Programming Technologies and Logic Element Design

Topics

- *What makes PLDs programmable?*
- *Why are they capable of realizing different combinational and sequential logic?*
- *What are SRAM-based FPGA, antifuse-based FPGA, MUX-based FPGA, …?*
- *Choices of programming technologies?*
- *Typical logic element structures?*

Programmable Switches

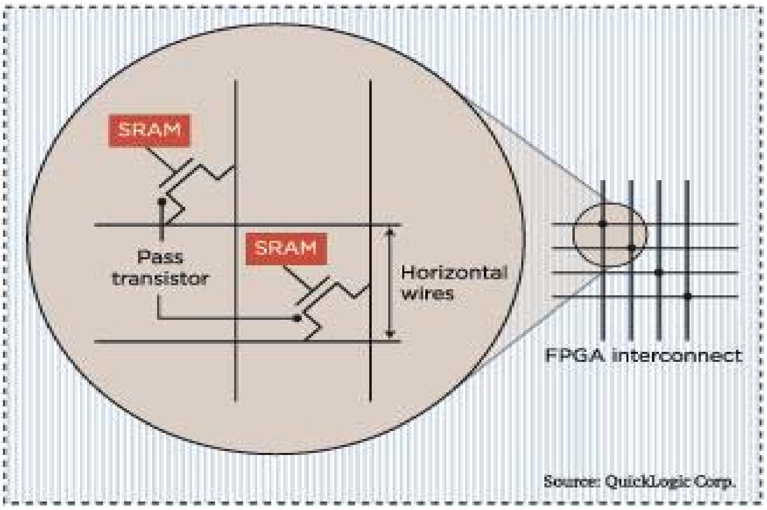
- Programmable switches are used for connections of wire segments in a FPLD.
- A FPLD contains over millions of programmable switches.
- So they should
  - consume as little chip area as possible
  - have low ON resistance and very high OFF resistance
  - contribute low parasitic capacitance
  - be easy to fabricate in a large number reliably.

Programming Switches

- SRAM-based
  - volatile
  - reprogrammable
- Antifuse-based
  - non-volatile
  - non-reprogrammable
- Flash-based
  - non-volatile
  - reprogrammable
- Which one is better???

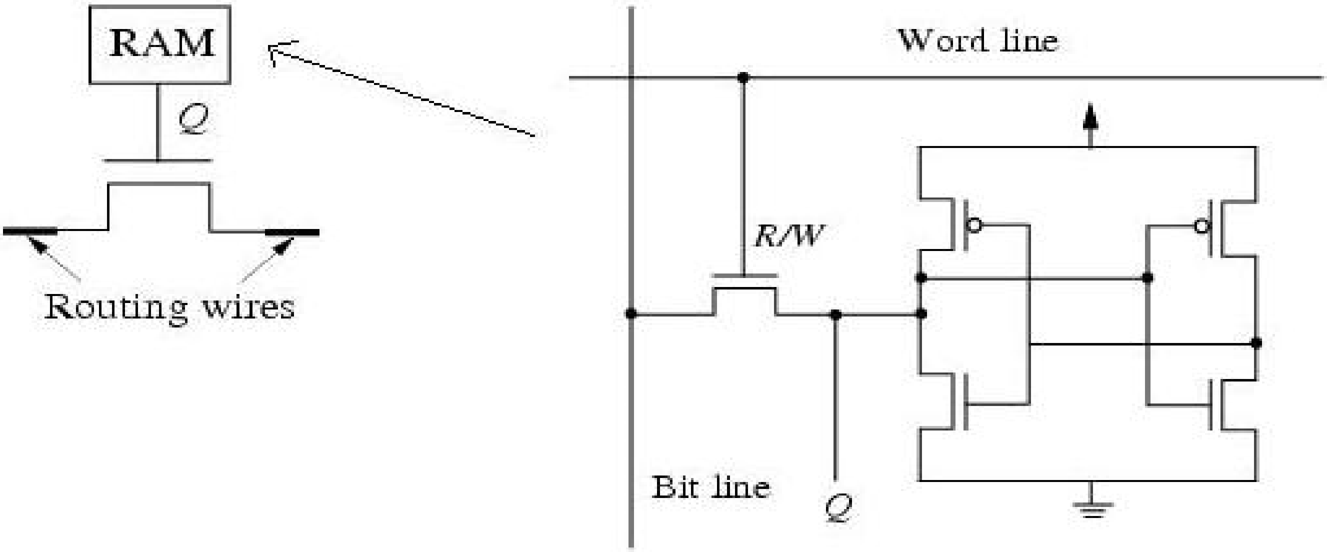
SRAM-based Programmable Connection

- Use a SRAM cell to control a pass transistor or multiplexer.
- Bit in SRAM determines state of connection.



