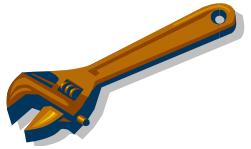


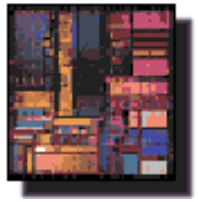




Digital Systems Design Automation

Unit 1: Course Introduction and Overview

Lecture 1.4: Taxonomy of Integrated Circuits



Anand Raghunathan

raghunathan@purdue.edu

© Anand Raghunathan

Outline

1.1 Moore's Law

1.2 Design Complexity and need for EDA

1.3 Course Overview

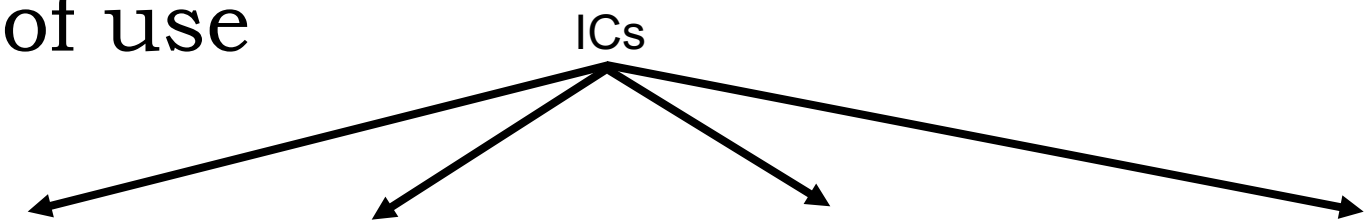
1.4 Taxonomy of integrated circuits

1.5 Levels of abstraction in IC design

1.6 A quick tour of logic level design automation

Taxonomy of Integrated Circuits

- By scope of use

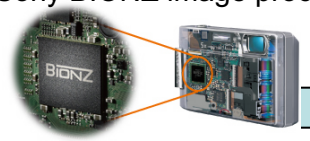


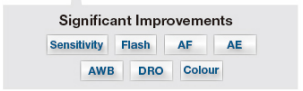
Application Specific IC (ASIC)

Designed for one end system

E.g., Chips used in Digital Cameras, Printers, Gaming Consoles

Sony BIONZ image processor





Application Specific Standard Product (ASSP)

