

PLAs

- A PLA (*programmable logic array*) replicates a 2-level *and-or* structure many times in a programmable array
 - Parts of a CPU's datapath or next-state logic can be built out of PLAs
 - Small circuits can be built out of PLAs
- At the input of each gate, there's a "fuse" which can be left whole, or broken
 - The designer can control which inputs go to each **and** gate, and which outputs of the **and** gates go to each **or** gate
- A PLA can be either
 - Mask programmable — customer orders a programmed PLA from manufacturer
 - Field programmable — customer can program PLA (once)

2

Spring 2006, Lecture 21

PLAs

- A PLA (*programmable logic array*) replicates a 2-level *and-or* structure many times in a programmable array

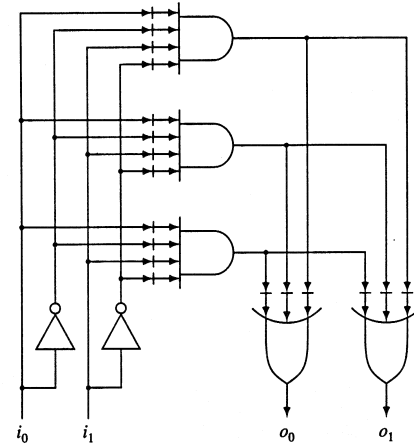


Diagram from *Computer Systems*, Maccabe, Irwin 1993

- This PLA has 2 inputs, 2 outputs, and can represent up to 3 product terms

3

Spring 2006, Lecture 21

PLA Example

- This is an *abstract* diagram of a PLA with 6 inputs, 4 outputs, which can represent up to 12 product terms

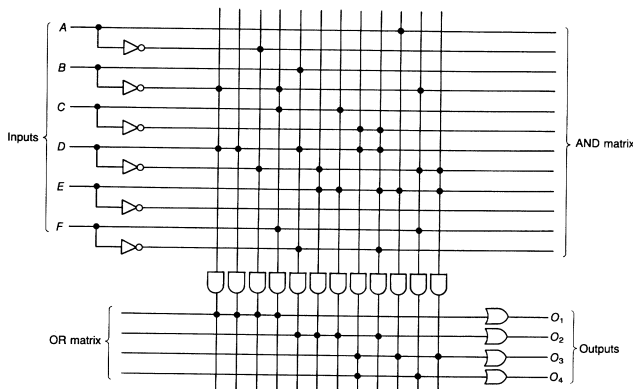


Diagram from *Digital Design*, Johnson & Karim, PWS-Kent 1987

- Try the Java KMap->PLA animation at <http://tech-www.informatik.uni-hamburg.de/applets/kvd>

4

Spring 2006, Lecture 21

Field-Programmable Logic Device

- The next evolutionary step beyond the PLA is the *field-programmable logic device* (FPLD), also called the:
 - Field-programmable gate array (FPGA)
 - Complex programmable logic device (CPLD)
- FPLD characteristics
 - Based on either an array of PLA-like *and-or* structures, or on look-up tables
 - May include not only this combinational logic, but also D (or more complex) flip-flops to more easily build sequential circuits
 - Many can be "programmed" repeatedly
 - Connect I/O to buses, connect buses to cells, control cell functions
 - Available in different sizes up to millions of gates (speed/density/cost tradeoffs)

5

Spring 2006, Lecture 21

Programming Using Antifuses

- An *antifuse* is normally open (“off”); when enough current (5–15mA) passes through it it closes (“on”)
 - Current melts a thin insulating dielectric and forms a permanent silicon link
 - Disadvantage — can only program once
 - Programmed in a special hardware device
 - An antifuse FPLD may contain 750,000 antifuses, but only about 2% of them typically need to be programmed
 - Takes about 5-10 minutes for each chip
- Advantages:
 - Small — about the size of a via
 - Low resistance, low capacitance = fast

6

Spring 2006, Lecture 21

Programming Using EPROMs & EEPROMs (Floating Gates)

