# 2.3.) Симулация на сода машината в симулатора Active-HDL Sim на Warp

=> Стартирайте Warp апликацията като натиснете Start, след това изберете съответното име на апликацията от *Programs -> Cypress -> Warp menu*.

Най-напред в текстовия редактор Galaxy се създава нов проект drink1, в който да се включи създаденият в ACTIVE-HDL FSM, VHDL файл binctr.vhd.

### 2.3.1. Създаване на нов проект drink1 в Galaxy

Стартирайте *Galaxy*.

=> Изберете **File** -> **New**.

=> Изберете **Project [Target-Device]** и натиснете **OK**, както е показано на Фигура 13.

New	? ×
New Text File	ОК
Project [Target - Device] Project [Target - Library]	Cancel

Фигура 13 Диалогов прозорец за нов файл и проект.

=> Изберете VHDL като тип на проекта, Project Type.

5	Project Type	
al a	Project Name:	
6	Project Path: C:\Mariya Damyanova\soda machine s HDL F	Browse
	< <u>B</u> ack <u>N</u> ext >	Cancel

Фигура 14 Въвеждане на име на проекта.

=> Въведете името на проекта, "drink1".

=> Въведете пътя до проекта както следва.

Въведете c:\your name\soda machine.

=> Натиснете Next за да извикате Add Files wizard. Add Files wizard се използва за копиране на набор от VHDL файлове в текущия проект. Пропуснете тази стъпка и натиснете Next, за да извикате Target Device wizard.

Забележка – Затварянето на Project Information wizard създава проект без избрани файлове и уреди. Ще можете да добавите файлове и да изберете уред от Project менюто по-късно.

Избраният диалогов прозорец на **Target Device wizard** показва всички Cypress PLD-та в дървоводна форма, която показва следната йерархия: семейство схеми (SPLD, CPLD, и т.н.), подсемейство схеми (Ultra37000, Flash, MAX, и т.н.), име на фамилията схеми и тип на пакета. Потребителят може да разгледа дървото на схемите. От лявата страна на диалоговия прозорец са наличните схеми, а от дясната са наличните пакети за избраната схема.

Device:	Package:	
<ul> <li>Ultra 37000</li> <li>c37032</li> <li>c37064</li> <li>c37128</li> <li>c37192</li> <li>c37256</li> <li>c37384</li> <li>c37512</li> <li>c37032v</li> <li>c37064v</li> </ul>	CY37256P160-154AC CY37256P160-125AC CY37256P160-125AI CY37256P160-125UMB CY37256P160-83AC CY37256P160-83AI CY37256P160-83UMB CY37256P208-154NC CY37256P208-125NC CY37256P208-125NI CY37256P208-83NC CY37256P208-83NI	
Device/package: c37256, CY37256P160- In-System Reprogrammable (TM) CPLD Number of macrocells = 256 Lead Thin Plastic Quad Flat Pack fmax = 83MHZ, Temp Range:Commercial (	33AC D C to +70 C)	

Фигура 15 Диалогов прозорец за избиране на целеви уред със Cypress PLD-та в дървовидна форма.

- => Изберете последователно CPLD -> Ultra37000 -> c37256.
- => Изберете пакета СҮЗ7256Р160-83АС в дясната рамка.
- => Натиснете **Finish**, за да създадете проекта.

=> Натиснете **Yes**, за да съхраните проекта.

Сега създадохте проектна директория за Вашите разработки на име *Вашето име*. Всички файлове, отнасящи се до това ръководство трябва да бъдат поставени в тази проектна директория.

## 2.3.2. Добавяне на VHDL файл:

=> Използвайте **Project** -> **Add Files** от менюто. Така достигате до диалоговия прозорец (Фигура 16) където можете да изберете VHDL файлове. Файловете се добавят в реда, в който бъдат избрани.

=> Натиснете бутона **browse**, за да навигирате до проектната директория.

=> Изберете binctr.vhd файл-а и натиснете Add.

4. Натиснете OK в диалоговия прозорец Add Files to Project.

**Забележка** – За да добавите файлове от различна директория, натиснете бутона **browse**, за да навигирате до проектната директория.

I Files to Project			?
Files in the Project directory:		Files in the Projec	t
binetr.vhd	>> Add	binctr.vhd	
	<b>KK</b> Remove		
Add files by copying files from other direc	ctories;		Browse
You will be able to add files to the project Files'' menu item from the "Project" me	et at a later st enu.	age of your design b	y selecting "Add
	<	<u>B</u> ack <u>N</u> ext	> Cancel

Фигура 16 Диалогов прозорец за избор на файл за компилиране

### 2.3.3. Компилиране и синтезиране на Top-Level файл

В следващите страници, Вие ще стартирате Warp, за да генерирате JEDEC файл за специфична таргетна схема, в случая за CY7C37256 macrocell CPLD.