Re-used across multiple similar end-systems

E.g., Mobile phone application processors

Domain Specific Product

Designed for an application domain

E.g., DSPs, Network processors, GPUs, TPUs

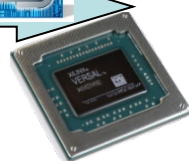
Increasing

Increasing

General Purpose

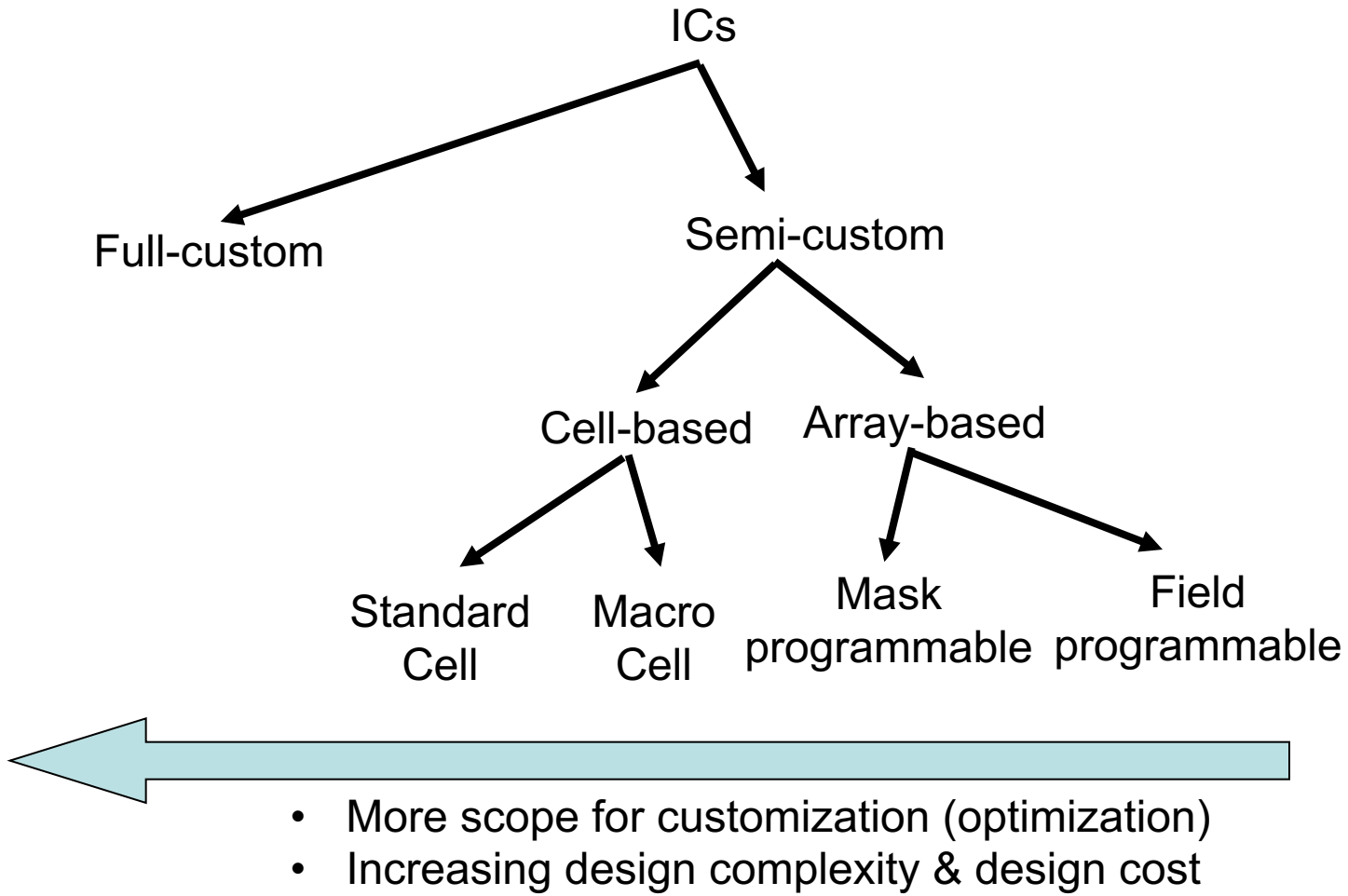
Usable across a wide range of end-systems and applications

E.g., Microprocessors, Memory, FPGAs



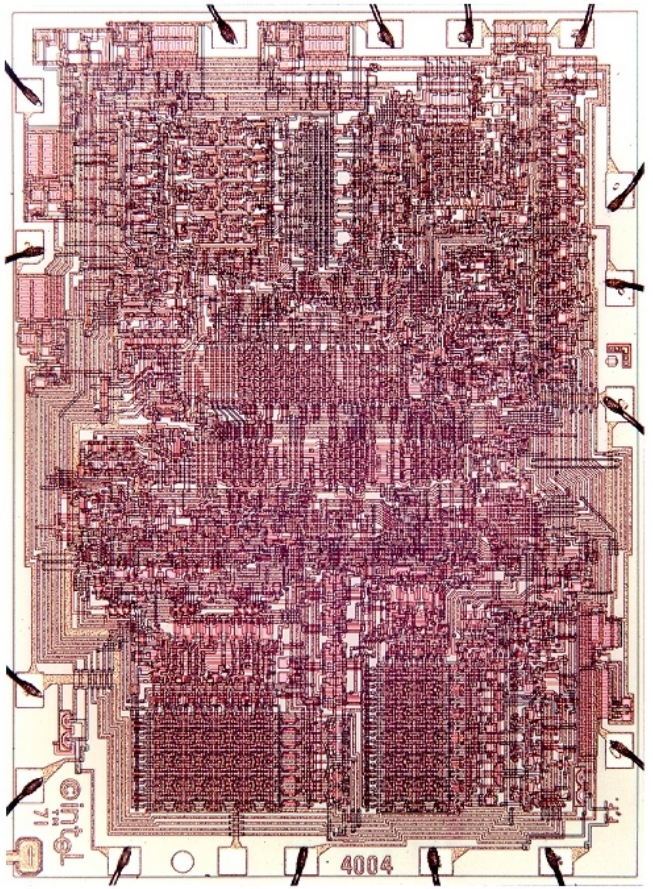
Taxonomy of Integrated Circuits

- By degree of customization during design



Full-Custom Design

- Very high level of customization down to layout (high efficiency)
- High design effort
- Not feasible for complex (e.g., Billion-transistor) ICs



