Lecturer: Alexander Jäger

Course of studies: Technische Informatik

Student number: 3158849

Date: 30.01.2015

- What are FPGAs
- Fields of applications
- Basic FPGA Design Flow
- Vivado Standard Design Flow
- Incremental Compile
- Test Setup & Results
- Conclusion

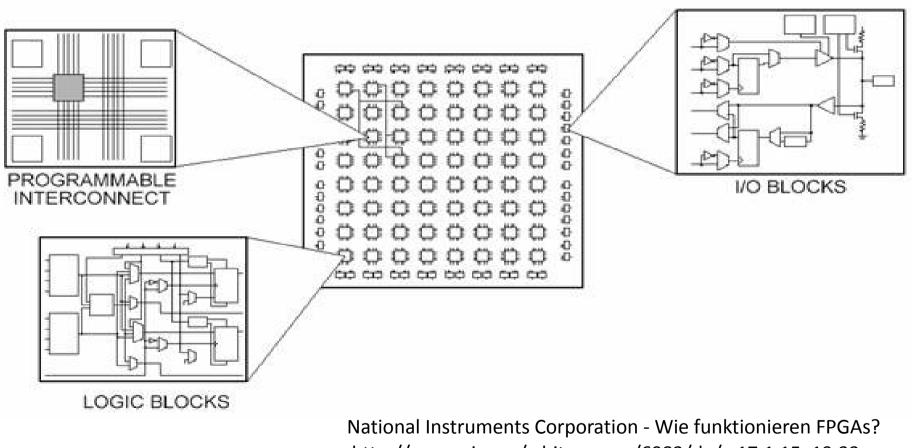


The field-programmable gate array (FPGA) is a semiconductor device that can be programmed after manufacturing. Instead of being restricted to any predetermined hardware function, an FPGA allows you to program product features and functions [...]. You can use an FPGA to implement any logical function ...

Altera Corporation - FPGAs , http://www.altera.com/products/fpga.html , 17.1.15, 19:21



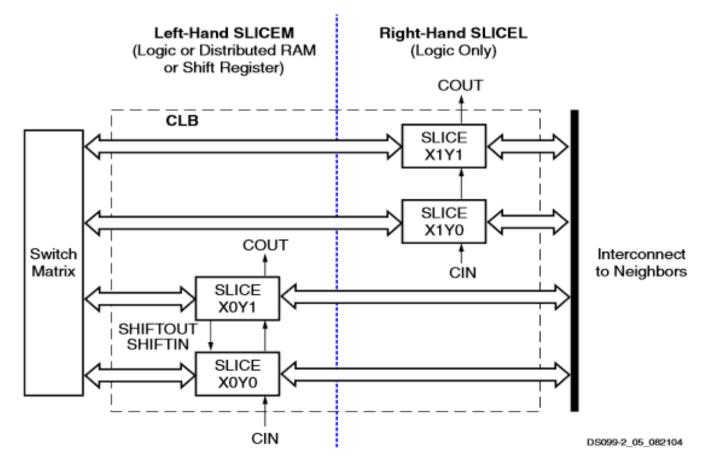
Structure:



http://www.ni.com/white-paper/6983/de/ , 17.1.15, 19:33



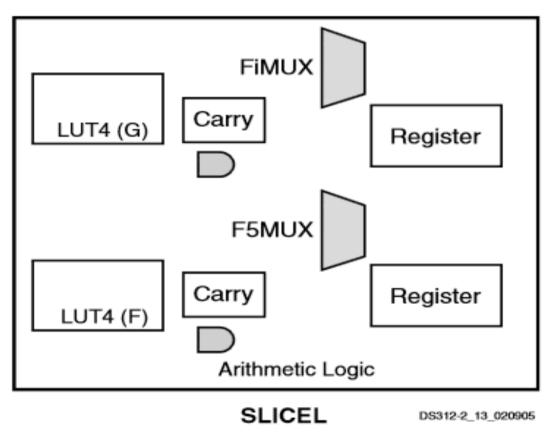
Structure of a logic block:



Xilinx Inc. - Spartan-3E FPGA Family Data Sheet, 19.7.2013, P.23



Structure of a SLICE:



Xilinx Inc. - Spartan-3E FPGA Family Data Sheet, 19.7.2013, P.23

- What are FPGAs
- Fields of applications
- Basic FPGA Design Flow
- Vivado Standard Design Flow
- Incremental Compile
- Test Setup & Results
- Conclusion



- Digital systems with small to medium quantities
- Prototyping of digital systems for evaluation and verification

Fields of applications



Reasons:

- All possible digital functions can be implemented
- User programmable
- Easy changes of the Implementation
- No mask costs



