Проектиране на вградени автомобилни електронни системи
Лабораторно упражнение №8

Работа с Xilinx Vivado и Vitis. Синтезиране на микропроцесорна система върху FPGA. Работа с Етернет модул.

==========

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: 08_eth → 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.

6. 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.

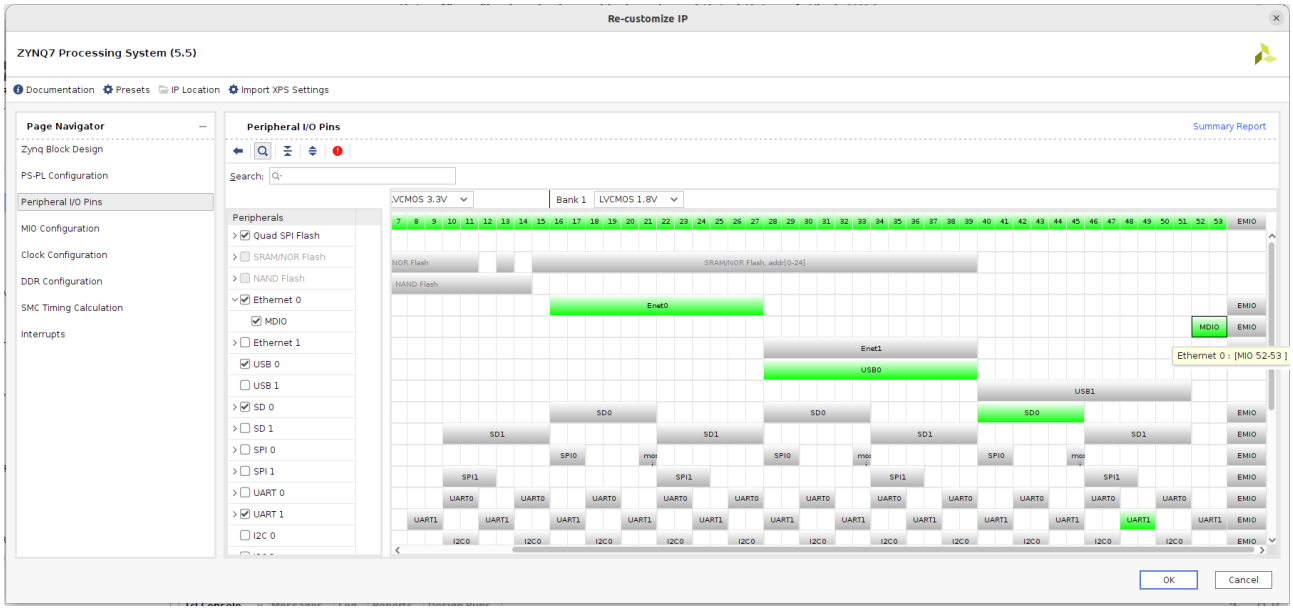
7. Вдясно → Diagram → right-click → Add IP → Search → Processor System Reset → double click.

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

9. Вдясно →Diagram → зелена лента → Designer Assitance available -> Run Connection Automation → Слага се отметка на "All Automation". Натиска се OK.

10. Щраква се два пъти върху блока “ZYNQ7 Processing System” → в “Page navigator” →Peripheral I/O pins → натиска се стрелката на Ethernet 0 →