Intel 4004

Cell Based Design

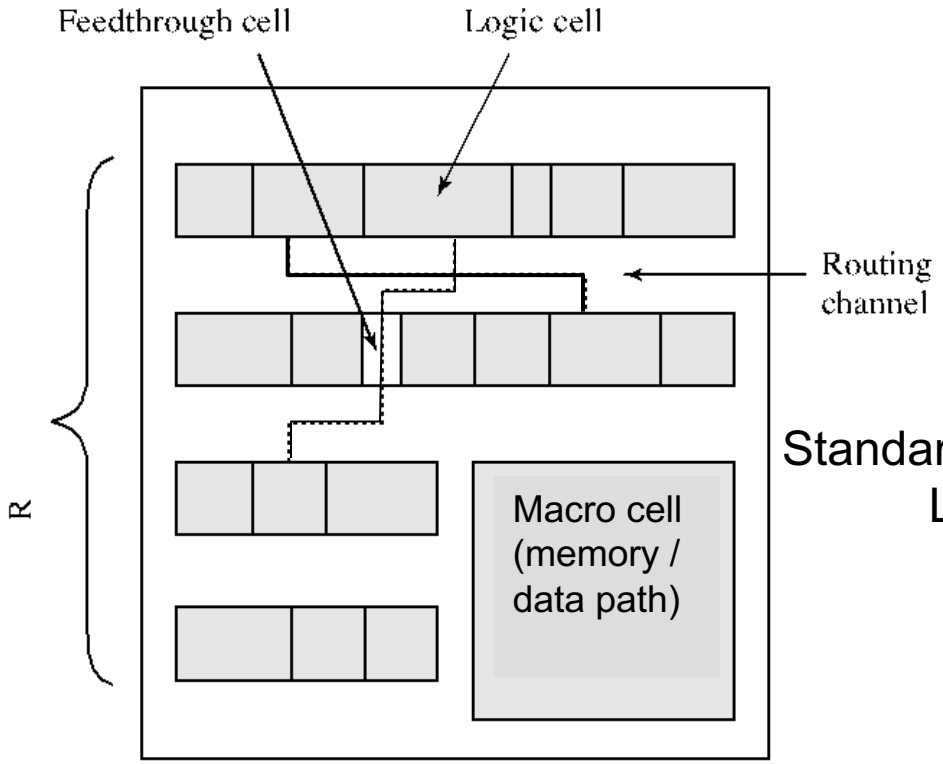
- Cell library contains pre-designed and pre-characterized circuit blocks (e.g., gates, latches, flip-flops, I/O buffers)
- Circuits constructed by instantiating and connecting cells from library
  - Design rules ensure that circuits will work once fabricated
- Library contains different views of each cell (logical/transistor/physical)
  - Timing/power/area models to be used by EDA tools

