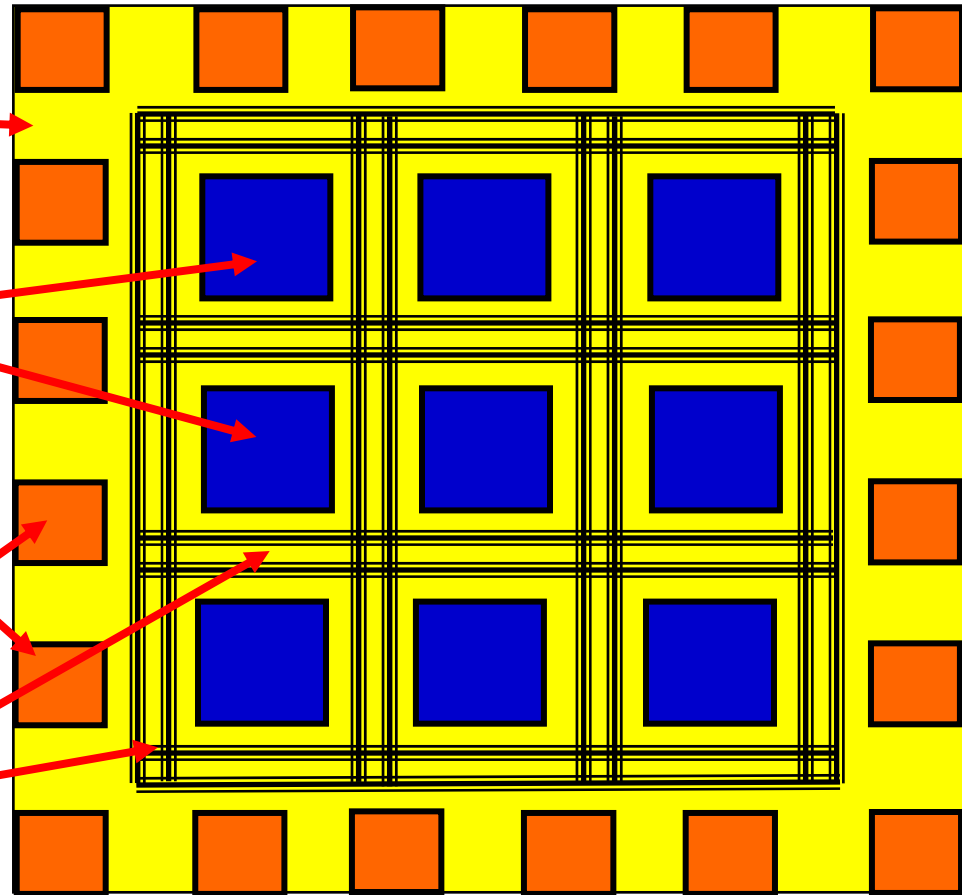


# Field Programmable Gate Arrays

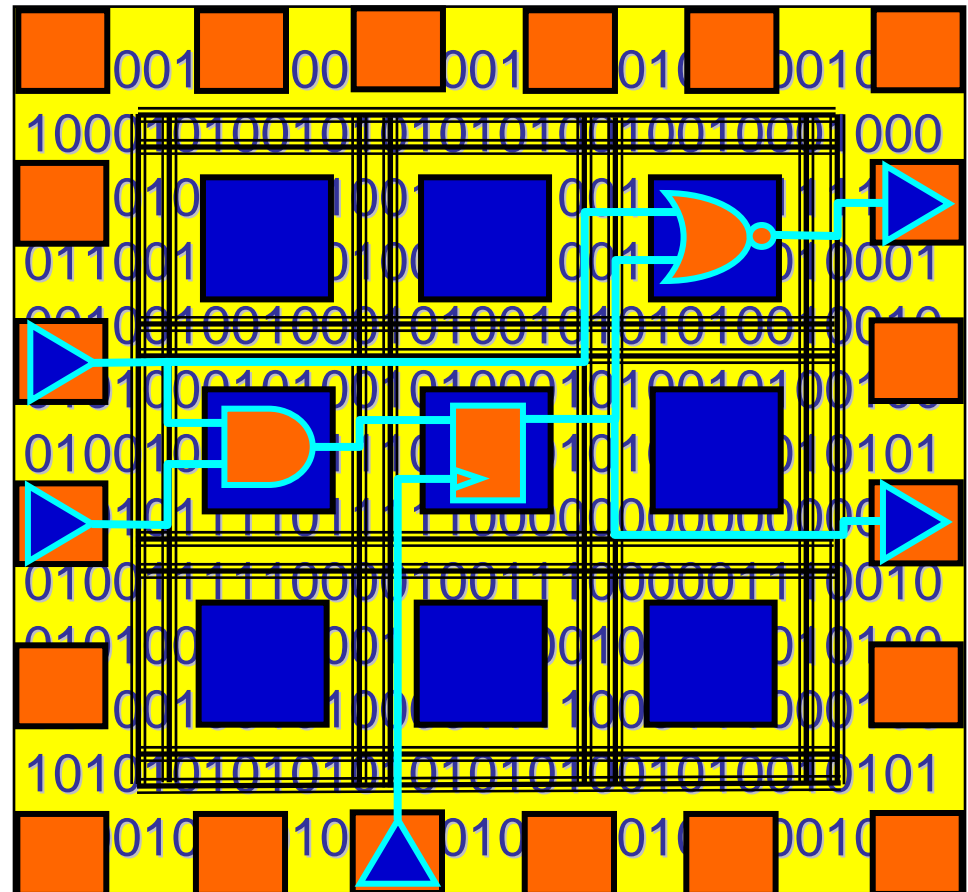
- ❑ Configuration Memory
- ❑ Programmable Logic Blocks (PLBs)
- ❑ Programmable Input/Output Cells
- ❑ Programmable Interconnect



Typical Complexity = 5 million – 1 billion transistors

# Basic FPGA Operation

- Writing configuration memory  $\Rightarrow$  defines system function
  - ❖ Input/Output Cells
  - ❖ Logic in PLBs
  - ❖ Connections between PLBs & I/O cells
- Changing configuration memory data  $\Rightarrow$  changes system function
  - ❖ Can change at anytime
  - ❖ Even while system function is in operation



