FPGA Programming

Tutorial

Wojciech Jalmuzna
Technical University of Lodz

LLRF 2011

17-20 October 2011, DESY, Hamburg
INTRODUCTION

• What is an FPGA
• Short FPGA History
• Programming Language vs. Hardware Description Language
• Conclusions ?

Files can be found at:
http://tesla.desy.de/~wjalmuzn/llrf11.tgz
What is FPGA? (1)

FPGA stands for Field Programmable Gate Array

The best way to show You what is an FPGA is to compare it with micro-controller, DSP or CPU
What is FPGA? (2)

Figure: Maxim 8051-Compatible uProcessor
What is FPGA? (3)

In comparison with uProcessor structure of FPGA is regular.

We have logic blocks, which contain “gates” and flip-flops.

We have interconnections, which can build connection between logic blocks,

We have IO Cells which are used to interface with outer world.

Figure: General FPGA block diagram
Short FPGA History

Market size

- 1985: First commercial FPGA technology invented by Xilinx
- 1987: $14 million
- ~1993: >$385 million
- 2005: $1.9 billion
- 2010 estimates: $2.75 billion

Market size corresponds to FPGA size (available logic gates)

Everything started with ~9000 gates and till now this amount increased by almost 1000 times. Additional embedded peripherals are now available (internal memories, fast arithmetic units, gigabit links)
Programming Language vs. HDL

Can we say that we are “programming” FPGA?

Description written in a programming language is translated into machine code, which is executed sequentially by a “program sequencer” of a target unit (CPU, DSP, uC).

Since FPGA has no predefined higher level structure (only flip-flops and logic gates) “program” should be rather called “structure description”.

Wojciech Jalmuzna, LLRF2011
Which X is better?

<table>
<thead>
<tr>
<th>X</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VENDOR</td>
<td>Altera, Xilinx ?</td>
</tr>
<tr>
<td>HDL</td>
<td>VHDL, Verilog, SystemVerilog, SystemC ?</td>
</tr>
<tr>
<td>TOOLS</td>
<td>Altera Quartus, Xilinx ISE, Synplicity ?</td>
</tr>
</tbody>
</table>

