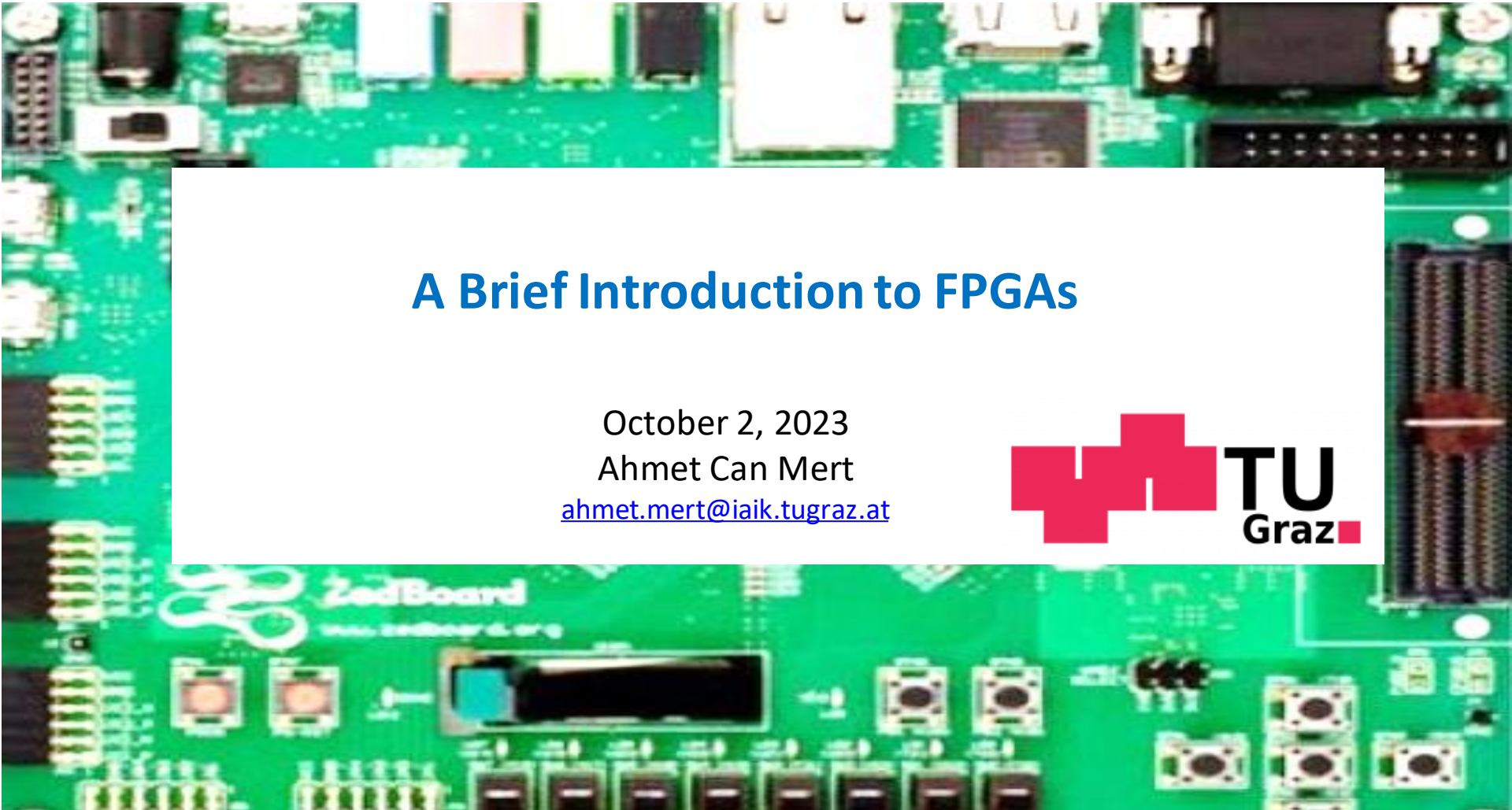


A Brief Introduction to FPGAs

October 2, 2023

Ahmet Can Mert
ahmet.mert@iaik.tugraz.at

TU Graz

FPGA: Definition

- FPGA: **F**ield **P**rogrammable **G**ate **A**rrays
  - An array of logic cells with programmable control signals and a programmable routing grid
  - It can be configured/programmed to perform any digital circuits
    - Microcontroller
    - Video processor
    - Crypto miner
  - How to configure?
    - Define its behavior using Hardware Description Languages
    - Compile and download it to FPGA

FPGA: Definition

- FPGA: **F**ield **P**rogrammable **G**ate **A**rrays
  - Configurable Logic Blocks (CLBs)
  - Programmable Interconnects
  - Programmable I/O blocks
  - Other functional blocks: DSPs, Block RAMs, Ultra RAMs, ...

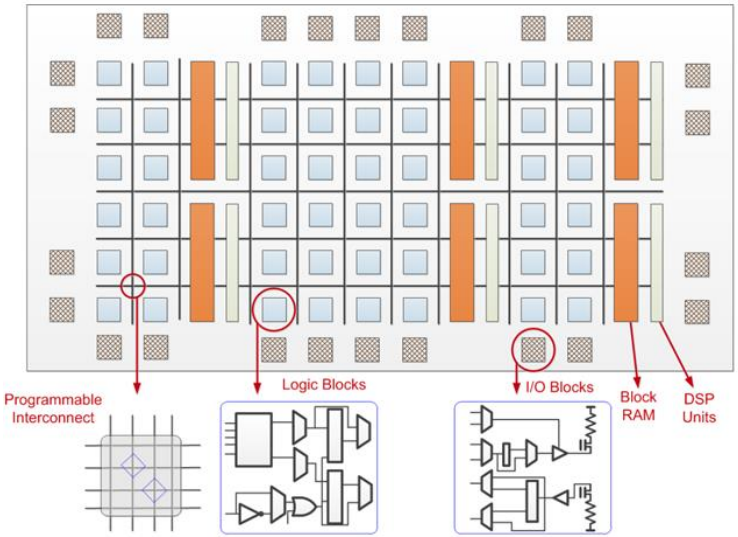


Image is retrieved from [https://towardsdatascience.com/introduction-to-fpga-and-its-architecture-20a62c14421c]

FPGA: A Brief History of FPGAs

- Some major inventions that led to FPGAs
  - 1957 – PROM (programmable read-only memory)
  - 1960 – MOSFETs (metal–oxide–semiconductor field-effect transistor)

FPGA: A Brief History of FPGAs

- Some major inventions that led to FPGAs
  - 1957 – PROM (programmable read-only memory)
  - 1960 – MOSFETs (metal–oxide–semiconductor field-effect transistor)
  - 1970 – EPROM
  - 1975 – PLA (Programmable Logic Array)
  - 1977 – PAL (Programmable Array Logic)

FPGA: A Brief History of FPGAs

- Some major inventions that led to FPGAs
  - 1957 – PROM (programmable read-only memory)
  - 1960 – MOSFETs (m r field-effect transistor)
  - 1970 – EPROM
  - 1975 – PLA (Progra
  - 1977 – PAL (Progra