# Combinational Logic Functions

□ Gates are combined to create complex circuits

□ Multiplexer example

❖ If  $S = 0$ ,  $Z = A$

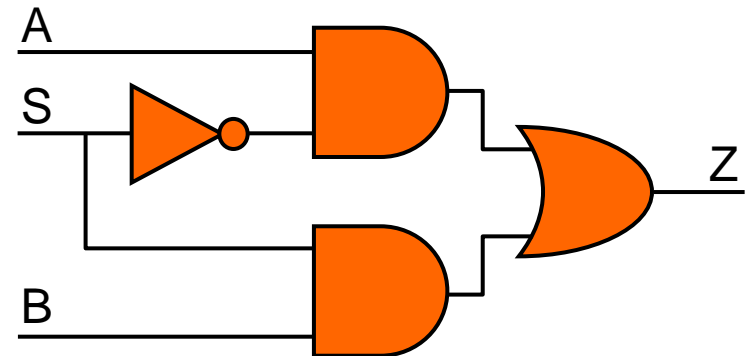
❖ If  $S = 1$ ,  $Z = B$

❖ Very common digital circuit

❖ Heavily used in FPGAs

✓ S input controlled by configuration memory bit

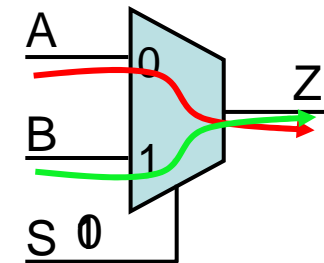
✓ We'll see it again



Truth table

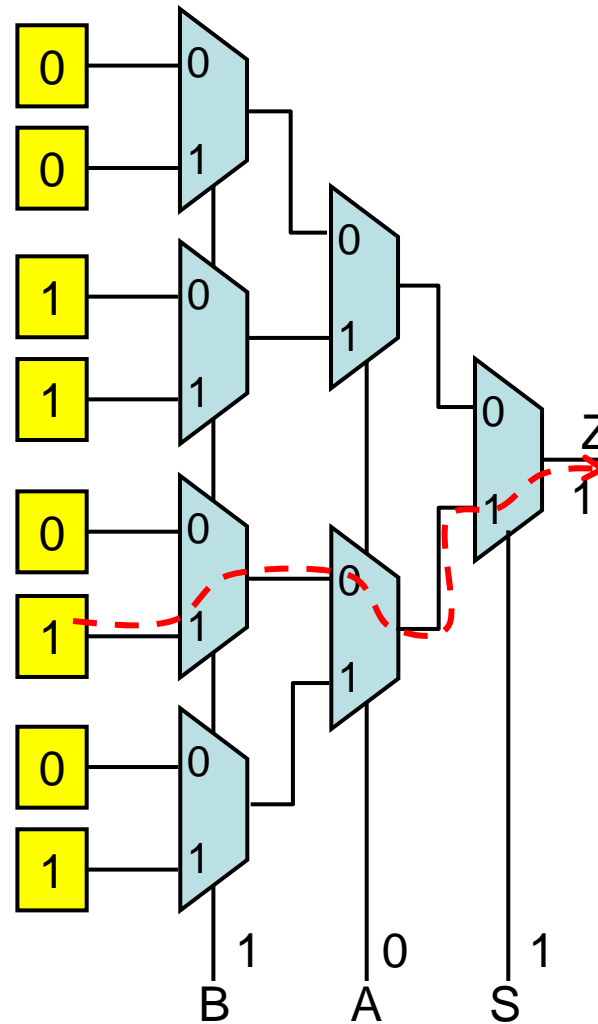
S	A	B	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Logic symbol

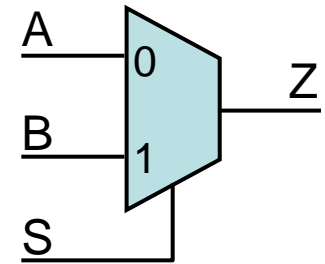


# Look-up Tables

- ❑ Recall multiplexer example
- ❑ Configuration memory holds outputs for truth table
- ❑ Internal signals connect to control signals of multiplexers to select value of truth table for any given input value