#### 2.3.4. Избиране на опции за компилиране

Опциите за компилиране се намират в менюто Project. Изберете Project -> Compiler Options и изберете таба Synthesis.

1	
Goal	I ech Mapping
C Area	Float Pins
• Speed	Float Nodes
Optimization Effort	F Retain XOR's
C None	Factor Logic
C Normal	🗖 Keep Polarity
	Disable Lateb Supthesis
• Exhaustive	Disable Laton Synthesis
Exhaustive	Global Low Power Mode
Exhaustive     I/0	Global Low Power Mode
Exhaustive      I/0      Disable Bus-Hold	Global Low Power Mode C D C T C Optimal
Exhaustive      I/0      Disable Bus-Hold      Voltage:	Global Low Power Mode C D C T Optimal Node Cost: 10 ▼
Exhaustive      I/0      Disable Bus-Hold      Voltage:          C 3.3V © 5.0V	Global Low Power Mode Choose FF Type CD CT Optimal Node Cost: 10 ▼
Exhaustive      I/0      Disable Bus-Hold      Voltage:          ① 3.3V ④ 5.0V      Default Slew Rate:	Global Low Power Mode Global Low Power Mode GDCT Optimal Node Cost: 10 ▼
<ul> <li>Exhaustive</li> <li>I/O</li> <li>☐ Disable Bus-Hold</li> <li>Voltage:</li> <li>○ 3.3V</li> <li>○ 5.0V</li> <li>Default Slew Rate:</li> </ul>	Global Low Power Mode Global Low Power Mode C D C T Optimal Node Cost: 10
<ul> <li>Exhaustive</li> <li>I/O</li> <li>Disable Bus-Hold</li> <li>Voltage:</li> <li>○ 3.3V</li> <li>○ 5.0V</li> <li>Default Slew Rate:</li> <li>○ Fast</li> <li>○ Slow</li> </ul>	Global Low Power Mode Global Low Power Mode CDCT Optimal Node Cost: 10
<ul> <li>Exhaustive</li> <li>I/O</li> <li>Disable Bus-Hold</li> <li>Voltage:</li> <li>© 3.3V</li> <li>© 5.0V</li> <li>Default Slew Rate:</li> <li>© Fast</li> <li>© Slow</li> <li>Unused Outputs:</li> </ul>	Global Low Power Mode Global Low Power Mode C D C T Optimal Node Cost: 10 Simulation Timing Model: Active-HDLSim/Active-VHDL

Фигура 17 Диалогов прозорец с опциите за компилиране и избран таб Synthesis.

• Настройка на неизползвани изходи

Warp Ви дава възожността да настроите всички неизползвани изходни изводи, които имат неизползвани макроклетки като '1'и, '0'и или 'Z'и. Това е глобална опция и не може да бъде отнесена към база синал по сигнал. Тази опция е полезна, за довеждането на всички не използвани изводи до определено логическо ниво.

=> Под Unused Outputs опциите от **Project -> Compiler Options -> Synthesis** таб, изберете **Z**, опциията по подразбиране за оставяне на всички неизползвани I/O изводи да имат три състояния.

Забележка – Когато използвате MAX340 EPLD и FLASH370 CPLD, е препоръчително да ползвате външни pull-ups за неизползваните I/O изводи.

• Избиране на Tech Mapping опции

Докато компилира регистрираните уравнения, fitter-ът използва директивата от Choose FF types, за да синтезира уравнения. Оставянето на опцията **Opt** избрана е най-добрият избор. Това позволява на fitter-а да избира между D-FF и T-FF имплементиране и тогава избира имплементацията, която използва наймалко ресурси.

=> От **Tech Mapping** опциите в таба **Synthesis**, изберете опцията **Optimal**.

Има още няколко други полезни опции от Tech Mapping options, които са обяснени в Galaxy online help, достъпни от **Galaxy Help** менюто.

• Избиране на Timing Model за Active-HDL Sim

Warp съдържа Active-HDL Sim timing simulator. Можете да проверите синтезирания проект, използвайки Active-HDL Sim, като създадете Active-HDLSim/ Active-VHDL времеви модел на проекта, използвайки Warp. Файлът binctr.vhd трябва да бъде компилиран с Active-HDLSim/Active-VHDL timing model, преди да бъде използван Active-HDL Sim.

=> Изберете Project -> Compiler Options -> Synthesis таб и натиснете Timing Model менюто в полето Simulation.