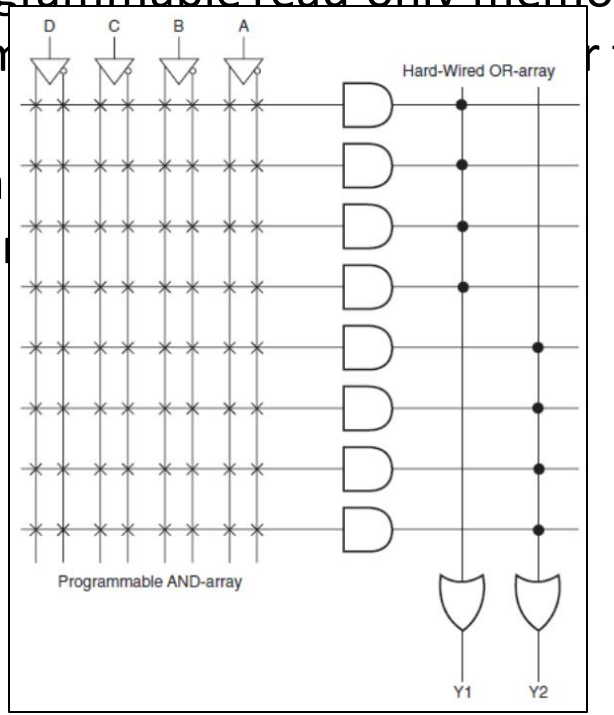


Image is retrieved from [https://www.electronics-tutorial.net/programmable-logic-devices/programmable-array-logic/]

FPGA: A Brief History of FPGAs

- Some major inventions that led to FPGAs
  - 1957 – PROM (programmable read-only memory)
  - 1960 – MOSFETs (metal–oxide–semiconductor field-effect transistor)
  - 1970 – EPROM
  - 1975 – PLA (Programmable Logic Array)
  - 1977 – PAL (Programmable Array Logic)
  - 1977 – EEPROM
  - 1981 – Flash Memory
  - 1985 – FPGA (Field Programmable Gate Arrays)

FPGA: First FPGA

- First FPGA: **XC2064**
  - In 1980s, engineers were trying to get the most out of each transistor
  - Ross Freeman came up with a different approach:
    - "*A chip packed with transistors that formed loosely organized logic blocks with connections that could be configured and reconfigured with software.*"
    - A big problem: Sometimes, some of transistors will not be used!

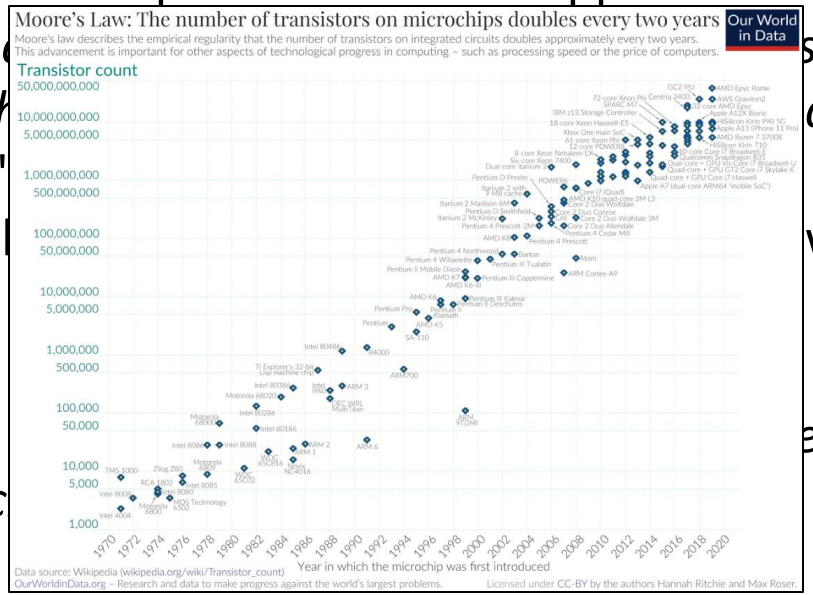
Ross Freeman: "*Moore's Law would eventually make transistors so cheap that no one would care.*"

Ross Freeman
Xilinx Co-founder

FPGA: First FPGA

- First FPGA: **XC2064**
  - In 1980s, engineers were trying to get the most out of each transistor
  - Ross Freeman came up with a different approach:
    - "*A chip pa* *sely organized logic blocks with* *d and reconfigured with software.*"
    - A big probl *will not be used!*

Ross Freeman: *e transistors so c*

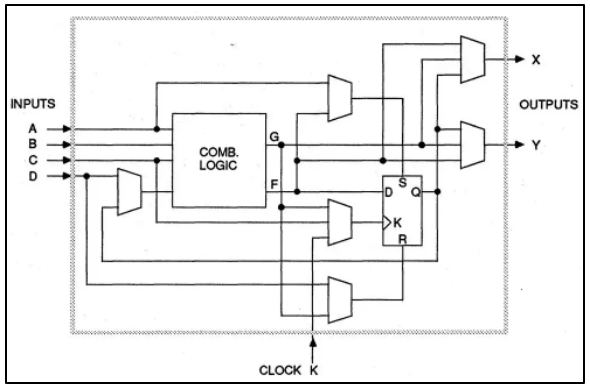


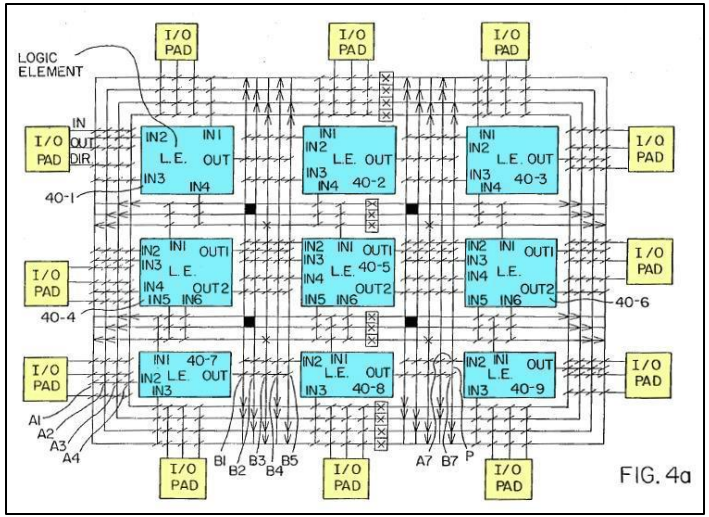
Ross Freeman
Xilinx Co-founder

FPGA: First FPGA

- First FPGA: **XC2064**
  - 8x8 grid of CLBs (Image from the Patent is a simplified 3x3 version)

  - Each CLB has 4 inputs and 2 outputs
    - Comb. Logic (**Lookup Table – LUT**)
    - D-FF (memory element)
    - Multiplexers (MUX)



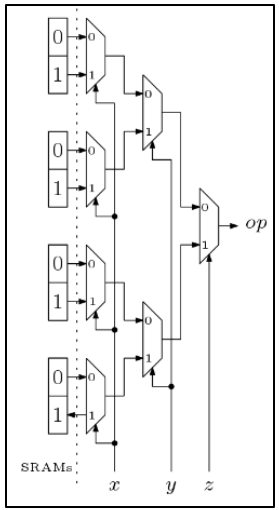


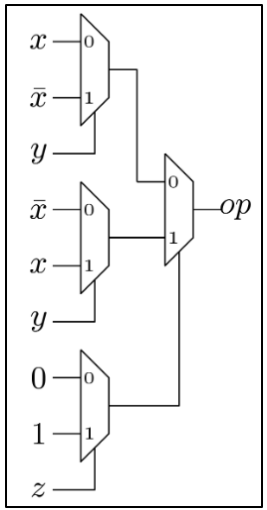
FPGA connections from Xilinx FPGA Patent

Images are retrieved from [https://semiwiki.com/fpga/290990-reverse-engineering-the-first-fpga-chip-xilinx-xc2064/]

FPGA: First FPGA

- **CLB: Configurable Logic Block**
  - LUT-based vs MUX-based CLBs





- How are CLBs configured?