Multiplexer

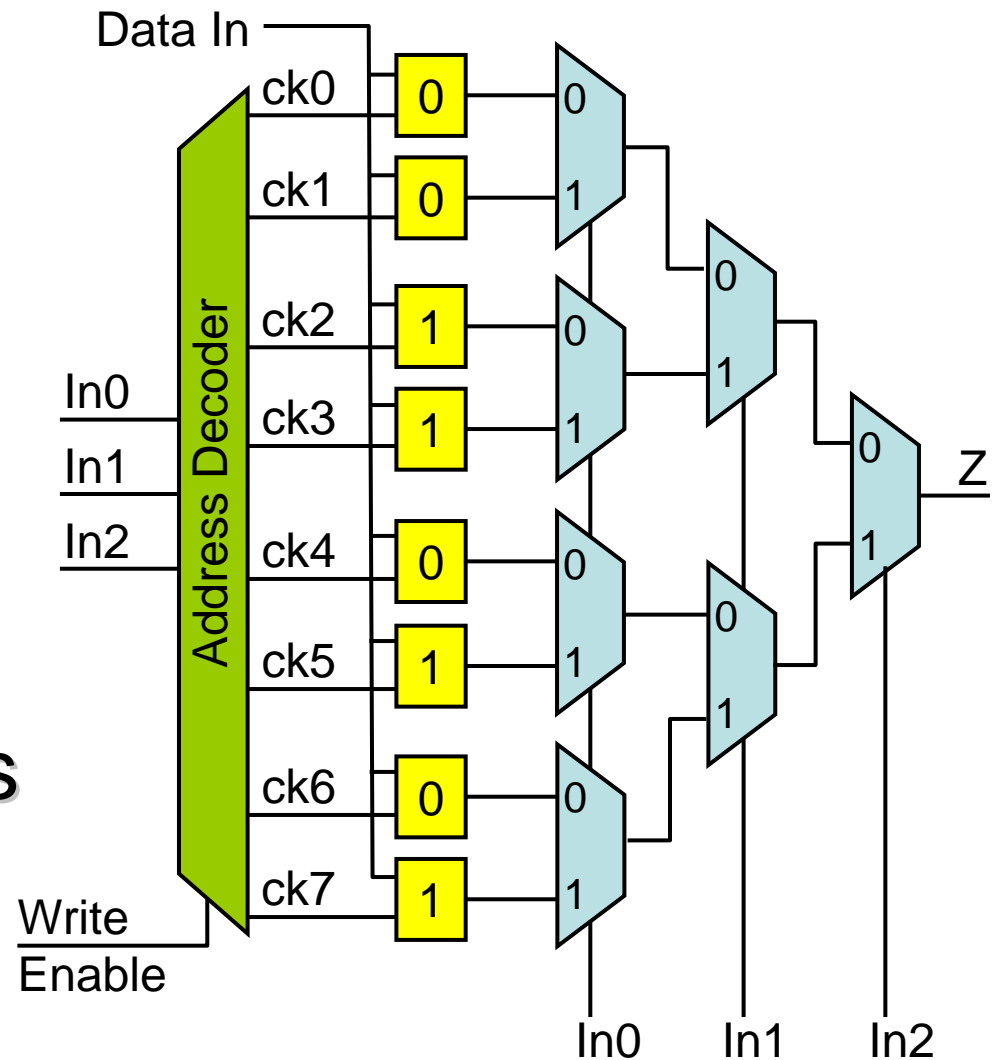


Truth table

S	A	B	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

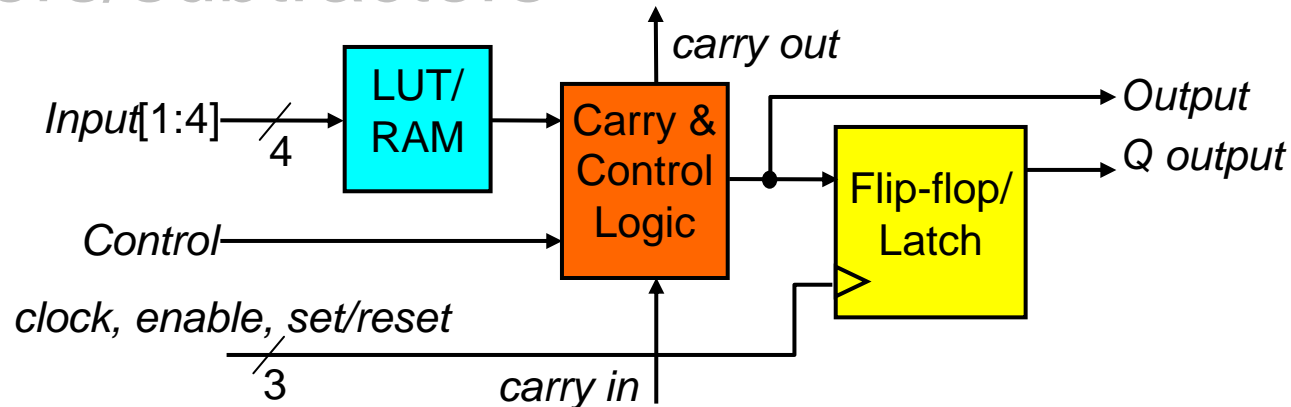
# Look-up Table Based RAMs

- ❑ Normal LUT mode performs read operations
- ❑ Address decoder with write enable generates clock signals to latches for write operations
- ❑ Small RAMs but can be combined for larger RAMs



# Basic PLB Architecture

- ❑ Look-up Table (LUT) implements truth table
- ❑ Memory elements:
  - ❖ Flip-flop/latch
  - ❖ Some FPGAs - LUTs can also implement small RAMs
- ❑ Carry & control logic implements fast adders/subtractors



