Einführung in FPGAs und VHDL

- smalltalk 2014-07-23 -

Benjamin Koch

Überblick

- Unterschied zu imperativen Programmiersprachen
- Wie ist ein FPGA aufgebaut?
- Überblick über VHDL
- Wie fängt man am besten an?

Imperative Programmiersprachen

- Python

```
a,b = b,a
```

```
c = a*42+b
```

- C

```
int tmp = a;
a = b;
b = tmp;
```

```
int c = a*42+b;
```

Assembler (a,b = b,a)

```
18: 55                    push   %rbp
19: 48 89 e5              mov    %rsp,%rbp
1c: 89 7d ec              mov    %edi,-0x14(%rbp)
1f: 89 75 e8              mov    %esi,-0x18(%rbp)
22: 8b 45 ec              mov    -0x14(%rbp),%eax
25: 89 45 fc              mov    %eax,-0x4(%rbp)
28: 8b 45 e8              mov    -0x18(%rbp),%eax
2b: 89 45 ec              mov    %eax,-0x14(%rbp)
2e: 8b 45 fc              mov    -0x4(%rbp),%eax
31: 89 45 e8              mov    %eax,-0x18(%rbp)
34: 5d                    pop    %rbp
35: c3                    retq
```

Assembler (c = a*42+b)

```
 0: 55                    push   %rbp
 1: 48 89 e5              mov    %rsp,%rbp
 4: 89 7d ec              mov    %edi,-0x14(%rbp)
 7: 89 75 e8              mov    %esi,-0x18(%rbp)
 a: 8b 45 ec              mov    -0x14(%rbp),%eax
 d: 6b c0 2a              imul   $0x2a,%eax,%eax
10: 03 45 e8              add    -0x18(%rbp),%eax
13: 89 45 fc              mov    %eax,-0x4(%rbp)
16: 5d                    pop    %rbp
17: c3                    retq
```