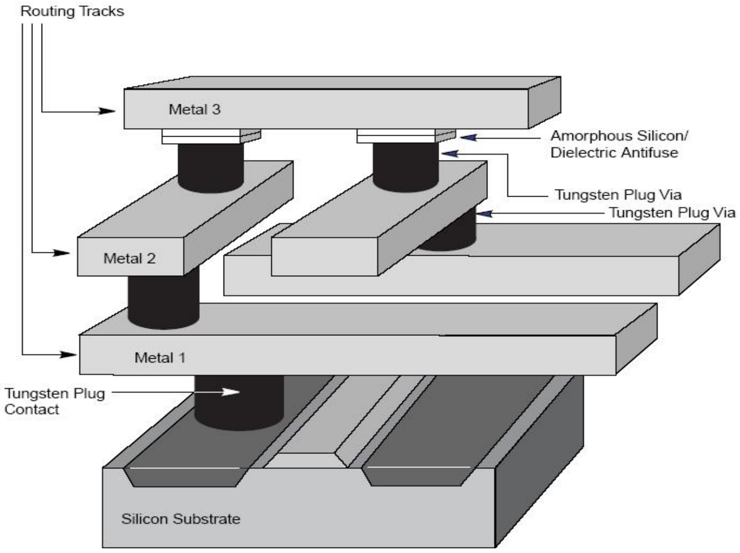
Source: QuickLogic Corp.

SRAM-based Programmable Connection



Antifuse-based Programmable Connection

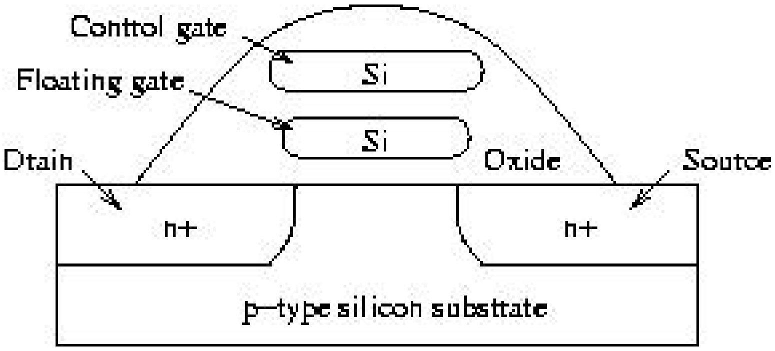
- Use antifuse as programmable switch.
- Antifuse – the opposite of regular fuse i.e., open until a programming current is forced through it.



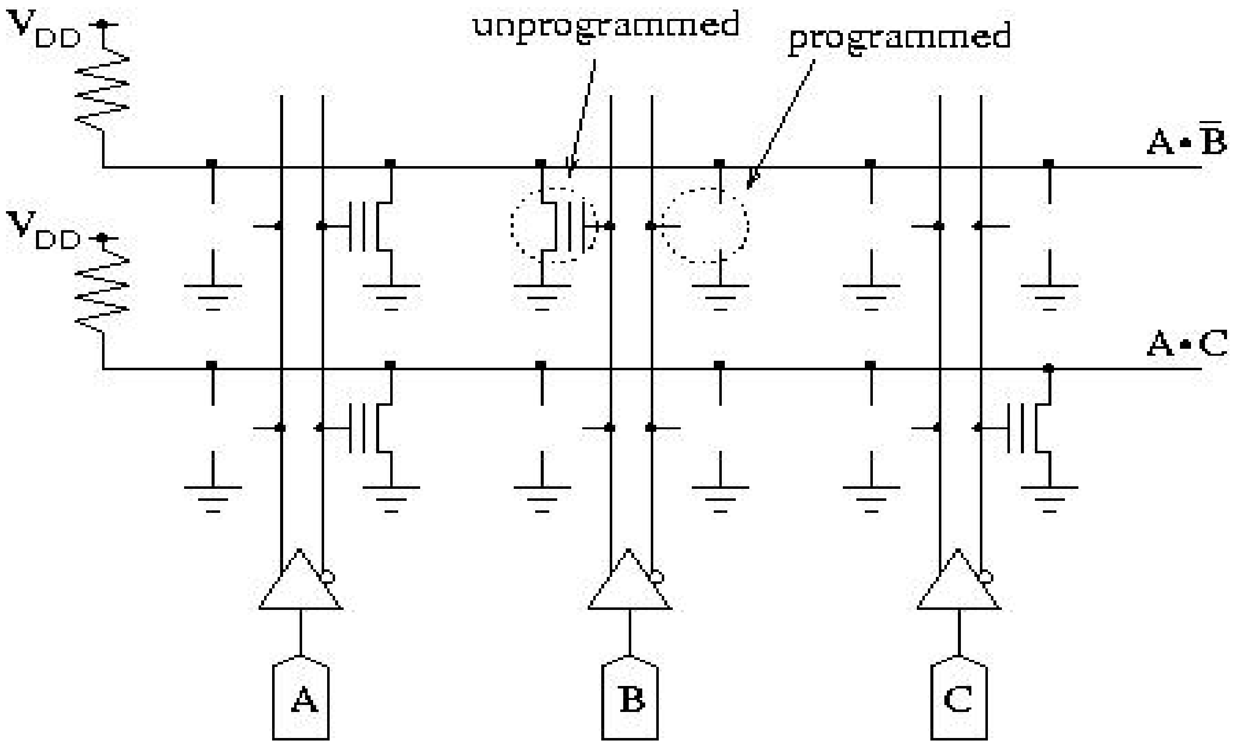
Source: Actel Corp.