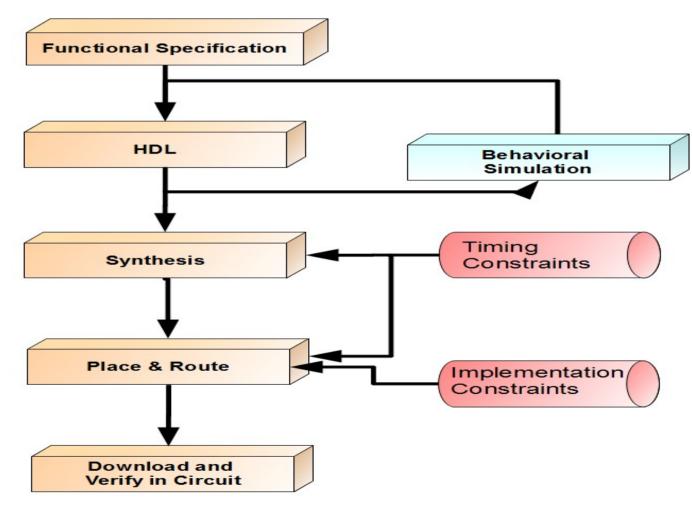
Disadvantages:

- No (flexible) analogue elements
- FPGA is slower and needs more power than an ASIC with same function
- Price per chip in High Volume Production relatively high

- What are FPGAs
- Fields of applications
- Basic FPGA Design Flow
- Vivado Standard Design Flow
- Incremental Compile
- Test Setup & Results
- Conclusion



FPGA Basic Design Flow

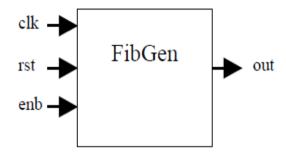




Functional Specification Minimal Example:

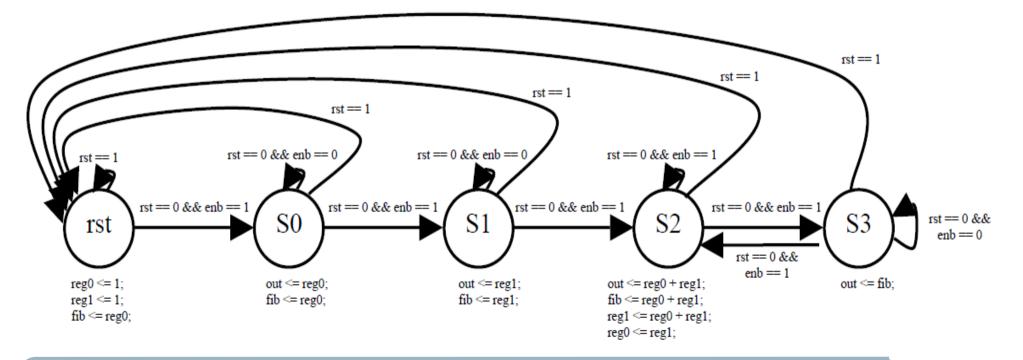
- Implement a Fibonacci number generator
- Inputs: Reset (positive), Enable, Clock
- Outputs: 16-Bit Fibonacci Number
- Target Frequency: 300 Mhz
- Frequency of sampling device: 200 Mhz
- IO-Standard: LVCMOS25





rst = 1 - resets to the beginning of the Fibonacci sequence enb = 1 - FibGen outputs Fibonacci sequence on every clk cycle

enb = 0 - FibGen stops and outputs the Fibonacci number last outputted





```
28 //
29 // Copyright (c) 2006 Susan Lysecky, University of Arizona
30 // Permission to copy is granted provided that this header remains
31 // intact. This software is provided with no warranties.
32 //
33
34 module FibGen(clk, rst, enb, out);
35
36
     input clk. rst. enb:
37
     output [16:0] out:
38
     reg [16:0] out:
39
40
     // states
41
     parameter S0 = 3'b000:
     parameter S1 = 3'b001;
42
43
     parameter S2 = 3'b010;
44
     parameter S3 = 3'b011;
45
     parameter 54 = 3'b100;
46
47
     // used to initialize registers
48
     49
     50
51
     reg [16:0] reg_0 = Zero_16;
52
     reg [16:0] reg_1 = One_16;
53
     reg [16:0] fib = Zero_16:
54
55
     reg [2:0] State;
56
57
     always @ (posedge rst or posedge clk)
58
     begin
59
        if( rst == 1 )
60
        begin
61
           req_0 = Zero_{16};
62
           reg_1 = 0ne_{16};
63
           fib = Zero_{16};
64
65
           State <= SO;
           out <= Zero_16;</pre>
66
67
        end
```