CPU - Overview

Figure 2-1. Block Diagram

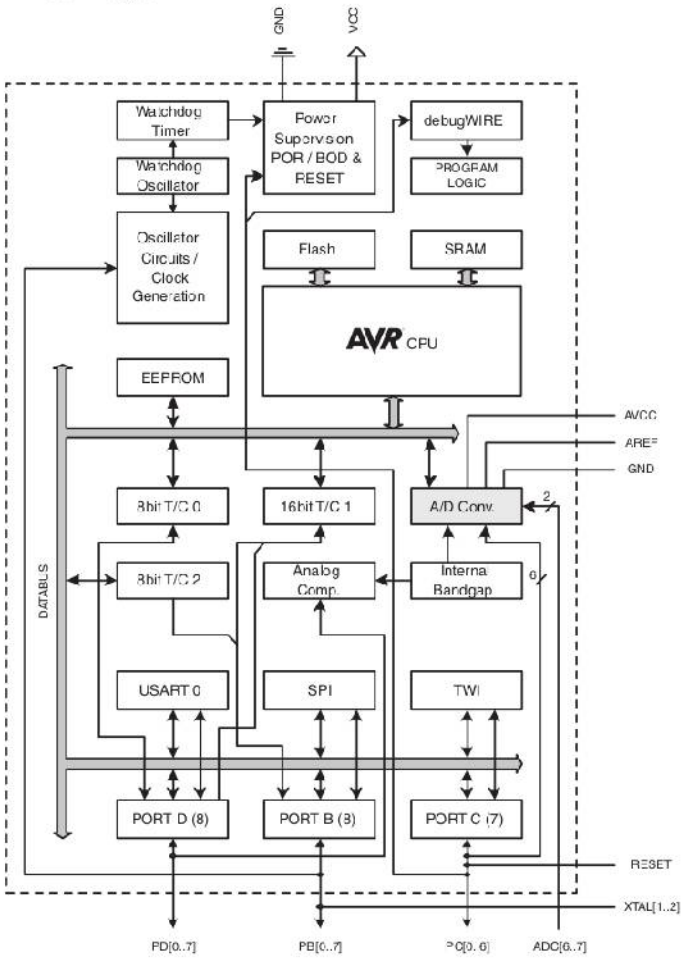


Image: Datasheet for ATmega328

CPU – Core

Figure 6-1. Block Diagram of the AVR Architecture

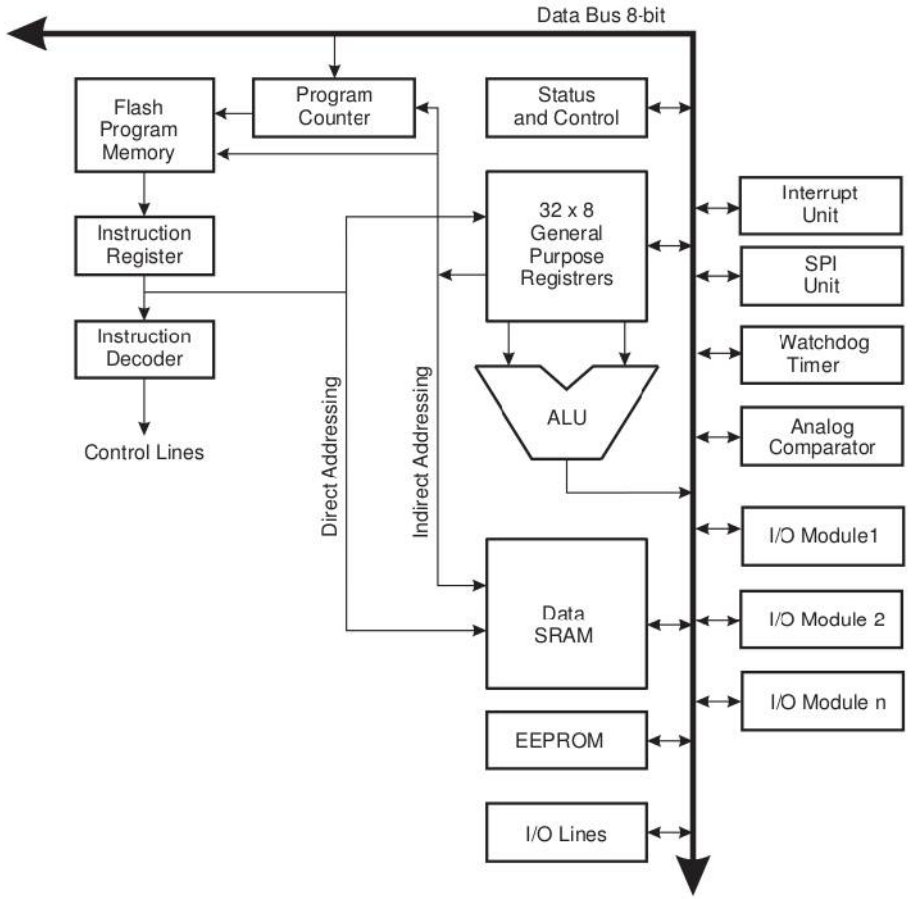


Image: Datasheet for ATmega328

CPU – Register-Transfer Level

Figure 13-2. General Digital I/O[(1)]

