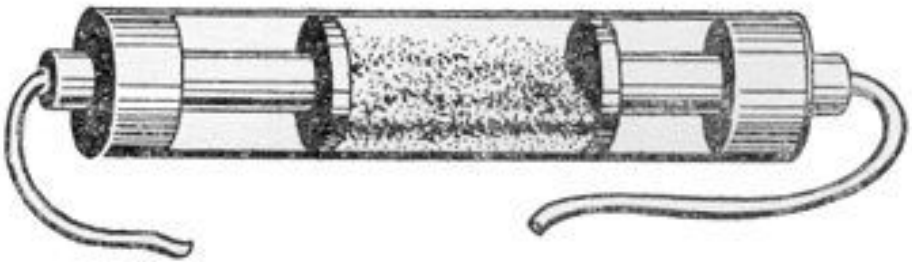
Introduction to FPGA's

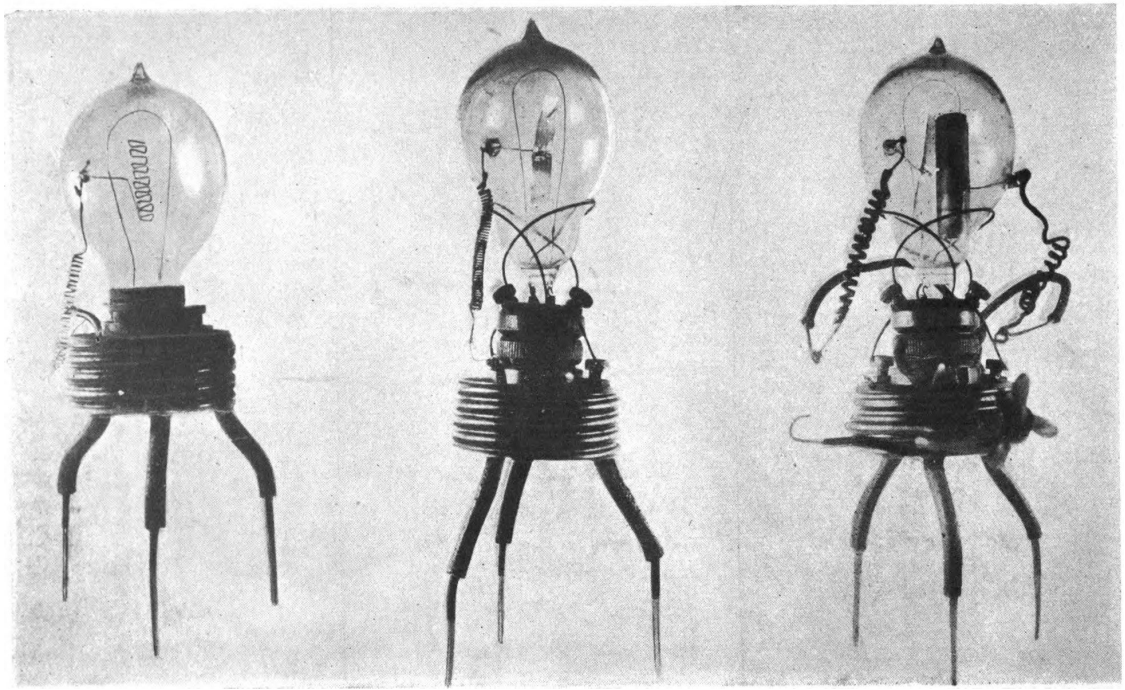
(and how we got there)

11th August 2022

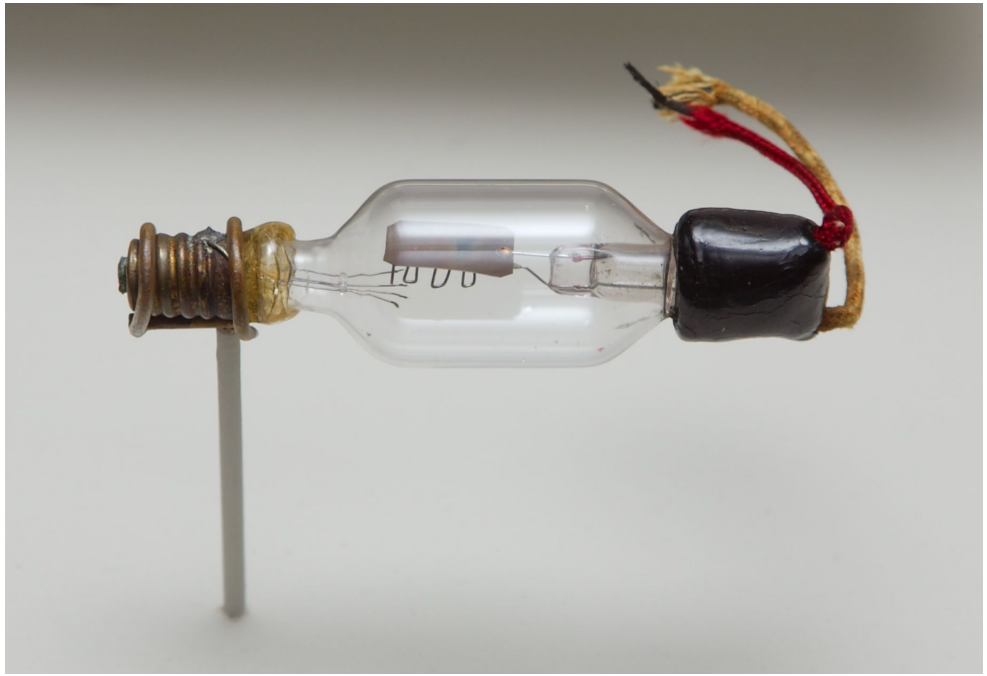
In the beginning



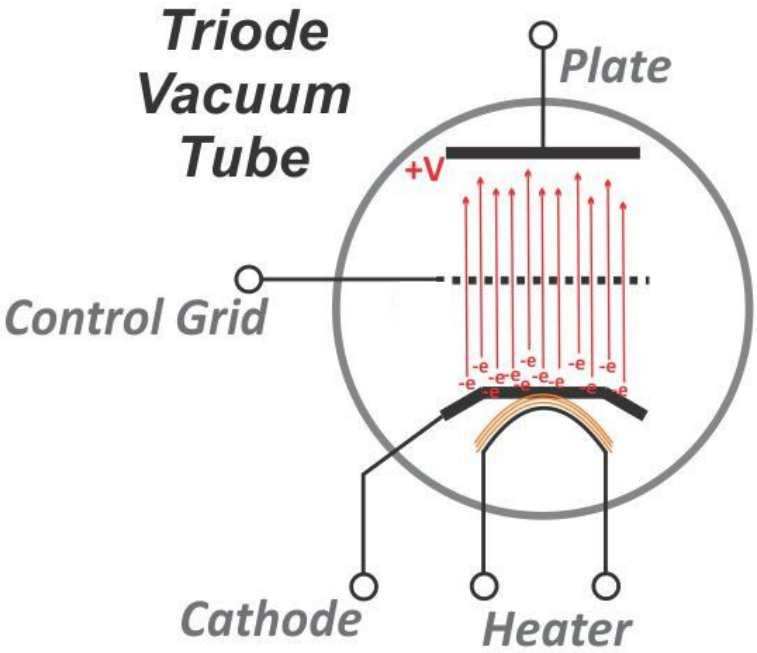
Enter the Thermionic Valves

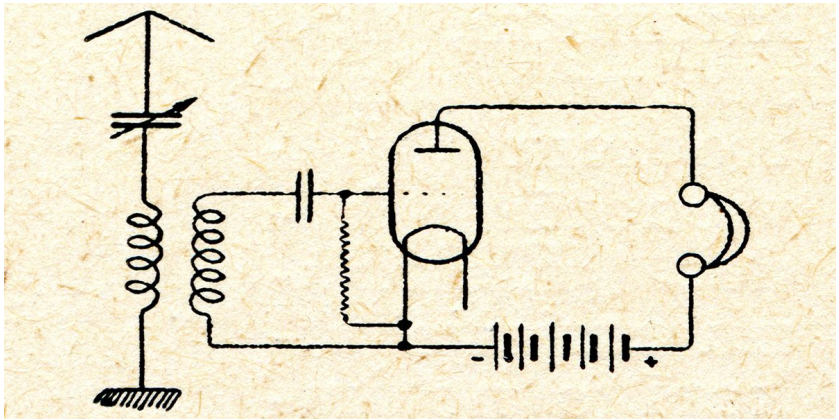


Tubes



Triode Receivers





Vacuum Tube Logic (VTL)