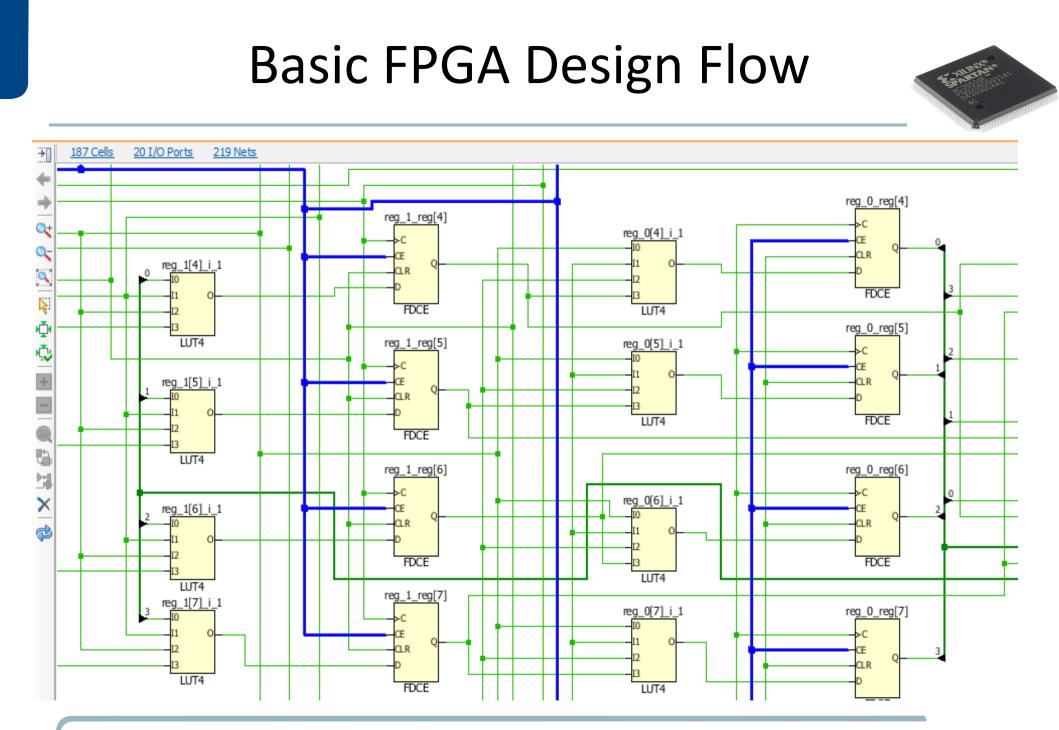
73	else
74	begin
75	case(State)
76	50
77	begin
78	// determine next state
79	if(enb == 1)
80	_ State <= S1;
81	else
82	State <= S0;
83	
84	// assign output value
85	<pre>out <= reg_0;</pre>
86	fib <= reg_0;
87	end
88	
89	S1:
90	begin
91	// determine next state
92	if (enb == 1)
93	_ State <= S2;
94	else
95	State <= S1;
96	
97	// assign output value
98	<pre>out <= reg_1;</pre>
99	fib <= reg_1;
L00	
L01	end
L02	
L03	S2:
L04	begin
L05	// determine next state
L06	if(enb == 1)
L07	State <= S2;
L08	else
L09	State <= S3;
L10	
111	<pre>// update values and assign output value</pre>
L12	<pre>out <= reg_0 + reg_1;</pre>
L13	<pre>fib <= reg_0 + reg_1;</pre>
L14	reg_0 <= reg_1;
L15	<pre>reg_1 <= reg_0 + reg_1;</pre>
L16	end

```
13 module Testbench:
14
15
      reg clk_t, rst_t, enb_t;
16
      wire [16:0] out_t;
17
      FibGen FibGen_1(clk_t, rst_t, enb_t, out_t);
18
19
20
      always
21
      begin
22
         clk_t <= 0;
23
         #25:
24
         clk_t <= 1;
25
         #25:
26
      end
27
28
      initial
29
      begin
30
31
         // reset
32
         rst_t <= 1; enb_t <= 0;</pre>
33
         #100:
34
35
         //case 0
         rst_t <= 0; enb_t <= 0;</pre>
36
37
         #100 $display("out_t = %b", out_t);
38
39
         //case 1
40
         enb_t <= 1;
41
         #500 $display("out_t = %b", out_t);
42
43
         //case 2
44
         enb_t <= 0;
45
         #100 $display("out_t = %b", out_t);
46
47
         //case 3
48
         rst_t <= 1; enb_t <= 1;</pre>
         #100 $display("out_t = %b", out_t);
49
50
51
      end
52 endmodule
```



1 create_clock -period 3.333 -name clk -waveform {0.000 1.667} [get_ports clk]
2 create_clock -period 5.000 -name clk_virt -waveform {0.000 2.500}
3 set_clock_groups -name out_clk -asynchronous -group [get_clocks clk_virt]
4

