

Programmable Logic Devices

- Read Only Memory (W 800-821)
- Programmable Logic Arrays (W 370-373)
- Programmable Array Logic (W 373-378)
- Field Programmable Gate Arrays (W 850-858)



Implementation Options for Digital Logic

- Assembly of SSI and MSI parts on PC boards.
 - » mostly obsolete; still useful when just a few parts needed
- Programmable Logic Devices (PLD)
 - » variety of types, with different size and performance characteristics; largest have over 10^6 gate “equivalents”
 - » CAD tools enable simulation and automate device programming
- Application Specific Integrated Circuits (ASIC)
 - » design methods similar to PLDs
 - HDLs and simulation with synthesis using standard cell library
 - plus physical design - placement of logic components, routing
 - » can augment with custom design of critical components
 - » higher performance, greater logic density
 - » custom IC fabrication – suitable for high production volumes

Programmable Logic Devices

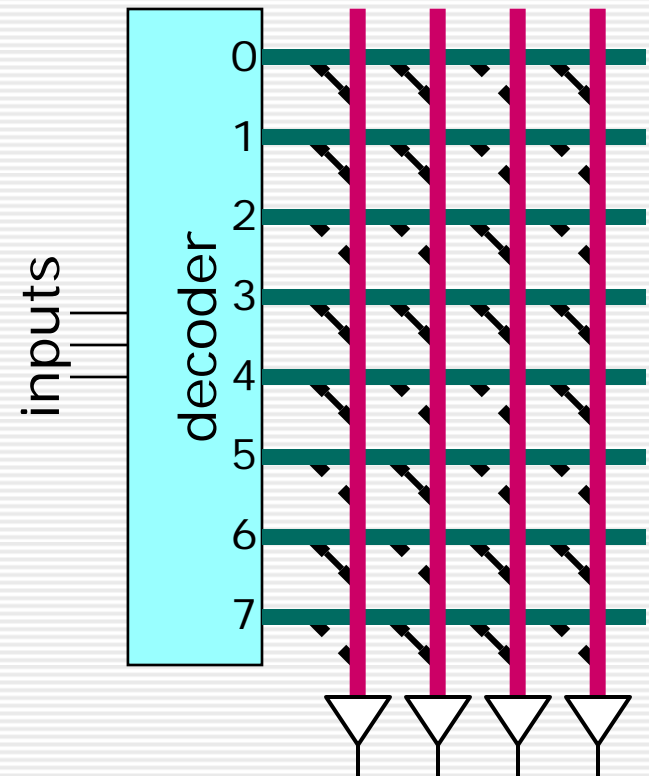
- Simple logic arrays
 - » implement 2 level logic circuits (AND/OR)
 - » based on regular array structure
 - » several types
 - Read Only Memories (ROMs and PROMs)
 - Programmable Logic Array (PLA)
 - Programmable Array Logic (PAL)
- Field Programmable Gate Arrays (FPGA)
 - » many copies of common building block
 - » each block can be configured for different logic functions and typically includes a flip flop and a 4 input function generator
 - » programmable interconnect
 - » often includes SRAM blocks
 - » largest FPGAs have about 100K flip flops, 100K function generators and 10 Mb of SRAM

Read-Only Memory

- Any random access memory can implement logic.
 - » store the logic function's truth table in memory
 - » example: use 4 bit RAM to get 2 input AND by storing 0 at locations 00, 01, 10 and 1 at location 11
 - » with 2^m words of 1 bit each can implement any logic function with m inputs
 - » memory with 2^m words of w bits each can implement w different logic functions of same m inputs
- *Read-Only Memory* (ROM) operates like a random access memory, but cannot be written to.
 - » stored data is determined when device is manufactured
 - » higher density and faster than comparable RAMs
 - » non-volatile storage - data retained when power is off
 - » programmable ROMs are written with PROM *programmer*

