

# Free Hardware Implementation of Ogg Theora Video Encoder

Andrey Filippov, Ph.D

Elphel, Inc.



# Background

Started as a system based on embedded Linux, Elphel cameras dramatically increased performance by applying methods of FOSS design to the FPGA code:

- 2001 – GNU/Linux-based Model 303 camera with pure software implementation of JPEG compression. Resolution – 1280x1024, frame rate - 0.2 fps;
- 2003 – Model 313 camera uses the same processor, JPEG compression performed by a reconfigurable FPGA. Resolution - 2048x1536, frame rate at maximal resolution – 9fps (1280x1024 at 22 fps – almost 100 faster);

# Project Goals

**Advanced network camera** should provide a combination of the following features:

- **High resolution** is important so a camera with a wide angle lens will not miss an important event;
- **High frame rate** common for the analog cameras should be maintained;
- **Low bit rate** is critical for the systems that combine high resolution with high frame rate.

# Project goals

# Selection of the video compression algorithm

	<b><i>Video Compression Efficiency</i></b>	<b><i>Computa- tional Complexity</i></b>	<b><i>Free to Implement</i></b>
<b><i>MJPEG</i></b>	low	medium	yes
<b><i>MPEG</i></b>	high	high	no
<b><i>Ogg Theora</i></b>	high	high	yes

# Design goals

# High performance video encoding is a computationally-intensive procedure:

- **pure software** solution is not practical – 1280x1024 @ 30fps data requires power of 4-5GHz universal x86 CPU;
- **off-the-shelf compressor ASICs** are not available for the high resolution video and Ogg Theora format;
- **custom ASICs** are expensive to develop and impossible to upgrade;
- **FPGA implementation** combines “hardware” performance with the flexibility of the software.

System Architecture

**FPGA implementation** provides adequate computational performance and allows usage of the new emerging video formats before they are finalized.

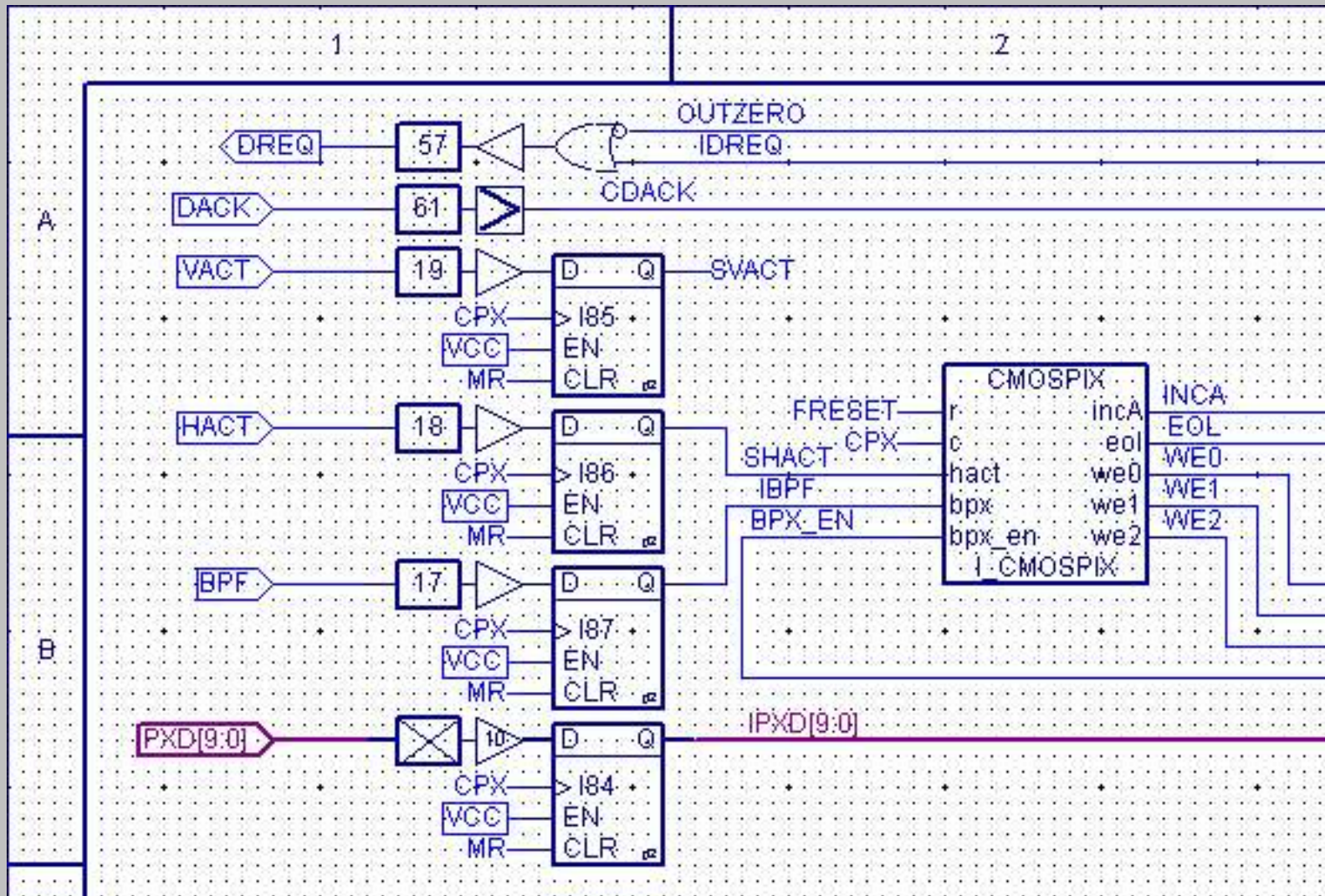
Reprogrammability of such devices allows code modifications after the hardware is built.

