1. Vytvorenie nového projektu

- 1.1. Spusťte Vivado HLS: vyberte Start > Všetky programy > Xilinx Design Tools > Vivado HLS 2015.4.
- 1.2. V spustenom programe kliknite na **Create New Project**. Otvorí sa dialógové okno pre vytvorenie nového projektu.
- 1.3. Kliknite na Browse... v poli Location, pre umiestnenie projektu vyberte cestu c:\Vivado_projects\lab4 a následne kliknite na OK.
- 1.4. Projekt nazvite fir.prj.
- 1.5. Kliknite na Next.
- 1.6. V okne Top Function nazvite funkciu fir.
- 1.7. Pre výber zdrojových súborov kliknite na Add Files..., z priečinka c:\Vivado_projects\lab4 vyberte súbory *fir.c* a *fir_coef.dat* a kliknite na Open.
- 1.8. Kliknite na Next.
- 1.9. Kliknite na Add Files... pre pridanie súborov na testbench. Z priečinka
 c:\Vivado_projcts\lab4 vyberte súbor *fir_test.c* a kliknite na Open.
- 1.10. Kliknite na Next.
- 1.11. V okne Solution Configuration ponechajte názov riešenia solution1 a periódu taktu nastavte na 10 (pre ZedBoard) alebo na 8 (pre Zybo). Pole Uncertainty ponechajte prázdne, ZedBoard prijíma ako predvolenú hodnotu 1.25 a Zybo 1.
- 1.12. Kliknite na ... pre výber súčastí. V danom filtri zvoľte súčasť xc7z020clg484-1 (ZedBoard) alebo xc7z010clg400-1 (Zybo) a kliknite na OK: Family: Zynq

Sub-Family: **Zynq** Package: **clg484** (ZedBoard) or **clg400** (Zybo) Speed Grade: **–1**

1.13. Kliknite na Finish.

Novovytvorený projekt môžete vidieť na karte Explorer. Rozbaľte jednotlivé podpriečinky aby ste videli aké položky sa v nich nachádzajú.

1.14. Rozbal'te priečinok Source a dvojitým kliknutím otvorte súbor fir.c.

```
1#include "fir.h"
 3 void fir (
 4 data_t *y,
   data_t x
 6){
     const coef_t c[N+1]={
 8 #include "fir_coef.dat"
0
      };
10
11
12 static data_t shift_reg[N];
13
    acc_t acc;
14 int i;
15
16 acc=(acc_t)shift_reg[N-1]*(acc_t)c[N];
17 loop: for (i=N-1;i!=0;i--) {
      acc+=(acc_t)shift_reg[i-1]*(acc_t)c[i];
shift_reg[i]=shift_reg[i-1];
18
19
20 }
21 acc+=(acc_t)x*(acc_t)c[0];
22
    shift_reg[0]=x;
23 *y = acc >> 15;
```

Obrázok 1 Posudzovaný návrh

Jeho obsah sa otvorí v informačnom okne. FIR filter očakáva x ako príklad vstupu a smerník pre výpočet príkladu výstupu. Oba sú zadefinované typom data_t. Koeficienty sú načítané v poli c typu coef_t zo súboru nazývaného fir_coef.dat v aktuálnom adresári. Sekvenčný algoritmus je aplikovaný a akumulovaná hodnota je vypočítaná v premennej acc, ktorá je typu acc_t.

1.15. Dvakrát kliknite na **fir.h** na karte *Outline* pre otvorenie jeho obsahu v informačnom okne.

```
1#ifndef _FIR_H_
2#define _FIR_H_
3#include "ap_cint.h"
4#define N 58
5#define SAMPLES N+10 // just few more samples then number of taps
6 typedef short coef_t;
7 typedef short data_t;
8 typedef int38 acc_t;
9 #endif
10
```

Obrázok 2 Hlavičkový súbor

Hlavičkový súbor zahŕňa ap_cint.h, takže používateľom definovaná šírka (s ľubovoľnou presnosťou) dát sa môže používať. Je tu taktiež definovaný počet točiek (N), počet vzoriek, ktoré sa budú generovať (v testbenchi) a údaje typu coef_t, data_t a acc_t. Coef_t a data_t sú taktiež typu short (16 bit). Pretože algoritmus iteruje cez 59 točiek, je tu možnosť rastu 6 bitov a preto je acc_t definované ako int38. Pretože acc_t je väčšie ako vzorka a koeficient šírky, odovzdávajú sa ešte pre použitím (ako na riadku 16, 18 a 21 v súbore fir.c).