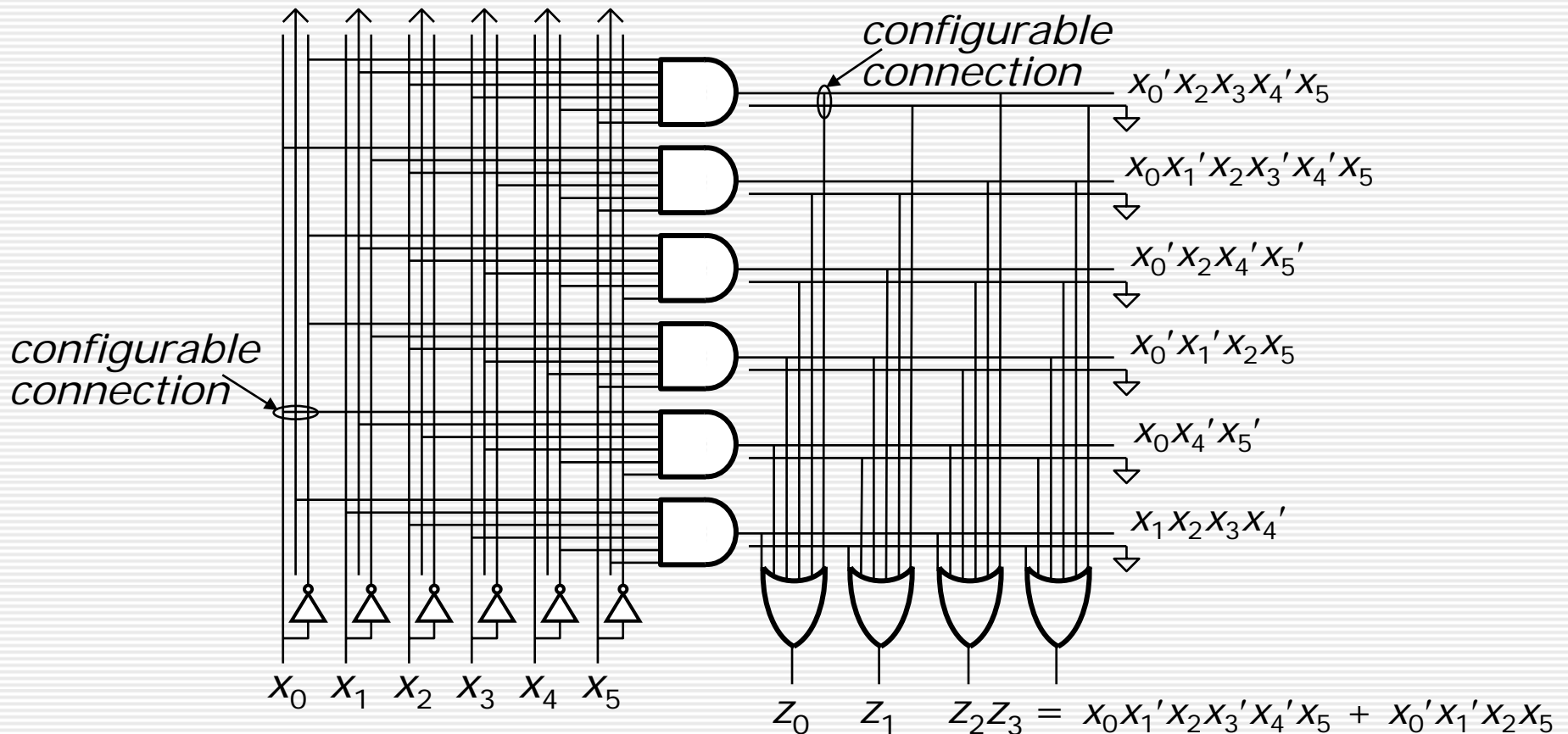
Implementation of ROMs

- ROM can be implemented using orthogonal arrangement of wires.
 - » optional connection at each intersection
 - » decoder puts logic '1' on exactly one of the horizontal wires - this can be detected at output if connection present
- Some PROMs are configured by breaking connections.
 - » high voltage placed across one input and one output at a time
 - » high current flow causes "fuse" at intersection to "blow"
- Other PROMs can be erasable and reprogrammed (EPROMs).



