Anexa 1

ISE Quick Start Tutorial, adaptat pentru ISE 14.7

Pornirea mediului ISE

Dublu click pe icoana de pe desktop, sau se pornește din Start \rightarrow Programs \rightarrow Xilinx Design Tools \rightarrow ISE Design Suite 14.7 \rightarrow ISE Design Tools \rightarrow Project Navigator

Atenție (!): Asigurați-vă că nu deschideți versiunea mai veche Xilinx ISE 9.2i care poate fi de asemenea prezentă pe stațiile de lucru din laborator.

Acces la Help

Pentru a accesa Help-ul optați pentru una din variantele următoare:

- Apăsați **F1** pentru a vizualiza Help-ul pentru o anumită unealtă sau funcție selectată
- Lansați **ISE Help Topics** din meniul Help. Veți accesa informații diverse despre crearea și mentenanța întregului ciclu de dezvoltare în ISE.



Figure 1: ISE Help Topics

Crearea unui nou proiect

Creați un nou proiect ISE, care va fi dedicat pentru dispozitivul FPGA prezent pe placa de dezvoltare Basys Spartan-3E.

Pentru a crea noul proiect:

- 1. Selectați din meniu **File → New Project...** Se va deschide New Project Wizard.
- 2. Introduceți test_env în câmpul Name.

- 3. Introduceți sau accesați o locație (cale de director, amintiți-vă regulile!) pentru proiectul nou. Un subdirector test_env se va crea automat.
- 4. Verificați ca valoarea din câmpul Top-Level Source Type este HDL.
- 5. Click **Next** pentru a merge la pagini cu proprietățile dispozitivului pentru implementare și testare.
- 6. Completați proprietățile conform listei de mai jos:
 - Product Category: All
 - Family: **Spartan 3E**
 - Device: XC3S100E
 - Package: TQ144
 - Speed Grade: -4
 - Top-Level Module Type: HDL
 - Synthesis Tool: XST (VHDL/Verilog)
 - Simulator: ISim (VHDL/Verilog)
 - Preferred language: VHDL
 - Lăsați valorile implicite în celelalte câmpuri.

Crearea unui fișier sursă VHDL

- 1. Click în meniu pe **Project\New Source**.
- 2. Selectați VHDL Module în lista Source type.
- 3. Introduceți numele fișierului *test_env* (! Nu e obligatoriu sa fie acelasi nume cu proiectul). Acest nume îl va avea și entitatea creată automat în acest fișier.
- 4. Verificați că este bifată căsuța Add to project.
- 5. Click Next.
- 6. Acum declarați porturile pentru entitatea principală ce se va crea, completând informația pentru porturi ca în figura de mai jos. Atenție: aceste porturi sunt definite în mod particular pentru placa Basys, fiind în principiu suficiente pentru majoritatea proiectelor dezvoltate pe parcursul acestui semestru.

> New Source Wiz	zard						×
Define Module Specify ports	s for module.						
Entity name	test_env						
Architecture name	Behavioral						
	Port Name	Directi	on	Bus	MSB	LSB	_
dk		in	•				
btn		in	•		3	0	
sw		in	•		7	0	
led		out	•		7	0	
an		out	•		3	0	
cat		out	•	✓	6	0	
dp		out	•				
		in	•				
		in	•				
		in	•				
		in	•				-
More Info			< <u>B</u> a	ack	<u>N</u> ext >	Can	cel

Figura 2: Definirea porturilor prin interfața Xilinx

7. Click **Next**, reverificați sumarul declarațiilor de porturi (apăsați **Back** dacă sunt necesare corecții) și apăsați **Finish** pentru a finaliza crearea noului fișier sursă.

Observație: se poate sări peste pasul de definire a porturilor în interfață (pas 6), caz în care se pot declara (sau modifica/corecta) porturile in secțiunea de declarare a entității create în noul fișier.

Fișierul sursă care conține declararea entității *test_env* și arhitectura ei este afișat acum în mediul ISE, iar în zona **Hierarchy** apare ca modulul principal (Top Module) al proiectului curent.

De reținut: în proiectele cu mai multe surse, daca se schimbă accidental entitatea Top Module, se poate seta alta ca Top Module prin click dreapta pe sursa dorită în **Hierarchy,** după care click pe **Set as Top Module**.

