FPGA based Design

Landscape of hardware design

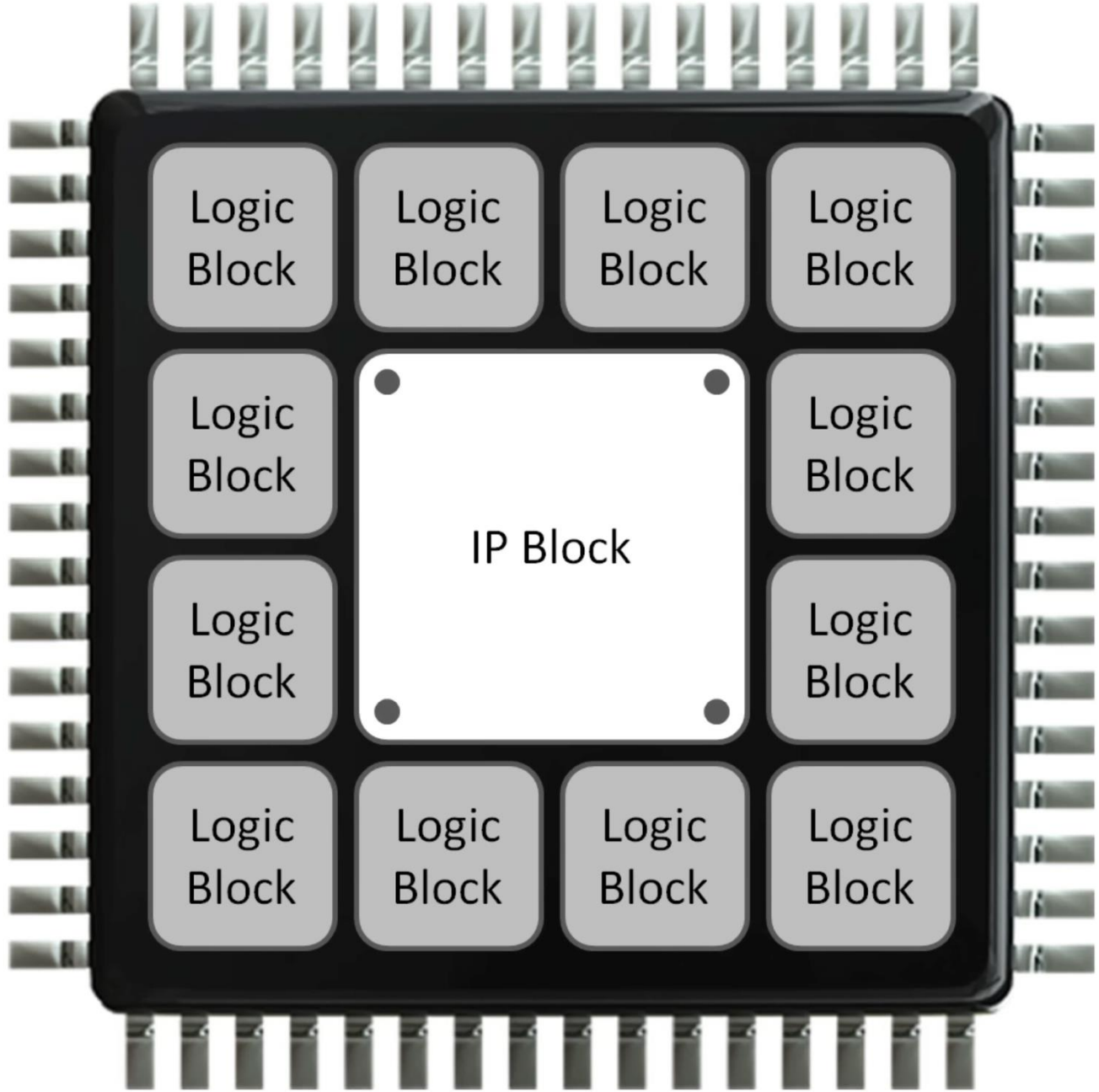
Generality provides flexibility at the cost of complexity

Comparison of CPU, FPGA, and Microcontroller

FPGAs for Emulation

Why use an FPGA?