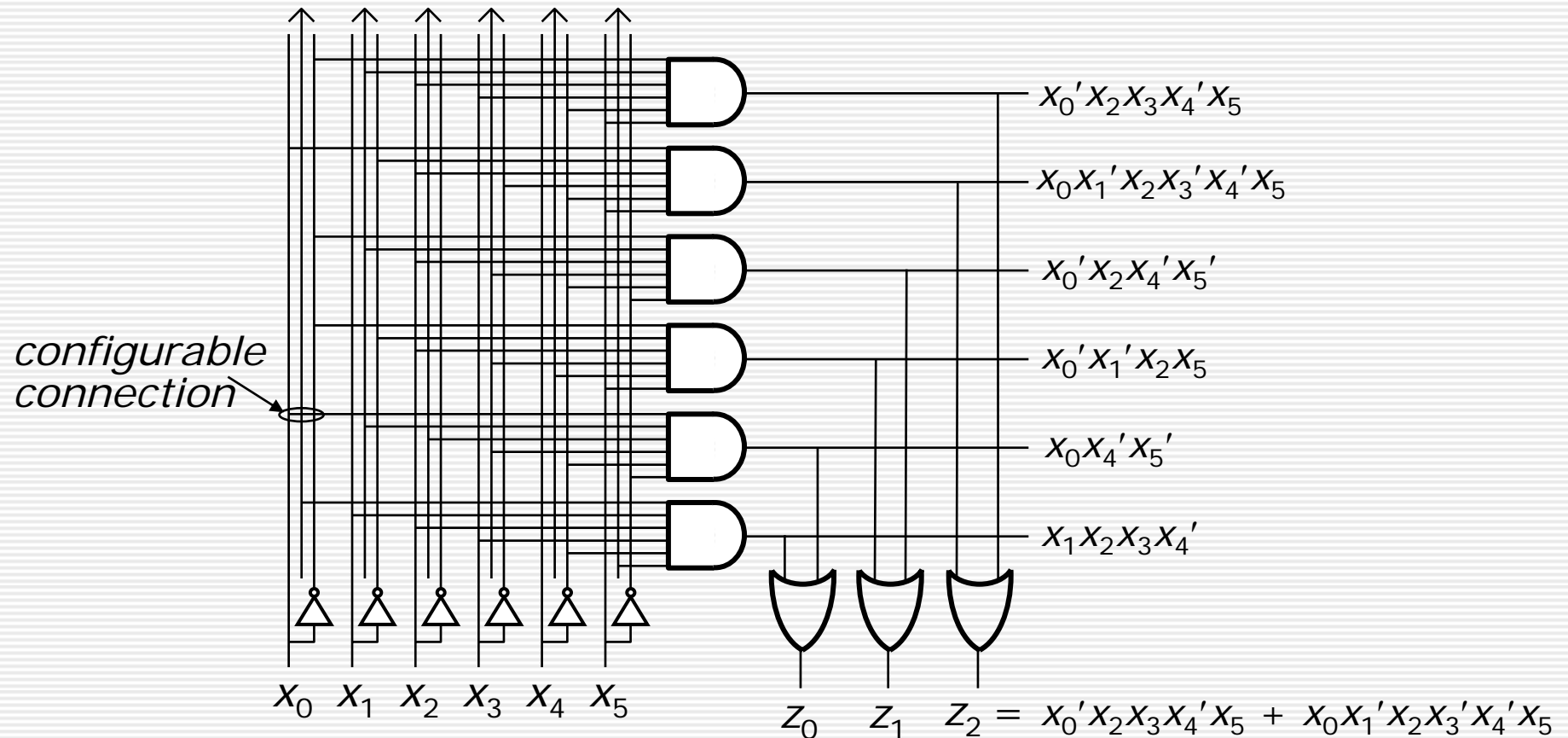
- stored functions
 - » $\sum_m(0,1,3,4,6)$,
 - » $\sum_m(0,1,3,5,7)$,
 - » $\sum_m(2,3,6,7)$,
 - » $\sum_m(0,3,4,6)$

Programmable Logic Arrays



- PLAs have configurable “AND-plane” & “OR-plane”.
- Can implement any 2-level AND-OR circuit.
- Efficient physical implementation in CMOS.

