FPGAs

What do those letters mean?

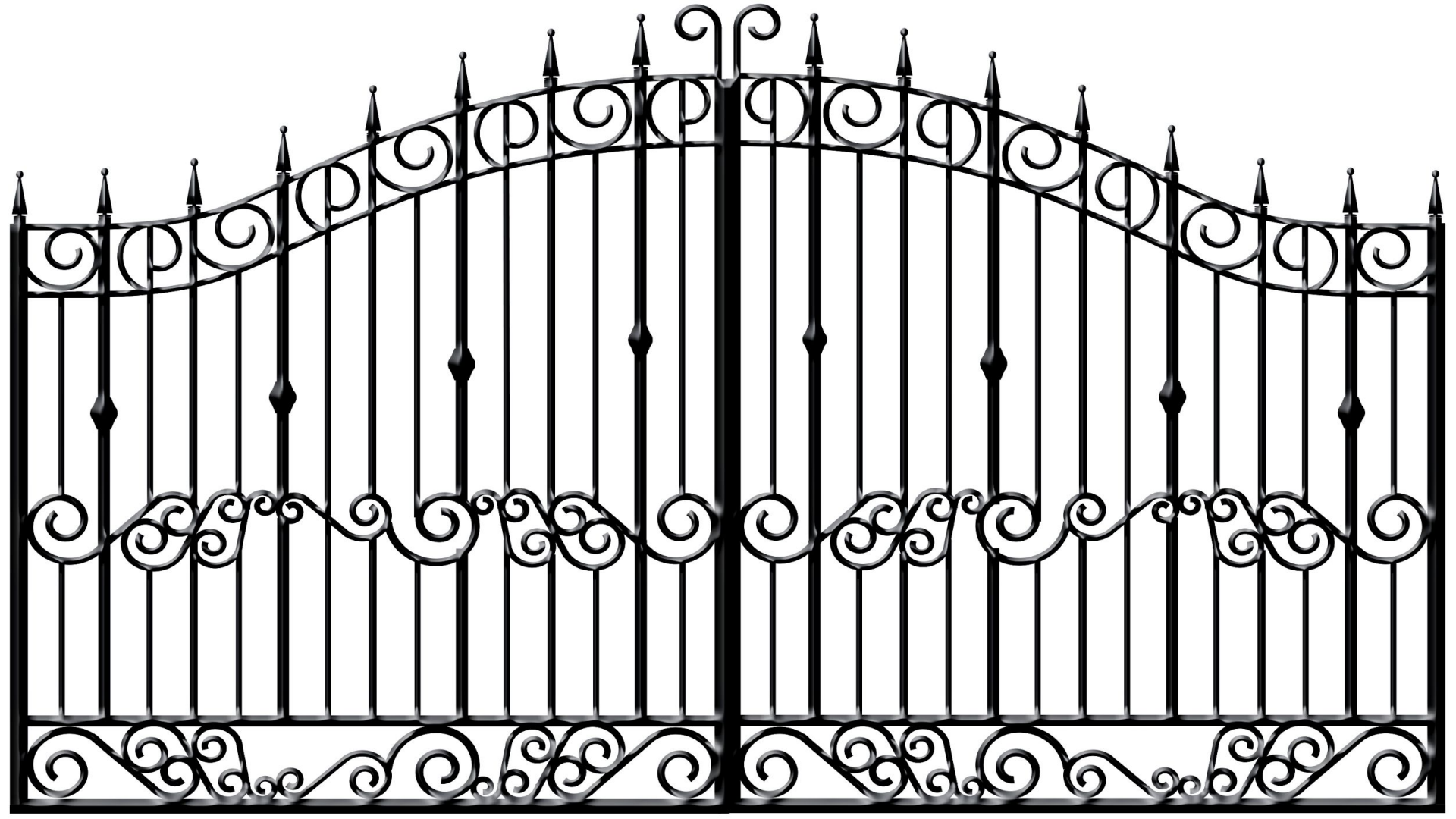
Field

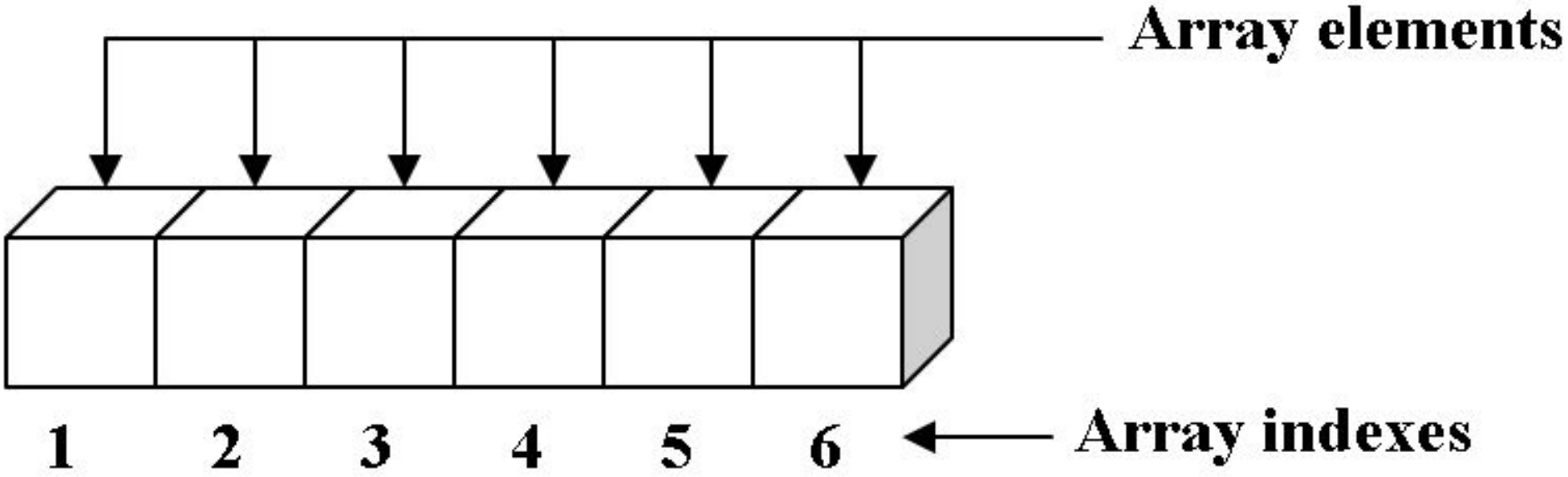
Programmable

Gate

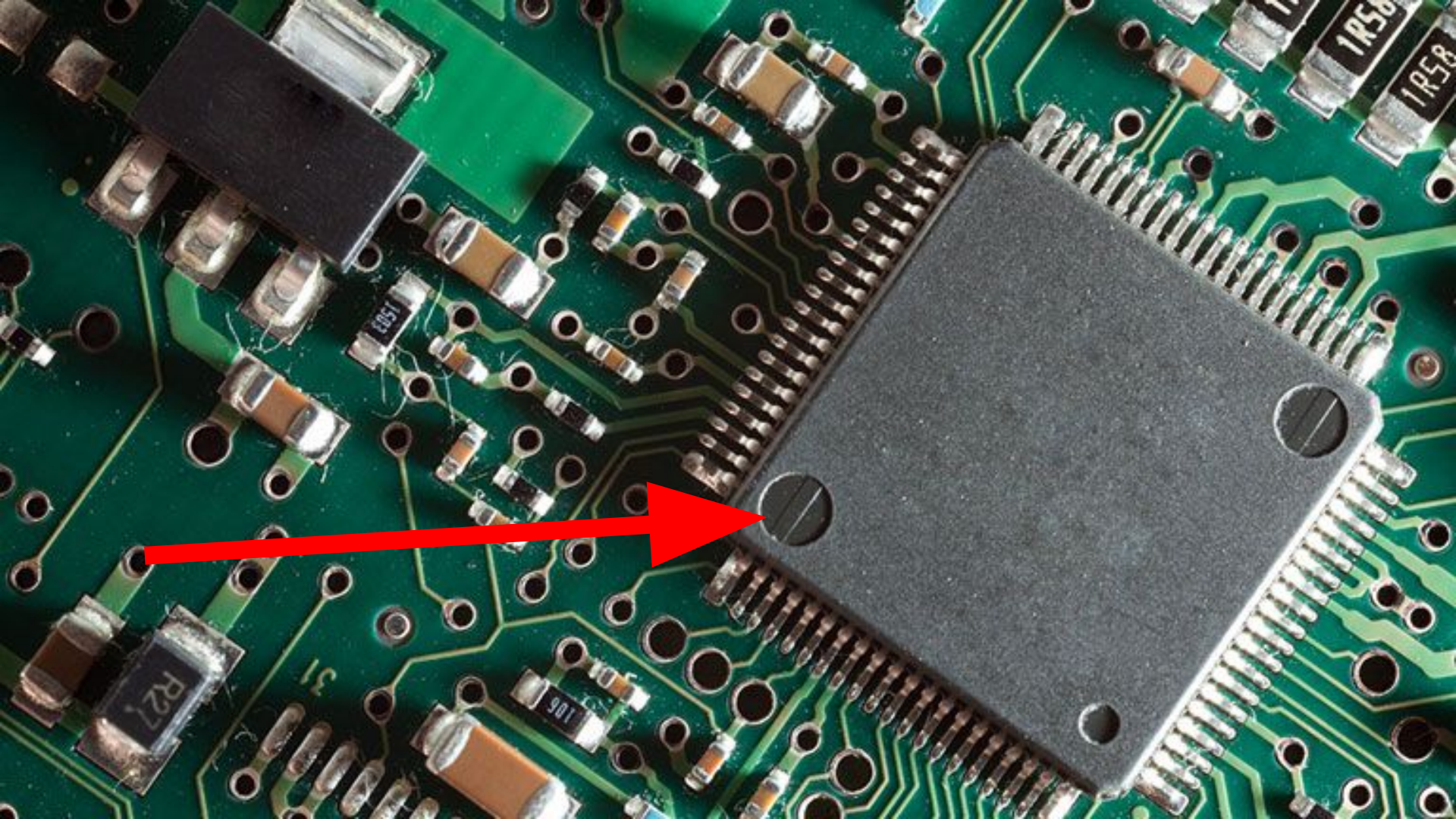
Array

tempString = tempString
.join(
tempString = tempString
value dataCal = "".join("{:02x}
,str(dataCal)) string4rep
ricName = line print(
flagCheckRicname
"RT_CHAIN_EVENT" flagCheckRicname
"RT": buff = myEvent+"\t"+RTRDATA
(path) filename = "vtest.spec"
Input4RTAvTEST',ignore_errors=
xlsx'): fName = os.path.bas





One-dimensional array with six elements


R21

What this thing does

Integrated Circuit or “chip”

