

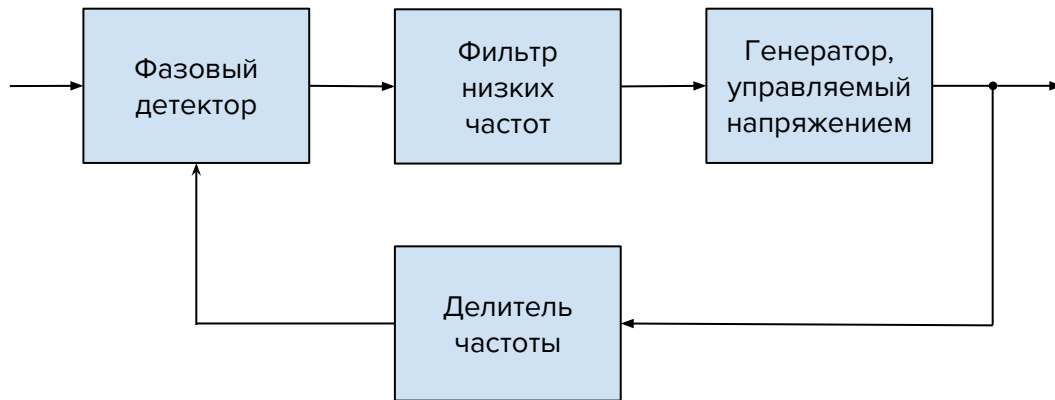
Фазовая автоподстройка частоты

ФАПЧ

ФАПЧ (Phase-Locked Loop, PLL) — система, содержащая генератор, фаза которого автоматически подстраивается под фазу входного сигнала или отклоняется от нее по требуемому закону

Применение:

- Модуляция, демодуляция
- Определение частоты и фазы принимаемого сигнала
- Умножение частоты



Altera Cyclone IV EP4CE6E22C8N

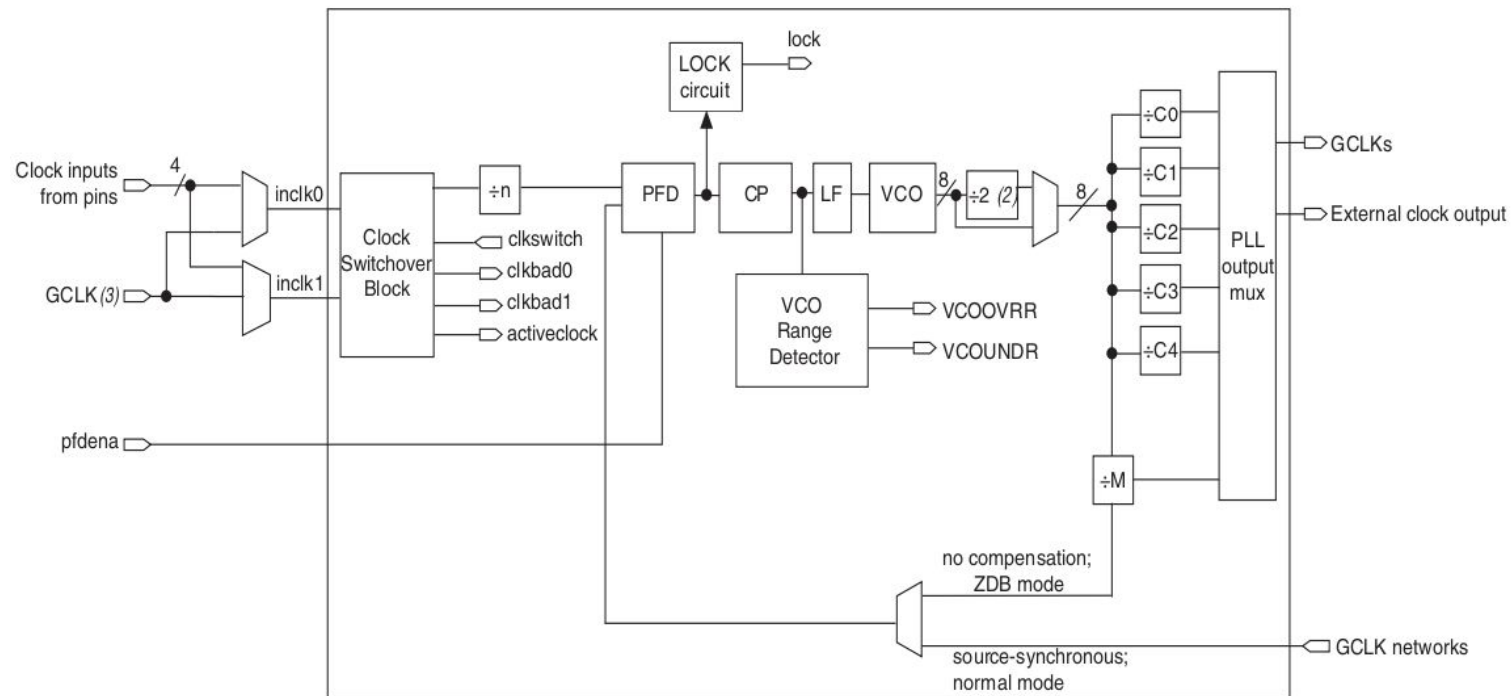
Ресурсы:

- Логические элементы
- Блоки памяти
- Умножители
- Блоки ФАПЧ (PLL)
- Порты ввода-вывода