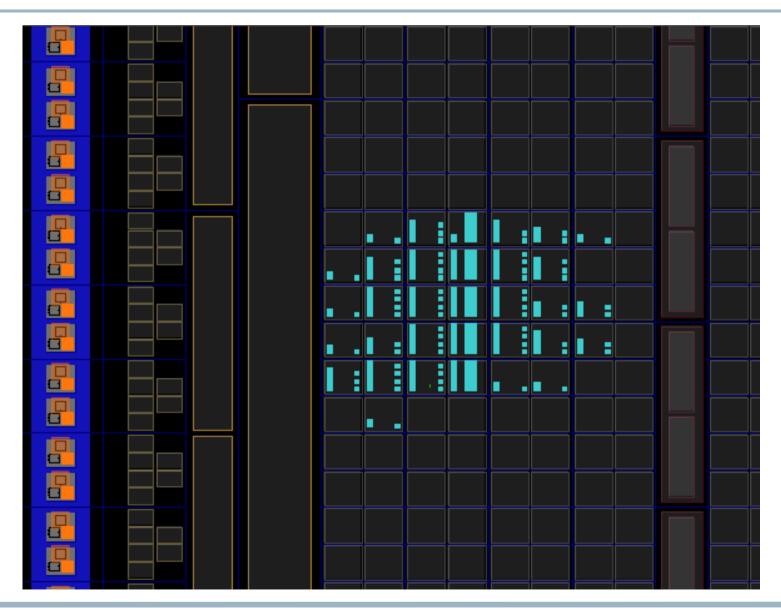
5 set_input_delay -clock [get_clocks clk] -min -add_delay 1.000 [get_ports enb] 6 set_input_delay -clock [get_clocks clk] -max -add_delay 1.500 [get_ports enb] 7 set_input_delay -clock [get_clocks clk] -min -add_delay 1.000 [get_ports rst] 8 set_input_delay -clock [get_clocks clk] -max -add_delay 1.500 [get_ports rst] 9 set_output_delay -clock [get_clocks clk_virt] -min -add_delay 0.900 [get_ports {out[*]}] 10 set_output_delay -clock [get_clocks clk_virt] -max -add_delay 1.000 [get_ports {out[*]}] 11

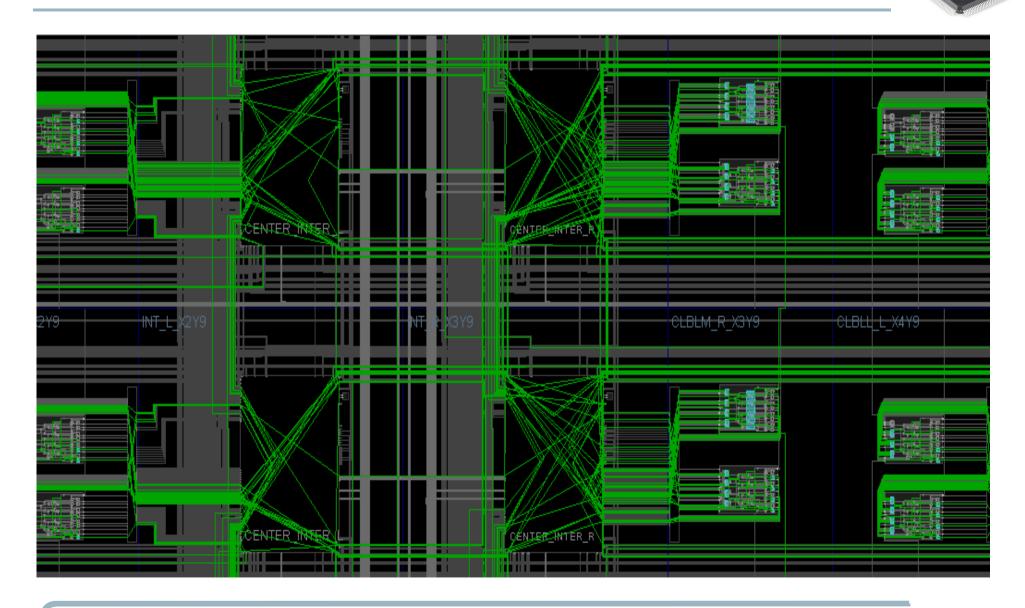




11 set property PACKAGE_PIN R16 [get ports {out[8]}] 12 set property PACKAGE PIN T16 [get ports {out[7]}] 13 set property PACKAGE_PIN W16 [get ports {out[6]}] 14 set property PACKAGE PIN Y16 [get ports {out[5]}] 15 set property PACKAGE_PIN W14 [get ports {out[4]}] 16 set property PACKAGE_PIN Y14 [get ports {out[3]}] 17 set property PACKAGE_PIN V15 [get ports {out[2]}] 18 set property PACKAGE PIN W15 [get ports {out[1]}] 19 set property PACKAGE PIN T15 [get ports {out[0]}] 20 set property PACKAGE PIN U15 [get ports rst] 21 set property IOSTANDARD LVCMOS25 [get ports {out[16]}] 22 set property IOSTANDARD LVCMOS25 [get ports {out[15]}] 23 set property IOSTANDARD LVCMOS25 [get ports {out[14]}] 24 set property IOSTANDARD LVCMOS25 [get ports {out[13]}] 25 set property IOSTANDARD LVCMOS25 [get ports {out[12]}] 26 set property IOSTANDARD LVCMOS25 [get ports {out[11]}] 27 set property IOSTANDARD LVCMOS25 [get ports {out[10]}] 28 set property IOSTANDARD LVCMOS25 [get ports {out[9]}]