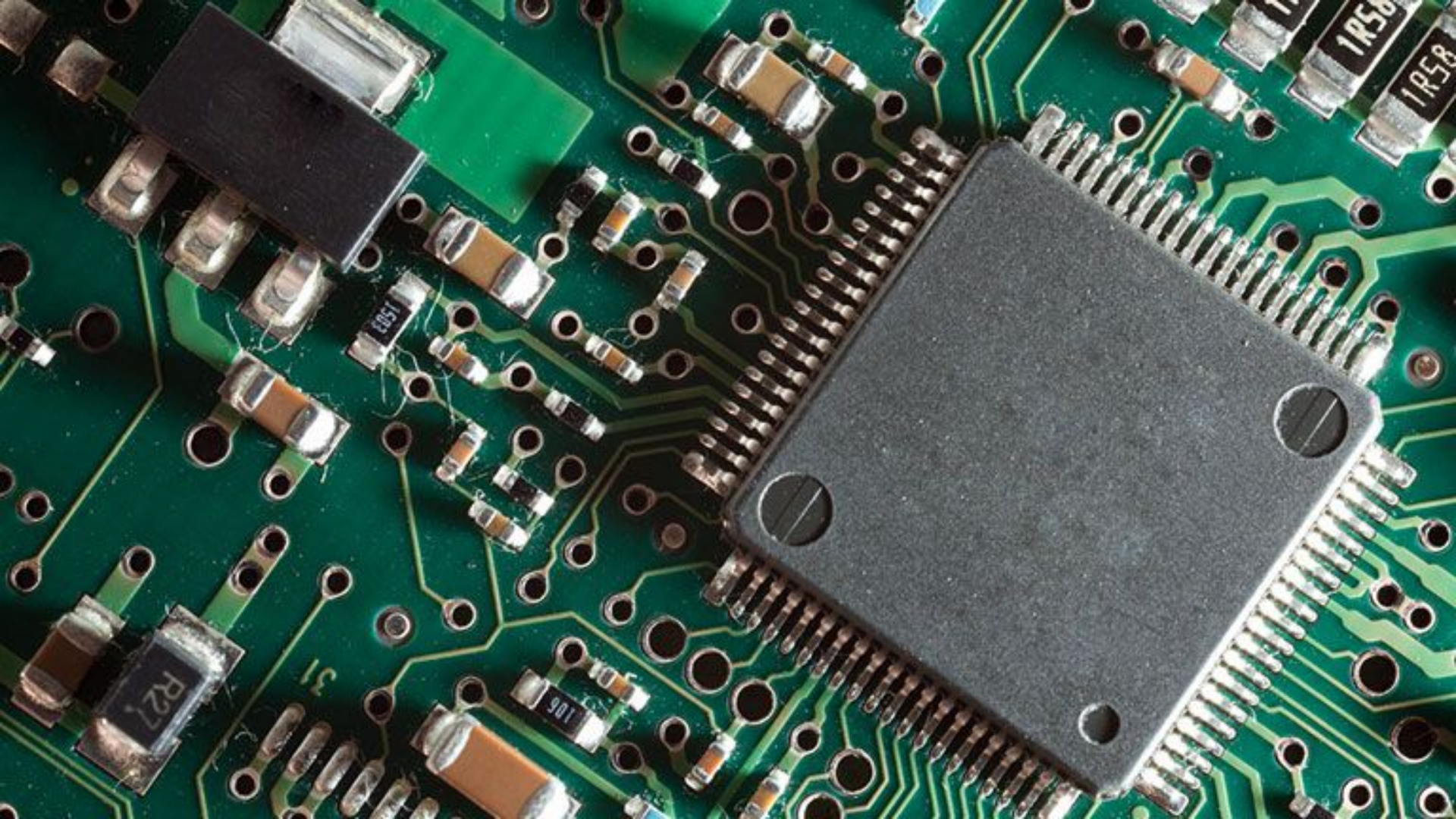
Field programmable: firmware can be modified “in the field”