=> Изберете Active-HDLSim/Active-VHDL.

=> Натиснете бутона **OK** button , за да потвърдите избраното по-горе и затворете диалоговия прозорец **Compiler Options**.

• Избиране на Top-Level файл

След като необходимият VHDL файл е добавен към проекта, може да бъде избран top level файл. Само top-level файлове са реално синтезирани и в проекта е позволен само един top-level файл. => Натиснете binctr.vhd.

=> Изберете Project -> Set Top или натиснете бутона Set Top, достъпен от project toolbar.

=> След като top-level файл е избран, файловият символ в йерархията ще има символ за йерархия.

Забележка – За да изберете top level файл, файлът трябва да бъде избран в подпрозореца Project

drink1 [Device - CY37256P160	0-83AC] - Galaxy - [binctr.vhd]
Eile Edit View Format Project	<u>Compile Templates Bookmarks Tools Window Help</u>
0 🛎 🖬 🕼 🖁 🗸 🖻 🖻 🎒	│ A ¶ ¼ ∉ ∉ ≣ ≝
Source Files - Project: drink1	<pre>7 library IEEE; 8 use IEEE.std_logic_1164.all; 9 10 package binctr_pkg is 11 component binctr 12 port (reset, get_drink, clk: in std_logic; 13 give_drink: inout std_logic;) 14 empty: inout std_logic); 15 end component; 16 end binctr_pkg; 17 18 library IEEE; 19 use IEEE.std_logic_1164.all; 20 21 library CYPRESS; 22 use CYPRESS.std_arith.all; 23 use CYPRESS.lpmpkg.all; 24 entity binctr is 25 port (CLK: in STD_LOGIC; 26 get_drink: in STD_LOGIC; 27 reset: in STD_LOGIC; 28 empty: inout STD_LOGIC; 29 give_drink: inout STD_LOGIC); 30 end; 31 32 architecture binctr_arch of binctr is 33</pre>
🛅 So 🛛 🏹 Hi 🛛 🛄 Ou	binctr.vhd

Фигура 18 Избор на top-level файл

• Компилиране и синтезиране на файл

=> Във вашия Galaxy прозорец изберете **Compile -> Project**, за да започнете компилирането или натиснете бутона **Compile Project** от project менюто. Warp започва компилацията и синтеза на дизайна върху CY7C37256 и извежда съобщения, за да Ви информира за напредъка в подпрозореца за резултати. Опцията **Compile - > Project** рекомпилира автоматично само тези файлове, които са модифицирани от последната компилация.

<ul> <li></li></ul>	8 use IEEE.std_logic_1164.all;
P binctr_pkg     e binctr     a binctr_arch     Rom Files	<pre>9 10 package binctr_pkg is 11 component binctr 12 port (reset, get_drink, clk: in std_log 13 give_drink: inout std_logic; 14 empty: inout std_logic); 15 end component;</pre>
	16 end binctr pkg;
	17
	18 library IEEE;
	19 use IEEE.std_logic_1164.all;
	20
	21 library CYPRESS;
	22 use CYPRESS.std_arith.all;
	23 use CYPRESS.lpmpkg.all;
	24 entity binctr is
	25 port (CLK: in STD_LOGIC;
	26 get_drink: in STD_LOGIC;
	27 reset: in STD_LOGIC;
	28 empty: inout STD_LOGIC;
	29 give_drink: inout STD_LOGIC);
	30 end;
	31
	32 architecture binctr_arch of binctr is
10 March 10	33
🖹 <b>So</b> 🔼 Hi 🌆 Ou	binctr.vhd
WARP done.	
Compilation successfu	1.
genvhdl -s 1164 VHDL	-i "binctr.vhd"
Running: disasm-v -f	-lstd_logic binctr
genvhdl completed	40 - 19 - 5 San (19 - 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 -

```
Done.
```

Фигура 19 Прозорец за резултати показва успешно компилиране.

Тази операция генерира два файла от особен интерес:

• binctr.jed: използва се за програмиране на схемата СY37256.

• <u>binctr.rpt</u>: съдържа изходна (*pinout*) и времева (*timing*) информация, заедно с друга информация относно финалния синтезиран проект. Можете да видите .rpt файла и изходните файлове, създадени от Warp като натиснете таба изглед на картина (output view) от подпрозореца Project.

Различни секции от report файл-а могат да бъдат достигнати като натиснете знака плюс и изберете наличните секции.

Забележка – Ако се появи грешка при компилирането, убедете се, че текстът на Вашия binctr.vhd файл е въведен точно както е показано по-рано в тази глава -- или, копирайте файла от <warp path>\examples\your name\vhdl directory – и после стартирайте Warp отново.