# A Simple PLB

## Two 3-input LUTs

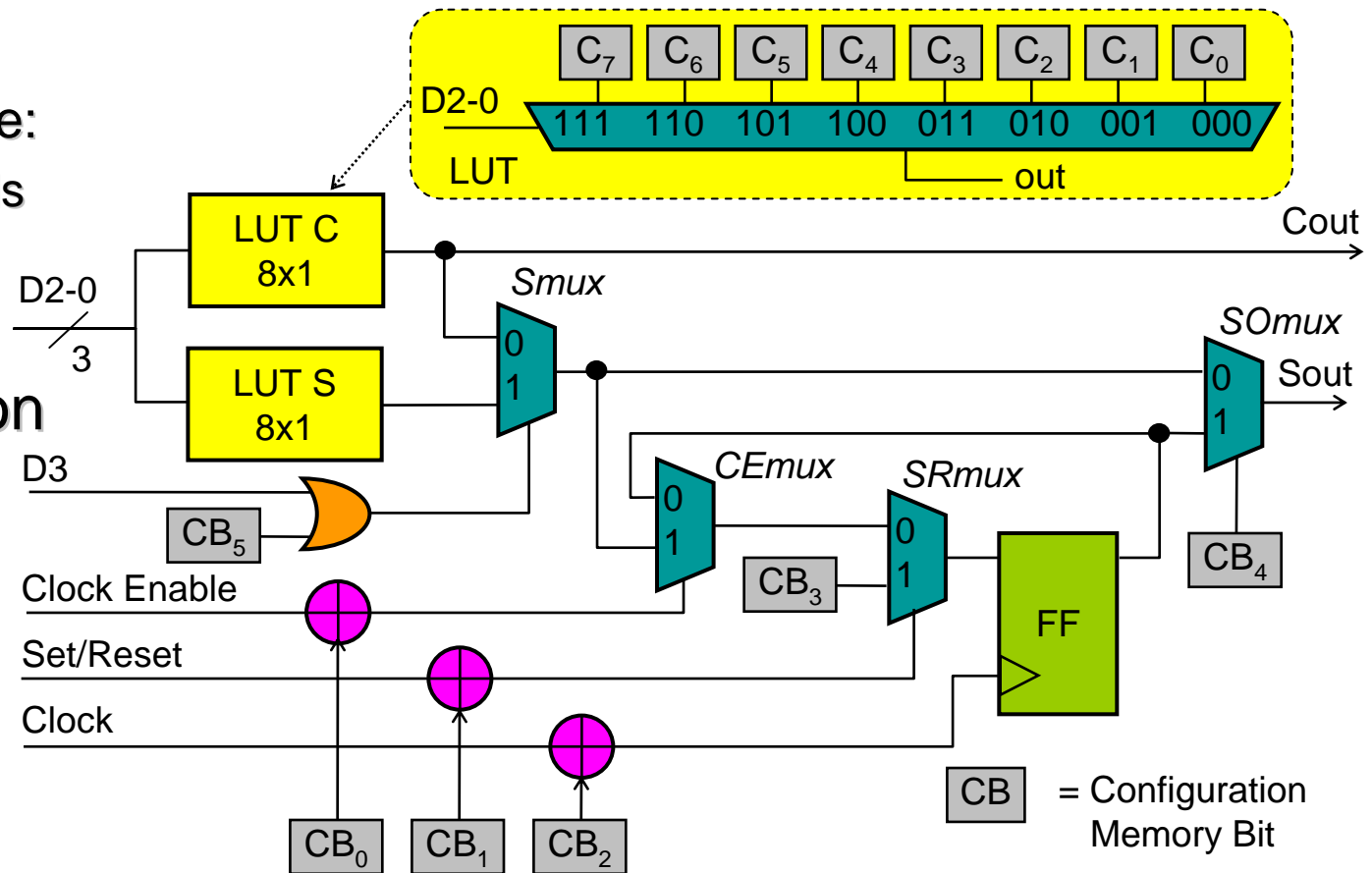
- Can implement any 4-input combinational logic function

## 1 flip-flop

- Programmable:
  - Active levels
  - Clock edge
  - Set/reset

## 22 configuration memory bits

- 8 per LUT
  - C0-7
  - S0-7
- 6 control bits
  - CB0-5



# Xilinx FPGAs

## ❑ Virtex and Spartan 2

- ❖ Array of 96 to 6,144 PLBs
  - ✓ 4 LUTs/RAMs (4-input)
  - ✓ 4 FF/latches
- ❖ 4 to 32 4K-bit dual-port RAMs

