

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK

Rendszerarchitektúrák labor Xilinx EDK

Raikovich Tamás

BME MIT





Labor tematika (Xilinx EDK)

- 1. labor:
 - A Xilinx EDK fejlesztői környezet ismertetése
- 2. labor:
 - Egyszerű processzoros rendszer összeállítása
 - Egyszerű szoftver alkalmazások készítése
- 3. labor:
 - Saját periféria illesztése
 - Megszakításkezelés
 - HW/SW együttes fejlesztés (debugger, ChipScope)



Témakörök

- Beágyazott rendszerek
- MicroBlaze processzor
- EDK alapok
- Gyári és saját IP-k hozzáadása
- Szoftverfejlesztés
- HW és SW együttes fejlesztése

BME-MIT

A ChipScope egy, az FPGA tervbe integrálható logikai analizátor, amely az FPGA terv belső jeleinek vizsgálatára használható

- Előnye a normál logikai analizátorhoz képest
 - Az igényeknek megfelelően konfigurálható
 - Nem kell kivezetni a jeleket FPGA I/O lábakra
 - Nem mindig van erre a célra elegendő I/O láb
- Hátránya a normál logikai analizátorhoz képest
 - FPGA erőforrásokat használ
 - Kevesebb erőforrás marad a terv számára
 - Kisebb kapacitású eszközök esetén korlátozottabb funkció

FPGA labor

A ChipScope logikai analizátor

A ChipScope rendszer felépítése:

• ICON

- JTAG kommunikáció
- Vezérlés
- ILA

- Logikai analizátor
- ChipScope Pro Analyzer SW



A ChipScope ILA bekötési példa (busz monitorozás):



Match Unit: a vizsgálandó jeleket összehasonlítja a megadott feltételekkel

- Basic: =, <> műveletek
- Advanced: =, <>, >, >=, <, <= műveletek
- Range: az Advaced műveleteken felül
 - "tartományon belül esik" művelet (in range)
 - "tartományon kívül esik" művelet (not in range)
- Mindhárom típus detektálhat éleket is
 - Basic with edges, Advanced with edges, Range with edges
- Eseményszámlálók:

- Pontosan n előfordulás
- Legalább n előfordulás
- Legalább n egymást követő (folyamatos) előfordulás

- Trigger feltétel (Trigger Condition)
 - A Match Unit események logikai vagy sorrendi kombinációja
 - Kijelöli a kezdőpontot a mintavételezett adatokat tartalmazó ablakban
- Tárolási feltétel (Storage Qualification Condition)
 - A Match Unit események logikai vagy sorrendi kombinációja
 - Eldönti, hogy kell-e tárolni az adott mintát



ChipScope beillesztése a processzoros rendszebe:

BME-M

- Debug menü → Debug configuration... → Add ChipScope Peripheral...
- A PLB busz monitorozására fogjuk használni: PLB IBA hozzáadása

System	Basic		
Monitor Hardware Signals → Debug Software Application → microblaze_0 → JTAG UART JTAG UART Add ChipScope Peripheral	<section-header><section-header><section-header><section-header><text><list-item><list-item><list-item></list-item></list-item></list-item></text></section-header></section-header></section-header></section-header>	Add New ChipScope Peripheral ChipScope cores can be added to monitor the various signals in achieve by adding a new core: To monitor PLB v4.6 bus signals (adding PLB IBA) To monitor arbitrary system level signals (adding ILA) To provide JTAG-based virtual input/output (adding <u>V</u> IO)	side the chip during run-time. Please indicate what you want
Delete ChipScope Peripheral			
Help	ОК Са		OK Cancel

ChipScope alap beállítások:



ChipScope haladó beállítások:

ystem	Basic Advanced	
Monitor Hardware Signals Chipscope_plbv46_iba_0 Debug Software Application Chimscroblaze_0 Miscellaneous Chipton JTAG UART	User System Buses Trigger Out and IBA Storage Trigger Out and IBA Storage Trigger Out and IBA Storage Trigger Out and IBA Storage PLB Address PLB Address PLB Data Signal storing enabled for the write data bus PLB Slave Control Bus Signal storing enabled for the write data bus PLB Slave Error Status Match unit type for the write data bus PLB Master Control Bus Match occurance counter width for the write data bus PLB Master Size And Type Enable probing the read data bus Deleting Internal port microblaze_0:DBG_STOP Deleting Internal port microblaze_0:DBG_STOP Deleting Internal port microblaze_0:DBG_STOP Deleting Internal port microblaze_0:DBG_STOP	
Add ChieGone Devictory 1	Information:	
<u>A</u> dd ChipScope Peripheral Delete ChipScope Peripheral	A new chipscope_icon instance is added into the system.	