проследява се редът на MDIO → вдясно е избрано EMIO → **избира се MDIO → OK.**



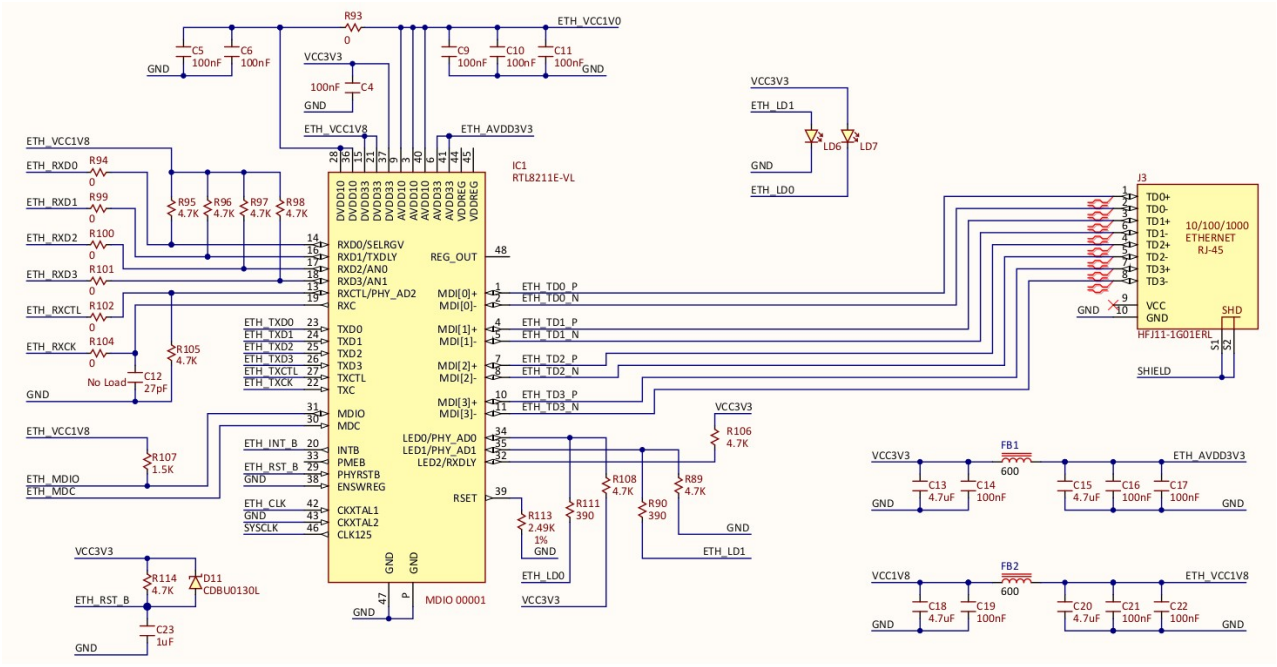
В “Page navigator” → MIO Configuration → I/O Peripherals → маха се отметката на → USB0. Проверяват се връзките на MII интерфейса:

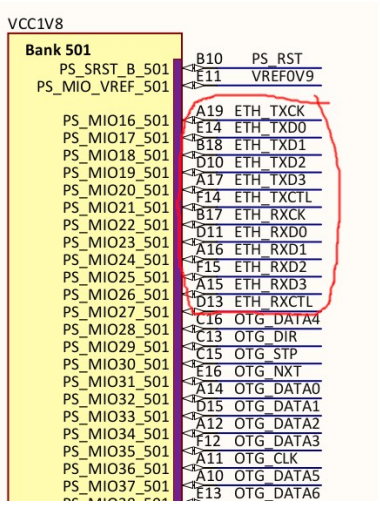
MIO52 ↔ mdc
MIO53 ↔ mdio

MIO16 ↔ tx_clk
MIO17 ↔ txd[0]
MIO18 ↔ txd[1]
MIO19 ↔ txd[2]
MIO20 ↔ txd[3]
MIO21 ↔ tx_ctl

MIO22 ↔ rx_clk
MIO23 ↔ rxd[0]
MIO24 ↔ rxd[1]
MIO25 ↔ rxd[2]
MIO26 ↔ rxd[3]
MIO27 ↔ rx_ctl

Забележете IC1 – RTL8211E-VL, който реализира физическия слой (PHY) на Етернет интерфейса. Той включва транслиране и кодиране на сигналите, разбъркване на данните (scramble) и договаряне на връзката.