1.16. Dvakrát kliknite na fir_test.c v priečinku testbench pre otvorenie jeho obsahu v informačnom okne.
Všimnite si, že testbench otvára fir_impulse.dat v režime písania a odosiela impulzy (prvá vzorka je 0x8000).

2. Spustenie C simulácie

2.1. Vyberte **Project** > **Run C Simulation** alebo kliknite na ina paneli nástrojov a v dialógovom okne simulácie kliknite na OK.

Testbench bude skompilovaný pomocou kompilátoru apcc a vygeneruje sa súbor **csim.exe**. Po spustení csim.exe sa zobrazí výstup v konzolovom okne.

3. Syntetizácia návrhu

- 3.1. Syntetizácia návrhu s predvolenými nastaveniami. Prezeranie výsledku syntetizácie a odpovedanie na pripravené otázky.
- 3.1.1. Vyberte Solution > Run C Synthesis > Active Solution pre spustenie syntetizácie.
- 3.1.2. Po skončení syntetizácie bude k dispozícii niekoľko súborov so správami a výsledky syntetizácie sa zobrazia v informačnom okne.
- 3.1.3. Správa syntetizácie zobrazuje výkon a odhad zdrojov ako aj odhadovanú latenciu návrhu.

3.1.4. Prezrite si výslednú správu a odpovedajte na nasledujúce otázky:

Odhadovaná perióda taktu:	
Najhorší prípad latencie:	
Celková hodnota DSP48E:	
Celková hodnota BRAM:	
Celková hodnota FF:	
Celková hodnota LUT:	

3.1.5. Správa tiež zobrazuje rozhranie signálov najvyššej úrovne generované nástrojmi.

Interface						
Summary						
RTL Ports	Dir	Bits	Protocol	Source Object	C Type	
ap_clk	in	1	ap_ctrl_hs	fir	return value	
ap_rst	p_rst in 1	1	ap_ctrl_hs	fir	return value	
ap_start	in	1	ap_ctrl_hs	fir	return value	
ap_done	out	1	ap_ctrl_hs	fir	return value	
ap_idle	out	1	ap_ctrl_hs	fir	return value	
ap_ready	out	1	ap_ctrl_hs	fir	return value	
у	out	16	ap_vld	У	pointer	
y_ap_vld	out	1	ap_vld	У	pointer	
x	in	16	ap_none	x	scalar	

Obrázok 3 Generované signály rozhrania

Môžete vidieť, že návrh očakáva vstup x ako 16-bitový skalár a výstup y cez ukazovateľ 16-bitových údajov. Je tu taktiež signál ap_vld, ktorý úkazuje, kedy je výsledok platný.

- 3.2. Pridanie direktívy PIPELINE pre slučky a opätovná syntetizácia návrhu. Prezeranie výsledku syntetizácie.
- 3.2.1. Uistite sa, že je **fir.c** otvorený v informačnom okne.
- 3.2.2. Prepnite sa na kartu **Directive** a použite direktívu **PIPELINE** pre **loop**.
- 3.2.3. Vyberte Solution > Run C Synthesis > Active Solution pre spustenie procesu syntézy.
- 3.2.4. Po skončení syntézy sa v informačnom okne zobrazí výsledok jej výsledok.

3.2.5. Všimnite si, že sa latencia znížila na 63 taktových periód. Spotreba DSP48 a BRAM ostala nezmenená, avšak spotreba LUT a FF mierne stúpla.

4. Spustenie RTL/V co-simulácie

- 4.1. Vyberte Solution > Run C/RTL Cosimulation alebo kliknite na ☑. Otvorí sa dialógové okno C/RTL Co-simulation.
- 4.2. Vyberte možnosť Verilog a kliknite na OK.

Spustí sa co-simulácia, vygeneruje sa a skompiluje niekoľko súborov a následne sa simuluje návrh. V konzolovom okne môžete vidieť jej vývoj. Po skončení simulácie môžete vidieť výslednú správu, ktorá ukazuje úspešnosť simulácie a hlásená latencia bola 63.