There is no correct or incorrect answer :)
Design Flow

STAGES

HDL Design Entry

Simulation

Synthesis

Implementation

Timing Analysis

Hardware

TOOLS

Any text or schematic editor, System Generator

ModelSim, ActiveHDL, RivieraPRO, gHDL, ISIM

PrecisionRTL, Synplicity, xst

ngdbuild, map, par

Timing Analyzer, PlanAhead

Impact, Chipscope
Design Entry - FIR

Describing in VHDL – simple behavioral description (as you would write in high level programming language)

```vhdl
... elsif rising_edge(P_I_CLK) then
    for I in CON_ORDER-1 downto 1 loop
        SIG_ADC(I) <= SIG_ADC(I-1);
    end loop;
    SIG_ADC(0) <= P_I_DATA;
    v_sum := (others => '0');

    for I in 0 to CON_ORDER-1 loop
        v_mult := SIG_ADC(I)*SIG_COEF(I);
        v_sum  := v_sum + v_mult;
    end loop;
    P_O_DATA <= v_sum;
...```

VHDL: Simple behavioral FIR implementation
Filter is implemented as an ADC pipeline. Each element is multiplied by a filter coefficient and all results are summed up. The operation takes one clock cycle.
Simulation Testbench

Test-bench preparation. Some syntax which cannot be synthesized

ADC SIM module has been created to provide data stream

Data from simulator has been exported to VCD file for visualization
Dedicated Module

ROM

Filter Input
Reference Output

STATE

Trigger

[Graphs showing waveforms]
Simulation - FIR
Simulation - FIR
Result Analysis - FIR
Synthesis - FIR

What is Synthesis? Conversion from some abstract description to logic gates. Special macros and FPGA built-in blocks are also detected and used. This is first step to implement design for FPGA.

- Modifications to test-bench
- Chipscope Pro for debugging
Synthesis Report

Macro Statistics

# MACs : 31
16x16-to-32-bit MAC : 31
# Multipliers : 1
16x16-bit multiplier : 1
# Registers : 512
Flip-Flops : 512

Report: FIR Macro Statistics

Timing Summary:
---------------
Speed Grade: -1

Minimum period: \textbf{66.385ns} (Maximum Frequency: \textbf{15.064MHz})
Minimum input arrival time before clock: 2.670ns
Maximum output required time after clock: 3.344ns
Maximum combinational path delay: No path found

Report: FIR Macro Statistics
Xilinx Schematic Viewer (1)

Limitations: Even for simple design RTL schematic can be huge. The best is to know which path or module part you want to observe.
FIR Schematic Viewer (2)
To Hardware

- Timing Constraints
- Implementation process
- Timing reports
- Advanced timing analysis (planAhead)
- Design verification (chipscope)
Timing Constraints

Basic timing constraint is PERIOD.

During timing analysis SETUP and HOLD violations are checked.

They cause flip-flops to go metastable
After synthesis, you run design implementation, which comprises the following steps:

**Translate** - merges the incoming netlists and constraints into a single design file.

**Map** - fits the design into the available resources on the target device, and optionally, places the design.

**Place and Route** - places and routes the design to the timing constraints.

**Generate Programming File** - creates a bitstream file that can be downloaded to the device.
Timing Analysis (1)

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Check</th>
<th>Worst Case</th>
<th>Best Case</th>
<th>Timing</th>
<th>Timing Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>* TS_P_I_V5_FPGA_CLK_P = PERIOD TIMEGRP &quot;TN&quot;</td>
<td>SETUP</td>
<td>-56.815ns</td>
<td>66.815ns</td>
<td>48</td>
<td>2727120</td>
</tr>
<tr>
<td>M_P_I_V5_FPGA_CLK_P&quot; 10 ns HIGH 50%</td>
<td>HOLD</td>
<td>0.255ns</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TS_U_TO_J = MAXDELAY FROM TIMEGRP &quot;U_CLK&quot;</td>
<td>SETUP</td>
<td>13.034ns</td>
<td>1.966ns</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TO TIMEGRP &quot;J_CLK&quot; 15 ns</td>
<td>HOLD</td>
<td>1.439ns</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TS_U_TO_U = MAXDELAY FROM TIMEGRP &quot;U_CLK&quot;</td>
<td>SETUP</td>
<td>14.103ns</td>
<td>0.897ns</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TO TIMEGRP &quot;U_CLK&quot; 15 ns</td>
<td>HOLD</td>
<td>0.568ns</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TS_J_CLK = PERIOD TIMEGRP &quot;J_CLK&quot; 30 ns H</td>
<td>SETUP</td>
<td>19.371ns</td>
<td>10.629ns</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IGH 50%</td>
<td>HOLD</td>
<td>0.324ns</td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PATH &quot;TS_D2_TO_T2_path&quot; TIG</td>
<td>SETUP</td>
<td>N/A</td>
<td>2.783ns</td>
<td>N/A</td>
<td>0</td>
</tr>
<tr>
<td>PATH &quot;TS_J2_TO_D2_path&quot; TIG</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>PATH &quot;TS_J3_TO_D2_path&quot; TIG</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>PATH &quot;TS_J4_TO_D2_path&quot; TIG</td>
<td>MAXDELAY</td>
<td>N/A</td>
<td>3.602ns</td>
<td>N/A</td>
<td>0</td>
</tr>
<tr>
<td>PATH &quot;TS_J_TO_D_path&quot; TIG</td>
<td>SETUP</td>
<td>N/A</td>
<td>3.917ns</td>
<td>N/A</td>
<td>0</td>
</tr>
<tr>
<td>PATH &quot;TS_D_TO_J_path&quot; TIG</td>
<td>SETUP</td>
<td>N/A</td>
<td>3.451ns</td>
<td>N/A</td>
<td>0</td>
</tr>