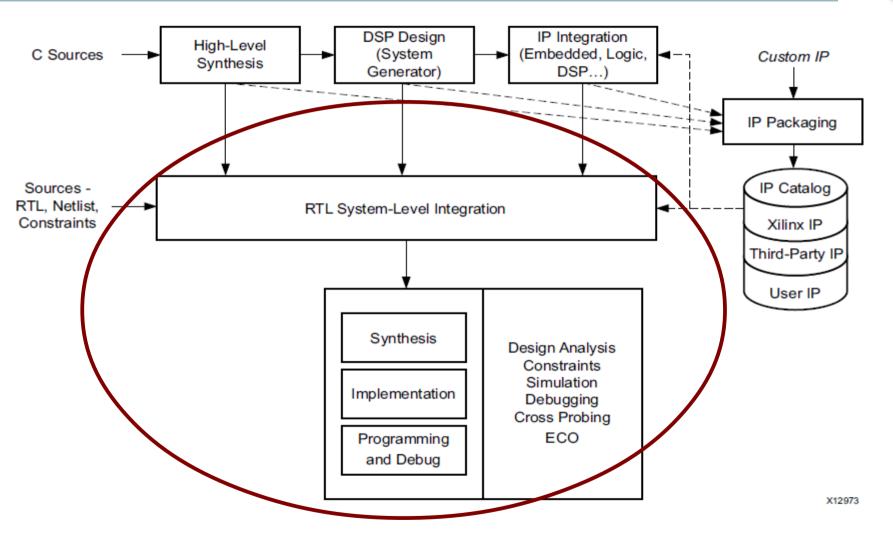






- What are FPGAs
- Fields of applications
- Basic FPGA Design Flow
- Vivado Standard Design Flow
- Incremental Compile
- Test Setup & Results
- Conclusion

Vivado Standard Design Flow



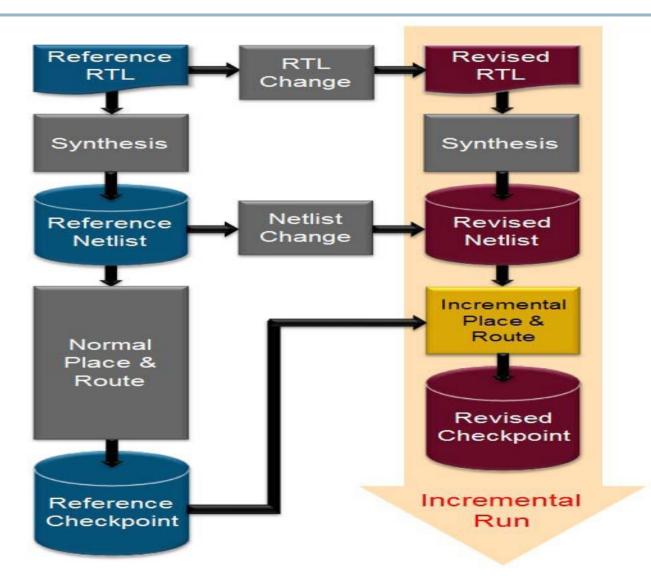
Xilinx Inc. - Vivado Design SuiteUser Guide Design Flows Overview, 1.10.14, P.6