Example: Nangate 15nm open cell library based on FreePDK

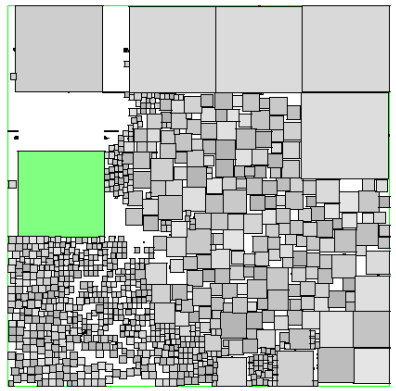
COMBINATIONAL CELLS			
Cell Name	*#In*	*Boolean Function*	*Drive Strengths*
BUF	1	$O = A$	X1, X2, X4, X8, X12 and X16
INV	1	$O = !A$	X1, X2, X4, X8, X12 and X16
AND2	2	$O = A * B$	X1 and X2
AND3	3	$O = A * B * C$	X1 and X2
AND4	4	$O = A * B * C * D$	X1 and X2
OR2	2	$O = A + B$	X1 and X2
OR3	3	$O = A + B + C$	X1 and X2
OR4	4	$O = A + B + C + D$	X1 and X2
NAND2	2	$O = !(A * B)$	X1 and X2
NAND3	3	$O = !(A * B * C)$	X1 and X2
NAND4	4	$O = !(A * B * C * D)$	X1 and X2
NOR2	2	$O = !(A + B)$	X1 and X2
NOR3	3	$O = !(A + B + C)$	X1 and X2
NOR4	4	$O = !(A + B + C + D)$	X1 and X2
MUX2	3	$O = A * !Sel + B * Sel$	X1
XOR2	2	$O = A \oplus B$	X1
XNOR2	2	$O = !(A \oplus B)$	X1
AOI21	3	$O = !((A * B) + C)$	X1 and X2
AOI22	4	$O = !((A * B) + (C + D))$	X1 and X2
OAI21	3	$O = !((A + B) * C)$	X1 and X2
OAI22	4	$O = !((A + B) * (C + D))$	X1 and X2
FA	3	$S = A \oplus B \oplus Cin$, $Cout = A \oplus B * Cin + A * B$	X1
HA	2	$S = A \oplus B$, $Cout = A * B$	X1
SEQUENTIAL CELLS			
Cell Name	*#In*	*Description*	*Drive Strengths*
DFFRNQ	3	D flip-flop with asynchronous $!reset$	X1
DFFSNQ	3	D flip-flop with asynchronous $!set$	X1
SDFFRNQ	5	D flip-flop with scan and asynchronous $!reset$	X1
SDFFSNQ	5	D flip-flop with scan and asynchronous $!set$	X1
LHQ	2	High enable Latch	X1
ADDITIONAL CELLS			
Cell Name	*#In*	*Description*	*Drive Strengths*
CLKBUF	2	Clock buffer	X1, X2, X4, X8, X12 and X16
TBUF	2	Tri-state buffer	X1, X2, X4, X8, X12 and X16
FILL	-	Filler cell	X1, X2, X4, X8 and X16
CLKGATETST	3	Clock gate with test pin	X1
ANTENNA	-	Antenna cell	-
FILLTIE	-	Cell to tie the wells	-
TIEH	-	Tie-high cell	-
TIEL	-	Tie-low cell	-

Cell Based Design

- Standard cells must have the same height
  - Eases physical design
  - Layout organized into rows of cells with routing channels in between them
- Macro cells do not have this restriction
- Common to mix both macro cells (e.g., memories and datapath components) with standard cells (random logic)