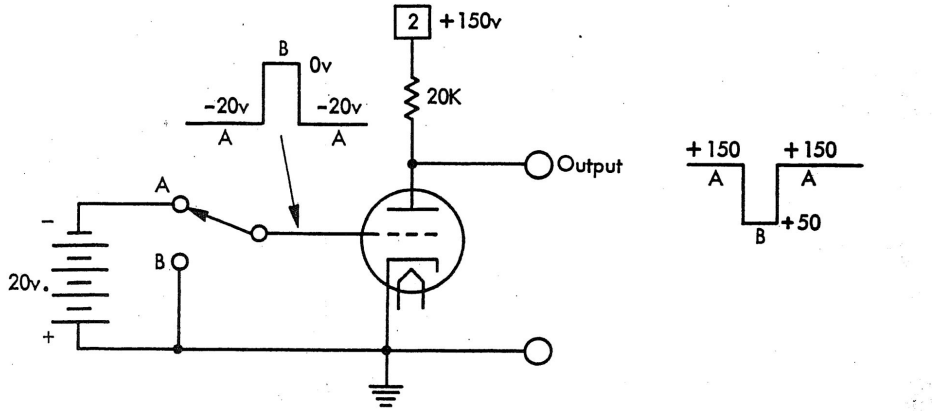
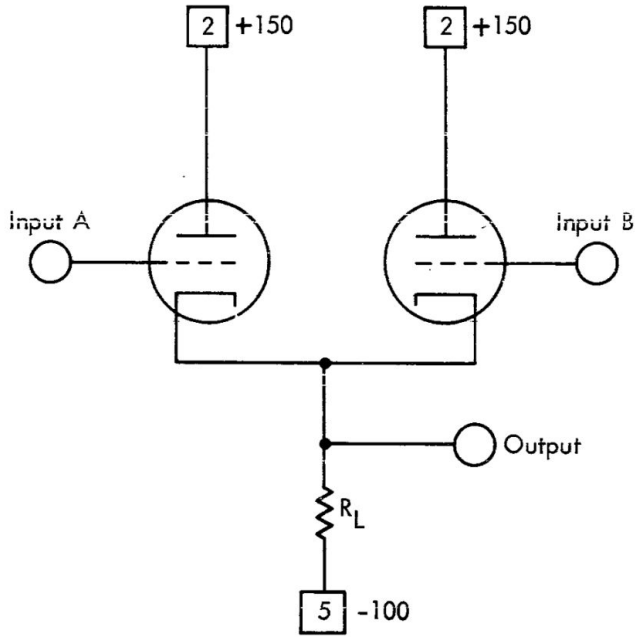


Figure 21. Basic Inverter Circuit

http://www.qrp.gr/technology/logic/5484436-Basic-Principles-of-Vacuum-Tube-Logic-Circuits.pdf

“OR” Gate



Input		Output	
A	B		
−25	−25	−22	(Mixer No Signal Condition)
+25	−25	+26	
−25	+25	+26	
+25	+25	+26	(Switch No Signal Condition)

VTL Flip Flop

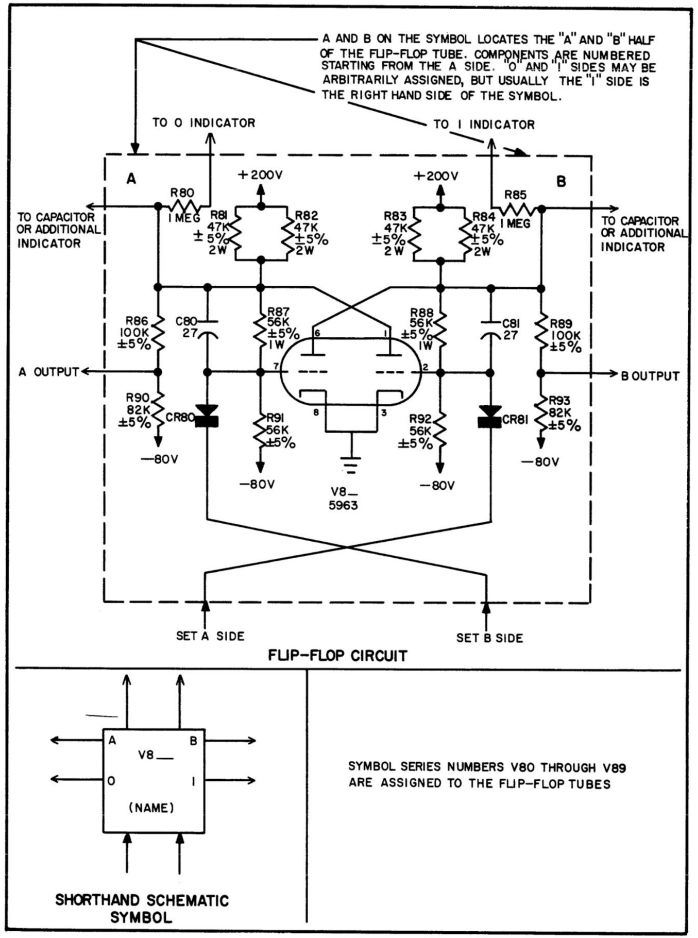


Figure 3. STANDARD Flip-flop Stage
PX 131

Digital Circuits using tubes

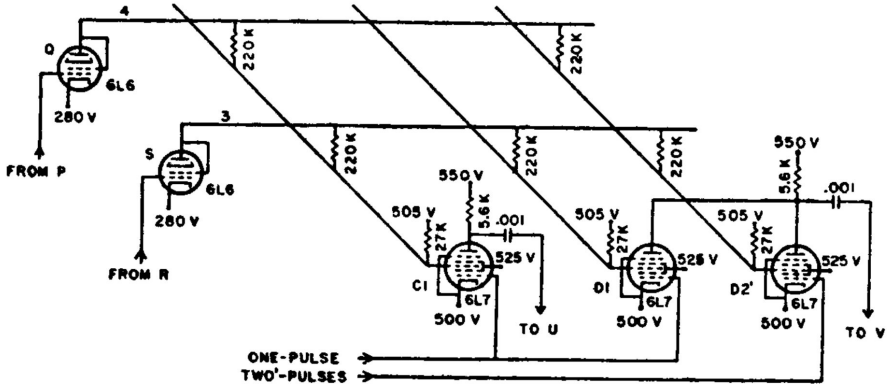
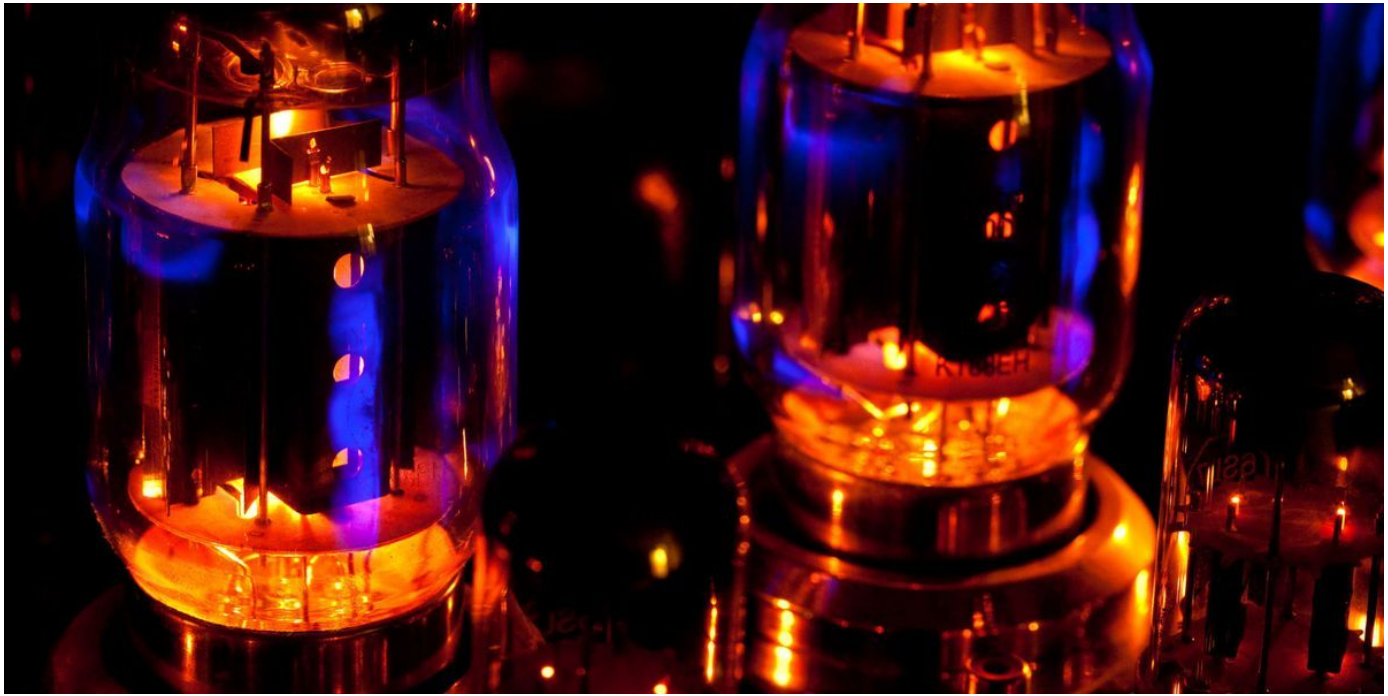


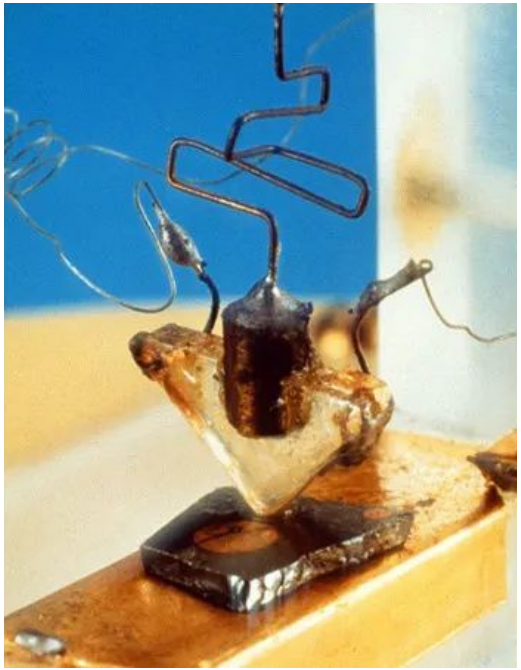
Fig. 7—Cross section of multiplication table.