</tbody>
</table>

VHDL: Simple behavioral FIR implementation
Timing Analysis (2)
Timing Analysis (3)

<table>
<thead>
<tr>
<th>Location</th>
<th>Delay type</th>
<th>Delay(ns)</th>
<th>Physical Resource</th>
<th>Logical Resource(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLICE_X15Y43.DQ</td>
<td>Tck</td>
<td>0.450</td>
<td>fir1/SIG_ADC_0_15</td>
<td>fir1/SIG_ADC_0_15</td>
</tr>
<tr>
<td>DSP48_X1Y17.A15</td>
<td>net (fanout=11)</td>
<td>0.527</td>
<td>fir1/SIG_ADC_0_15</td>
<td>fir1/SIG_ADC_0_15</td>
</tr>
<tr>
<td>DSP48_X1Y17.PCOUT9</td>
<td>Tdspdo_APCOUT_M</td>
<td>3.832</td>
<td>fir1/Mmult_v_sum_mult0000</td>
<td>fir1/Mmult_v_sum_mult0000</td>
</tr>
<tr>
<td>DSP48_X1Y18.FCIN9</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Mmult_v_sum_mult00000_PC0</td>
<td>fir1/Mmult_v_sum_mult00000_PC0</td>
</tr>
<tr>
<td>DSP48_X1Y18.PCOUT9</td>
<td>Tdspdo_PFCINPCOUT</td>
<td>2.013</td>
<td>fir1/Maddrsub_v_sum_mult0001</td>
<td>fir1/Maddrsub_v_sum_mult0001</td>
</tr>
<tr>
<td>DSP48_X1Y19.FCIN9</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Maddrsub_v_sum_mult0001_P</td>
<td>fir1/Maddrsub_v_sum_mult0001_P</td>
</tr>
<tr>
<td>DSP48_X1Y19.PCOUT9</td>
<td>Tdspdo_PFCINPCOUT</td>
<td>2.013</td>
<td>fir1/Maddrsub_v_sum_mult0002</td>
<td>fir1/Maddrsub_v_sum_mult0002</td>
</tr>
<tr>
<td>DSP48_X1Y20.FCIN9</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Maddrsub_v_sum_mult0002_P</td>
<td>fir1/Maddrsub_v_sum_mult0002_P</td>
</tr>
<tr>
<td>DSP48_X1Y20.PCOUT9</td>
<td>Tdspdo_PFCINPCOUT</td>
<td>2.013</td>
<td>fir1/Maddrsub_v_sum_mult0003</td>
<td>fir1/Maddrsub_v_sum_mult0003</td>
</tr>
<tr>
<td>DSP48_X1Y21.FCIN9</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Maddrsub_v_sum_mult0003_P</td>
<td>fir1/Maddrsub_v_sum_mult0003_P</td>
</tr>
<tr>
<td>DSP48_X1Y21.PCOUT9</td>
<td>Tdspdo_PFCINPCOUT</td>
<td>2.013</td>
<td>fir1/Maddrsub_v_sum_mult0004</td>
<td>fir1/Maddrsub_v_sum_mult0004</td>
</tr>
<tr>
<td>DSP48_X1Y22.FCIN9</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Maddrsub_v_sum_mult0004_P</td>
<td>fir1/Maddrsub_v_sum_mult0004_P</td>
</tr>
<tr>
<td>DSP48_X1Y22.PCOUT9</td>
<td>Tdspdo_PFCINPCOUT</td>
<td>2.013</td>
<td>fir1/Maddrsub_v_sum_mult0005</td>
<td>fir1/Maddrsub_v_sum_mult0005</td>
</tr>
</tbody>
</table>

Phase Error (PE): 0.000000
### Timing Analysis (4)

<table>
<thead>
<tr>
<th>Component</th>
<th>Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP48_X1Y43.PCIN9</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Maddsub_v_sum_mult0025</td>
</tr>
<tr>
<td>DSP48_X1Y43.PCOUT9</td>
<td>Tdspdo_PCINPCOUT</td>
<td>2.013</td>
<td>fir1/Maddsub_v_sum_mult0025_P</td>
</tr>
<tr>
<td>DSP48_X1Y44.PCIN9</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Maddsub_v_sum_mult0026</td>
</tr>
<tr>
<td>DSP48_X1Y44.PCOUT9</td>
<td>Tdspdo_PCINPCOUT</td>
<td>2.013</td>
<td>fir1/Maddsub_v_sum_mult0026_P</td>
</tr>
<tr>
<td>DSP48_X1Y45.PCIN9</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Maddsub_v_sum_mult0027</td>
</tr>
<tr>
<td>DSP48_X1Y45.PCOUT9</td>
<td>Tdspdo_PCINPCOUT</td>
<td>2.013</td>
<td>fir1/Maddsub_v_sum_mult0027_P</td>
</tr>
<tr>
<td>DSP48_X1Y46.PCIN9</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Maddsub_v_sum_mult0028</td>
</tr>
<tr>
<td>DSP48_X1Y46.PCOUT9</td>
<td>Tdspdo_PCINPCOUT</td>
<td>2.013</td>
<td>fir1/Maddsub_v_sum_mult0028_P</td>
</tr>
<tr>
<td>DSP48_X1Y47.PCIN9</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Maddsub_v_sum_mult0029</td>
</tr>
<tr>
<td>DSP48_X1Y47.PCOUT9</td>
<td>Tdspdo_PCINPCOUT</td>
<td>2.013</td>
<td>fir1/Maddsub_v_sum_mult0029_P</td>
</tr>
<tr>
<td>DSP48_X1Y48.PCIN0</td>
<td>net (fanout=1)</td>
<td>0.000</td>
<td>fir1/Maddsub_v_sum_mult0030</td>
</tr>
<tr>
<td>DSP48_X1Y48.CLOCK</td>
<td>Tdspdck_PCINP</td>
<td>1.301</td>
<td>fir1/Maddsub_P_0_DATA_mult000</td>
</tr>
</tbody>
</table>

**Total:** 66.500ns (65.973ns logic, 0.527ns route) (99.2% logic, 0.8% route)
Timing Analysis (5)
Timing Analysis (6)
Timing Analysis (7)
How to solve timing problems?

If timing violation is not big (range of 1/10 of nanosecond) it may be enough to change optimization options or rerun implementation several times.

In case of our violation (~56 ns) the only solution is to go back to design entry stage.
32-input adder has been implemented as a cascade of 2 input adders (5 levels – 5 clock cycles of additional delay)
Chipscope Pro for HW Verification

Internal Logic Analyzer – its functionality can be compared to big and expensive logic analyzers, but it is much simple to use.

During design entry stage you connect signals you want to monitor.

After FPGA is programmed you define trigger conditions and examine your data :)

Be careful – it occupies resources and can change circuit behavior especially when timing constraints are defined in a wrong way.
Chipscope Pro for HW Verification
Chipscope Pro for HW Verification
Advanced Issues

- More timing constraints (OFFSET, multi-cycle paths)
- Source synchronous data transfers
- Multiple clock domains
Thank You for Your Attention

Files can be found at:
http://tesla.desy.de/~wjalmuzn/llrf11.tgz