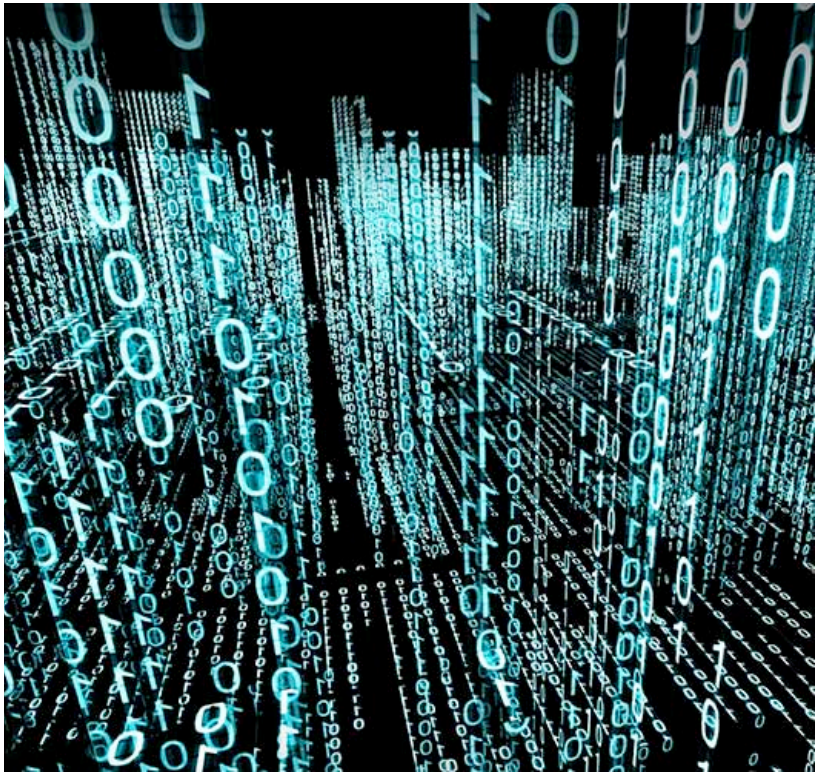


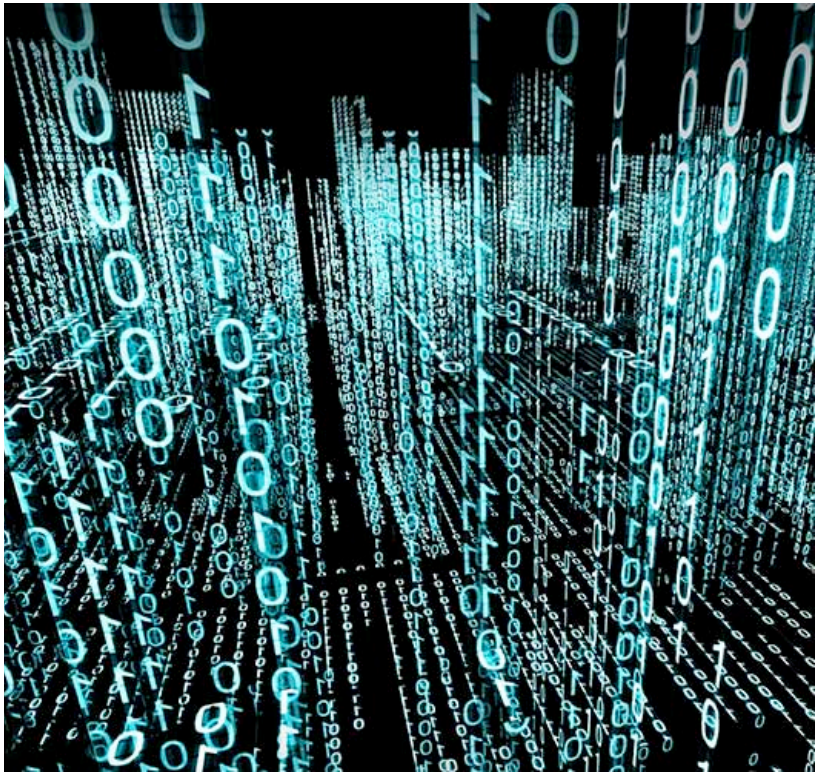
EECS 151/251A

Spring 2024

Digital Design and Integrated Circuits

Instructor:
J. Wawrzynek

Lecture 5: ASICs, FPGAs



Implementation Alternatives

ASICs

FPGAs

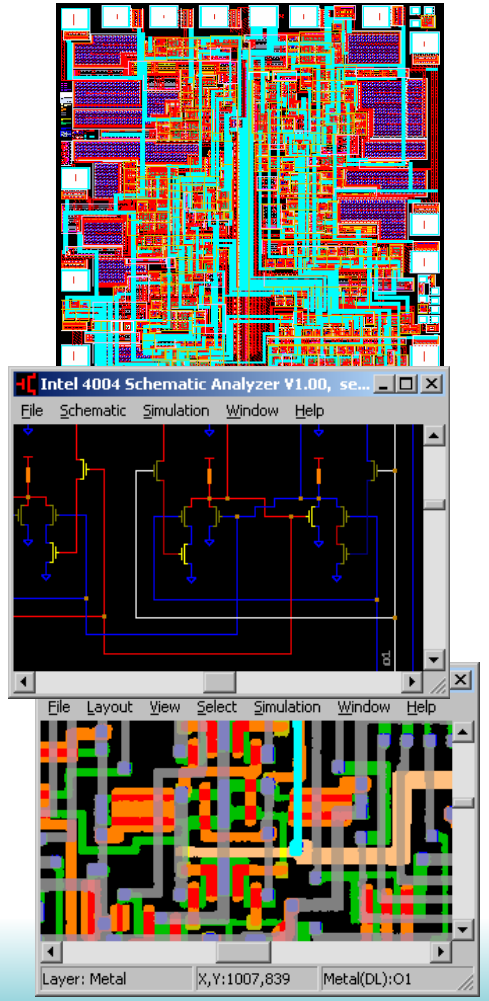
Implementation Alternative Summary

Full-custom:	All circuits/transistors layouts optimized for application.
Standard-cell (ASIC):	Small function blocks/"cells" (gates, FFs) automatically placed and routed.
Gate-array (structured ASIC):	Partially prefabricated wafers with arrays of transistors customized with metal layers or vias.
FPGA:	Prefabricated chips customized with loadable latches or fuses.
Microprocessor:	Instruction set interpreter customized through software.
Domain Specific Processor:	Special instruction set interpreters (ex: DSP, NP, GPU, TPU).

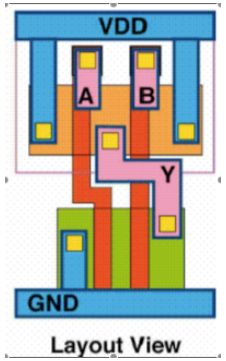
*Wh*at are the important metrics of comparison?

Full-Custom

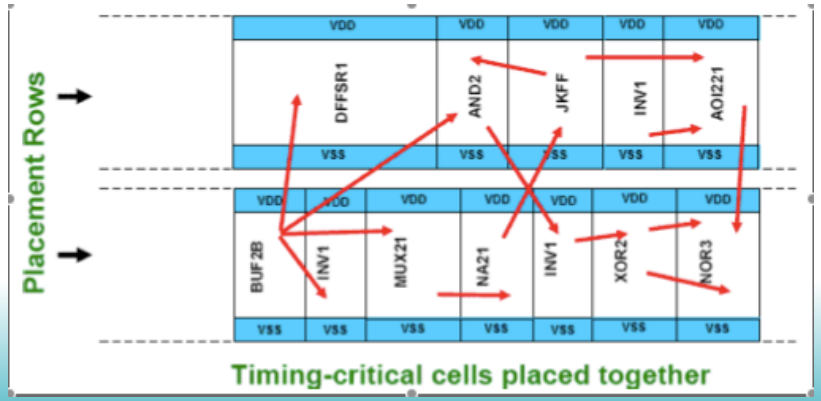
- Circuit styles and transistors are custom sized and drawn to optimize die, size, power, performance.
- High NRE (non-recurring engineering) costs
  - Time-consuming and error prone layout
- Hand-optimizing the layout can result in small die for low per unit costs, extreme-low-power, or extreme-high-performance.
- Common today for **analog design**.
- High NRE usually restricts use to highly-constrained and cost insensitive markets.

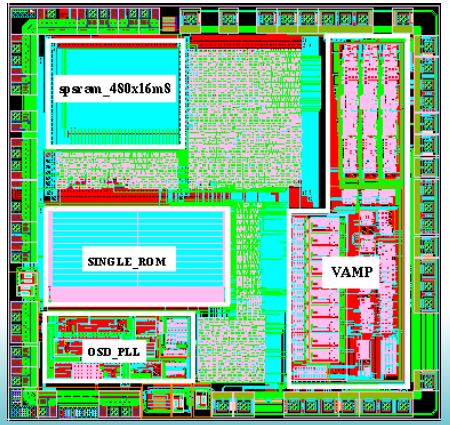


Standard-Cell ASIC Design



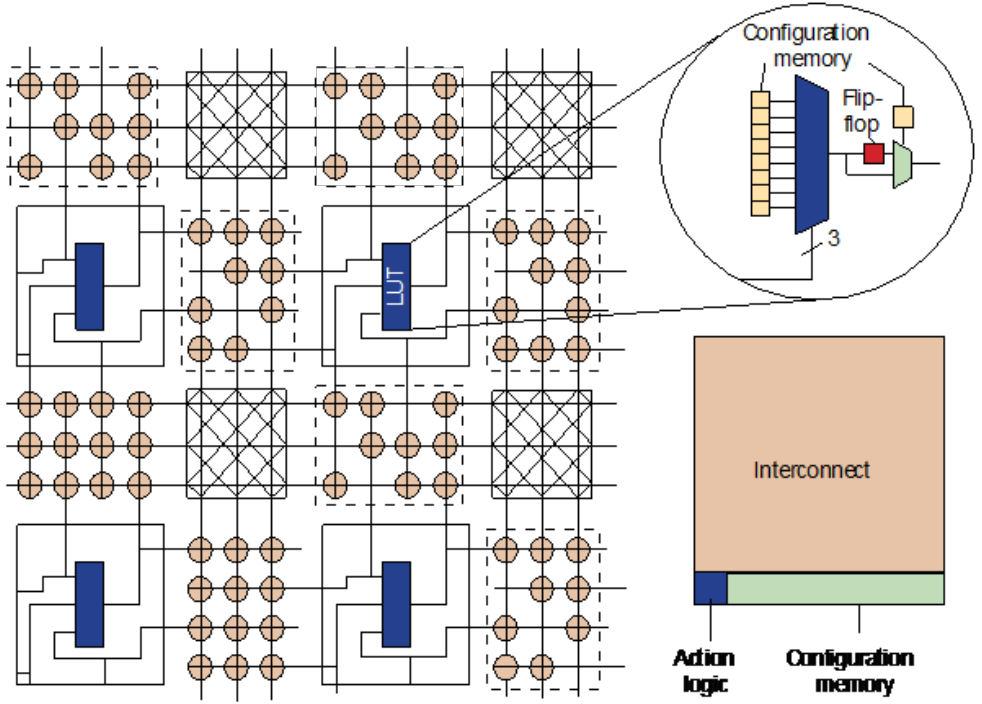
- Based around a set of pre-designed (and verified) cells
  - Ex: NANDs, NORs, Flip-Flops, counters slices, buffers, …
- Each cell comes complete with:
  - layout (perhaps for different technology nodes and processes),
  - Simulation, delay, & power models.
- Chip layout is automatic, reducing NREs (usually no hand-layout).
- (Slightly) less optimal use of area and power, leading to higher per die costs than full-custom.
- Commonly used with other predesigned blocks (large memories, I/O blocks, etc.)