Images are retrieved from [https://digitalsystemdesign.in/fpga-basic-overview/]

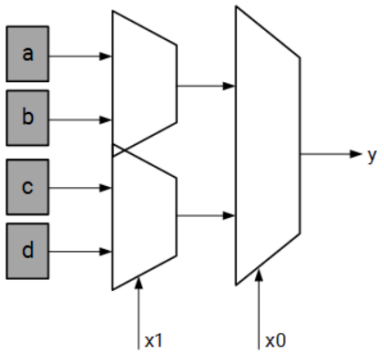
FPGA: First FPGA

- **LUT: Lookup Table**
  - Inside logic block, FPGAs do not have gates. Instead, they have lookup tables with reconfigurable outputs.
  - These lookup tables are used to implement truth table of an expression.
  - It is capable of implementing any logic function of *N* Boolean variables.
    - In modern XIlinx FPGAs, *N* is generally 6.

FPGA: First FPGA

- **LUT: Lookup Table**
  - Inside logic block, FPGAs do not have gates. Instead, they have lookup tables with reconfigurable outputs.
  - These lookup tables are used to implement truth table of an expression.
  - It is capable of implementing any logic function of *N* Boolean variables.
    - In modern XIlinx FPGAs, *N* is generally 6.
- Example: AND gate with *N*=2

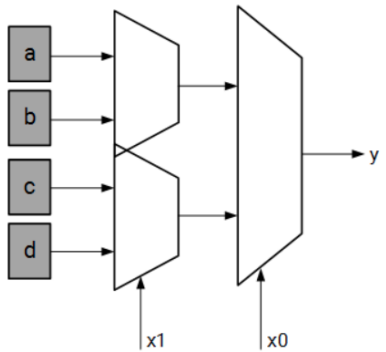
x0	x1	y
0	0	
0	1	
1	0	
1	1	



FPGA: First FPGA

- **LUT: Lookup Table**
  - Inside logic block, FPGAs do not have gates. Instead, they have lookup tables with reconfigurable outputs.
  - These lookup tables are used to implement truth table of an expression.
  - It is capable of implementing any logic function of *N* Boolean variables.
    - In modern XIlinx FPGAs, *N* is generally 6.
- Example: AND gate with *N*=2

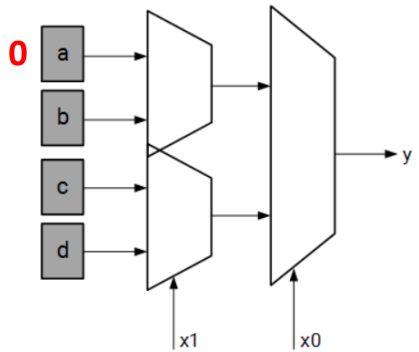
x0	x1	y
0	0	0
0	1	0
1	0	0
1	1	1



FPGA: First FPGA

- **LUT: Lookup Table**
  - Inside logic block, FPGAs do not have gates. Instead, they have lookup tables with reconfigurable outputs.
  - These lookup tables are used to implement truth table of an expression.
  - It is capable of implementing any logic function of *N* Boolean variables.
    - In modern XIlinx FPGAs, *N* is generally 6.
- Example: AND gate with *N*=2

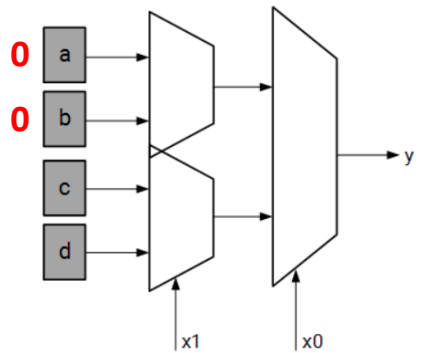
x0	x1	y
0	0	0
0	1	0
1	0	0
1	1	1



FPGA: First FPGA

- **LUT: Lookup Table**
  - Inside logic block, FPGAs do not have gates. Instead, they have lookup tables with reconfigurable outputs.
  - These lookup tables are used to implement truth table of an expression.
  - It is capable of implementing any logic function of *N* Boolean variables.
    - In modern XIlinx FPGAs, *N* is generally 6.
- Example: AND gate with *N*=2

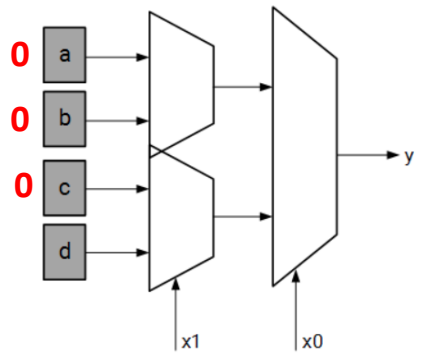
x0	x1	y
0	0	0
0	1	0
1	0	0
1	1	1



FPGA: First FPGA

- **LUT: Lookup Table**
  - Inside logic block, FPGAs do not have gates. Instead, they have lookup tables with reconfigurable outputs.
  - These lookup tables are used to implement truth table of an expression.
  - It is capable of implementing any logic function of *N* Boolean variables.
    - In modern XIlinx FPGAs, *N* is generally 6.
- Example: AND gate with *N*=2

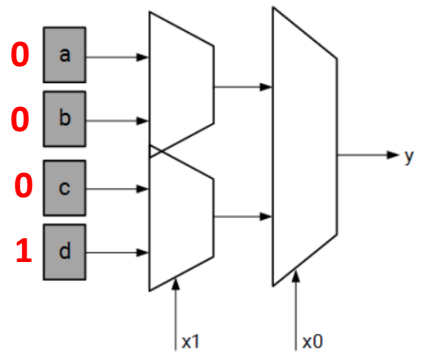
x0	x1	y
0	0	0
0	1	0
1	0	0
1	1	1



FPGA: First FPGA

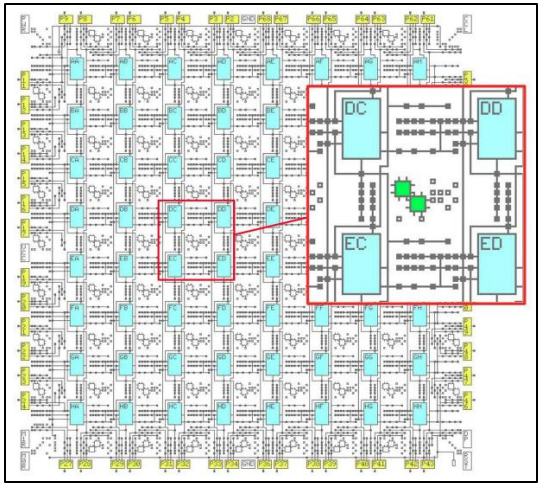
- **LUT: Lookup Table**
  - Inside logic block, FPGAs do not have gates. Instead, they have lookup tables with reconfigurable outputs.
  - These lookup tables are used to implement truth table of an expression.
  - It is capable of implementing any logic function of *N* Boolean variables.
    - In modern XIlinx FPGAs, *N* is generally 6.
- Example: AND gate with *N*=2

