Проектиране на вградени автомобилни електронни системи
Лабораторно упражнение №5

Работа с Xilinx Vivado и Vitis. Синтезиране на микропроцесорна система върху FPGA. Работа с I^2C модул. Мултиплексиране на изводите с Constraints файл.

==========

1. Превключете джъмпера вдясно на платката на позиция JTAG. Свържете µUSB кабел към PROG/UART USB куплунга. Включете платката от ключа ON/OFF.

2. От страничния панел на Ubuntu изберете бутон “Show Applications”, след което в полето “Type to search” напишете Vivado и натиснете с ляв бутон на мишката иконката на програмата.

3. Create Project → Next → Project name: 05_i2c → Next → RTL Project + "Do not specify sources at this time" → Next → таб Boards: избира се Zybo (не Zybo Z7-10, не Zybo Z7-20, а само Zybo) → Next → Finish

ЗАБЕЛЕЖКА: работната маса с платка Zybo Z7-10 (вдясно на Етернет куплунга трябва да има 2 HDMI конектора; ако има един HDMI и един VGA значи, че е само Zybo) трябва да избере Zybo Z7-10 от това меню.

4. Вляво → Flow navigator → Create block design → OK

5. Вдясно → Diagram → right-click → Add IP → Search → ZYNQ7 Processing System → double click

8. Вдясно → Diagram → натиска се и се задържа ляв бутон върху FCLK_CLK0 сигнала и се свързва с M_AXI_GP0_ACLK, след това се пуска левия бутон.

9. Вдясно → Diagram → right-click → Add IP → Search → Processor System Reset → double click

10. Вдясно → Diagram → right-click → Add IP → Search → AXI IIC → double click

11. Щраква се два пъти върху блока “ZYNQ7 Processing System” → в “Page navigator” се отива на раздел “Interrupts” → слага се отметка на “Fabric interrupts” → в подраздела “PL-PS Interrupt Ports” се слага отметка на “IRQ_F2P [15:0]” → OK

12. В същия прозорец, отива се в “Page Navigator” → MIO Configuration → слага се отметка на I/O Peripherals → UART1 и се проверяват връзките MIO48 ↔ tx, MIO49 ↔ rx → OK

13. Свързва се “interrupt” изхода на I2C модула (iic2intc_irpt) с входа на ZYNQ7 (IRQ_F2P).
Натиска се ляв бутон върху извода и се задържа, докато се тегли връзката. Когато се свърже с другия пин, трябва да се пусне левия бутон.

14. Вдясно →Diagram → зелена лента → Designer Assitance available → Run Block Automation → Слага се отметка на "All Automation".

15. Вдясно →Diagram → зелена лента → Designer Assitance available -> Run Connection Automation → Слага се отметка на "All Automation".
Избира се axi_iic_0 / IIC → вдясно се избира Options → Select board part interface → падащо меню → Custom.
Натиска се OK.

16. Щраква се двукратно върху ZYNQ7 Processing System → Page Navigator → MIO Configuration → I/O Peripherals → маха се отметката от Етернет модула ENET0.

17. Подрежда се блоковата схема с бутон Regenerate Layout.

18. Вдясно→Diagram→лента с бутони → Validate Design (F6) → "Validation successful. There are no errors or critical warnings in this design." → OK

19. Централно→в Block design прозореца, натиска се таб-а Sources → Design sources → right-click на design_1.bd → Create HDL Wrapper (създава VHDL описание на новосъздадената система) → Let Vivado manage wrapper and auto-update → OK

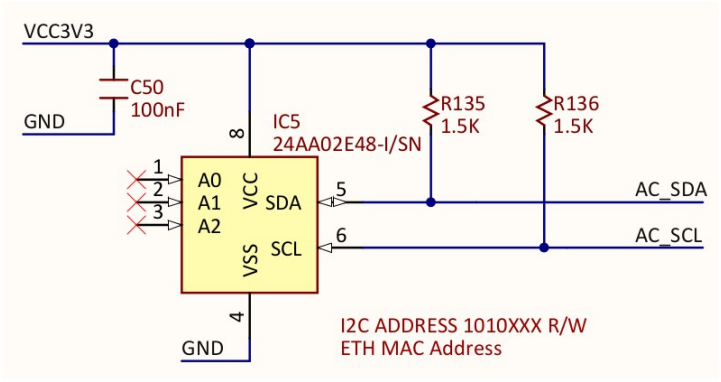
20. File → Add Sources → Add or create constraints → Next → Add files → укажете пътя до 00_ZYBO_Master.xdc (има го в директорията на настоящото лабораторно) → OK → сложете отметка на “Copy constraints files into project” → Finish.

