Vendor-agnostic probing of FPGA designs For efficient run-time debugging



Alexander Wirthmüller aw@mpsitechnologies.com

Introduction

About me

- Based in Munich
- Diploma in Electrical Engineering
- Founder and Director at MPSI Technologies
- MPSI Technologies: make Embedded Software development more pleasant – by replacing repetitive tasks with model-based source code generation
- Senior Staff Engineer with Symeo / indie Semiconductor (industrial radar)







FPGA Design Probing Scope | IDE-based vs. generic | Platforms

- For CPU code, iterations (compile + link) are relatively fast:
 Use debugger iff problem encountered (unexpected behavior, segmentation fault, ...)
- Unit and integration tests are mandatory in quality projects
- Test-driven development also exists (but is the exception)
- FPGA designs, are costly (time-wise) to iterate, also harder to grasp for the human brain: Logic test-benches and simulation are part of every design
- Complexity starts with integration [of designs from other vendors]
- Full design simulation and randomized testing still possible (and done)
- "Outside-world" triggers and concurrency often hold surprises → case for design probing



FPGA Design Probing Scope | IDE-based vs. generic | Platforms

Vendor offerings

- Altera: Signal Tap Logic Analyzer, Signal Probe + System Console
- AMD: Virtual I/O + Vivado
- Efinix: Virtual I/O + Efinity / GTKwave
- Lattice: Reveal + Diamond Microchip: SmartDebug + Windows UI
- → all require proprietary IP blocks and/or software, most use the JTAG interface as debug port

This work

- General-purpose tracking of 15-bit wide signals with (time-domain) compression
- FSM state tracking + statistics with compression
- Read-out to CPU and .vcd output for visualization in GtkWave

Key enablers

- PHY-agnostic CPU (host) interface with bytecode protocol
- Vendor-agnostic instantiation of memory primitives
- Powered by Open Source WhizniumDBE



FPGA Design Probing Scope | IDE-based vs. generic | Platforms

Lattice CrossLink-NX eval board (cleb)

- Device LIFCL-40-9BG400C
- Host: PC / NUC (UART-over-USB)
- X DDR memory



Efinix Titanium Ti180 dev kit (tidk)



- Device Ti180M484I3
- Host: Buildroot Linux on soft RISC-V (AXI/32)
- ✓ 256 MB LPDDR4 SDRAM

Avnet ZUBoard 1CG (zudk)

- Based on Zynq UltraScale+ 1CG
- Device ZU1CG-1SBVA484I
- Host: Yocto Linux on ARMv8 (AXI/64)
- ✓ 1GB DDR4 SDRAM





Example #1: Pushbutton De-bounce Context | Capture | Gptrack_Easy_v1_0 | Data format | Readout



Vendor-agnostic probing of FPGA designs For efficient real-time debugging



Example #1: Pushbutton De-bounce Context | Capture | Gptrack_Easy_v1_0 | Data format | Readout





Example #1: Pushbutton De-bounce

Context | Capture | Gptrack_Easy_v1_0 | Data format | Readout





Example #1: Pushbutton De-bounce Context | Capture | Gptrack_Easy_v1_0 | Data format | Readout

• 32-bit words, raw vs. burst flag, two captures (raw) vs. capture + duration (burst)

	Seq	<u>uence buffer</u>																					
	31		24	23			16	15						8	7							0	
quad 0 (raw)	0		capt	/15	(t=1)			0					ca	pt/	15 (t=0)							
quad 1																							
quad 2 (burst)	1		burs	tCap	ot/15				burst	Cnt: t	rkclk	cycles	no c	han	ge (I	up te	o 0.6	66 m	ıs @	100	MH	z)	
quad sizeSeqbuf/4-1																							



Example #1: Pushbutton De-bounce Context | Capture | Gptrack_Easy_v1_0 | Data format | Readout

• Three commands for trigger selection, acquisition start and status polling

```
mgptrack.select(
```

staTixVTrigger={void,btn0.rising,btn0.falling,ackInvTkclksrcSetTkst.rising},
stoTixVTrigger={void,btn0.rising;btn0.falling;ackInvTkclksrcSetTkst.rising})

mgptrack.set(rng={false,true},TCapt=[uint32])

(tixVState{idle,arm,acq,done}) = mgptrack.getInfo()

• One buffer transfer for resulting sequence buffer

(file) = seqbufMgptrackToHostif.read(reqlen[uint32])



Example #2: CPU-FPGA Transaction Context | Capture | Fsmtrack_Easy_v1_0 | Data format | Readout

• PHY-agnostic CPU host interface with byte protocol to invoke commands / write/read buffers





Example #2: CPU-FPGA Transaction

Context | Capture | Fsmtrack_Easy_v1_0 | Data format | Readout

• FSM states for individual sub-transfers and mapping to ID (C enum style)

type stateOp_t is (
 stateOpInit,
 stateOpInit,
 stateOpIdle,
 stateOpRxopA, stateOpRxopB, stateOpRxopC, stateOpRxopD, stateOpRxopE,
 stateOpTxpollA, stateOpTxpollB, stateOpTxpollC, stateOpTxpolLD, stateOpTxpolLE, stateOpTxpolLF, stateOpTxpolLG,
 stateOpTxA, stateOpTxE, stateOpTxB, stateOpTxC, stateOpTxD, stateOpTxF, stateOpTxG,
 stateOpTxbufA, stateOpTxbufB, stateOpTxbufC, stateOpTxbufD, stateOpTxbufE, stateOpTxbufF, stateOpTxbufG, stateOpTxbufA, stateOpRxB, stateOpRxD,
 stateOpRxA, stateOpRxB, stateOpRxC, stateOpRxD,
 stateOpRxbufA, stateOpRxB, stateOpRxD,
 stateOpRxbufA, stateOpRxbufB, stateOpRxbufC, stateOpRxbufD, stateOpRxbufE, stateOpRxbufF,
 stateOpRxbufA, stateOpRxbufB, stateOpRxbufC, stateOpRxbufB, stateOpRxbufE,
 stateOpRxbufF,
 stateOpRxbufA, stateOpRxbufB, stateOpRxbufC,
 stateOpRxbufA, stateOpRxbufF,
 stateOpRxbufA, stateOpRxbufB, stateOpRxbufC,
 stateOpRxbufA,
 stateOpRxbufA,

stateOp_dbg <= x"00" when stateOp=stateOpInit
 else x"10" when stateOp=stateOpIdle
 else x"20" when stateOp=stateOpRxopA
 else x"21" when stateOp=stateOpRxopB
 else x"22" when stateOp=stateOpRxopC
 else x"23" when stateOp=stateOpRxopD
 else x"24" when stateOp=stateOpRxopE
 else x"30" when stateOp=stateOpTxpollA
 else x"31" when stateOp=stateOpTxpollB
 ...
 else x"80" when stateOp=stateOpInv
 else (others => '1');