5. Nastavenie IP-XACT adaptéra

- 5.1. Uistite sa, že je súbor fir.c otvorený v informačnom okne.
- 5.2. Prepnite sa na kartu **Directive**.
- 5.3. Pravým tlačidlom myši kliknite na x a vyberte možnosť Insert Directive...
- 5.4. V zobrazenom dialógovom okne vyberte direktívu INTERFACE.
- 5.5. Kliknite na tlačidlo vedľa mode (optional). Vyberte s_axilite.
- 5.6. V poli bundle (optional) zadajte fir_io a kliknite na OK.

Vivado HLS Directive Directive	Editor
Destination Source File Directive File	
Options mode (optional): register (optional): depth (optional):	s_axilite
port (required): offset (optional):	×
bundle (optional):	fir_io
Help	Cancel OK

Obrázok 4 Výber AXI4LiteS adaptéra a pomenovanie zväzku

5.7. Rovnako použite direktívu INTERFACE (vrátane zväzku (bundle)) pre výstup y.

Directive INTERFACE	•
Destination	
Source File	
Directive File	
Options	
mode (optional):	s_axilite 🔹
register (optional):	
depth (optional):	
port (required):	у
offset (optional):	
bundle (optional):	fir_io
Help	Cancel OK

Obrázok 5 Výber AXI4LiteS adaptéra a pomenovanie zväzku pre výstup y

5.8. Použite direktívu **INTERFACE** pre modul **fir** zahrnutím signálov ap_start, ap_done a ap_idle ako súčasť zbernicového adaptéra.

Directive	
INTERFACE	•
Destination	
Source File	
 Directive File 	
Options	
mode (optional):	s_axilite 🔹
register (optional):	
depth (optional):	
offset (optional):	
bundle (optional):	fir_io

Obrázok 6 Výber AXI4LiteS adaptéra a pomenovanie zväzku pre funkciu kontrolných signálov

Všimnite si, že vyššie uvedené kroky 5.3 až 5.8 vytvorili adresnú mapu pre x, y, ap_start, ap_valid, ap_done a ap_idle, ktoré môžu byť prístupné cez softvér. Striedavo, signály ap_start, ap_valid, ap_done, ap_idle môžu byť generované ako samostatné porty na jadro bez použitia direktívy RESOURCE na moduli fir. Tieto porty sa pripájajú na procesor systému pomocou dostupných GPIO IP.

6. Generovanie IP-XACT adaptéra

6.1. Keďže boli pridané direktívy, je bezpečné znovu spustiť syntézu návrhu. Vyberte Solution > Run C Synthesis > Active Solution.

Skontrolujte či je v zhrnutí rozhrania na konci správy syntézy viditeľné rozhranie, ktoré bolo vytvorené.

6.2. Akonáhle je syntéza návrhu dokončená, vyberte **Solution > Export RTL**. Otvorí sa dialógové okno Export RTL.

xport RTL	e	8
Format Selection		
IP Catalog	▼ Configuration	
Options Evaluate Verilog	•	
	Do not show this dialog box as	gai
		-

Obrázok 7 Dialógové okno Export RTL

- 6.3. Kliknite na OK, následne sa vygeneruje adaptér IP-XACT.
- 6.4. Po skončení behu rozbaľte priečinok **impl** na karte Explorer a pozorujte jednotlivé vytvorené adresáre: ip, verilog a vhdl.



```
Obrázok 8 Generovaný adaptér IP-XACT
```

Rozbal'te priečinok **ip** a pozorujte niekoľko súborov a podadresáre. Jeden podadresár je adresár s ovládačmi, ktorý sa skladá z hlavičkových, c, tcl, mdd a makefile súborov. Ďalší súbor je zip súbor, ktorý je odkladacím priestorom ip.



Obrázok 9 Adresáre adaptéra

6.5. Zatvorte Vivado HLS výberom **File > Exit**.

7. Vytvorenie projektu Vivado

7.1. Spustenie Vivado TCl Shell a spustenie poskytnutého tcl skriptu pre vytvorenie prvotného systému zameraného na Zedboard (zariadenie xc7z020clg484-1) alebo Zybo (zariadenie xc7z010clg400-1).

Ak chcete vytvoriť systém od nuly, potom postupujte podľa pokynov uvedených v dodatku a následne pokračuje nižšie krokom 7.2.