That makes FOSS development model relevant, and the freedoms of GNU/GPL applicable to this border area between software and hardware.

# System Architecture

# Designing with FPGA

Simple FPGA designs can be developed using hardware approach – schematic capture.



Software approach works better for complex designs, we use Verilog Hardware Description Language (HDL) for Elphel products.

```
34 ** Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
35 ** -----**
36 */
37 module idct_1d      (clk,      // twice pixel clock, (@posedge)
38                    en,        // enable (only for simulation?)
39                    start,     // 1 cycle long with the first data
40                    skip,     // preserve timing but save power on reducing regis
41                    y,        // [15:0] 16-bit (signed) input data
42                    xa,       // [15:0] 16-bit (signed) output data for X3, X0, X
43                    xb,       // [15:0] 16-bit (signed) output data for X4, X7, X
44                    xvld     // output data pair valid
45                    );
46 parameter C3= 54491;
47 parameter S3= 36410;
48 parameter C4= 46341;
49 parameter C6= 25080;
50 parameter S6= 60547;
51 parameter C7= 12785;
52 parameter S7= 64277;
53 |
54 input      clk;

109 assign xa[15:0] = ar[15:0];
110 assign xb[15:0] = sr[15:0];
111
112 always @ (posedge clk) begin
113     first_run <= en && ((start && !skip) || (first_run && (cntr[3:0] != 4'hf)));
114     last_run  <= en && ((first_run && (cntr[3:0] == 4'hf)) || (last_run && (cntr[3:0]
115
116     if      (start )                cntr[3:0] <= 4'h0;
117     else if (!(first_run || last_run ||
118             first_skip || last_skip)) cntr[3:0] <= 4'hf;
119     else                                cntr[3:0] <= cntr[3:0] + 1;
120 | case (cntr[3:0] | {4{start}})
121     4'hf: {mir en, mir sel[1:0]} <= 3'h4;
```

# FPGA vs Software

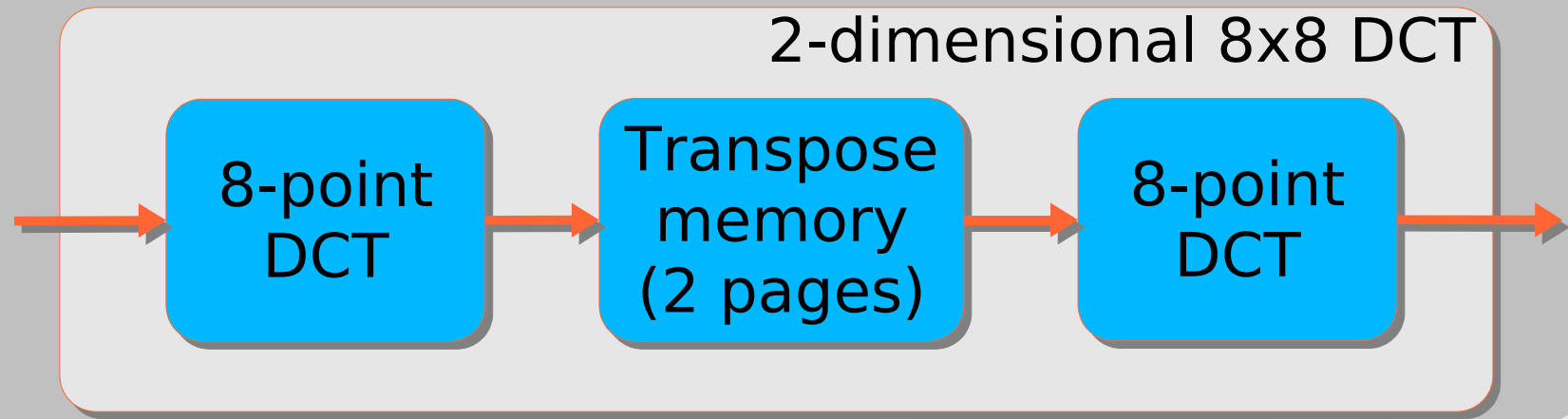
There are some fundamental differences between FPGA and CPU operation that make programming them different:

- **Semi-analog nature of the hardware.**  
Propagation delays of the signals are analog even for the binary-level signals;
- All parts of the **FPGA operate in parallel** at the same time. Different HDL operators are mapped to different parts of the chip contrary to the CPUs that execute them at different times.

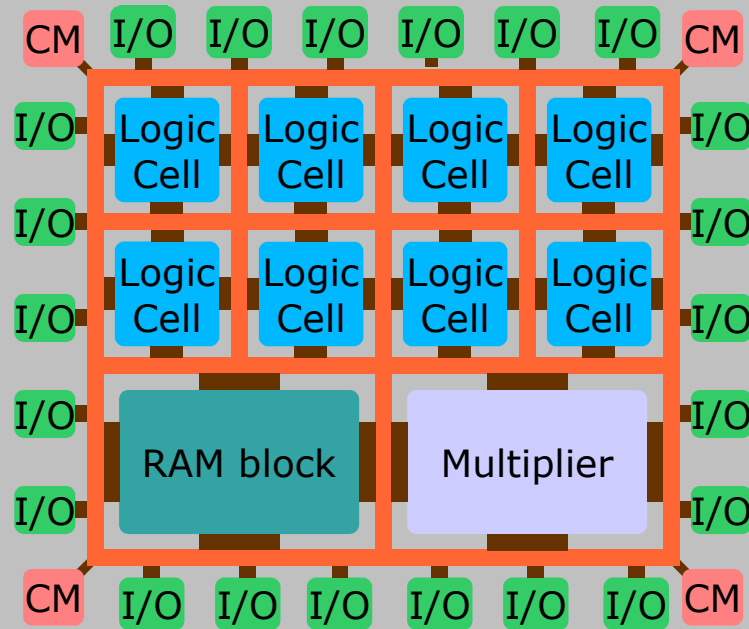
Programming FPGA

# Parallelism in FPGA

MISD parallel architecture (pipe-like chains) is more common in FPGA than SIMD that is used for the multimedia extensions in modern CPUs. But generally they are MIMD, of course.