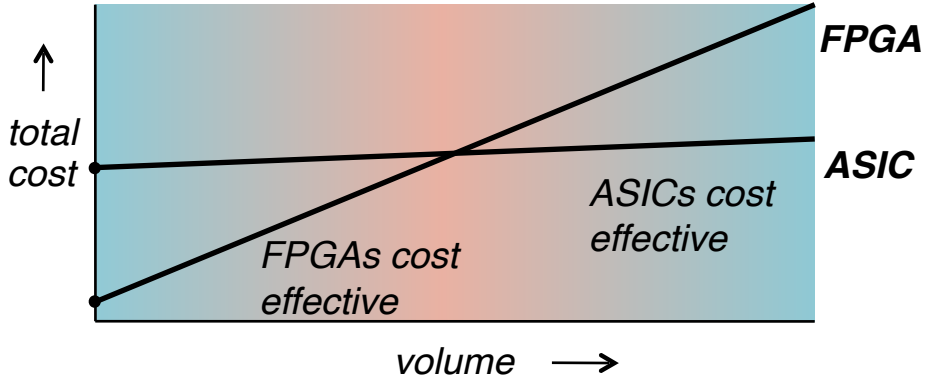
Field Programmable Gate Arrays (FPGA)

- *Two-dimensional array of simple logic- and interconnection-blocks.*
- *Typical architecture: Look-up-tables (LUTs) implement any function of n-inputs (n=3 in this case).*
- *Optional connected Flip-flop with each LUT.*



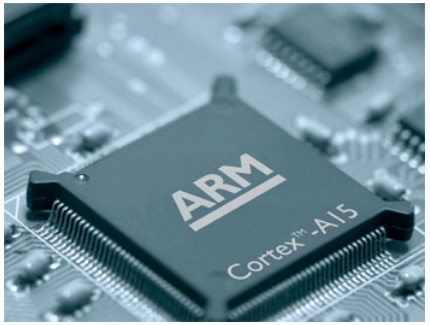
- Fuses, EPROM, or Static RAM cells are used to store the "configuration".
  - Here, it determines function implemented by LUT, selection of Flip-flop, and interconnection points.
- Many FPGAs include special circuits to accelerate adder carry-chain and many special cores: RAMs, MAC, Enet, PCI, SERDES, CPUs, ...

FPGA versus ASIC



- **ASIC:** Higher NRE costs (10's of $M). Relatively Low cost per die (10's of $ or less).
- **FPGAs:** Low NRE costs. Relatively low silicon efficiency ⇒ high cost per part (> 10's of $ to 1000's of $).
- **Cross-over volume** from cost effective FPGA design to ASIC was often in the 100K range.

Microprocessors / Microcontrollers



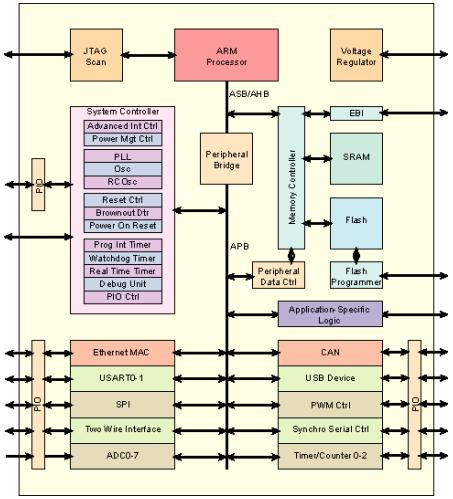
- Where relatively low performance and/or high flexibility is needed, a viable implementation alternative:
  - Software implements desired function
  - “Microcontroller”, often with built in nonvolatile program memory and used as single function.
- Furthermore, instruction set processors (microprocessors) are a ubiquitous “abstraction” level.
  - “Synthesizable” RTL model (“soft core”, available in HDL)
  - Often mixed into other digital designs
- Their implementation hosted on a variety of implementation platforms: standard-cell ASICs, FPGA, other processors?

§	Assembler
	ADD{cond}{S} Rd, Rn, <Operand2>
	ADC{cond}{S} Rd, Rn, <Operand2>
5E	QADD{cond} Rd, Rm, Rn
5E	QDADD{cond} Rd, Rm, Rn
	SUB{cond}{S} Rd, Rn, <Operand2>
	SBC{cond}{S} Rd, Rn, <Operand2>
	RSB{cond}{S} Rd, Rn, <Operand2>
	RSC{cond}{S} Rd, Rn, <Operand2>
5E	QSUB{cond} Rd, Rm, Rn
5E	QDSUB{cond} Rd, Rm, Rn
2	MUL{cond}{S} Rd, Rm, Rs
2	MLA{cond}{S} Rd, Rm, Rs, Rn
M	UMULL{cond}{S} RdLo, RdHi, Rm, Rs
M	UMLAL{cond}{S} RdLo, RdHi, Rm, Rs
6	UMAAL{cond} RdLo, RdHi, Rm, Rs

System-on-chip (SOC)

- Pre-verified block designs, standard bus interfaces (or adapters) ease integration - lower NREs, shorten TTM.

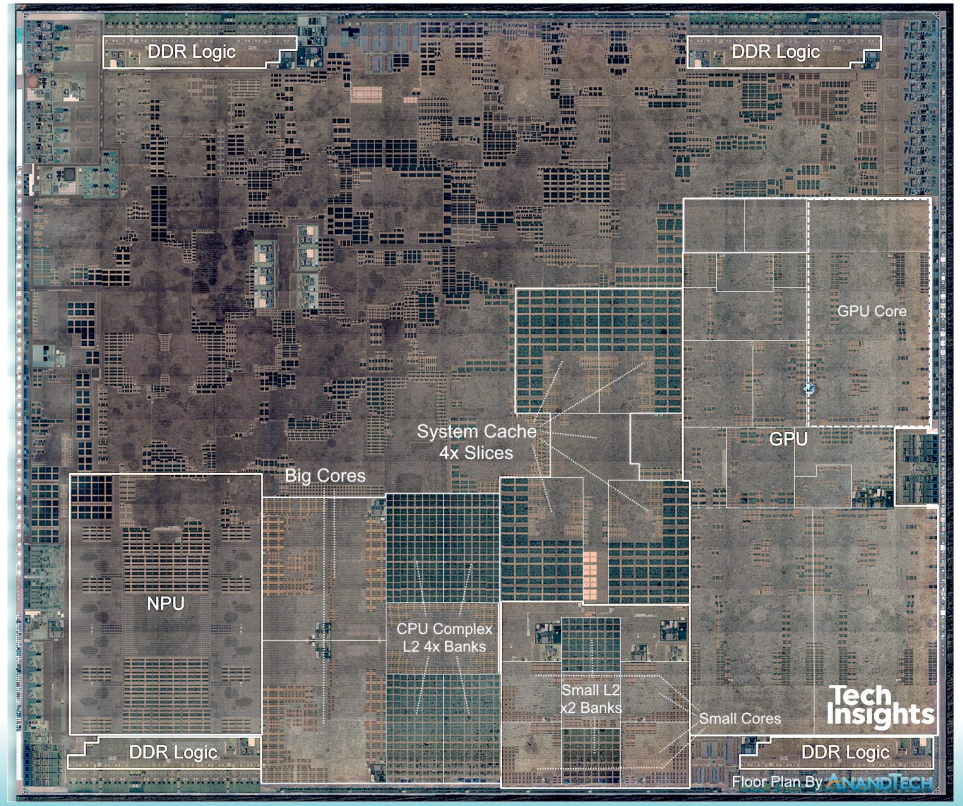
- *Brings together: standard cell blocks, custom analog blocks, processor cores, memory blocks, embedded FPGAs, …*
- *Standardized on-chip buses (or hierarchical interconnect) permit "easy" integration of many blocks.*
  – *Ex: AXI, AMBA, Sonics, …*
- *"IP Block" business model: Hard- or soft-cores available from third party designers.*
- *ARM, inc. is the shining example. Hard- and "synthesizable" RISC processors.*
- *ARM and other companies provide, Ethernet, USB controllers, analog functions, memory blocks, …*



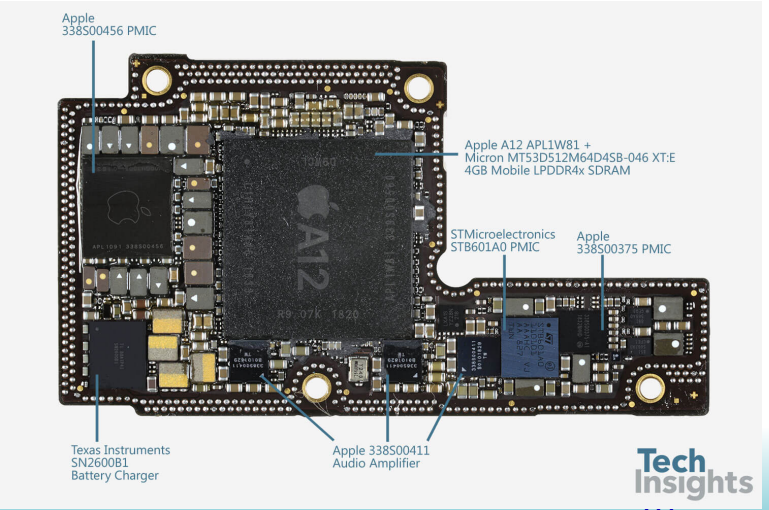
Qualcomm Snapdragon

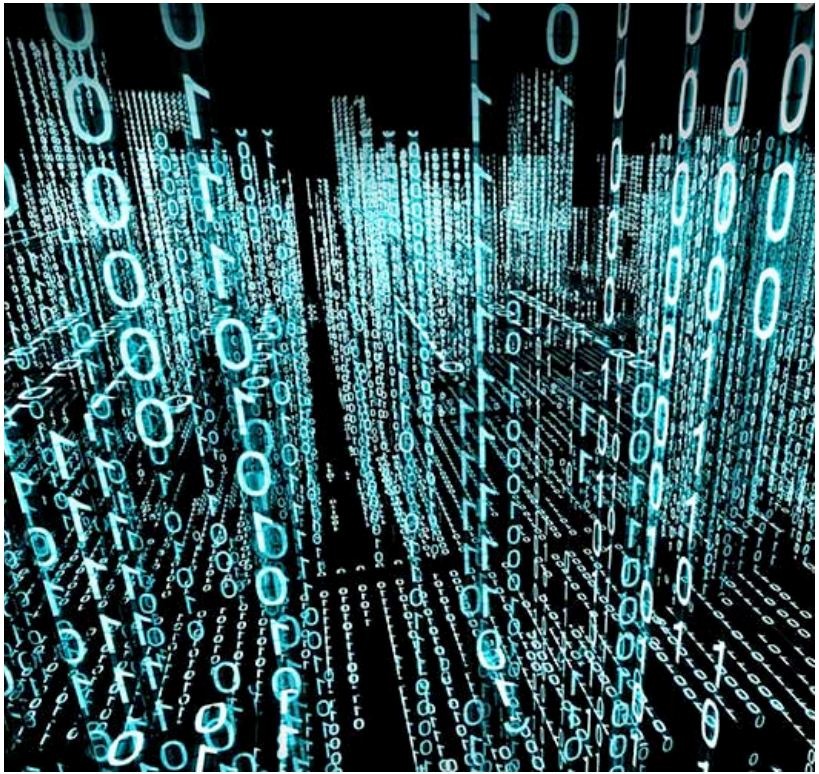
Modern(-ish) System-On-A-Chip (SOC)

- Apple A12 Bionic
  - *7nm CMOS, Up to 2.49GHz*



- *2x Large CPUs*
- *4x Small CPUs*
- *GPUs*
- *Neural processing unit (NPU)*
- *Lots of memory*
- *DDR memory interfaces*





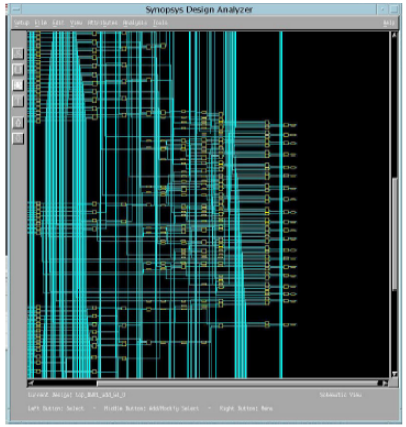
ASICs

Verilog to ASIC layout flow

- "push-button" approach

```
module adder64 (a, b, sum);
  input  [63:0] a, b;
  output [63:0] sum;

  assign sum = a + b;
endmodule
```

