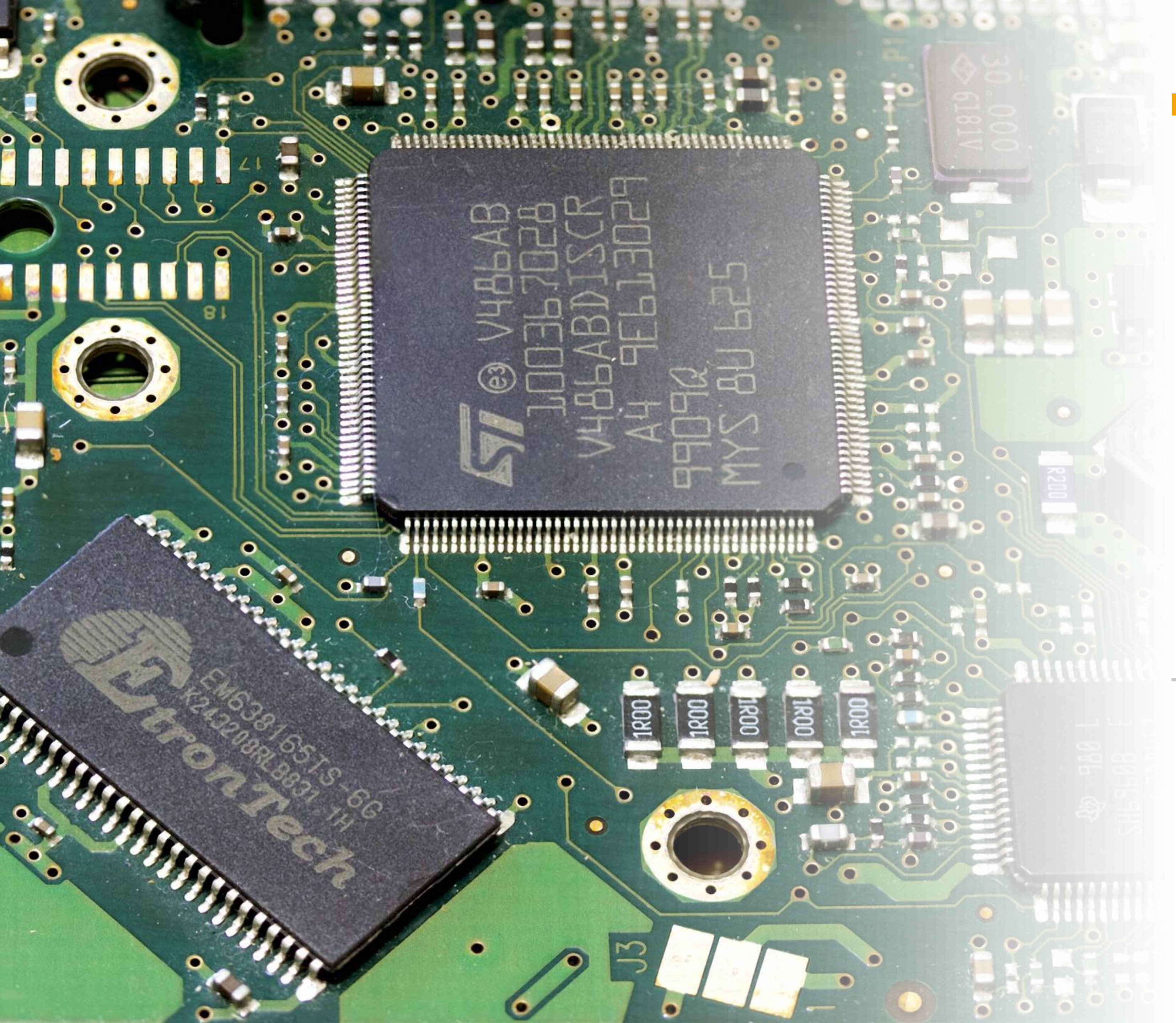


Digital Logic and System Design

6. FPGA

COL215, I Semester 2022-2023
Venue: LHC 111
'E' Slot: Tue, Wed, Fri 10:00-11:00

