The Brainf*ck CPU Project

An Introduction to FPGA Development using VHDL

http://www.clifford.at/bfcpu/

Clifford Wolf

ROCK Linux - http://www.rocklinux.org

CNGW - http://www.cngw.orig

LINBIT - http://www.linbit.com

Overview
- Building custom hardware
- The Brainf*ck CPU

Programmable Hardware

Introduction to VHDL

Overview

LINBIT Information Technologies GmbH

Building custom hardware

Overview
- Building custom hardware
- The Brainf*ck CPU

Programmable Hardware

Introduction to VHDL

With programable hardware (such as PLDs and FPGAs) it is possible to

- Instantly test hardware designs with almost no prototyping costs
- Simply "upload" hardware designs (bitstream files) to a chip like it would be software
- Have much fun with building hardware without touching a soldering gun

With HDLs (such as Verilog and VHDL) it is possible to

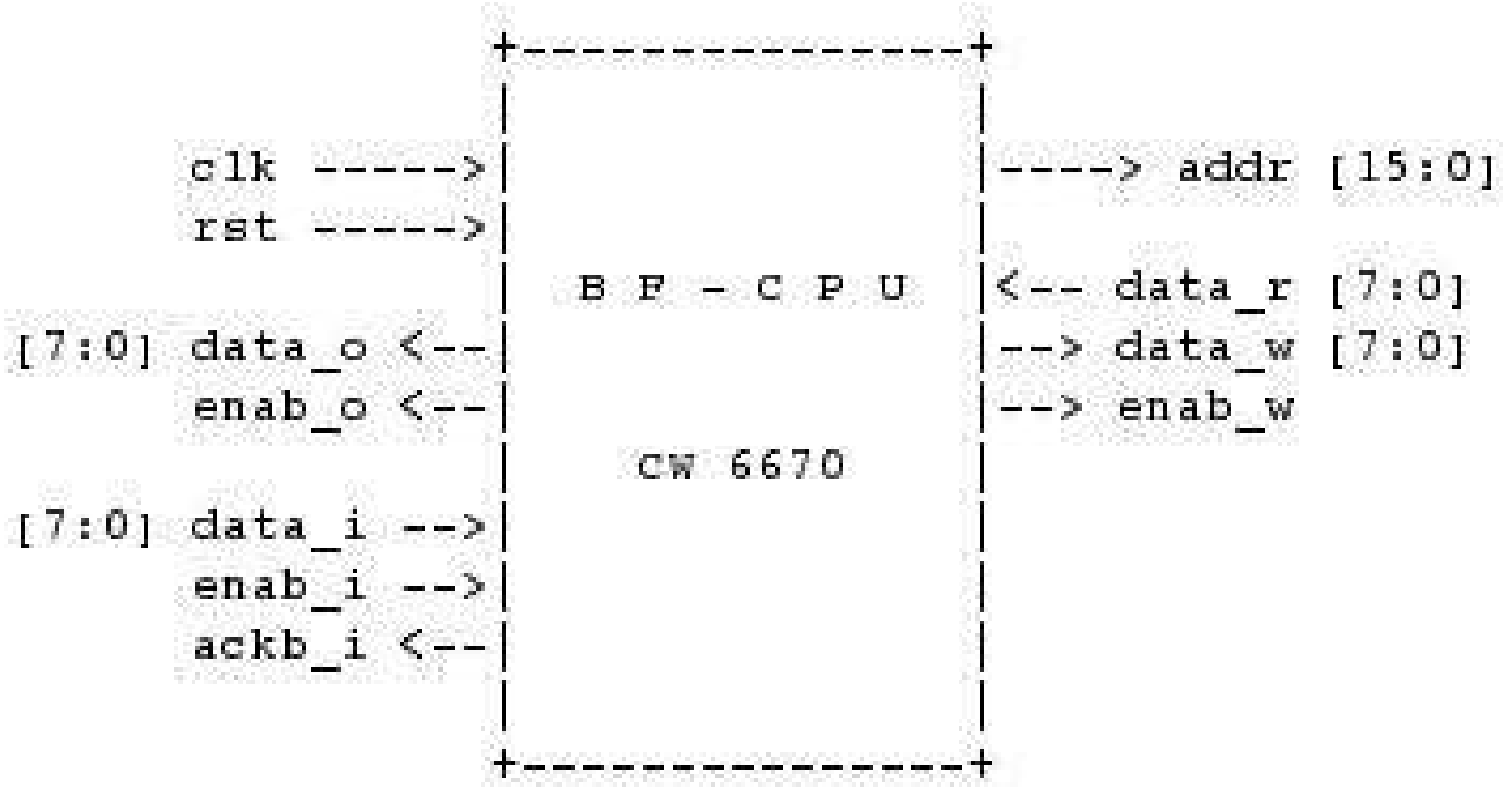
- Describe a hardware design like program source describes the behavior of a program
- Automatically create bitstream files for different chips from the same (portable) source
- Simulate the behavior of your hardware on various levels