11. В същия прозорец → “Page navigator” → MIO Configuration → Проверяват се връзките на UART1 интерфейса:

MIO48 ↔ tx
MIO49 ↔ rx

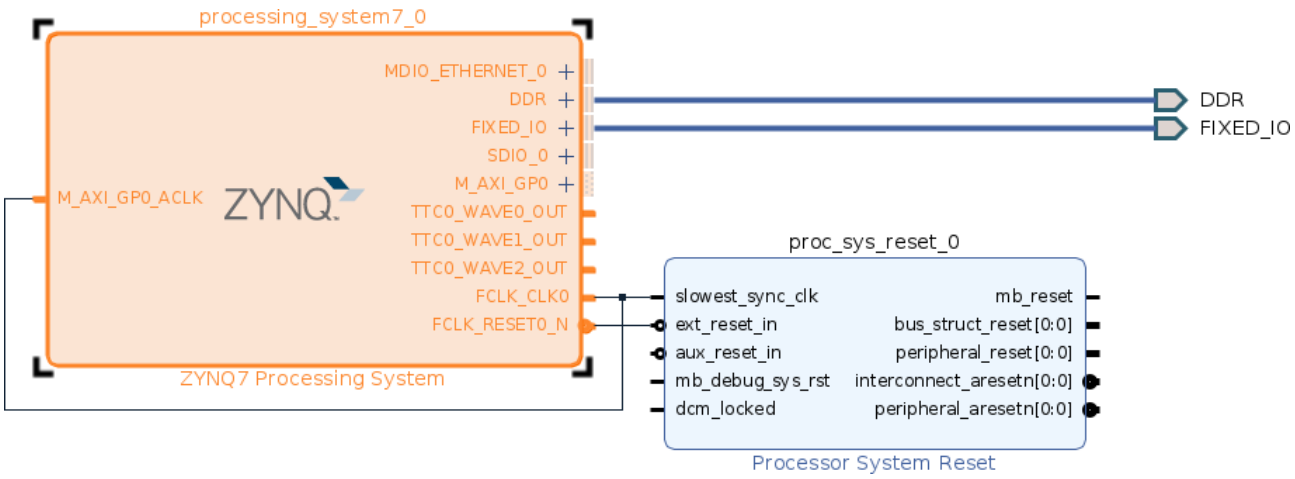
12. Подрежда се блоковата схема с бутон Regenerate Layout.

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

14. 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

Блоковата схема на системата трябва да изглежда така:



15. 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.

16. File → Export → Export hardware → Next → Include bitstream → Next → Next → Finish

==========
==========

17. Tools → Launch Vitis IDE

18. 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.

19. File → New → Platform project → Platform project name: 08_eth_pla → Next → таб "Create new platform from hardware" → Browse → избира се пътя до проекта 08_eth, създаден от Vivado → design_1_wrapper.xsa → Open → Finish.

20. Ще се използва библиотеката с отворен сорс код lwIP, която реализира TCP/IP протокола. Тази библиотека не е добавена по подразбиране към проекта. Затова: вляво на средата до таб Explorer има друга таб, Assistant → избира се проекта 08_eth_pla [Platform] → десен бутон → Open Platform Editor → избира се “standalone on ps7_cortexa9_0 → Board Support Package → централно ще се появи таб с бутон “Modify BSP Settings” → натиска се този бутон → слага се отметка на Supported libraries / lwip211.

21. В същия прозорец отдясно в графа Overview трябва да се появи standalone/lwip211 → dhcp_options → lwip_dhcp → Value = true.

ВНИМАНИЕ: ако “autonegotiation link speed” не минава успешно, трябва в този прозорец да изберете temac_adapter_options/phy_link_speed → Value = 1000 Mbps [1] → OK.

ВНИМАНИЕ: ако “autonegotiation link speed” не минава успешно, и ако рутерът ви е с 100-мегабитови портове, изберете phy_link_speed = 100 → OK.

22. Отворете във Vitis файла:

workspace_vitis/08_eth_pla/export/08_eth_pla/sw/08_eth_pla/standalone_domain/bspinclude/include/lwipopts.h

и проверете дали макроса LWIP_DHCP е равен на 1:

```
#define LWIP_DHCP 1
```

23. Проверете дали макроса CONFIG_LINKSPEED_AUTODETECT в

workspace_vitis/08_eth_pla/export/08_eth_pla/sw/08_eth_pla/standalone_domain/bspinclude/include/lwipopts.h

е равен на 1:

```
#define CONFIG_LINKSPEED_AUTODETECT 1
```

24. Вляво → Project explorer → избира се 08_eth_pla (Out-of-date) → right-click → Build Project.