- 7.1.1. Otvorte Vivado Tcl Shell výberom Start > Všetky programy > Xilinx Design Tools
 > Vivado 2015.4 Tcl Shell
- 7.1.2. V shell okne zmeňte adresár na c:/Vivado_projects/lab4 pomocou príkazu cd.

7.1.3. Spusťte poskytnutý skript pre vytvorenie pôvodného systému so zed_autio_ctrl a GPIO perifériami zadaním nasledujúcich príkazov:

source zed_audio_project_create.tcl pre ZedBoard alebo

source zybo_audio_project_create.tcl pre Zybo

Skript sa spustí a pôvodný systém, uvedený nižšie, bude vytvorený.



Obrázok 10 Návrh blokov so zed_audio_ctrl a spoje pre ZedBoard



Obrázok 11 Návrh blokov so zed_audio_ctrl a spoje pre Zybo

7.2. Pridanie HLS IP do IP katalógu

7.2.1. Na karte Flow Navigator kliknite na Project Settings pod Project Manager.

- 7.2.2. Kliknite na ikonu **IP**.
- 7.2.3. Kliknite na tlačidlo + (adresár lab4/ip_repo už bol pridaný). Prejdite na c:\Vivado_projects\lab4\fir.prj\solution1\impl\ip a kliknite na Select.
 Adresáre budú naskenované a pridané v okne *IP Repositories* a bude zistený jeden IP záznam.
- 7.2.4. Kliknite na **OK**.
- 7.2.5. Kliknite na OK pre potvrdenie nastavení.
- 7.3. Vytvorenie inštancie jadra fir_top dvakrát, jednu pre každú stranu kanála, do spracovania systémového názvu inštancie ako fir left a fir_right.
- 7.3.1. Kliknite na ikonu Add IP ^P a vyhľadajte Fir v katalógu zadaním Fir a dvojitým klikom na *Fir* pridajte inštanciu.
- 7.3.2. Kliknite na tlačidlo Add IP to Block Design, ak je zobrazené. Všimnite si, že pridaná IP má HLS logo, ktoré indikuje, že bola vytvorená od Vivado HLS.
- 7.3.3. Vyberte pridané inštancie v diagrame a zmeňte ich názov na fir_left zadaním názvu do poľa *Name* na formulári *Block Properties* vľavo.
- 7.3.4. Rovnako pridajte ostatné inštancie z HLS IP, a pomenujte ich fir_right.
- 7.3.5. Kliknite na Run Connection Automation a vyberte All Automation.
- 7.3.6. Kliknite na /fir_left/s_axi_fir_io a /fir_right/ s_axi_fir_io, overte, že budú obe pripojené na M_AXI_GP0 a kliknite na OK.
- 7.3.7. Rovnako opäť kliknite na Run Connection Automation, vyberte /fir_right/ s_axi_fir_io a kliknite na OK.

- 7.4. Umožnenie portov PSL-PL Interrupt > portov IRQ_F2P. Pridanie inštancie concat IP s dvoma jednobitovými vstupnými portmi. Pripojenie vstupných portov na prerušenie portov z dvoch FIR inštancií a výstupných portov na port IRQ_F2P z inštancie processing system7 0.
- 7.4.1. Dvakrát kliknite na inštanciu processing_system7_0 pre otvorenie formulára.
- 7.4.2. Vyberte Interrupt v l'avej časti okna, kliknite vpravo na políčko Fabric Interrupts.
- 7.4.3. Rozbal'te vstupy *Fabric Interrupts > PL-PS Interrupt Ports > IRQ_F2P* (vpravo) a kliknite na políčko *IRQ_F2P[15:0]*.
- 7.4.4. Kliknite na OK.
- 7.4.5. Pridajte inštanciu concat IP.
- 7.4.6. Pripojte prerušené porty každej inštancie FIR na dve vstupné porty inštancie *xlconcat_0*.
- 7.4.7. Pripojte výstupný port inštancie *xlconcat_0* na port IRQ_F2P inštancie *processing_system7_0*.
 Momentálny návrh môžete vidieť nižšie (možno bude potrebné kliknúť na tlačidlo regenerate).





Obrázok 12 Úplný návrh

7.5. Overenie adries a validácie návrhu. Generovanie súboru system_wrapper a pridanie uvedeného Xilinx Design Contraints (XDC).

7.5.1. Kliknite na Address Editor a rozbaľte processing_system7_0 > Data, ak je to potrebné.

Generované mapy adries sú uvedené nižšie.