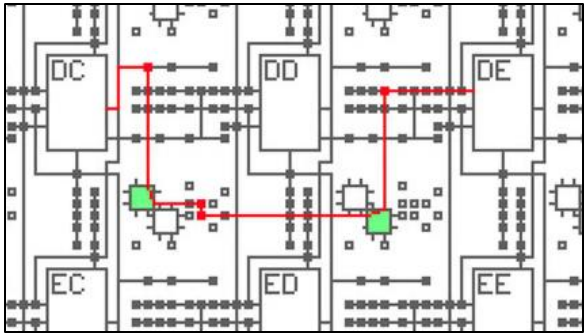
x0	x1	y
0	0	0
0	1	0
1	0	0
1	1	1



FPGA: First FPGA

- The interconnects (*routing*)
  - There are several horizontal and vertical line segments.
  - Interconnect points allow connections to be made between a horizontal line and a vertical line, allowing arbitrary paths to be created.
    - Switch matrices
- Example: First FPGA (**XC2064**)



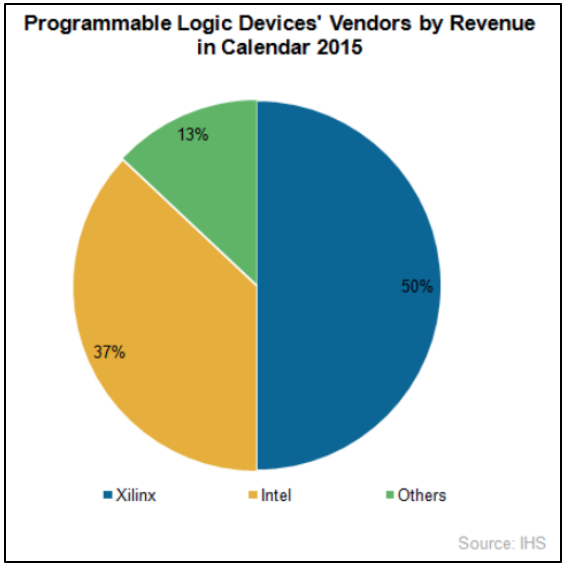


Images are retrieved from [https://semiwiki.com/fpga/290990-reverse-engineering-the-first-fpga-chip-xilinx-xc2064/]

FPGA: Modern FPGAs

- Modern FPGAs have …
  - More CLBs
  - Larger LUTs (6-input LUTs)
  - Custom building blocks (DSPs, BRAMs, Advanced I/O, ...)

- Major FPGA Vendors
  - Xilinx (acquired by AMD for 49B$ in 2022)
  - Altera (acquired by Intel for 15B$ in 2015)
  - Others (Lattice semiconductor etc.)



FPGA: Modern FPGAs

- Example: Xilinx FPGA portfolio

- Virtex Ultrascale+ FPGA Product Table:

	Foundation							58G PAM4		
Device Name	VU3P	VU5P	VU7P	VU9P	VU11P	VU13P	VU19P	VU23P	VU27P	VU29P
System Logic Cells (K)	862	1,314	1,724	2,586	2,835	3,780	8,938	2,252	2,835	3,780
CLB Flip-Flops (K)	788	1,201	1,576	2,364	2,592	3,456	8,172	2,059	2,592	3,456
CLB LUTs (K)	394	601	788	1,182	1,296	1,728	4,086	1,030	1,296	1,728
Max. Dist. RAM (Mb)	12.0	18.3	24.1	36.1	36.2	48.3	58.4	14.2	36.2	48.3
Total Block RAM (Mb)	25.3	36.0	50.6	75.9	70.9	94.5	75.9	74.3	70.9	94.5
UltraRAM (Mb)	90.0	132.2	180.0	270.0	270.0	360.0	90.0	99.0	270.0	360.0
DSP Slices	2,280	3,474	4,560	6,840	9,216	12,288	3,840	1,320	9,216	12,288
Peak INT8 DSP (TOP/s)	7.1	10.8	14.2	21.3	28.7	38.3	10.4	4.1	28.7	38.3
PCIe® Gen3 x16	2	4	4	6	3	4	0	0	1	1
PCIe Gen3 x16/Gen4 x8 / CCIX[1]	–	–	–	–	–	–	8	4	–	–
150G Interlaken	3	4	6	9	6	8	0	0	8	8
100G Ethernet w/ KR4 RS-FEC	3	4	6	9	9	12	0	2	15	15
Max. Single-Ended HP I/Os	520	832	832	832	624	832	1,976	572	676	676
Max. Single-Ended HD I/Os	0	0	0	0	0	0	96	72	0	0
GTY 32.75Gb/s Transceivers	40	80	80	120	96	128	80	34	32	32
GTM 58Gb/s PAM4 Transceivers	–	–	–	–	–	–	–	4	48	48
100G / 50G KP4 FEC	–	–	–	–	–	–	–	2 / 4	24 / 48	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	-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	-1 -2

FPGA: Why should I learn FPGA?

- FPGAs are now everywhere
  - IoT to Data Centers

Amazon EC2 F1 instances use FPGAs to deliver custom hardware accelerations.

FPGA: Modern FPGAs

- Example: Xilinx FPGA portfolio

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

Alveo-U250 Board
~7,700$

- Virtex Ultrascale+ FPGA Product Table:

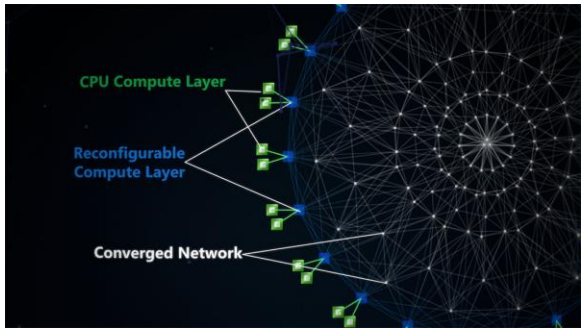
	Foundation							58G PAM4		
Device Name	VU3P	VU5P	VU7P	VU9P	VU11P	VU13P	VU19P	VU23P	VU27P	VU29P
System Logic Cells (K)	862	1,314	1,724	2,586	2,835	3,780	8,938	2,252	2,835	3,780
CLB Flip-Flops (K)	788	1,201	1,576	2,364	2,592	3,456	8,172	2,059	2,592	3,456
CLB LUTs (K)	394	601	788	1,182	1,296	1,728	4,086	1,030	1,296	1,728
Max. Dist. RAM (Mb)	12.0	18.3	24.1	36.1	36.2	48.3	58.4	14.2	36.2	48.3
Total Block RAM (Mb)	25.3	36.0	50.6	75.9	70.9	94.5	75.9	74.3	70.9	94.5
UltraRAM (Mb)	90.0	132.2	180.0	270.0	270.0	360.0	90.0	99.0	270.0	360.0
DSP Slices	2,280	3,474	4,560	6,840	9,216	12,288	3,840	1,320	9,216	12,288
Peak INT8 DSP (TOP/s)	7.1	10.8	14.2	21.3	28.7	38.3	10.4	4.1	28.7	38.3
PCIe® Gen3 x16	2	4	4	6	3	4	0	0	1	1
PCIe Gen3 x16/Gen4 x8 / CCIX[1]	–	–	–	–	–	–	8	4	–	–
150G Interlaken	3	4	6	9	6	8	0	0	8	8
100G Ethernet w/ KR4 RS-FEC	3	4	6	9	9	12	0	2	15	15
Max. Single-Ended HP I/Os	520	832	832	832	624	832	1,976	572	676	676
Max. Single-Ended HD I/Os	0	0	0	0	0	0	96	72	0	0
GTY 32.75Gb/s Transceivers	40	80	80	120	96	128	80	34	32	32
GTM 58Gb/s PAM4 Transceivers	–	–	–	–	–	–	–	4	48	48
100G / 50G KP4 FEC	–	–	–	–	–	–	–	2 / 4	24 / 48	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	-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	-1 -2

FPGA: Why should I learn FPGA?

- FPGAs are now everywhere
  - IoT to Data Centers

Amazon EC2 F1 instances use FPGAs to deliver custom hardware accelerations.

Microsoft's Catapult Project

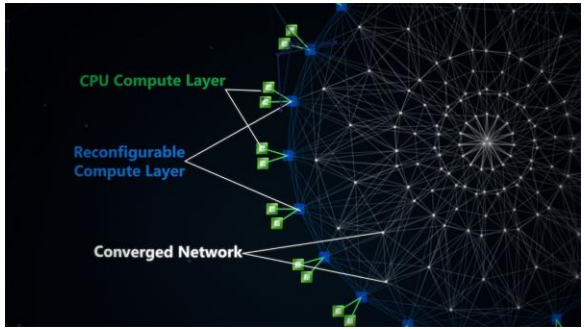


FPGA: Why should I learn FPGA?

- FPGAs are now everywhere
  - IoT to Data Centers

Amazon EC2 F1 instances use FPGAs to deliver custom hardware accelerations.

Microsoft's Catapult Project



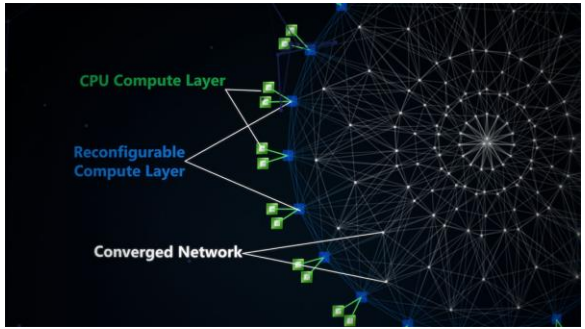
Intel Ships Xeon Skylake Processor with Integrated FPGA

FPGA: Why should I learn FPGA?

- FPGAs are now everywhere
  - IoT to Data Centers

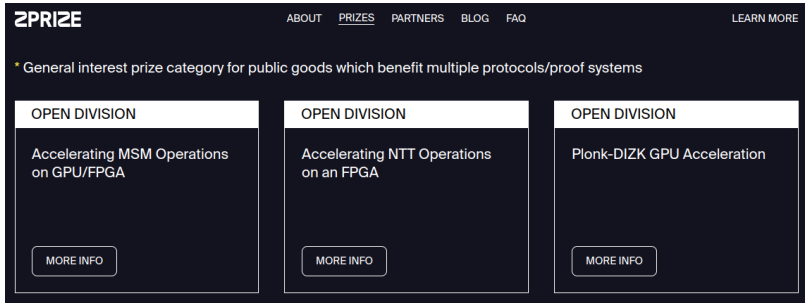
Amazon EC2 F1 instances use FPGAs to deliver custom hardware accelerations.

Microsoft's Catapult Project



Intel Ships Xeon Skylake Processor with Integrated FPGA

FPGAs are used for accelerating complex computations in blockchain technology



FPGA: Why should I learn FPGA?

- Growing market
  - 2022: 8B$ market
  - 2027: 15B$ market (estimated)
  - Job market is not saturated

- Many research direction
  - Acceleration of computation-intensive tasks
    - Machine Learning, DNN
    - Cryptography
    - Video Processing
  - FPGA security
  - High level synthesis

FPGA: Where are FPGAs used?

- FPGAs could be …
  - A target platform
  - A prototype platform for ASIC or SoC

FPGA: Where are FPGAs used?

- FPGAs could be …
  - A target platform
  - A prototype platform for ASIC or SoC

- FPGAs are used in a variety of applications
  - Machine Learning, AI
  - Cybersecurity
  - Video Processing
  - Automotive
  - Space Technology
  - Finance

FPGA: Advantages and Disadvantages

Advantages

- Programmability
- Fast design time (time to market)
- Low design cost
- Suitable for automation
- Parallel processing
- Prototyping
- System on chip

Disadvantages

- Low performance
- Large area
  (due to reconfigurability)
- High power consumption

FPGA: FPGA vs ASIC

FPGA

- Simple design cycle
- Short development time
- High adaptability
- Low NRE cost

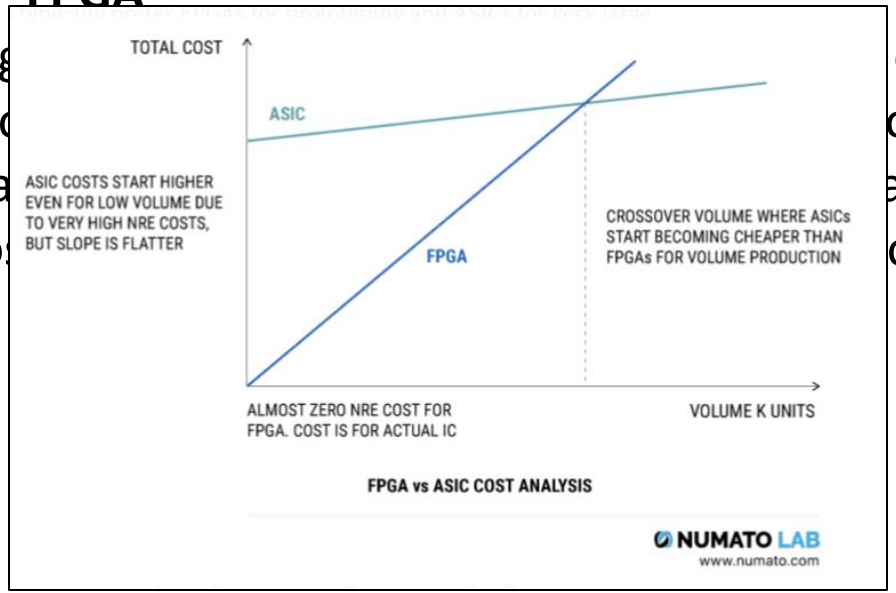
ASIC

- Difficult to design
- Long development time
- Low adaptability
- High NRE cost

FPGA: FPGA vs ASIC

FPGA **ASIC**

- Simple desig design
- Short develo opment time
- High adapta ability
- Low NRE co ost