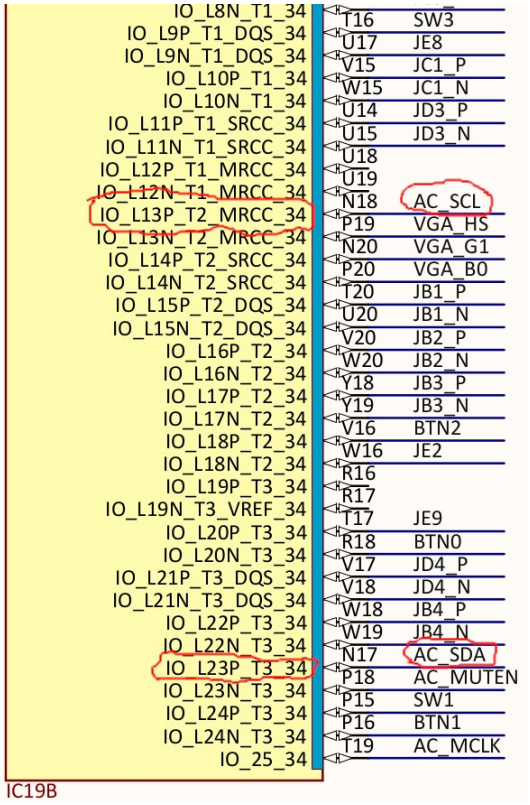
21. Горе, вляво → таб Sources → Constraints → constrs_1 → щракнете двукратно върху 00_ZYBO_Master.xdc, за да се отвори constraints файла. В него се описват връзките между вътрешните сигнали на FPGA и изводите на корпуса на FPGA.

Гледайки принципната схема на ZYBO, трябва да се проследят връзките на I2C EEPROM паметта с изводите на FPGA.

В constraints файла се откоментират следните редове:

#Audio Codec/external EEPROM IIC bus
#IO_L13P_T2_MRCC_34
set_property PACKAGE_PIN N18 [get_ports ac_scl]
set_property IOSTANDARD LVCMOS33 [get_ports ac_scl]
#IO_L23P_T3_34
set_property PACKAGE_PIN N17 [get_ports ac_sda]
set_property IOSTANDARD LVCMOS33 [get_ports ac_sda]





след което се натиска Ctrl + s от клавиатурата, за да се запазят промените.

ВНИМАНИЕ! Имената на сигналите в дадения constraints файл са примерни (ac_scl, ac_sda). Реалните имена могат да се видят от Design Sources → design_1_wrapper → design_1_i → двукратно щракване върху design_1.v

Пример – ако в wrapper-а са записани следните сигнали:

```
module design_1
   (DDR_addr,
    DDR_ba,
    DDR_cas_n,
...
```

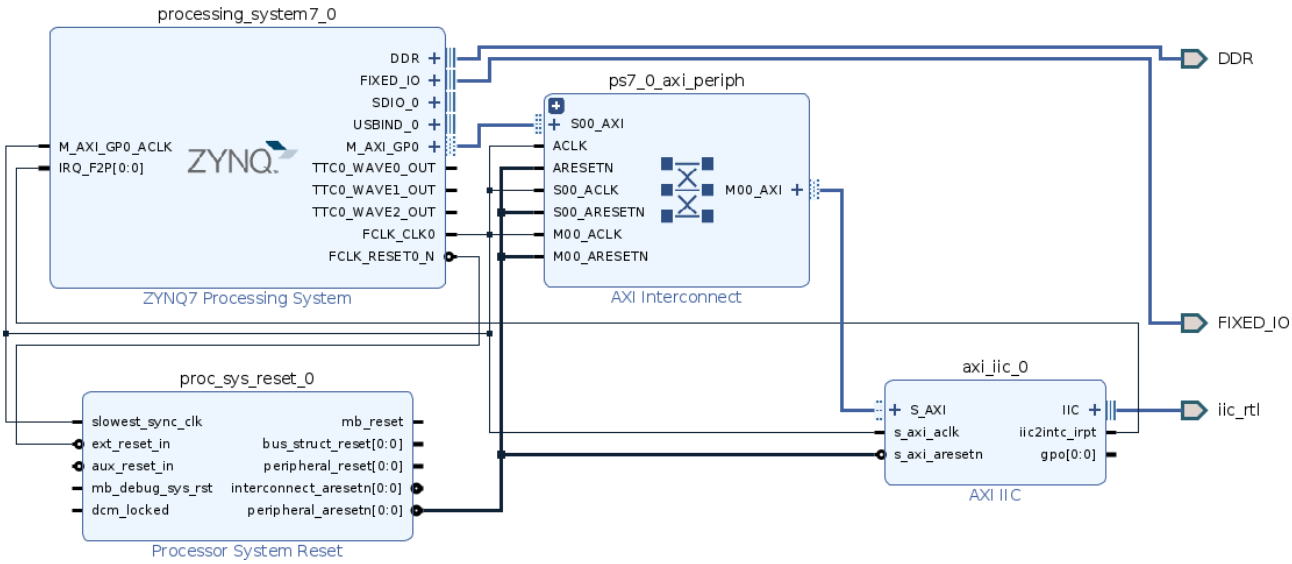
```
    iic_rtl_scl_i,
    iic_rtl_scl_o,
    iic_rtl_scl_t,
    iic_rtl_sda_i,
    iic_rtl_sda_o,
    iic_rtl_sda_t);
...
```