Vivado Standard Design Flow

Novigate Novig	<u>Edit Flow Tools Window La</u>	ayout View Help											Q.+ Sear	rch comma	ands
Implementation Constraints Wilds Wilds Constraints Wilds Wilds Constraints Wilds Constraints Wilds Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints Wilds Constraints <			Σ 🎯 📀 🍕 🔀	🛯 🧐 🔚 Default La	yout	-)	K 🚸 🔭	E)					-		
Project Hanager @ Project Settings @ Add Sources @ Language Templates @ Language Templates @ Densition Settings @ Cenerate Block Design @ Generate Block	w Navigator «	Implemented	Design - impl_9_entries96	xc7vx690tffg1761-2											
Vroject Hanager W v720_top Ge Project Statings W v720_top W v720_top W v720_top W v720_top <th></th> <th>Netlist</th> <th></th> <th>_ [</th> <th>l 🗠 🗡</th> <th>ΣΡ</th> <th>roject Summa</th> <th>ry ×</th> <th>Device</th> <th>e 🗙 🐼 hmc_e</th> <th>controller_</th> <th>top.v</th> <th>×</th> <th></th> <th>000</th>		Netlist		_ [l 🗠 🗡	ΣΡ	roject Summa	ry ×	Device	e 🗙 🐼 hmc_e	controller_	top.v	×		000
Vroject Hanager W v720_top Ge Project Statings W v720_top W v720_top W v720_top W v720_top <th></th> <th>🔀 🖄 🖪</th> <th></th> <th></th> <th></th> <th>÷∎</th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>		🔀 🖄 🖪				÷∎									
^A Add Sources ^A carde Sources ^A corder Sources					*										
We doublet We doublet Sources Sources<	ô Project Settings	😟 🖓 🔂 Nets (2019				태				XOYS) X1 Y	9			
Windless	🚰 Add Sources				=	+									
P Integrator	🖓 Language Templates					\rightarrow				SY UX		8			
Implementation Implem	E IP Catalog			er_top)						Mova		_			
P Integrator															
	P Integrator			en_top)	-					ХПУР	X1Y	6			
Open Block Design Marketiss Generate Block Design Minulation Synthesis Description: Windo Implementation Defaults Description: Windo Implementation Defaults Synthesis Description: Synthesis Design Runs Constraints WNS Manuel Design Manuel Defaults Manuel Design Manuel Defaults Manuel Design Manuel Defaults Manuel Design Runs Constraints Manuel Design Name Manuel Design Constraints Manuel Design Manuel Design Manuel Design Constraints Manuel Design Manuel Design Manuel Design Manuel Design Manuel Design Manuel Design Manuel Design Manue	ở Create Block Design					9									
Generate Block Design implementation Run Properties implementation Settings implementation Run Properties implementation Settings implementatel Design		& Sources	🕅 Netlist							X0Y5	X1 Y				
implantion impla_g_entries96 impla_g_entries96 impla_g_entries96		Implementation F	un Properties	- 1	л с ×				Ì						
minulation	Generate block Design									, xorz		4			
Implementation Name: mpl_9_entries96 Implementation Implementatio	imulation										و کنا ا		- 414		
• Run Simulation • Run Simulation • Run Allysis • Open Elaborated Design • Synthesis • Synthesis • Open Synthesised Design • Mingle_entries256 ic constrs_1 • onstraints • Synth_9_=entries266 ic constrs_1 • onstraints • Synth_9_=entries266 ic constrs_1 • onstraints • onstraints • onstraints • onstraints • onstraints	Simulation Settings	/ impi_9_entries	90			6					1 N X1Y				
RTL Analysis Description: Wixdo Implementation Defaults Synthesis Synthesis Synthesis Synthesis Sopen Synthesis Design Runs Implementation Sopen Synthesis Sope	-	Name:	impl 9 entries96			396				I NOV			2 F F F		
RTL Analysis Det to the excitation of the properties of	www.kun Sinulation				=					AU12		-			
 	RTL Analysis	Part:	xc7vx690tffg1761-2 (ac	tive)						X0Y1	1 X1Y	1			
Synthesis Synthesis Open Synthesized Design Name Constraints WNS TNS WHS THS TPWS Failed Routes LUT FF BRAM DSP Start Implementation Implementation Settings Implementation Implement	Den Flaborated Design	Description:	Vivado Implementation Defa	aults 🙁	-										
Synthesis Settings		<	III							ΧΟΥΟ	X1 Y	o			
Synthesis Settings Name Constraints WNS TNS WHS THS TPWS Failed Routes LUT FF BRAM DSP Start Implementation Implementation Settings Implementation Impl	Synthesis	General Prop	erties Options Log Repo	orts Messages		2									
Nume Constraints WNS TNS WHS THS TPWS Failed Routes LUT FF BRAM DSP Start Implementation Implementation Settings Implementation Implementation 0.00 0.00 0.00 0.00 0.00 0.00 0.00 17.14 0.00 1/5/15 4: Implementation Implementation Settings Implementation Implementation 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1/5/15 4: Implementation Implementation Implementation Implementation Implementation 0.00 0.00 0.00 0.00 0.00 0.00 0.00 1/5/15 4: Implementation Implem	🏀 Synthesis Settings														
Name Constraints WNS TNS WHS THS TPWS Failed Routes LUT FF BRAM DSP Start Implementation Implementation Implementation Settings Implementation Settings Implementation 4.28 2.02 1.09 0.00 1/5/15 6: 0.00 0.00 0.00 0.00 0.00 1/5/15 6: 0.00 1/5/	Run Synthesis														
implementation implementation Settings				Constraints	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	DSP	
mplementation impl_8_entries256_ic constrs_1 0.10 0.00 0.05 0.00 0 5.31 3.07 17.14 0.00 1/5/15 8: implementation Settings impl 9_entries96 constrs_1 0.11 0.00 0.00 0 0 5.17 2.85 17.14 0.00 1/6/15 11: implementation impl 9_entries96_ic (active) constrs_1 0.11 0.00 0.00 0 0 5.17 2.85 17.14 0.00 1/6/15 11: implemented Design impl 9_entries96_ic (active) constrs_1 0.10 0.00 0.00 0 0.17 2.85 17.14 0.00 1/6/15 12: implemented Design impl 9_entries96_ic (active) constrs_1 0.10 0.00 0.00 0.517 2.85 17.14 0.00 1/6/15 12: implementations implemented Runs implementation implementation 0.00 0.00 0.00 0.00 1/6/15 12: implementations implementation implementation implementation 0.00 0.00 0.00 0.00 0.00 1/6	B open of the leaded bearging	🔀 📋 🦾 🔏 imp		_	0.06	0.00	0.01	0.00	0.00						1/5/15 6:
Implementation Settings Implementation Settings Implementation Settings Implementation Implemen	mplementation			_	0.10	0.00	0.05	0.00	0.00						
Run Implementation Implementation Implementation Implemented Design Implement	•			_	0.10	0.00	0.05	0.00	0.00						
Implemented Design				-	0.11	0.00	0.01	0.00	0.00		_				
rogram and Debug		📄 🗸 synth	_9_entries96_ic (active)	constrs_1							4.15	1.79	1.09	0.00	1/6/15 12:2
rogram and Debug	Implemented Design			constrs_1	0.10	0.00	0.06	0.00	0.00		0 5.17	2.85	17.14	0.00	1/6/15 12:3
⁽¹⁾ ⁽¹	rogram and Debug			-ll 200 to :2-							0.00	0.00	0.00	0.00	11/1/10/1
Generate Bitstream	-														
Generate Bitstream				-											11/17/14 10:
	Cenerate Bitstream														