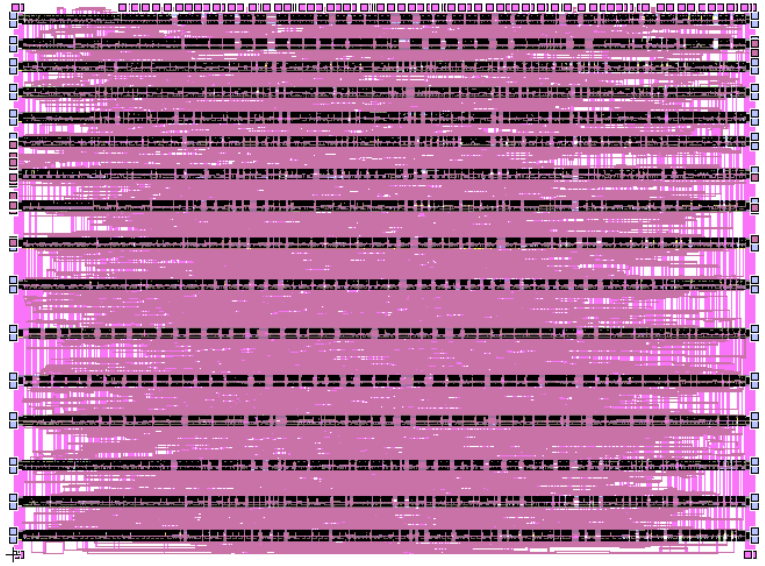
Standard Cell Layout



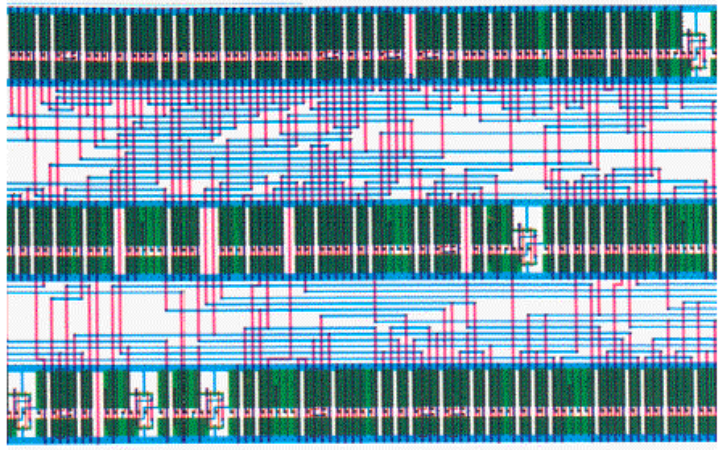
Macro Cell Layout

Standard Cell Based Design: Examples

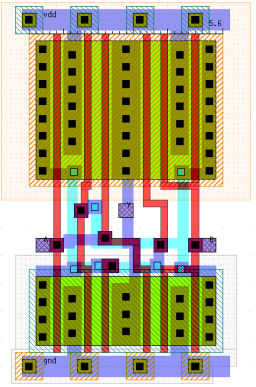