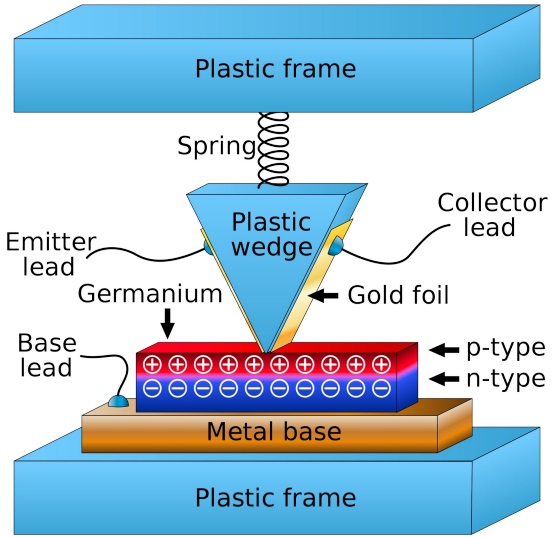
http://archive.computerhistory.org/resources/text/Knuth_Don_X4100/PDF_index/k-8-pdf/k-8-r5367-1-ENIAC-circuits.pdf

End of Part 1

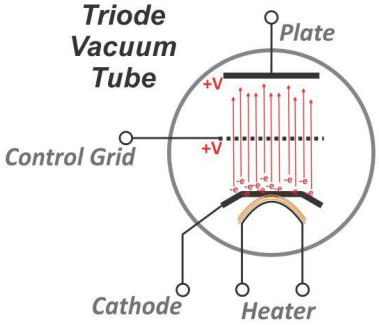


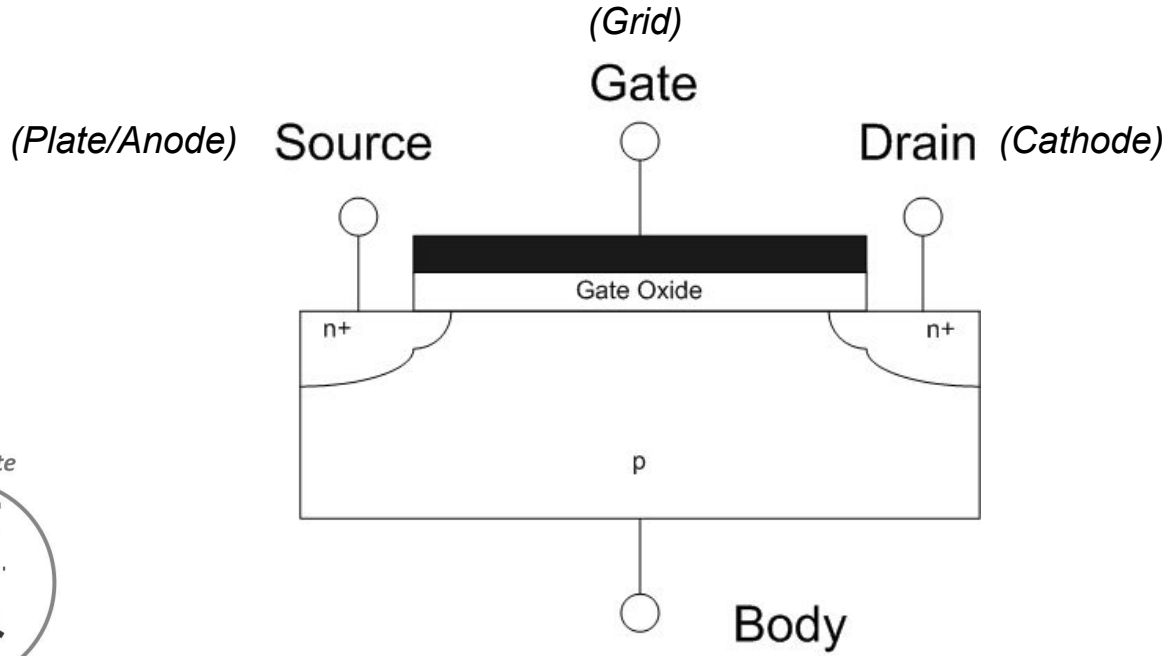
Transistors

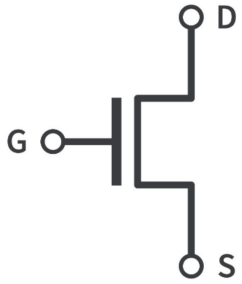




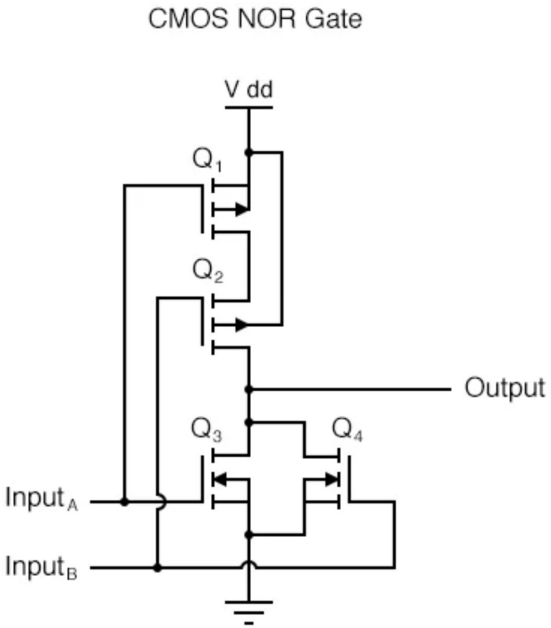
Field Effect Transistor (FET)