25. File → New → Application project → Next → "Select a platform from repository" → Избира се 08_eth → Next → Application project name: 08_eth_app → Next → Next → “lwIP Echo Server” → Finish.

26. Щраква се двукратно с ляв бутон върху директорията src в проекта 08_eth_app_system/08_eth_app → src → echo.c

27. В текстовия редактор на Vitis ще има заредена примерна програма, използваща socket-и и TCP/IP комуникация, за да направи exo на TCP/IP терминал.

28. Очаква се exo сървърът да има **IP дадено от рутера** (вижте UART терминала) и **порт номер 7**.

ВНИМАНИЕ: в main( ) функцията има масив:

```
unsigned char mac_ethernet_address[] =
     { 0x00, 0x0a, 0x35, 0x00, 0x01, 0x02 };
```

който съдържа MAC адрес на устройството. Променете този MAC, така че той да е различен за различните работни маси.

29. Вляво, Project explorer → избира се 08_eth_app_system → right-click → Build project.

30. Вляво, Project explorer → избира се 08_eth_app_system → right-click → Debug as → Launch Hardware.

31. Свържете LAN кабел към куплунга J3 на Zybo от едната страна и мрежа с DHCP сървър от другата страна (рутер).

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:

```
[12:27:07:041] -----lwIP TCP echo server ------
[12:27:07:041] TCP packets sent to port 6001 will be echoed back
[12:27:07:057] Start PHY autonegotiation
[12:27:07:057] Waiting for PHY to complete autonegotiation.
[12:27:11:080] autonegotiation complete
[12:27:11:080] link speed for phy address 1: 1000
[12:27:11:080] unable to determine type of EMAC with baseaddress 0xE000B000
[12:27:12:257] Board IP: 192.168.0.108
[12:27:12:257] Netmask : 255.255.255.0
[12:27:12:257] Gateway : 192.168.0.1
[12:27:12:257] TCP echo server started @ port 7
[12:27:12:257]
```

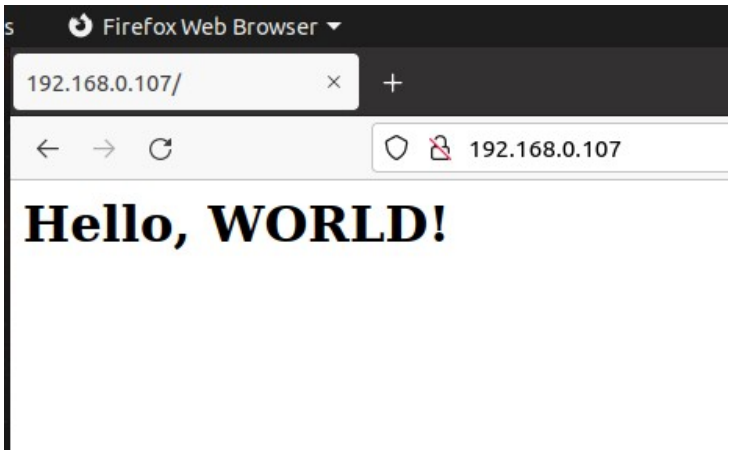
35. Отворете терминал с CTRL + ALT + t и напишете команда за стартиране на TCP/IP терминал, например:

```
telnet 192.168.0.108 7
```

след което въведете произволен текст. Ако примерът се е заредил успешно, трябва да виждате буквите на изречението, което пишете (exo). С Enter затваряте socket-а.

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

37. Припомнете си ученическите години и напишете уеб страница на HTML, която да се поддържа от сървъра на FPGA. Опитайте да достъпите страницата с уеб браузър от локалната мрежа (от външни IP-та няма да стане, защото портовете са затворени). Напомняне: уеб-браузърите използват **порт 80** и изобразяват низовете на HTML чак след **като се затвори socket-а** на връзката.



* * *

[1] https://digilent.com/reference/learn/programmable-logic/tutorials/zybo-getting-started-with-zynq-server/start
[2] https://github.com/Xilinx/embeddedsw/issues/70

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