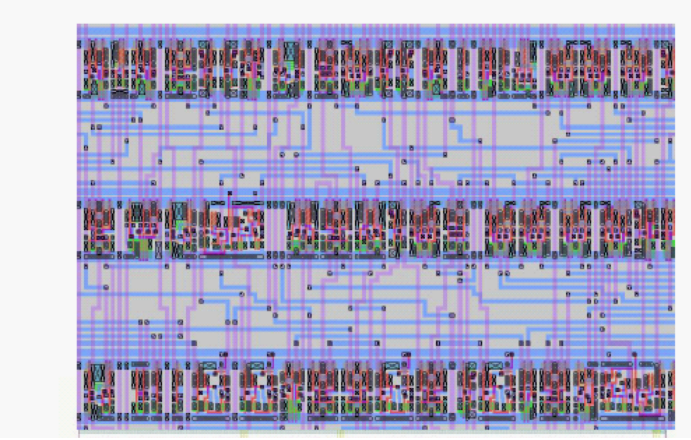


After Synthesis

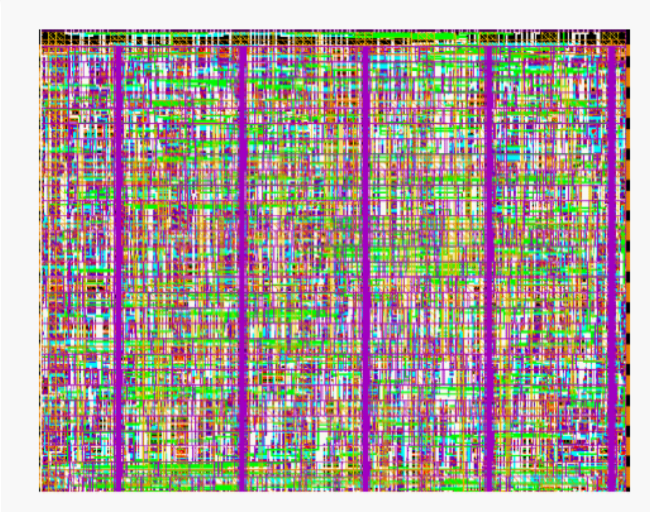
After Placement

After Routing

Standard cell layout methodology



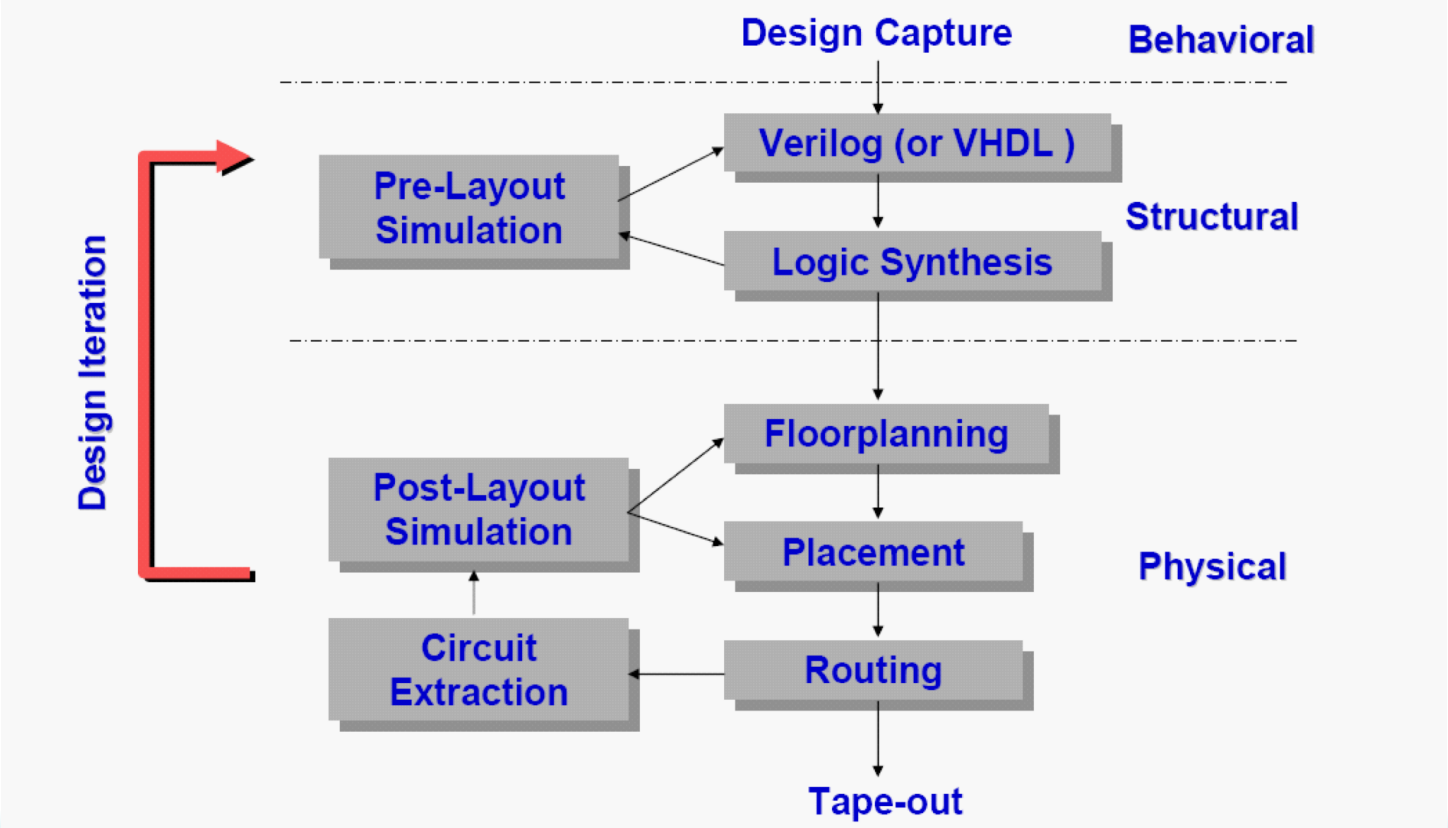
1um, 2-metal process



Modern sub-100nm process
"Transistors are free things that fit under wires"

- With limited # metal layers, dedicated routing channels were needed
- Now, many layers and wires routed over cells. Currently area often dominated by wires

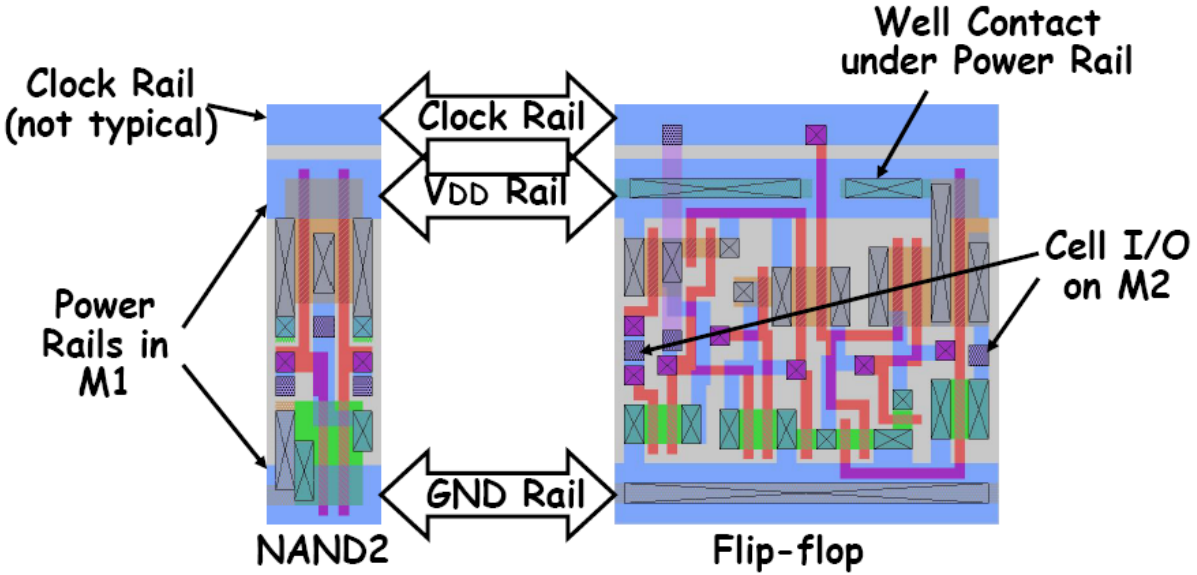
The ASIC flow



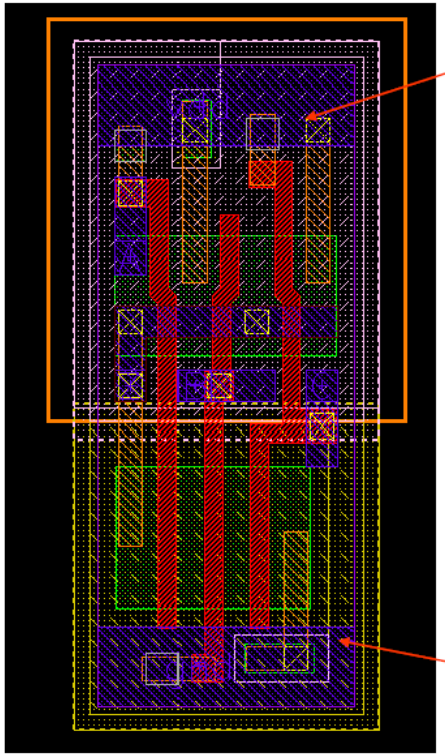
Standard cell design

Cells have standard height but vary in width

Designed to connect power, ground, and wells by abutment



Standard cell characterization



Power Supply Line (V_{DD})

Delay in (ns)!!

Path	1.2V - 125°C	1.6V - 40°C
$In1—t_{pLH}$	$0.073+7.98C+0.317T$	$0.020+2.73C+0.253T$
$In1—t_{pHL}$	$0.069+8.43C+0.364T$	$0.018+2.14C+0.292T$
$In2—t_{pLH}$	$0.101+7.97C+0.318T$	$0.026+2.38C+0.255T$
$In2—t_{pHL}$	$0.097+8.42C+0.325T$	$0.023+2.14C+0.269T$
$In3—t_{pLH}$	$0.120+8.00C+0.318T$	$0.031+2.37C+0.258T$
$In3—t_{pHL}$	$0.110+8.41C+0.280T$	$0.027+2.15C+0.223T$

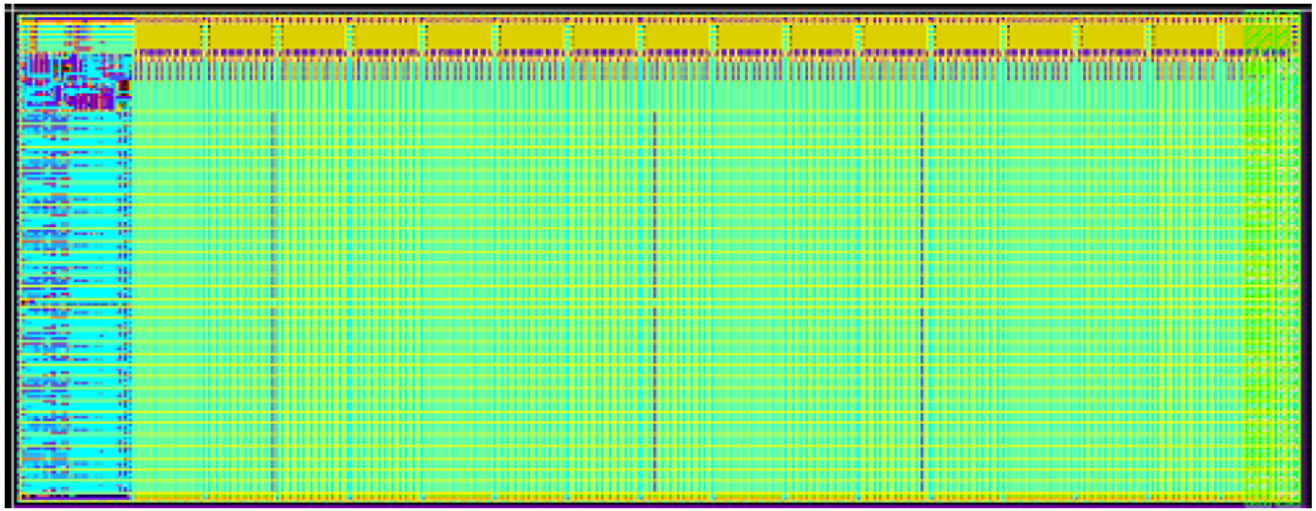
3-input NAND cell
(from ST Microelectronics):
C = Load capacitance
T = input rise/fall time

Ground Supply Line (GND)

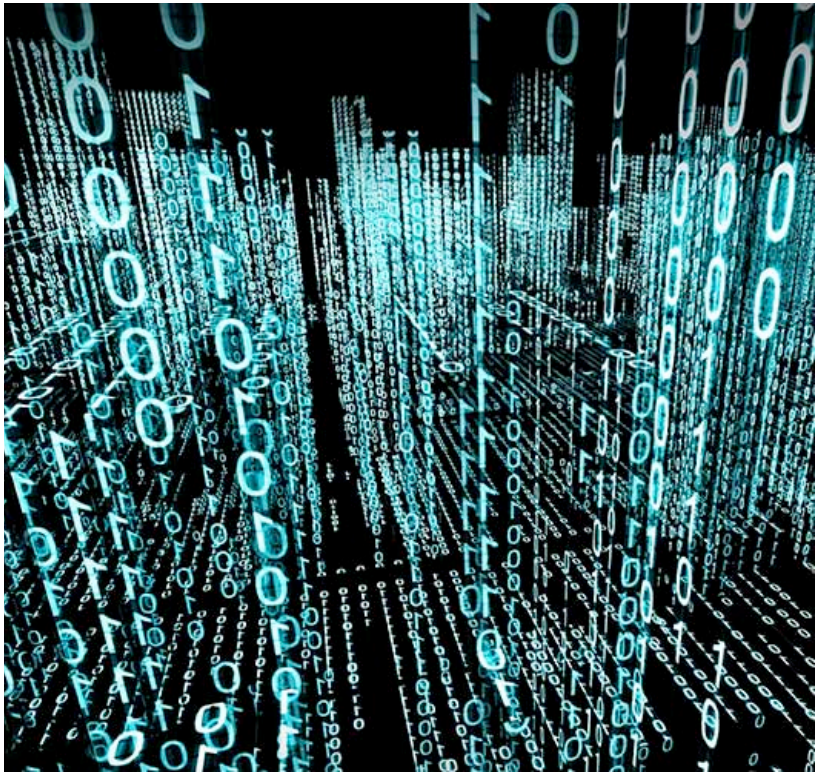
- **Each library cell (FF, NAND, NOR, INV, etc.) and the variations on size (strength of the gate) is fully characterized across temperature, loading, etc.**