The Brainf*ck CPU

Overview
- Building custom hardware
- The Brainf*ck CPU

Programmable Hardware

Introduction to VHDL



A minimalistic CPU with 8 bit data-bus and 16 bit address-bus which can execute brainf*ck code. The VHDL design file for it has 260 lines of code. The optimised variation of the CPU with a 1 byte internal data cache has 340 lines of code.

Overview

Programmable Hardware
- PALs
- PLDs and CPLDs
- FPGAs
- Recommended reading

Introduction to VHDL

Programmable Hardware

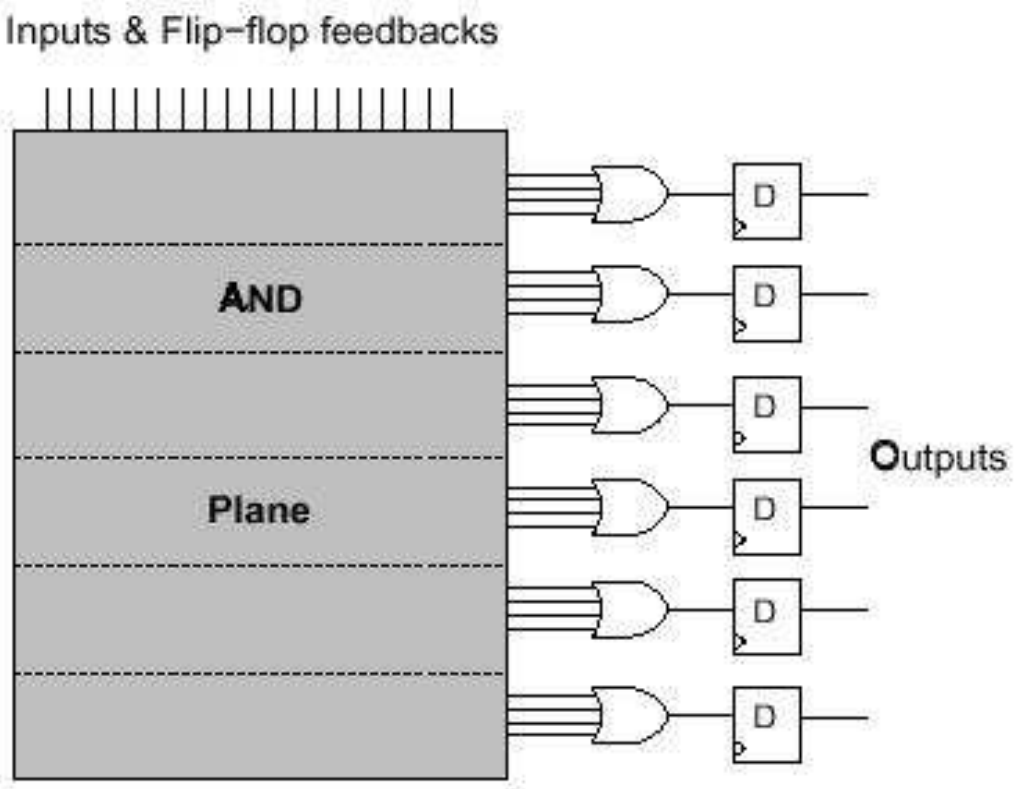
PALs

Overview

Programmable Hardware
- PALs
- PLDs and CPLDs
- FPGAs
- Recommended reading

Introduction to VHDL

A PAL is a very simple and limited programable hardware device:



It's programm only consists of a single truth table. Which is implemented as one big unit with d-flipflops (connected to one global clock) on the output lines.