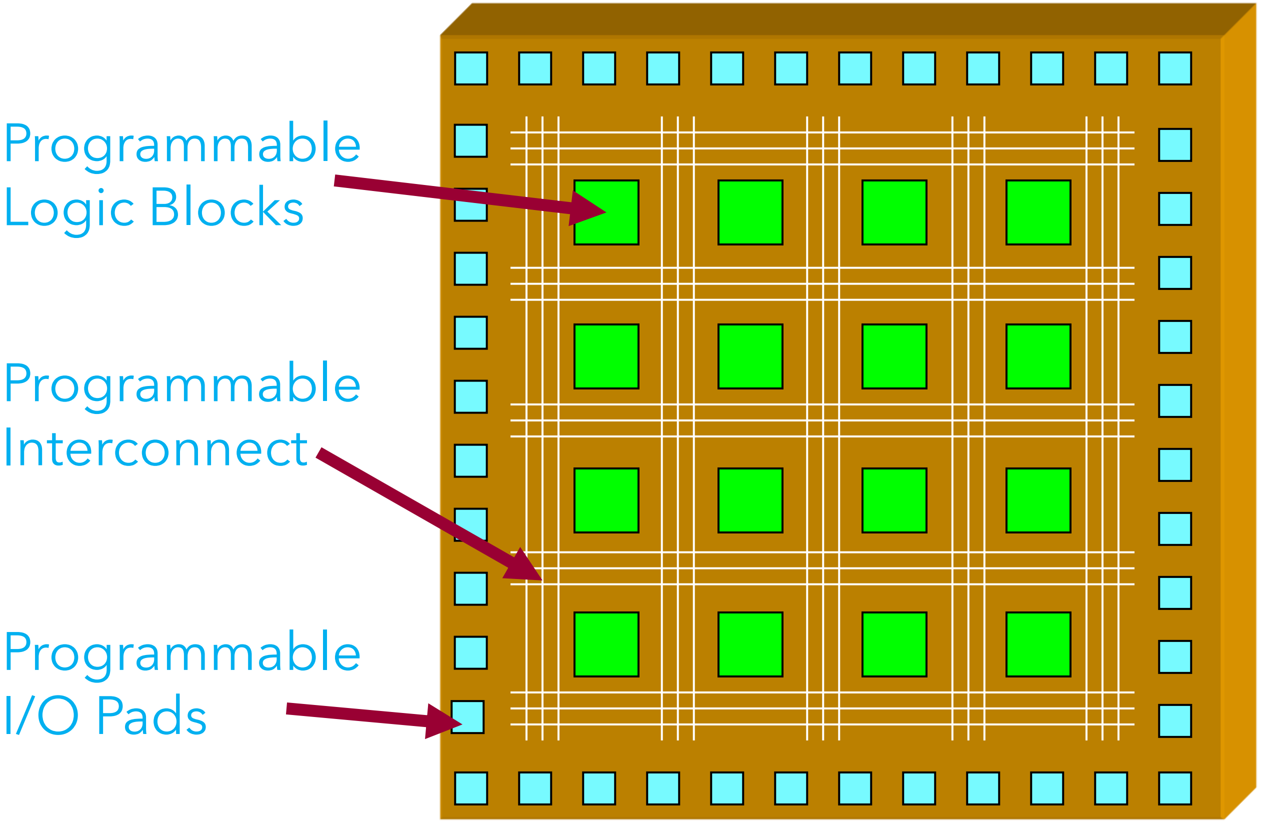
Instructor: Preeti Ranjan Panda
panda@cse.iitd.ac.in
www.cse.iitd.ac.in/~panda/
Dept. of Computer Science & Engg., IIT Delhi