32-bit ALU standard cell layout



Three rows of standard cells

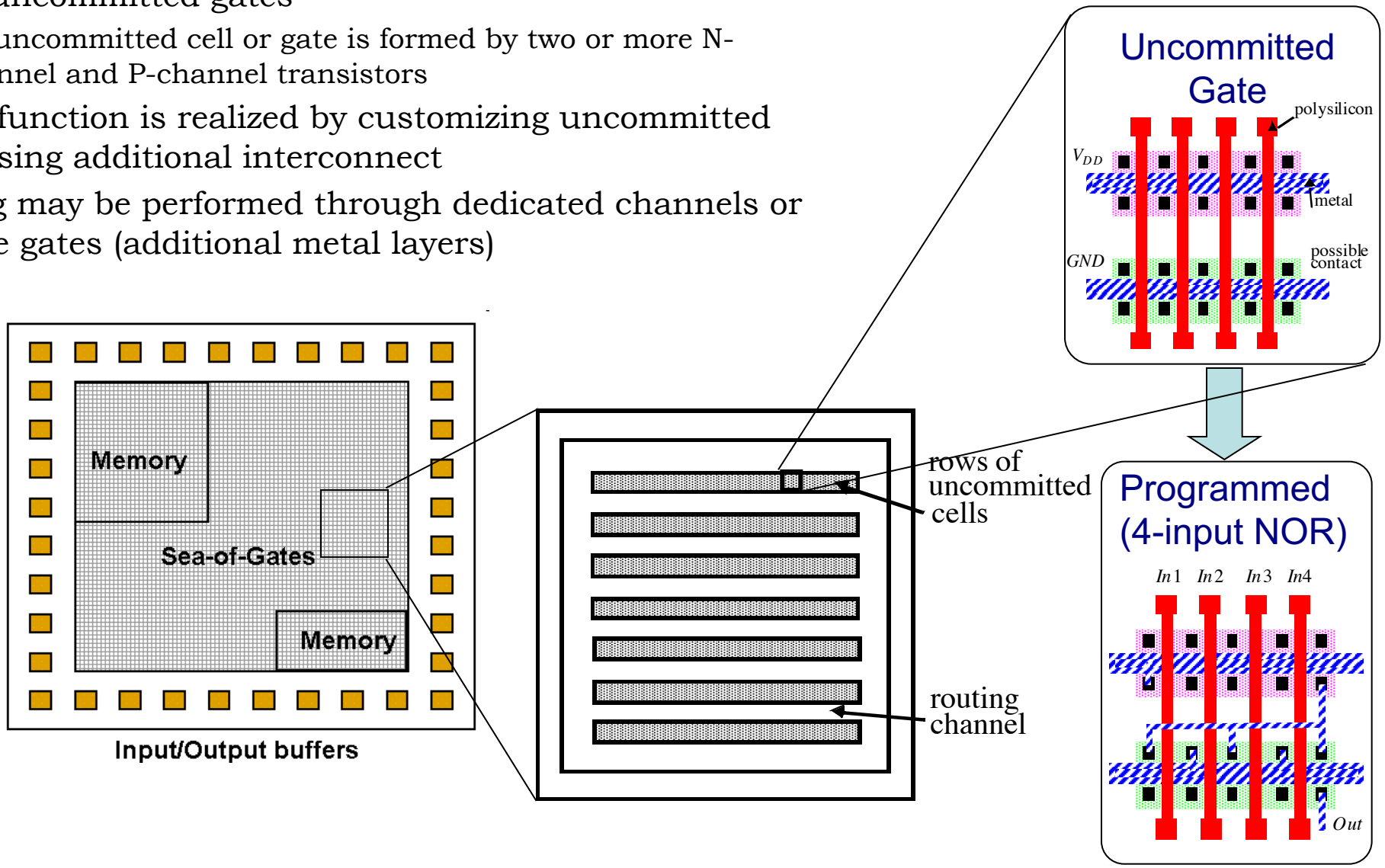


2-input XOR gate



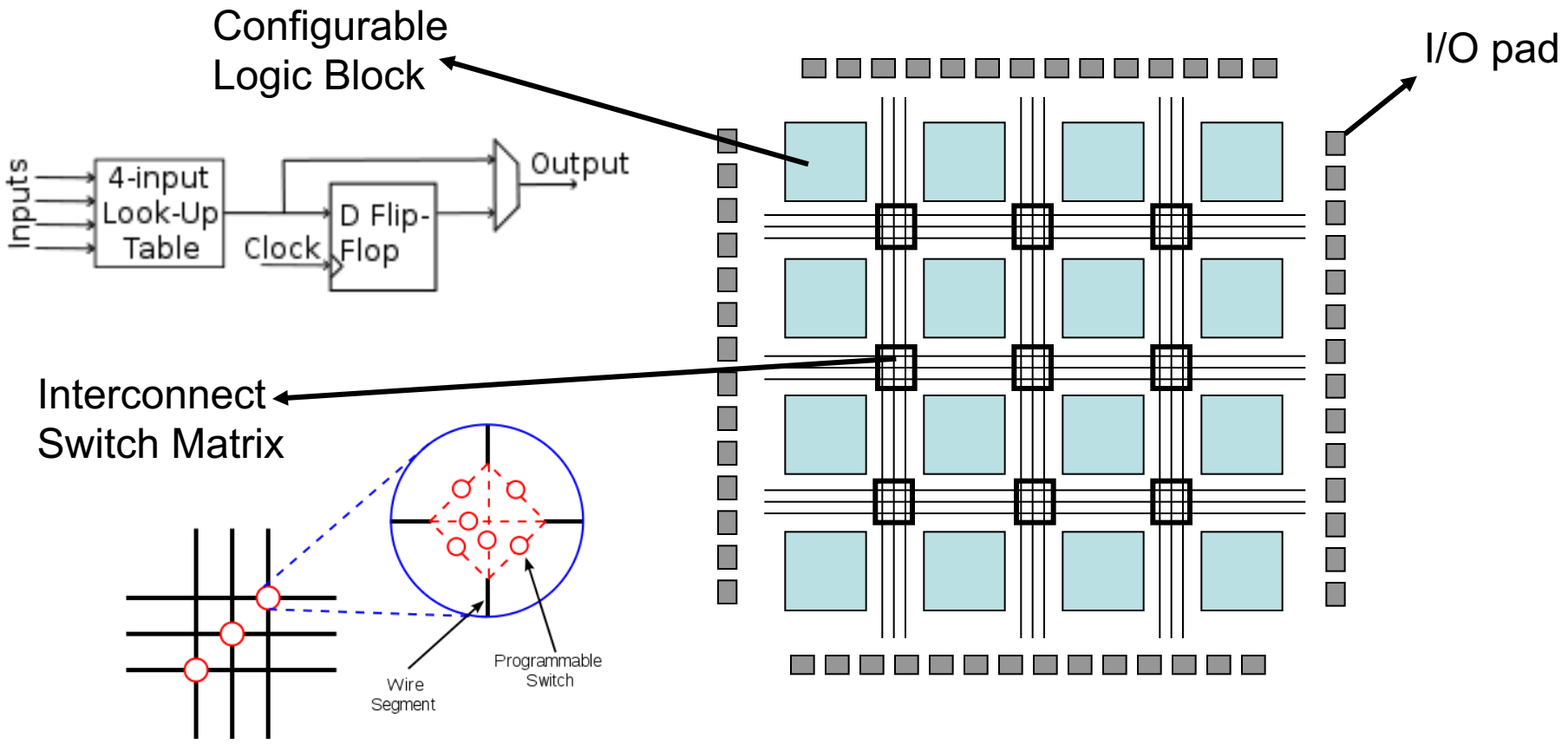
Mask-programmable Gate Array / Sea-of-gates

