

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

Mikrorendszerek tervezése Debug lehetőségek

Fehér Béla Raikovich Tamás

BME-MI



FPGA labor

Bevezetés

- A debuggolás a beágyazott rendszerek fejlesztésének elengedhetetlen része
 - Alapszintű ellenőrzés és tesztelés
 - A hibák megkeresése és javítása
- Módszerek

BME-MI

- Hardver debuggolása: logikai analizátor
- Szoftver debuggolása: hardveres debug egység
 - A CPU vezérlését veszi át, normál működésnél tiltott
 - Hard processzoros rendszerek beépítve tartalmazzák
 - Szoft processzorok esetén általában külön periféria
- Debuggolás típusok
 - Funkcionális debuggolás
 - Teljesítmény debuggolás/analízis (profiling)

Bevezetés

- Az FPGA-k korlátozott belső láthatóságot biztosítanak
 - Hogy férjünk hozzá a buszhoz?
 - Hogy figyeljük a memóriát és a regisztereket?
- Hardveres debug eszköz
 - Vivado logikai analizátor
- Szoftveres debug eszköz

BME-MI

- Software Development Kit
- A szoftverfejlesztésnél volt már szó róla
- Kapcsolat a hoszt PC-vel egyetlen JTAG interfészen keresztül
 - Szoftver letöltése és debuggolása
 - MicroBlaze esetén: MicroBlaze Debug Module (MDM) periféria
 - Hardver debuggolása a belső logikai analizátorral
 - HW és SW együttes debuggolás is lehet (cross-trigger)

SDK – Debug perspektíva (ismétlés)



SDK – Debug funkciók (ismétlés)

A debugger funkciói

- Töréspontok elhelyezése és eltávolítása:
 - Dupla kattintás a sor száma mellett a szürke területen

32 //Clear the timer interrupt flag. o 33 csr = XTmrCtr_GetControlStatusReg(XPAR_AXI_TIMER_0_BASEADDR, 0); 34 XTmrCtr_SetControlStatusReg(XPAR_AXI_TIMER_0_BASEADDR, 0, csr);

- A program végrehajtás vezérlése
 - Resume: a program futásának folytatása
 - Suspend: a program futásának megállítása
 - Step Into: az aktuális forráskód sor végrehajtása
 - Függvényhívás esetén belép a függvénybe
 - Step Over: az aktuális forráskód sor végrehajtása
 - Függvényhívás esetén lefut a függvény, nem lép be a függvénybe
 - Step Return: a futás leáll a függvényből való kilépéskor
 - Run to Line: futtatás a kijelölt forráskód sorig

SDK – Debug funkciók (ismétlés)

A debugger funkciói

Forráskód nézet (C, disassembly)

- Töréspontok elhelyezése, eltávolítása
- Változók értékeinek megtekintése
 - Vigyük az egérkurzort a változó fölé
- Debug nézet
 - Veremkeret
 - A program végrehajtás vezérlése
- Variables: a lokális változók listája, értékeik módosítása
- Breakpoints: töréspontok engedélyezése, tiltása
- Registers: a CPU regiszterek listája, értékeik módosítása
- Expressions: kifejezések értékének figyelése (watch)
- Memory: memóriatartalom megjelenítése, módosítása

BME-MIT

A Vivado logikai analizátor felépítése

ICON (Integrated CONtroller)

- JTAG kommunikáció
- Vezérlés

BME-MIT

• ILA (Int. Logic Analyzer)

- Logikai analizátor
- VIO (Virtual I/O)
 - Belső FPGA jelek meghajtása
 - Belső FPGA jelek monitorozása
- Vivado HW manager
 - Konfiguráció, megjelenítés



Az ILA egység belső felépítése

BME-MIT



FPGA labor

A trigger komparátor a vizsgálandó jeleket összehasonlítja a megadott feltételekkel

- Egyenlőség (==) és nem egyenlőség (!=) vizsgálata
- Kisebb (<), kisebb vagy egyenlő (<=), nagyobb (>) és nagyobb vagy egyenlő (>=) relációk vizsgálata
- Felfutó (R) és lefutó (F) él detektálása
- Don't care (X), nincs érték vizsgálat
- Értéktartomány vizsgálata két komparátorral
 - "tartományon belül esik" művelet
 - "tartományon kívül esik" művelet

BME-MI

- Trigger feltétel (Trigger Condition)
 - A trigger komparátor események logikai (AND, OR, NAND, NOR) vagy sorrendi kombinációja
 - Kijelöli a kezdőpontot a mintavételezett adatokat tartalmazó ablakban
 - Trigger állapotgép összetett feltételekhez
- Tárolási feltétel (Storage Qualification Condition)
 - A trigger komparátor események logikai (vagy sorrendi) kombinációja
 - Eldönti, hogy kell-e tárolni az adott mintát
- A minták tárolása Blokk-RAM-ba történik