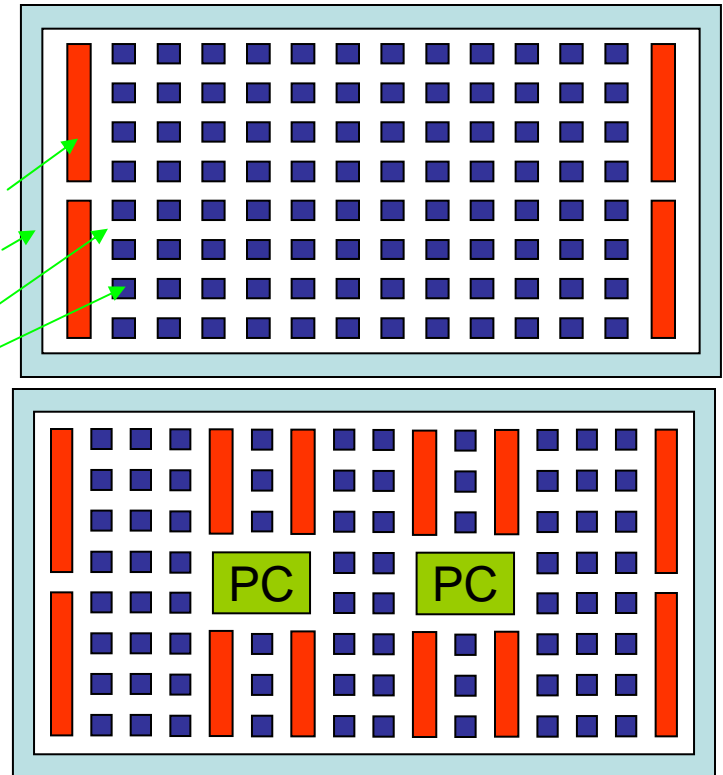
## ❑ Virtex II, Virtex II Pro

- ❖ Array of 352 to 11,204 PLBs
  - ✓ 8 LUTs/RAMs (4-input)
  - ✓ 8 FF/latches
- ❖ 12 to 444 18K-bit dual-port RAMs
- ❖ 12 to 444 18×18-bit multipliers
- ❖ 0 to 2 PowerPC processor cores

## ❑ Virtex 4

- ❖ Array of 1,536 to 22,272 PLBs
  - ✓ 4 LUTs/RAMs (4-input)
  - ✓ 4 LUTs (4-input)
  - ✓ 8 FF/latches
- ❖ 48 to 552 18K-bit dual-port RAMs
  - ✓ Also operate as FIFOs
- ❖ 32 to 512 DSP cores include:
- ❖ 0 to 2 PowerPC processor cores

Special cores  
I/O cells  
Routing  
PLBs



## ❑ Spartan 3

- ❖ Array of 192 to 8,320 PLBs
  - ✓ 4 LUTs/RAMs (4-input)
  - ✓ 4 LUTs (4-input)
  - ✓ 8 FF/latches
- ❖ 4 to 104 18K-bit dual-port RAMs
- ❖ 4 to 104 18×18-bit multipliers

# Ranges of FPGA Resources

FPGA Resource		Small FPGA	Large FPGA
Logic	PLBs per FPGA	256	25,920
	LUTs and flip-flops per PLB	1	8
Routing	Wire segments per PLB	45	406
	PIPs per PLB	139	3,462
Specialized Cores	Bits per memory core	128	36,864
	Memory cores per FPGA	16	576
	DSP cores	0	512
Other	Input/output cells	62	1,200
	Configuration memory bits	42,104	79,704,832

