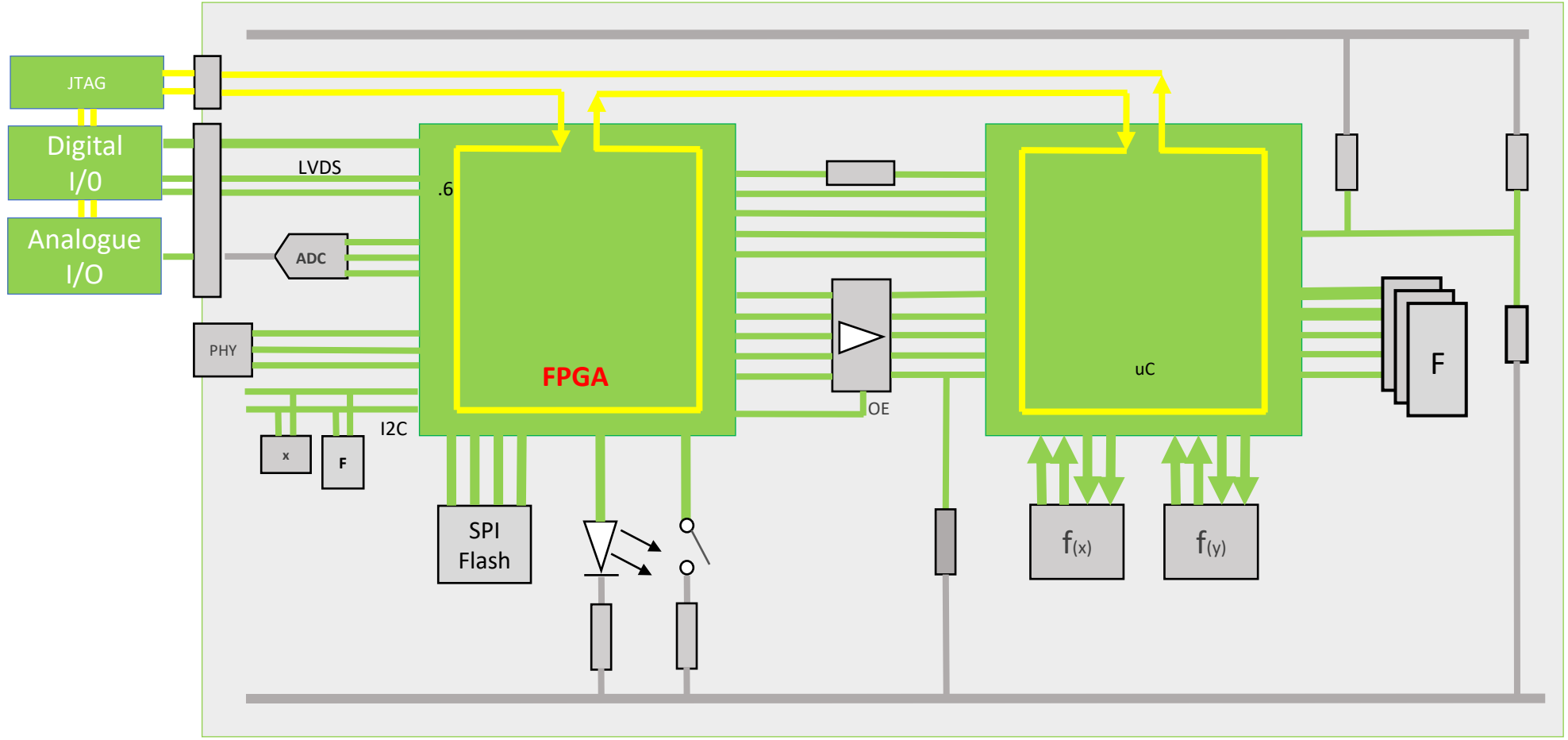
In-System Programming – micro controllers