Table 1-1. Resources for the Cyclone IV E Device Family

Resources	EP4CE6	EP4CE10	EP4CE15	EP4CE22
Logic elements (LEs)	6,272	10,320	15,408	22,320
Embedded memory (Kbits)	270	414	504	594
Embedded 18 × 18 multipliers	15	23	56	66
General-purpose PLLs	2	2	4	4
Global Clock Networks	10	10	20	20
User I/O Banks	8	8	8	8
Maximum user I/O ⁽¹⁾	179	179	343	153

ФАПЧ в Altera Cyclone IV E



ALTPLL IP

Installed IP

Project Directory

No Selection Available

Library

- Basic Functions
 - Arithmetic
 - Bridges and Adaptors
 - Clocks; PLLs and Resets
 - ALTCLKCTRL Intel FPGA IP
 - PLL**
 - ALTPLL**
 - ALTPLL_RECONFIG
 - Configuration and Programming
 - I/O
 - Miscellaneous
 - On Chip Memory
 - Simulation; Debug and Verification
- DSP
- Interface Protocols
- Memory Interfaces and Controllers
- Processors and Peripherals
- University Program

Search for Partner IP

ALTPLL [About](#) [Documentation](#)

1 Parameter Settings 2 PLL Reconfiguration 3 Output Clocks 4 EDA 5 Summary

General/Modes Inputs/Lock Bandwidth/SS Clock switchover

Currently selected device family: Cyclone IV E ☐ Match project/default

Able to implement the requested PLL

General

Which device speed grade will you be using?

☐ Use military temperature range devices only

What is the frequency of the inclk0 input? MHz

☐ Set up PLL in LVDS mode Data rate: Mbps

PLL Type

Which PLL type will you be using?

☒ Fast PLL ☐ Enhanced PLL ☐ Select the PLL type automatically

Operation Mode

How will the PLL outputs be generated?

☒ Use the feedback path inside the PLL

☒ In normal mode

☐ In source-synchronous compensation Mode

☐ In zero delay buffer mode

☐ Connect the fbmimic port (bidirectional)

☐ With no compensation

☐ Create an 'fbin' input for an external feedback (External Feedback Mode)

Which output clock will be compensated for?

Diagram showing PLL configuration for Cyclone IV E:

inclk0 (reset) → PLL → c0 (locked)

PLL Parameters:

clk	Ratio	Ph (dp)	PC (%)
c0	1/1	0.00	50.00

Cyclone IV E

ALTPLL IP

ALTPLL About Documentation

1 Parameter Settings 2 PLL Reconfiguration 3 **Output Clocks** 4 EDA 5 Summary

clk c0 > clk c1 > clk c2 > **clk c3** > clk c4

c0 - Core/External Output Clock
Able to implement the requested PLL

☒ Use this clock

Clock Tap Settings

☐ Enter output clock frequency: Requested Settings: 100.00000000 [MHz] Actual Settings: 75.000000

☒ Enter output clock parameters:

Clock multiplication factor: 3 Actual Settings: 3

Clock division factor: 2 Actual Settings: 2

Clock phase shift: 0.00 [deg] Actual Settings: 0.00

Clock duty cycle (%): 50.00 Actual Settings: 50.00

Note: The displayed internal settings of the PLL is recommended for use by advanced users only

Per Clock Feasibility Indicators: c0 c1 c2 c3 c4

Cancel < Back Next > Finish

Файл pll_inst.v

```
pll pll_inst(
    .inclk0(inclk0_sig),
    .c0(c0_sig)
);
```

File	Description
<input checked="" type="checkbox"/> pll.v	Variation file
<input checked="" type="checkbox"/> pll.ppf	PinPlanner ports PPF file
<input type="checkbox"/> pll.inc	AHDL Include file
<input type="checkbox"/> pll.cmp	VHDL component declaration file
<input type="checkbox"/> pll.bsf	Quartus Prime symbol file
<input checked="" type="checkbox"/> pll_inst.v	Instantiation template file
<input checked="" type="checkbox"/> pll_bb.v	Verilog HDL black-box file

top.v

.....

```
wire clk;  
pll pll_inst(  
    .inclk0(CLK), // 50 МГц  
    .c0(clk)      // 75 МГц  
);
```

```
wire [15:0]hd_data;  
cpu_top cpu_top(  
    .clk(clk),  
    .data_out(hd_data)  
);
```

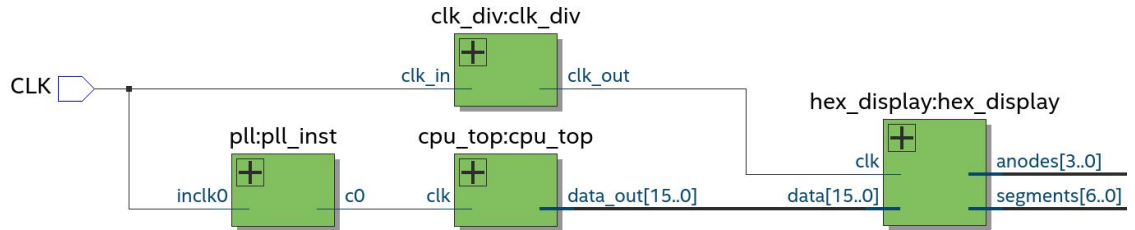
.....

Из файла fib.s

.....

```
_delay:  
    addi    t5, t5,    128    # increment delay counter  
    bne     t5, zero,   _delay # next delay loop (~670ms)
```

.....



GitHub

github.com/viktor-prutyanov/drec-fpga-intro