Field Programmable Gate Array

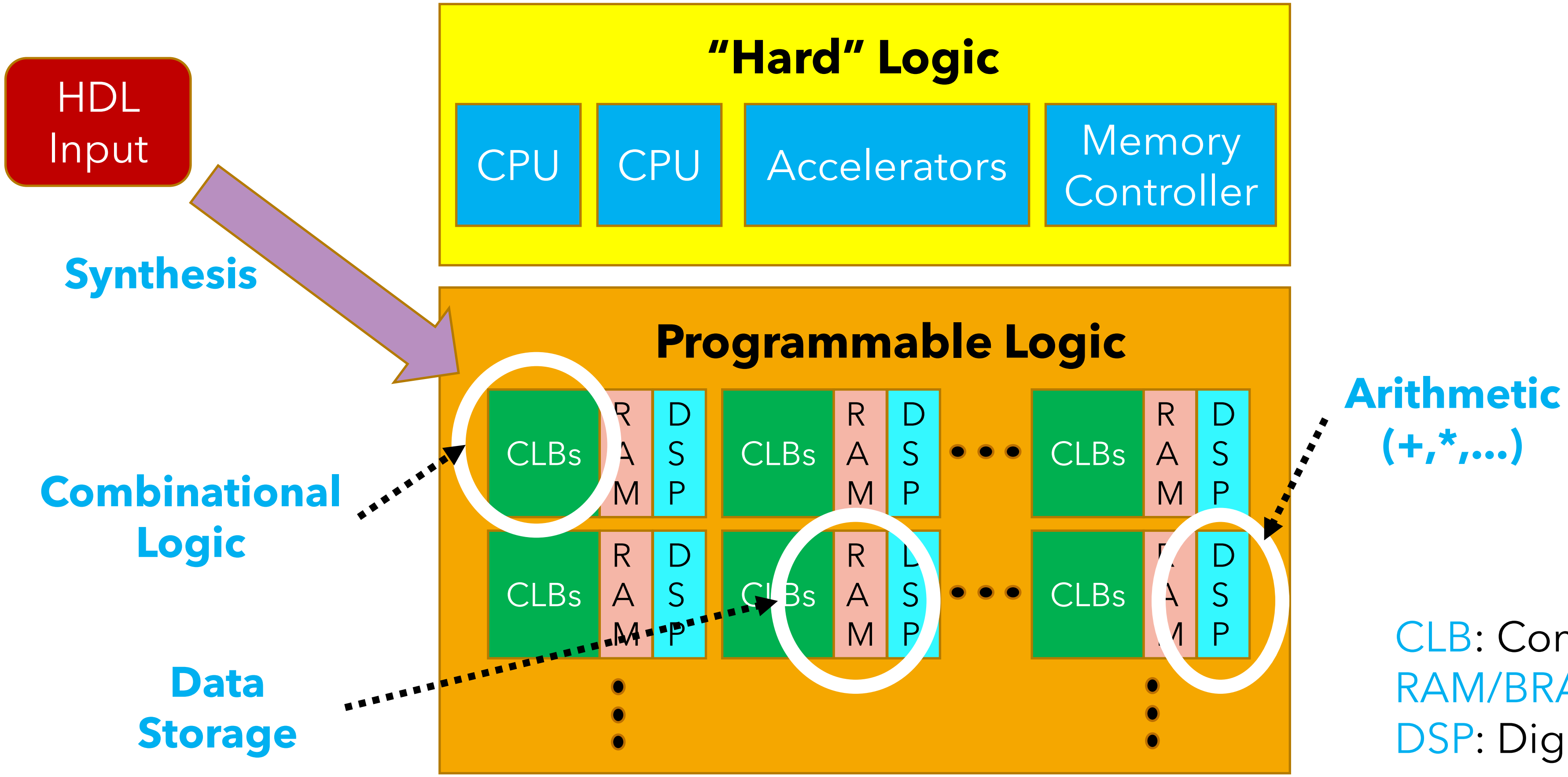
- **Field Programmable**
  - Customised/Programmed by designer
  - vs. “Mask Programmable”: customised by foundry
- **Gate Array**
  - Design methodology: pre-fabricated gates, connected later
- Customisation/Programming process **simple/cheap**
  - Design turnaround time: minutes/hours
- FPGA chips **produced in bulk**
  - independent of functionality