Flash-based Programmable Connection

- Use EEPROM transistor as programmable switch.
- An n-channel EEPROM switch is programmed (electrons get trapped on floating gate) by applying higher-than-operation voltages to the drain and the control gate.



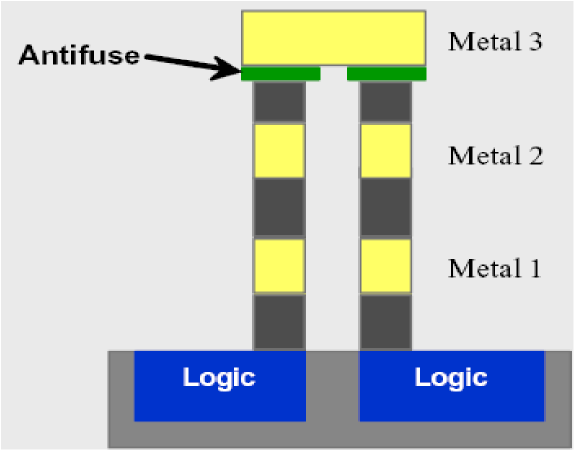
Flash-based Programmable Connection



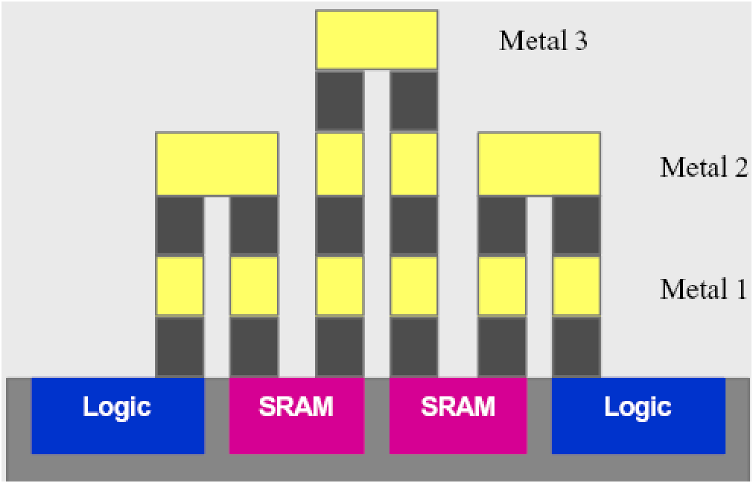
Programming Technologies Comparison

Technology	SRAM	Poly-diffusion antifuse	Metal-metal antifuse	Flash
Manufacturing complexity	+	-	-	-
Size	largest	small	occupy no device floor	medium
ON resistance	medium	small	smallest	medium
OFF capacitance	medium	small	smallest	medium
Reprogrammable?	Yes	No	No	Yes
Volatile?	Yes	No	No	No