в constraint файла трябва да се запише:

```
#Audio Codec/external EEPROM IIC bus
#IO_L13P_T2_MRCC_34
set_property PACKAGE_PIN N18 [get_ports iic_rtl_scl_io]
set_property IOSTANDARD LVCMOS33 [get_ports iic_rtl_scl_io]

#IO_L23P_T3_34
set_property PACKAGE_PIN N17 [get_ports iic_rtl_sda_io]
set_property IOSTANDARD LVCMOS33 [get_ports iic_rtl_sda_io]
```

Получената блокова схема трябва да изглежда така:



21. Вляво → Flow navigator → Generate bitstream → Yes → OK → изчаква се няколко минути (докато завърши синтеза) → View reports → OK

ВНИМАНИЕ: долу, централно, в таб Log може да наблюдавата съобщенията от синтеза. Най-горе, вдясно на Vivado прозореца ще видите иконка на въртящ се зелен часовник. Докато тя е видима, значи трябва да се изчака.

22. File → Export → Export hardware → Next → Include bitstream → Next → Next

→ Finish

==========

==========

23. От страничния панел на Ubuntu изберете бутон “Show Applications”, след което в полето “Type to search” напишете Vitis Classic и натиснете с ляв бутон на мишката иконката на програмата.

24. Избира се път до workspace за фърмуерния проект→ Launch

ВНИМАНИЕ: възможно е да има останали фърмуерни проекти от минали групи. В таб-а Explorer на средата Vitis със задържане на CTRL от клавиатурата изберете с ляв бутон на мишката всички проекти, след което натиснете десен бутон на мишката и Delete. Ако проектите ще се използват, махнете отметката от “Delete project contents on disk (cannot be undone)” и натиснете OK.

25. File → New → Platform project → Platform project name: 05_i2c → Next → таб "Create new platform from hardware" → Browse → избира се пътя до проекта 05_i2c, създаден от Vivado → design_1_wrapper.xsa → Open → Finish

26. Вляво→Project explorer → избира се 05_i2c → right-click → Build Project

27. File → New → Application project → Next → "Select a platform from repository" → Избира се 05_i2c → Next → Application project name: 05_i2c_app → Next → Next → “Hello World” → Finish

28. Щраква се двукратно с ляв бутон върху директорията src в проекта 05_i2c_app_system/05_i2c_app → src → helloworld.c