Atenție la părintele *test_env* în ierarhie: numele este format din proprietățile dispozitivului pe care îl veți încărca. Pentru Basys, trebuie să fie **xc3s100e-4tq144**. Dacă nu coincide, înseamnă ca ați sărit peste pasul 6 la **Crearea unui nou proiect**. Dublu click pe părinte și introduceți proprietățile respective.

\geq	ISE Project Navigator (P.20131013) - D:\users	:\flori	n\2003-2004\Te	aching\ac2015\lab\teste\test_env\te	est_env.xise - [test_env.vhd]			X
E	Ele Edit Yew Project Source Process Tools	: <u>W</u> in	dow Layout Hel;					<u>- 0 x</u>
ll L]≫⊟₽ \$]%00× ¤(24	»] 🎤 ሯ 🌮	878 <u> </u>	🎤 🕅 🕨 🖄 🛣 🗸			
Desk	gn ↔⊡∄×		2 C	ompany:				×
	View: Vi	Þ	3 E	ngineer:				
8	Herarchy		4 c	reate Date: 12:14:10 02/	20/2015			
81	- 😇 test_env	10	6 D	esign Name:				
	E xc3s100e-5cp132		7 M	odule Name: test_env - H	Sehavioral			
00	Table rest_env - benevicien (rest_env.vi		8 P	roject Name:				
6			10 T	ool versions:				
12		1	11 D	escription:				
		%	12					
_		34	13 D	ependencies:				
		36	15 R	evision:				
		-	16 R	evision 0.01 - File Created	1			
		G	17 A	dditional Comments:				
		$ \odot $	18					
			20 11br	ary IEEE;				
			21 use	IEEE.STD_LOGIC_1164.ALL;				
			22					
	▲		23 0	rithmetic functions with Si	ary declaration if using igned or Unsigned values			
	No Processes Running		25us	e IEEE.NUMERIC STD.ALL;				
-	Deserve has an Debusined		26					
ΨĘ,	Processes: test_env - benavioral		27 0	ncomment the following libs	ary declaration if instantiating			
앬	Design Summary/Reports Design Utilities		28 a 2911	brary UNISIM;	is code.			
91	🕀 🈼 User Constraints		30us	e UNISIM.VComponents.all;				
-	B 00 Synthesize - XST		31					
ш	Implement Design Generate Programming File		32 enti	ty test_env is				
	Configure Target Device		34	btn ; in STD LOGIC	VECTOR (3 downto 0);			
	Analyze Design Using ChipScope		35	sw : in STD_LOGIC_V	ECTOR (7 downto 0);			
			36	led : out STD_LOGIC	VECTOR (7 downto 0);			
			37	an : out STD_LOGIC	VECTOR (3 downto 0);			
			39	dp : out STD LOGIC)	,			
			40 end	test_env;				
			41					
			42 arch	itecture Benavioral of test	_env 1s			
			44 begi	n				-
			4					
>	Start Cart Design The Files		,	test env.vhd		Design Summary (out of date)	X	
_	· · · · · · · · · · · · · · · · · · ·				<u> </u>			
Cons	sole						-	+ 0 5 ×
18	INFO:HDLCompiler:1061 - Parsing INFO:ProjectMont - Parsing desig	whDi m hi	erarchy com	sers/florin/2003-2004/feach	ing/ac2015/1ab/teste/test_env/tes	_env.vnd" into library wor	rĸ	*
1								
1								
L.								Y
면		-	The Develop					<u> </u>
	Console 😈 Errors 🗘 Warnings 🖓 H	na in F	Ties Results					10.25 Col 1 . MOI

Figura 3: Noul proiect în ISE

Asigurați-vă că următoarele librării sunt incluse în zona de declarare a librăriilor în test_env:

use IEEE.STD_LOGIC_ARITH.ALL; use IEEE.STD_LOGIC_UNSIGNED.ALL;

Dacă nu sunt incluse, adăugați-le.