Interconnect: Antifuse vs SRAM-based



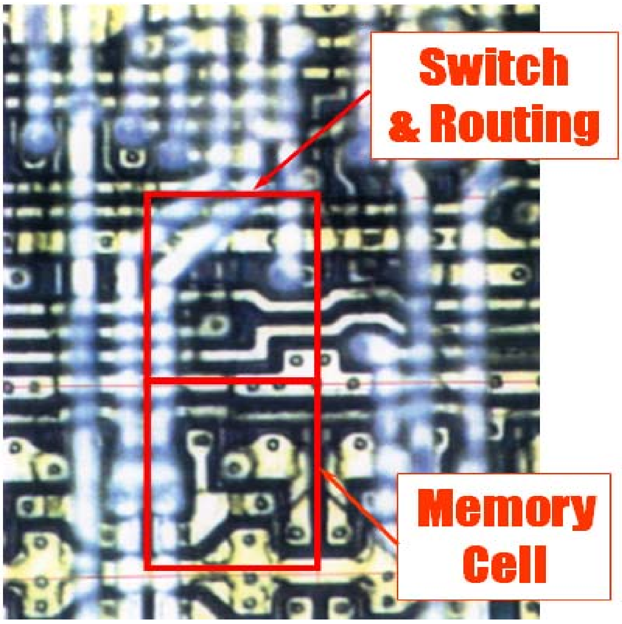
Antifuse-based interconnect



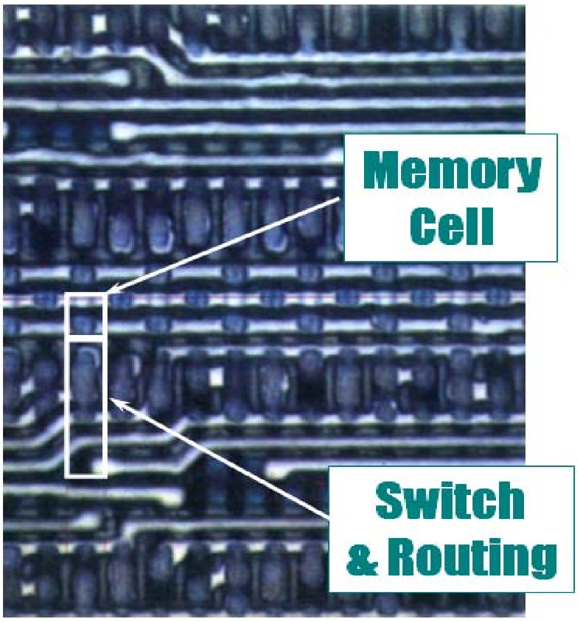
SRAM-based interconnect

Source: Actel Corp.

Size Comparsion of SRAM vs Flash



SRAM-based PLD

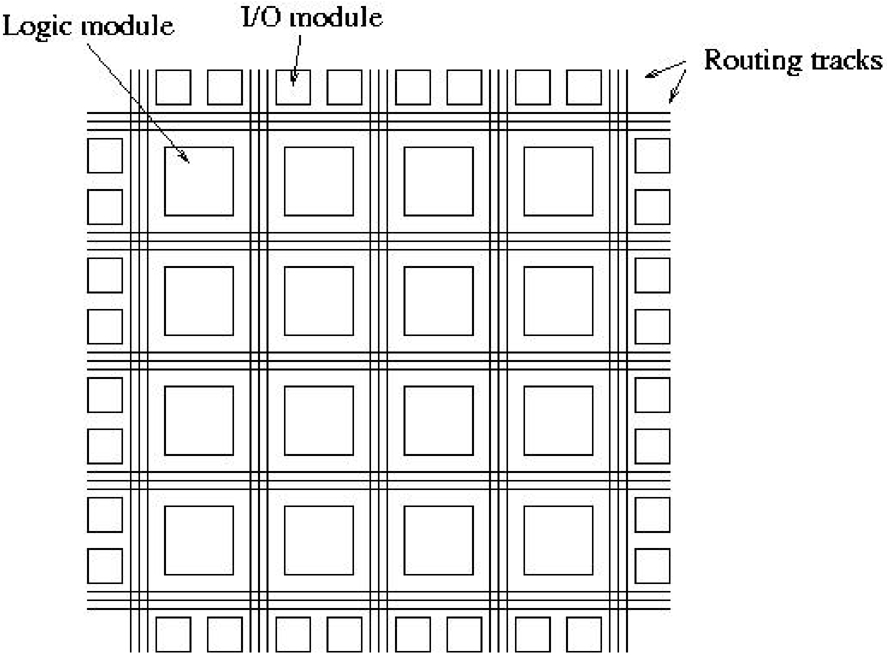


flash-based PLD

Source: Actel Corp.