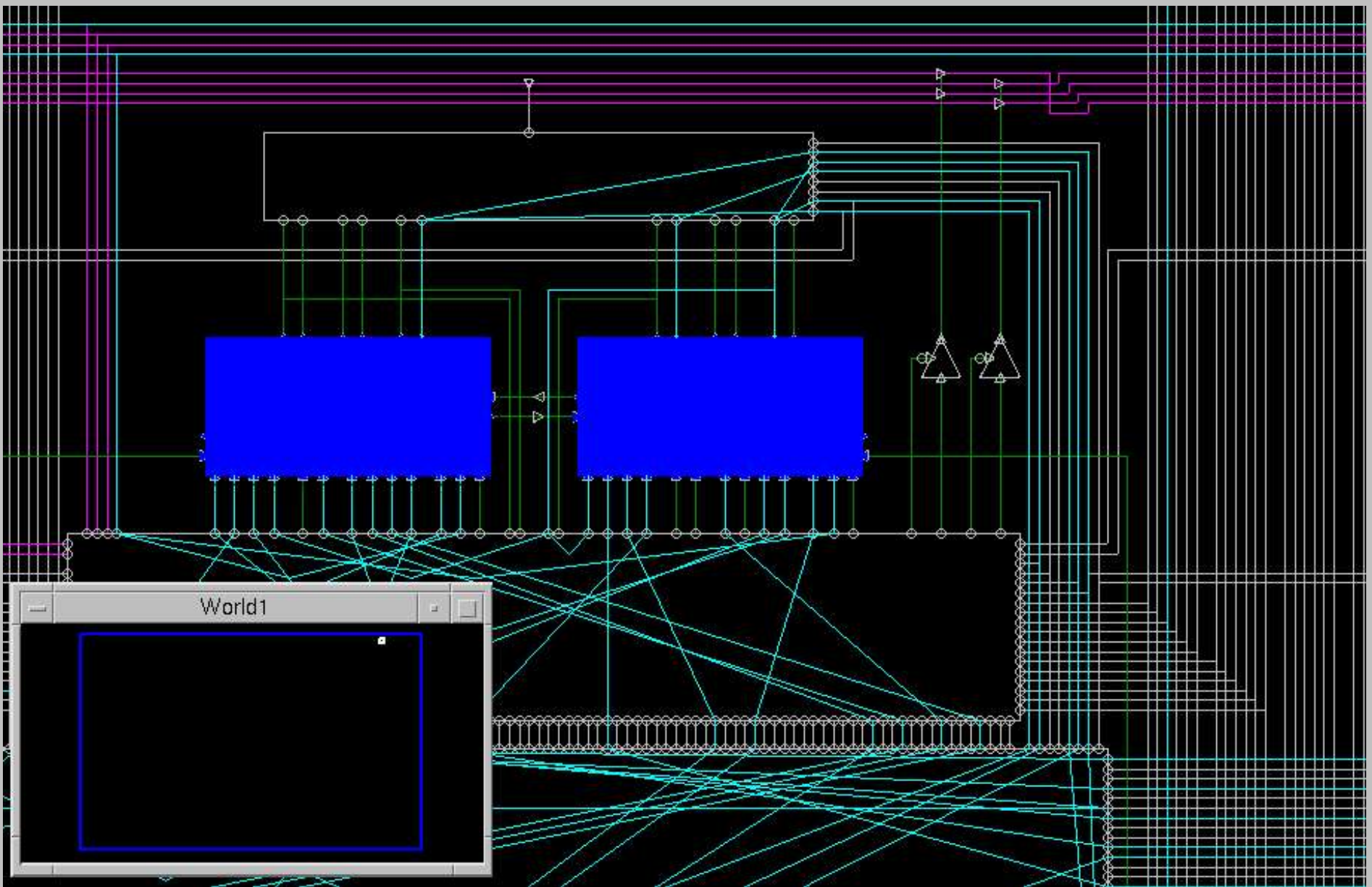


# Programming FPGA



FPGA used in the camera (Xilinx® Spartan-3) has

- 7680 universal logic cells (64% currently used);
- 24 18kbit dual port RAM blocks (83% used);
- 24 18x18 bit multipliers (54% used);
- 173 I/O blocks and (67% used);
- 4 digital clock management modules (50% used)



FPGA detail – two logic cells, switch-boxes and routing

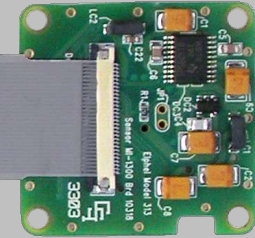
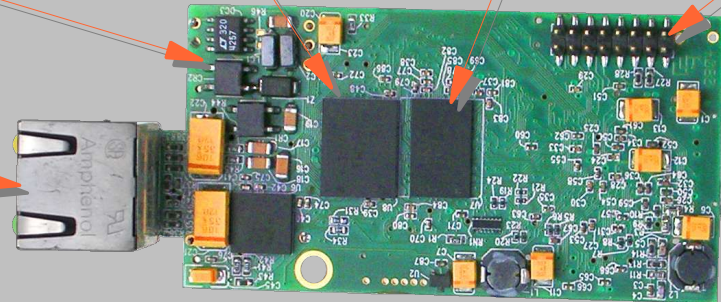
Flash  
16MB