Ако грешки се появят в подпрозореца за резултат:

=> Натиснете таба *Errors and Warnings*, намиращ се отдолу на прозореца за резултати Galaxy output.

=> Натиснете два пъти върху съобщението за грешка, за да го изберете.

=> Отидете в прозореца за редактиране VHDL editor window. Курсорът трябва да се намира на реда, който е предизвикал грешката. Използвайте бутоните за грешка Next и Previous, за да локализирате други грешки.

#### 2.3.5. Симулиране на поведението на проекта

В това ръководство Вие ще изпълните следните стъпки:

- Стартиране на Active-HDL Sim.
- Отваряне на файл-а binctr.vhd file в папка your name/soda machine/vhd/.
- Определяне на of the стимулиращи сигнали (stimulus signals) в симулацията.
- Симулиране на проекта.

• Преглед и анализ на резултатите с цел да се установи какво се е случило.

- Стартиране на Active-HDL Sim
- => Стартирайте Active-HDL Sim като изберете Tools -> Active-HDL Sim.



Фигура 20 Избиране на Tools -> Active-HDL Sim

=> Отворете файла binctr.vhd file като изберете File -> Open VHDL и потърсите папка your name/soda machine/vhd.

=> Натиснете **binctr.vhd**.

=> Натиснете **ОК**. Прозорецът Active-HDL Sim трябва да прилича на фигура 21.

File Search View D	esign Simulation	eform3 Waveform Tools Help
≌ 📽 🖬   🕱		▶ ▶I ▶ 100ns 🔆 📢 No simulation
]% <b>B B</b>   10 9	> ] 🖟 Q ↔	<u>ଭ</u>   ବ୍ ବ୍ ବ୍   ୩୦୦୦   ୬୦୦୦୦   ୫୫.୫   ୫୦୦୦   ୫୦୦୦
Name	Value Sti	🗽 т 20 т 40 т 60 т 80 т 100 т 120 т 140 т 160 т 180 т 20
🏾 🕱 waveform3 /		4
<pre>&gt; Design: 14 &gt; Design: Opd &gt; Compilation &gt; File: C:\Ma &gt; Compile End &gt; Compile Ard &gt; Top-Level &gt; Entity =&gt; &gt; Compile sud &gt; ELBREAD: E. &gt; ELBREAD: E.</pre>	23, 24 Àïðè ening design n with file ariya Damyan tity "binctr chitecture " unit detecte binctr ccess O Erro laboration p laboration t	ë 2009 ä. ""C:\Mariya Damyanova\soda machine s HDL FSM\vhd\b: reorder ova\soda machine s HDL FSM\vhd\binctr.vhd " DSMB" of Entity "binctr" d rs 0 Warnings Analysis time : 0.0 [s] rocess. ime 0.0 [s].

Фигура 21 Първоначален прозорец Active-HDL Sim за binctr.vhd.

• Задаване на входните източници и на сигналите за изчертаване върху времедиаграмата

=> Изберете Waveform -> Add Signals.

=> Натиснете два пъти **clk** в десния прозорец на диалоговия прозорец **Add Signals**, показан на фигура 22.

/	1	92 object(s)		
≓ 💶 Root :	~	Name	Value	^
🕀 🔁 INST_0 : c37koreg	=	₽- clk	Unavailable	
🕀 🤹 INST_1 : c37koreg		🗢 empty	Unavailable	
INST_2: c37koreg		₽- get_drink	Unavailable	
∎		🗢 give_drink	Unavailable	
+ + INST_4: c3/koreg		D- reset	Unavailable	
		<sup>™</sup> curstateSBV_I	) Unavailable	
		<sup>™</sup> curstateSBV_	l Unavailable	
		ז jed_node2486	Unavailable	
🛨 🔁 INST_9:c37kmux	~	™ jed_node2492	Unavailable	~

Фигура 22 Диалогов прозорец Add Signals с налични сигнали за създаване на изглед.

=> Натиснете два пъти следните сигнали, за да ги добавите към Вашия нов изглед в следната последователност:

reset, get\_drink, get\_diet, clk, empty, give\_drink.

=> Натиснете Add.

**Забележка** – Сигналите в лявата част на диалоговия прозорец Add Signals dialog box ще са различни в зависимост от избраните пакет и фамилия схеми.

• Задаване на параметрите на входните източници (Stimulus Signal)

Трябва да определите стойностите на следните входни източници: clk, reset, get\_drink.

Задайте clk сигнала за равно разпределени, редуващи се високи и ниски стойности.

=> Изберете сигнала clk в лявата част на прозореца waveform.