BME-MIT

ILA egység

- Részletek: Integrated Logic Analyzer v6.2 PG (PG172)
- Belső FPGA jelek vizsgálata utólagos analízishez
- Konfigurálható számú bemenet (Probe)
 - GUI esetén max. 64, TCL parancsokkal max. 1024
 - Felhasználás: trigger, adat vagy mindkettő
- Bemenetenként konfigurálható számú trigger komparátor
- Sorrendi triggerelési lehetőség
 - Trigger állapotgép
- Tárolási feltétel megadható
- Cross-trigger lehetőség
 - Trigger IN, Trigger OUT
- A trigger feltétel helye az ablakon belül megadható

BME-MIT



System ILA egység

- Részletek: System ILA v1.0 Product Guide (PG261)
- A Normál ILA képességei itt is megvannak
- Konfigurálható natív és interfész bemenet típusok
 - Minden bemenetre külön-külön megadható
 - Interfész esetén protokoll szintű információk megjelenítése a felhasználói felületen
 - AXI4 és AXI4-Stream protokoll ellenőrzés



FPGA labor

VIO egység

- Részletek: Virtual Input/Output v3.0 PG (PG159)
- Belső FPGA jelek meghajtása és figyelése valós időben
- A portok száma és mérete konfigurálható

BME-MIT

Opcionális aktivitás detektorok a bemeneteken



Logikai analizátor egységek hozzáadása

- Példányosítás a HDL kódban
- Hozzáadás az IP Integrator-ban (ez javasolt)
 - Az egység hozzáadható az IP katalógusból
 - A vizsgálandó jelek interaktívan beköthetők
- Jelek megjelölése vizsgálatra (Mark Debug)
 - Megadás a HDL kódban: (* mark_debug = "true" *)
 - Megadás TCL parancs segítségével
 - Megadás az XDC fájlban
 - Jel megjelölése az IP Integrator-ban

BME-MI



FPGA labor

Logikai analizátor egységek hozzáadása

"Mark Debug" esetén a szintézis után adható a rendszerhez és konfigurálható a logikai analizátor

- A szintetizált rendszer megnyitása
- Flow Navigator vagy Tools menü → Set Up Debug
- Vizsgálandó jelek kiválasztása, az órajel tartomány és a bemenet típus (adat, trigger) megadása
- ILA egység konfigurálása

A logikai analizátor konfigurálása

- Trigger mode: a trigger esemény forrása
 - BASIC_ONLY: egyszerű trigger feltétel
 - Alap relációk és ezek AND/OR kapcsolata
 - ADVANCED_ONLY: összetett trigger feltétel megadása trigger állapotgéppel
 - TRIG_IN_ONLY: TRIG_IN bemenet
 - BASIC_OR_TRIG_IN
 - ADVANCED_OR_TRIG_IN
- TRIG_OUT mode: a TRIG_OUT kimenet használati módja
 - DISABLED: nincs használva a kimenet
 - TRIGGER_ONLY: a trigger feltétel jelzése
 - TRIG_IN_ONLY: a TRIG_IN bemeneten érkező jelzés továbbítása
 - TRIGGER_OR_TRIG_IN

BME-MIT

Status -	hw_ila_1	? _ 🗆
;		
BASIC_	ONLY	~
DISABL	ED	~
S		
	BASIC	~
'S:	1	[1 - 1024]
1:	1024	✔ [1 - 1024]
window:	512	[0 - 1023]
)	ms	
	Status - 1 BASIC_ DISABL s s: n: window:	Status - hw_ila_1 BASIC_ONLY DISABLED S BASIC S: 1 1 1024 window: 512 ms

A logikai analizátor konfigurálása

- Capture mode: tárolási feltétel
 - ALWAYS: minden minta tárolásra kerül
 - BASIC: alap relációk és ezek AND/OR kapcsolata
- Number of windows: a tárolási ablakok száma
 - A trigger élesítésre kerül a következő ablakra, amíg tele nem lesz a mintákat tároló memória
- Window data depth: egy ablak hány mintát tárol
- Trigger position in window: a trigger esemény helye az ablakon belül