- What are FPGAs
- Fields of applications
- Basic FPGA Design Flow
- Vivado Standard Design Flow
- Incremental Compile
- Test Setup & Results
- Conclusion

Incremental Compile



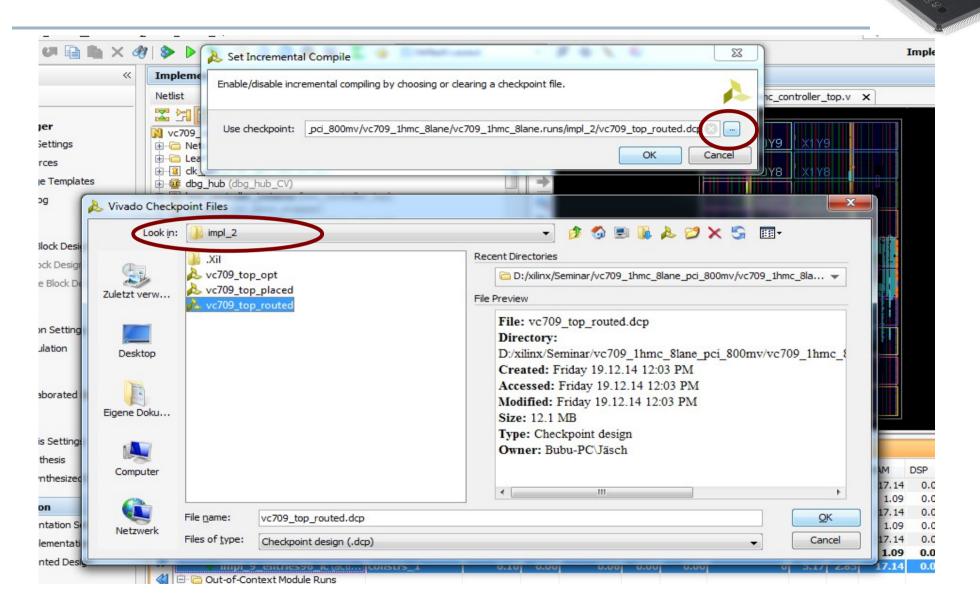
Xilinx Inc. -Vivado Design Suite User Guide Implementation 15.10.14 P.83

Incremental Compile

	Create New Runs					
	This wizard will guide you through the process of creating and launching multiple synthesis or/and implementation runs. You can then apply different sets of command options, or strategies, to these runs and launch them all at once.					
	What type runs do you want to create?					
	Synthesis					
	Implementation					
VIVADO.	⊚ <u>B</u> oth					
VIVADO.	To continue, click Next					

Eart. Wxc/vxosotrig1/01/		_		11.6	W1
Description: Vivado Implementation	Defaults 🛞	9	Implementation Run Properties	Strg+E	
< III		X	Delete	Entf	n i
General Properties Options Log	Reports Messages		Make Active		
Design Runs			Change Run Settings		
			Set Incremental Compile		
Name	Constraints		Save As Strategy		ed Routes
🔀 🐘 🔏 impl_8_entries256	constrs_1		Save As Strategy		
synth_8_entries256_ic	constrs_1	₿	Open Run		
Minipi_o_enules250_ic	constrs_1		Launch Runs		
synth_9_entries96	constrs_1	K	Reset Runs		
✓ impl_9_entries96	constrs_1				-
synth_9_entries96_ic (act			Launch Next Step: write_bitstream		
	cti constrs_1		Launch Step To	•	
🕻 🖻 🛅 Out-of-Context Module Runs			Reset to Previous Step: route_design		
⊕ √ ck_gen_200_to_i2c_synth		2	Generate Bitstream		-
a via_i_synui_i	ila_1	-			-
Image: Image	n_1 hmc_transceiver_	•	Display Run Log		
		B	Display Run Reports		
🔪 🔲 Td Console 📔 🗩 Messages 🗌 🕅	Log 🗋 Reports 🗐	P	Display Run Messages		
		D	Copy Run		
		\$	Create Runs		
erte Objekt. Zum Skalieren ziehen Sie die Ziehpunkte. Zum 🛛			Open Run Directory		
🛛 🕑 🖳			Export to Spreadsheet		