Programmable Array Logic



- PAL is similar to PLA but fixed OR-plane.
- Simpler to program and cheaper implementation.
- Limited number of terms in each output.

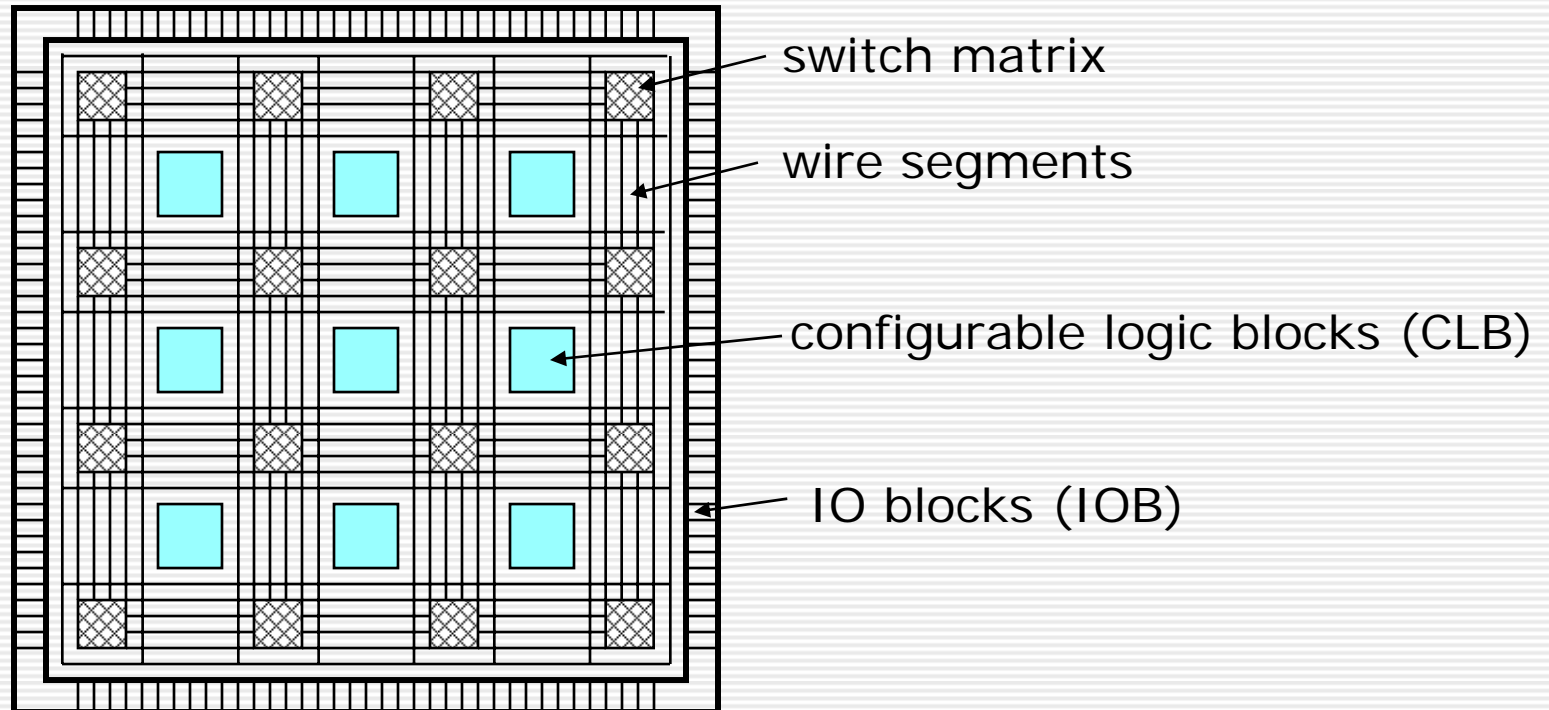
Comparison of PROMs, PLAs & PALs

- Can view PROMs, PALs as restricted forms of PLA.
 - » PROMs are logically equivalent to PLA with AND-plane that generates all minterms and configurable OR-plane
 - » PAL is logically equivalent to PLA with fixed OR-plane in which each output is the OR of a subset of the ANDs
- Different implementations means different capabilities.
 - » so, one needs to match device capabilities to the characteristics of logic equations being generated
 - » consider number inputs & outputs, total number of different terms (PLAs), number of different terms per output (PALs)
 - for given number of inputs/outputs, PROMs most flexible