- => Задръжте бутона Ctrl, докато избирате reset и get\_drink.
- => Натиснете с десен бутон и изберете опцията Stimulators.

=> Натиснете clk в рамката Signals frame.

=> Изберете "Clock" като Stimulator type от менюто.

=> Във времевата диаграма натиснете върху една от левите страни на подпрозореца. Полето с 1 трябва да стане светло сиво, за да индикира, че е избрано. Погледнете фигура 23.

ignals:	Stimulator type:	Strength:
Name C	Clock	Override
□ reset □ get_drink	0 Fre	• 100ns • 50 %
 Display paths		

Фигура 23 Времева диаграма за дефиниране на входния източник clk, 1 е избрано.

=> Натиснете бутона **Арріу**.

Дефинирайте reset да се покачва за определено време. За да го направите:

- => Изберете сигнала reset от подпрозореца Signals.
- => Изберете "Formula" като Stimulator type от менюто.
- => Въведете следното в полето "Enter Formula":
- =>0 0, 1 75 ns, 0 125 ns
- => Натиснете бутона **Арр**у.

Дефинирайте get\_drink да се покачва за четири непоследователни периода.

- => Изберете сигнала get\_drink от подпрозореца Signals.
- => Изберете "Formula" като Stimulator type от менюто.
- => Въведете следното в полето "Enter Formula":

=> 0 0, 1 175 ns, 0 225 ns, 1 375 ns, 0 425 ns, 1 575 ns, 0 625 ns, 1 775 ns, 0 825 ns

=> Натиснете бутона **Apply**.

Дефинирайте reset да се покачва за един покачващ се период след последната get\_drink заявка.

=> Изберете signala reset от подпрозореца Signals.

=> Изберете "Formula" като Stimulator type от менюто.

=> Добавете следния текст към формулата след 0 125 ns:

=>00, 1 1675 ns, 0 1725 ns

=> Натиснете бутона **Apply**.

Дефинирайте get\_drink да се покачва за един покачващ се период, респ., след втория reset.

=> Изберете сигнала get\_drink от подпрозореца Signals.

=> Добавете следния текст към формулата след 0 825 ns:

=>00, 1 1775 ns, 0 1825 ns

=> Натиснете бутона **Apply** и след това **Close**.

Когато е завършено, резултатът трябва да изглежда като фигура 24.

Signals Hotkeys Predefined		? ×
Signals: Name Type ✓ clk Clock ✓ reset Formula ✓ get_cola Formula ✓ get_diet Formula	Stimulator type: Formula Enter formula: 0 0, 1 975 ns, 0 1025 n Accept Active-CAD format.	Strength: Override
Display paths	Apply	
		Close

Фигура 24 Диалогов прозорец Active-HDL Sim stimulators за **binctr.awf** с нагласени всички сигнали

• Стартиране на симулацията

=> За да симулирате проекта, натиснете менюто до полето 100ns на лентата с инструменти и натиснете бутона **Run For**.

Забележка – За да реинициализирате симулацията, натиснете бутона Restart Simulation на лентата с инструменти или го изберете от менюто Simulation -> Restart Simulation и изберете Waveforms -> Clear all Waveforms.

Забележка – Може да искате да промените резолюцията на монитора, за да вместите всички дейности по функциите на един екран. Изберете View -> Zoom -> Out.

Резултатите трябва да изглеждат подобно на фигура 25:

\$ <b>₽</b>  % X <b>B B</b>  ¢		
ame	Value	Sti 1 200 1 400 1 600 1 801810 ns 100 1 1200 1
₽- reset	0	Fo
₽- get_drink	1	
¤- clk	1	
🗢 empty	1	
🗢 give_drink	0	
waveform3	/	

Фигура 25 Резултати от симулацията на binctr.vhd в Active-HDL Sim

Симулацията стартира, като машината за напитки е празна. Забележете състоянието на сигнала empty при стартирането на симулацията.

Когато сигналът reset се покачи в началото на симулацията, сигналът empty намалява. Сега машината е готова да дава питиета.

Машината дава три напитки в отговор на първите три заявки за питиета. Забележете връзката между пулсовете на сигналите get\_drink и give\_drink. След следващата заявка за питие, машината обаче не дава питие; отделението за питиета е празно.

Когато две от напитките в отделението вече липсват, сигналът empty се покачва. Той остава висок, докато сигналът reset отново се покачи, казвайки на машината, резервоарите са попълнени. Следващата заявка е изпълнена.

Забележка – Ако резултатът от симулацията Ви не е верен, уверете се, че сигналите Ви започват и свършват наполовина на качващите се и падащите периоди.