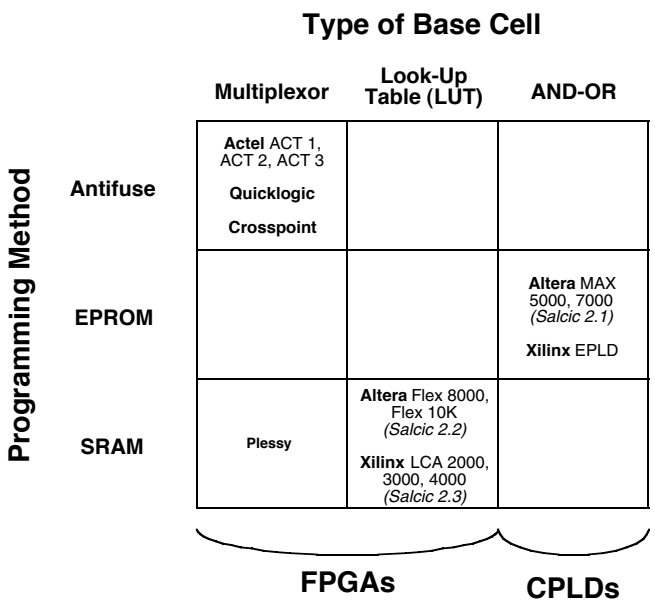
Types of FPLDs

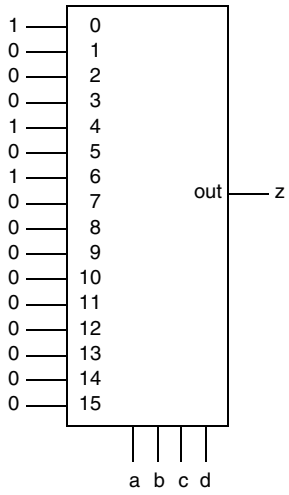


- Layout / routing
  - Row-based: Actel
  - Matrix-based: Altera, Quicklogic, Xilinx

Implementing a Truth Table Using a Multiplexor

- Besides and-or structures, another alternative is to use a 4-input multiplexor

a	b	c	d	x
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

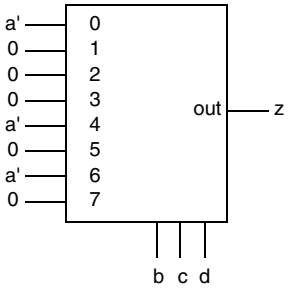


- Any function of N inputs can be implemented using a 2^N to 1 multiplexor

Implementing a Truth Table Using a Multiplexor (cont.)

- An alternative is to "fold" the truth table, and tie each input to either 1, 0, or the MSB, and only use a 3-input multiplexor

a	b	c	d	x
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

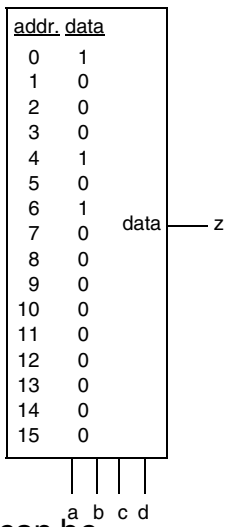


- Any function of N inputs can be implemented using a 2^{N-1} to 1 multiplexor
  - Some FPLDs are based on multiplexors, and attach simple gates to selector lines

Implementing a Truth Table Using a ROM

- Yet another alternative is to use a ROM

a	b	c	d	x
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



- Any function of N inputs can be implemented using a 2^Nx 1 bit ROM
  - Some FPLDs are based on static RAMs (SRAMS) loaded at power-up; these are said to use *look-up tables* (LUTs)