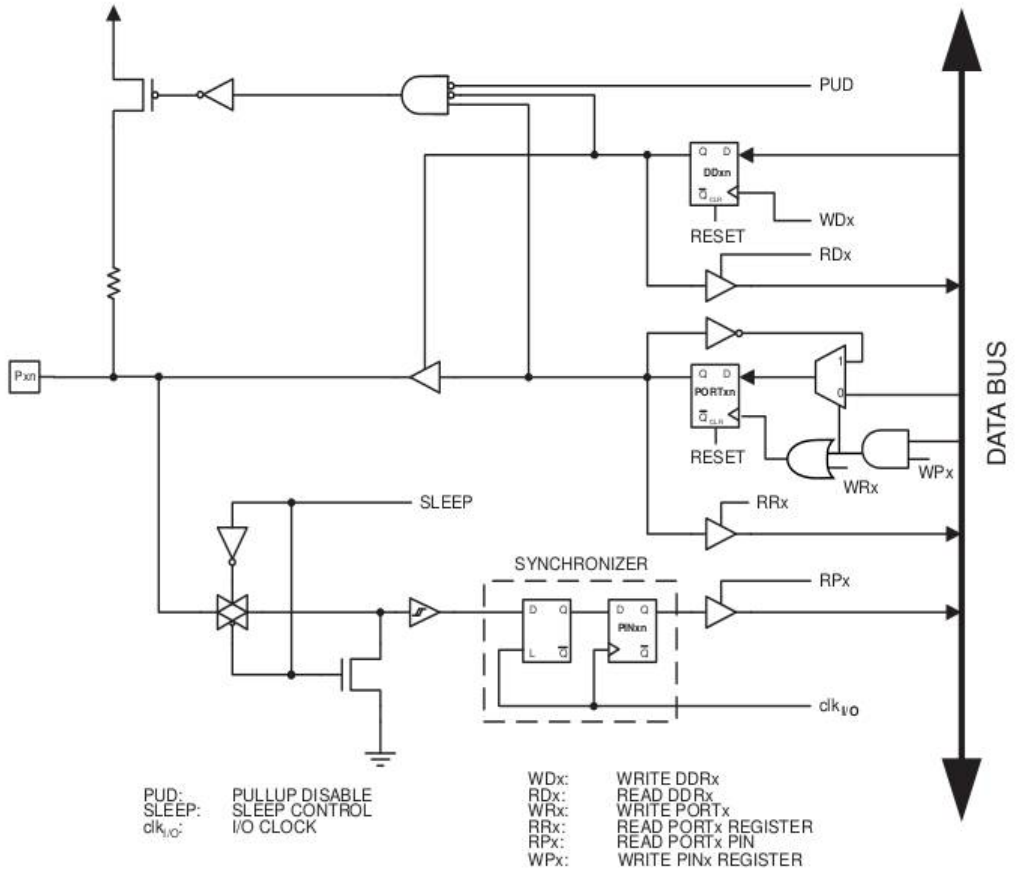


Image: Datasheet for ATmega328

CPU – Circuit Level

Figure 13-1. I/O Pin Equivalent Schematic

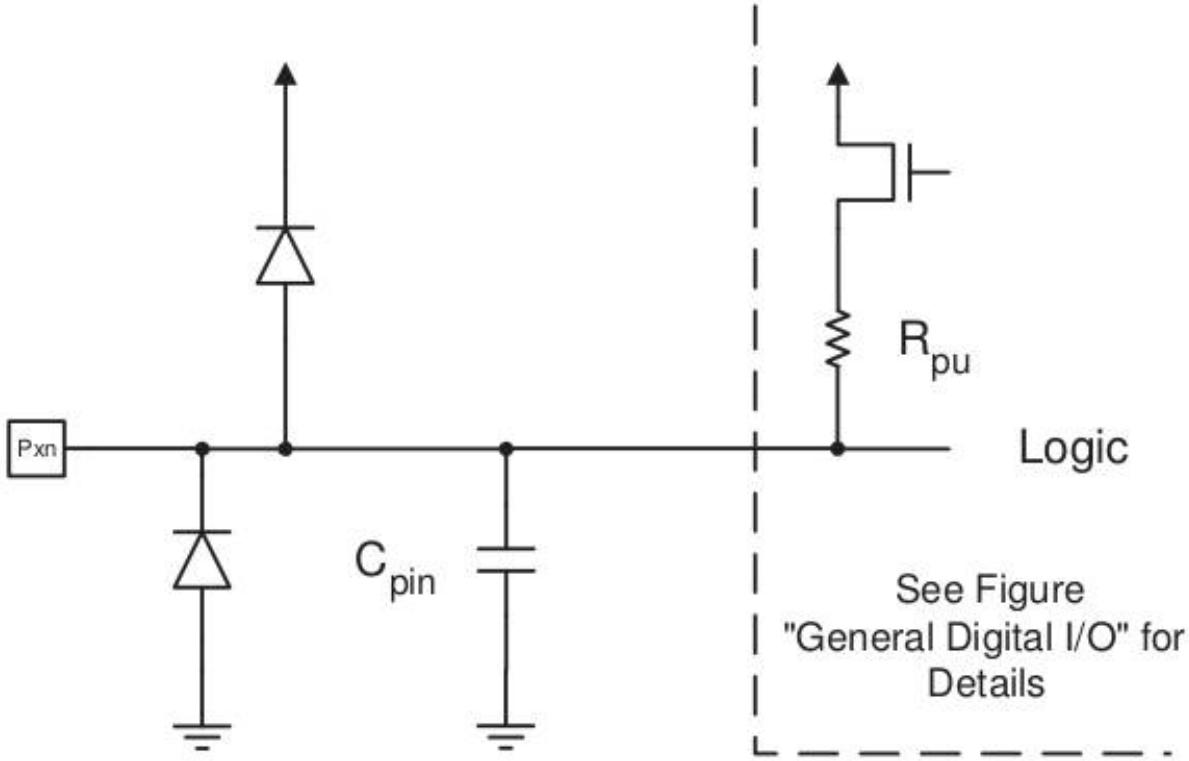


Image: Datasheet for ATmega328

Überblick

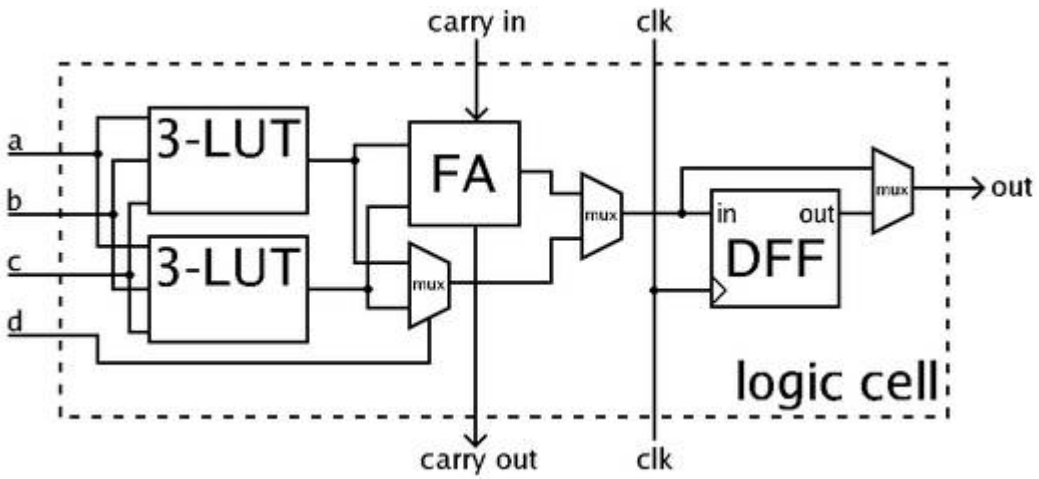
- Unterschied zu imperativen Programmiersprachen
- **Wie ist ein FPGA aufgebaut?**
- Überblick über VHDL
- Wie fängt man am besten an?

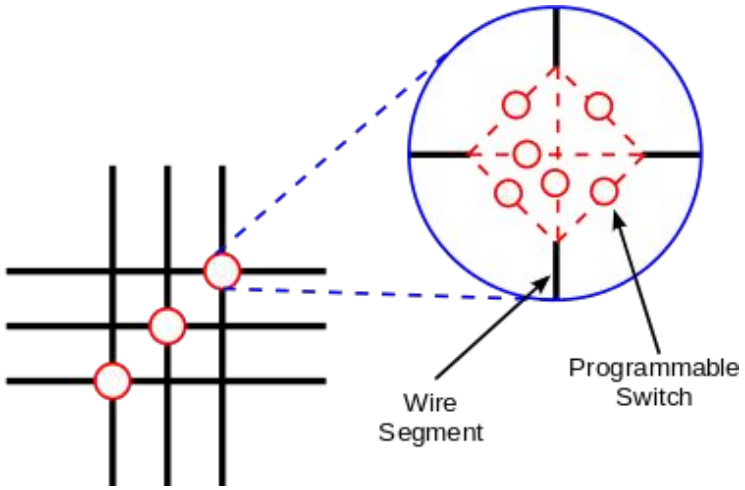
FPGA

- ASIC sind teuer; 74HCxxx sind aufwändig
- FPGAs:
  - Massenproduktion → günstig
  - Wiederbeschreibbar (“field-programmable”)
- Vorteile gegenüber CPU:
  - Anpassbar, z.B. Pins und Peripherie
  - Parallelität
  - Optimierbar auf Geschwindigkeit, Preis, Energie

Aufbau eines FPGAs

- Logic: Look-Up Tables, Flip-Flops
- Interconnect
- Block RAM, DSP Blocks, etc.





Images: Wikipedia

Überblick

- Unterschied zu imperativen Programmiersprachen
- Wie ist ein FPGA aufgebaut?
- **Überblick über VHDL**
- Wie fängt man am besten an?

VHDL (a,b = b,a)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity swap is
   port ( a_in, b_in  : in  std_logic_vector(31 downto 0);
          a_out,b_out : out std_logic_vector(31 downto 0);
          clk : in std_logic);
end swap;

architecture behavioral of swap is
begin
  a_out <= b_in;
  b_out <= a_in;
end behavioral;
```

VHDL (a,b = b,a)

```
architecture behavioral of swap is
  component bram is
     port ( aclk         : in std_logic;
            write_enable : in std_logic;
            address      : in std_logic_vector(0 downto 0);
            data         : inout std_logic_vector(31 downto 0)
     );
  end component;
  signal a,b,data : std_logic_vector(31 downto 0);
  signal state : std_logic_vector(2 downto 0) := "000";
  signal write_enable : std_logic;
begin
  inst : bram
    port map (aclk => clk, write_enable => write_enable, ...);
  ...
end behavioral;
```

VHDL (a,b = b,a)