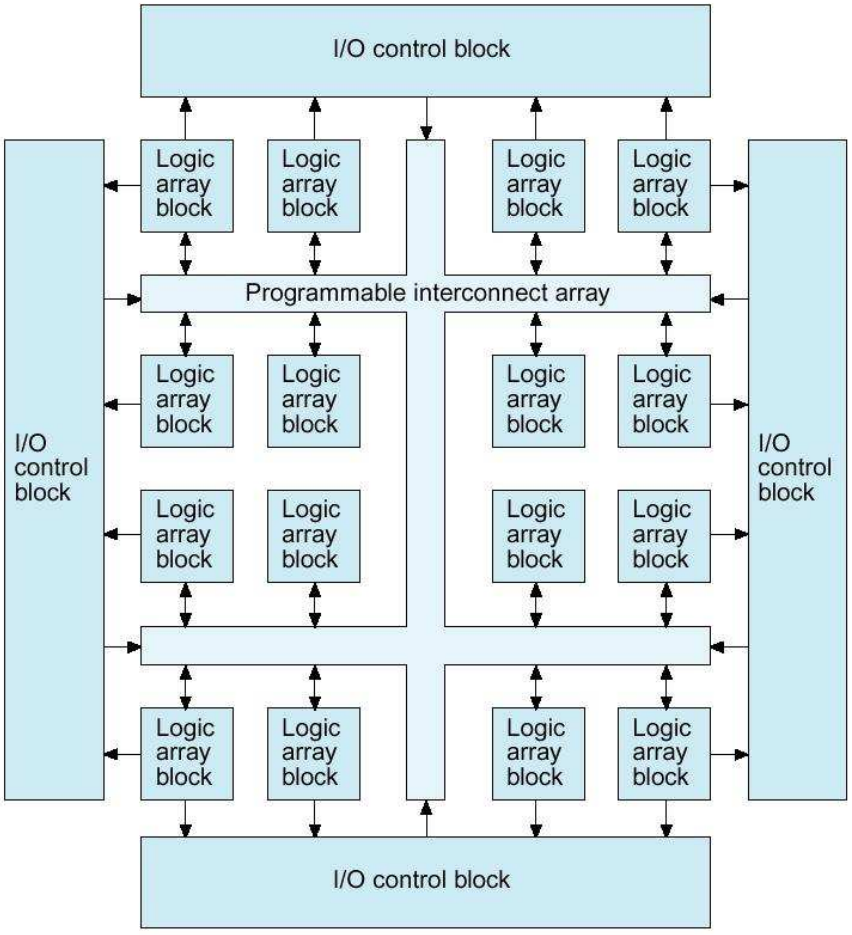
PLDs and CPLDs

Overview

Programmable Hardware
- PALs
- PLDs and CPLDs
- FPGAs
- Recommended reading

Introduction to VHDL

A PLD consists of macrocells which are something like better PALs and one central interconnection logic. PLDs are usually used to connect a high number of pins with a more or less simple logic at very low cost. Typical PLD applications are "glue logic" for connecting other ASICS.