- Sea of uncommitted gates
  - An uncommitted cell or gate is formed by two or more N-channel and P-channel transistors
- A logic function is realized by customizing uncommitted gates using additional interconnect
- Routing may be performed through dedicated channels or over the gates (additional metal layers)



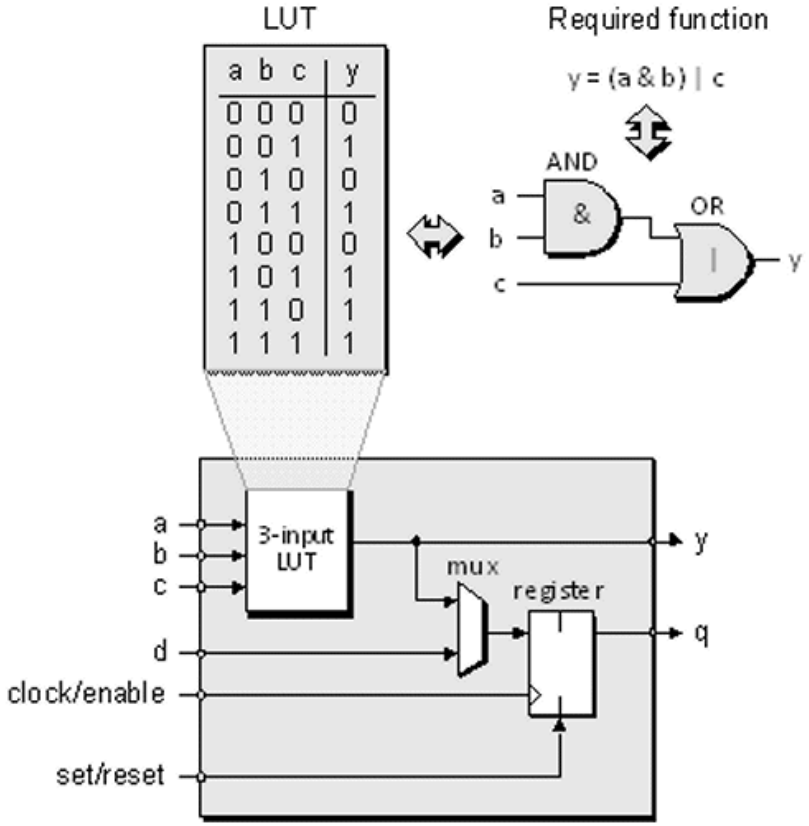
Field Programmable Gate Arrays (FPGAs)

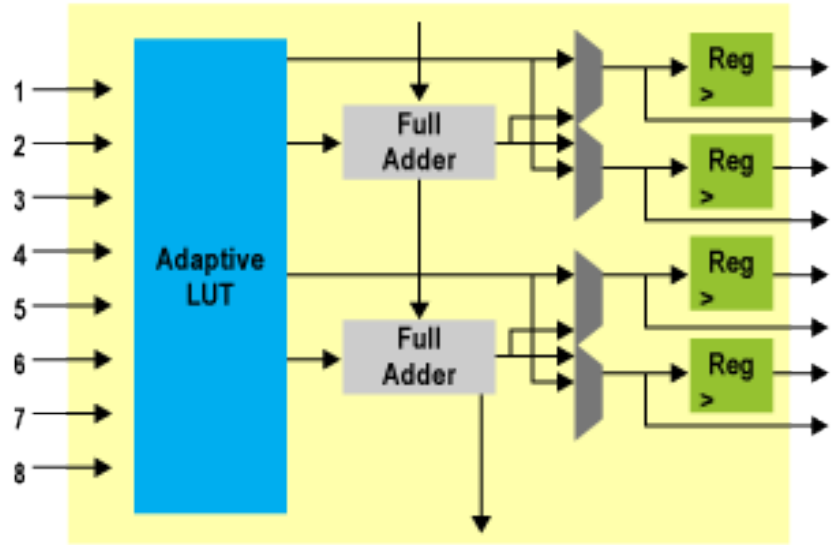
- FPGAs are integrated circuits that can be “programmed” after manufacturing
- Consist of Configurable Logic Blocks (CLBs) and Programmable Interconnect



