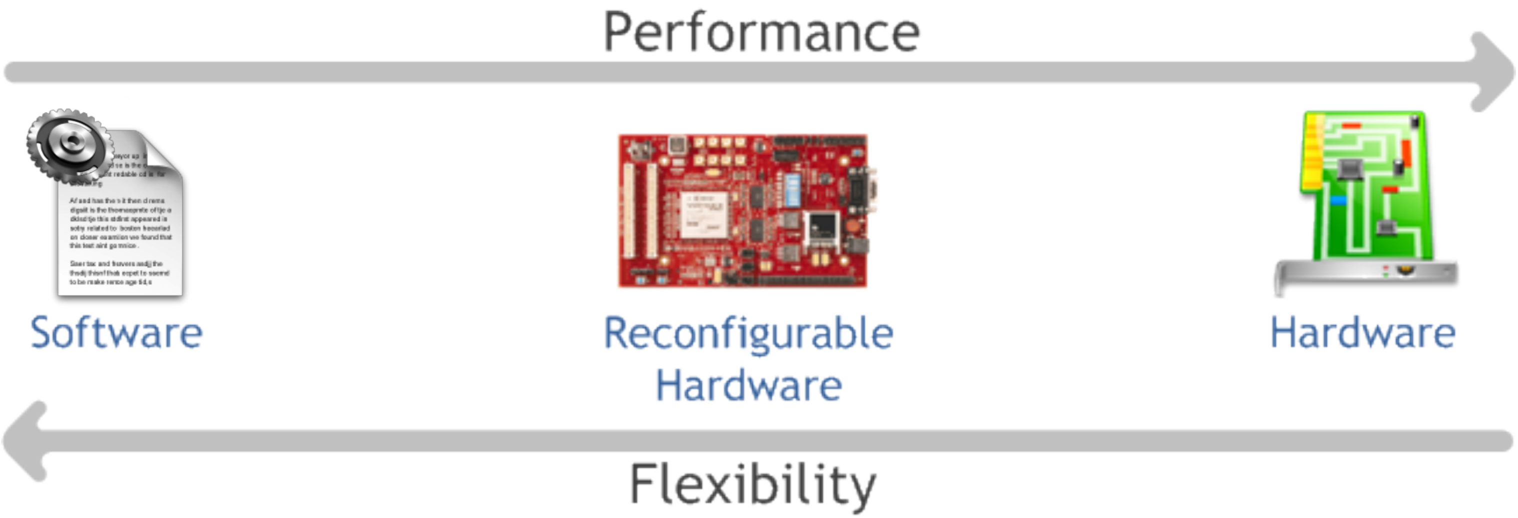
A very short introduction to FPGAs

Alessandro A. Nacci
alessandro.nacci@polimi.it

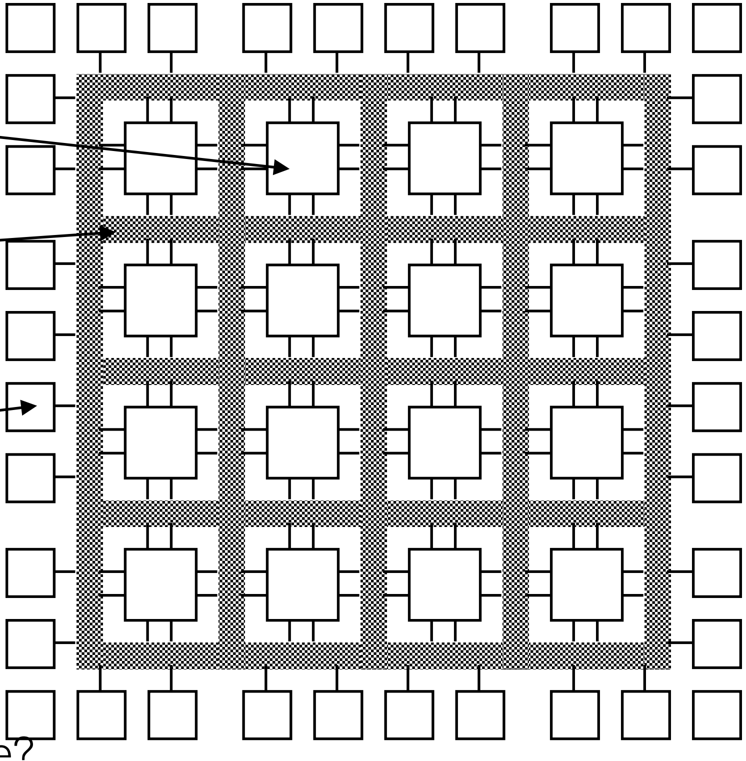
Reconfigurable Hardware



“Reconfigurable computing is intended to fill the gap between hardware and software, achieving potentially much higher performance than software, while maintaining a higher level of flexibility than hardware”

(K. Compton and S. Hauck, *Reconfigurable Computing: a Survey of Systems and Software*, 2002)