System SDRAM  
32 MB

Extension board  
connector

IEEE 802.3af  
PoE circuitry

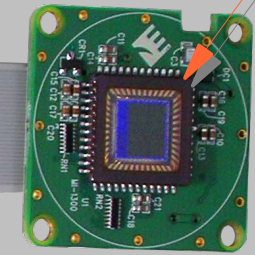
10/100 Mbps  
LAN connector



Axis ETRAX100LX SoC  
running GNU/Linux

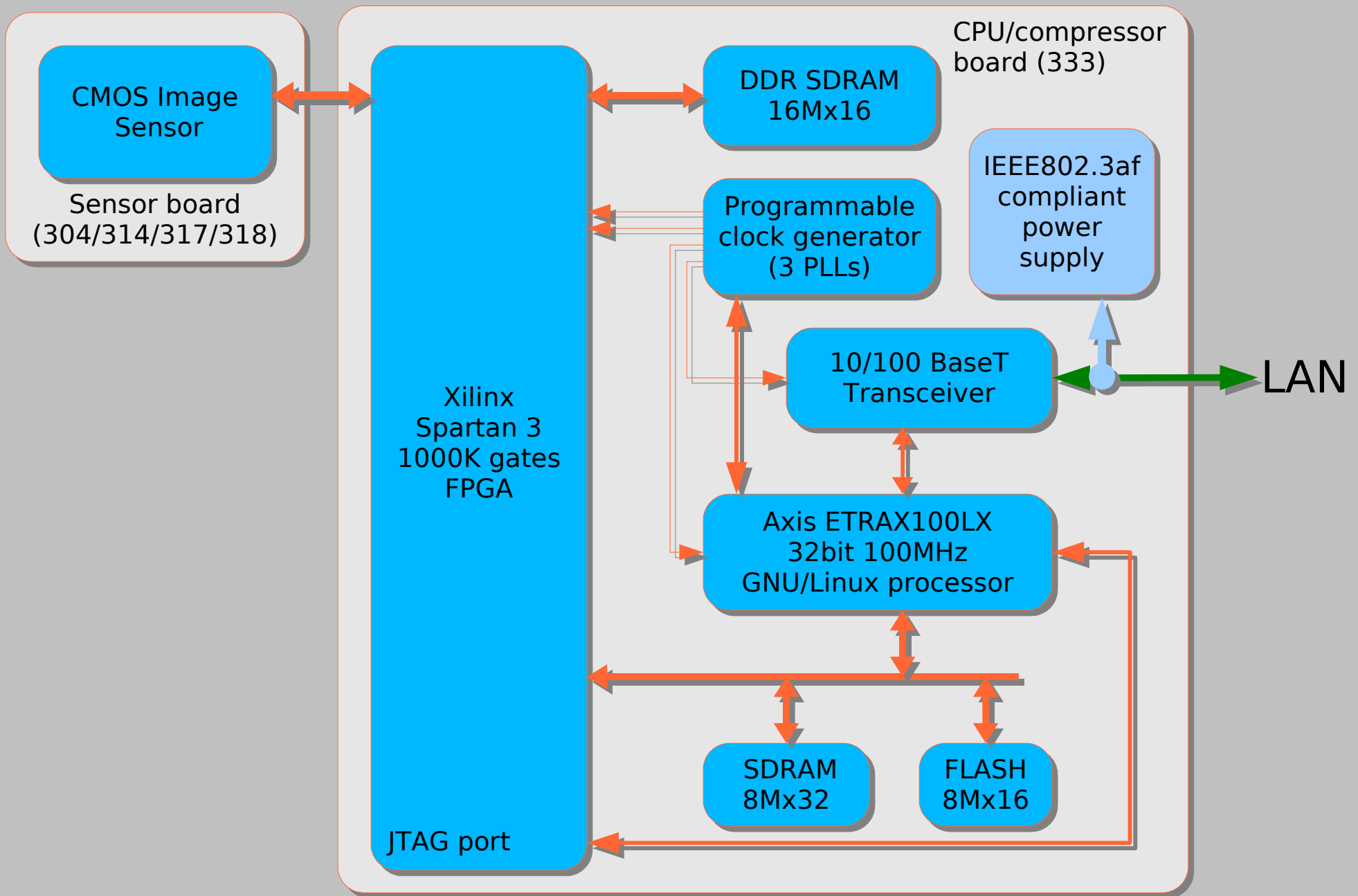
Dedicated to FPGA  
DDR SDRAM (32MB)