(De revenit aici când se va cere...) Folosirea Language Templates (VHDL) – optional

Următorul pas în procesul de creare a noii surse VHDL este adăugarea descrierii comportamentale pentru entitatea Top Level a proiectului. O modalitatea facilă (pentru început, până vă reamintiți limbajul VHDL) este să folosiți exemple de cod din ISE Language Templates pe care să le particularizați pentru entitatea voastră.

- 1. Plasați cursorul de editare sub begin în secțiunea architecture a entității unde doriți să adăugați codul.
- 2. Deschideți Language Templates prin selectarea din meniu a Edit → Language Templates...
- 3. Navigați în ierarhie prin simbolul "+", către exemplele de cod: VHDL → Synthesis Constructs → Coding Examples → ...
- 4. Selectați componenta dorită în ierarhie, apoi click dreapta → Use in File. Acest pas va copia modelul de cod în fișierul sursă unde ați plasat cursorul de editare.

- 5. Închideți Language Templates.
- 6. Înlocuiți denumirea implicită a semnalelor din codul inserat cu denumirea semnalelor din entitatea pe care o descrieți.

Editarea finala / sintetizarea sursei VHDL (descrierea comportamentului)

- 1. Adăugați componente (cu *component*) și / sau declarații de semnal în secțiunea de declarații între architecture și begin.
- 2. Adăugați restul codului (instanțierea componentelor *port map*, descrierea comportamentală *process* sau atribuiri concurente, etc.) între begin și end.
- 3. Pentru acest prim proiect adăugați următoarele atribuiri concurente după begin:

led <= sw; an <= btn; dp <= '0'; cat <= (others=>'0');

- 4. Salvați fișierul cu File → Save or Ctrl + S.
- 5. În zona *Hierachy* selectați entitatea Top Level, în cazul de față *test_env*.
- 6. Verificați că sintaxa VHDL este corectă: în zona Processes (stânga-mijloc în ISE) dublu click pe **Synthesize XST → Check Syntax**
- 7. Corectați eventualele erori care sunt raportate în zona Errors (partea de jos a ISE). Începeți procesul de corectură cu prima eroare!
- 8. Sintetizați proiectul: dublu click pe Synthesize XST
- 9. Vizualizați circuitul rezultat sub formă schematică (relevant mai ales pentru următoarele proiecte, unde vor fi circuite mai complexe): dublu click pe Synthesize XST → View RTL Schematic. Pe dialogul care se deschide asigurați-vă ca e selectată varianta a doua (*Start with a schematic of the top level block*), apăsați OK. Se va deschide o schemă bloc a circuitului principal, dublu click pe el pentru a vedea organizarea internă. Ar trebui să recunoașteți cel puțin o parte a componentelor declarate! Aceasta este o primă metodă de verificare ca ați descris corect componentele dorite.

Acum ați finalizat procesul de creare a sursei VHDL, fără erori de sintaxă!

Imlementarea proiectului si verificarea constrângerilor

- 1. Selectați sursa *test_env* zona **Hierarchy**.
- 2. Implementați proiectul: dublu click pe Implement Design în zona Processes.
- 3. Dacă totul decurge fără erori, în drept cu **Implement Design** va avea un semn de bifat verde. Dacă există avertismente va apare un triunghi galben cu semn de exclamare. Corectați eventualele erori și avertismente (o parte din avertismente nu necesită corectură, dacă sunt irelevante: ex. anumite semnale sunt declarate dar nu sunt legate încă în proiect).
- 4. Deschideți sumarul proiectării prin dublu click pe procesul **Design Summary/Reports** în zona **Processes**. Consultați rapoartele (Summary,

Timing constraints, etc.) pentru a vedea și analiza diferitele aspecte de performanță ale proiectului: resurse utilizate pe de dispozitivul țintă (placa), aspecte legate de ceas, etc. Chiar dacă pentru acest prim circuit simplu aceste rapoarte sunt mai puțin relevante, în proiectele viitoare ele vor fi necesare.