“Macro” modules / cells

256×32 (or 8192 bit) SRAM Generated by hard-macro module generator



- Generate highly regular structures (entire memories, multipliers, etc.) with a few lines of code
- Verilog models for memories automatically generated based on size

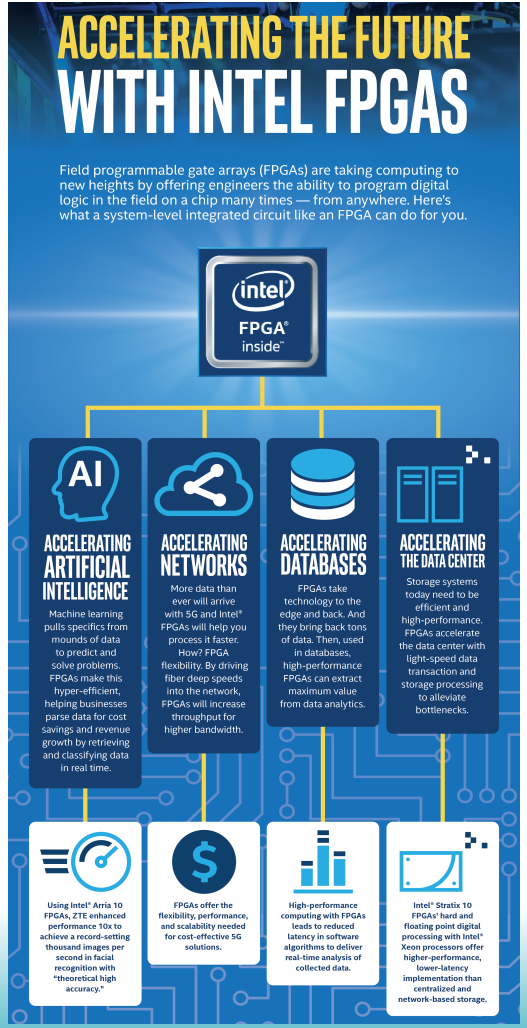


FPGAs

FPGAs are in widespread use

Far more different designs are implemented in FPGAs than in custom chips.

And in the Data-Center



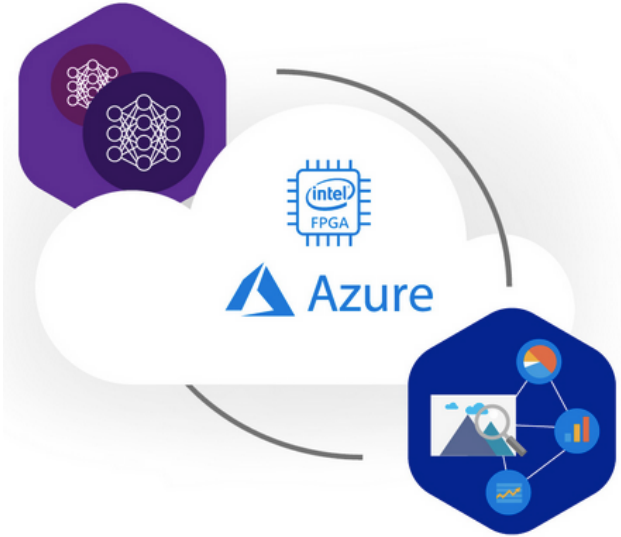
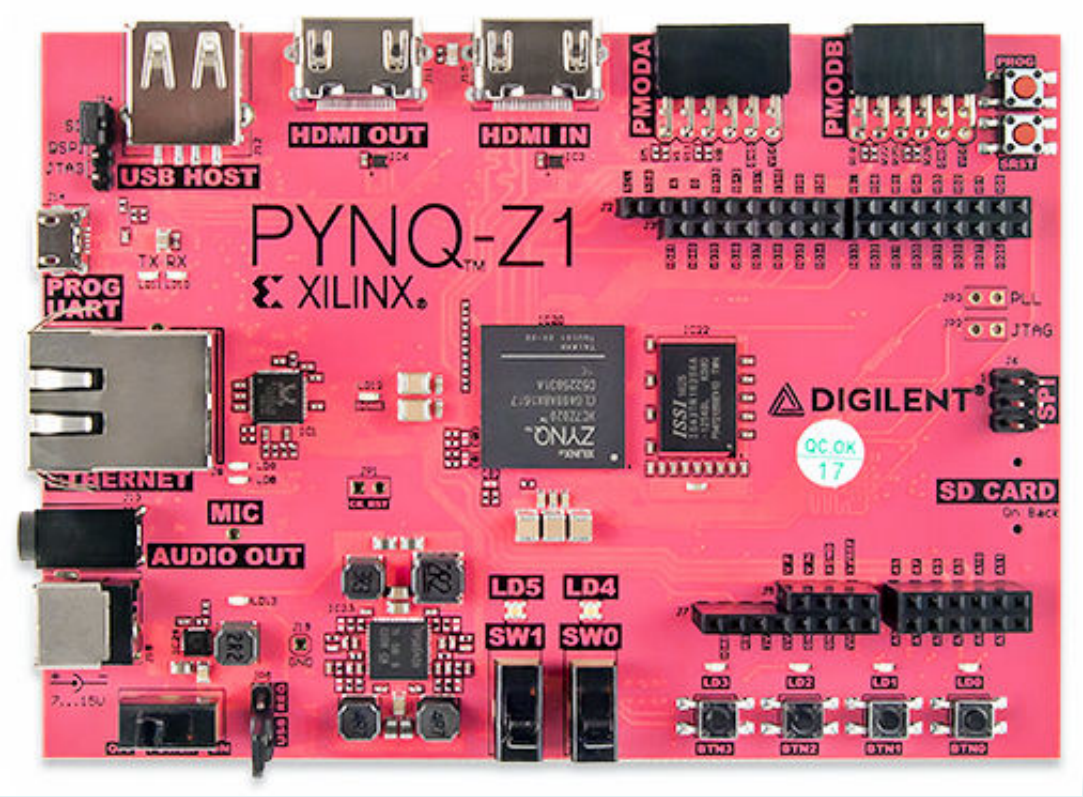
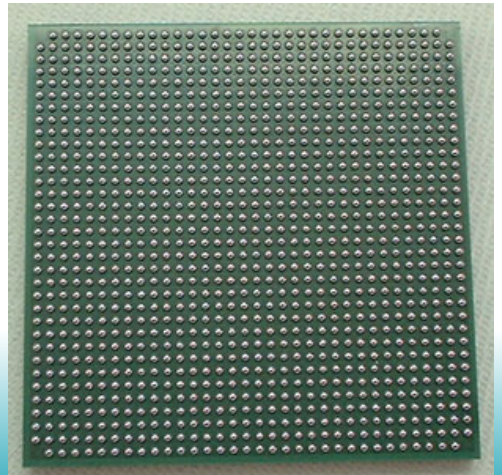


Image credit: Copyright © 2018 Microsoft Corporation. All rights reserved.

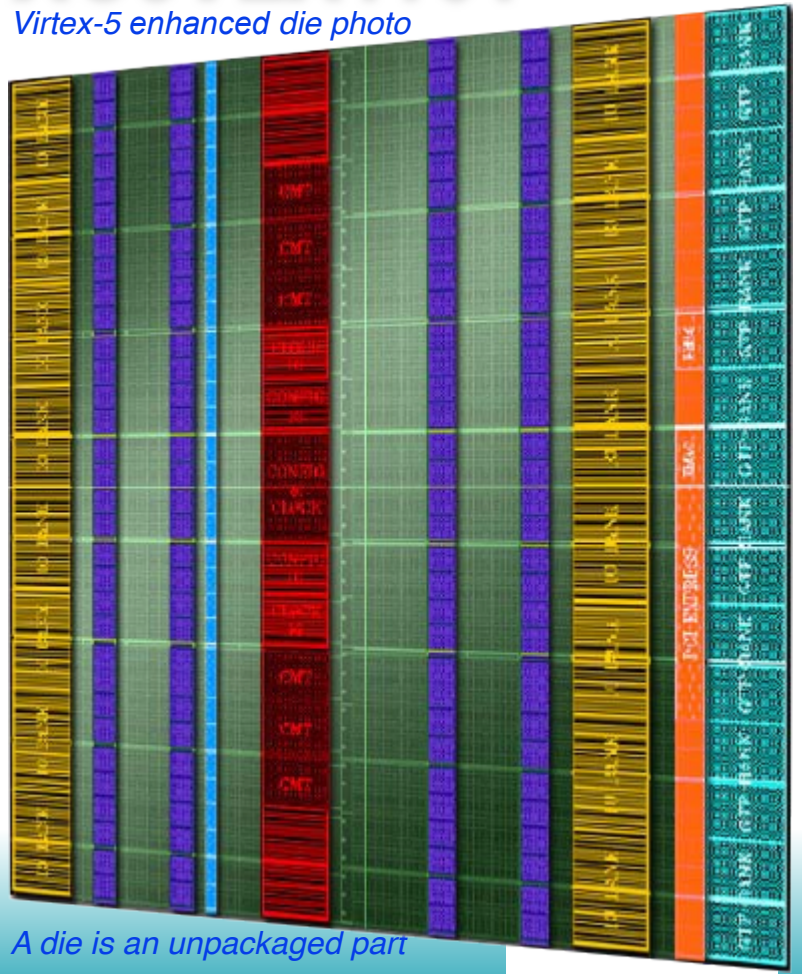
EECS151/251A FPGA Lab Board



FPGA: Xilinx Virtex-5 XC5VLX110T



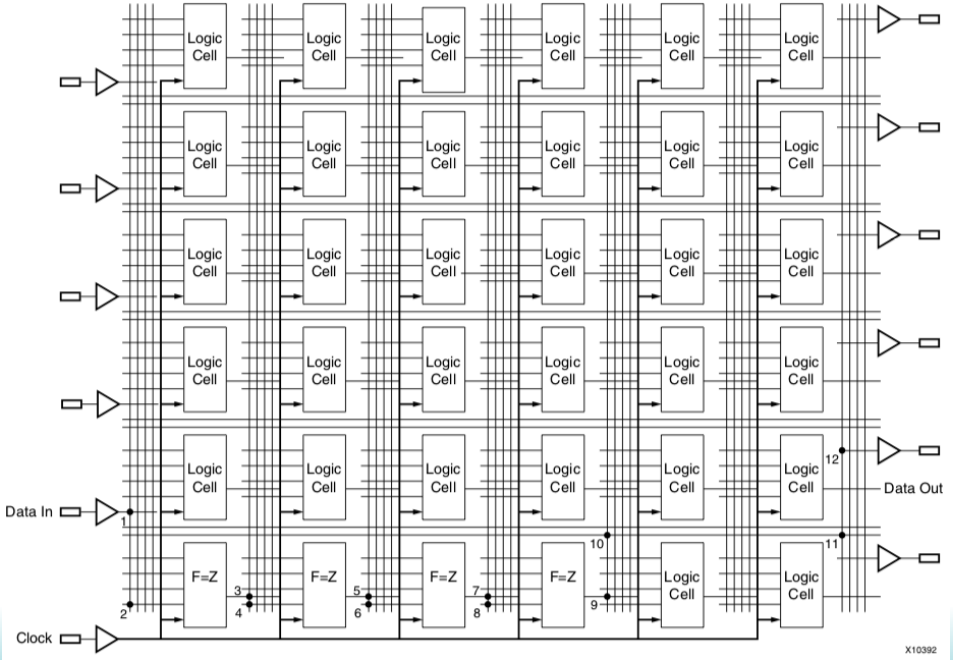
Virtex-5 enhanced die photo



A die is an unpackaged part

FPGA Overview

- Basic structure: two-dimensional array of logic blocks and flip-flops with a means for the user to configure (program):
    1. the interconnection between the logic blocks,
    2. the function of each block.