Cell	Slave Interface	Base Name	Offset Address	Range	High Address
G processing_system7_0					
er la Data (32 address bits : 46	S AXI	Reg	0x4120 0000	64K 1	0x4120 FFFF
zed_audio_ctrl_0	S_AXI	reg0	0x43C0_0000	64K -	0x43C0_FFFF
fir_left	s_axi_fir_io	Reg	0x43C1_0000	64K *	Cx43C1_FFFF
	s_axi_fir_io	Reg	0x43C2_0000	64K *	0x43C2_FFFF

a) ZedBoard

Cell		Slave Interface	Base Name	Offset Address	Rang	e	High Address
9 processing	system7_0						
🖃 🛄 Data (32	address bits : 0x40000	000 [1G])					
- wa and gp	io_0	S_AXI	Reg	0x4120_0000	64K	٠	0x4120_FFFF
- ** zybo_i	udio_ctrl_0	S_AXI	reg0	0x6000_0000	64K		Ox6000_FFFF
fir_left		s axi fir io	Reg	0x43C0_0000	64K		0x43C0_FFFF
- sa fir rig	nt	s and fir io	Reg	0x43C1 0000	64K		0x43C1 FFFF

b) Zybo

Obrázok 13 Generované mapy adries

- 7.5.2. Spusťte *Design Validation* (**Tools > Validate Design**) a overte, či nie sú prítomné žiadne chyby.
- 7.5.3. V pohľade *sources*, kliknite pravým tlačidlom na blok súboru diagramu, system.bd, a vyberte Create HDL Wrapper pre aktualizáciu súboru HDL wrapper. Po zobrazení výzvy, kliknite na OK s výberom nastavenia *Let Vivado manage wrapper and auto-update*.
- 7.5.4. Kliknite na Add Sources na karte *Flow Navigator*, vyberte Add or Create Constraints a kliknite na Next.
- 7.5.5. Kliknite na Add Files, vyhľadajte priečinok c:\Vivado_projects\lab4, vyberte zed_audio_constraints.xdc alebo zybo_audio_constraints.xdc.
- 7.5.6. Kliknite na **Finish** pre pridanie súboru.

- 7.5.7. Kliknite na Generate Bitstream vo Flow Navigator pre spustenie syntézy, implementácie a bistream gerovania procesov.
- 7.5.8. Kliknite na Save a Yes, ak sa otvorí výzva na spustenie procesu.
- 7.5.9. Keď je generovanie bitov dokončené, zobrazí sa okno s výberom nastavenia Open Implemented Design. Kliknite na Cancel.

8. Export do SDK a vytvorenie projektu aplikácie

- 8.1. Vyberte File > Export > Export Hardware...
- 8.2. Uistite sa, že je vybraté nastavenie **Include Bitstream** a kliknite na **OK**. Cieľový adresár nechajte nastavený na lokálny adresár projektu.
- 8.3. Vyberte File > Launch SDK.
- 8.4. Kliknite na OK.
- 8.5. V SDK vyberte File > New > Board Support Package.
- 8.6. Kliknite na Finish s predvolenými nastaveniami (so samostatným operačným systémom).
 Tým sa otvoria nastavenia platformy softvéru zobrazujúce výber operačného systému a knižníc.
- 8.7. Kliknite na OK pre potvrdenie predvolených nastavení, ak chceme vytvoriť platformu projektu standalone_bsp_0 bez nutnosti podpory akýchkoľvek dodatočných knižníc. Generovanie knižnice sa spustí na pozadí a vytvorí sa súbor xparameters.h v adresári c:\Vivado_projects\lab4\audio\audio.sdk\standalone_bsp_0\ps7_cortexa9_0\include\.
- 8.8. Vyberte File > New > Application Project.

- 8.9. Zadajte **TestApp** ako názov projektu a pre *Board Support Package* vyberte **Use Existing** (standalone_bsp by mala byť jediná možnosť).
- 8.10. Kliknite na Next, vyberte *Empty Application* a kliknite na Finish.
- 8.11. Vyberte **TestApp** v okne projektu, kliknite pravým tlačidlom na priečinok src a vyberte možnosť **Import**.
- 8.12. Rozbal'te kategóriu General a dvakrát kliknite na File System.
- 8.13. Vyhľadajte priečinok c:\Vivado_projects\lab4 a kliknite na OK.
- 8.14. Vyberte zed_testapp.c a zed_audio.h pre ZedBoard alebo zybo_testapp.c a zybo_audio.h pre Zybo a kliknite na Finish pre pridanie súborov do projektu. Program by mal byť úspešne zostavený.