CMOS FET Gates



https://www.allaboutcircuits.com/textbook/digital/chpt-3/cmos-gate-circuitry/

FPGA’s

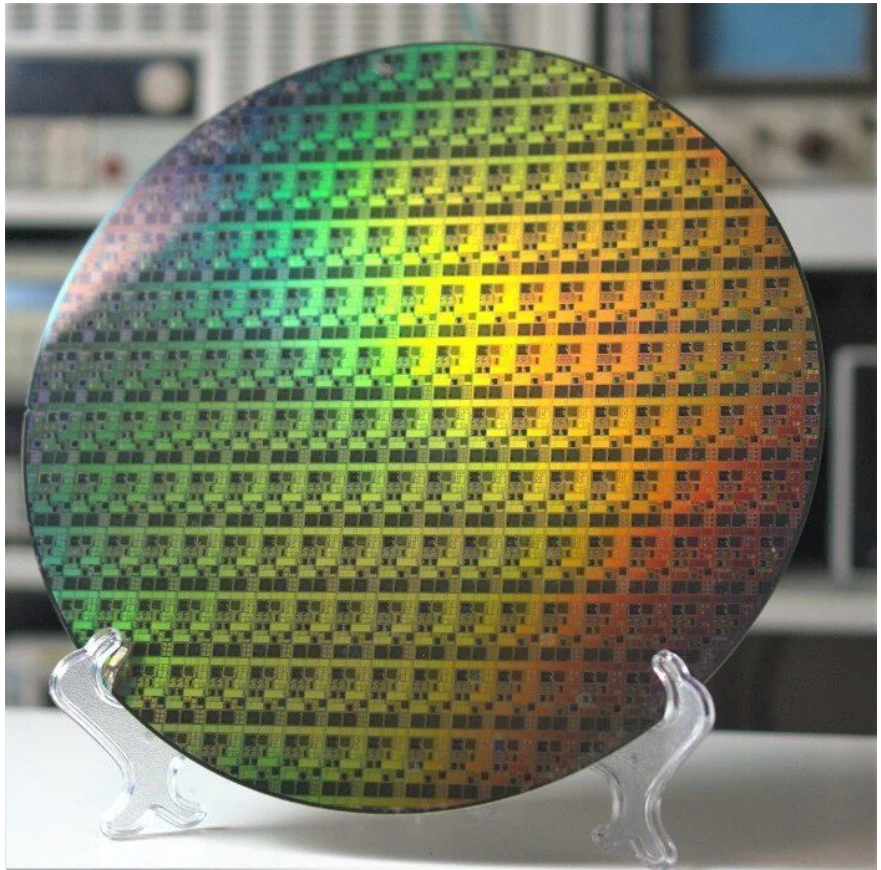
Field

Programable

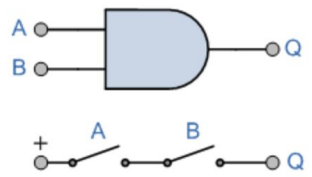
Gate

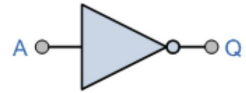
Arrays

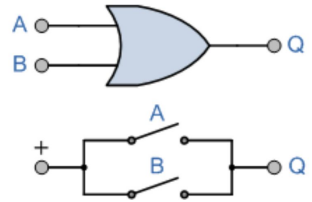
Silicon Chips



Logic Gates

Symbol	Truth Table		
	B	A	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1

Symbol	Truth Table	
	A	Q
	0	1
	1	0

Symbol	Truth Table		
	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1

AND

In A	In B	Out
0	0	0
1	0	0
0	1	0
1	1	1

NAND

In A	In B	Out
0	0	1
1	0	1
0	1	1
1	1	0

OR

In A	In B	Out
0	0	0
1	0	1
0	1	1
1	1	1

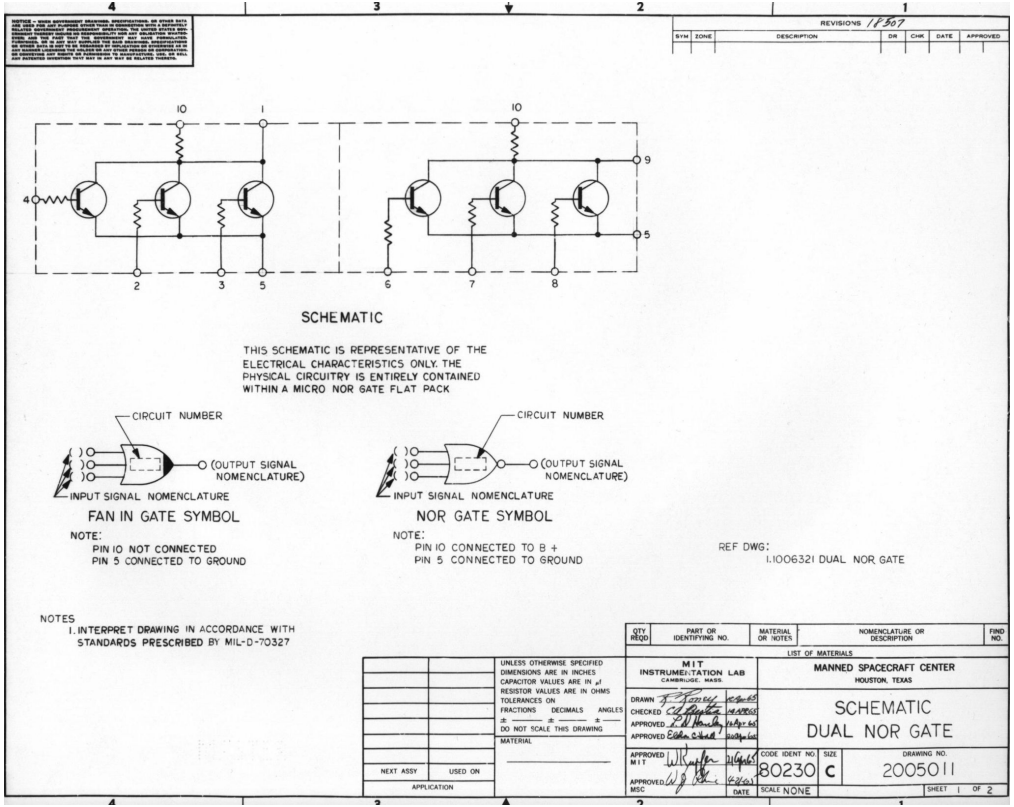
NOR

In A	In B	Out
0	0	1
1	0	0
0	1	0
1	1	0

XOR

In A	In B	Out
0	0	0
1	0	1
0	1	1
1	1	0

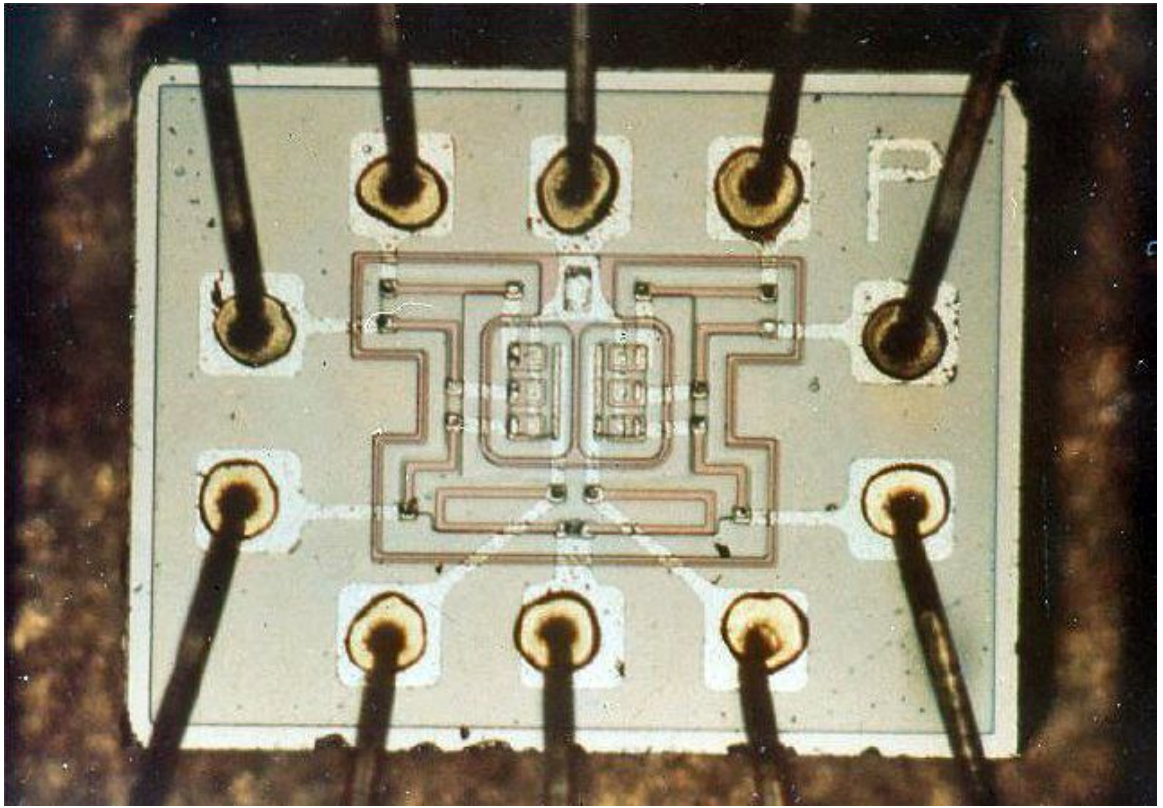
Apollo Guidance Computer 3 input NOR Gate



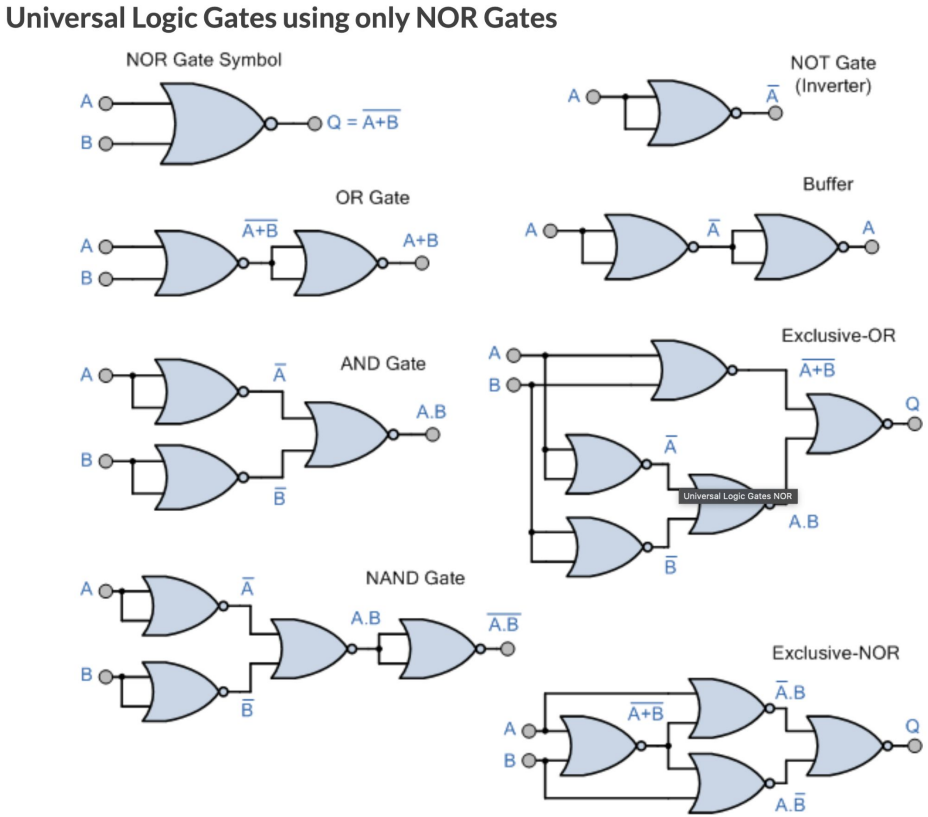
Inputs			outputs
W	X	Y	Z = W. X. Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