Example #2: CPU-FPGA Transaction Context | Capture | Fsmtrack_Easy_v1_0 | Data format | Readout

- DUT comprises host interface and connected module 10 kHz clock source
- Trigger on first received byte from CPU
- Capture FSM progress and tkst (time stamp) update
- CPU-triggered command is tkclksrc.setTkst(tkst=0)





Example #2: CPU-FPGA Transaction

Context | Capture | Fsmtrack_Easy_v1_0 | Data format | Readout

		•															GTKV	Vave - te	st/2024	40702_v	cd/cmdi	nv.txt.vc	d		
	×		Ē	00	¢	G	≪	\gg	<u></u>	•>>	From	: 0 sec		To:	270 ns			C							
·	SST					Tin	ne					0								10	00 ns				
	- m	dut						h	ostif	State0	2	rxopA	rxopB	rxopC	rxopD	rxA	r*xB		rxC	rxD	inv		idle	rxopA	
								tkc	lksrcS	State0	b	run										(inv		run	
						tko	lksrc	GetT	kstTks	t[7:0]		4C										00			
L																									
F	Гуре	Signa	ls																						
5	tring	hostif	StateOp																						
۷	vire	tkclksr	cGetTks	tTkst[7:	0]																				
5	tring	tkclksr	cStateO	p																					



Example #2: CPU-FPGA Transaction Context | Capture | Fsmtrack_Easy_v1_0 | Data format | Readout

• FSM state statistics output (different example)

```
states covered: 0x00 (init), 0x01 (idle), 0x10 (runA), 0x11 (runB), 0x20 (done)
states missed: 0x12 (runC), 0x30 (reset)
first state occurrence in ms:
[0.000000] 0x00 (init)
[0.000010] 0x01 (idle)
[0.003320] 0x10 (runA)
[0.003348] 0x11 (runB)
[0.019200] 0x20 (done)
state occurrence and share:
0x00 (init): 88.0% (1760/2000)
0x01 (idle): 0.1% (2/2000)
0x10 (runA): 3.3% (66/2000)
0x11 (runB): 6.6% (132/2000)
0x20 (done): 2% (40/2000)
```



Example #2: CPU-FPGA Transaction Context | Capture | Fsmtrack_Easy_v1_0 | Data format | Readout

- Mux'ed capture port
- Selectable trigger
- Count buffer
- First occurrence buffer
- Variable size sequence buffer
- Tracking clock (trkclk)
- Master clock for readout (mclk)
- Adaptive readout width





Example #2: CPU-FPGA Transaction

Context | Capture | Fsmtrack_Easy_v1_0 | Data format | Readout

-		Cor	int huf	for																			-		-	-				+		
		<u></u>																							-		-					
		31						24	23							16	15							8	7							0
	state 0x00	cnt	: trkclk	cycles	s in s	tate	0x00) fro	m st	tart t	o st	op t	rigge	er (e	.g. u	p to	43 s	. @ 1	100	MHz)											
	max. 0xFF																															
		<u>Firs</u>	t occu	rrence	e buf	ffer													_									_				
				_				~ 4	~ ~							10	4.5								-			_				
	state 0x00	31		elle ere	elee f		fr	24	23			فم اند	at a l	2200)	16	15	haf			0E			8	/	4/:	a va li					0
	state 0x00	isu	DCC: UTK	сік су	cies i	rom	Tron	n su	artu	ngge	run	uist	ate	JXUU	0000	ursi	ort	nen	irst t	ime;	UXF	FFFF	TTT	rese	rve	u / II	ivai	a				
	max_0xEE																								1							
	IIIda. OATT																															
		Sec	uence	buffe	er																									-		
		31						24	23							16	15							8	7							0
	quad 0 (raw)		s	tate(t	:=3)						sta	te(t	=2)						sta	te(t	=1)						st	ate(t=0)			
	quad 1																															
	quad 2 (burst)			0xFl	F						bu	rstSt	tate				bı	urst	Cnt:	trkc	k cy	cles	in bu	irst	State	e (up	to (0.66	ms	@1	00 M	Hz)
	quad sizeSeqbuf/4-1	ater				/ 1000	alid																				_					
		sta	te uxFF	reser	ved	/ inv	alid																					_				



• Three commands for source/trigger selection, acquisition start and status polling

```
mfsmtrack0.select(tixVSource={hostifOp},
    staTixVTrigger={void,hostifRxAXIS_tvalid,ackInvTkclksrcSetTkst},
    stoTixVTrigger={void,hostifRxAXIS_tvalid,ackInvTkclksrcSetTkst})
```

mfsmtrack0.set(rng={false,true},TCapt=[uint32])

(tixVState{idle,arm,acq,done},coverage[blob32]) = mfsmtrack0.getInfo()

- Three buffer transfers for resulting count, first occurrence and sequence buffers
 - (file) = cntbufMfsmtrackOToHostif.read(reqlen[uint32])
 - (file) = fstoccbufMfsmtrack0ToHostif.read(reqlen[uint32])
 - (file) = seqbufMfsmtrack0ToHostif.read(reqlen[uint32])



Example #3: Time-multiplexed DDR Memory Access Context | Capture | CDC & Performance

- Ddrmux module template to share DDR memory access across multiple read/write ports
- AXI4 burst mode transfers (here: 16 beats @ 16 byte)
- Deterministic transfers from / to client buffers perturbed by random trafgen traffic
- Capture request / acknowledge handshakes



Example #3: Time-multiplexed DDR Memory Access Context | Capture | CDC & Performance





Example #3: Time-multiplexed DDR Memory Access Context | Capture | CDC & Performance

• Trigger signals are CDC'ed before the input mux, otherwise (true) dual port RAM's achieve the CDC between tracking and read-out

- All inputs are registered, timing closure achieved at up to 434 MHz (on Xilinx MPSoC)
- Most critical (not in this example): count buffer due to read _and_ write on trkclk



WhizniumDBE Module Templates

 WhizniumDBE module templates used across the presented examples (cf. <u>https://github.com/mpsitech/wted-WhizniumDBE-Template-Evaluation-Device</u>)

Module template	Features
Axislave_v1_0 *)	AXIlite to AXI stream, var. width
Crcspec_v3_0	16-/32-bit poly CRC with var. width AXI stream
Ddrmux_Easy_v1_0	Multiplexed DDR memory access with burst transfers across subset of AXI4, transfer statistics accumulation with CPU read-out CROSS-VENDOR PHY-AGNOSTIC
Debounce_v1_0	Digital pushbutton debouncer
Dpram_efnx_v1_0	Dual-port RAM generator for Efinix with asymmetric port widths
Dpebram_lttc_v1_2 *)	Dual-port EBRAM wrapper for Lattice
Dpbram_xlnx_v8_4	Dual-port BRAM wrapper for Xilinx
Fsmtrack_Easy_v1_0	FSM tracker with trigger and capture manifolds, compression and statistics accumulation, CPU read-out CROSS-VENDOR