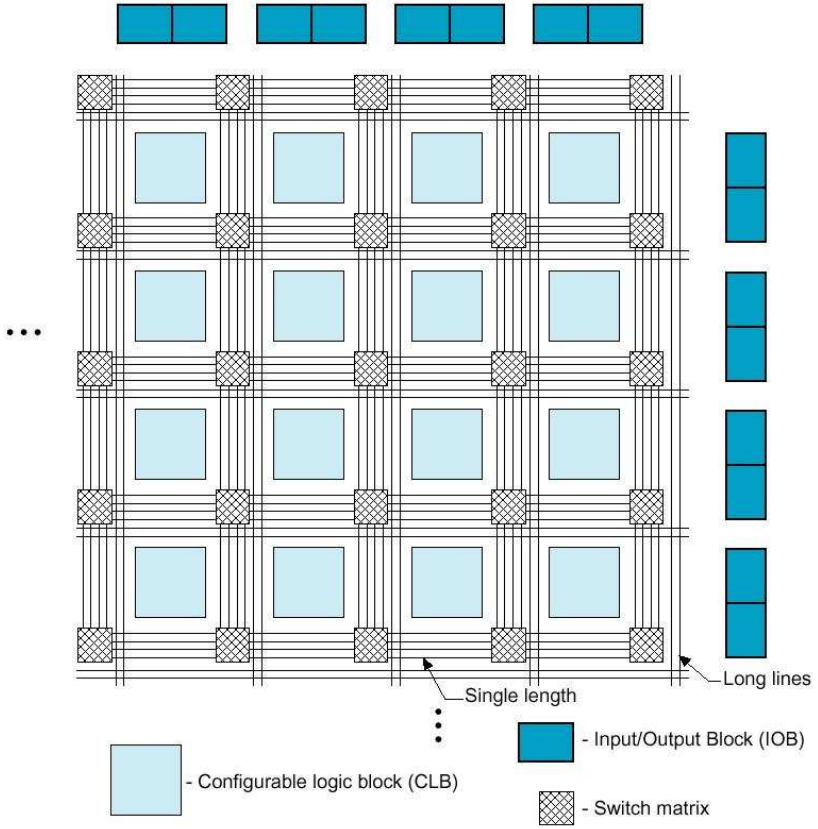
FPGAs

Overview

Programmable Hardware
- PALs
- PLDs and CPLDs
- FPGAs
- Recommended reading

Introduction to VHDL

FPGAs consist of even more complex logic blocks connected by a very complex interconnection matrix. FPGAs can implement a very complex logic on one chip. Typical applications are CPUs and DSPs up to very complex SoC setups.



Recommended reading

Overview
Programmable Hardware
- PALs
- PLDs and CPLDs
- FPGAs
- Recommended reading

Introduction to VHDL

The Xilinx "Detailed Functional Description" Datasheets

E.g.: "Spartan-IIE 1.8V FPGA Detailed Functional Description"
http://direct.xilinx.com/bvdocs/publications/ds077_2.pdf