```
p : process(clk, IO, R) is
begin
  case state is
    when "000"   =>            address <= "0"; write_enable <= '0';
    when "001"   => a <= data; address <= "1"; write_enable <= '0';
    when "010"   => b <= data;
    when "011"   => data <= a; address <= "1"; write_enable <= '1';
    when "100"   => data <= b; address <= "0"; write_enable <= '1';
    when others => write_enable <= '0';
  end case;

  if rising_edge(clk) then
    state <= std_logic_vector(unsigned(state) + "001");
  end if;
end process;
```

Circuit (c = a*42 + b)

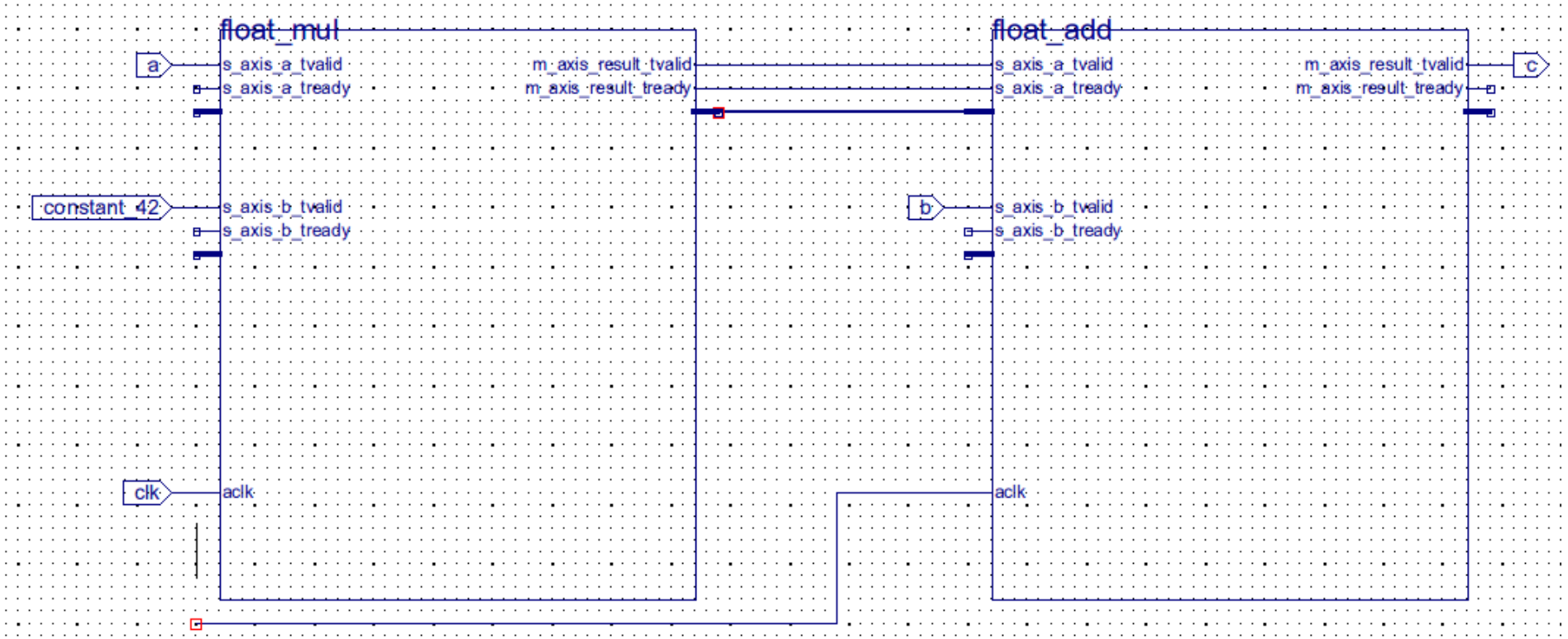


Image: Xilinx ISE

Geht das nicht einfacher?

High-Level Synthese – C zu VHDL

- Vivado HLS – teuer, aber gut

Prozessor im FPGA, z.B.

- ZPUino
- MicroBlaze

Überblick

- Unterschied zu imperativen Programmiersprachen
- Wie ist ein FPGA aufgebaut?
- Überblick über VHDL
- **Wie fängt man am besten an?**

Software

- Hersteller: Xilinx oder Altera
- Xilinx ISE
  - kostenlose Webpack Lizenz
  - Netzwerklizenzen in der Uni

Hardware

- Papilio ($80)
- Spartan-3 Board ($170)
- Parallela ($100 auf Kickstarter)
- CPLD-Evaluation-Board (20€, Pollin)

Fertig.