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence the term field-programmable. The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC). Circuit diagrams were previously used to specify the configuration, but this is increasingly rare due to the advent of electronic design automation tools.
FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects allowing blocks to be wired together. Logic blocks can be configured to perform complex combinational functions, or act as simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.[1] Many FPGAs can be reprogrammed to implement different logic functions, allowing flexible reconfigurable computing as performed in computer software.
FPGAs have a remarkable role in embedded system development due to their capability to start system software development simultaneously with hardware, enable system performance simulations at a very early phase of the development, and allow various system trials and design iterations before finalizing the system architecture.[2]

Classical FPGA architecture

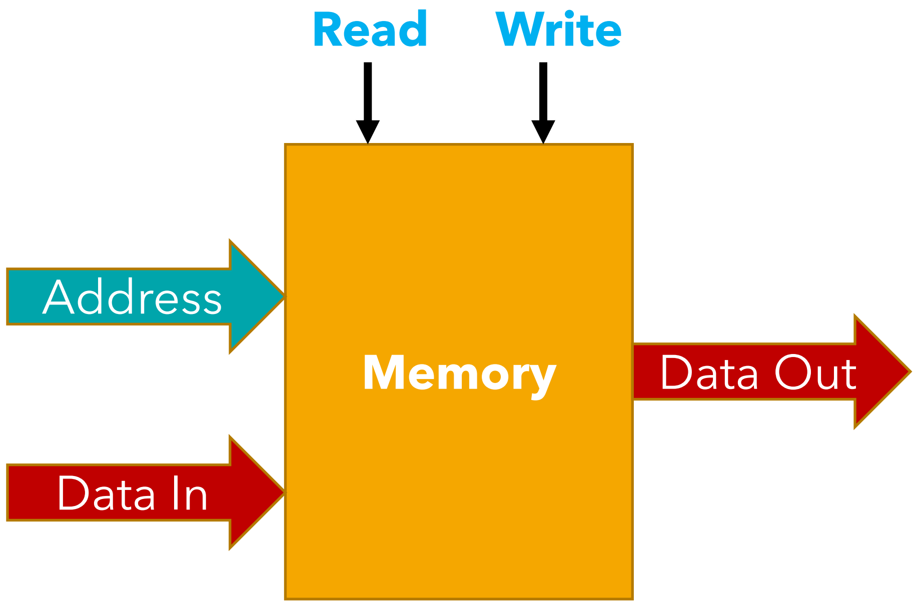


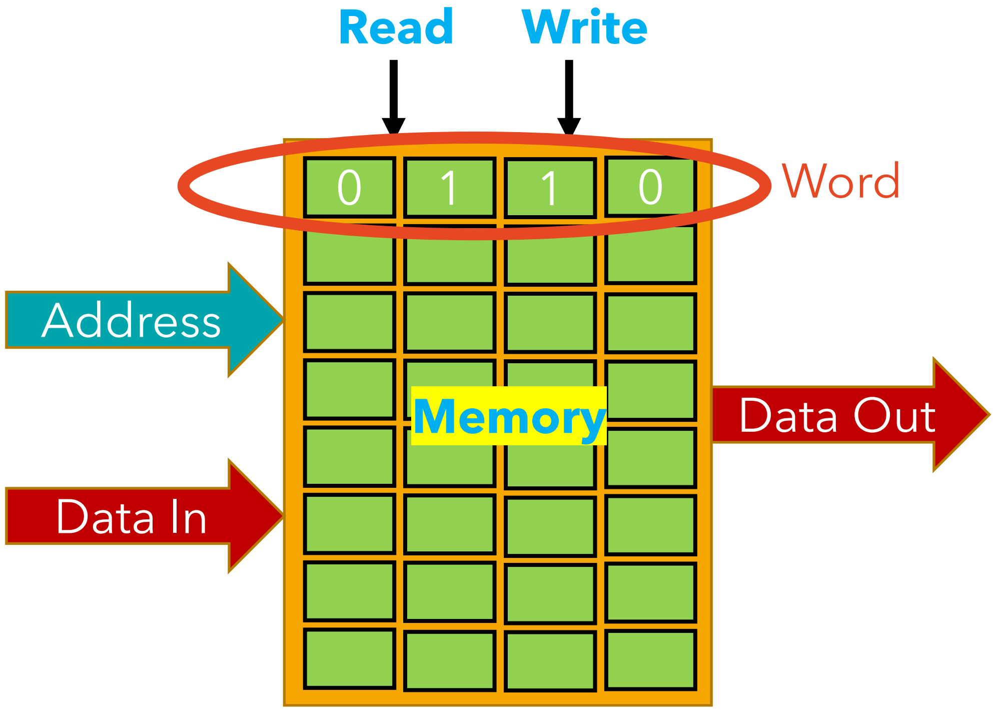
Modern FPGA Architecture:
Xilinx Zynq 7000 Series