9. Overenie návrhu v hardvéri

- 9.1. Len Zybo: Uistite sa, že JP7 je dané do možnosti USB napájania.
- 9.2. Pripojte micro-usb kábel medzi PC a JTAG port základnej dosky.
- 9.3. Pripojte audio kábel medzi konektor Line In a reproduktory (slúchadlá) počítača.
- 9.4. Pripojte slúchadlá do konektora Line Out na ZedBoard alebo do konektora HPH Out na dosku Zybo. Dosku zapnite.
- 9.5. Vyberte Xilinx Tools > Program FPGA.
- 9.6. Uistite sa, že je bistream system_wrapper.bit vybratý a pole súboru BMM je prázdne.
- 9.7. Kliknite na **Program**.

To bude konfigurovať FPGA.

- 9.8. Dvakrát kliknite na corrupted_music_4KHz.wav alebo na nejaký iný súbor na prehratie, ktorý je možné prehrať nainštalovaným prehrávačom. Prepnite ho do režimu nepretržitého prehrávania.
- 9.9. Kliknite pravým tlačidlom myši na *TestApp* na karte Project Explorer a vyberte **Run As** > Launch On Hardware (GDB).

Program sa stiahne a spustí. Ak chcete počúvať poškodený signál, vypnite SW0. Ak chcete počúvať filtrovaný signál, zapnite SW0.

9.10. Ak budete hotový, ukončte program kliknutím na Terminate (červený štvorec) na karte konzoly SDK. Vypnite dosky a ukončte SDK a Vivado použitím File > Exit.

Odpovede

Odhadovaná perióda taktu:	6.38 ns
Najhorší prípad latencie:	175 clock cycles
Celková hodnota DSP48E:	3
Celková hodnota BRAM:	0
Celková hodnota FF:	168
Celková hodnota LUT:	106

Dodatok

10. Vytvorenie projektu pomocou Vivado GUI

- 10.1. Otvorte Vivado výberom Start > Všetky programy > Xilinx Design Tools > Vivado
 2015.4
- 10.2. Kliknite na **Create New Project**. Otvorí sa dialógové okno vytvorenia nového projektu. Kliknite na **Next**.
- 10.3. Kliknite na Browse pre výber umiestnenia projektu, vyberte c:\Vivado_projects\lab4 a kliknite na Select.

- Do poľa *Project Name* zadajte ako názov projektu audio. Uistite sa, že políčko *Create Project Subdirectory* je zaškrtnuté. Kliknite na Next.
- 10.5. Vyberte **RTL Project** vo formulári Project Type a kliknite na **Next**.
- 10.6. Ako cieľový jazyk a jazyk simulátora vyberte Verilog a kliknite na Next.

Specify H to your p	s IDL and netlist files, o roject. Create a new	r directories containing HDI source file on disk and add	and netlist files, to add it to your project. You	1
[Add Files	Add Directories	Create File	
Scan an	d add RTL include file	s into project		
Copy so	urces into project			
Add sou	rces from subdirector	ies		
Target lan	guage: Verilog 💌	Simulator language: Verilo	og ₹	
				Coursel

Obrázok 14 Pridanie zdrojov do nového projektu

- Kliknite dvakrát na Next aby ste preskočili dialógové okná pridávania existujúcich IP a pridávania obmedzení.
- 10.8. Na formulári *Default Part*, vyberte Boars a buď vyberte Zedboard Zynq Evaluation and Development Kit alebo Zybo. Kliknite na Next.

Ak nevidíte položky Zybo a chcete sa zamerať na dosku Zybo, potom si prosím prečítajte súbor readme_zybo.docx a nainštalujte súbory zybo dosky v inštalačnom adresári Vivado.

10.9. Skontrolujte súhrn projektu a kliknite na Finish, prázdny projekt Vivado sa vytvorí.

11.Vytvorenie systému použitím IP iterátora

11.1. Použitie IP iterátora pre vytvorenie nových blokov návrhu a generovanie ARM Cortex-A9 procesora založeného hardvérom systému. 11.1.1. Vo *Flow Navigator*, kliknite na **Create Block Design** pod IP Iterator.