- EPROM programming & operation:
 - To program: a high programming voltage is applied, semi-permanently turning the transistor off (“open”)
 - To erase: the transistor is exposed to UV light, which returns the transistor to normal operation (“closed”)
 - Can be reprogrammed many times
- EEPROMs are similar, but are erased electrically
 - Faster to erase than EPROM, and can be done “in-circuit”
 - Requires larger cell than EPROM
- Advantages
 - Can be programmed repeatedly, in-circuit
 - Fairly small — requires only 1 transistor

7

Spring 2006, Lecture 21

Programming Using Static RAMs (SRAMs)

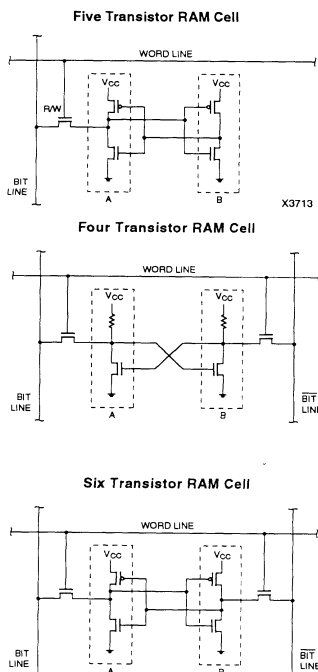


Figure from *Field-Programmable Gate Array Technology*, Trimberger, Kluwer, 1994

8

Spring 2006, Lecture 21

Programming Using Static RAMs (SRAMs) (cont.)

- Disadvantages:
 - Must load configuration from ROM, disk, etc. on power-up
 - Large — requires several transistors
- Advantages:
 - Can be programmed repeatedly, in-circuit
 - Can be programmed quickly (< 1ms)
 - Part has been 100% tested at factory
 - Same basic process as CMOS, so quickly takes advantage of new fab processes
 - CMOS also requires less power than circuits requiring pull-up resistors
- As with other techniques, can be used to connect I/O to buses, connect buses to cells, control cell functions
 - Also used to construct RAM blocks, stacks, queues, and other memory blocks

9

Spring 2006, Lecture 21

Types of FPLDs

		Type of Base Cell		
		Multiplexor	Look-Up Table (LUT)	AND-OR
Programming Method	Antifuse	Actel ACT 1, ACT 2, ACT 3 Quicklogic Crosspoint		
	EPROM			Altera MAX 5000, 7000 (Salicic 2.1) Xilinx EPLD
	SRAM	Plessey	Altera Flex 8000, Flex 10K (Salicic 2.2) Xilinx LCA 2000, 3000, 4000 (Salicic 2.3)	
		FPGAs		CPLDs

Layout / routing

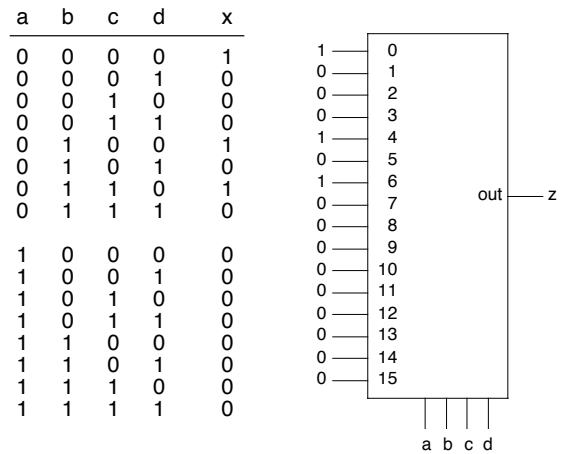
- Row-based: Actel
- Matrix-based: Altera, Quicklogic, Xilinx

10

Spring 2006, Lecture 21

Implementing a Truth Table Using a Multiplexor

- Besides and-or structures (see 1st slide), an alternative is to use a 16-input mux