CLB: Combinational Logic Block
RAM/BRAM: Random Access Memory
DSP: Digital Signal Processing unit

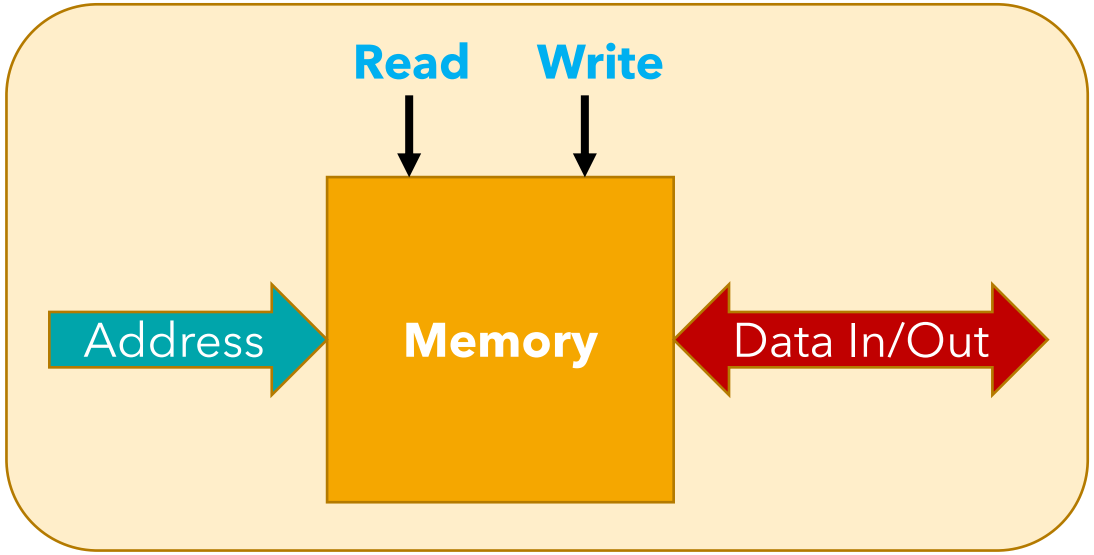
Memory



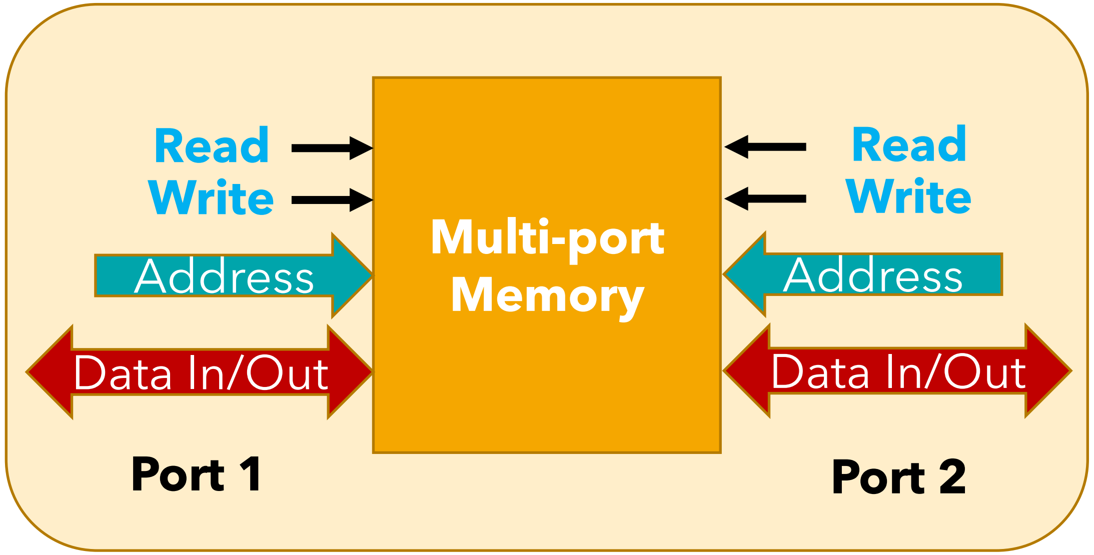


Word Size: 4 bits
Memory Size: 8 words

Memory: General Architectures



Bi-directional Data Bus

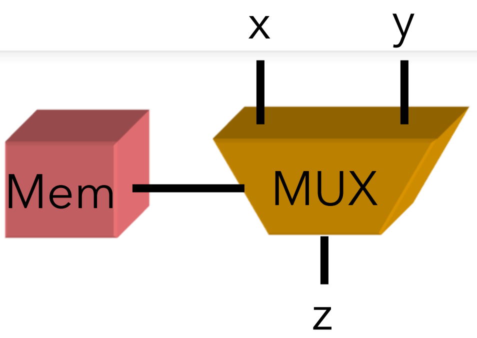


Multiple Simultaneous Accesses

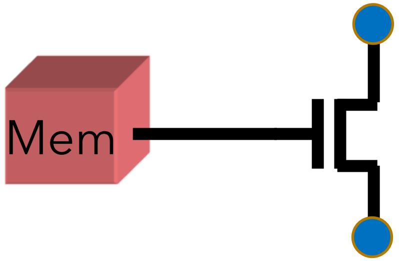
Read & Write on Same Address: CHAOS

FPGA Programming: SRAM

- **SRAM**: Static Random Access Memory
  - Stores a bit (0/1)
- **Programming**:
  - writing 0 or 1 into SRAM cell