Obrázok 15 Vytvorenie IP iterátora blokov návrhu

- 11.1.2. Zadajte system ako názov návrhu a kliknite na OK.
- 11.1.3. IP z katalógu možno doplniť rôznymi spôsobmi. Kliknite na Add IP v správe v hornej časti panela diagramu alebo kliknite na ikonu Add IP na bočnej lište, stlačením Ctrl + I, alebo kliknite pravým tlačidlom myši na ľubovoľné miesto v okne diagramu a vyberte Add IP.
- 11.1.4. Akonáhle je IP katalóg otvorený, zadajte "zy" do vyhľadávača, nájdite a dvakrát kliknite na položku ZYNQ7 Processing System alebo kliknite na položku a stlačte kláves Enter pre pridanie do návrhu. Pridá sa Zynq blok.



Obrázok 16 Zynq IP blok

- 11.1.5. V hornej časti okna diagramu je oznamujúca správa, že je k dispozícii pomoc asistenta. Kliknite na **Run Block Automation** a vyberte /processing_system7_0.
- 11.1.6. Kliknite na OK, keď sa zobrazí výzva na spustenie automatizácie.

Všimnite si, že sa externé porty automaticky pridali pre DDR a Fix IO, akonáhle je automatizácia blokov dokončená; niektoré ďalšie predvolené porty boli tiež pridané do bloku.



Obrázok 17 Zyną blok s portmi DDR a Fixed IO

11.1.7. V diagrame bloku, dvakrát kliknite na Zynq blok, aby sa otvorilo okno prispôsobenia pre Zynq spracovanie systému.

Diagram bloku Zynq by mal byť teraz otvorený, zobrazuje rôzne konfigurovateľné bloky spracovania systému.

V tejto fáze môže dizajnér kliknúť na rôzne konfigurovateľné bloky (zvýraznené na zeleno) a zmeniť konfiguráciu systému.

- 11.2. Konfigurácia I/O periférnych blokov pre použitie periférií UART 1 a I2C 1, vypnutie ostatných nežiaducich periférií. Zrušenie označenia políčka Timer 0. Umožnenie FCLK_CLK1, PL fabric clock a frekvencií buď 10.000 MHz pre ZedBoard alebo 12.288 MHz pre Zybo.
- 11.2.1. Vyberte *MIO Configuration* (vľavo) pre otvorenie konfiguračného formulára a rozbaľte *I/O Peripheral* v pravej časti.
- 11.2.2. Kliknite na políčko periférie *I2C 1*. Zrušte zaškrtnutie *USB0*, *SD 0*, *ENET 0*, *GPIO* > *GPIO MIO*, tie nebudeme potrebovať.

- 11.2.3. Rozbal'te skupinu Application Processing Unit, v zozname vyberte kartu MIO Configuration a zrušte označenie pre Timer 0.
- 11.2.4. Vyberte *Clock Configuration* na l'avej strane tabule, rozbal'te položku *PL Fabric Clocks* (vpravo) a zaškrtnite políčko *FCLK_CLK1*.
- 11.2.5. Zmeňte hodnotu *Requested Frequency* FCLK_CLK1 na 10.000 MHz pre ZedBoard alebo 12.288 MHz pre Zybo.

Peripheral I/O Pins		Component	Clock Source	Requested Frequ
MIO Configuration	"E	Processor/Memory Clock	s	
Clock Configuration		PL Fabric Clocks		
DDR Configuration		- FCLK_CLK0	IO PLL .	100.000000
CMC Timing Calculation		FCLK_CLK1	IO PLL 🔻	10.000000 💿
SMC Timing Calculation		- FCLK_CLK2	JO PLL	50.000000
Interrupts		FCLK_CLK3	JO PLL	50

a) ZedBoard

\$	Component 1		Clock Source	Actual Frequency	Range(MHz)	
8	Đ	Timers				
	Đ	System Debug Clocks				
	÷.	Processor/Memory Clocks				
	PL Fabric Clocks					
		FCLK_CLK3	IO PLL	50	50.000000	0.100000 : 250.000000
		FCLK_CLK2	IO PLL	50	50.000000	0.100000 : 250.000000
		FCLK_CLK1	IO PLL 🔻	12.288	12.280702	0.100000 : 250.000000
		FCLK_CLK0	IO PLL 🔻	100	100.000000	0.100000 : 250.000000

b) Zybo

Obrázok 18 Povolenie a nastavenie FCLK_CLK1 frekvencie

11.2.6. Kliknite na OK.

Všimnite si, že Zynq blok zobrazí len príslušné porty.