http://klabs.org/history/ech/agc_schematics/

Apollo 3 input NOR gate Integrated Circuit

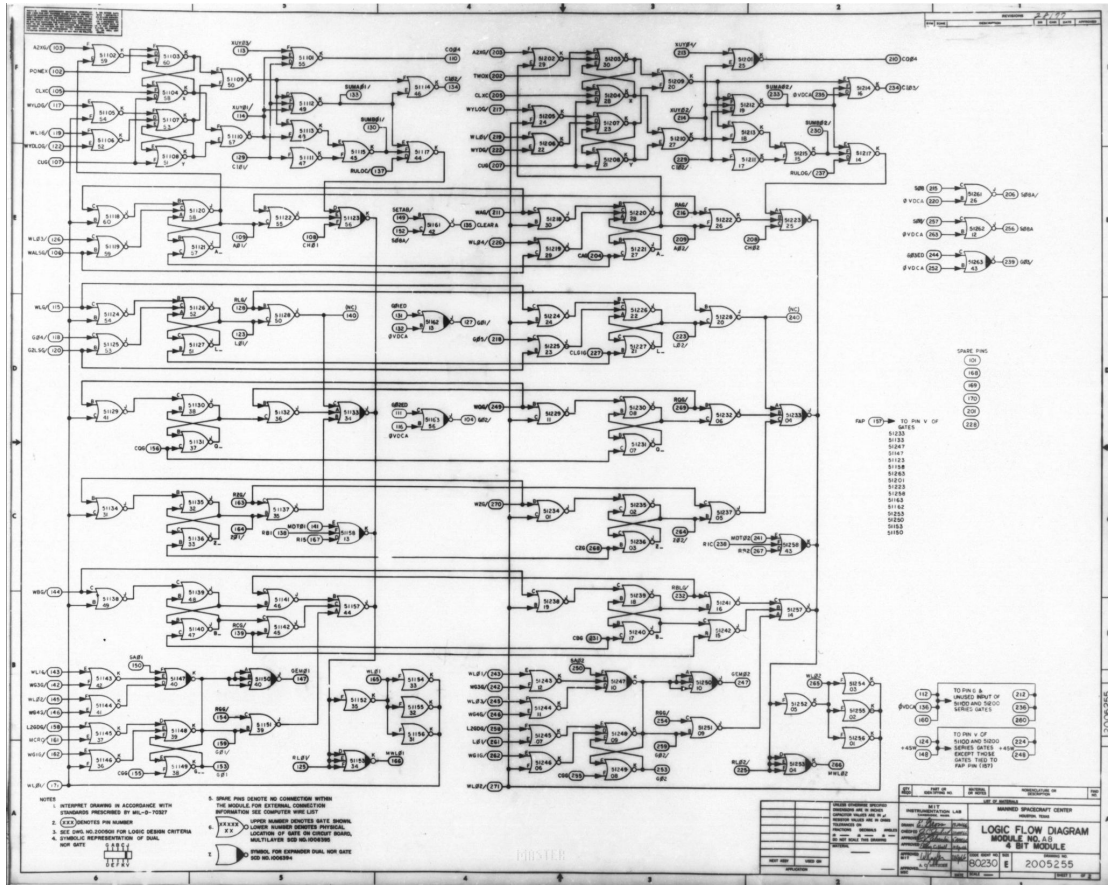


Logic using only NOR Gates



https://www.electronics-tutorials.ws/logic/universal-gates.html

Navigating to the Moon on NOR Gates



https://www.ibiblio.org/apollo/

Look Up Table (LUT)

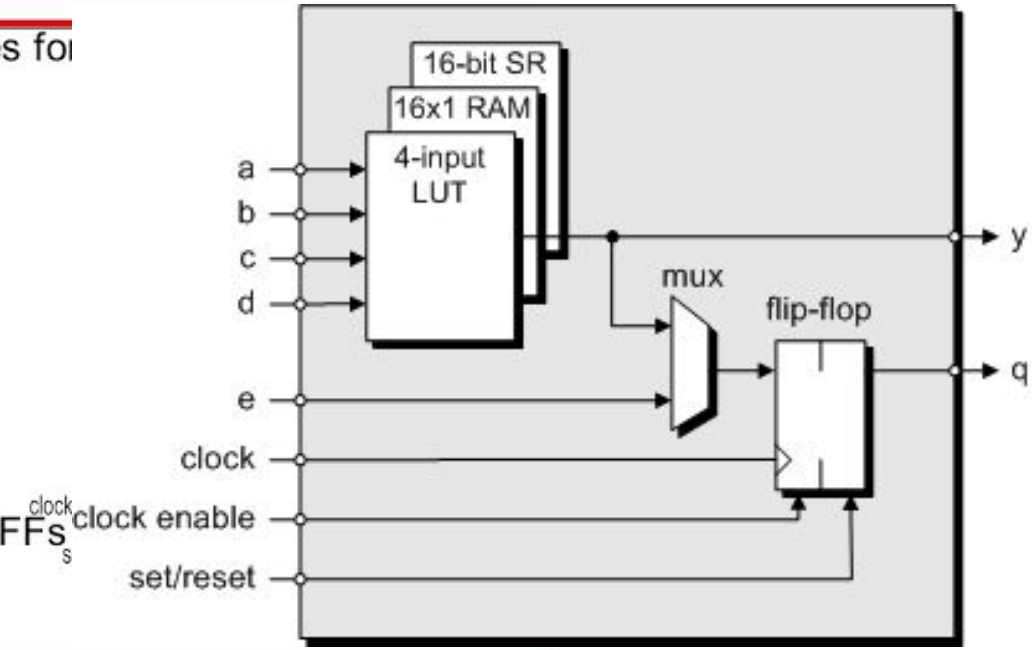
LUT

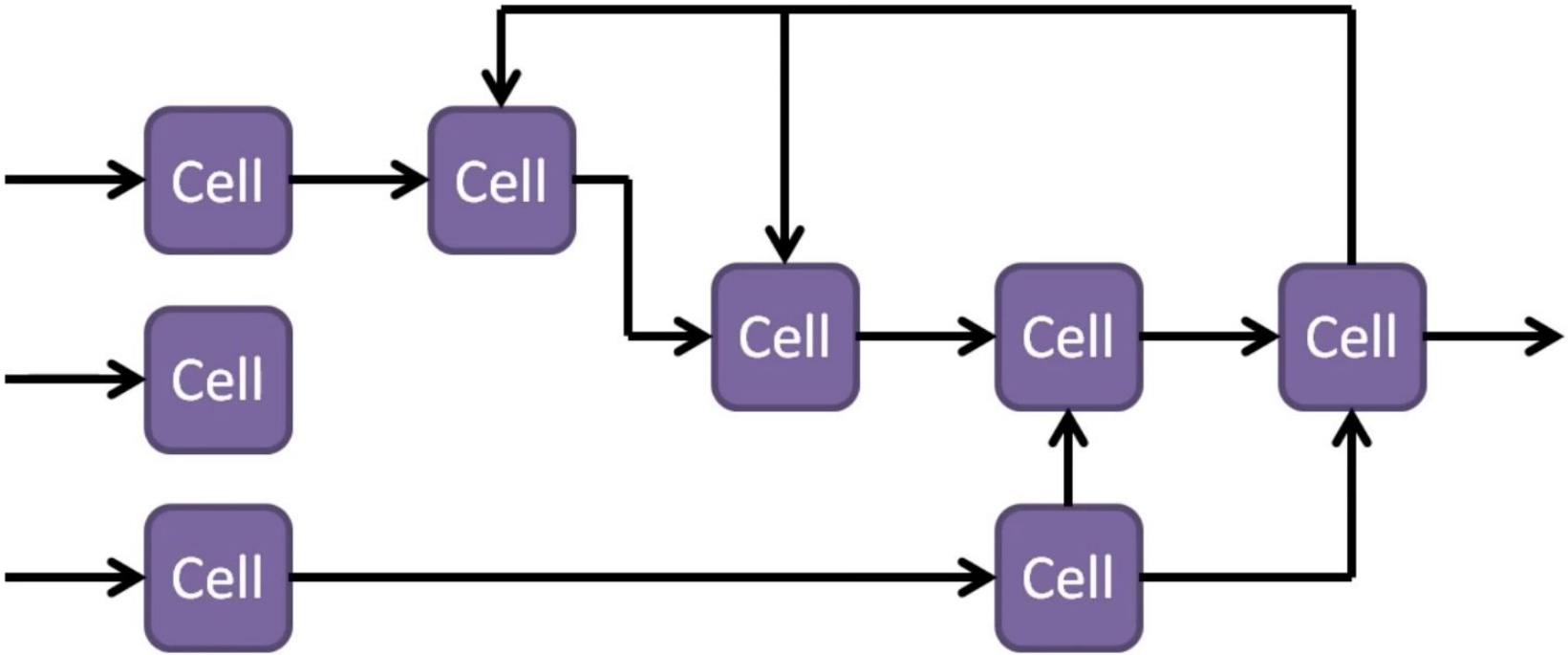
In A	In B	In C	In D	Out
0	0	0	0	?
1	0	0	0	?
0	1	0	0	?
...				
1	0	1	1	?
0	1	1	1	?
1	1	1	1	?