  - this, in turn, causes:
    - selections
    - connections



Programming the logic



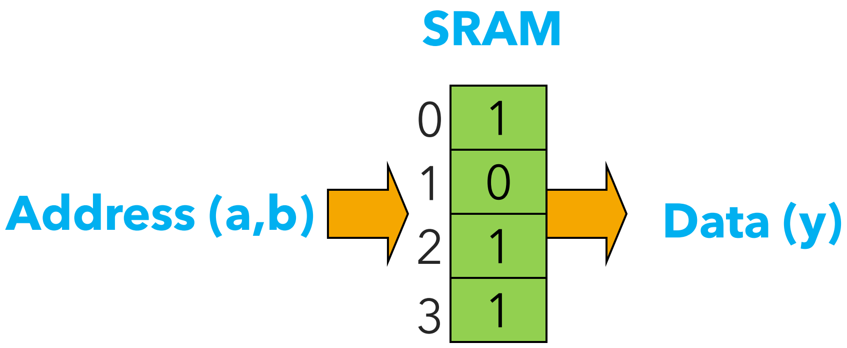
Programming the interconnection (short/open)

SRAM-based FPGA

- Each logic block consists of an SRAM
- An SRAM with 2^n bits can implement ANY function of n inputs
  - Use inputs as address
  - Store value in locations

Truth Table of Function

a b	y
0 0	1
0 1	0
1 0	1
1 1	1



0	1
1	0
2	1
3	1

Xilinx 7-Series Architecture

- 64-bit Look-up Tables (LUT)
    - All functions of 6 inputs
- What if we wanted 2 functions of 5 inputs instead?