Programmable Fabrics

- Logic cells
- Interconnect

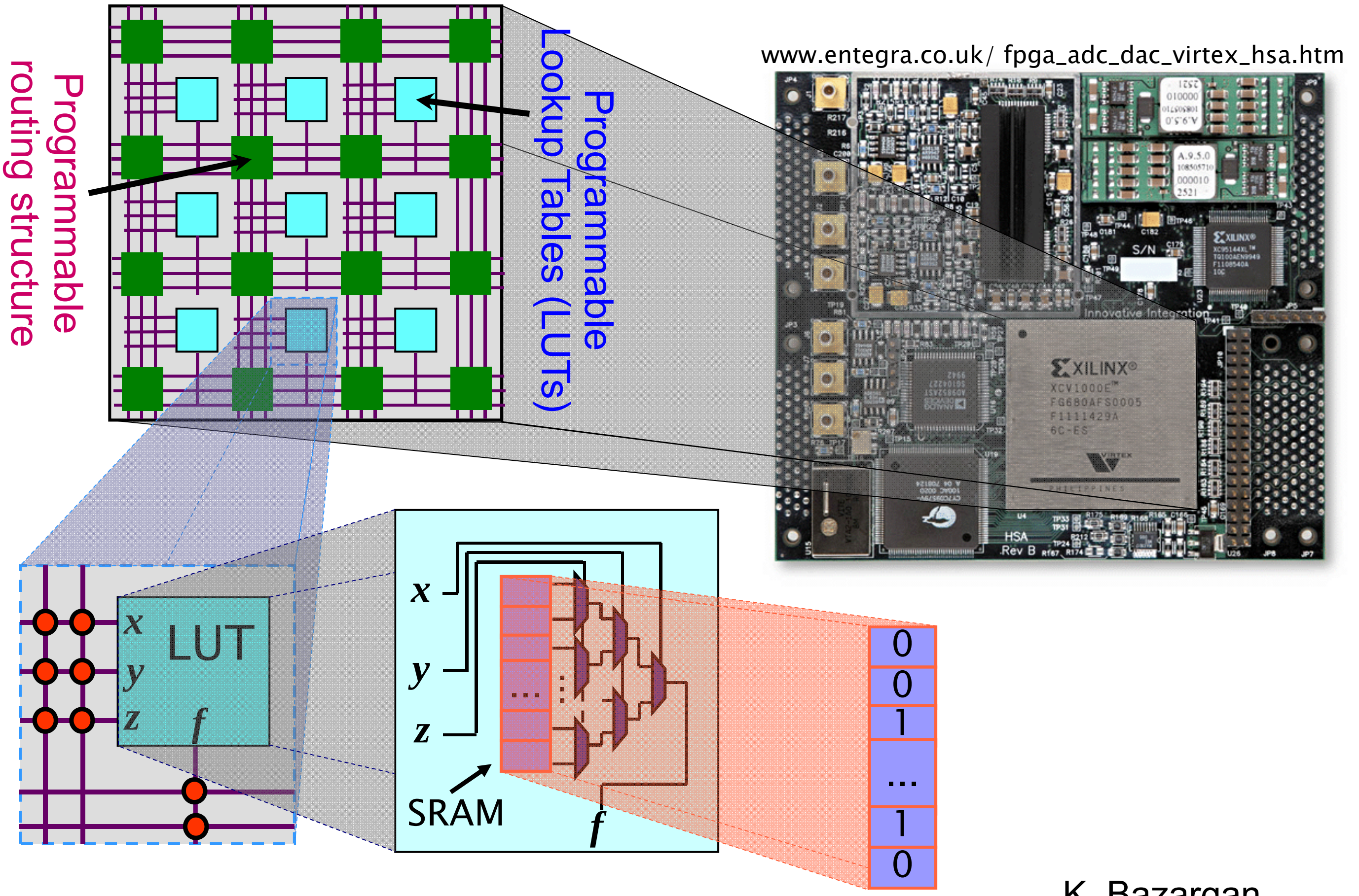


FPGA

Logic Cell

- A logic element has a fixed number of inputs and outputs.
- A LE can implement a certain set of functions.
- FPGA’s LE
  - SRAM-based (lookup table based)
  - MUX-based
- CPLD’s LE
  - Programmable array logic (PAL) based

SRAM-Based FPGA

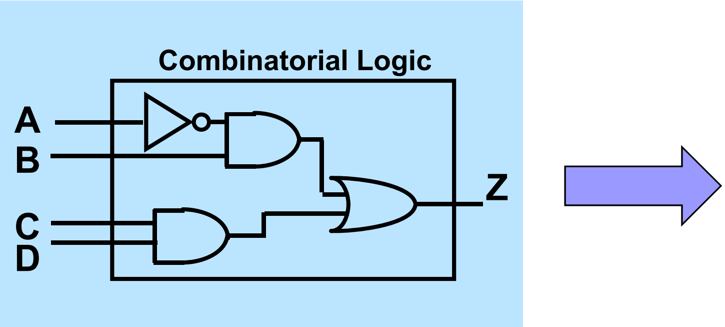

Programmable routing structure
Programmable Lookup Tables (LUTs)
x
y
z
LUT
f
SRAM
...
0
0
1
...
1
0