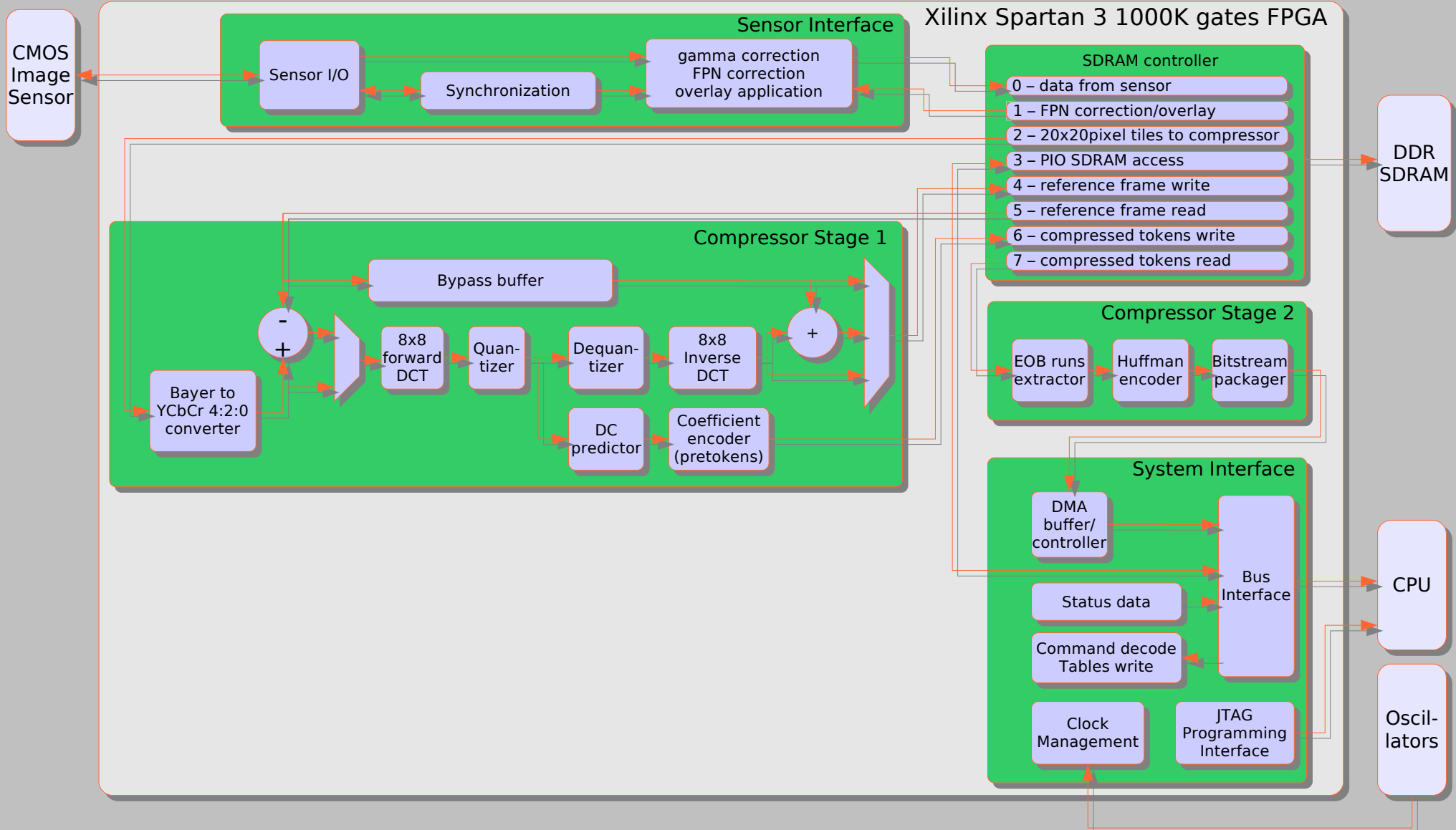
2048x1536  
10-bit CMOS  
image sensor



1 million gates  
Xilinx Spartan® 3  
Reconfigurable FPGA

3-PLL programmable  
clock generator





# Results

Third generation of Elphel cameras uses a million-gate Xilinx Spartan-3 FPGA to implement a free high-performance Ogg Theora video codec. With a 3 megapixel CMOS image sensor it can run:

- 2048x1536 at 12 fps;
- 1600x1200 at 20 fps;
- 1280x1024 at 30 fps;
- smaller frames – several hundred fps (42MPix/sec of Bayer-encoded sensor color data).

Currently only a subset of Theora features is implemented, but reconfigurable FPGA allows future code updates (not limited to Elphel developments).

# Conclusions

- **Reconfigurable FPGAs** provide performance that is often 100 times that of universal processors of comparable size and power consumption;
- Elphel products offer software developer **opportunity to experiment** in this area - all the FPGA source code of Elphel cameras is available under GNU/GPL;
- **Ogg Theora video codec** is a new one and is not yet as popular as some traditional ones, but the combination of high performance and freedom to use and modify makes it an excellent choice for video on GNU/Linux.