FPGA: FPGA vs ASIC

FPGA

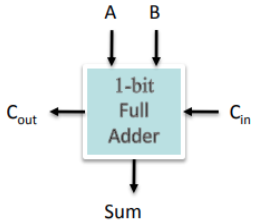
- Simple design cycle
- Short development time
- High adaptability
- Low NRE cost
- Low performance
- Large area
- High power consumption

ASIC

- Difficult to design
- Long development time
- Low adaptability
- High NRE cost
- High performance
- Low area
- Low power consumption

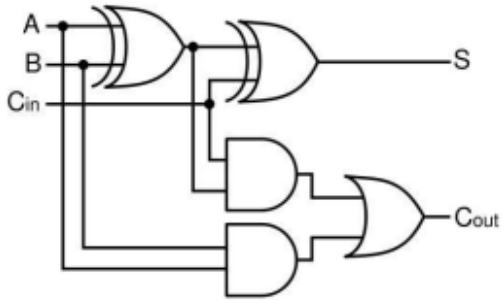
FPGA: FPGA vs ASIC

- In ASIC, an expression is implemented using gates (AND, OR, etc.) while FPGAs map the expression into lookup tables
  - Example: Full Adder



S = A XOR B XOR Cin
Cout = (A AND B) OR (A XOR B AND)

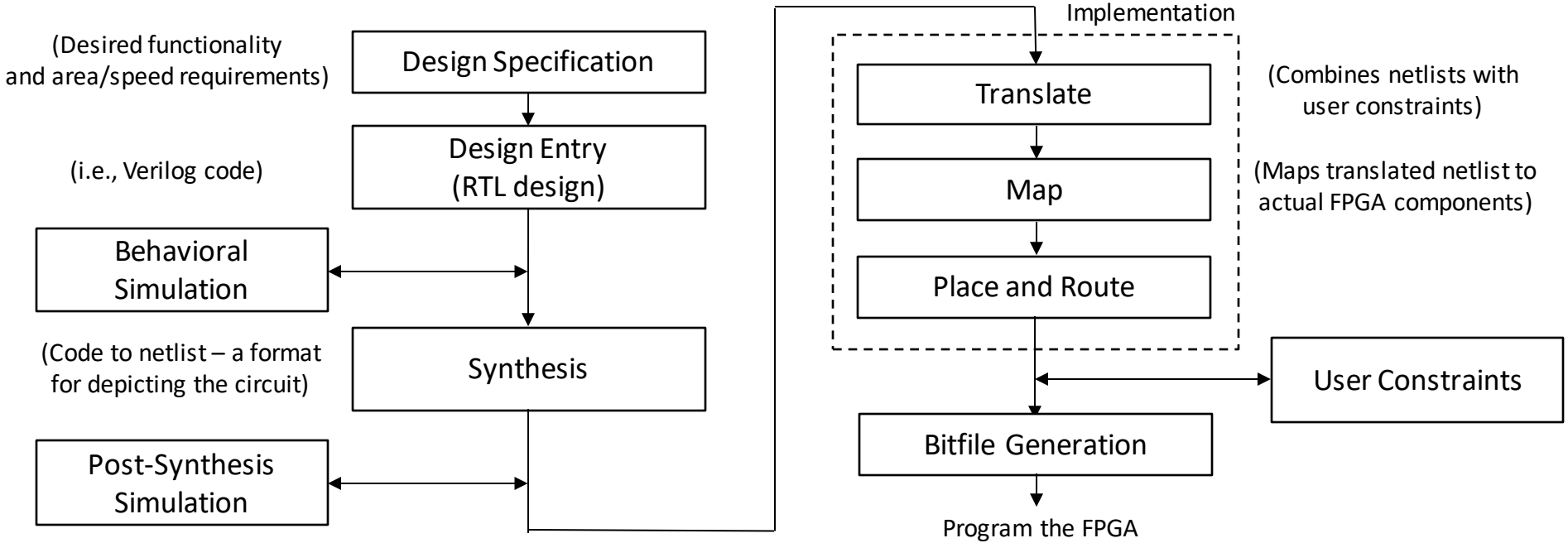
ASIC



FPGA with 4-input LUTs?

FPGA: Design Flow

- Design specification to FPGA programming



FPGA: Components (a bit more in detail)

- FPGA: **F**ield **P**rogrammable **G**ate **A**rrays
  - Configurable Logic Blocks (CLBs)
  - Programmable Interconnects
  - Programmable I/O blocks
  - Other functional blocks: DSPs, Block RAMs, Ultra RAMs, ...

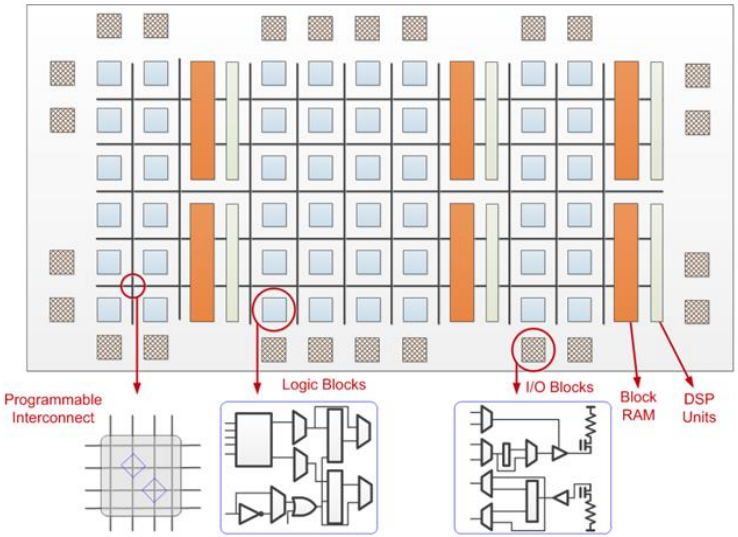


Image is retrieved from [https://towardsdatascience.com/introduction-to-fpga-and-its-architecture-20a62c14421c]