- HW és SW együttes debuggolása
 - A ChipScope trigger kimenete le tudja állítani a program futását: töréspont a debuggerben
 - A debuggerben beállított töréspontra futás trigger eseményként szolgál a logikai analizátornak
- Új elemek a rendszerben
 - chipscope_icon_0: JTAG kommunikáció, vezérlés
 - chipscope_plbv46_iba_0: PLB busz analizátor
- A ChipScope hozzáadása után újra kell generálni a huzalozási listát és a konfigurációs bitfolyamot

- Indítsuk el a ChipScope Pro Analyzer programot
- Kattintsunk az Open Cable/Search JTAG Chain gombra
 - A JTAG láncban található eszközök azonosítása
- Kattintsunk az OK gombra: elindul az analizátor
 - Az alapértelmezett trigger beállításokkal
 - Az alapértelmezett hullámforma beállításokkal

	JTAG Chain	<u>D</u> evice	<u>W</u> indow	<u>H</u> elp
Iew Projec TAG Chair	.t		4	

BME-MI'

С	hipScop	oe Pro Analyzer				X
	JTAG C	hain Device Order:				
	Index	Name	Device Name	IR Length	Device IDCODE	USERCODE
	0	MyDevice0	XC3S500E	6	41c22093	
	1	MyDevice1	XCF04S	8	f5046093	
	2	MyDevice2	XC2C64A	8	06e5e093	
						Advanced >>
		Ok	Cancel	Read USE	ERCODEs	

A PLB busz jelek importálása

- − File menü → Import → Select New File
- implementation\chipscope_plbv46_iba_0_wrapper\chipscope _plbv46_iba_0.cdc fájl megnyitása
- Töröljük a meglévő jeleket a hullámforma ablakban
- Adjuk hozzá a szükséges PLB jeleket a hullámformához
 - − Jobb klikk a jelen → Add to View → Waveform



• A trigger és a tárolási feltételek beállítása

2	Trigge	er Setup - DEV:0 MyDevi	ice0 (XC3S500E) UNIT:0 My	ILAO (ILA)				¤″⊠
ž		Match Unit	Function		Value		Radix	Counter
atch	🔶 МЗ:	TRIG3	>			0000_0010_0000_0000	Hex	disabled 🔶
E	Add	Active	Trigger Conditio	n Name		Trigger Condition Equation		Output Enable
ġ	Del	۲	TriggerCondi	tion0			Disabled 📮	
 Capt 	Type:	Window	Windows:	1	Depth: 32	•	Position:	0
ure :	Storag	ge Qualification:		M1 && N	/12 && M3			

- A trigger élesítése
 - − Trigger Setup menü → Run vagy a ▶ gomb
- A debugger elindítása és várakozás a beállított feltételek teljesülésére



 A trigger feltétel teljesülése után a megadott számú minta tárolásra kerül és megjelenik a hullámforma ablakban

Waveform - DEV:0	MyDe	vice0 ((XC3S250E) UNIT:0 M	MLAO (ILA)				o o [X
Bus/Signal	x	0	2 -7	-2	3	8 	13	18	
PLB_RNW	0	0							1
- PLB_SaddrAck	0	0							
- PLB_SwrDAck	0	0							
- PLB_SwrComp	0	0							
⊶ PLB_ABus	8142)	8142)			{	81420000			
∽ PLB_wrDBus	018FI	01BFI	00FFE015		01FEF915	X	01FDA415	(01	-
•	• •	• •	•					•	
			X: -32	• •	0: -32	↓	Δ(X-0): 0		

Példa a ChipScope és a debugger együttes használatára:

- Timer megszakításos alkalmazás: álljon le a program futása, ha a LED-ekre legalább 0x10 értéket írunk ki
- HW/SW Co-Debug legyen engedélyezve (XPS: debug beállítások)
- Töröljük a hullámforma ablakban az összes jelet
- A hullámformához adjuk hozzá a következő jeleket:
 - PLB_RNW: írás (0) / olvasás (1) kiválasztó jel
 - PLB_PAValid: a cím érvényességét jelzi
 - PLB_SaddrAck: a cím nyugtázó jel
 - PLB_SwrDAck: az írási adatok nyugtázó jele
 - PLB_SwrComp: az írási adatátvitel végét jelzi
 - PLB_ABus: címbusz
 - PLB_wrDBus: írási adatbusz

A Match Unit-ok beállítása:

- Match Unit 0 (M0)
 - PLB hibajelek és PLB reset jel
 - MicroBlaze programszámláló és halted jelzés
 - Művelet: ==
 - Mindegyik bit legyen X
- Match Unit 1 (M1)
 - PLB vezérlő és nyugtázó jelek
 - Művelet: ==
 - Akkor legyen trigger, ha a címet nyugtázta a periféria
 - PLB_RNW: 0, PLB_PAValid: 1, PLB_SaddrAck: 1, a többi bit legyen X
- Match Unit 2 (M2)
 - PLB címbusz
 - Művelet: ==
 - Érték: a GPIO LED periféria címe (a formátumot állítsuk HEX-re)

A Match Unit-ok beállítása:

- Match Unit 3 (M3)
 - PLB írási adatbusz
 - Művelet: >=
 - A formátumot állítsuk HEX-re
 - Érték: 0000_0010 (64 bites esetben: 0000_0010_0000_0000)
- Match Unit 4 (M4)
 - PLB olvasási adatbusz
 - Művelet: ==
 - A formátumot állítsuk HEX-re
 - Érték: XXXX_XXXX (64 bites esetben: XXXX_XXXX_XXXX_XXXX)



A trigger és a tárolási feltételek beállítása:

- Trigger feltétel: M1 && M2 && M3
 - A címet nyugtázta a periféria (M1) és
 - A címbuszon a GPIO LED periféria címe van (M2) és
 - Az adat értéke nagyobb vagy egyenlő mint 0x10 (M3)
 - Break eseményhez: Output Enable legyen Pulse (high)
- Capture
 - Type: Window
 - Windows: 1
 - Depth: a minták száma (legyen például 64)
 - Position: a trigger esemény pozíciója (legyen például 20)
 - Tárolási feltétel: All Data

- Indítsuk el a debuggert, majd pedig élesítsük a triggert
- A beállított trigger feltétel teljesülésekor
 - A hullámforma ablakban megjelennek a minták
 - Leáll a program futása (break)

		_	-6	-5	-4	-3	-2 -	1 🗖		12	3	4	5	6	7	8	9							
Bus/Signal	X	0		Ī			Ī	Í			Ī		Ī	Ī		Ī	Ī							
PLB_RNW	0	0							\square															
- PLB_PAValid	0	0																						
- PLB_SaddrAck	0	0																						
- PLB_SwrDAck	0	0																						
- PLB_SwrComp	0	0																						
⊶ PLB_ABus	0000	0000	00404	X8					8140	0000					<u> Xoooo</u>	<u>1160</u>	00000							-
∽ PLB_wrDBus	00001	0000){§	X				0000	0021					<u> Xoooc</u>	001DC	00	00000	Ţ						
I I		∢ ►	•														•							
				>	: -20		•		0: -2	0	ŀ	• •	Δ(X-	0):0]							
								18	X	Gpio	Dis	crete	⊇Vrit	te(&	gpio	, 1,	leď	/alue);						
								19 20	,	- / h - ma	-	∍bít,	áota.	1760	tör	1600								
							\$	21	C;	sr =	XTm	rCtr	mGet	Con	trol	Stat	usRe	(XPAR)	KPS TI	IMER	O BA	SEAD	DR. C	n
								22	X	TmrCt	tr m?	SetCo	ontro	olSt	atus	Reg ()	XPAR	XPS TIM	IER O	BASE	EADDR	ι, ο,	csr)	;
							8	23}			_					- 1	-			-				