Simplified version of FPGA internal architecture

Why are FPGAs Interesting?

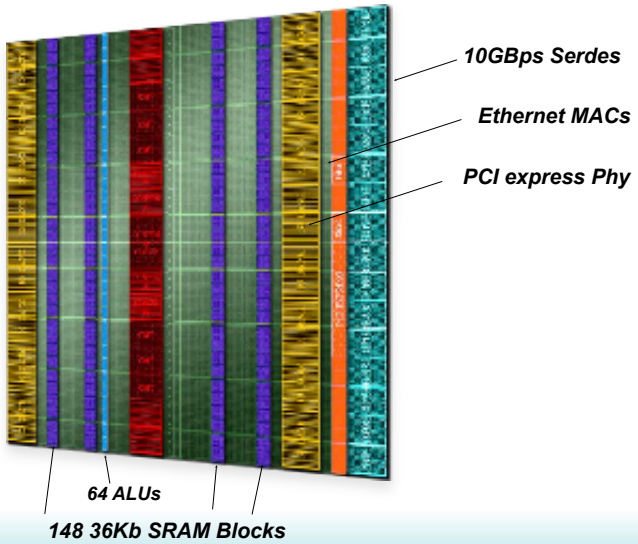
- Technical viewpoint:
  - For hardware/system-designers, like ASICs - only better: "Tape-out" new design every few minutes/hours.
  - "reconfigurability" or "reprogrammability" may offer other advantages over fixed logic?
    - In-field reprogramming? Dynamic reconfiguration? Self-modifying hardware, evolvable hardware?

Of course, the higher flexibility comes at the expense of larger die area, slower circuits, and more energy per operation.

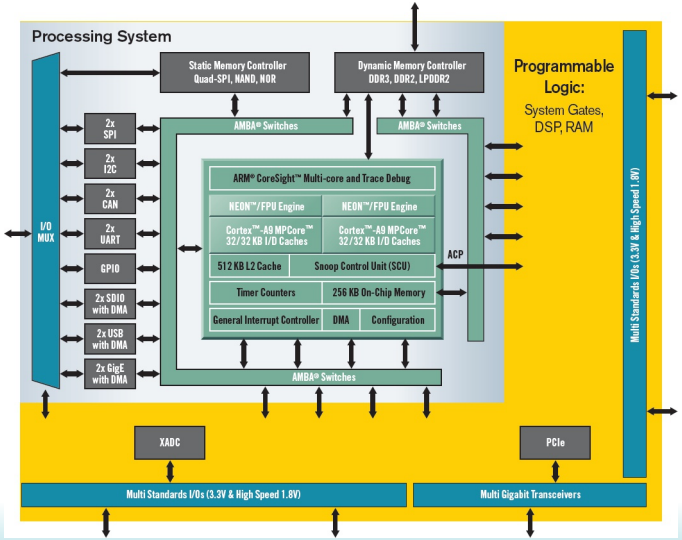
Why are FPGAs Interesting?

- Logic capacity now only part of the story: on-chip RAM, high-speed I/Os, “hard” function blocks, ...
- Modern FPGAs are “reconfigurable systems on a chip”

Xilinx Virtex-5 LX110T

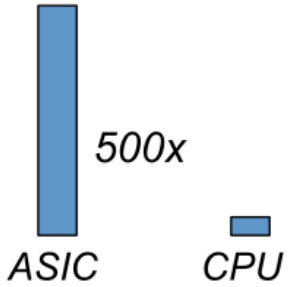


Xilinx ZYNQ - embedded ARM cores

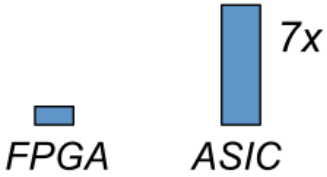


Energy Efficiency of CPU versus ASIC versus FPGA

Rehan Hameed, Wajahat Qadeer, Megan Wachs, Omid Azizi, Alex Solomatnikov, Benjamin C. Lee, Stephen Richardson, Christos Kozyrakis, and Mark Horowitz. Understanding sources of inefficiency in general-purpose chips. SIGARCH Comput. Archit. News, 38:37–47, June 2010.

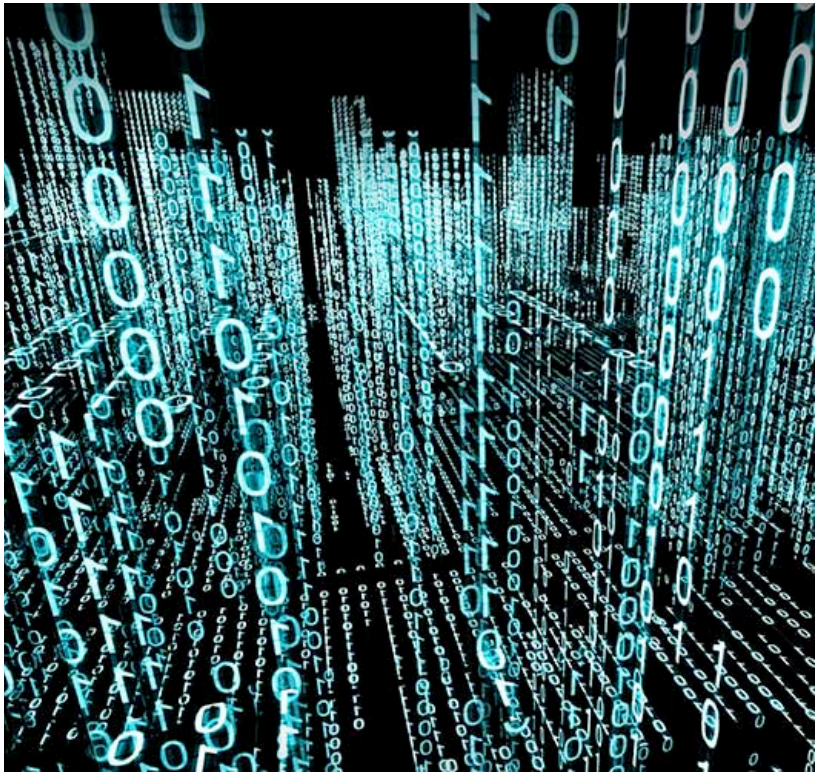


Ian Kuon and Jonathan Rose. Measuring the gap between fpgas and asics. In Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays, FPGA '06, pages 21–30, New York, NY, USA, 2006. ACM



$\therefore$ *FPGA : CPU = 70x*

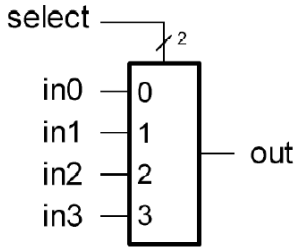
Similar story for performance efficiency



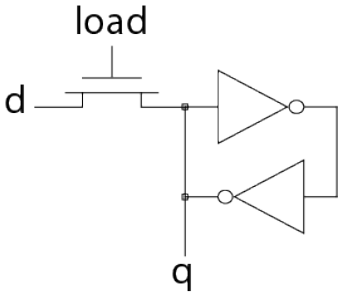
FPGA Internals

Background for upcoming technical details

Review: mux or multiplexor is a combinational logic circuit that chooses between 2^N inputs under the control of N control signals.

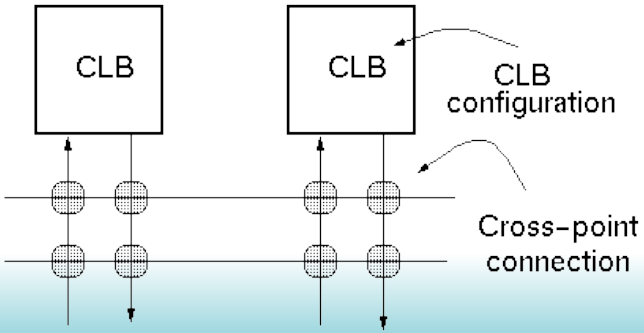


A latch is a 1-bit memory (similar to a but "level sensitive" not edge-triggered, closer to an SRAM storage cell).



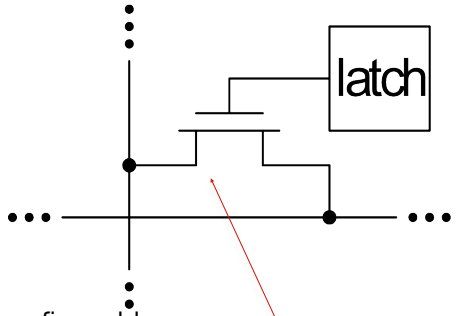
FPGA Programmability

- FPGA programmability allows users to:
  1. define function of configurable logic blocks (CLBs),
  2. establish interconnection paths between CLBs
  3. set other options, such as clock, reset connections, and I/O.
- Most FPGAs have **"SRAM based"** programmability.



Programmable Cross-points

- Latch-based (Xilinx, Intel/Altera, ...)

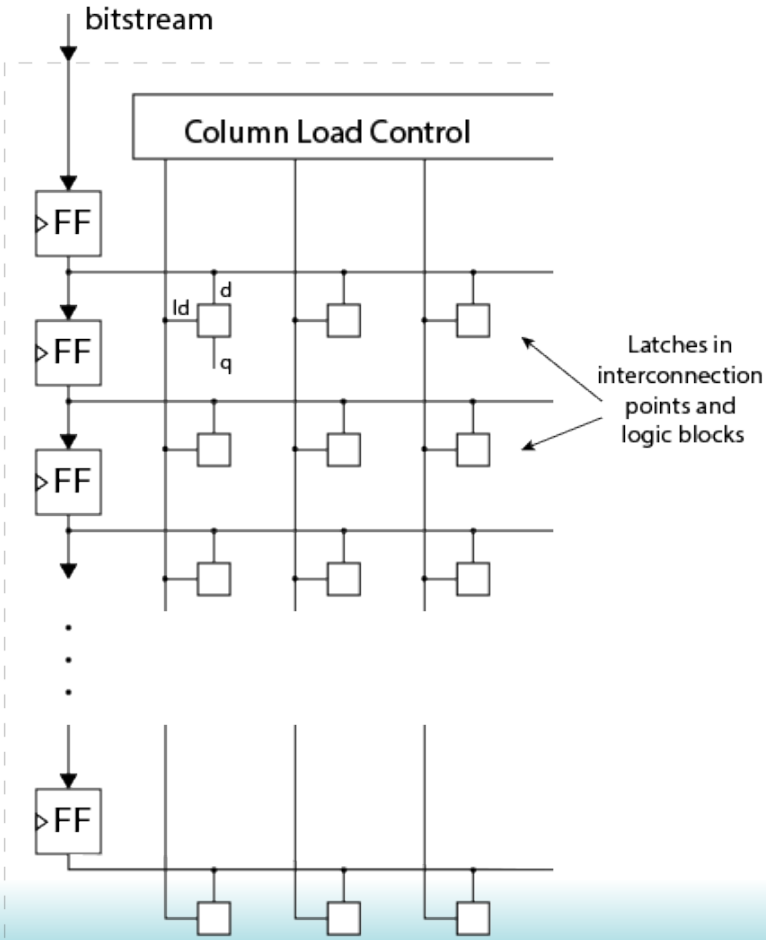


MOSFET used as a "switch"

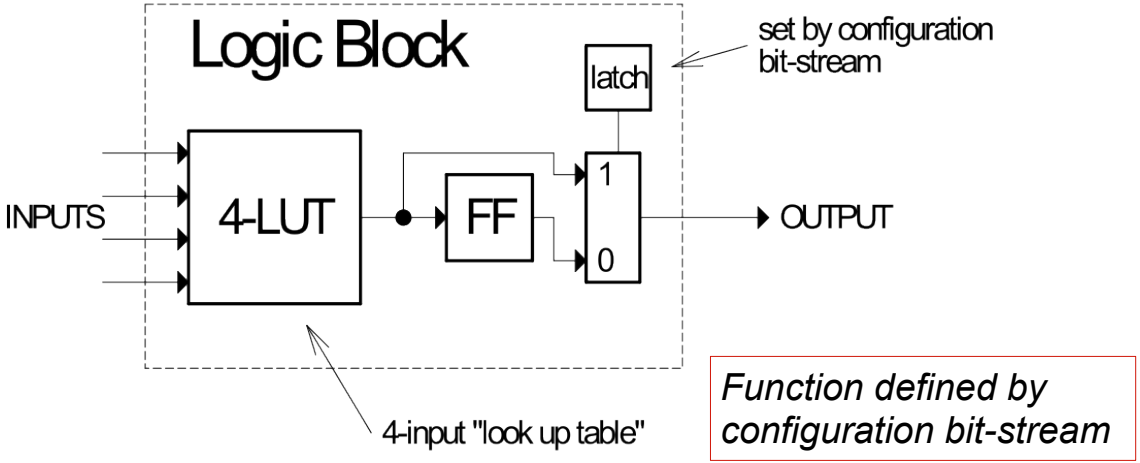
\+ reconfigurable
– volatile
– relatively large.