 - » performance characteristics and cost also differ
- Most parts include flip flops, for sequential circuits.

Field Programmable Gate Arrays

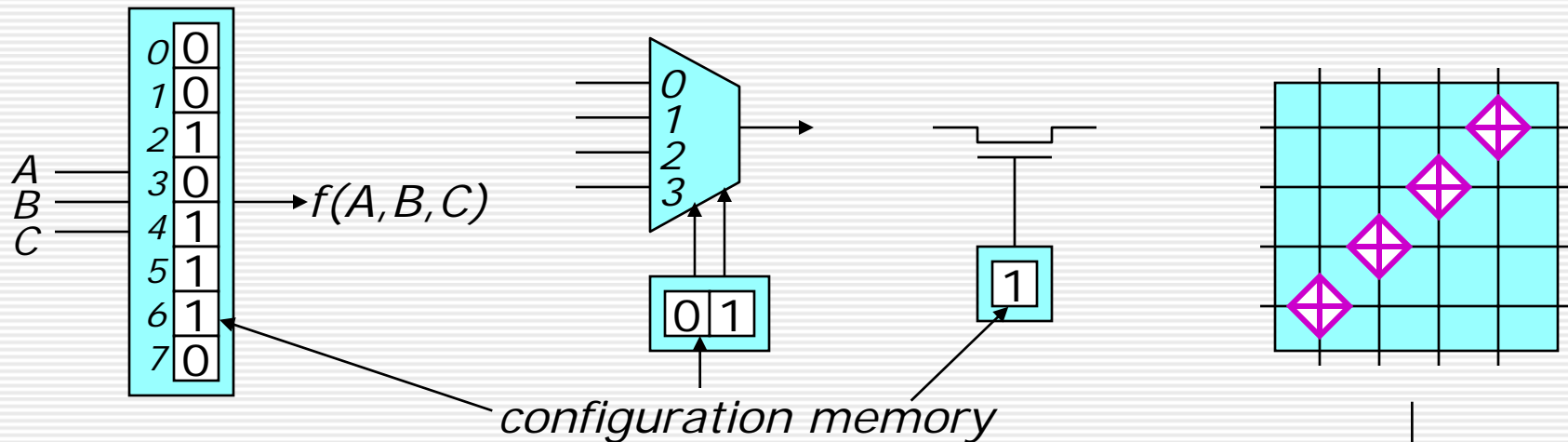
- FPGAs can be used to construct more complex circuits.
- Chip contains a large number (tens of thousands) of configurable logic building blocks.
 - » typically each block includes a 4 input function generator, a flip flop and some “glue” logic
 - » CAD tools map high level circuit to basic blocks, configuring function generators & other configurable elements as needed
- Programmable interconnect used to wire logic blocks.
 - » wire segments connected to logic blocks and to other wire segments by configurable switches
 - » CAD tools determine switch configuration needed to provide right connectivity
- CAD tools perform *mapping, placement, routing*.
 - » routing information used in timing analysis & simulation