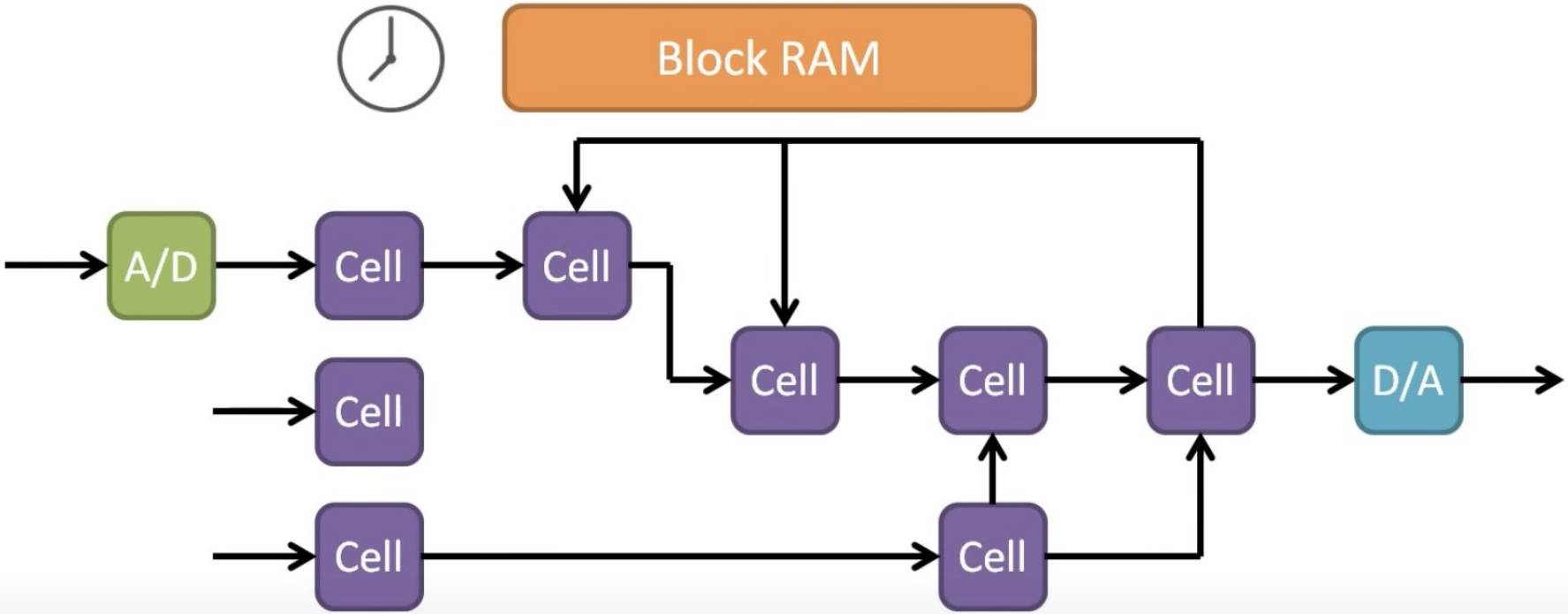
Cells

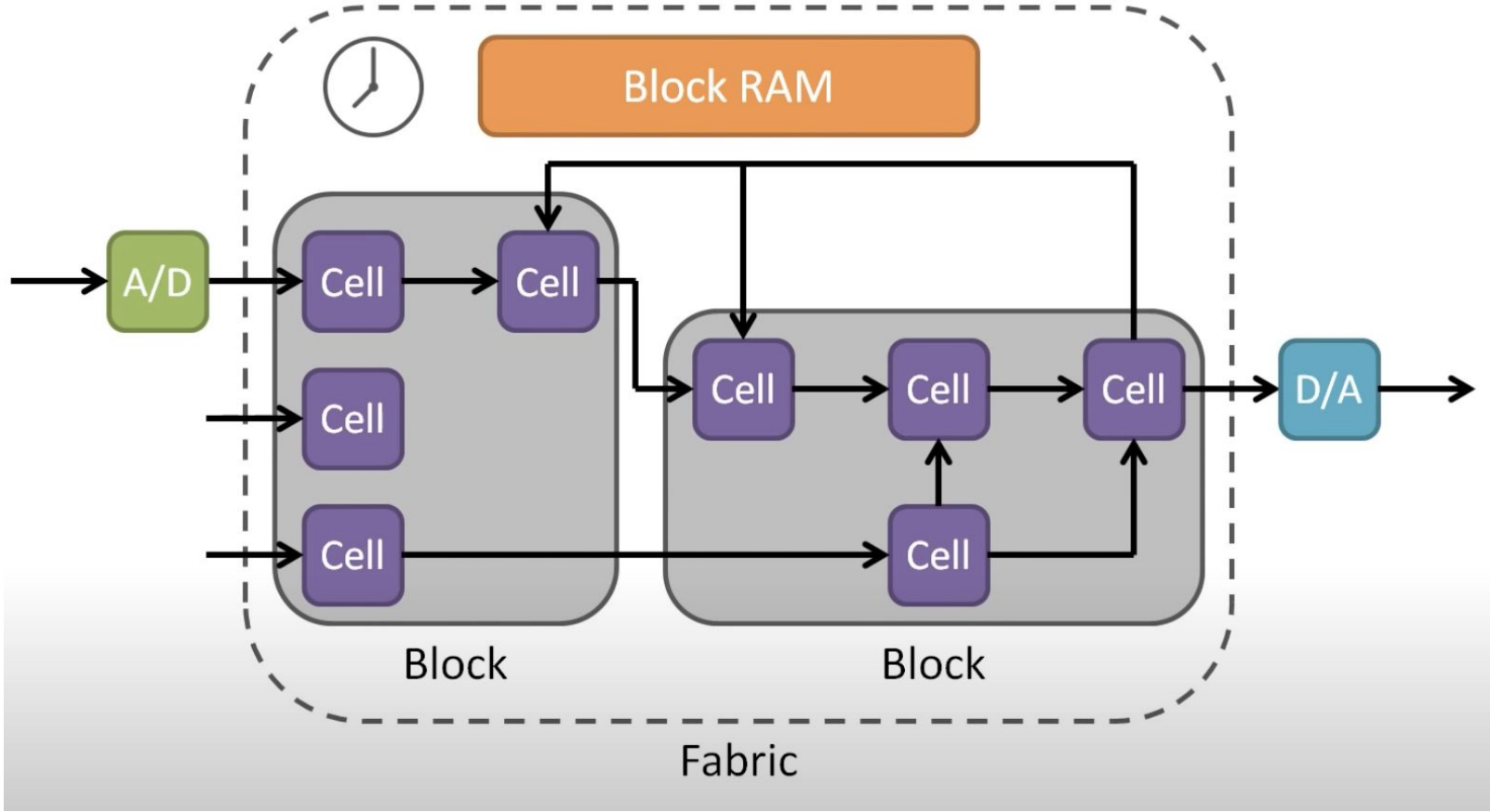
Xilinx logic cell

- Each vendor has its own names for things.
- Xilinx call it logic cell (LC), comprises:
  - 4-input LUT
  - MUX
  - Register
- Clock can be configured rising versus falling
- Register can be configured as FFs or as a latch.
- Altera call it logic element (LE)

