3.3.) Програмиране на платката с програмата ISR (In-System-Reprogrammable)

Използваната развойна система позволява многократно пре-програмиране на схемите. Най-напред платката е свързана към компютъра чрез кабела за програмиране.



Фигура 5 Кабел за програмиране C3ISR.02

Кабелът за програмиране трябва да не е свързан, преди включването на захранващото устройство. Фигура 5 показва използвания кабел за програмиране.

Подаване на напрежение към платката

Захранващата клема приема 2,5 мм женски контакт. Може да се използва 5волтов преобразувател на напрежение, за да захранва Delta39K ISR прототипната платка. Преобразувателят трябва да разполага със съответен женски контакт или да бъде снабден отделно с такъв от магазин за електроника. Ние сме използвали специализирао захранване за програмируеми схеми. Захранващата клема е обозначена на платката като: V_{CC}. Лампичката на име DS1 показва кога има ток към V_{CC}. И двете схеми Delta39K и Ultra37000 оперират с 5 волтово захранване, което трябва да бъде доставено чрез V_{CC}.

3.3.1. Стартиране на програмата ISR

Стартирайте софтуера чрез избиране на *Start -> Programs -> Cypress -> ISR Programming Software Version 3.0 -> ISR Programming Software* от Windows десктопа.

Фигура 24 показва изглед на програмата ISR.



Фигура 26 Потребителски интерфейс на програмата ISR

В горната част на прозореца е разположена заглавната лента **Title Bar**. Точно под заглавната лента е меню лентата **Menu Bar**, която изброява имената на всички менюта. Кликайки с левия бутон на мишката върху име на меню, се показват всички подменюта, достъпни от това меню. Лентата с инструменти **Tool Bar** се намира под меню лентата. Лентата с инструменти Ви позволява да изберете най-често използваните команди като натиснете бутона.

Главният прозорец е под лентата с инструменти. Този прозорец е разделен на две части чрез Разделителната лента, Window Splitter Bar. Можете да промените относителната гоелмина разделените прозорци като влачите разделителя нагоре и надолу с левия бутон на мишката. Горната част на разделения прозорец изобразява веригата (daisy chain) като мрежа от клетки, а долната част е прозорецът, показващ текущия резултат. Прозорецът за резултати показва информация, предупреждения и съобщения за грешки. Можете да влачите хоризонталната и вертикалната лента за позициониране по някой от прозорците. Веригата от устройства(о) е представена чрез мрежа от клетки. Всяка редица в мрежата съответства на едно устройство. Във всяка редица има пет колони от клетки, а именно: Позиция на схемата (Device Position), Схема (Device), Операция (Operation), Име на файл (Filename), Потребителски код (Hex) (User Code (Hex)) и компресия на данни (Data Compression). Колоните са обозначени в първата (заглавна) редица. Клетките в първата колона описват относителната позиция на устройството във веригата. Втората колона е наречена Схема (Device). Кликайки с левия бутон на мишката върху клетка от колоната **Device** изскача кутия с комбинации, от която потребителят може да избере целевото устройство. По подобен начин, кликайки с левия бутон на мишката върху клетка от колоната **Operation**, се появява избирателна кутия, която изброява всички възможни операции за конкретното устройство. Можете да изберете всяка една от предложените операции. Четвъртата колона се нарича **Program Filename**.

Ако изберете за устройството някоя от следните операции Program & Verify, Verify Pattern или Read Device, трябва да посочите съответен JEDEC или Нех файл в тази колона.

Всяка клетка в колоната **Program Filename** е поле, което може да се редактира и позволява да въведете пълното име и път до JEDEC или Нех файла. Кликането върху полето в колоната

User Code (Hex) column активира редактируемото поле и показва стойността на потребителския код, която се съдържа във входящия файл, ако има такъв. В редактируемото поле може да бъде въведена различна стойност, за да бъде програмирана в схемата. В тази клетка също така има и кутийка за отметка (check box) с етикет Lock. Ако кутийката не е отметната, ISR апликацията ще програмира устройството с потребителския код, съдържащ се в програмния файл, в противен случай се използва потребителския код в клетката. Ако кутийката не е отметната и ако няма потребителския код в програмния файл, потребителският код ще бъде програмиран като нулева стойност. Кутийката за отметка (checkbox) в колоната Data Compression води до съставянето на STAPL файл, използвайки ACA компресия при писане на данни в STAPL файла.

Забележка – Трябва да изберете устройство, преди да можете да видите списъка с налични операции в съответната клетка от колоната Operation.

Първо попълваме полето за име и път до програмния файл **Program Filename.** Отваря се диалогов прозорец като на фиг.27, в който избираме .jed файла.

Look jn: i demultiplexor 02-06-2009 i demultiplexor i c37256v i screenshots i vhd i demultiplexor.jed	
demultiplexor lc37256v screenshots vhd demultiplexor.jed	
ic37256v screenshots vhd demultiplexor.jed	
screenshots vhd demultiplexor.jed	
→ vhd → demultiplexor.jed	
demultiplexor.jed	
File name: demultiplexor.jed	OK
Files of human All Deserves Files (% is d.% have)	

Фигура 27 Избиране на Program File

3.3.2. Избиране на настройки за програмирането на платката

След като заредим програмния файл demultiplexor.jed, автоматично се попълва и полето от колоната Device, както е показано на фиг.28.