- Internal uC Flash



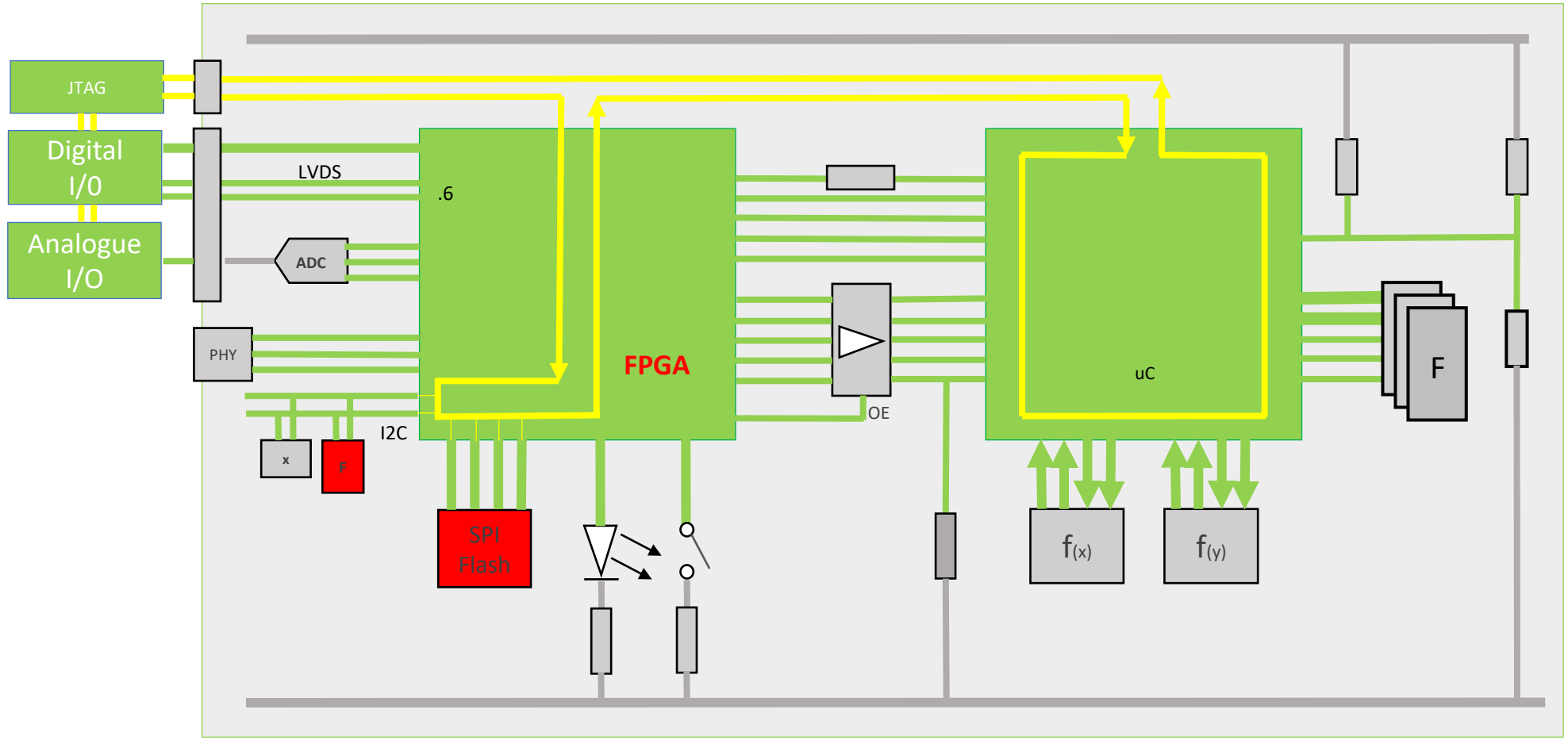
In-System Programming – logic devices

- SVF
- JAM
- STAPL
- Jedec
- IEEE 1532



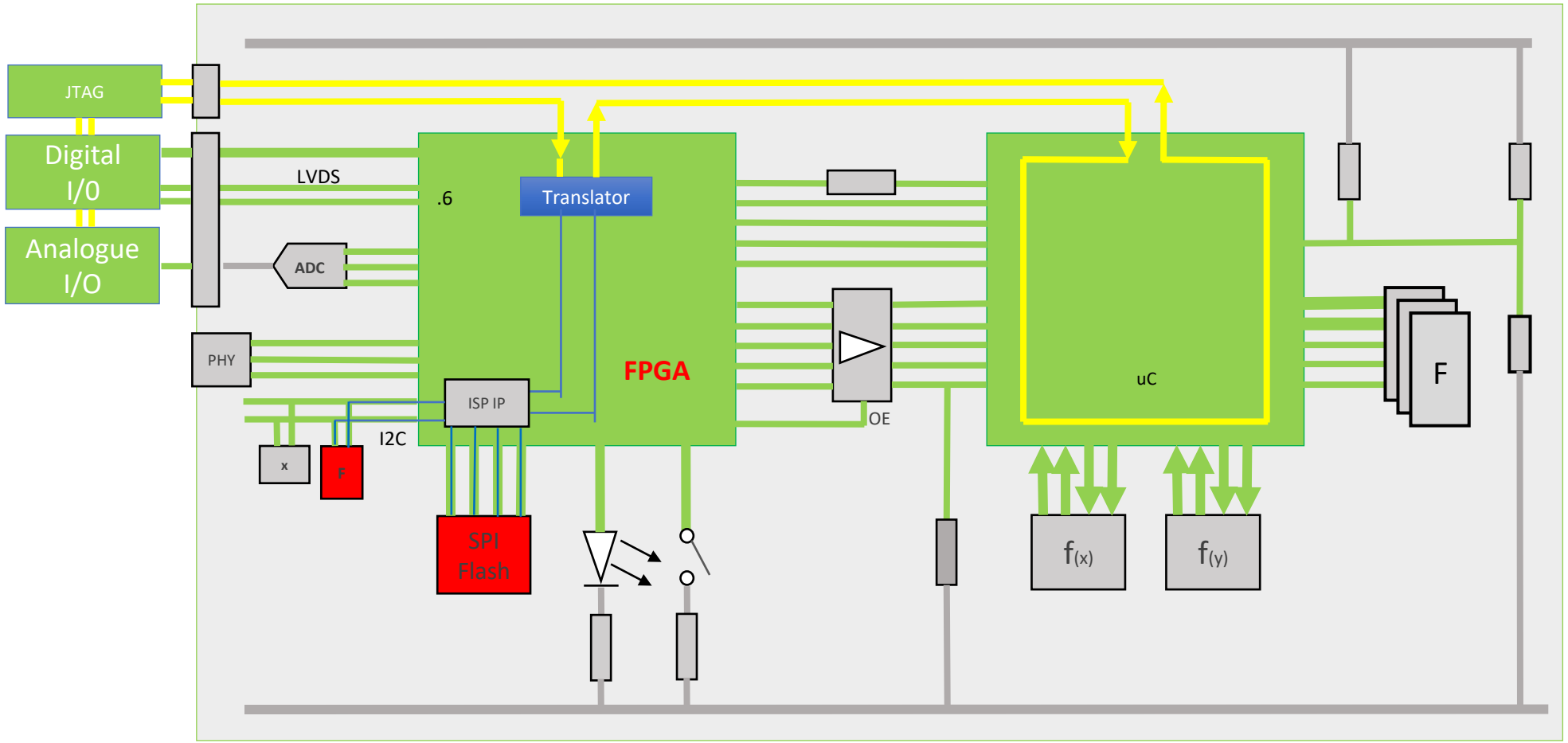
In-System Programming – via short chain

- Flash
- I²C Flash
- SPI Flash



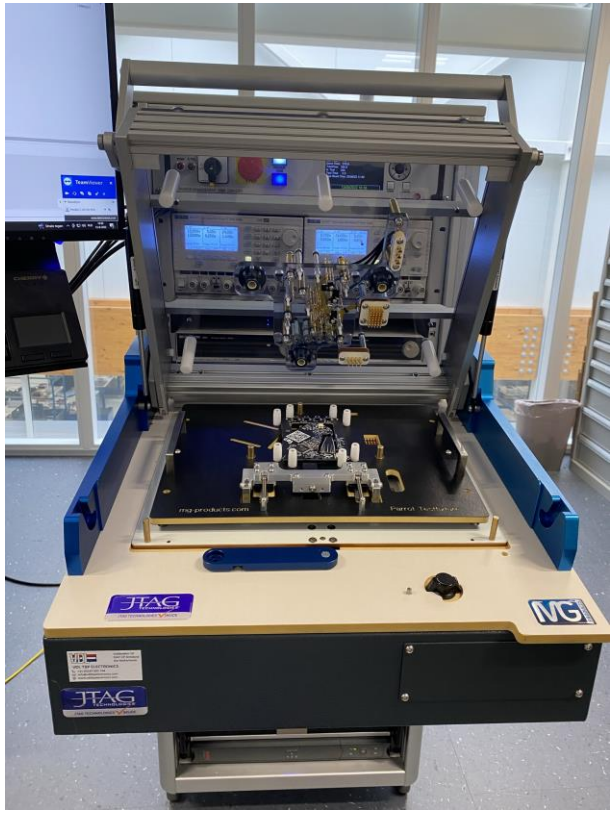
In-System Programming – via embedded programmer

- I²C Flash
- SPI Flash
- Other…



Benefits of Boundary scan

- **Affordable**
- **High coverage**
- **No firmware required**
- **No or minimal number of test pads** (test fixtures with less connections, less EMC issues)
- **Can be used during all product stages** (prototype, pre-production, production, field repairs)
- **Typically, JTAG interface is already available in your digital design(s)**
- **Testability and programmability can be determined at schematic stage**



Stand 9D079


We are boundary-scan.®
JTAG Technologies B.V.
Headquarter
Boschdijk 50
5612 AN Eindhoven
+31 40 2950870
info@jtag.com
www.jtag.com
www.jtaglive.com
Customers in 50+ countries
Worldwide support
2,500+ customers
25+ years in the heart of electronics
10,000+ systems sold