FPGA: Other functional components

- 7-series overview (Artix-7)
  - Modern FPGAs have diverse resources

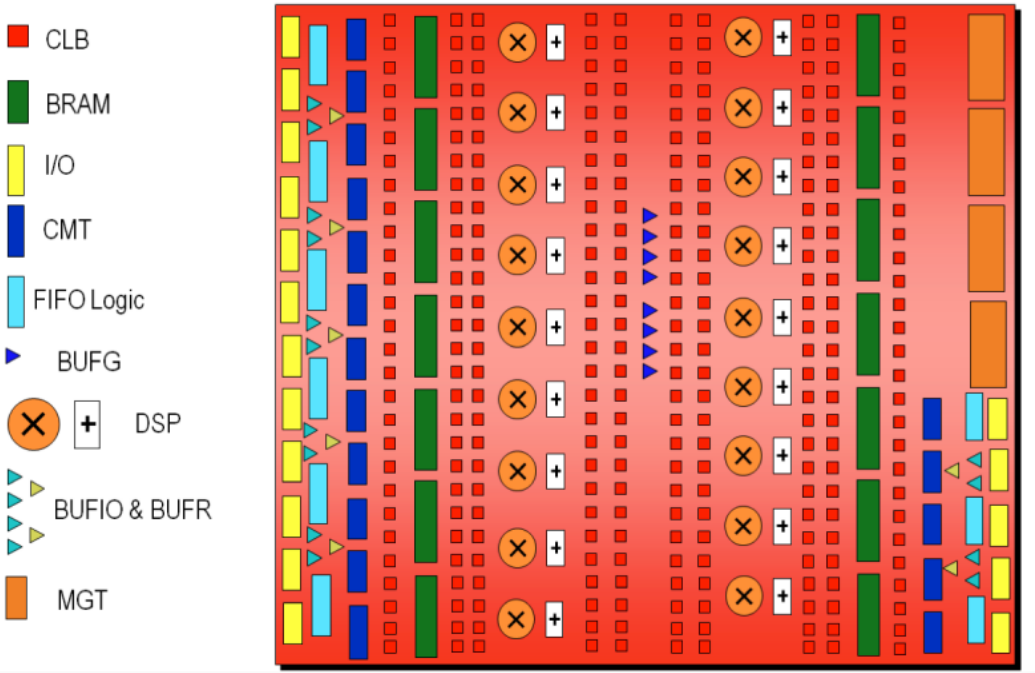


Image is retrieved from [https://xilinx.eetrend.com/files-eetrend-xilinx/forum/201509/9204-20390-7_series_architecture_overview.pdf]

FPGA: Other functional components

- Configurable Logic Block (CLB)
  - Has 2 slices
    - SLICEM (LUT can be used for logic and memory)
    - SLICEL (LUT can be used only for logic)
  - Connected to interconnect through switch matrix

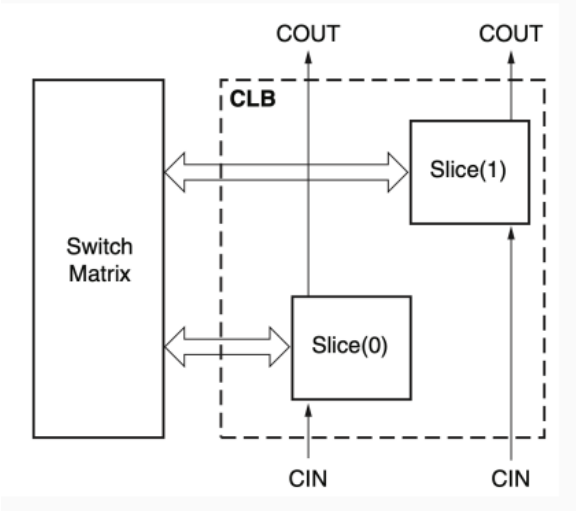
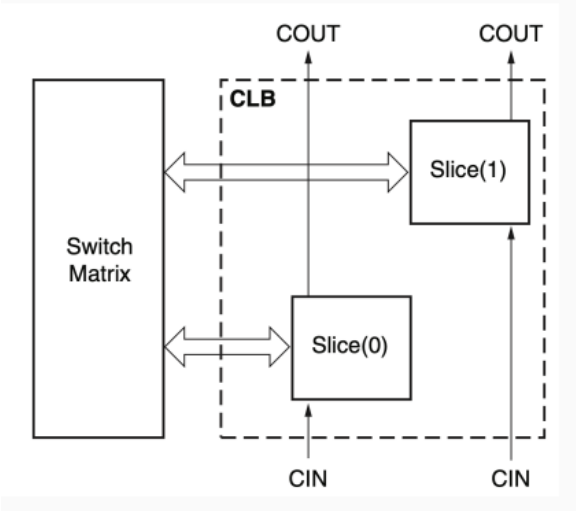


Image is retrieved from [https://xilinx.eetrend.com/files-eetrend-xilinx/forum/201509/9204-20390-7_series_architecture_overview.pdf]

FPGA: Other functional components

- Configurable Logic Block (CLB)
  - Has 2 slices
    - SLICEM (LUT can be used for logic and memory)
    - SLICEL (LUT can be used only for logic)
  - Connected to interconnect through switch matrix



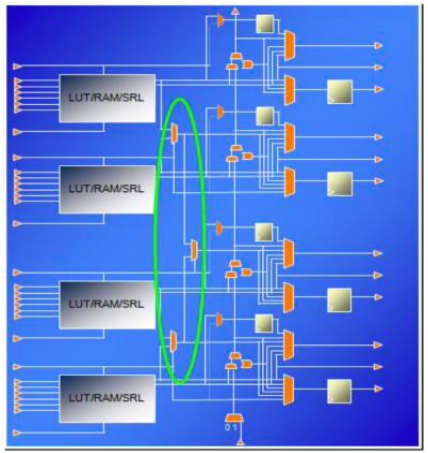
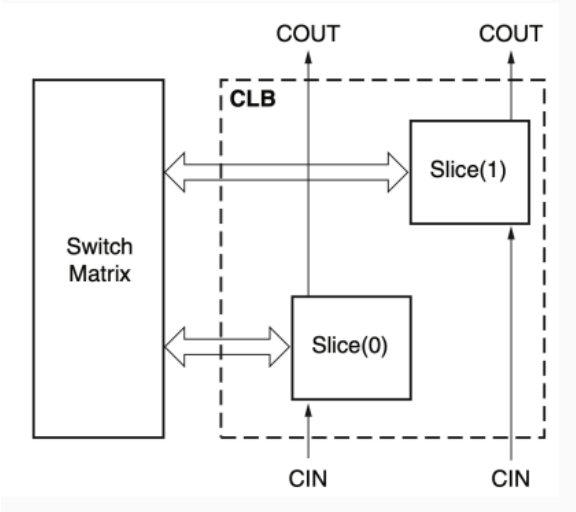


Image is retrieved from [https://xilinx.eetrend.com/files-eetrend-xilinx/forum/201509/9204-20390-7_series_architecture_overview.pdf]

FPGA: Other functional components

- Configurable Logic Block (CLB)
  - Has 2 slices
    - SLICEM (LUT can be used for logic and memory)
    - SLICEL (LUT can be used only for logic)
  - Connected to interconnect through switch matrix