WhizniumDBE Module Templates

Module template	Features (cont'd)
Gptrack_Easy_v1_0	15-bit general purpose tracker with trigger manifold and compression, CPU read-out CROSS-VENDOR
Hostif_Easy_v1_0	Connection to CPU host with byte protocol decoding PHY-AGNOSTIC
Ident_Easy_v1_0	Block / IP identification including version and Git hash with CPU read-out
Rgbled_v1_0	Driver for RGB888 in to PWM LED out
Timeout_v1_0 *)	Watchdog timer
Tkclksrc_Easy_v1_0	10 kHz clock source for time-stamping and timeout tasks with CPU read-out
Top_v1_0	Clock and reset generation, top-level routing CROSS-VENDOR
Uartrx/tx_v2_0 *)	UART RX/TX to AXI stream

 Plus, an awesome 3rd party core: neoTRNG for random number generation (BSD3 license) https://github.com/stnolting/neoTRNG

*) auto - instantiated



WhizniumDBE ("Device Builder's Edition") is

- NOT high-level synthesis (HLS), not a compiler
- NOT your typical generator framework
- NOT a visual design tool



WhizniumDBE ("Device Builder's Edition") is

- NOT WhizniumDBE is
- NOT ... a user-extensible framework written in C++, that for a given RTL design
- NOT ... interprets its structure and features, specified in text-based model files
 - ... composes and maintains a fine-grained RTL model *) in a SQL database
 - ... then is able to write VHDL and C++ code based on it
 - ... taking into account manual code contributions of previous design versions

*) from hierarchical structure down to FSM's incl. state transitions, CDC fabric, generics/ports/signals/variables



- Clean, ergonomic, source code structure ("tasteful naming conventions", etc.)
- Parametrized templates for standard components (e.g. SPI, GPIO, CRC, Git-Ident; 35 and counting)
- Custom templates can interact with the model / module surroundings while a design is composed (not just simple files with placeholders)
- The applicable vendor('s primitives) can be an auto-derived template parameter
- Scope extends beyond the FPGA world with **WhizniumSBE** (Service Builder's Edition)





• Regular RTL workflow (including use of vendor IDE's) augmented by "source code tree iteration"



Vendor-agnostic probing of FPGA designs For efficient real-time debugging



• Modular description (IexWdbeMdl.txt), CRC template, polynomial changed



	Start Einfügen	Zeichnen Seitenlayout Fo	rmeln Daten Überprüfen	Ansicht Automatisieren			C Kommer	ntare 🕼 Freigeben 🗸
ncept Key	Einfügen	alibri \checkmark 12 \land K \bigcup \checkmark \bigtriangleup \checkmark	A [×] = = = ₽ ↓ × = = = ⊡ × = = ≥ ≫ × × × ×	Standard ✓ Image: Standard ✓	Bedingte Formatierung V Als Tabelle formatieren V Zellenformatvorlagen V	Einfügen v E Löschen v E Format v	D v Bearbeiten	Add-Ins Daten analysieren
	A249 🛔 🗙 🗸	/ fx						
	A	В	С	D	E			F
Modular dosc	221		trigs	reqInvFiniterxRelease;reqInv	/FinitetxSend			
would ucst	222		sizeSeqbuf	4				
	223		ImelAMModulePar.end					
	224	ImelMModule.end						
	225 fpga	xczu6cg-1ffvc900e	mplet	Ethernet frame to FSO pack	et true	vivado		
	226	ImelMModule.srefIxVBasetype	hsrefSupRefWdbeMModule	srefTplRefWdbeMModule	sref	Comment		
	227	top		top_v1_0	mplet_core			
	228		ImeIAMModulePar.x1SrefKKey	Val				
	229		aresetNNotP	true				
	230		fAclk	200000				
	231		aclkDiffNotSng	false				
	232		clks	mclk;ethrxclk;ethtxclk;fsorxc	clk;fsotxclk			
	233		fClks	200000;125000;125000;15	56250;156250			
	234		clkIntNotExts	true;false;false;true;true				
	235		clkDiffNotSngs	false;false;false;false;false				
	236		ImeIAMModulePar.end					
	237	ectr	mplet_core	ident_Easy_v1_0	ident	version identif	ier	
	238		ImeIAMModulePar.x1SrefKKey	Val				
	239		hash	c20dcef				
	240		who	aw.mpsi				
	241		cfg	fMclk.nat.uint32;fEthrxclk.n	at.uint32;fFsorxclk.nat.	uint32;RCli.nat.uir	nt8;RFso.nat.uin	t8
	242		ImeIAMModulePar.end					
	243	ehostif	mplet_core	hostif_Easy_v1_0	hostif	APU via AXI H	PM0 FPD	
	244		ImeIAMModulePar.x1SrefKKey	Val				
	245		phytype	axi				
	246		wA	40				
	247		wD	64				
	248		ImeIAMModulePar.end					
	249	ectr	mplet_core		client	Ethernet inter	face	
	250	oth	mplet_core;client	crcspec_v3_0	ethcrc	calculate		
	251		ImelAMModulePar.x1SrefKKey	Val				
	252		poly	12345678				
	253		wD	64				
	254		ImeIAMModulePar.end					
	255	oth	mplet_core;client		lenrand	random numb	er generator for	r frame size
	256	oth	mplet_core;client;lenrand	neotrng_v1_0	neotrng			
	257	ectr	mplet_core	ddrmux_Easy_v1_0	ddrif	DDR AXI trans	fer arbiter via A)	KI HPO FPD
	258		ImelAMModulePar.x1SrefKKey	Val				
	-							
	IexWdbell	Adl fstd +						