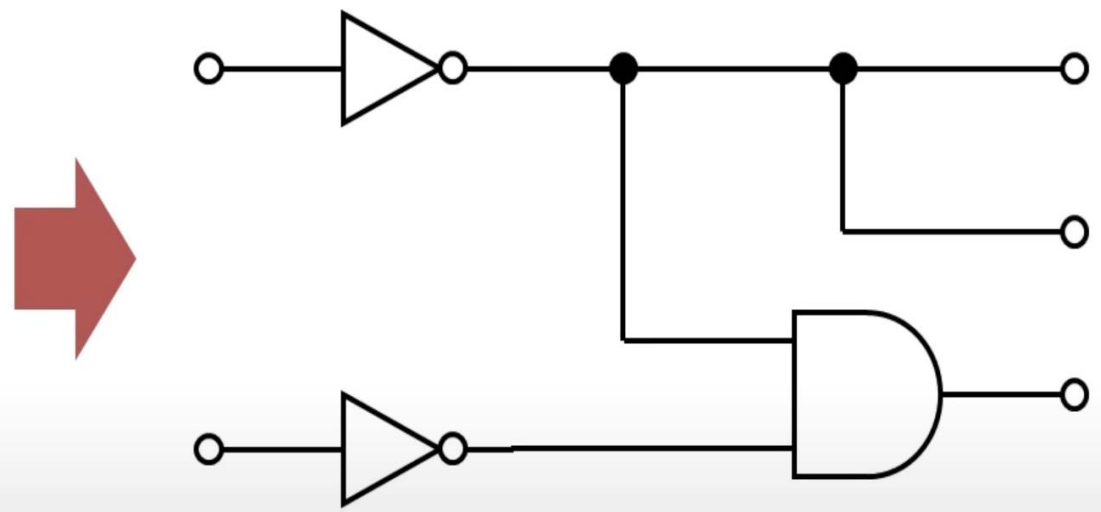
- Custom, reconfigurable digital logic circuits
- Add peripherals
- Modify CPU
- Speed for specific computations
- Prototype (e.g. path to production for ASIC)


Logic Block
Logic Block
Logic Block
Logic Block
Logic Block
IP Block
Logic Block
Logic Block
Logic Block
Logic Block
Logic Block
Logic Block
Logic Block
Logic Block

Hardware Description Language (HDL)

Example of Verilog

```
1  // Module: button 0 lights up 2 LEDs, button 0 and 1 light up another LED
2  module and_gate (
3
4      // Inputs
5      input   [1:0]   pmod,
6
7      // Output
8      output  [2:0]   led
9  );
10
11     // Wire (net) declarations (internal to module)
12     wire not_pmod_0;
13
14     // Continuous assignment: replicate 1 wire to 2 outputs
15     assign not_pmod_0 = ~pmod[0];
16     assign led[1:0] = {2{not_pmod_0}};
17
18     // Continuous assignment: NOT and AND operators
19     assign led[2] = not_pmod_0 & ~pmod[1];
20
21 endmodule
```



VHDL

- 1983
- Department of Defense
- Strongly-typed
- More verbose

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

-- Entity declaration

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 definition

architecture andLogic of andGate is

 begin

    Y <= A AND B;

end andLogic;
```

Verilog

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

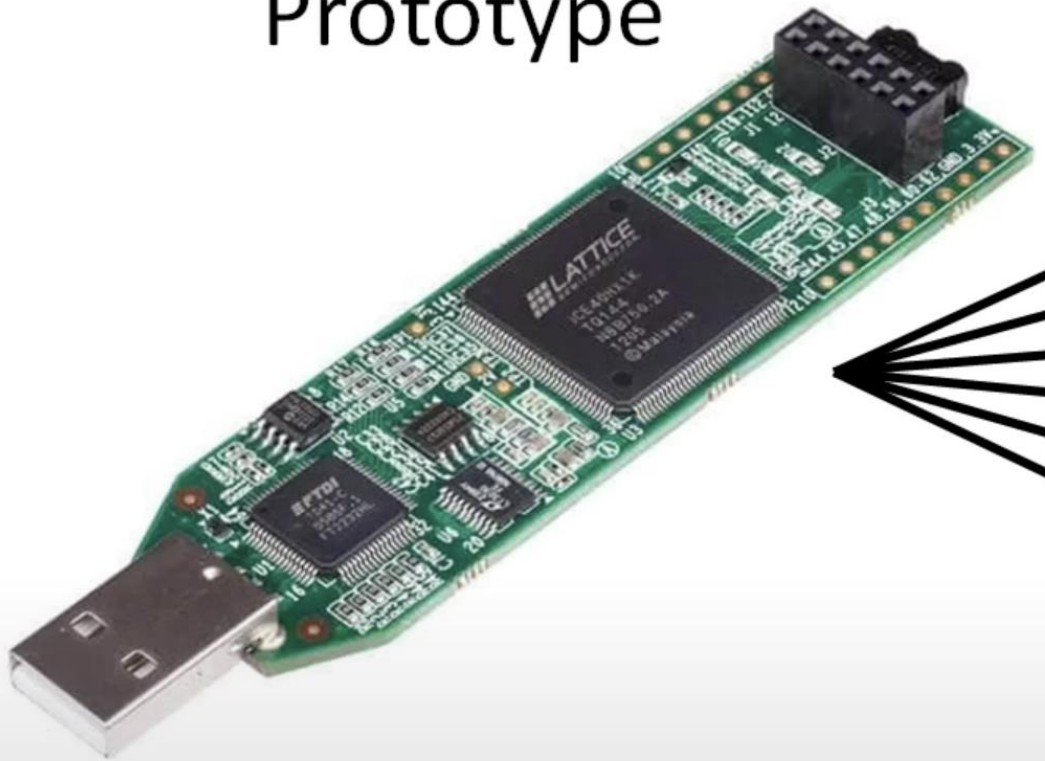
```verilog
// Module: AND gate example
module and_gate (