# What is Built-In Self-Test?

□ **Basic idea:** Add circuitry to IC or PCB to facilitate testing itself

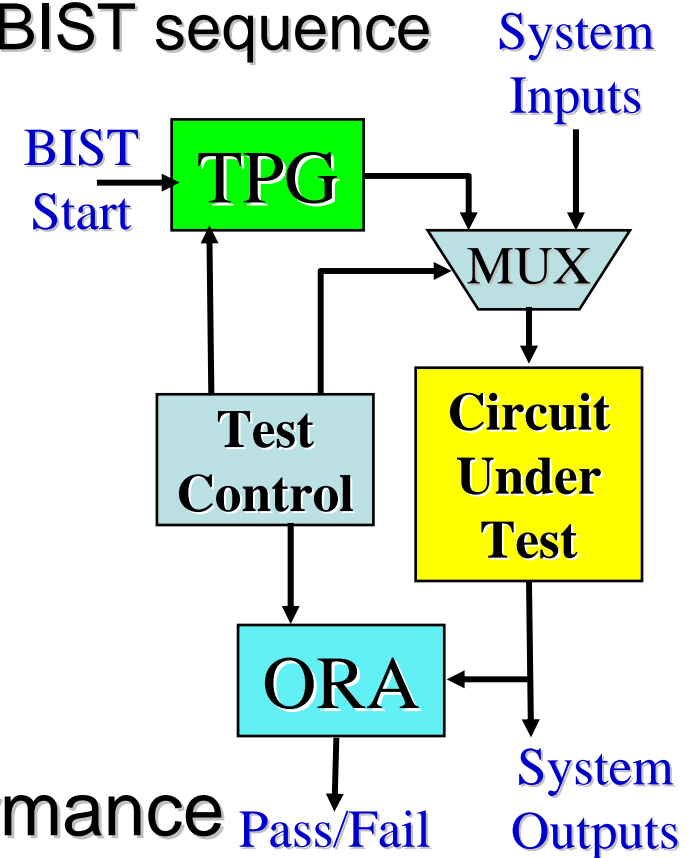
- ❖ Only power and clock needed during BIST sequence
- ❖ Pass/Fail result reported at end of BIST sequence
  - ✓ No need for external test equipment

□ Necessary components:

- ❖ Test Pattern Generator (TPG)
- ❖ Output Response Analyzer (ORA)
- ❖ For system level use:
  - ✓ Test controller
  - ✓ Input isolation

□ **Benefits:** low testing time & cost

□ **Penalties:** area overhead, performance

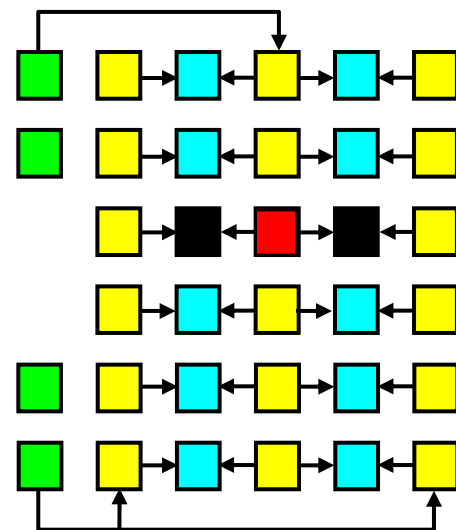
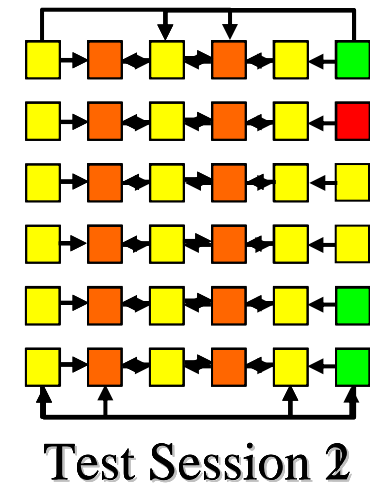


# BIST for FPGAs

- ❑ **Basic idea:** reprogram FPGA to test itself
- ❑ BIST logic disappears after test
  - ❖ No area overhead or performance penalties
- ❑ Applicable to all levels of testing
  - ❖ A generic test for a generic component
  - ❖ Independent of system function
- ❑ Good diagnostic resolution
  - ❖ **Logic:** Look-Up Table (LUT) or flip-flop
  - ❖ **Routing:** wire segment or switch
  - ❖ Reconfigure system function for fault-tolerance
- ❑ **Cost:** memory to store BIST configurations