Vendor-agnostic probing of FPGA designs For efficient real-time debugging



•••		\leftrightarrow \rightarrow \bigcirc		tsbereich)		
Ch	EXPLORER		nd ↔ Client.vhd fstd_0_2_5 • fpgafstd/ fpg	gafstd/mplet ×		$\cdots \square \oslash \mathbb{P} \leftrightarrows \rightarrow \lor \land \land$
U	∨ UNBENANNT (ARBEITSBEREICH)	fstd > fpgafstd > mplet > \equiv C	Client.vhd			
Q	\sim fstd	85 architecture Rtl of Clien	t is	90	architecture Rtl of Client is	
/	> _mdl	251 252 basis		257	han in	
የօ	> _rls	252 begin 253		258	begin	
6	> ezdevfstd	254		260		
~	✓ fpgafstd	255 sub-module instantia	tion	261	sub-module instantiation	
	> iftest	257		263		
04.080329) ifteet m1	258 myEthcrc : Crc04C11DB7	.64	264+	myEthcrc : Crc12345678_64	
	> intest_init	259 port map (265	port map (
⊑⊙	> mparq	260 reset => resetEthrx	clk,	266	reset => resetEthrxclk,	
	> mparq_m1	261 mclk => ethrxclk, 262		267	mclk => ethrxclk,	
μ ^O	✓ mplet	263 AXIS_tready => ethc	rcAXIS_tready,	269	AXIS_tready => ethcrcAXIS_tready,	
ய		264 AXIS_tvalid => ethc	rcAXIS_tvalid,	270	AXIS_tvalid => ethcrcAXIS_tvalid,	
	= Axisiave_v1_0.vnd	265 AXIS_tdata => ethcr	cAXIS_tdata,	271	AXIS_tdata => ethcrcAXIS_tdata,	
0	Client.vhd	266 AXIS_tkeep => ethcr	cAXIS_tkeep,	272	AXIS_tkeep => ethcrcAXIS_tkeep,	
	E Crc04C11DB7_64.vhd	267 AXIS_tlast => ethcr	cAXIS_tlast,	273	AXIS_tlast => ethcrcAXIS_tlast,	
	\equiv Crc1021 64 vbd	268 269 crc => etherco.		274	crc => etherco	
		270 validCrc => validEt	hcrco	275	validCrc => validEthcrco	
	= Crc1021_128.vhd	271);		277);	
		272		278		
	E Crc12345678 64.vhd	273 myLenrand : Lenrand		279	myLenrand : Lenrand	
	= Ddrifyhd	274 generic map (280	generic map (
	= Dam.viia	275 w => 14,		281	w => 14,	
	Decode.vhd	270		282	low -> 46	
	Encode.vhd	278 high => 10184.		285	high => 10184 .	
	= Fec.vhd	279		285		
	= Oth und	280 NBit => 14, dete	rmined by high-low	286	NBit => 14, determined by high-low	
	= Gtn.vna	281 NByte => 2 ceil(NBit/8)	287	NByte => 2 ceil(NBit/8)	
	Hostif.vhd	282)		288)	
Q	≡ Ident.vhd	283 port map (289	port map (
0	= Latency.vhd	285 mclk => mclk		290	mclk => mclk.	
070		286		292	me ex -> me exy	
203		287 strbRand => strbLen	rando,	293	<pre>strbRand => strbLenrando,</pre>	
	> ZEITACHSE	288 rand => lenrando		294	rand => lenrando	

Vendor-agnostic probing of FPGA designs For efficient real-time debugging



FPGA Conference Europe 2024

				*-1->				
•••			\leftarrow \rightarrow \bigcirc Unbenannt (Arbeitsberg	eich)				
Ŋ	EXPLORER	Client.vhd	\equiv Client.vhd \leftrightarrow Client.vhd fstd_0_2_5 · fpgafstd/ fpgafstd/mpl	et	Cro	:040	C11DB7_64.vhd ↔ Crc12345678_64.vhd × ↑	$\downarrow \rightleftharpoons \P \bowtie \square \cdots$
	V UNBENANNT (ARBEITSBEREICH)	fstd > fpgafs	std > mplet > \equiv Crc12345678_64.vhd					1.0
Q	∨ fstd	45 begin		45	beg	in		
	> _mdl	56 proces 57 vari	s (reset, mclk, stateOp) able first: boolean;	56	5 p	roce var	ess (reset, mclk, stateOp) riable first: boolean;	_
J.S	> ezdevfstd	58 50 borin		58	3	ogi-		
	✓ fpgafstd	60 if r	reset='1' then	60		if	reset='1' then	
Ŕ	> iftest	62	accop <= stateopiliit;	62	2	S	cateop ~ stateopinit;	
	> iftest_m1	63 cr	<pre>cc_sig <= (others => '0');</pre>	63	3	c	<pre>rrc_sig <= (others => '0');</pre>	
ΓÖ	> mparq	65 Va	<pre>statuc_sig <= '0';</pre>	64	5	v	allourc_sig <= '0';	
	> mparq_m1	66 fi	rst := true;	66	5	f	irst := true;	
B	✓ mplet	68 elsi	f rising_edge(mclk) then	68	3	els	if rising_edge(mclk) then	
	Axislave_v1_0.vhd	69 if	stateOp=stateOpInit then	69	9	i	f state0p=state0pInit then	
T		70 71	<pre>crc_sig <= (others => '0'); validCrc sig <= '0';</pre>	70			<pre>crc_sig <= (others => '0'); validCrc sig <= '0';</pre>	
	E Crc04C11DB7_64.vhd	72		72	2			
		73 74	<pre>first := true;</pre>	73	1		first := true;	
	E Crc1021_128.vhd	75	<pre>stateOp <= stateOpCapt;</pre>	75	5		<pre>state0p <= state0pCapt;</pre>	
		76	sif stateOn=stateOnCant then	76	5		lsif stateOn=stateOnCant then	
		78	if AXIS_tvalid='1' and (first or validCrc_sig='1') then	78	3		if AXIS_tvalid='1' and (first or validCrc_sig=':	1') then
	≣ Ddrif.vhd	79	case AXIS_tkeep is	79			case AXIS_tkeep is	
	≣ Decode.vhd	80	<pre>crc_sig(31) <= AXIS_tdata(7) xor AXIS_tdata(5) xor AXIS tda</pre>	80	, L+		<pre>wmen fillilli => crc_sig(31) <= AXIS_tdata(7) xor AXIS tdat</pre>	ta(6) xor AXIS_tdata(4)
	Encode.vhd	82-	<pre>crc_sig(30) <= AXIS_tdata(6) xor AXIS_tdata(4) xor AXIS_tdata(4)</pre>	82	2+		<pre>crc_sig(30) <= AXIS_tdata(6) xor AXIS_tdata</pre>	ta(5) xor AXIS_tdata(3)
	≣ Fec.vhd	83— 84—	<pre>crc_sig(29) <= AXIS_tdata(5) xor AXIS_tdata(3) xor AXIS_tdata(3) xor AXIS_tdata(2) xor AXIS_tdata(4) xor AXIS_tdata(4) xor AXIS_tdata(5) xor AXIS_tdata</pre>	83	3+ 1+		<pre>crc_sig(29) <= AXIS_tdata(7) xor AXIS_tdat crc sig(28) <= AXIS tdata(6) xor AXIS tdat</pre>	ta(5) xor AXIS_tdata(4) ta(4) xor AXIS tdata(3)
	≣ Gth.vhd	85-	<pre>crc_sig(27) <= AXIS_tdata(6) xor AXIS_tdata(3) xor AXIS_tdata</pre>	85	5+		<pre>crc_sig(27) <= AXIS_tdata(6) xor AXIS_tdata</pre>	ta(5) xor AXIS_tdata(4)
		86— 97—	$crc_sig(26) = AXIS_tdata(5) xor AXIS_tdata(2) xor AXIS_tdata(2) xor AXIS_tdata(2) xor AXIS_tdata(5) xor AXIS_tdata(5)$	86	5+ 7+		crc_sig(26) <= AXIS_tdata(7) xor AXIS_tdat	ta(5) xor AXIS_tdata(4)
Q	≣ Ident.vhd	88-	<pre>crc_sig(24) <= AXIS_tdata(7) xor AXIS_tdata(5) xor AXIS_tdata(6) xor AXIS_tdata(7) xor AXIS_tdata</pre>	88	8+		crc_sig(24) <= AXIS_tdata(7) xor AXIS_tdata(7) x	ta(5) xor AXIS_tdata(4)
V	≣ Latency.vhd	89—	<pre>crc_sig(23) <= AXIS_tdata(6) xor AXIS_tdata(5) xor AXIS_tdata</pre>	89	+		<pre>crc_sig(23) <= AXIS_tdata(7) xor AXIS_tdat</pre>	ta(6) xor AXIS_tdata(4)
st	> GLIEDERUNG	90-	<pre>crc_sig(22) <= AXIS_tdata(5) xor AXIS_tdata(4) xor AXIS_tdata(5) xor AXIS_tdata(4) xor AXIS_tdata(5) xor AXIS tdata(4) xor AXIS tdata(5) xor AXIS tdata</pre>	90	ν+ ι+		<pre>crc_sig(22) <= AXIS_tdata(b) xor AXIS_tdat crc sig(21) <= AXIS tdata(7) xor AXIS tdat</pre>	ta(5) xor AXIS_tdata(3) ta(5) xor AXIS tdata(4)
203	> ZEITACHSE	92-	<pre>crc_sig(20) <= AXIS_tdata(7) xor AXIS_tdata(6) xor AXIS_tdata</pre>	92	2+		<pre>crc_sig(20) <= AXIS_tdata(0) xor AXIS_tdata</pre>	ta(14) xor AXIS_tdata(1)
		03	crc sig(10) ~ AYIS tdata(6) yor AYIS tdata(5) yor AYIS tda	03	2		crc sig(10) <- AVIS tdata(7) yor AVIS tdat	ta(6) vor AVIS tdata(4)