Xilinx FPGA Organization



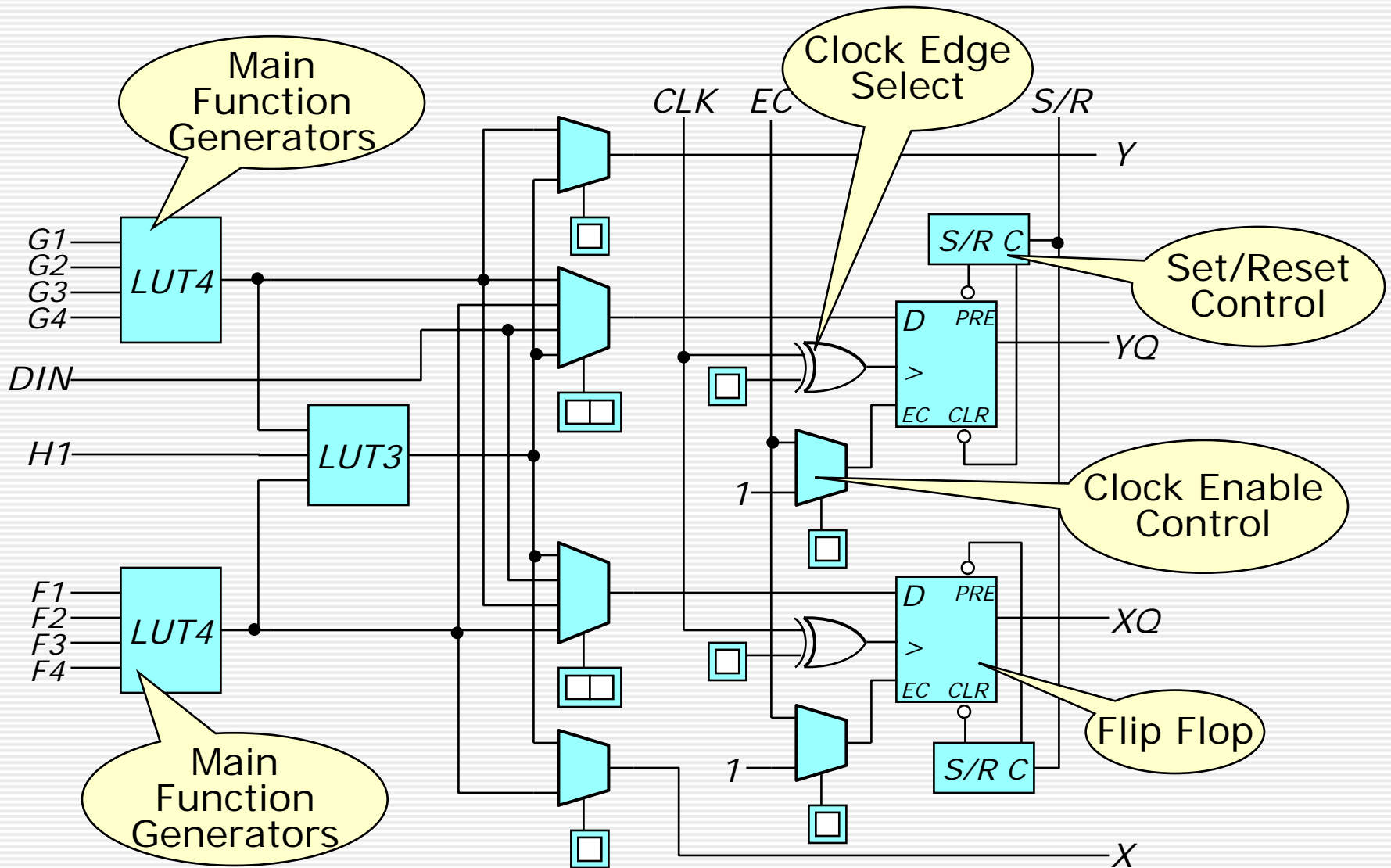
- CLBs can be connected to “passing” wires.
- Wire segments connected by switch matrix.
- Long wire segments used to connect distant CLBs.
- Configuration information stored in SRAM bits that are loaded when power turns on.