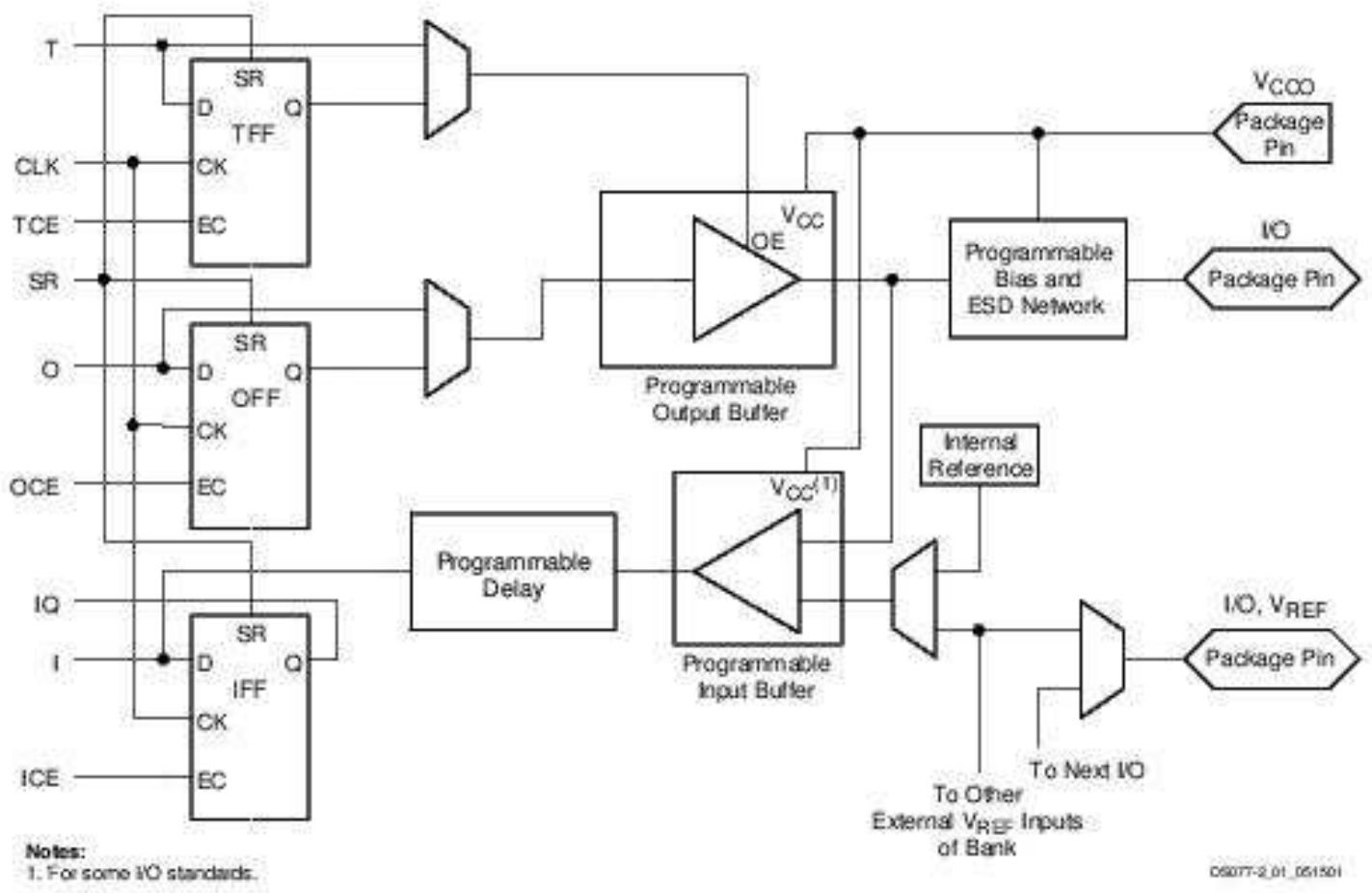


Figure 2: **Spartan-IIE Input/Output Block (IOB)**

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

Introduction to VHDL

LINBIT Information Technologies GmbH

What is VHDL

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

- VHDL is the "VHSIC Hardware Description Language"
- VHSIC is a "Very High Speed Integrated Circuit"
- VHDL was originally design to document circuits
- Later on, programs have been developt to generate ciruct designs from VHDL code
- Only a small subset of correct VHDL code can be used to synthesize designs

A 2 bit counter (1)

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

A simple clocked 2 bit counter (00, 01, 10, 11, 00, ..):

- 2 Output signals: D1 (lower bit) and D2 (higher bit)
- The following truth table shows how to calculate new D1 and D2 from the old values:

D2	D1	D2,	D1,
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

D2	D1	D2,
0	0	0
0	1	1
1	0	1
1	1	0

D1	D1,
0	1
1	0
0	1
1	0

So it turns out: D2 xor D1 not D1

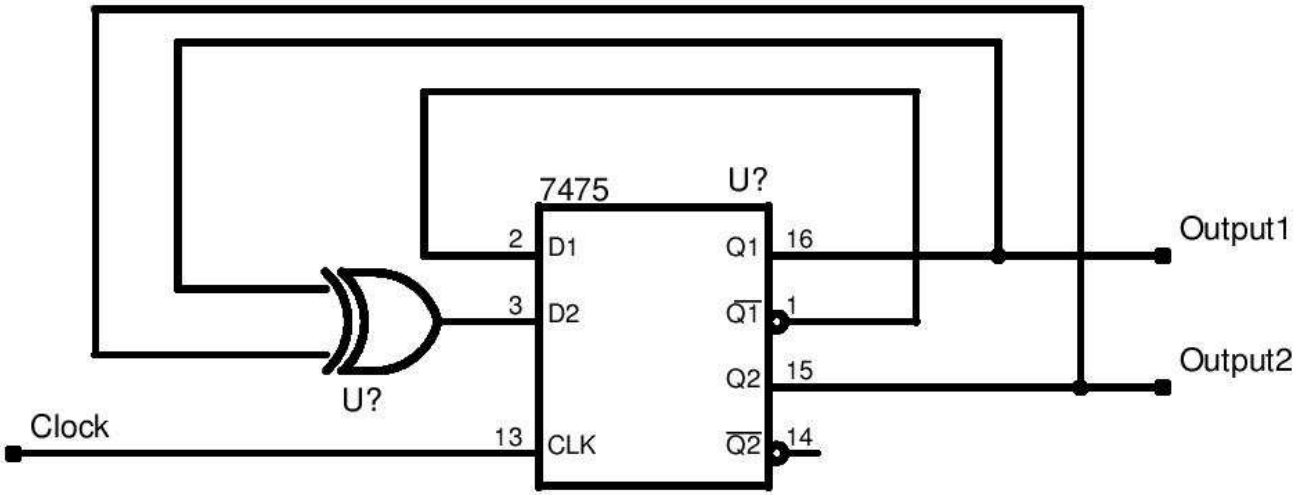
A 2 bit counter (2)