Field-Programmable Gate Arrays



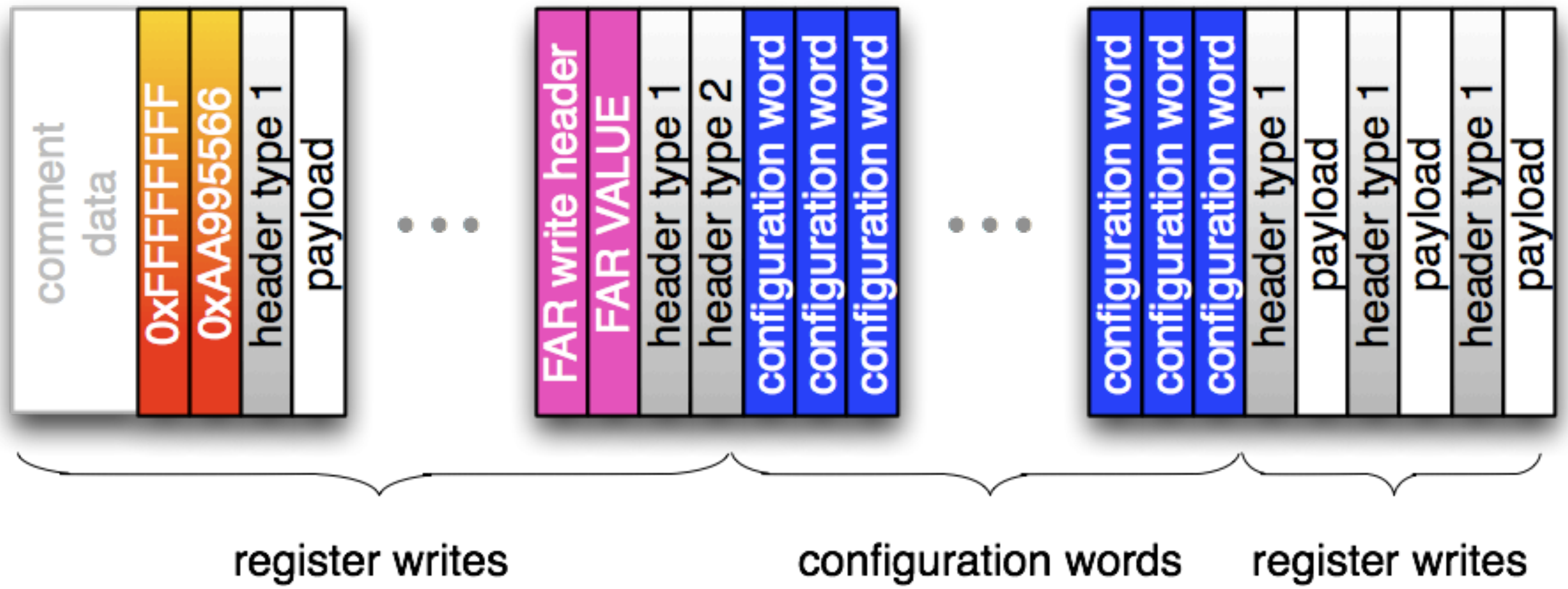
- Logic blocks
  - to implement combinational and sequential logic
- Interconnect
  - wires to connect inputs and outputs to logic blocks
- I/O blocks
  - special logic blocks at periphery of device for external connections

- Key questions:
  - how to make logic blocks programmable?
  - how to connect the wires?
  - *after the chip has been fabbed*

Configuration Bitstream

- The configuration bitstream must include ALL CLBs and SBs, even unused ones
- CLB0: 00011
- CLB1: ?????
- CLB2: 01100
- CLB3: XXXXX
- SB0: 000000
- SB1: 000010
- SB2: 000000
- SB3: 000000
- SB4: 000001