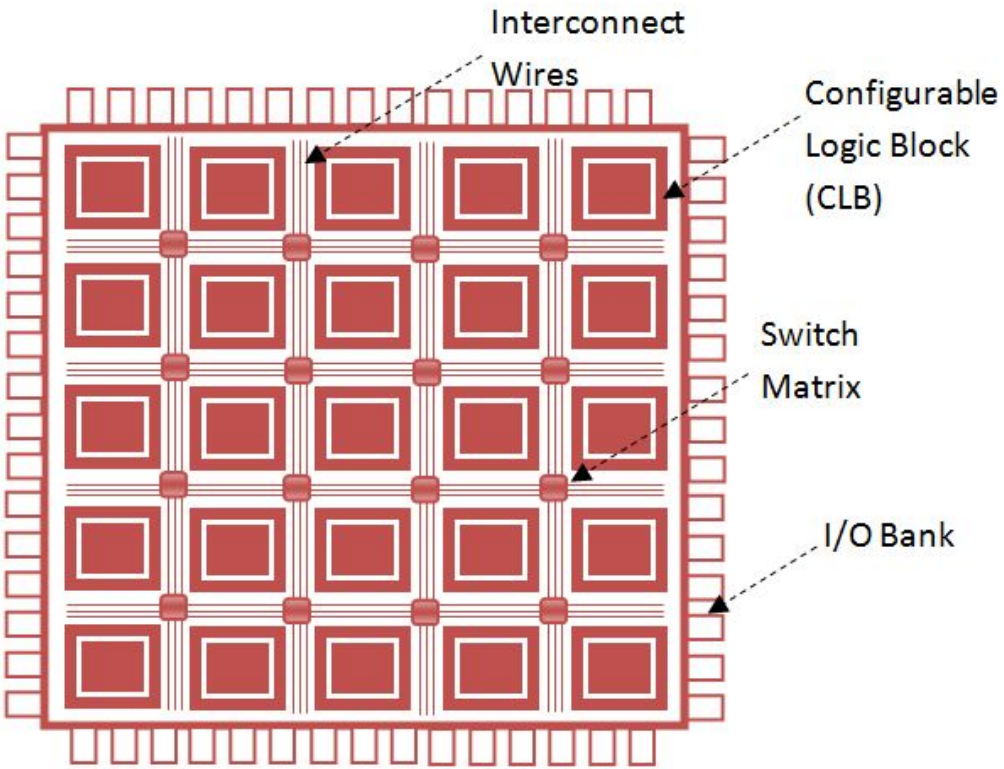



Cell
Cell
Cell
Cell
Cell
Cell
Cell


Block RAM
A/D
Cell
Cell
Cell
Cell
Cell
Cell
Cell
Cell
D/A


Block RAM
A/D
Cell
Cell
Cell
Cell
Cell
Cell
Cell
Cell
D/A
Block
Block
Fabric

Routing and Interconnect



FPGA Use Cases

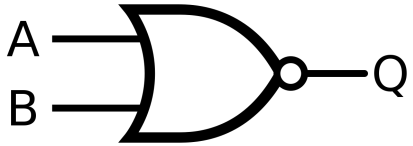
- Communication Equipment performing various Digital Signal Processing Tasks, such as filtering and computing the Fourier Transform to transform between the time and frequency domain and handle high data rates.
- Networking eg using embedded HARD IP in the FPGA, e.g. 100G and 400G Ethernet
- Quick way to prototype design as they can be reconfigured over and over again, ahead of producing an Application Specific Integrated Circuit (ASIC).
- Training Neural Networks
- Applications with massive IO requirements, e.g. Led Cubes
  https://twitter.com/esden/status/1210714541643812864?s=20&t=kiLFLvsxAShGoD0V5TwdBA

Why FPGA?

- Create Custom Reconfigurable digital Logic Circuits in an FPGA, to manipulate and perform computations on very large vectors e.g. 512 bit for massive data throughput
- Optimised digitized digital circuits for performing specialised computations like calculating the fast fourier transform in only a few clock cycles.
- You can use an FPGA as an external chip to supplement your processor (Hardware Acceleration)
- Modify a CPU e.g. Open Source RiscV processor to support some specialised function e.g. ability to multiply and add in a single clock cycle

How to create an FPGA design

Hardware Description Language (HDL)



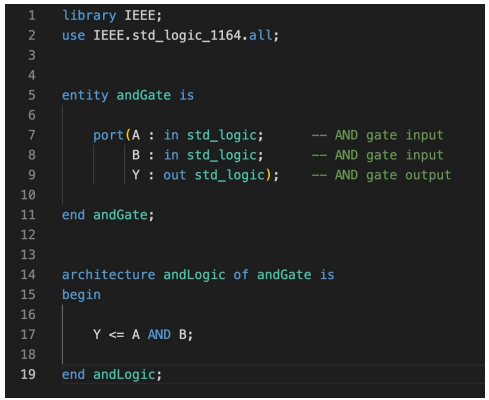
A	B	Q = A nor B
0	0	1
0	1	0
1	0	0
1	1	0

```
1   library IEEE;
2   use IEEE.std_logic_1164.all;
3
4   entity nor_gate is
5       port(A: in std_logic;
6            B: in std_logic;
7            Q: out std_logic);
8   end nor_gate;
9
10  architecture norLogic of nor_gate is
11   begin
12      Q <= not(A OR B);
13  end norLogic;
14
15
```

Hardware Design Languages (HDL)

VHDL

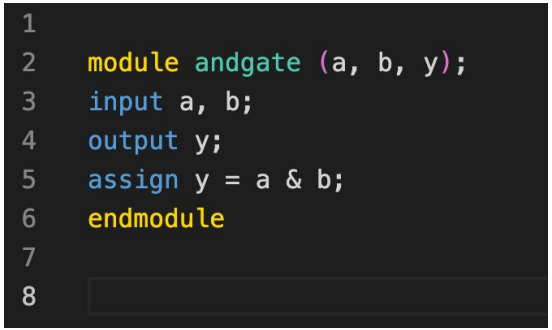
- 1983
- Department of Defence
- Strongly-typed
- More Verbose



```vhdl
library IEEE;
use IEEE.std_logic_1164.all;


entity andGate is

    port(A : in std_logic;      -- AND gate input
         B : in std_logic;      -- AND gate input
         Y : out std_logic);    -- AND gate output

end andGate;


architecture andLogic of andGate is
begin

    Y <= A AND B;

end andLogic;
```

Verilog

- 1984
- Gateway Design Automation
- Weakly-typed
- C-like



```verilog

module andgate (a, b, y);
input a, b;
output y;
assign y = a & b;
endmodule


```

HDL Languages continued

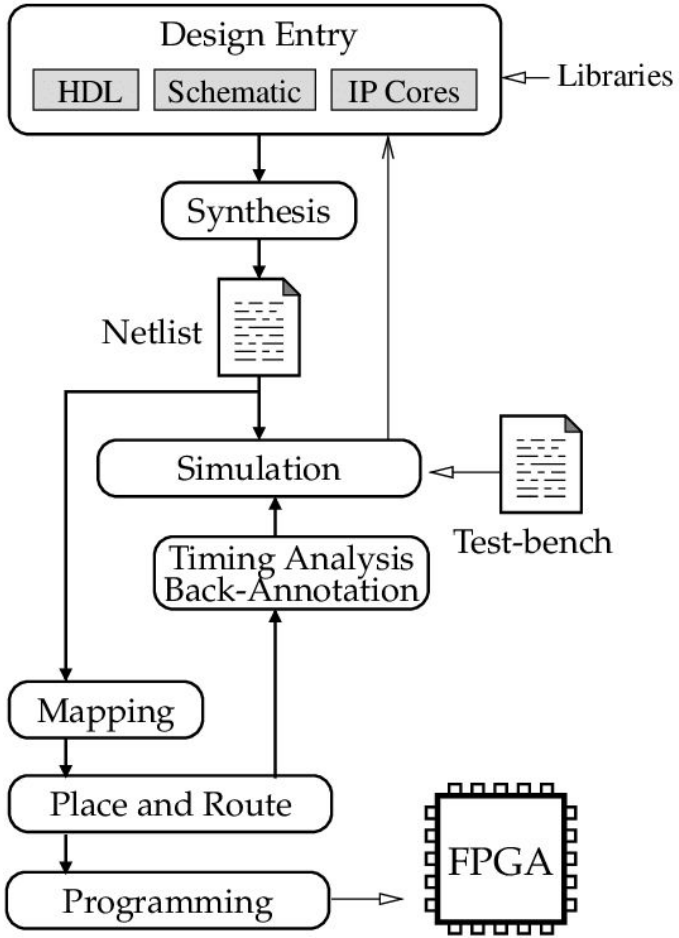
- System Verilog
  - Extends the functionality of Verilog-2005
  - Adds features for testing/verification (test benches)
- High Level Synthesis (HLS)
  - Automatically convert a functional design to RTL
  - C, C++, MATLAB, etc