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

The 2 bit counter as circuit:



And this is the VHDL code for the same thing:

```
process (clock) begin
    if rising_edge(clock) then
        output1 <= not output1;
        output2 <= output1 xor output2;
    end if;
end process;
```

LINBIT Information Technologies GmbH

Signals in VHDL

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

- Signals are single wires or buses in the circuit.
- Buses (multiple wires with one name) are usually called "signal vectors".
- There are multiple signal types (vitally important: bit and std_logic).
- Usually a signal my only have one single source (exception: std_ulogic).
- Many signal types have predefined operations (like "xor", "not", "+" or "*").
- Signals can be assigned a value using the "<=" operator

LINBIT Information Technologies GmbH

Prozesses in VHDL (1)

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

- Processes are one way to group VHDL statements to a logical unit.
- A dependency list contains all signals the statements in the process depend on.

```
process (mysignal2, mysignal3) begin
    mysignal1 <= mysignal2 xor mysignal3;
end process;
```

- Clocked processes are the prefered way to use d-flip-flops in the design:

```
process (clk) begin
    if rising_edge(clk) then
            mysignal1 <= mysignal2 xor mysignal3;
    end if;
end process;
```

Processes in VHDL (2)

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

- Clocked processes may also contain code for reset signals:

```
process (clk, rst) begin
    if rst = '1' then
        mysignal1 <= x"00";
    elsif rising_edge(clk) then
        mysignal1 <= mysignal1 + 1;
    end if;
end process;
```

- Remember: Only a small subset of synthactically and gramatically correct code can actually used to synthesize a real hardware design.
- No more than one "if rising_edge(clk)" per process.
- That "if" must surrond all other instructions in the process.
- The only allowed variation is the check for a reset signal.

Processes in VHDL (3)

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

- Processes may always contain code which implies a feedback of outputs. The first code fragment implies the else-tree of the 2nd one:

```
process (clk) begin
        if rising_edge(clk) then
                if enable = 1 then
                        mysignal1 <= mysignal1 + 1;
                end if;
        end if;
end process;

process (clk) begin
        if rising_edge(clk) then
                if enable = 1 then
                        mysignal1 <= mysignal1 + 1;
                else
                        mysignal1 <= mysignal1;
                end if;
        end if;
end process;
```

LINBIT Information Technologies GmbH

Variables in VHDL

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

- Variables may be used just as variables in a traditional program.
- Variables are assigned values using the ":=" operator.
- Variables are always local to a process.

```
process (clk)
        variable temp : std_logic_vector std_logic_vect
begin
        if rising_edge(clk) then
                temp := mysignal1;
                temp := temp + 25;
                temp := temp * mysignal2;
                if mysignal3 = '0' then
                        temp := mysignal2;
                end if;
                mysignal1 <= temp;
                temp := temp + 844;
                mysignal2 <= temp;
        end if;
end process;
```

LINBIT Information Technologies GmbH

Entities, archs and components

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and component
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

- A VHDL project is structured in so-called entities.
- An "architecture" is the concrete implementation of an entity.
- An entity may have multiple architectures (e.g. optimized for size or speed).
- When an entity is used in another entity, it is called a component.
- Components and signal types can be imported from libraries.

LINBIT Information Technologies GmbH

VHDL Example (1)

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits


```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity demo is
        port(
                clk : in std_logic;
                rst : in std_logic;
                dat : out std_logic_vector (15 downto 0)
        );
end demo;
```

VHDL Example (2)

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

```
architecture demo_arch of demo is
        signal val : std_logic_vector (15 downto 0);
begin
        process (clk, rst) begin
                if rst = '1' then
                        val <= (others => '0');
                elsif rising_edge(clk) then
                        val <= val + 1;
                end if;
        end process;
        dat <= val;
end demo_arch;
```