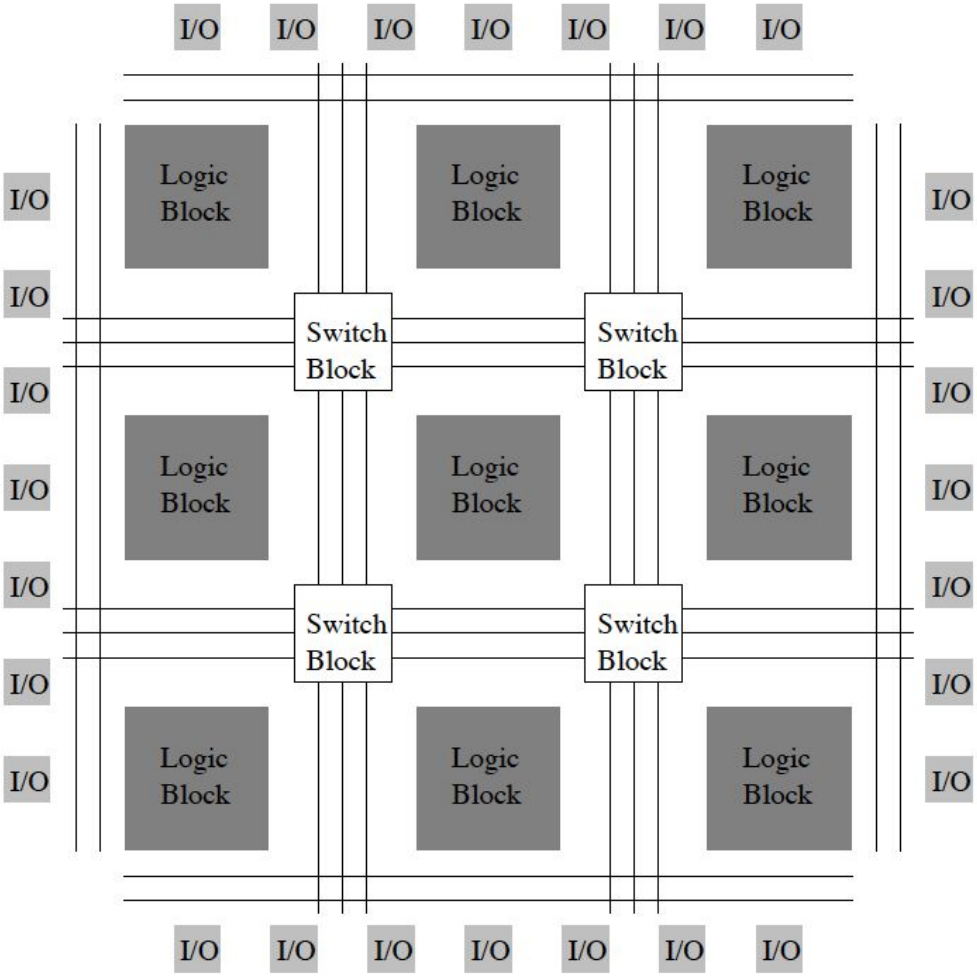
It has no intended function