www.entegra.co.uk/ fpga_adc_dac_virtex_hsa.htm

K. Bazargan

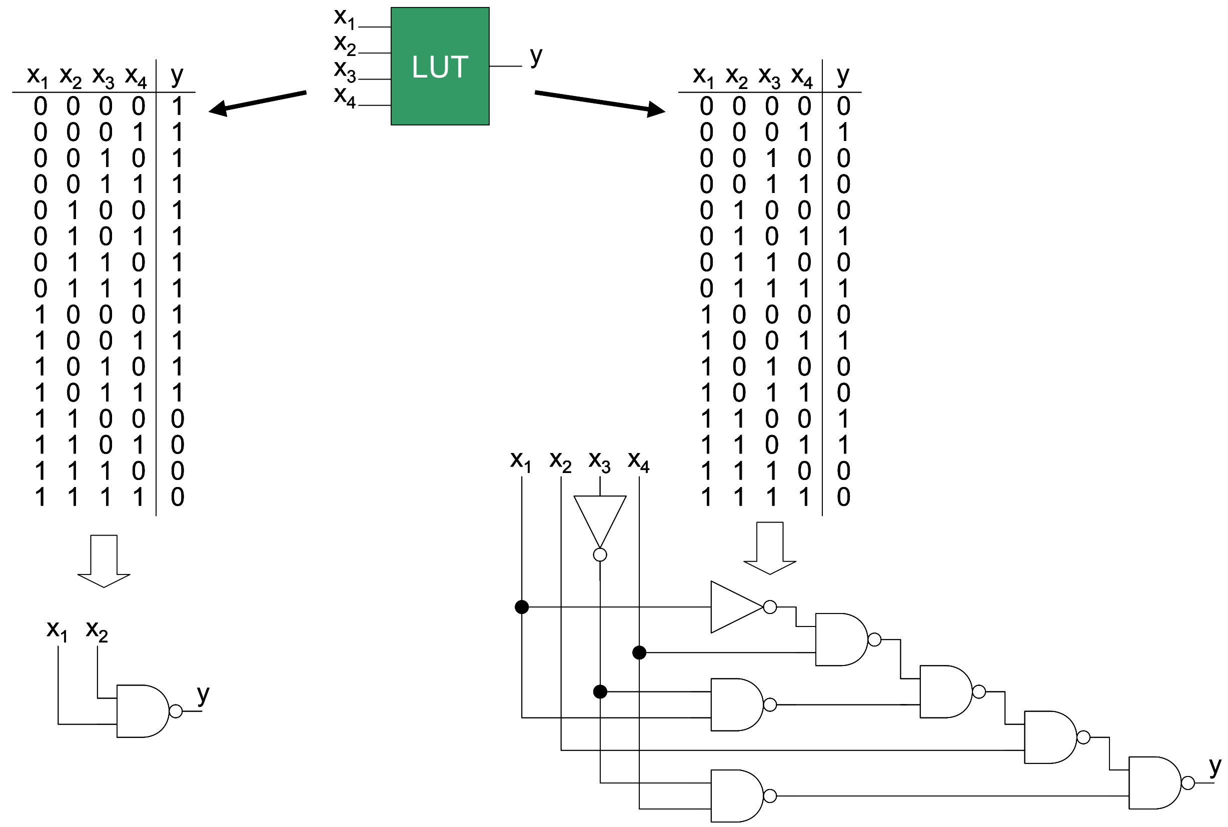
Look-Up Table (LUT)

- Combinatorial logic is stored in look-up tables (LUTs)
  - Capacity is limited by the number of inputs, not by the complexity
- Delay through the LUT is constant



A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
	.	.	.	
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