    // Inputs
    input   pmod_0,
    input   pmod_1,

    // Outputs
    output  led_0
);

    // Continuous assignment
    assign led_0 = ~pmod_0 & ~pmod_1;

endmodule
```

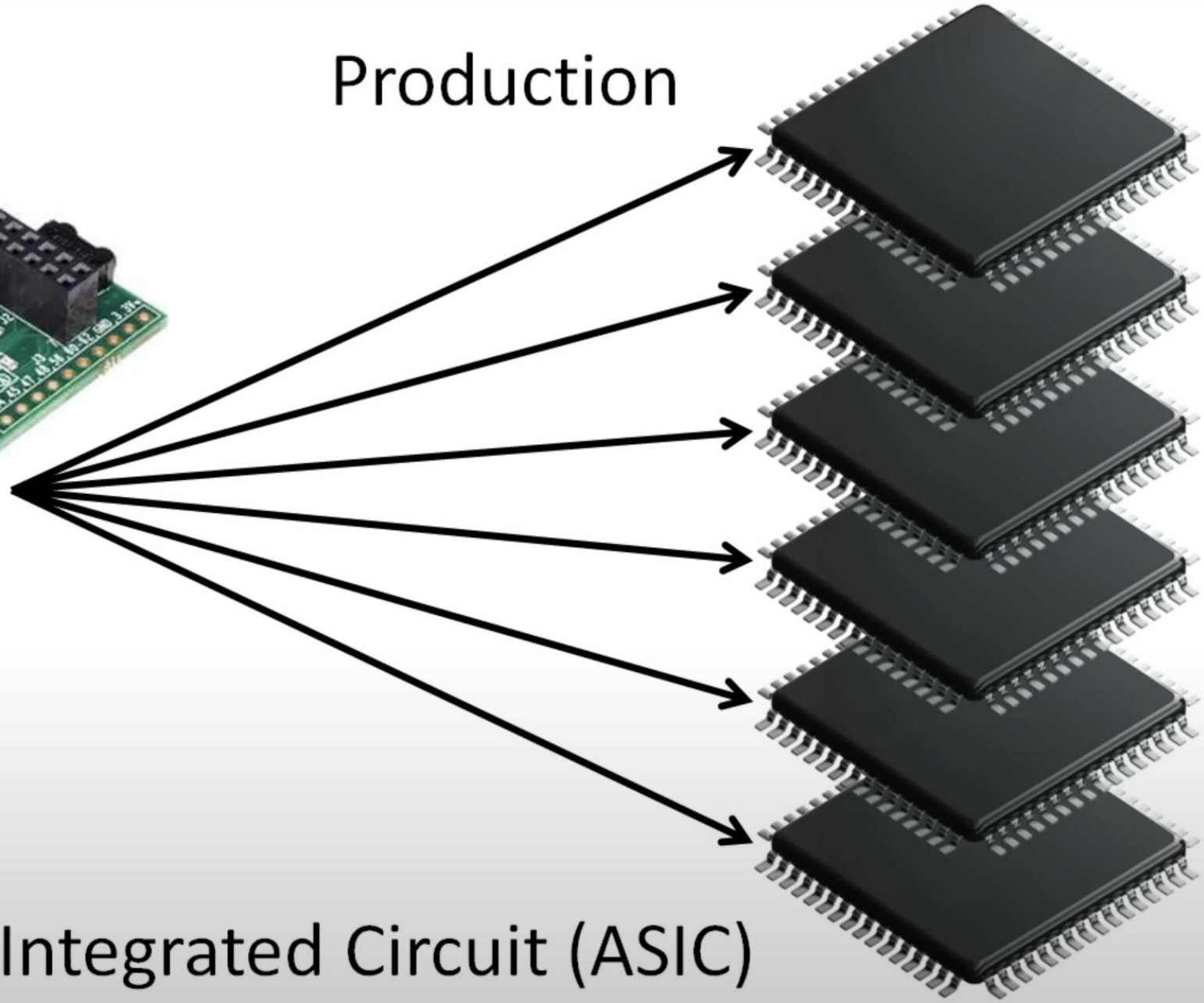
Register Transfer Level (RTL)

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

Production

Prototype





Application-Specific Integrated Circuit (ASIC)

FPGA Design Flow