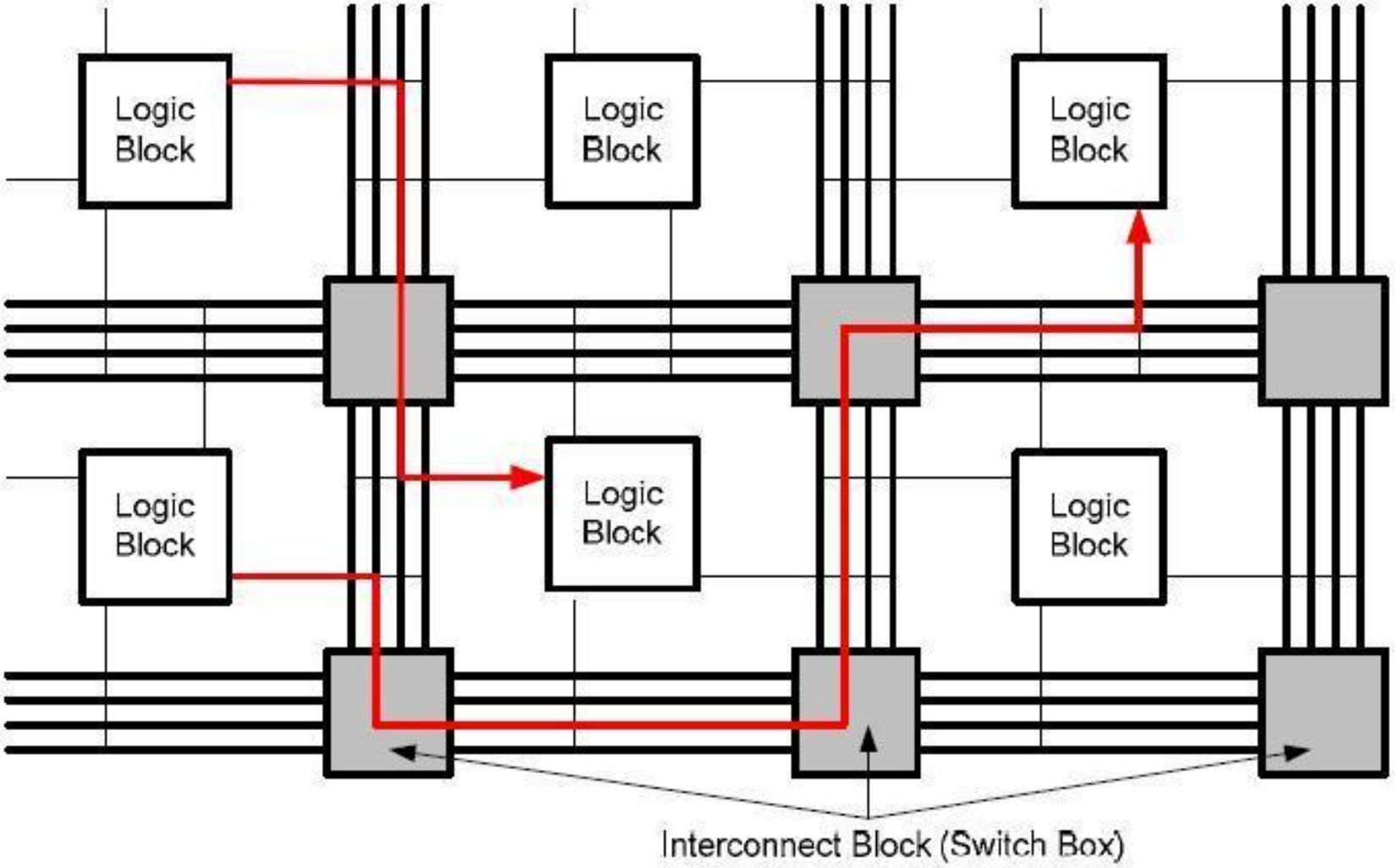
Closest you can get to designing your own chip