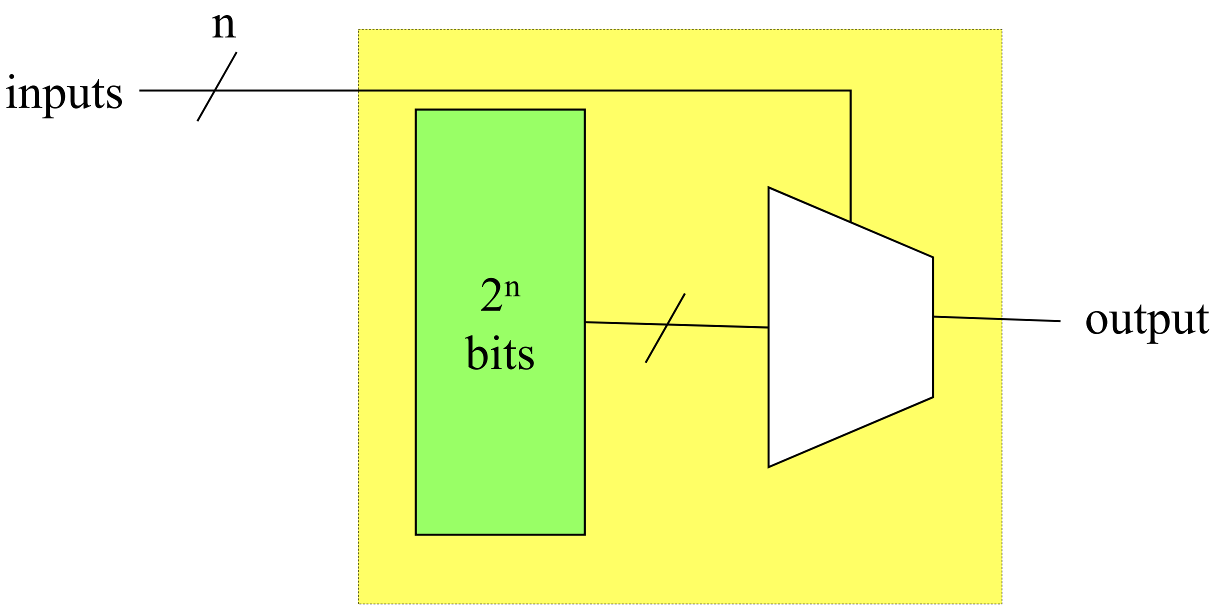
LUT Functionality



x_1	x_2	x_3	x_4	y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

x_1	x_2	x_3	x_4	y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

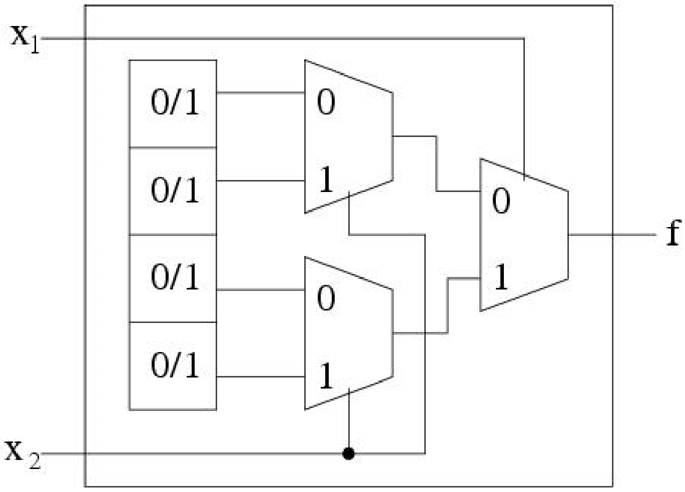
LUT Structure



n-input LUT

Use SRAM for LUT

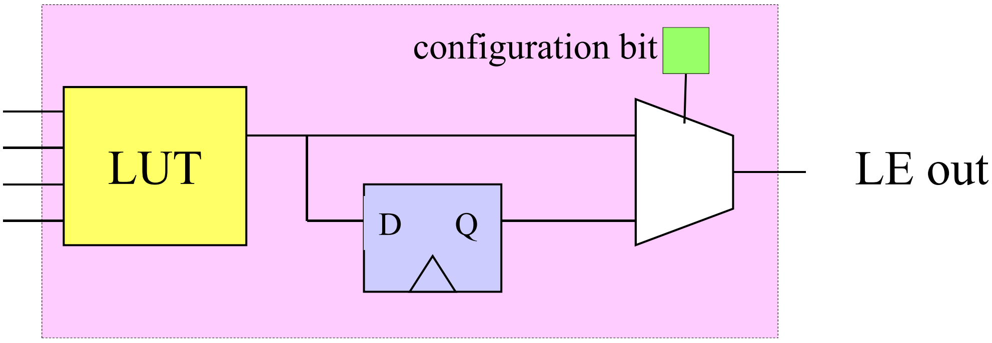
- Use SRAM as lookup table for combinational function.



What configuration will get f = $x_1' x_2 + x_1 x_2'$?

SRAM (LUT)-Based Logic Cell

- Logic cell includes combinational function + flip-flop(s).
- Flip-flop is selected into the circuit as needed.



Logic cell

Evaluation of SRAM-based LUT

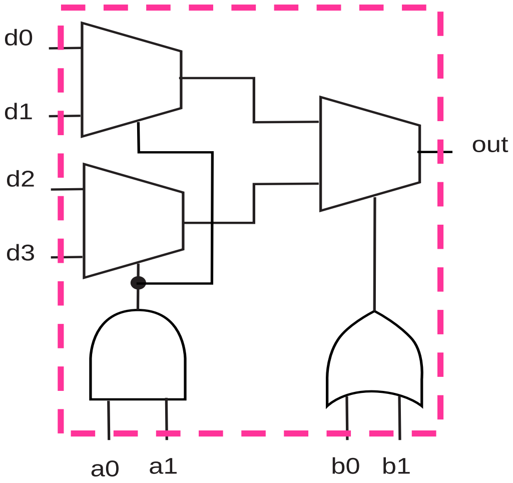
- k-input LUT can handle all functions of ≤ k inputs.
- All logic functions take the same amount of space.
- SRAM is larger than static gate equivalent of function.
- All functions have the same delay.
- Burns power at idle (leakage power)
- Shutdown not practical since power must be maintained to retain configuration.