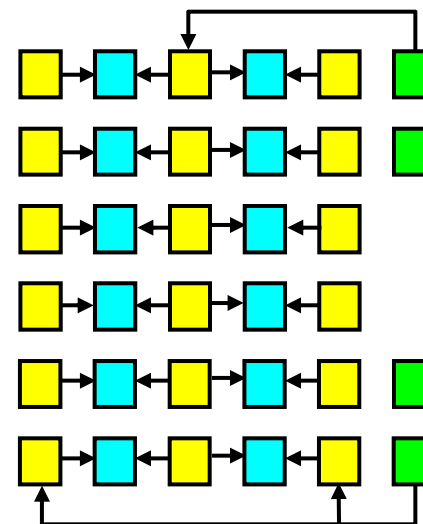
# Built-In Self-Test of FPGA Logic

- Program PLBs to functions as
  - ❖ Test Pattern Generators (TPGs)
    - ✓ 6 to 12-bit counter
  - ❖ Output Response Analyzers (ORAs)
    - ✓ Comparator and a latch
  - ❖ Blocks Under Test (BUTs)
- Flip BIST architecture to complete testing



Test Session 1

■ = TPG  
■ = BUT  
■ = ORA



Test Session 2

# Fault Injection Emulator

□ Faulty FPGA are difficult to find

❖ 1 FPGA with faulty PLB & 2 FPGAs with faulty routing

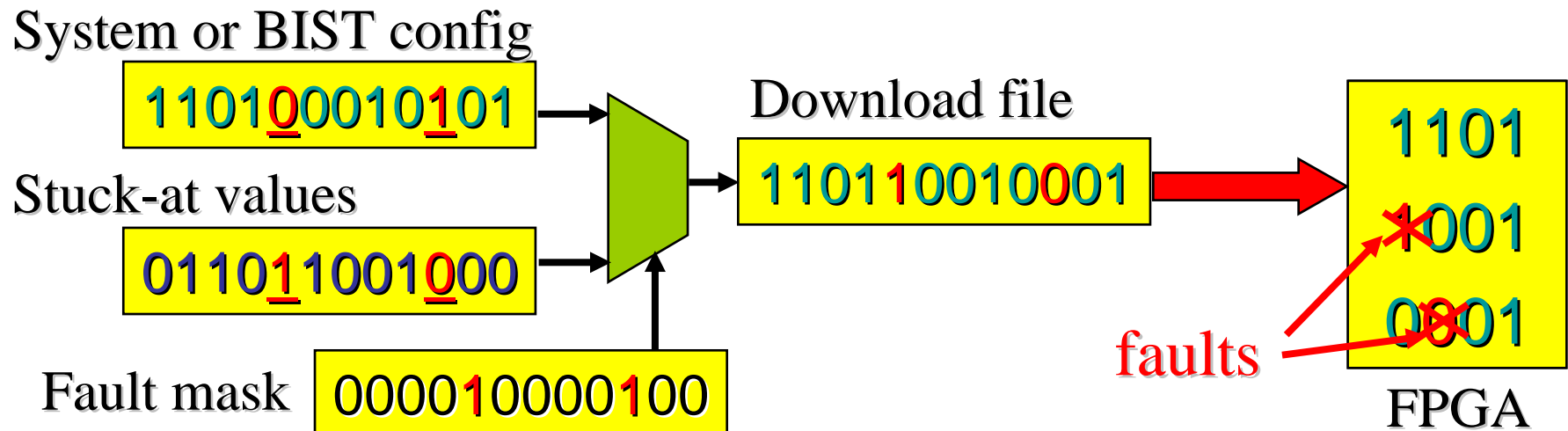
□ We created a Fault Injection Emulator

❖ Intercepts & modifies configuration bits prior to download

❖ Fault Emulator can create multiple faults in:

✓ PLBs: LUTs, flip-flops, etc.

✓ Interconnect: shorts and opens in wires



# FPGA BIST Demonstration

## □ Graphic User Interface

- ❖ Shows what is happening inside FPGA during BIST

## □ Fault Injection Emulator

- ❖ Inserts faults into configuration data file
- ❖ Emulated faults are downloaded with BIST configuration

## □ Diagnostic algorithm applied to BIST results

- ❖ Identifies faulty resource