Vehicol-agnostic probing of FFOA designs



• Command set and buffer transfers (lexWdbeCsx.txt), command added to module "Client"



ncept Key f	Einfügen	Calibri F K U	 12 → A[*] A[*] 12 → A[*] A[*] 12 → A[*] A[*] 12 → A[*] A[*] 	= = = ₽ × = = = ⊡ × = = ≫ ×	Standard ✓ ₩ ↓ 000 × % 9 500 -300	Bedingte Formatierung Als Tabelle formatieren Zellenformatvorlagen v	 ✓ I Einfüg ✓ I Einfüg ✓ Ei	gen v len v at v Beart) v Booten Add-	Ins Daten analysiere	'n	
	A255	× √ fx	-	-		-	-				*	
	A	В	LmolMV(actoritom1 crof	D	E	F	G	н	4	J	к .	
Command set	232		nc	offline	connent							len
	233		ready	ready								
	234		active	transceiving								
	235		ImelMVectoritem1.end									
	236	ImelMVecto	1.end									
	237	ImelMModu	sref									
	238	mplet core	client									
	239		ImelMController.									
	240		٨									
	241			ImelMCom	n sref	sreflxVRettype	sreflyrRefWo	srefRvrRefW	srefRerRefW	Comment		
	242			0	configTrafgen	void						
	243				ImeIAMCommandInvpar2.sref	sreflxWdbeVPartype	srefRefWdbe	Length	Defval	srefRefWdbe	Comm	
	244				randNotFix	_bool					randor	
	245				expTRep	uint8					in mclk	
	246				NPayload	uint16					in byte	
	247				destH	uint16					MAC d	
	248				destL	uint32						
	249				ImeIAMCommandInvpar2.end							
	250			0	set	void						
	251				ImeIAMCommandInvpar2.sref	sreflxWdbeVPartype	srefRefWdbe	Length	Defval	srefRefWdbe	Comm	
	252				rng	_bool						
	253				trafgenNotEth	_bool						
	254				ImeIAMCommandInvpar2.end							
	255			0	doSomething	void						
	256				ImeIAMCommandInvpar2.sref	srefIxWdbeVPartype	srefRefWdbe	Length	Defval	srefRefWdbe	Comm	
	257				what	uint8						
	258				ImeIAMCommandInvpar2.end							
	259			ImelMComr	nand2.end							
	260		ImelMController.end									
	261	mplet_core	ddrif									
	262	mplet_core	fec									
	263		ImelMController.									
	264		^									
	265			ImelMComr	n sref	sreflxVRettype	sreflvrRefWo	srefRvrRefW	srefRerRefW	Comment		
	266			0	config	void						
	267				ImelAMCommandInvpar2.sref	sreflxWdbeVPartype	srefRefWdbe	Length	Defval	srefRefWdbe	Comm	
	268				strength	uint8					in net v	