Configurable Logic Block

- CLB contains a lookup table (LUT) that stores truth table of logic function to be implemented
- More complex CLBs contain an LUT, arithmetic circuits, flip-flop(s), and multiplexers

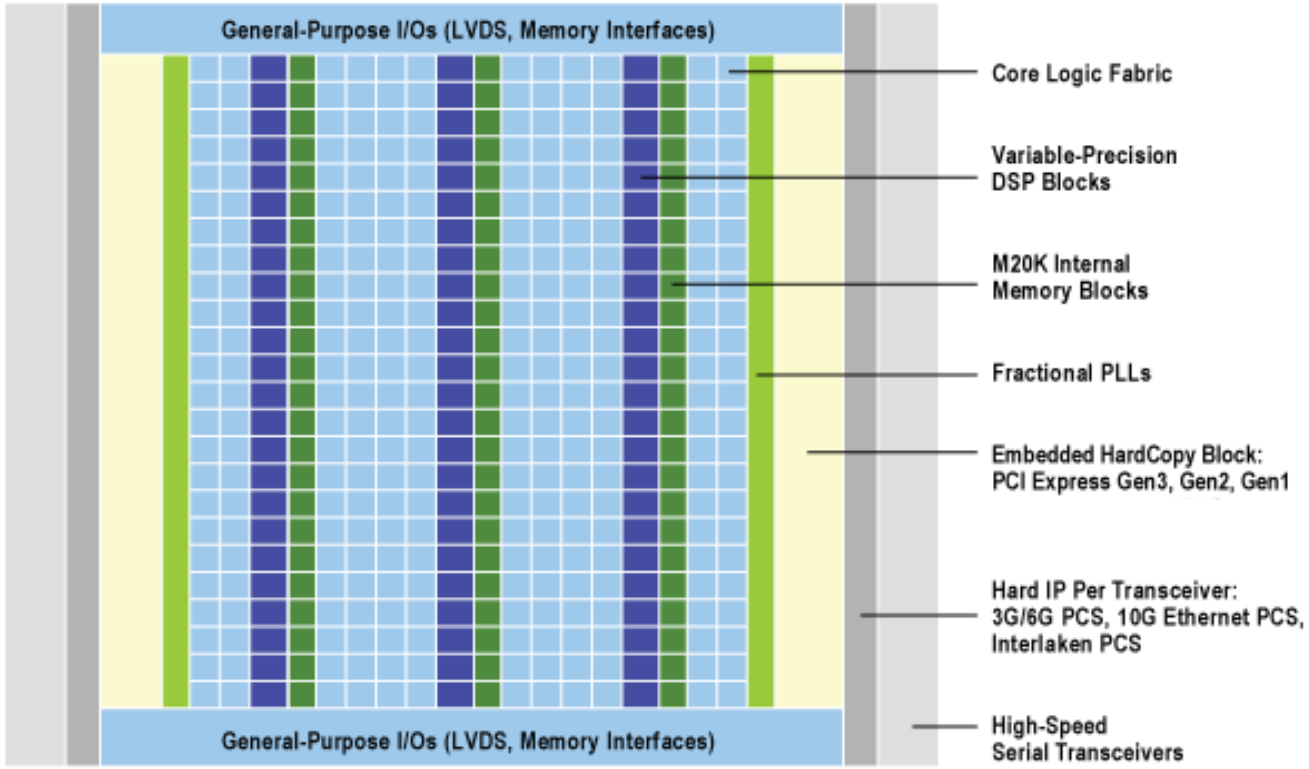




Stratix V Adaptive Logic Module (Source: Intel)

FGPA: Contemporary Architecture

- Modern FPGAs contain hardwired circuits for arithmetic, memory, I/O, and even microprocessors



Stratix V FPGA architecture (Source: Intel)

Question

- Logic level design automation tools are used for
  A. Standard Cell based designs
  B. Macro cell based designs
  C. Mask programmable gate arrays
  D. Field programmable gate arrays

Answer: E. All of the above!