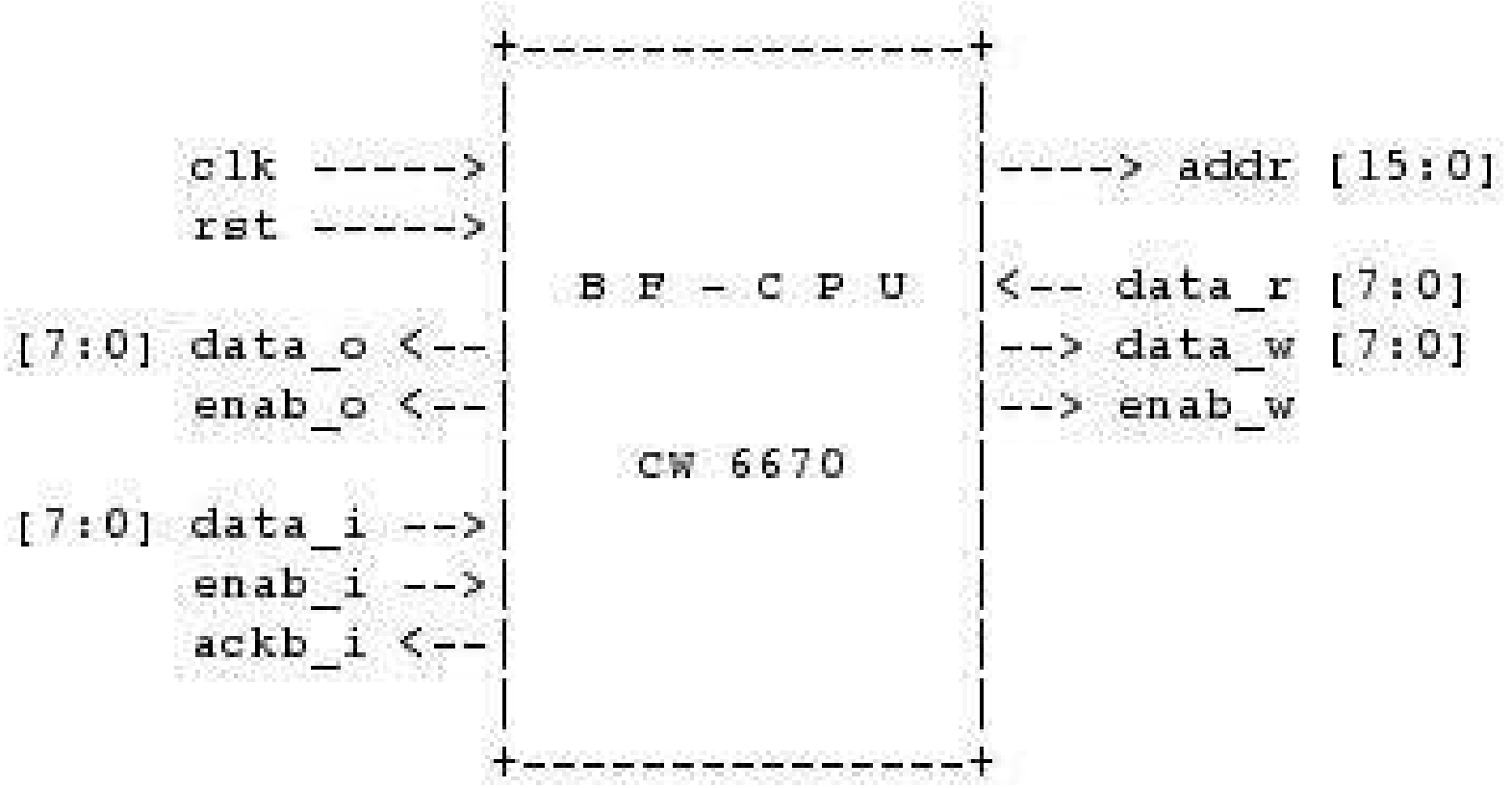
The Brainf*ck CPU

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

[Discussion the Brainf*ck CPU VHDL code]



LINBIT Information Technologies GmbH

Credits

Overview

Programmable Hardware

Introduction to VHDL
- What is VHDL
- A 2 bit counter (1)
- A 2 bit counter (2)
- Signals in VHDL
- Prozesses in VHDL (1)
- Processes in VHDL (2)
- Processes in VHDL (3)
- Variables in VHDL
- Entities, archs and components
- VHDL Example (1)
- VHDL Example (2)
- The Brainf*ck CPU
- Credits

- LINBIT Information Technologies GmbH: http://www.linbit.com/
- The ROCK Linux Project: http://www.rocklinux.org/
- Clifford Wolf: http://www.clifford.at/

http://www.clifford.at/bfcpu/