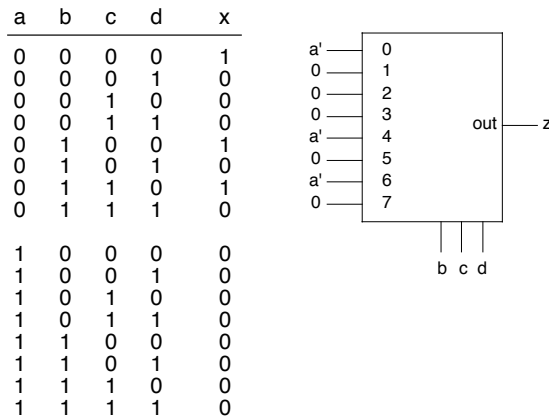
- Any function of N inputs can be implemented using a 2^N to 1 multiplexor

12

Spring 2006, Lecture 21

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 8-input multiplexor



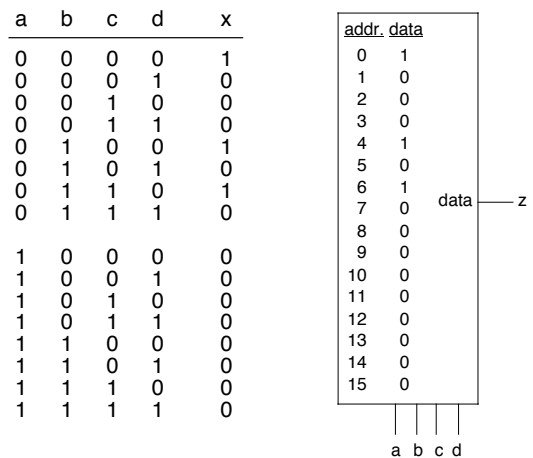
- Any function of N inputs can be implemented using a 2^{N-1} to 1 multiplexor
 - Some FPLDs are based on multiplexors, and attach simple gates to selector lines

13

Spring 2006, Lecture 21

Implementing a Truth Table Using a ROM

- Yet another alternative is to use a ROM



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

14

Spring 2006, Lecture 21

Row-Based Layout

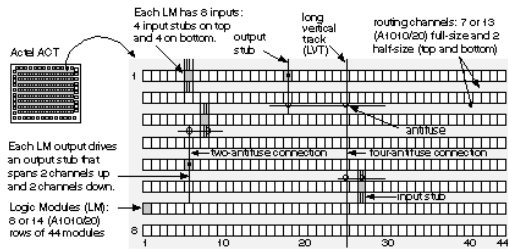


Figure from *Application-Specific Integrated Circuits*, Smith, Addison-Wesley, 1997

■ 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

16

Spring 2006, Lecture 21

Matrix-Based Layout

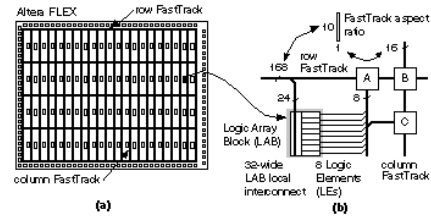


Figure from *Application-Specific Integrated Circuits*, Smith, Addison-Wesley, 1997

■ 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

17

Spring 2006, Lecture 21

Antifuse Routing

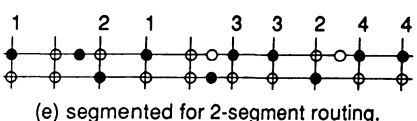
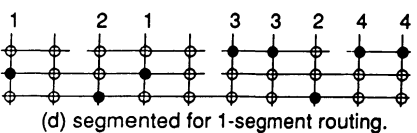
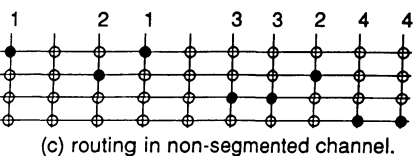
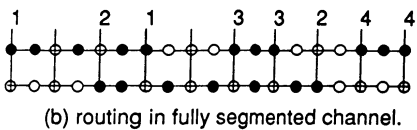
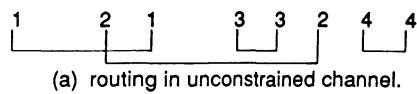


Figure from *Field-Programmable Gate Array Technology*, Trimberger, Kluwer, 1994

18

Spring 2006, Lecture 21

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

19

Spring 2006, Lecture 21