Trigger feltétel		Trigger Setup - hw_ila_1 & Capture Q + = D + LO	setup - hw_	ia_1 apcsolat		Tárolási fe feltéte	eltétel I beállí	beállítá: ításhoz h	s (a triggo asonló)	er ? _ 0
Dealitas		Name	Operation	ator	Radio	C	Value		Port	Comparator Usage
(BASIC)		slot_0 : microblaze_0_axi_dp : AWADI	DR ==	~	[H]	~	4000_00	• 00	probe4[31:0]	1 of 2
		slot_0 : microblaze_0_axi_dp : AWRE/	ADY ==	↑ ~	[B]	~	1	~	probe13[1]	1 of 2
Jelek	۲	slot_0 : microblaze_0_axi_dp : AWVAL	.ID ==	Reláció	[B]	~	1 F	igyelend	lő érték	1 of 2
BME-MIT		A			1		- 75			

Példa – AXI4-Lite busz monitorozása

- LED GPIO periféria a 0x4000000 címen
- GPIO adatregiszter írás figyelése System ILA egységgel
 - Trigger: AWADDR=0x40000000 & AWVALID=1 & AWREADY=1



BME-MIT /

- Hardver és szoftver együttes debuggolása
 - Hol a hiba? A hardverben vagy a szoftverben?
- HW→SW: ILA trigger esemény a Trigger OUT porton
 - Töréspont a CPU-n futó programban
- SW→HW: a CPU-n futó program töréspontra fut
 - Trigger esemény az ILA számára a Trigger IN porton
- Szükséges beállítások

BME-MT

- MicroBlaze CPU: Extended Debug
- MDM: Enable Cross Trigger
- ILA: Trigger Out Port, Trigger In Port

- Az ILA és az MDM összekapcsolása
 - ILA TRIG_OUT port → MDM TRIG_IN port
 - MDM TRIG_OUT port → ILA TRIG_IN port
- ILA trigger esemény jelzése a MicroBlaze CPU felé
 - − Vivado-ban → TRIG_OUT mode: TRIGGER_ONLY
- ILA trigger bemenet (TRIG_IN) figyelembe vétele
 - Trigger mode: BASIC_OR_TRIG_IN
 - Trigger mode: TRIG_IN_ONLY



HW→SW: ILA trigger esemény hatására a program futása megáll a triggert okozó utasítást követő utasításnál



20

SW→HW: töréspontra fut a program, amely trigger esemény az ILA számára (a helyét az ablak közepe vagy vége felé állítsuk be)

	for	(;;)									
	{										
		MEM32 (XPAR GPIO LED DISP BASEA	DDR + XGPIO	DATA OF	FSET) = c	nt;					
≈		<pre>sleep(1);</pre>							-		
		cnt = (cnt + 1) & 0xff;					Sec. 2.4				
	}					_					
		ILA Status: Idle									
		Name	Value		490	495	, 500 <u>,</u>	505		510	í.
		∨ 🛋 slot_0 : microblaze_0_axi_dp : Interface	_				Active	Act			
		> 🗮 slot_0 : microblaze_0_a: Write Transactions (-				Write (ID:0	Addr:0x4		— —	
		> 🔚 slot_0 : microblaze_0_axi_dp : AR Channel	No Read Addr Cn			1	l <mark>o Read Addr Cm</mark>	ds			
	111	> 🔚 slot_0 : microblaze_0_axi_dp : R Channel	No Read Data Be			N) Read Data Bea	ats			
		∨ 🛋 slot_0 : microblaze_0_axi_dp : AW Channel	_				Addr Cmd			— —	
	200	🔓 slot_0 : microblaze_0_axi_dp : AWVALID	0								
		Islot_0 : microblaze_0_axi_dp : AWREADY	0								
		> 😻 slot_0 : microblaze_0_axi_dp : AWADDR	0000064c	0000057	20 X X X	00001	4000	0000	XiiXii	00)0054c
		> 😽 slot_0 : microblaze_0_axi_dp : AWPROT	Data Secure Unp			Data	a Secure Unpriv:	leged			
		> 😽 slot_0 : microblaze_0_axi_dp : AW_CNT	0		0		X			0	
		🗸 🔚 slot_0 : microblaze_0_axi_dp : W Channel	_							_	
		🔓 slot_0 : microblaze_0_axi_dp : WVALID	0								
		Islot_0 : microblaze_0_axi_dp : WREADY	0								
		> 😽 slot_0 : microblaze_0_axi_dp : WDATA	000002b8	0000	00000 X	X	000000	000		00) 00 2Ъ8
		> 😽 slot_0 : microblaze_0_axi_dp : WSTRB	f	f	1	X	f		(1)		f
		> 📕 slot_0 : microblaze_0_axi_dp : B Channel	_								