Static CMOS gate vs. LUT

- Number of transistors:
  - NAND/NOR gate has 2n transistors.
  - 4-input LUT has 80 transistors in SRAM, 96 in multiplexer.
- Delay:
  - 4-input NAND gate has 9τ delay.
  - SRAM decoding has 21τ delay.
- Power:
  - Shutdown of static CMOS logic can reduce leakage power.
  - Shutdown of SRAM not practical since power must be maintained to retain configuration.

MUX-based Logic Cell

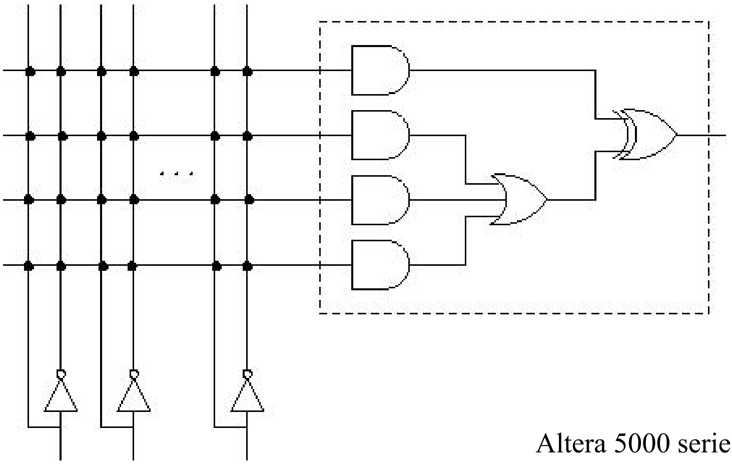
- Consists of a few multiplexers and simple logic gates.
- Programmed by connecting variables or constants to the inputs of the logic cell.



What configuration will get f = $x_1'x_2+x_1x_2'$?

PAL-based Logic Cell

- CPLDs use logic cells evolved from a PAL-based architecture.
- Each logic cell consists of wide fan-in (20 to over 100 inputs) AND gates feeding into an OR gate.



Altera 5000 series macrocell (simplified)

Idea behind each Logic Cell Type

- SRAM-based
  - □ Use SRAM to store the truth table of a function
- MUX-based
  - □ Use multiplexers as function generator
- PAL-based
  - □ Use a sum of product approach to implement function

SRAM-based FPGAs

- Program logic functions & interconnects using SRAM.
- Advantages
  - □ Reprogrammable
  - □ Dynamically reconfigurable
  - □ Uses standard processes
- Disadvantages
  - □ Higher static power dissipation than antifuse and flash, and shutdown is not practical since power must be maintained to retain configuration.
  - □ Require loading from external device at power up (extra device, extra time, extra power).
  - □ Possible to steal, disrupt configuration bits.

Antifuse-based FPGAs

- Use antifuse for programmability.
- Logic blocks in antifuse-based FPGAs are generally based on multiplexing (hence also called MUX-based FPGAs).
- Advantages
  - □ Non-volatile
  - □ Permanent (suitable for high radiation environment)
  - □ Lower static power consumption and shutdown is ok
  - □ Design security
- Disadvantages
  - □ Non-reprogrammable
  - □ Require more than standard processes

Flash-based FPGAs

- Use flash-based switches for programmability.
- Logic blocks generally based on multiplexing.
- Advantages
  - □ Non-volatile
  - □ Lower static power consumption and shutdown is ok
  - □ Design security
  - □ Reprogrammable
- Disadvantage
  - □ Require more than standard processes
- Remark: Hybrid "flash-SRAM" FPGAs simply combine flash-based storage (hold configuration pattern) and SRAM-based logic structure on one chip

SRAM, Antifuse, and FLASH-FPGAs Compared

	SRAM	Antifuse	FLASH
Technology node	state-of-the-art	1 or more generations behind	1 or more generations behind
Reprogrammable	Yes	No	Yes
Volatile	Yes	No	No
External config. file	Yes	No	No
Instant-on	No	Yes	Yes
IP security	Acceptable (w/ encryption)	V. good	V. good
Size of config. cell	Large (6 transistors)	V. small	Medium-small (2 transistors)
Power consumption	High	Low	Medium
Radiation immunity	No	Yes	No

Reading

- “Flash Flood – inside FPGAs’ Non-volatile Companions”, *FPGA and Structured ASIC Journal*, www.fpgajournal.com
- “Design Security in Nonvolatile Flash and Antifuse FPGAs”, *D & R Industry Articles*, www.design-reuse.com