The configuration bitstream

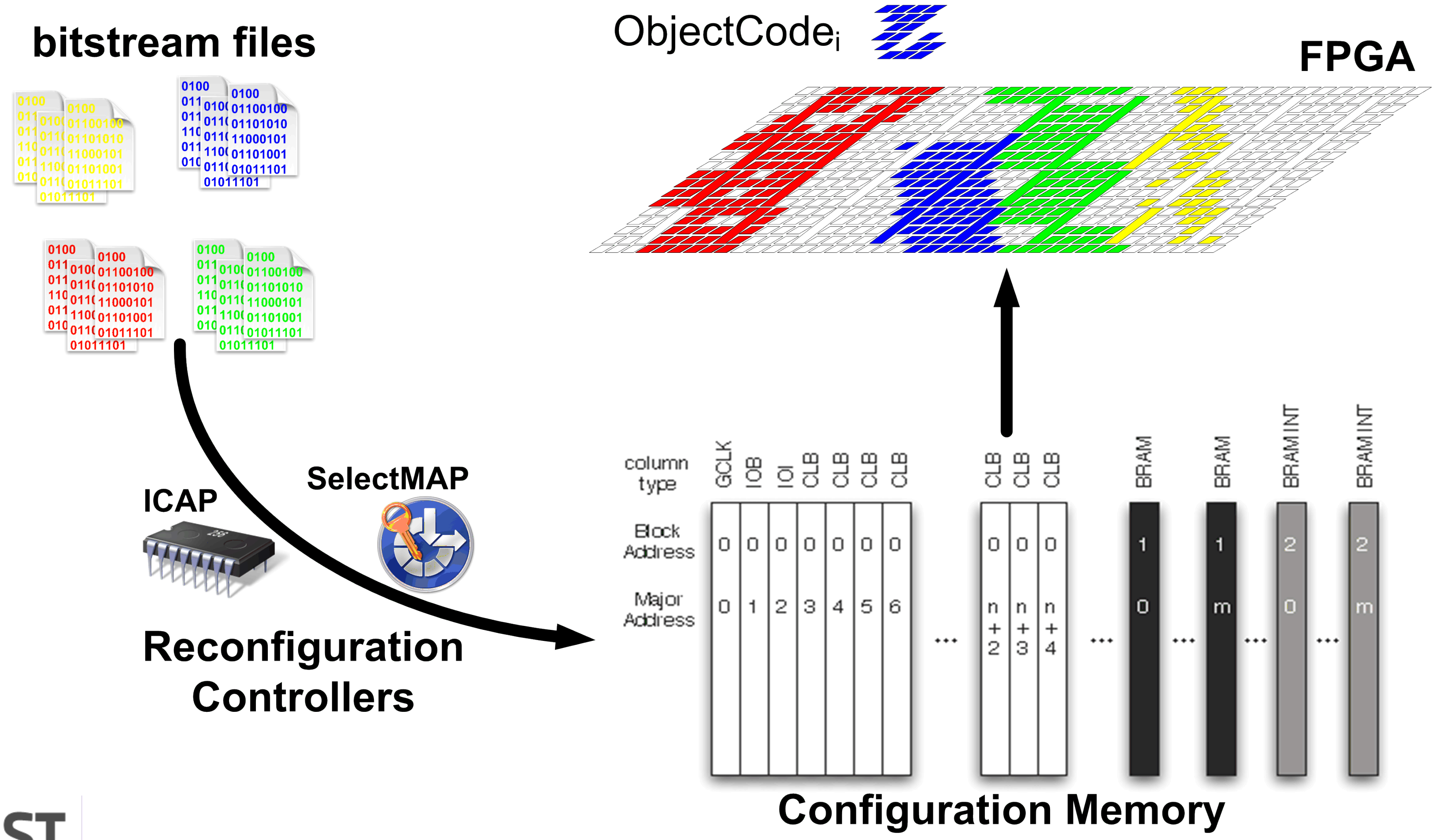


- Occupation must be determined only on the basis of
  – Number of configuration words
  – Initial Frame Address Register (FAR) value

Some Definitions

- Object Code: the executable active physical (either HW or SW) implementation of a given functionality
- Core: a specific representation of a functionality. It is possible, for example, to have a core described in VHDL, in C or in an intermediate representation (e.g. a DFG)
- IP-Core: a core described using a HD Language combined with its communication infrastructure (i.e. the bus interface)
- Reconfigurable Functional Unit: an IP-Core that can be plugged and/or unplugged at runtime in an already working architecture
- Reconfigurable Region: a portion of the device area used to implement a reconfigurable core

Xilinx FPGA and Configuration Memory



Reconfiguration Controllers

column type	GCLK	IOB	IOI	CLB	CLB	CLB	CLB	...	CLB	CLB	CLB	...	BRAM	...	BRAM	...	BRAM INT	...	BRAM INT
Block Address	0	0	0	0	0	0	0		0	0	0		1		1		2		2
Major Address	0	1	2	3	4	5	6		n + 2	n + 3	n + 4		0		m		0		m

Configuration Memory

FPGA EDA Tools

- Must provide a design environment based on digital design concepts and components (gates, flip-flops, MUXs, etc.)
- Must hide the complexities of placement, routing and bitstream generation from the user. Manual placement, routing and bitstream generation is infeasible for practical FPGA array sizes and circuit complexities.

Computer-aided Design

- Can't design FPGAs by hand
  - way too much logic to manage, hard to make changes
- Hardware description languages
  - specify functionality of logic at a high level
- Validation - high-level simulation to catch specification errors
  - verify pin-outs and connections to other system components
  - low-level to verify mapping and check performance
- Logic synthesis
  - process of compiling HDL program into logic gates and flip-flops
- Technology mapping
  - map the logic onto elements available in the implementation technology (LUTs for Xilinx FPGAs)

CAD Tool Path (cont’d)

- Placement and routing
  - assign logic blocks to functions
  - make wiring connections
- Timing analysis - verify paths
  - determine delays as routed
  - look at critical paths and ways to improve
- Partitioning and constraining
  - if design does not fit or is unroutable as placed split into multiple chips
  - if design it too slow prioritize critical paths, fix placement of cells, etc.
  - few tools to help with these tasks exist today
- Generate programming files - bits to be loaded into chip for configuration

QUESTIONS?