Incremental Compile





Short Summary:

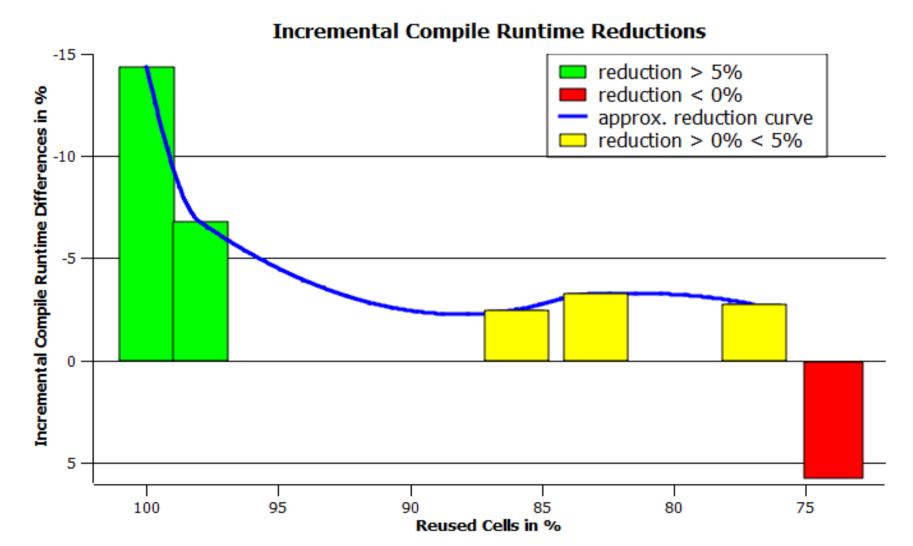
- A minimum of 85% match between original and new netlist required
- Design Checkpoint from previous Implementation needed
- Checkpoint can be (partially) placed or (partially) placed and (partially) routed

- What are FPGAs
- Fields of applications
- Basic FPGA Design Flow
- Vivado Standard Design Flow
- Incremental Compile
- Test Setup & Results
- Conclusion



- Starting Point: Fully implemented design with checkpoints
- List of small to big changes
- 2 runs for every list entry One with Standard Flow, One with Incremental Compile
- Compare: Runtime, Timing, Resource Utilization

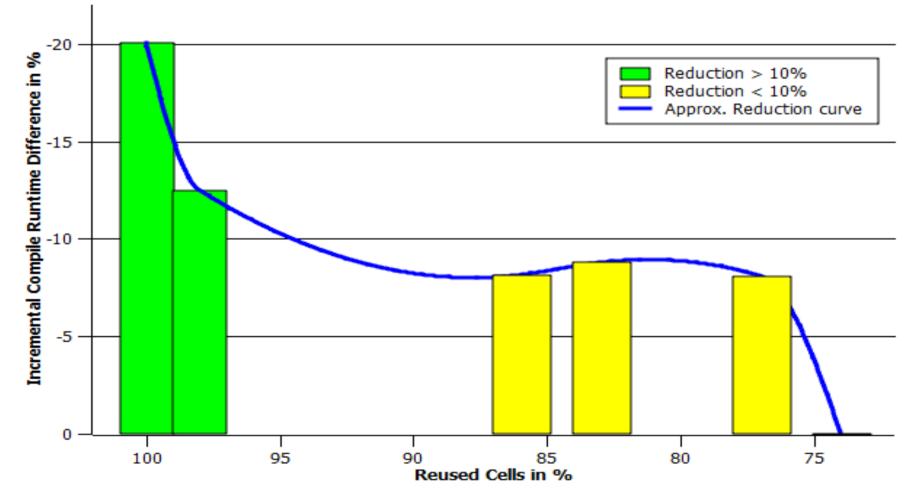




30/01/15



Calculated Theoretical Runtime Reductions





Runtime Facts:

- Best measured runtime reduction: 21.22 %
- Best theoretical reduction: 26.81 %
- Average runtime reduction: 6.19 %
- Additional runtime inducted trough Incremental Compile: 1:45 min



Timing results:

• No influence, all required timings met



Resource utilization:

- No influence on used Block Ram and Slice Registers
- Utilization of Slice LUTs, LUT FF-Pairs and used Slices stayed same or dropped a bit

- What are FPGAs
- Fields of applications
- Basic FPGA Design Flow
- Vivado Standard Design Flow
- Incremental Compile
- Test Setup & Results
- Conclusion

Conclusion



Incremental Compile:

- Easy to use
- Overall small runtime reductions
- Sometimes small resource utilization reductions
- => Only minor improvements, still recommended for usage

Discussion





Any Questions?

Xilinx IC: http://en.wikipedia.org/wiki/Xilinx#mediaviewer/File:Xilinx_Spartan-3E_%28XC3S500E%29.jpg

30/01/15