User Programmability

- **Latches store the configuration.**
- **Configuration *bitstream* is loaded under user control.**
- "partial reconfiguration": a selective part of the array can be reprogrammed without disturbing the other parts.
- Dynamic / runtime reconfiguration: reprogramming during a computation.
- **Most commonly the entire device is programmed when the system is booted.**



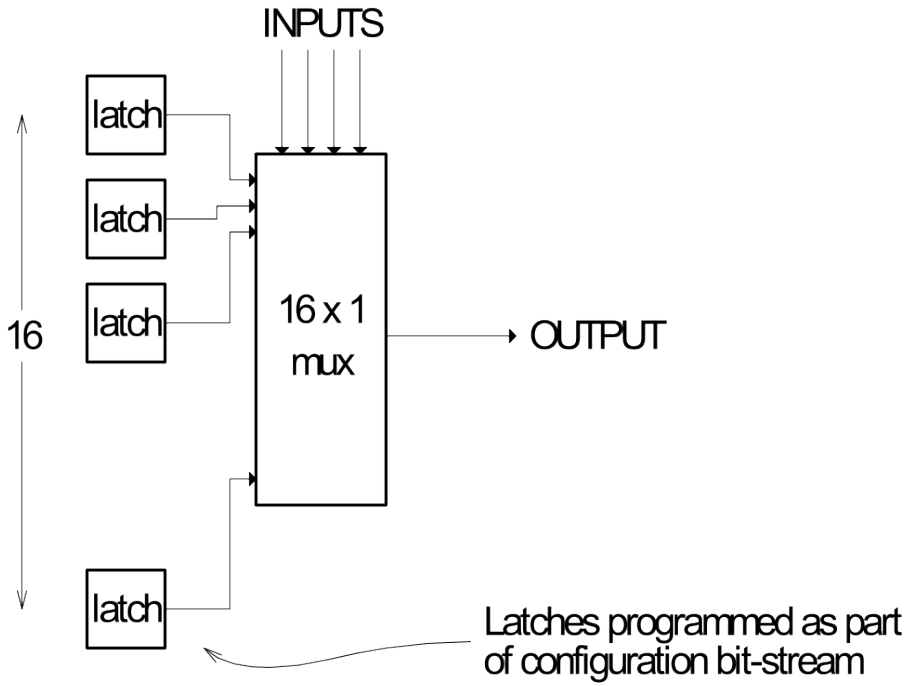
Simplified FPGA Logic Block



- Look up table (LUT)
  - implements any combinational logic function
- Register (Flip-flop)
  - optionally stores output of LUT

4-LUT Implementation

- LUT size named by number of inputs
- n-bit LUT is implemented as a 2^n x 1 memory:
  - inputs choose one of 2^n memory locations.
  - memory locations (latches) are loaded with values from user's configuration bit stream.
  - Inputs to mux control are the LUT inputs.
- Result is a general purpose "logic gate".
  - n-LUT can implement any function of n inputs!



LUT as general logic gate

- An n-LUT is a direct implementation of a function truth-table.
- Each latch location holds the value of the function corresponding to one input combination.

Example: 2-input functions

INPUTS	AND	OR
00	0	0
01	0	1
10	0	1
11	1	1

A 2-lut Implements any function of 2 inputs.

How many of these are there?

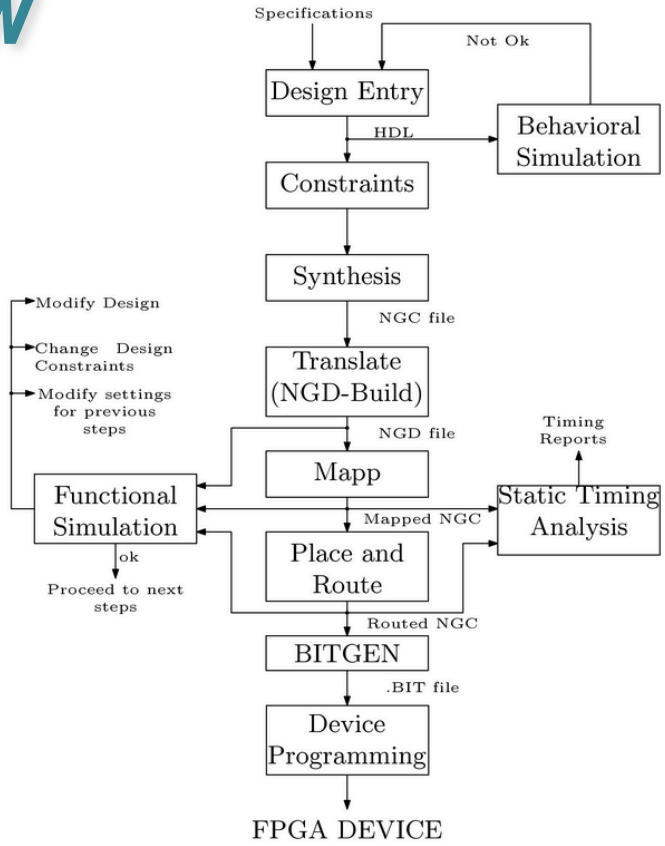
How many functions of n inputs?

Example: 4-lut

INPUTS	
0000	F(0,0,0,0) ← store in 1st latch
0001	F(0,0,0,1) ← store in 2nd latch
0010	F(0,0,1,0) ←
0011	F(0,0,1,1) ←
0011	
0100	•
0101	•
0110	•
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

FPGA Generic Design Flow

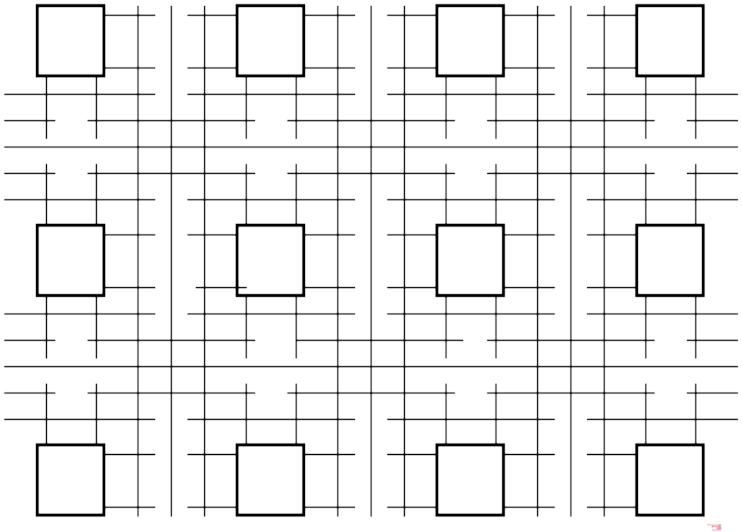
- Design Entry:
  - HDL (hardware description languages: Verilog, VHDL)
- Design Implementation:
  - Logic synthesis (in case of using HDL entry) followed by,
  - Partition, place, and route to create configuration bit-stream file
- Design verification:
  - Optionally use simulator to check function,
  - Load design onto FPGA device (cable connects PC to development board), optional “logic scope” on FPGA
  - check operation at full speed in real environment.

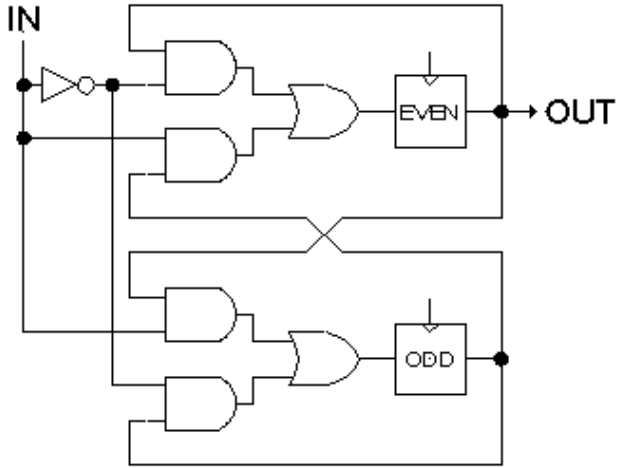


Example Partition, Placement, and Route

- Example Circuit:
  - collection of gates and flip-flops

- Simplified FPGA structure:





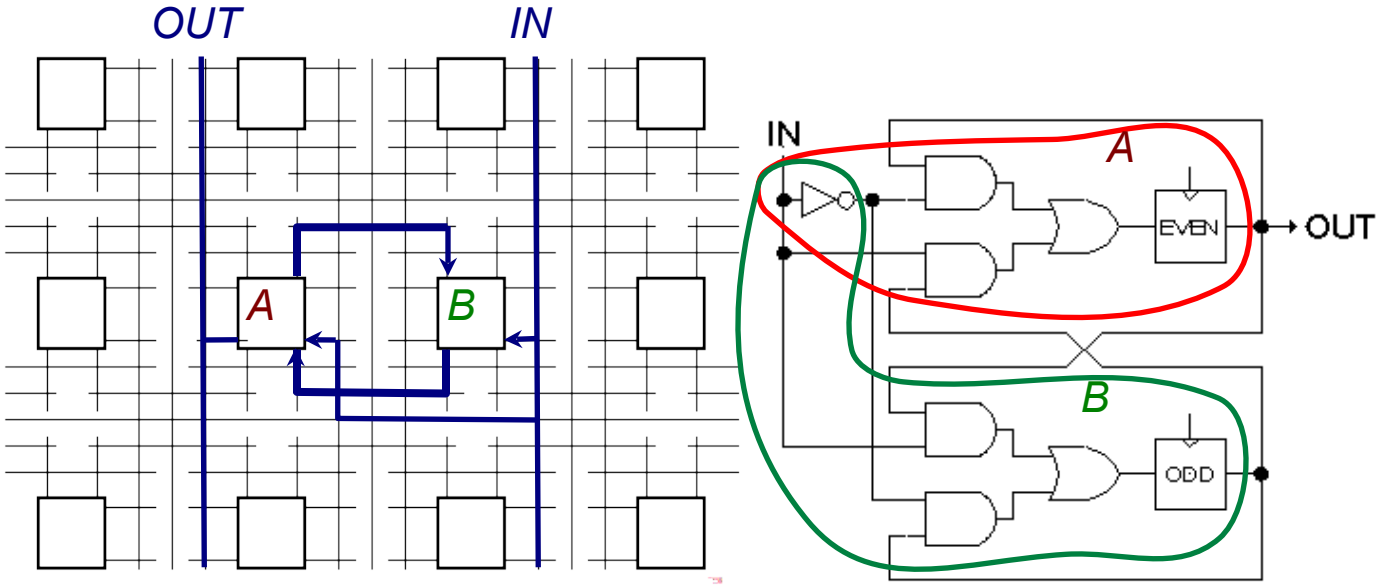
Circuit combinational logic must be "covered" by 4-input 1-output LUTs.

Flip-flops from circuit must map to FPGA flip-flops.
(Best to preserve "closeness" to CL to minimize wiring.)

Best placement in general attempts to minimize wiring.

Vdd, GND, clock, and global resets are all "prewired".