Configuring Logic

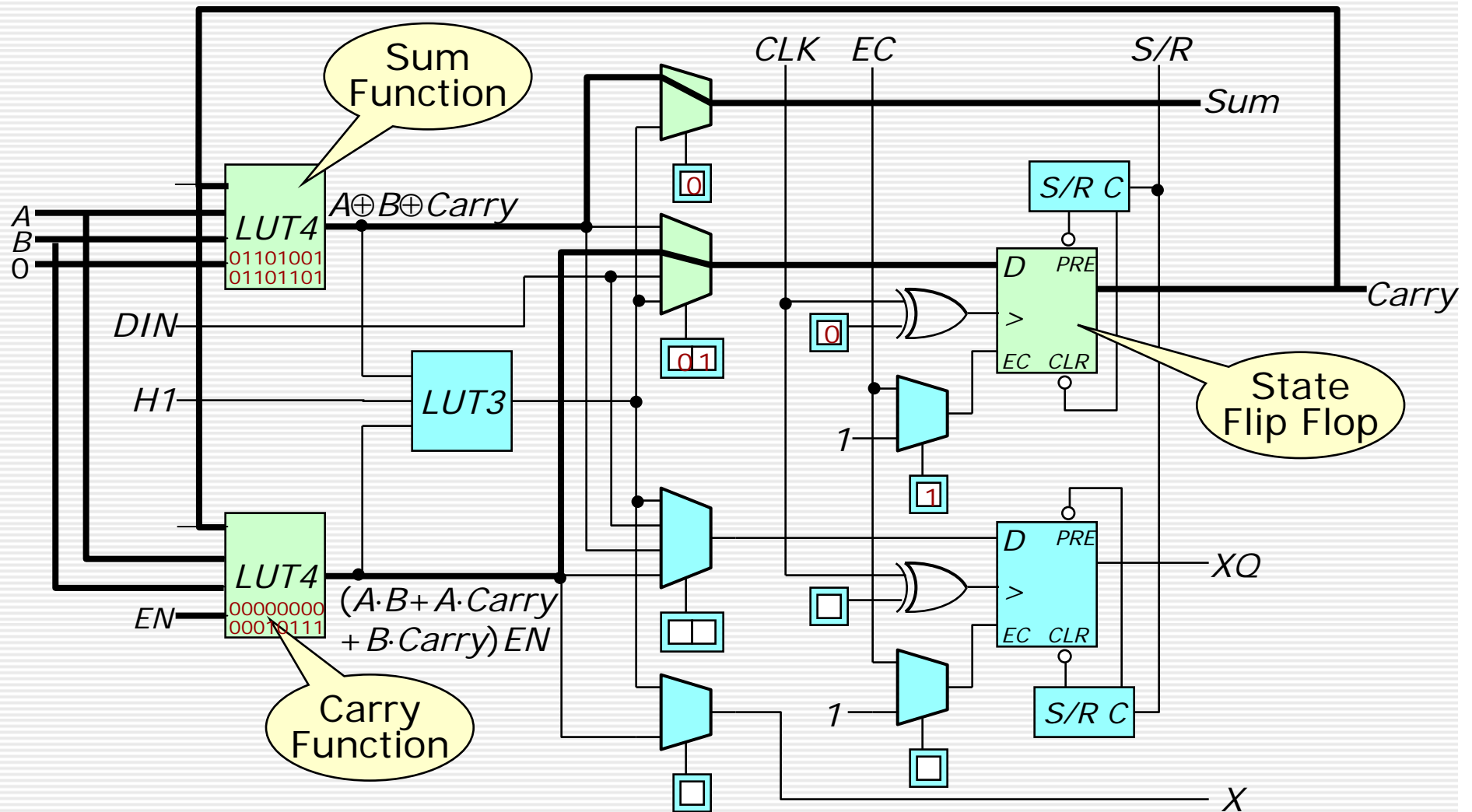


- Lookup table implements logic functions.
- Multiplexors and pass transistors implement routing.
- Switch matrix contains configurable clusters of pass transistors.
 - » provides wide variety of routing options

Xilinx Configurable Logic Block



Implementing Serial Adder in CLB



- Second flip flop still available (thru LUT3).