Vendor-agnostic probing of FPGA designs For efficient real-time debugging



EXPLORER ···· VINBENANNT (ARBEITSBEREICH) fstd >_mdl	<i>⊂ ⊂ Client.vhd</i> fstd > fpgafstd > n 15 entity Client 16 port (- → ,	eitsbereich) ogafstd/mplet	×	 ↑↓ =	
EXPLORER VUNBENANNT (ARBEITSBEREICH) () fstd > _mdl	<pre>E Client.vhd fstd > fpgafstd > n 15 entity Client 16 port (</pre>	E Client.vhd ↔ Client.vhd fstd_0_2_5 • fpgafstd/ f nplet > E Client.vhd	ogafstd/mplet	×		
✓ UNBENANNT (ARBEITSBEREICH) ✓ fstd > _mdl	<pre>fstd > fpgafstd > n 15 entity Client 16 port (</pre>	nplet > Client.vhd				- I W U
O ∨ fstd > _mdl	15 entity Client					
>_mdl	16 port (is		15 er	ntity Client is	
	20 P			16	port (
	42 reqInvCon	<pre>figTrafgen: in std_logic;</pre>		42	<pre>reqInvConfigTrafgen: in std_logic;</pre>	
∠ > _ris	43 ackInvCon	<pre>figTrafgen: out std_logic;</pre>		43	<pre>ackInvConfigTrafgen: out std_logic;</pre>	
> ezdevfstd	44 45	feerDeedNetFing is and logic method (7 domate 0).		44	andi-Technology Matrix, is and logic vertex(7 doubte 0).	
✓ fpgafstd	45 configTra	fgenExnTRep: in std logic vector(7 downto 0);		45	configTrafgenExpTRep: in std logic_vector(7 downto 0);	
> iftest	47 configTra	fgenNPavload: in std logic vector(15 downto 0);		47	configTrafgenNPavload: in std logic vector(15 downto 0);	
hiftoot m1	48 configTra	<pre>fgenDestH: in std_logic_vector(15 downto 0);</pre>		48	<pre>configTrafgenDestH: in std_logic_vector(15 downto 0);</pre>	
/ intest_init	49 configTra	<pre>fgenDestL: in std_logic_vector(31 downto 0);</pre>		49	<pre>configTrafgenDestL: in std_logic_vector(31 downto 0);</pre>	
≤S > mparq	50			50		
> mparq_m1	51 reqInvSet	: in std_logic;		51	reqInvSet: in std_logic;	
P ∨ mplet	52 ackinvset	: but sta_togic;		53	ackinvset: out std_togic;	
E Axislave_v1_0.vhd	54 setRng: i	n std_logic_vector(7 downto 0);		54	<pre>setRng: in std_logic_vector(7 downto 0);</pre>	
Section Client.vhd	55 settrarge	nNotetn: in sta_togic_vector(/ downto 0);		50 56+	setTrangenNoteth: in std_togic_vector(/ downto 0);	
= Crc04C11DB7 64 vbd				57+	<pre>reqInvDoSomething: in std_logic;</pre>	
			\rightarrow	58+	<pre>ackInvDoSomething: out std_logic;</pre>	
= Crc1021_64.vnd				59+		
	EC	///////////////////////////////////////		60+	<pre>doSomethingWhat: in std_logic_vector(7 downto 0);</pre>	
	57 fife tx r	data rdv: out std logic:		62	fife tx r data rdy: out std logic:	
E Crc12345678 64.vhd	58 fifo_tx_r	rd: in std_logic;		63	<pre>fifo_tx_r_rd: in std_logic;</pre>	
	59 fifo_tx_r	_valid: out std_logic;		64	<pre>fifo_tx_r_valid: out std_logic;</pre>	
⇒ Darit.vna	60 fifo_tx_r	<pre>_data: out std_logic_vector(7 downto 0);</pre>		65	<pre>fifo_tx_r_data: out std_logic_vector(7 downto 0);</pre>	
Decode.vhd	61 fifo_tx_r	_sop: out std_logic;		66	<pre>fifo_tx_r_sop: out std_logic;</pre>	
≣ Encode.vhd	62 fife ty r	_eop: out std_logic;		67	<pre>tito_tx_r_eop: out std_logic; fife tx_r_err; out std_logic;</pre>	
= Fec vhd	64 fife ty r	underflow: out std logic:		69	fife tx r underflow: out std logic:	
	65 fifo tx r	flushed: out std_logic;		70	<pre>fifo_tx_r_flushed: out std_logic;</pre>	
= Gtn.vna	66 fifo_tx_r	_control: out std_logic;		71	<pre>fifo_tx_r_control: out std_logic;</pre>	
	67			72		
≣ Ident.vhd	68 fifo_dma_	<pre>tx_end_tog: in std_logic;</pre>		73	<pre>fifo_dma_tx_end_tog: in std_logic;</pre>	
= Latency ybd	69 fifo_dma_	<pre>tx_status_tog: out std_logic;</pre>		74	<pre>fifo_dma_tx_status_tog: out std_logic; fifo_tx_r_status_in_std_logic vostor(2_downto_0);</pre>	
	70 1110_1X_1	_status: in stu_togic_vector(3 downto 0);		76	TITO_LX_I_SCALUS; IN SCU_LOGIC_VECTOR(3 downto 0);	
	72 fifo rx w	wr: in std logic;		77	fifo_rx_w_wr: in std_logic;	
> ZEITACHSE	73 fifo rx w	data: in std logic vector(7 downto 0):		78	fifo rx w data: in std logic vector(7 downto 0):	

Vendor-agnostic probing of FPGA designs For efficient real-time debugging



FPGA Conference Europe 2024

• Custom fine structure (IexWdbeFin.txt), FSM state added



			un ociterinaye			Redingte Formatierung v			
oncept Ke		Calibri	~ 12		Standard	Als Tabelle formatieren v	Löschen v V Sastiana Supposed		
	Einfugen 💞	F K U	• H• Ø	× ▲ ×	»¶ • 🕮 • %	7 500 300 IV Zellenformatvorlagen V	Format v Sortieren Suchen und	analysi	ieren
	A2280 🔹 🗙	√ fx							
Custom fin	A	В	С	D	E	F	G	н	1
	2252			ImelMEsm.end			//		
	2253		trafgen	ethrxclk	resetEthrxclk	false	state(init) or !txrng_sig_ethrxclk or !tra	afgentalse	traffic
	2254			ImelMFsm.srefixvDbgtapty	pe				
	2255			void	In the second state in the	D and	E alla	C	
	2256				Imelivirsmstate.irerke	an srer	Extip	Comment	
	2257				U	Init	taise	le1	Conda
	2258					ImerAlvirsmstateStep.sretFnxkerWdbeMFsms	turna sia ethnuelle extrafa 1 1 1 1 1 1 1 -	ip1	Cond2
	2259					init Idla	itxrng_sig_etnrxcik or tratgenNotetn_	athrxcik	
	2200					Intel AME and the Chan and	else		
	2201				0	imeiAMFsmstateStep.end	falsa		
	2202				U	Idle	tate Cond1	le1	Cond2
	2203					ImelAlvirsmstatestep.srerrnxkerwabewirsms	Itrafgen DandNietEis er stehl en rande	Ipi	Condz
	2264					dataA	ItrafgenkandNotFix or strbLenrando	start	
	2205				1	ImerAMFSmstateStep.end	falsa		
	2200				1		Taise	le 1	Cando
	2207					ImerAlvirsmstatestep.srerFnxkerwdbelviFsms	state Cond1	Ipi	Condz
	2268					datab		dataA	
	2269				1	ImelAWFsmstatestep.end	false		
	2270				1	Catab	Taise	Int	Canda
	22/1					ImelAMFsmstatestep.srefFnxkerWdbeMFsms	state Cond1	Ip1	Cond2
	2272					datac		datab	
	22/3					ImeiAMFsmstatestep.end	6-1		
	2274				1	datac	Taise	-	C
	2275					ImeiAiviFsmstatestep.srefFnxkerwdbewiFsms	state Cond1	ip1	Cond2
	2276					pause	client losourceAXIS_tready		wordi
	2277					datac	client I osourceAXIS_tready		ethcrcA
	2278					dataA	client losourceAxIs_tready		eise
	2279				0	ImelAWFsmstatestep.end	6-1	_	_
	2280				0	pause	Taise		0 10
	2281					ImerAlvirsmstatestep.srefrnxkerwabewirsms	state Condi	ipi	Cond2
	2282					Init	pause(trargenExpTRep)=1		
	2283					арог	problemOccurred		
	2284					pause	else	inc	
	2285				0	imeiAivirsmstateStep.end	6.1		
	2286				0	abort	taise		
	2287				ImelMFsmstate.end				
	2288			ImeIMFsm.end					