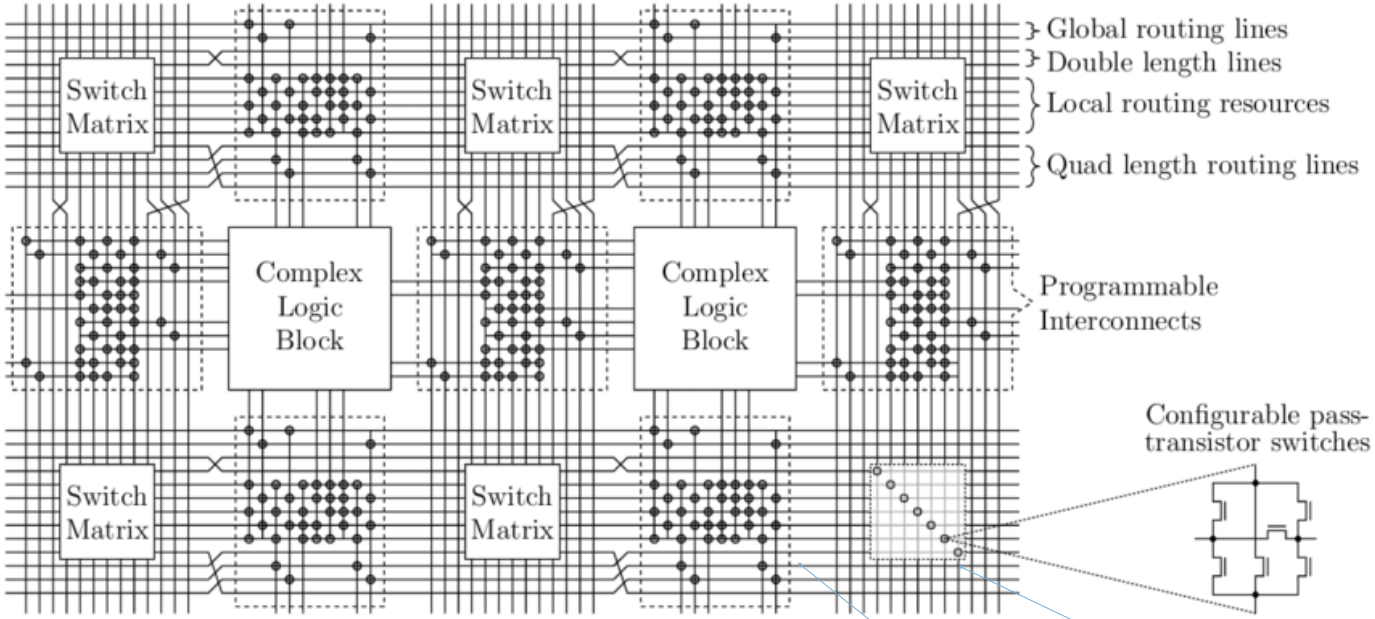
Example Partition, Placement, and Route



Two partitions. Each has single output, no more than 4 inputs, and no more than 1 flip-flop. In this case, inverter goes in both partitions.

Note: (with 4-LUTs) the partition can be arbitrarily large as long as it has not more than 4 inputs and 1 output, and no more than 1 flip-flop.

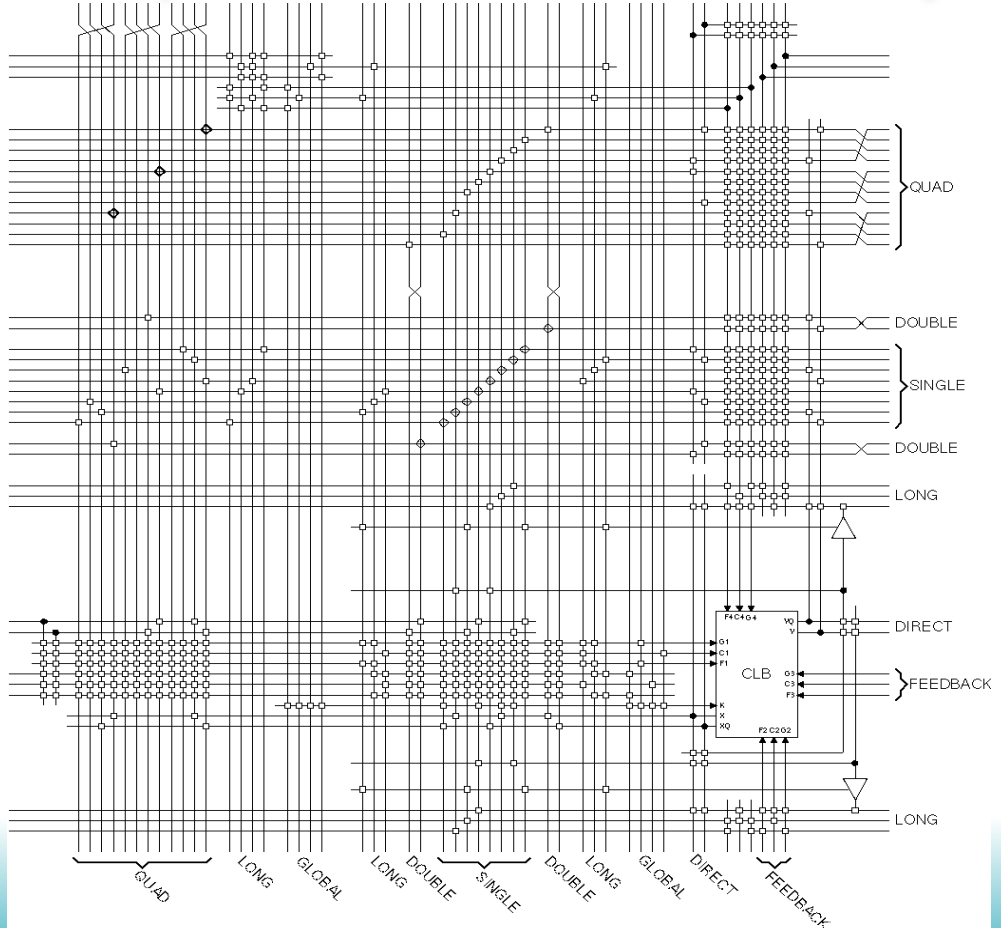
Configurable Interconnect



- Design Challenges (topology):
  - traversing long wires incurs delay and energy
  - switches (transistors) add significant delay
  - Mapping time

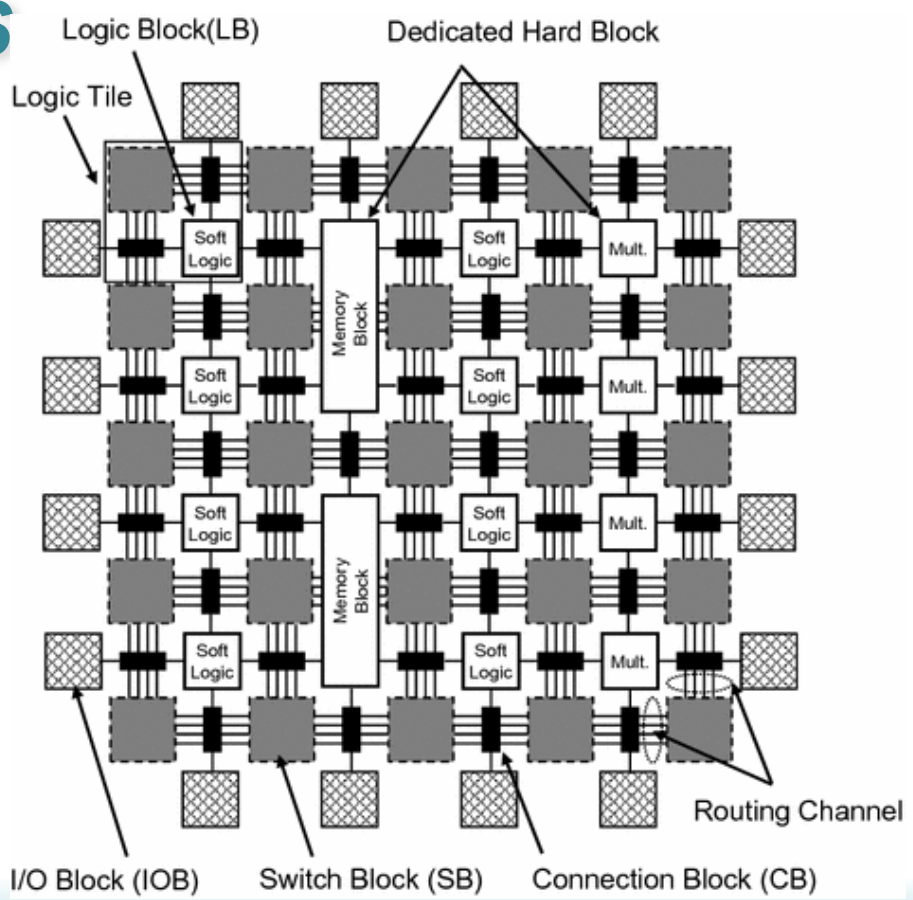
switch matrix could be more richly populated
"connection block"

Xilinx FPGAs (interconnect detail)



Embedded Hard Blocks

- Many important functions are not efficient when implemented in the reconfigurable fabric:
  - multiplication, large memory, …
- Dedicated blocks take relatively little area and therefore could go unused.



Colors represent different types of resources:

Logic

Block RAM

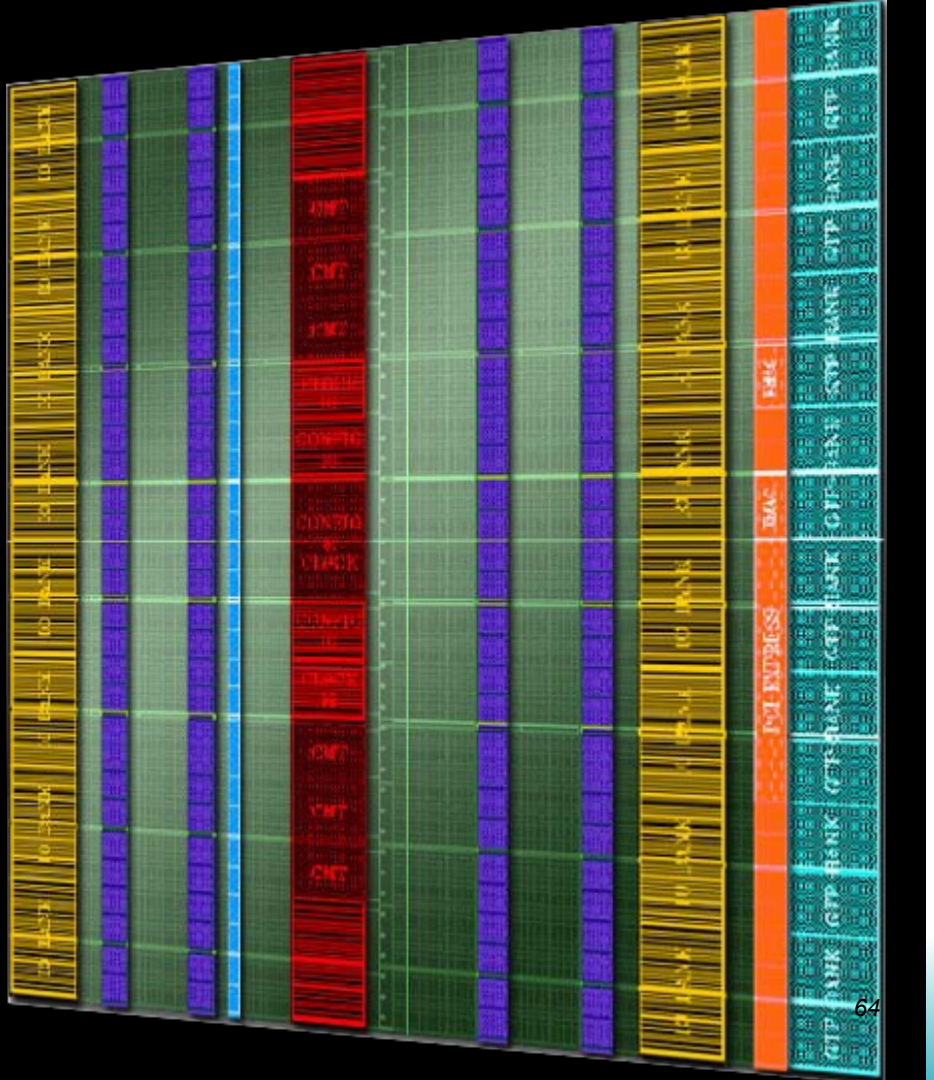
DSP (ALUs)

Clocking

I/O

Serial I/O + PCI

A routing fabric runs throughout the chip to wire everything together.