Stabilirea constrângerilor pentru locațiile de pini (atribuirea pinilor)

Specificați care locații de pini de pe placa de dezvoltare vor fi atribuiți porturilor din proiect (entitatea Top Level *test_env*). Există mai multe metode, mai jos aveți abordarea recomandată (fără probleme de compatibilitate):

- Adăugați un fișier user constraints din meniu Project\New Source, alegeți tipul sursei ca "Implementation Constraints file", introduceți numele fișierului la fel (nu e obligatoriu) ca entitatea Top Level test_env.
- Acum, in zona Hierarchy, fisierul test_env.ucf ar trebui să apară sub test_env.vhd (expandați cu "+"). Selectați test_env.ucf. În zona Processes, dublu click pe User Constraint->Edit Constraints (Text), pentru a edita în mod text test_env.ucf.
- 3. În general, atribuirea unui pin la un port existent se face folosind eticheta "net". În particular, pentru porturile definite în acest tutorial, găsiți constrângerile definite <u>aici</u>. Deschideți acest fișier într-un editor text, copiați / inserați conținutul în test_env.ucf. Examinați sintaxa pentru eticheta "net".

Pentru viitor, dacă se adaugă porturi noi, se pot defini noi constrângeri în fișierul *ucf* folosind eticheta "net", și manualul de referință pentru placa Basys (pentru identificarea locațiilor de pini).

Datorită modificării fișierului ucf, procesul **Implement Design** în zona **Processes** are un semn de întrebare portocaliu, semn că nu este actualizat. Rulați-l iar sau treceți la pasul următor (se va rula oricum automat).

Generarea fișierului de programare (*.bit)

- Mai întâi trebuie schimbată proprietatea FPGA Start-up Clock la JTAG Clock: click dreapta Generate Programming File, Process Properties → Startup Options → FPGA Start-Up Clock → JTAG Clock.
- Dublu click pe procesul Generate Programming File în zona Processes. Rularea acestui proces va duce automat la rularea proceselor anterioare (Implement Design, Synthesize – XST) dacă sursele au fost modificate de la ultima rulare.

În acest moment, dacă nu există erori, fișierul test_env.bit a apărut în directorul proiectului.

O eroare frecventă care apare este de tipul "No such site on the Device". Eroarea e din cauză că ați sărit peste pasul 6 la **Crearea unui nou proiect.** În acest caz dați dublu click pe părintele entității Top Level a proiectului, așa cum apare în zona Hierarchy. Introduceți proprietățile corecte.

Încărcarea proiectului (*.bit) pe placa de dezvoltare Spartan™-3E Basys

Dacă apar probleme în timp ce parcurgeți pașii următori, vedeți lista (după figură) problemelor frecvente și soluții.

- 1. Conectați placa la portul USB.
- 2. Porniți unealta Adept: Start \rightarrow Programs \rightarrow Digilent \rightarrow Adept.
- 3. Apăsați butonul Initialize Chain.
- 4. Navigați (browse) și selectați fișierul bit al proiectului.
- 5. Programați dispozitivul FPGA.

nfig	Test	Register I/0) File I/O	I/O Ex	Settin	gs	
FP	GA				-) [E	Browse	Program
XC3S	250E						
XCF	025				• [i	Browse	Program
			Initialize	Chain)		

Figure 4: Unealta Adept de la Digilent. Atenție, exemplul acesta este pentru placa Basys 2

Probleme legate de programare și eventuale soluții

Problemă

Placa Basys nu este recunoscută

Soluție (luați-le în ordine, și reporniți Adept-ul după fiecare încercare nereușită):

- a) Încercați alt port USB (față sau spate), dacă începe automat un proces de instalare a driver-ului și vă cere drepturi de administrator, cereți ajutorul profesorului.
- b) Verificați dacă placa nu necesită alimentare externă.
 - Dacă nu necesită (nu are marcajul "E" în poziția 1 din figura de mai jos) atunci asigurați-vă că Switch-ul (poziția 3 în imaginea de mai jos) este pe poziția VUSB
 - > Dacă necesită, atunci vezi explicațiile următoare



- 1. Folosiți o sursă de 3.3 V pentru alimentare în mufa 2.
- 2. Switch-ul din poziția 3 trebuie pozitionat pe pozitia *VEXT* pt. alimentare externa, sau pe *VUSB* altfel (pt. placile ok)
- c) Încercați cu alt cablu.
- d) Încercați altă placă (raportați asta profesorului).
- e) Schimbați stația de lucru.