https://docs.xilinx.com/r/en-US/ug1399-vitis-hls/Introduction-to-Vitis-HLS

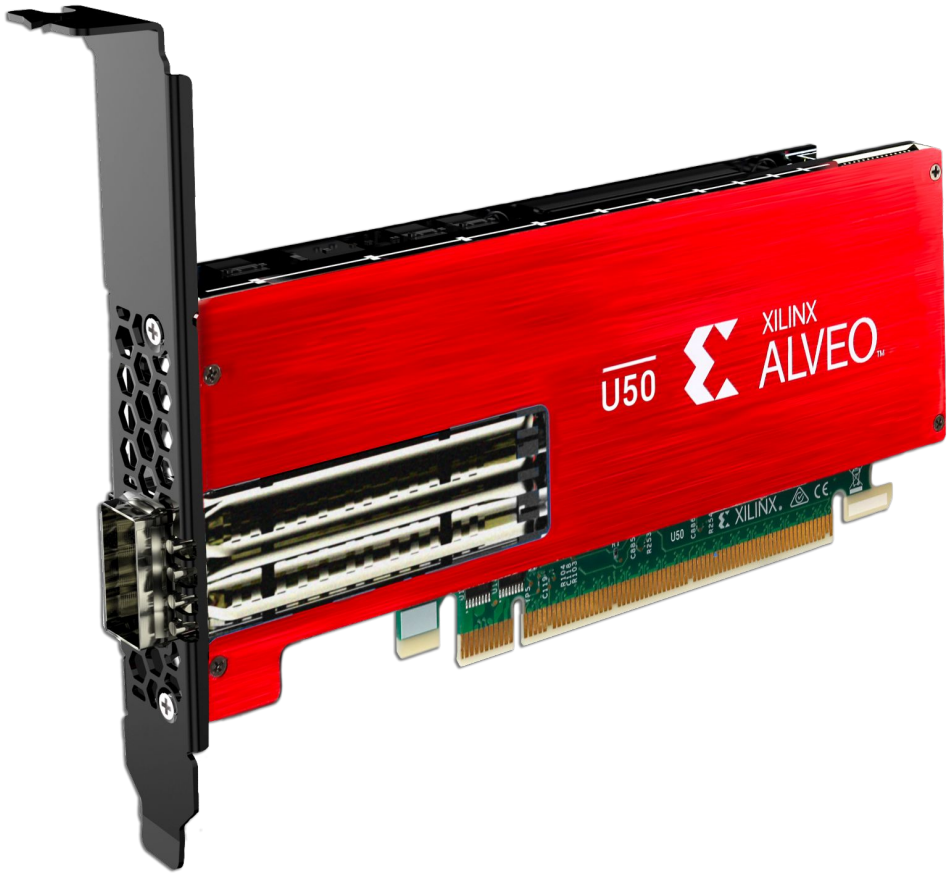
FPGA Design Primitives

- LUT’s
- D-Type Flip Flops
- Latches (not good design practice)
- MUX’s (made of ‘and’ & ‘or’ gates)
- RAMs (Distributed using LUT’s as RAM)
- Dual Port RAMs (Block RAM using embedded SRAM)
- RAMs (High Bandwidth Memory (HBM) DDR based on chip)
- FIFO’s (width and depth)
- DSP blocks (27b x 18b Multiply and Accumulate)
- HARD IP (eg PCIe, 100G Ethernet MAC)
- SerDes (GTY 12.5 Gbps per differential pair)

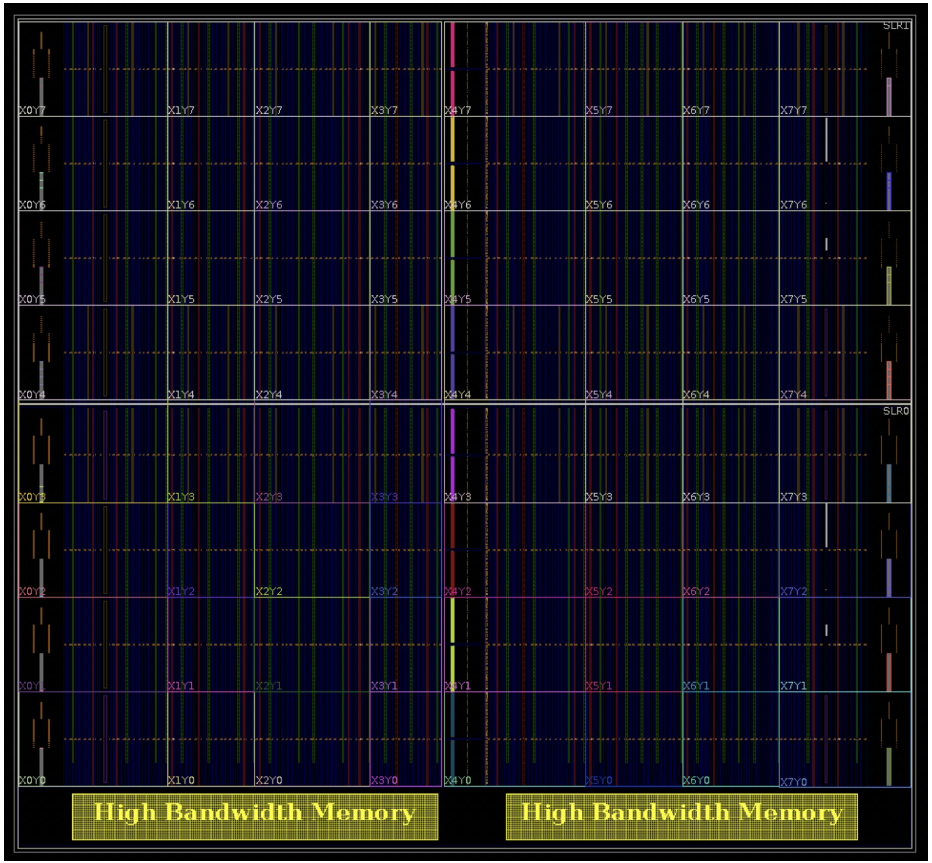
FPGA Design Flow



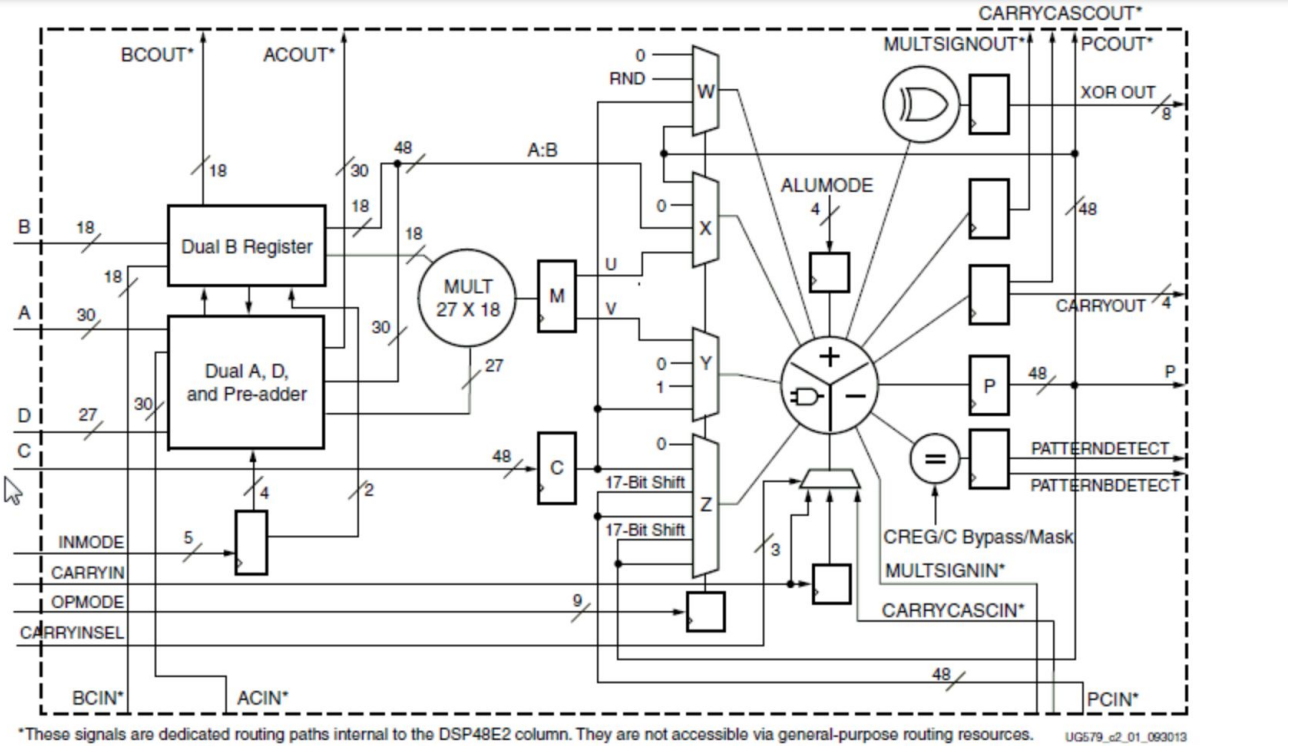
Xilinx Alveo



An Array of Programmable Gates



DSP48E2



*These signals are dedicated routing paths internal to the DSP48E2 column. They are not accessible via general-purpose routing resources.

UG579_c2_01_093013

[1] http://archive.computerhistory.org/resources/text/Knuth_Don_X4100/PDF_index/k-8-pdf/k-8-r5367-1-ENIAC-circuits.pdf

[2] http://www.qrp.gr/technology/logic/5484436-Basic-Principles-of-Vacuum-Tube-Logic-Circuits.pdf

[3] https://en.wikipedia.org/wiki/Apollo_Guidance_Computer

[4] https://www.xilinx.com/products/boards-and-kits/alveo/u50.html

[5] https://www.youtube.com/watch?v=lLg1AgA2Xoo