64

Virtex 6-LUTs: Composition of 5-LUTs

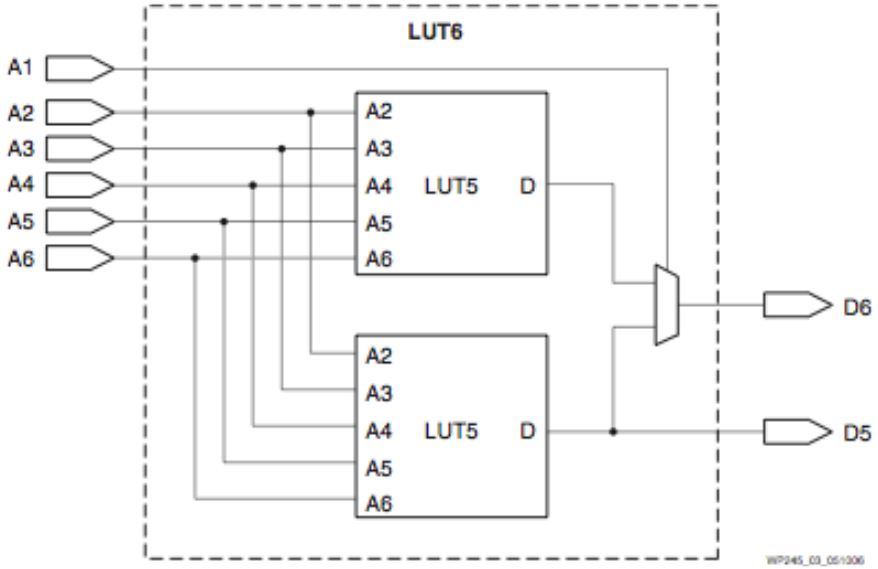


Figure 3: **Block Diagram of a Virtex-5 6-Input LUT**

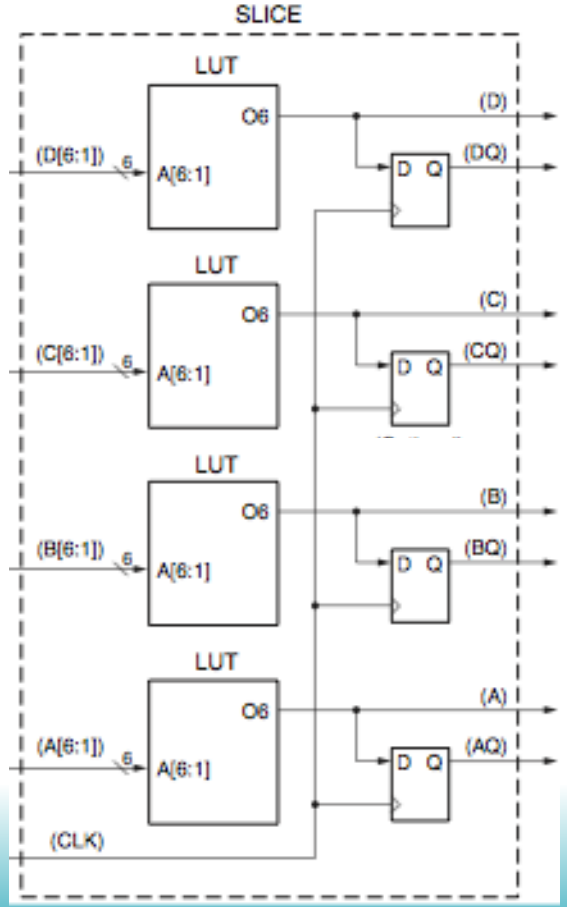
May be used as one 6-input LUT (D6 out) ...

... or as two 5-input LUTS (D6 and D5)

Combinational logic

(post configuration)

The simplest view of a slice



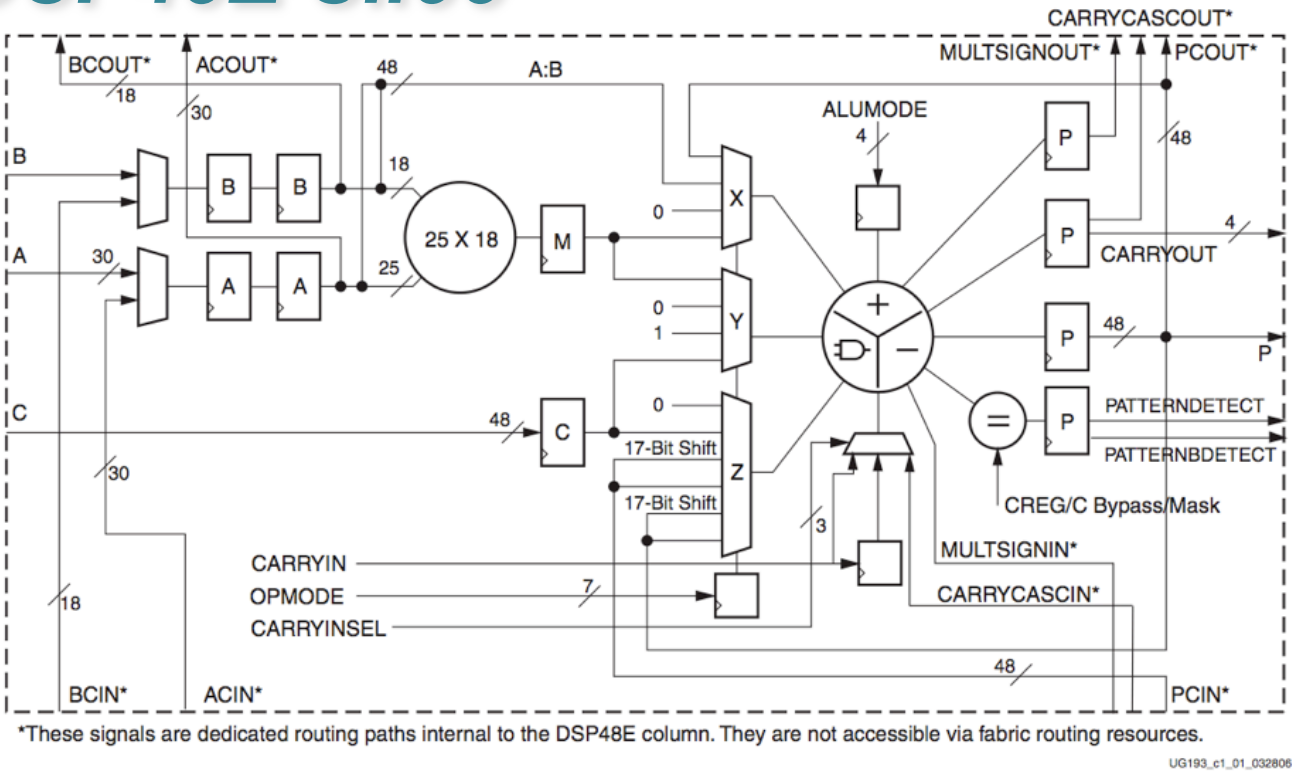
Four 6-LUTs

Four Flip-Flops

Switching fabric may see combinational and registered outputs.

An actual Virtex slice adds many small features to this simplified diagram. We show them one by one ...

Virtex DSP48E Slice



Efficient implementation of multiply, add, bit-wise logical.

Zynq-7000 SoC Data Sheet: Overview

DS190 (v1.11.1) July 2, 2018 **Product Specification**

Zynq-7000 SoC First Generation Architecture

The Zynq®-7000 family is based on the Xilinx SoC architecture. These products integrate a feature-rich dual-core or single-core ARM® Cortex™-A9 based processing system (PS) and 28 nm Xilinx programmable logic (PL) in a single device. The ARM Cortex-A9 CPUs are the heart of the PS and also include on-chip memory, external memory interfaces, and a rich set of peripheral connectivity interfaces.

Programmable Logic (PL)

Configurable Logic Blocks (CLB)

- Look-up tables (LUT)
- Flip-flops
- Cascadeable adders

36 Kb Block RAM

- True Dual-Port
- Up to 72 bits wide
- Configurable as dual 18 Kb block RAM

DSP Blocks

- 18 x 25 signed multiply
- 48-bit adder/accumulator
- 25-bit pre-adder

Programmable I/O Blocks

- Supports LVCMOS, LVDS, and SSTL
- 1.2V to 3.3V I/O
- Programmable I/O delay and SerDes

JTAG Boundary-Scan

- IEEE Std 1149.1 Compatible Test Interface

PCI Express® Block

- Supports Root complex and End Point configurations
- Supports up to Gen2 speeds
- Supports up to 8 lanes

Serial Transceivers

- Up to 16 receivers and transmitters
- Supports up to 12.5 Gb/s data rates

Two 12-Bit Analog-to-Digital Converters

- On-chip voltage and temperature sensing
- Up to 17 external differential input channels
- One million samples per second maximum conversion rate

Table 1: **Zynq-7000 and Zynq-7000S SoCs *(Cont'd)***

	Device Name	Z-7007S	Z-7012S	Z-7014S	Z-7010	Z-7015	Z-7020	Z-7030	Z-7035	Z-7045	Z-7100
	Part Number	**XC7Z007S**	**XC7Z012S**	**XC7Z014S**	**XC7Z010**	**XC7Z015**	**XC7Z020**	**XC7Z030**	**XC7Z035**	**XC7Z045**	**XC7Z100**
Programmable Logic	Xilinx 7 Series Programmable Logic Equivalent	Artix®-7 FPGA	Artix-7 FPGA	Artix-7 FPGA	Artix-7 FPGA	Artix-7 FPGA	Artix-7 FPGA	Kintex®-7 FPGA	Kintex-7 FPGA	Kintex-7 FPGA	Kintex-7 FPGA
	Programmable Logic Cells	23K	55K	65K	28K	74K	85K	125K	275K	350K	444K
	Look-Up Tables (LUTs)	14,400	34,400	40,600	17,600	46,200	53,200	78,600	171,900	218,600	277,400
	Flip-Flops	28,800	68,800	81,200	35,200	92,400	106,400	157,200	343,800	437,200	554,800
	Block RAM (# 36 Kb Blocks)	1.8 Mb (50)	2.5 Mb (72)	3.8 Mb (107)	2.1 Mb (60)	3.3 Mb (95)	4.9 Mb (140)	9.3 Mb (265)	17.6 Mb (500)	19.2 Mb (545)	26.5 Mb (755)
	DSP Slices (18x25 MACCs)	66	120	170	80	160	220	400	900	900	2,020
	Peak DSP Performance (Symmetric FIR)	73 GMACs	131 GMACs	187 GMACs	100 GMACs	200 GMACs	276 GMACs	593 GMACs	1,334 GMACs	1,334 GMACs	2,622 GMACs
	PCI Express (Root Complex or Endpoint)[3]		Gen2 x4			Gen2 x4		Gen2 x4	Gen2 x8	Gen2 x8	Gen2 x8
	Analog Mixed Signal (AMS) / XADC	2x 12 bit, MSPS ADCs with up to 17 Differential Inputs									
	Security[2]	AES and SHA 256b for Boot Code and Programmable Logic Configuration, Decryption, and Authentication									

State-of-the-Art - Xilinx FPGAs

45nm	28nm	20nm	16nm
SPARTAN-6	VIRTEX-7 KINTEX-7 ARTIX-7 SPARTAN-7	VIRTEX UltraSCALE KINTEX UltraSCALE	VIRTEX UltraSCALE+ KINTEX UltraSCALE+

Virtex Ultra-scale

Device Name	VU3P	VU5P	VU7P	VU9P	VU11P	VU13P	VU27P	VU29P	VU31P	VU33P	VU35P	VU37P
System Logic Cells (K)	862	1,314	1,724	2,586	2,835	3,780	2,835	3,780	962	962	1,907	2,852
CLB Flip-Flops (K)	788	1,201	1,576	2,364	2,592	3,456	2,592	3,456	879	879	1,743	2,607
CLB LUTs (K)	394	601	788	1,182	1,296	1,728	1,296	1,728	440	440	872	1,304
Max. Dist. RAM (Mb)	12.0	18.3	24.1	36.1	36.2	48.3	36.2	48.3	12.5	12.5	24.6	36.7
Total Block RAM (Mb)	25.3	36.0	50.6	75.9	70.9	94.5	70.9	94.5	23.6	23.6	47.3	70.9
UltraRAM (Mb)	90.0	132.2	180.0	270.0	270.0	360.0	270.0	360.0	90.0	90.0	180.0	270.0
HBM DRAM (GB)	–	–	–	–	–	–	–	–	4	8	8	8
HBM AXI Interfaces	–	–	–	–	–	–	–	–	32	32	32	32
Clock Mgmt Tiles (CMTs)	10	20	20	30	12	16	16	16	4	4	8	12
DSP Slices	2,280	3,474	4,560	6,840	9,216	12,288	9,216	12,288	2,880	2,880	5,952	9,024
Peak INT8 DSP (TOP/s)	7.1	10.8	14.2	21.3	28.7	38.3	28.7	38.3	8.9	8.9	18.6	28.1
PCIe® Gen3 x16	2	4	4	6	3	4	1	1	0	0	1	2
PCIe Gen3 x16/Gen4 x8 / CCIX[1]	–	–	–	–	–	–	–	–	4	4	4	4
150G Interlaken	3	4	6	9	6	8	6	8	0	0	2	4
100G Ethernet w/ KR4 RS-FEC	3	4	6	9	9	12	11	15	2	2	5	8
Max. Single-Ended HP I/Os	520	832	832	832	624	832	520	676	208	208	416	624
GTY 32.75Gb/s Transceivers	40	80	80	120	96	128	32	32	32	32	64	96
GTM 58Gb/s PAM4 Transceivers							32	48				
100G / 50G KP4 FEC							16 / 32	24 / 48				
Extended[2]	-1 -2 -2L -3	-1 -2 -2L -3	-1 -2 -2L -3	-1 -2 -2L -3	-1 -2 -2L -3	-1 -2 -2L -3	-1 -2 -2L -3	-1 -2 -2L -3	-1 -2 -2L -3	-1 -2 -2L -3	-1 -2 -2L -3	-1 -2 -2L -3
Industrial	-1 -2	-1 -2	-1 -2	-1 -2	-1 -2	-1 -2	-1 -2	-1 -2	–	–	–	–