Different Implementation Styles

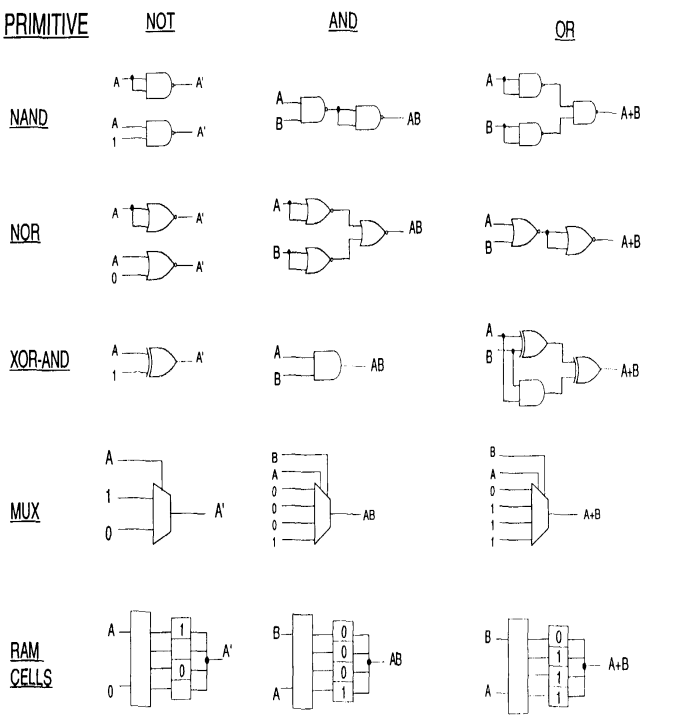


Diagram from *Digital Design Using Field Programmable Gate Arrays*, Chan & Mourad, Prentice Hall 1994

Row-Based Layout

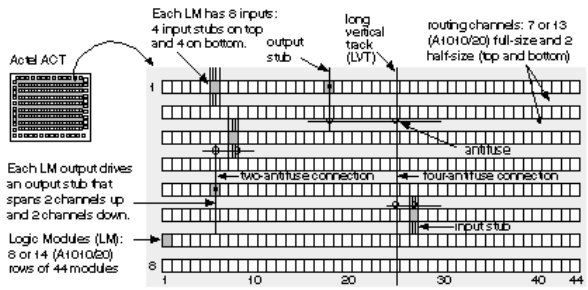


Figure from *Application-Specific Integrated Circuits*, Smith, Addison-Wesley, 1997

- Cells are arranged in rows
  - Horizontal channels between rows
  - Vertical channels above cells: some short, some long
  - Each *channel* contains a fixed number of *tracks*, each track holds one wire
    - Wires may be divided into fixed-length *segments* within each track
  - In figure above, cell inputs connect to horizontal wires, outputs to vertical wires

Matrix-Based Layout

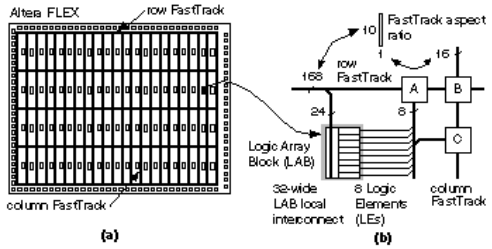


Figure from *Application-Specific Integrated Circuits*, Smith, Addison-Wesley, 1997

- Cells are arranged in an array (*matrix*)
  - Horizontal and vertical channels between cells
  - Each *channel* contains a fixed number of *tracks*, each track holds one wire
  - In figure above:
    - Cell inputs connect to horizontal tracks
    - Box A connects cell output(s) to horizontal tracks, and box C connects cell output(s) to vertical tracks
    - Box B acts as a switchbox between horizontal and vertical tracks

Antifuse Routing

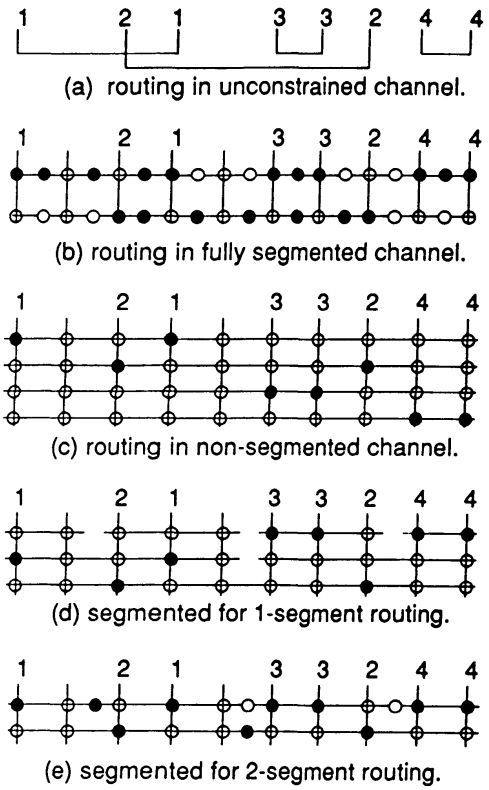


Figure from *Field-Programmable Gate Array Technology*, Trimberger, Kluwer, 1994

Antifuse Routing (cont.)

- Fully segmented
  - Switch at every cross point normally passes signals through vertically and horizontally, but can connect the vertical and horizontal tracks
  - Antifuse connects or disconnects the segments of the horizontal channel
- Non-segmented
  - Excessive area requirements
- 1-segment routing
  - Divides the tracks into segments of varying lengths, which allows each net to be routed in a track of more or less the appropriate size
- 2-segment routing
  - Allows track segments to be joined