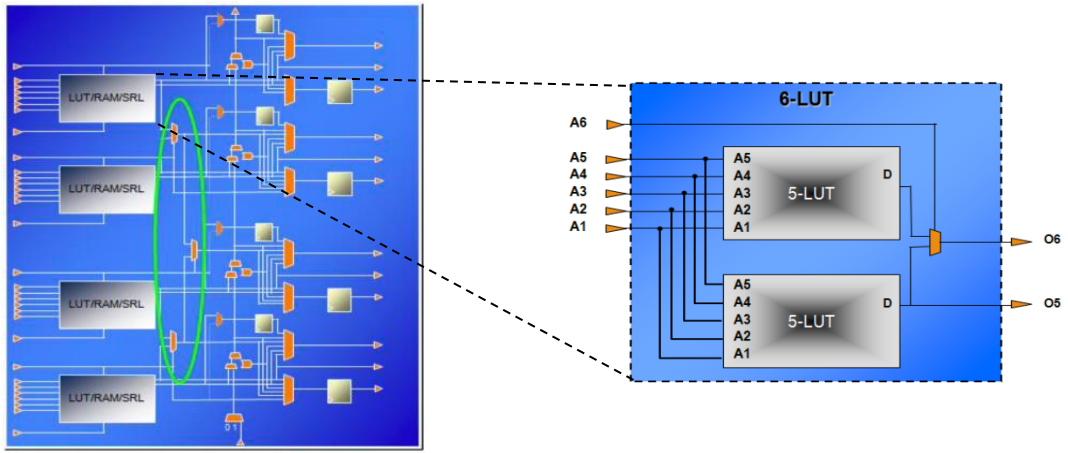


Image is retrieved from [https://xilinx.eetrend.com/files-eetrend-xilinx/forum/201509/9204-20390-7_series_architecture_overview.pdf]

FPGA: Other functional components

- DSP Slice
  - Full-custom and low power DSP units
  - Reconfigurable: P = (A±D) x B + C

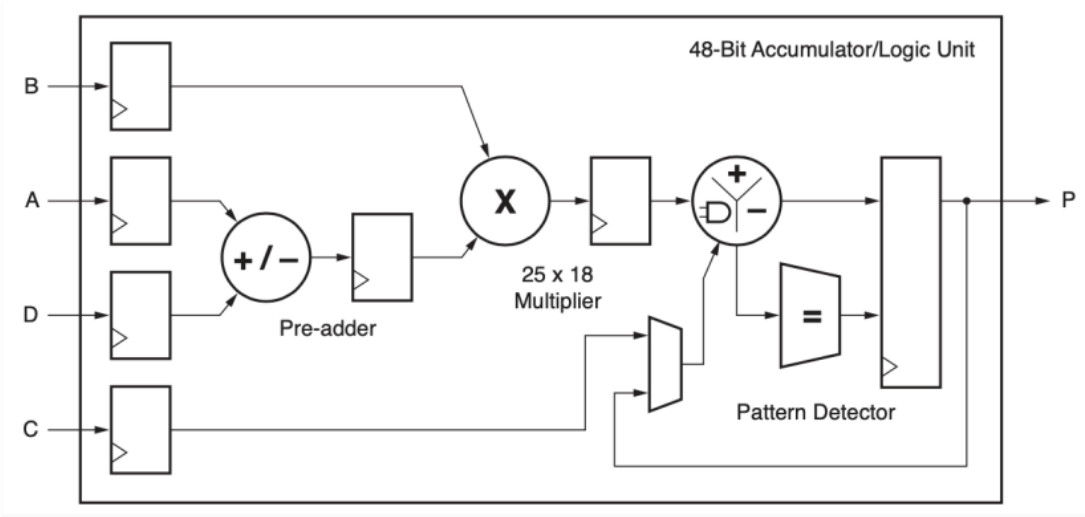


Image is retrieved from [https://xilinx.eetrend.com/files-eetrend-xilinx/forum/201509/9204-20390-7_series_architecture_overview.pdf]

FPGA: Other functional components

- Block RAM (BRAM)/Block ROM (BROM)
  - For storing large data
  - Configurable (2 x 18Kb or 1 x 36 Kb)
  - Two read/write ports
    - Input/output port: 18/36/72

- Advanced FPGAs have Ultra RAM (URAM)
  - 288 Kb

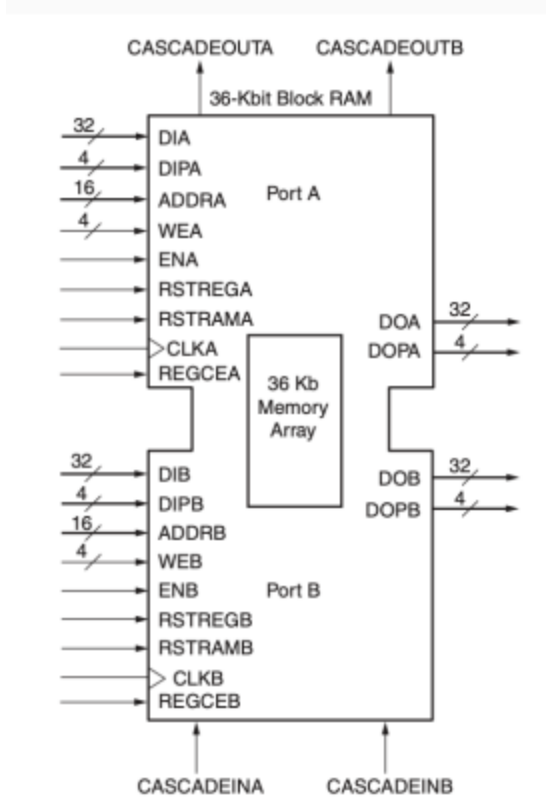


Image is retrieved from [https://xilinx.eetrend.com/files-eetrend-xilinx/forum/201509/9204-20390-7_series_architecture_overview.pdf]

FPGA: Other functional components

- Block RAM (BRAM)/Block ROM (BROM)
  - For storing large data
  - Configurable (2 x 18Kb or 1 x 36 Kb)
  - Two read/write ports
    - Input/output port: 18/36/72

- Advanced FPGAs have Ultra RAM (URAM)
  - 288 Kb

- FPGA boards might be coupled with processors
  - Xilinx PYNQ-Z2 has an ARM Cortex

- It is also possible to instantiate "soft" processor in FPGAs
  - Xilinx's Microblaze soft processor core

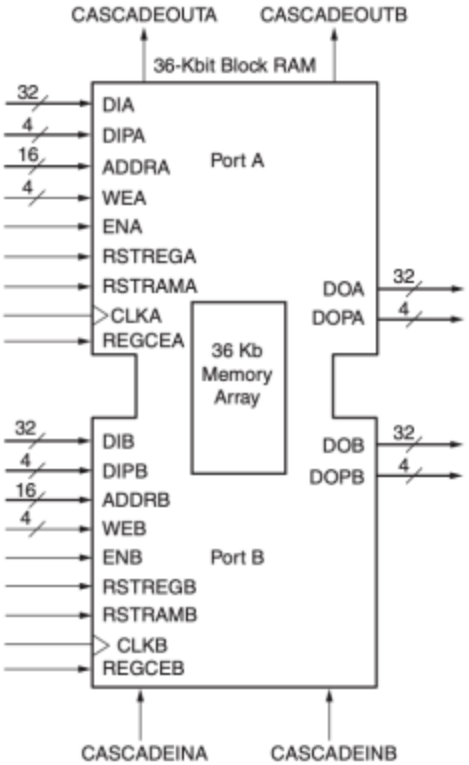


Image is retrieved from [https://xilinx.eetrend.com/files-eetrend-xilinx/forum/201509/9204-20390-7_series_architecture_overview.pdf]