29. В текстовия редактор на Vitis се въвежда следната програма [1]:

```
#include <stdio.h>
#include <string.h>
#include "platform.h"
#include "xil_printf.h"
#include "sleep.h"

#include "xparameters.h"
#include "xscugic.h"
#include "xil_exception.h"
#include "xiic.h"

#define EEPROM_ADDRESS        0x50 //7-bit address, 1010 000

XScuGic INTC0;
XIic I2C0;

u8 receiving_data;
u8 transmitting_data;
```

```c
void i2c_send_handler(XIic *instance_ptr){
	transmitting_data = 0;
}

void i2c_receive_handler(XIic *instance_ptr){
	receiving_data = 0;
}

void i2c_status_handler(XIic *instance_ptr, int event){

}

int main(){
	u8 read_buffer[256];
	u8 cmd_buffer[16];
	XScuGic_Config *intc_config;
	XIic_Config *iic_config;

	init_platform();

	xil_printf("Starting the I2C EEPROM example ...\n\r");

	intc_config = XScuGic_LookupConfig(XPAR_PS7_SCUGIC_0_DEVICE_ID);
	XScuGic_CfgInitialize(&INTC0, intc_config, intc_config->CpuBaseAddress);
	XScuGic_SetPriorityTriggerType(&INTC0, XPAR_FABRIC_IIC_0_VEC_ID, 0xA0,
0x3);
	XScuGic_Connect(&INTC0, XPAR_FABRIC_AXI_IIC_0_IIC2INTC_IRPT_INTR,
(Xil_ExceptionHandler) XIic_InterruptHandler, &I2C0);
	XScuGic_Enable(&INTC0, XPAR_FABRIC_AXI_IIC_0_IIC2INTC_IRPT_INTR);
	Xil_ExceptionInit();
	Xil_ExceptionRegisterHandler(XIL_EXCEPTION_ID_INT, (Xil_ExceptionHandler)
XScuGic_InterruptHandler, &INTC0);
	Xil_ExceptionEnable();

	iic_config = XIic_LookupConfig(XPAR_IIC_0_DEVICE_ID);
	XIic_CfgInitialize(&I2C0, iic_config, iic_config->BaseAddress);
	XIic_SetSendHandler(&I2C0, &I2C0, (XIic_Handler) i2c_send_handler);
	XIic_SetRecvHandler(&I2C0, &I2C0, (XIic_Handler) i2c_receive_handler);
	XIic_SetStatusHandler(&I2C0, &I2C0, (XIic_StatusHandler)
i2c_status_handler);
	XIic_SetAddress(&I2C0, XII_ADDR_TO_SEND_TYPE, EEPROM_ADDRESS);

	memset(read_buffer, 0x00, 256);

	receiving_data = 1;
	transmitting_data = 1;

	cmd_buffer[0] = 0xfa;
	XIic_Start(&I2C0);
	XIic_MasterSend(&I2C0, cmd_buffer, 1);
	while ((transmitting_data) || (XIic_IsIicBusy(&I2C0) == TRUE)) { }
	XIic_Stop(&I2C0);

	XIic_Start(&I2C0);
	XIic_MasterRecv(&I2C0, read_buffer, 2); //This API can't receive a single
byte, so get 2
	while ((receiving_data) || (XIic_IsIicBusy(&I2C0) == TRUE)) { }
	XIic_Stop(&I2C0);
```

```
	xil_printf("Eeprom read data: %02X\n", read_buffer[0]);

	while(1){ }

	cleanup_platform();

	return 0;
}
```

30. Вляво, Project explorer -> избира се 05_i2c_app_system -> right-click -> Build project

31. Вляво, Project explorer -> избира се 05_i2c_app_system -> right-click -> Debug as -> Launch Hardware

32. Отваря се терминал в Ubuntu с CTRL + ALT + T → Пише се ls /dev/tty и се натиска tab → "Display all 100 possibilities? (y or n)" въвежда се 'y' → **търси се системния файл, отговарящ на виртуалния RS232 порт** за дебъг съобщения (обикновено ttyUSB1, ВНИМАНИЕ на ttyUSB0 излиза виртуален порт за JTAG дебъгера, който не трябва да бъде отварян).

След като се види номера на виртуалния порт, в същия терминал се стартира RS232 терминал чрез командата

cutecom

33. В cutecom → Device: избира се съответния порт за дебъг съобщения /dev/ttyUSBx → Settings → 115200-8-N-1, no flow control -> Open

34. Във Vitis: натиска се бутон Resume (F8). След това в Cutecom трябва да се изпише:

```
Starting the I2C EEPROM example ...
[19:12:39:271] Eeprom read data: D8
```

35. За да спрете debug сесията във Vitis, натиснете Disconnect

36. Напишете програма, която прочита целия EUI-48 уникален номер, записан в ROM клетки на EEPROM паметта 24AA02E48.

37. Напишете програма, която прочита цялата EEPROM памет и принтира данните в следния формат:

```
[19:04:01:252] Starting the I2C EEPROM example ...
[19:04:01:252] Eeprom read data:
[19:04:01:268]
[19:04:01:268] 0x00: DF 4B FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:284] 0x10: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:284] 0x20: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:284] 0x30: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:300] 0x40: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:300] 0x50: FA FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:300] 0x60: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:300] 0x70: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:316] 0x80: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:316] 0x90: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:316] 0xA0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:332] 0xB0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:332] 0xC0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:332] 0xD0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:348] 0xE0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
[19:04:01:348] 0xF0: FF FF FF FF FF FF FF FF FF FF D8 80 39 5E 6B A1
```

* * *

[1] https://xilinx.github.io/embeddedsw.github.io/iic/doc/html/api/index.html

доц. д-р инж. Любомир Богданов, 2024 г.