Design and implement any digital function

Flexible




I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
Logic Block
Logic Block
Logic Block
Switch Block
Switch Block
Logic Block
Logic Block
Logic Block
Switch Block
Switch Block
Logic Block
Logic Block
Logic Block

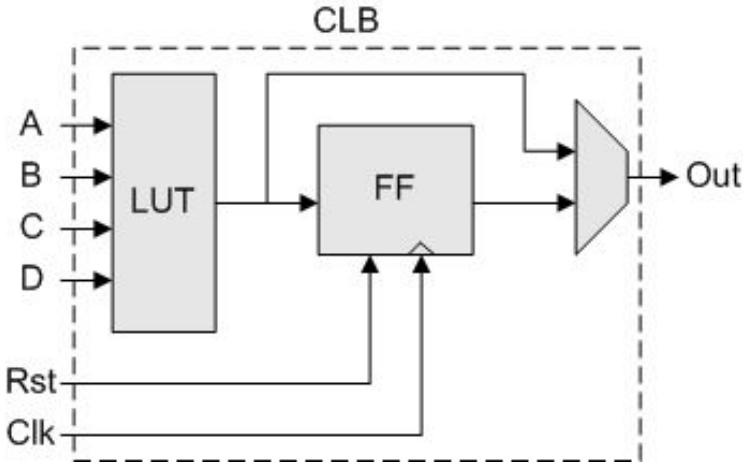

Logic Block
Logic Block
Logic Block
Logic Block
Logic Block
Logic Block
Interconnect Block (Switch Box)

Configurable Logic Block

There are more complex CLBs than this example...

LUT: Lookup table with four inputs and configuration fuses inside that can be programmed to perform a particular function.

FF: One or more flip flops connected to clock line and reset line. It has Q and not Q outputs to select which output is desired in latched/clocked logic.



I/O Blocks

Handles input/output and connects to pins on the chip

Complicated blocks, not just a buffer

Programmed to perform functions

Examples: Tristate input, differential pair drivers, different load voltage standards, flip flops and latches to drive DDR memory

Configuration

Configuration logic determined by user

Hooked up to external configuration flash memory

All states of fuses in switch blocks, CLBs, and IOBs

When FPGA is turned off, the programmed behavior disappears

Must load from external configuration memory to set everything up

Advantages

Do anything in digital domain

Super Fast, can be real time

Not a fixed processor

Why is this applicable to Parallel Programming?

Massively Parallel

Generally better than microcontrollers for these type applications

Microcontrollers have a lot of bottleneck as they process everything in sequence

FPGAs can process multiple inputs into multiple outputs at the same time

Links for references and more info

EEVblog #496 - What Is An FPGA?: https://youtu.be/gUsHwi4M4xE

Various projects with FPGAs: http://www.fpga4fun.com/

Course solely on FPGAs: http://venividiwiki.ee.virginia.edu/mediawiki/index.php/ClassECE6332Fall15GroupFPGA

Simplified block diagram of FPGAs. Source: B. Nezamfat dissertation