🔑 Unti	itled - Isr					
Eile Edit View Operations Iools Help						
	Device	Operation	Program Filename	User Code (Hex)	Data Compression	
1	CY37256VP160-66AC		C:\Mariya Damyan 🇨	0000 🗖 Lock	П ОИ	
2				🗖 Lock	П ON	
3				🗖 Lock	П ON	
4				🗖 Lock	П ON	
5				🗖 Lock	🗖 ON	
Ready						

Фигура 28 ISR прозореца след избиране на програмния файл

Попълваме и останалите полета в първата редица, както е показано на фиг.27. Като операция избираме **Program & Verify** – програмира и проверява схемата, зададена в Jedec файла, върху схемата.

Останалите операции, които програмата ISR поддържа за Ultra37000 схеми са:

- Verify Pattern — проверява програмираното устройство със схемата специфицирана в JEDEC файла.

- **Bypass Device** – Тази операция поставя устройството в bypass състояние (устройството не бива взето под внимание).

- Erase Device – Изтрива всички съединиени места в устройството (Erase all fuse locations in the device).

- Read Silicon ID – прочита силиконовото ID от устройството.

- Read User Code – Прочита 8-битовия потребителски идентификационен код ID от програмираното устройство.

- **Program Security Bit** – програмира бита за сигурност (security bit) на схемата. След като битът за сигурност е програмиран, проверяването, четенето и програмирането на бита за сигурност са деактивирани, но четенето на силиконовото ID, четенето на потребителския код и други функции все още са позволени.

- **Read Device** - Тази операция прочита съдържанието на схемата и записва JEDEC файла.

Въвеждаме потребителски код – **10FF**. Слагаме отметки на кутийките Lock и ON от полетата User Code (Hex) съотв. Data Compression.

🐴 Unti	tled - Isr					
Eile Edit View Operations Iools Help						
	Device	Operation	Program Filename	User Code (Hex)	Data Compression	
1	CY37256VP160-66AC	Program & Verify	C:\Mariya Damyan	10FF 🔽 Lock	N 🔽 N	
2				🗖 Loci	I ON	
3				🗖 Lock	N 🗖 N	
4				🗖 Loci	I ON	
5				🗖 Lock	C ON	
Ready						

Фигура 29 ISR прозореца с попълнените полета

След като всички полета са попълнени, от меню лентата избираме **Operations -** > **Compose**.

Операцията **Compose** потвърждава описанието на веригата и записва STAPL файл(ове) за всички схеми (в случая една). STAPL файловете се наричат

Filename_DEV#.STP, където # е позицията на избраното устройство във веригата. За всеки ISR файл се създава допълнителен STAPL файл Filename_VRFY.STP. Файлът се използва, за да изпраща запитвания на устройството(устройствата) на платката и да потвърди описанието на веригата. Всичките изходни файлове са създадени в същата директория, където се намира и ISR файлът.

Появява се прозорецът, показан на фиг.30

lsr	
1	Please save the project file before you select the Compose operation.

Фигура 30 Прозорец след изпълнението **Operations -> Compose**.

От меню лентата избираме File -> Save и съхраняваме файла като Demux_prog.isr.

След това отново избираме **Operations -> Compose**.

3.3.3. Резултати от програмирането на платката

Резултатът е изобразен на фиг.31.

🕫 Dem	ux_prog.isr - lsr						- 6 🛛	
Ele Edit View Operations Icols Help								
D 🖙 🖬 🥌 👗 🛍 🛍 🔯 🕼 s 🎗								
	Device	Operation	Program Filename	User Code (Hex)	Data Compression			
1	1 CY37256VP160-66AC Program & Verify C:\Mariya Damyan 10FF 🔽 Lock 🔽 ON							
2	2 FLock FON							
З	3 ELock EON							
4	4 FLock FON							
5	5 FLock FON							
Generating configuration file Warning: Row 2 - Device not selected. Skipping this device row. Warning: Row 4 - Device not selected. Skipping this device row. Warning: Row 5 - Device not selected. Skipping this device row. Warning: Row 5 - Device not selected. Skipping this device row. Error(s) 0. Warning(s) 4 Calling Composer C: Program Files/Cypress/ISR Programming Software Version 3.0.7/bin/compose.exe "Demux_prog.cfg" STAPL Composer Version 3.0.3.3 Copposing STAPL (1) (C) 2000-2001 Cypress Semiconductor Reading configuration file 'Demux_prog.cfg' SystEM: Composing file 'C: Mariya Damyanova/demultiplexor 02-06-2009/Demux_prog_vrfy.stp' Row 1: Composing file 'C: Mariya Damyanova/demultiplexor 02-06-2009/Demux_prog_dev1.stp' Composer created 2 stapl file(s) successfully.								
Ready								

Фигура 31 Резултати

След успешното създаване на STAPL файла(файловете), операцията *Operations->Play* ще стане активна. Изберете я, за да изпълните определените операции на всички схеми във веригата. Изходните съобщения са изобразени в прозореца за резултати, както и са записани във файла *Filename*.LOG.