Vendor-agnostic probing of FPGA designs For efficient real-time debugging



•••	$\leftarrow \rightarrow$	🔎 Unbenannt (Arbeitsbereicl	h)		
EXPLORER		Client.vhd fstd_0_2_5 • fpgafstd/ fpgafstd/m	plet X		$\square \And \mathbb{P} \leftrightarrows \downarrow \checkmark \blacksquare$
→ UNBENANNT (ARBEITSBEREICH)	fstd > fpgafstd > mplet > = Client.v	vhd			
○ ✓ fstd	349 setRng: in std_logi	ic_vector(7 downto 0);	355 -	setRng: in std_logic_vector(7 down	to 0);
> mdl	544 begin		550	begin	_
	715 elsif stateTrafgen=stateT	TrafgenDataC then	569	elsif stateTrafgen=stateTrafgenDataC the	1
	716 if clientToSourceAXIS_t	tready='1' then	721	if clientToSourceAXIS_tready='1' then	
> ezdevfstd	749 wordTrafgen := word	dTrafgen + 1; IP impl.trafgen.dataC.ne>	755	<pre>wordTrafgen := wordTrafgen + 1;</pre>	<pre>IP impl.trafgen.dataC.nextWord</pre>
✓ fpgafstd			756		
C > iftest	751 staterrargen <= sta 752 end if:	aterrargendataa;	758	end if:	
> iftest m1	753 end if:		759	end if;	
	754		760		
- mparq	755 elsif stateTrafgen=stateT	TrafgenPause then	761	elsif stateTrafgen=stateTrafgenPause ther	1
> mparq_m1	756 if pause(trafgenExpTRep 757 stateTrafgen <= state	p)='1' then	762	<pre>if pause(trafgenExpTRep)='1' then stateTrafgen <= stateTrafgenInit;</pre>	
└ v mplet	758	errargeninit,	764	staterrargen <= staterrargeninit,	
Axislave_v1_0.vhd		///////////////////////////////////////	765+	elsif problemOccurred then	
E Client vhd			766+	<pre>stateTrafgen <= stateTrafgenAbort;</pre>	
		///////////////////////////////////////	767+		
E Crc04CTDB7_64.vnd	759 else	ector(unsigned(nause) + 1); TP impl tr:	768	else	ause) + 1) TP impl trafgen
	761	cecor(unsigned(pudse) + 1); = if imperent	770	pause .= sta_togic_vector(unsigned(pa	use, i i, ii inpereturgen.
	762 stateTrafgen <= state	eTrafgenPause;	771	<pre>stateTrafgen <= stateTrafgenPause;</pre>	
≣ Crc8005_64.vhd	763 end if;		772	end if;	
- Crc12345678 64 ybd		///////////////////////////////////////	773+		
			775+	IP impl_trafgen_abort INSERT	1
= Ddrif.vhd	764 end if;		776	end if;	
Decode.vhd	765 end if;		777	end if;	
≡ Encode.vhd	766 end process;		778	end process;	
≡ Fec.vhd	767 IP impl.trafgen.rising	- END	779	— IP impl.trafgen.rising —— END	
= Oth yhd	769		781		
= Gui.viu	770 implementation: TX data st	treaming (txstr)	782	implementation: TX data streaming (txstr)	
= Hostif.vhd	771		783		
8 ≡ Ident.vhd	772		784		
	<pre>//3 IP impl.txstr.wiring F 774</pre>	KREGTN	785	IP impl.txstr.wiring RBEGIN	
ST2 > GLIEDERUNG	775 takes data from GEM (8 bit	t and re-packages it to 64 bit, adjusting	787	takes data from GEM (8 bit and re-packages	s it to 64 bit, adjusting tkee
205 XEITACHSE	776	· · · · · · · · · · · · · · · · · · ·	788		

Vendor-agnostic probing of FPGA designs For efficient real-time debugging



FPGA Conference Europe 2024

• WhizniumDBE Iterator (cross-platform Java tool accessing the WhizniumDBE daemon via it JSON API)



• WhizniumDBE Iterator (cross-platform Java tool accessing the WhizniumDBE daemon via it JSON



Full Video: https://content.mpsitech.cloud/FPGAEurope2024/WhizniumDBE_step.mp4

Vendor-agnostic probing of FPGA designs For efficient real-time debugging

API)



Conclusion & Outlook

- PHY- and vendor-agnostic probing of FPGA designs using Open Source software is feasible
- ... and long-overdue to replace proprietary solutions
- Mid-range (and larger) devices have sufficient capacity to implement this functionality
- TBD 1: integration of read-out and .vcd generation into the Open Source **WhizniumDBE** core library; also support merging results across clock domains
- TBD 2: addition of a **WhizniumSBE** capability to control the probing blocks from an (auto-generated) web UI

• Not on the roadmap: the "O" part of "Virtual I/O". While probing is universal, stimuli aren't ... and it is easy to design a project-specific, CPU-connected module using **WhizniumDBE**



Resources

- Both Whiznium tools are available free of charge on GitHub, including installation instructions
 <u>https://github.com/mpsitech/The-Whiznium-Documentation</u>
- The Open Source StarterKit ist available for various hardware platforms, with vendor-specific instructions also available on GitHub
- "The Whiznium Developer Experience" webinar series on Whiznium is on YouTube (google it)
- For advanced users WhizniumSBE/DBE cheat sheets are available which serve as reference for writing model files
 WhizniumDBE Cheat Sheet





Resources

- Both Whiznium tools are available free of charge on GitHub, including installation instructions
 <u>https://github.com/mpsitech/The-Whiznium-Documentation</u>
- The Open Source StarterKit ist available for various hardware platforms, with vendor-specific instructions also available on GitHub
- "The Whiznium Developer Experience" webinar series on Whiznium is on YouTube (google it)
- For advanced users WhizniumSBE/DBE cheat sheets are available which serve as reference for writing model files
- (New) home for cross-vendor cores
 <u>https://mpsitech.github.io/Laser-Scanner-By-Platform</u>
- Some more presentations on the topic
 <u>https://www.mpsitech.com/documentation/presentations</u>



Thank You! Questions?

Also, feel free to connect.