Egyéb hardver debug lehetőségek

Nagysebességű interfészek vizsgálata

- IBERT (Integrated Bit Error Ratio Tester) és Vivado Serial I/O Analyzer
 - A nagysebességű GTX/GTH/GTY transceiver-ek monitorozása és dinamikus átkonfigurálása
 - Minta generátor és ellenőrző a bithibaarány vizsgálatához (jelút minősége)
- MIG (Memory Interface Generator)

BME-MI

- A debug port vizsgálható logikai analizátorral
 - Kalibrációs és adat olvasási jelek



- A szoftver kritikus részeinek meghatározása
 - A profiler eszköz megméri, hogy a processzor mely függvények végrehajtásával tölti az idejét
 - Hasonló a hardveres időzítés analízishez
 - Információt ad arról, hogy mely szoftver rutinok lennének jelöltek a hardveres gyorsításhoz
- A függvények hatékonyságának javítása

BME-MIT

- Gyorsabb C kód (pl. pointer aritmetika mellőzése)
- Megvalósítás C helyett assembly nyelven
- Speciális CPU utasítások használata (pl. SIMD)

A funkció hardveres megvalósítása, gyorsítása

- Az SDK tartalmazza a GNU profiler (gprof) eszközt
- A profiler által adott mérési eredmények
 - Függvényhívási gráf: mely függvény mely függvényt hívott meg és hányszor
 - Hisztogram: összesen mennyi idő lett eltöltve az egyes függvények végrehajtásával

Name (location)	Time	Calls	Time/Call	%Time 👻
Summary	403.599ms	1	4	100.0%
MAD_F_MUL_28	189.99ms	102400	1.846us	46.85%
MAD_F_MUL_28 (lab5.c:62)	96.299ms			23.86%
MAD_F_MUL_28 (lab5.c:61)	87.900ms			21.78%
MAD_F_MUL_28 (lab5.c:63)	4.900ms			1.21%
■muldi3	107.799ms			26.71%
■ dot32	101.199ms	100	1.11ms	25.07%
dct32 (lab5.c:77)	45.699ms			11.32%
dct32 (lab5.c:78)	45.300ms			11.22%
dct32 (lab5.c:76)	9.100ms			2.25%
dct32 (lab5.c:71)	499.999us			0.12%
	499,999us			0.12%

BME-MIT

Hisztogram

Name (location) Time Calls Time/Call %Time 403.599ms Summary 00.0% MAD_F_MUL_28 102400 1.846us 189.99ms 102400 988ns parents 101.199ms 07% 102400 dct32 (lab5.c:77) 101.199ms 988ns _muldi3 107.799ms dct32 101.199ms 100 1.11ms children 102400 1.846us 189.99ms 102400 1.846us MAD_F_MUL_28 (lab5.c:61) 189.99ms 100 23.0us 0.57% parents 2.300ms main (lab5.c:117) 2.300ms 100 23.0us 0.57% 0.79% exit 3.199ms 0.57% 🖃 main 2.300ms 0 children 101.199ms 100 1.11ms 25.07% dct32 (lab5.c:71) 25.07% 101.199ms 100 1.11ms

Függvényhívási gráf

- A GNU profiler HW és SW erőforrásokat igényel
 - Hardveres időzítőt
 - Memóriaterületet a mérési eredmények számára
 - A programba profiling rutinok kerülnek
- Az időzítő adott időközönként megszakítást kér
 - A megszakításkezelő rutin a programszámláló alapján hisztogramot készít a memóriában
 - A megszakítás gyakorisága beállítható
- Minden függvényhívás megjelölésre kerül a fordító által a nyomon követhetőség végett
 - A kód strukturáltsága befolyásolja az analizálhatóságot
 - Pl. csak egy main() függvényből álló program alkalmatlan
 - Az algoritmust logikailag szét kell bontani függvényekre

A Board Support Package konfigurálása

- Processzor: drivers → microblaze_0 (vagy a megadott név)
 - Az extra_compiler_flags-hoz a -pg hozzáadása
- Operációs rendszer: standalone
 - Az enable_sw_intrusive_profiling=true beállítása, profile_timer megadása
- A szoftver alkalmazás konfigurálása
 - A kívánt alkalmazás projekt kiválasztása
 - Run menü → Run Configuration → Xilinx C/C++ Application (System Debugger): új konfiguráció vagy a már meglévő kiválasztása
 - − Application fül → Advanced options → Edit...
 - Enable Profiling (gprof) engedélyezése és a mintavételi frekvencia, valamint a mérési adatbuffer kezdőcímének beállítása
 - Az adatbuffer tartalmát a szoftver alkalmazás nem módosíthatja!
- A program futásának befejeztével a mérési eredmények a gmon.out fájlba íródnak és ez automatikusan megnyitásra kerül