11.3. Pridanie predpokladu I2C-based buď zed_audio_ctrl IP pre ZedBoard alebo zybo_audio_ctrl IP pre Zybo do IP katalógu.

11.3.1. Na karte Flow Navigator, kliknite na IP Catalog pod Project Manager. Otvorí sa IP katalóg.



Obrázok 19 Vyvolanie IP katalógu

11.3.2. Kliknite na IP Settings v IP katalógu.



Obrázok 20 Vyvolanie IP nastavení

- 11.3.3. Kliknite na Add Reppository.... Vyhľadajte adresár c:\Vivado_projects\lab4\ip_repo a kliknite na Select.
 Všimnite si, že položky zed_audio_ctrl a zybo_autio_ctrl sa zobrazia v poli *IP in Selected Repository*.
- 11.3.4. Kliknite na OK pre potvrdenie nastavení.

- 11.4. ZedBoard: inštancovanie zed_audio_ctrl a GPIO so šírkou 2 bity na kanáli 1 a šírkou 1 bit na kanáli 2.
 Zybo: inštancovanie zybo_audio_ctrl a GPIO so šírkou výstupu 1 bit len na kanáli 1 a šírkou vstupu 1 bit len na kanáli 2.
 Spustenie pripojenia automatizácie pre ich prepojenie.
- 11.4.1. Kliknite na Add IP I ak IP katalóg nie je otvorený a vyhľadajte AXI GPIO v katalógu zadaním gpi a dvakrát kliknite na položku AXI GPIO pre pridanie inštancie.
- 11.4.2. Kliknite na tlačidlo Add IP to Block Design.
- 11.4.3. Dvakrát kliknite na pridanie inštancie a zobrazí sa GUI Re-Customize IP.
- 11.4.4. Zmeňte šírku kanála 1 na 2 pre ZedBoard alebo šírku 1 output only pre Zybo.
- 11.4.5. Zaškrtnite políčko Enable Dual Channel, nastavte šírku na 1 input only a kliknite na OK.
- 11.4.6. Podobne pridajte inštanciu buď zed_audio_ctrl pre ZedBoard alebo zybo_audio_ctrl pre Zybo.
- 11.4.7. Všimnite si, že pomoc návrhu je k dispozícii. Kliknite na Run Connection Automation a vyberte /axi_gpio_0/S_AXI.
- 11.4.8. Kliknite na OK pre pripojenie sa do rozhrania M_AXI_GP0.
 Všimnite si, že ďalšie dva bloky, *Proc Sys Reset* a AXI Interconnect, sú automaticky pridané do návrhu.
- 11.4.9. Podobne, kliknite na Run Connection Automation a vyberte buď /zed_audio_ctrl_0/S_AXI pre ZedBoard alebo /zybo_audio_ctrl_0/S_AXI pre Zybo a kliknite na OK.

- 11.5. Vytvorenie IIC_1, GPIO, FCLK_CLK1 a buď zed_audio_ctrl alebo zybo_audio_ctrl vonkajších portov.
- 11.5.1. Zvoľte interface GPIO inštancie axi_gpio_0, kliknite na neho pravým tlačidlom myši vyberte Make External pre vytvorenie vonkajšieho portu. To vytvorí vonkajší port s názvom GPIO a pripojí ho k perifériám.
- 11.5.2. Zvoľte interface GPIO2 inštancie axi_gpio_0, kliknite na neho pravým tlačidlom myši a vyberte Make External pre vytvorenie vonkajšieho portu.
- 11.5.3. Podobne, vyberte jeden port buď inštancie zed_audio_ctrl_0 alebo inštancie zybo_audio_ctrl_0, aby boli vonkajšie.
- 11.5.4. Podobne urobte vonkajšími interface IIC_1 a port FCLK_CLK1 inštancie processing_system7_0.
 Súčasnú podobu návrhu môžete vidieť nižšie (možno budete musieť kliknúť na tlačidlo regenerovania ^(M)).



Obrázok 21 Výsledný návrh pre ZedBoard



Obrázok 22 Výsledný návrh pre Zybo