- <u>https://www.linkedin.com/in/wirthmua</u>
- <u>https://github.com/mpsitech</u>

Alexander Wirthmüller Founder & Director

Phone: +49 (89) 4524 3826 Mobile: +49 (175) 918 5480 E-Mail: aw@mpsitech.com

MPSI Technologies GmbH Agnes-Pockels-Bogen 1 80992 Munich Germany www.mpsitech.com



Gptrack / Fsmtrack model entry

lexWdbeMd	lv1.1.28				
ImelMUnit.s	srefIxVB srefSilRefWdbeMUnit	sref	Title	Easy	srefKToolch
fpga	lifcl-40-9bg400c	cleb	Lattice CrossLink-NX Evalua	tictrue	radiant
	ImelMModule.srefIxVBasetype	hsrefSupRefWdbeMModule	srefTplRefWdbeMModule	sref	Comment
	edbgctr	wted_core	fsmtrack_Easy_v1_0	mfsmtrack0	track host interface FSM
		ImeIAMModulePar.x1SrefKKey	Val		
		trkclk	mclk		
		ratioTrkclk	1		
		capts	hostif.op		
		trigs	hostifRxAXIS_tvalid.rising;ac	kInvTkclksrcSetTkst.rising	
		sizeSeqbuf	4		
		ImelAMModulePar.end			
	edbgctr	wted_core	fsmtrack_Easy_v1_0	mfsmtrack1	track other mclk-based FSM's
		ImelAMModulePar.x1SrefKKey	Val		
		trkclk	mclk		
		ratioTrkclk	1		
		capts	tkclksrc.op		
		trigs	hostifRxAXIS_tvalid.rising;ac	kInvTkclksrcSetTkst.rising	
		sizeSeqbuf	4		
		ImelAMModulePar.end			
	edbgctr	wted_core	gptrack_Easy_v1_0	mgptrack	track mclk-based signals
		ImelAMModulePar.x1SrefKKey	Val		
		trkclk	mclk		
		ratioTrkclk	1		
		capts	tkclk;rgb0_r;rgb0_g;rgb0_b;	btn0;btn0_sig;tkclksrcGet	TkstTkst[70]
		trigs	btn0.rising;btn0.falling;ackl	nvTkclksrcSetTkst.rising	
		sizeSeqbuf	4		



Hostif model entry

ImeIMUnit.srefIxVI	srefSilRefWdbeMUnit	sref	Title	Easy	srefKToolch
fpga	lifcl-40-9bg400c	cleb	Lattice CrossLink-NX Evaluati	true	radiant
	ImelMModule.srefIxVBasetype	hsrefSupRefWdbeMModule	srefTplRefWdbeMModule	sref	Comment
	ehostif	wted_core	hostif_Easy_v1_0	hostif	connection to host via FT2232H
		ImeIAMModulePar.x1SrefKKey	Val		
		phytype	uart		
		fSclk	115200		
		ImeIAMModulePar.end			

fpga	ti180m484i3	tidk	Efinix Titanium Ti180 devel	o true	efinity
	ImelMModule.srefIxVBasetype	hsrefSupRefWdbeMModule	srefTplRefWdbeMModule	sref	Comment
	ehostif	wted_core	hostif_Easy_v1_0	hostif	connection to RISC-V soft-cores running Linux via AXIIite
		ImeIAMModulePar.x1SrefKKey	Val		
		phytype	axi		
		wA	32		
		wD	32		
		ImeIAMModulePar.end			

fpga	xczu1cg-1sbva484e	zudk	Avnet Zynq UltraScale+ MPSo	true	vivado
	ImelMModule.srefIxVBasetype	hsrefSupRefWdbeMModule	srefTplRefWdbeMModule	sref	Comment
	ehostif	wted_core	hostif_Easy_v1_0	hostif	connection to Cortex-A53's running Linux via FPD AXIlite
		ImeIAMModulePar.x1SrefKKey	Val		
		phytype	axi		
		wA	40		
		wD	64		
		ImeIAMModulePar.end			



Efinix vs. Xilinx Fsmtrack

•	\leftarrow \rightarrow (♀ wted
EXPLORER	≡ Mfsmtrack0.vhd \leftrightarrow Mfsmtrack0.vhd/tidk/zudk ×	$\Box \Leftrightarrow \mathbb{P} \leftrightarrows \lor \land \Box$
\sim wted	fpgawted > zudk > ≡ Mfsmtrack0.vhd	
>< fpgawted	313 begin	331 begin
* ipgawied	315	333
√ tidk	316 sub-module instantiation	334 sub-module instantiation
Neotrng_v1_0.vhd	317	335
≣ Rgbled_v1_0.vhd	318	336
State vbd	319- myCntbuf : Dpram_size2kB_a32b32	337+ myCntbuf : Dpbram_size2kB_a32b32
	320 port map (338 port map (
IIdk.ist	321 resetA => reset,	
Tidk.vhd	322 CIKA => IICIK,	340
≡ Timeout v1 0.vhd	324 enA => enCntbuf.	341 enA => enCntbuf.
Theline Ferry of Outed	325- weA => '1',	
= TKCIKSFC_Easy_VI_0.Vhd	326	343
Trafgen.vhd	<pre>327— aA => aCntbuf_vec,</pre>	344+ addrA => aCntbuf_vec,
Wted core.vhd	328- drdA => open,	doutA => open,
N ZUCK	329- dwrA => dwrCntbuf,	346+ dinA => dwrCntbuf,
~ ZUUK	330 221 recetB => recet	34/
Axislave_v1_0.vhd	$\frac{332}{clkB} \Rightarrow mclk.$	348 clkB => mclk.
Elient.vhd	333	349
\equiv Crc8005 64 vbd	<pre>334 enB => enCntbufB,</pre>	350 enB => enCntbufB,
	335- weB => '0',	351+ weB => (others => '0'),
Ddrif.vhd	336	352
Debounce_v1_0.vhd	<pre>337- aB => aCntbufB_vec,</pre>	353+ addrB => aCntbufB_vec,
⊟ Hostif.vhd	338- drdB => drdCntbufB,	354+ doutB => drd(rtb)B,
= Island shad	339 dwrb => (others => 0)	355 (Uners => "0")
	341	357
Memgptrack.vhd	342— myFstoccbuf : Dpram size2kB a32b32	358+ mvFstoccbuf : Dobram size2kB a32b64
Mfsmtrack0.vhd	343 port map (359 port map (
E Mfsmtrack1ybd	344— resetA => reset,	
	<pre>345 clkA => mclk,</pre>	360 clkA => mclk,
Mgptrack.vhd	346	361
E Neotrng_v1_0.vhd	347 enA => enFstoccbuf,	362 enA => enFstoccbuf,
\equiv Rabled v1.0 vhd	349 WEA => 1 ,	364 WEA => (others => 1.7,
	350 aA => aEstoccbuf vec.	365+ addrA => aFstoccbuf vec.
/ GLIEDERUNG	351— drdA => open,	366+ doutA => open,
> ZEITACHSE	352- dwrA => dwrFstoccbuf,	367+ dinA => dwrFstoccbuf,

Vendor-agnostic probing of FPGA designs For efficient real-time debugging



FPGA Conference Europe 2024