



دانشكده مهندسي كامپيوتر

دستورکار **آزمایشگاه طراحی خودکار سیستمهای دیجیتال آزمایشگاه معماری کامپیوتر**(ویژه کارشناسی پیوسته-نیمسال اول ۹۸-۹۷)

> تهیه و ^{تد}وین: ا**میر باوفای طوسی** م**حسن سیاوشی** زمستان ۹۶





Model Sim.



٣

پیشگفتار

آزمایشگاه ابزارهای طراحی با کمک کامپیوتر که از دروس تخصصی گرایش معماری سیستمهای کامپیوتری میباشد، با هدف آشنایی عملی دانشجویان با زبانهای توصیف سخت افزار و استفاده از این زبانها برای پیادهسازی سیستمهای دیجیتال روی FPGA و نکات عملی آموزش داده میشود. در این آزمایشگاه دانشجویان با طراحی خودکار سیستمهای دیجیتال بزرگ و ابزارهای مورد نیاز برای شبیهسازی، سنتز، پیادهسازی و دیباگکردن این مدارها آشنا میشوند. دستورکار این آزمایشگاه در قالب ۱۵ آزمایش تهیه شده است.

در انتها از تمامی صاحبنظران در زمینه طراحی خودکار سیستمهای دیجیتال درخواست می شود پیشنهادات و نظرات خود را درخصوص این دستورکار در جهت بهبود و ارتقاء هرچه بیشتر دانش و تخصص دانشجویان عزیز و مهندسین آینده این مرز و بوم، از طریق ایمیلهای زیر با ما در میان بگذارند.

1- abavafat@yahoo.com

2- mohsensiavoshi69@gmail.com

رویه و مقررات آزمایشگاه قبل از شروع جلسات آزمایشگاه لازم است دانشجویان دستور کار فعلی را تهیه و به نکات زیر توجه نمایند: ۱- دانشجویان موظفاند هر جلسه از ابتدای شروع کلاس در آزمایشگاه حضور داشته باشند و عدم حضوردانشجو در دقایق اولیه کلاس بدون هماهنگی با مدرس آزمایشگاه به منزله غیبت در جلسه مذکور خواهد بود. ۲- هر گروه شامل ۲ نفر می باشد . ۳- هر گروه موظف است بعد از انجام هر آزمایش، یک گزارش کامل مربوط به آزمایش فعلی (۲-۳ صفحه) را در ابتدای جلسه بعدی تحویل دهد. (البته برخی از آزمایشها به بیش از یک جلسه نیاز دارد که لازم است بعد از اتمام کار، گزارش کامل تحویل داده شود) ۴- گزارش کامل باید موارد زیر را در بر بگیرد: الف: توضيح مفاهيم و بيان هدف از آزمايش انجام شده ب: خروجی های حاصل از آزمایش و نتایج شبیه سازی که توسط اعضای گروه در آزمایشگاه بدست آمده است. پ: خروجیهای تکمیلی به صورت کار در منزل ت: بررسی و بیان قابلیت سنتزیذیری بر روی بورد خ: مشکلاتی که در حین انجام آزمایش با آنها برخورد کردید و راه حل پیشنهادی برای رفع آنها ج: نتيجه گيري ۵- هر گروه در پایان ترم یک پروژه شبیهسازی یا توصیفی مرتبط با مفاهیم مطرح شده ارائه خواهد داد. ۶- فعالیت کلاسی هر فرد در کلاس بصورت مجزا رصد می شود. ۷- بارم نمرات برای محاسبه نمره نهایی هر دانشجو به صورت زیر خواهد بود:

عنوان	بارم نمره
فعالیت کلاسی و روند نتیجه گیری هر آزمایش و ارائه گزارش هر جلسه	١.
امتحان پایانی	۵
پروژه نهایی	۵+۱

۸- دانشجویان ملزم به رعایت اخلاق و آداب مهندسی و دانشجویی هستند و از انجام کارهای خلاف همچون کپی کردن طرح دیگران، ارائه نتایج ساختگی برای آزمایشها، استخدام دیگران برای انجام بخشی از کارهای مربوط به آزمایش ها و ... اکیداً پرهیز کنند .در صورت مشاهده این موارد ضمن دریافت نمره مردودی، برابر با مقررات دانشگاه و دانشکده برخورد خواهد شد.

فهرست مطالب

مقدمه: زبان توصيف سخت افزار **آزمایش ۱:** آشنایی با برد DE2_115 شرکت Altera و ویژگیهای این برد **آزمایش ۲:** طراحی و پیادهسازی مدارهای ترکیبی و ترتیبی با استفاده از امکانات شماتیک **آزمایش ۳:** طراحی یک مدارجمع کنندہ کامل **آزمایش ۴:** طراحی ضربکننده و تکنیکهای برنامه نویسی VHDL آزمایش ۵: طراحی و ییادهسازی ساعت RTC **آزمایش ۶:** پیادهسازی مدارهای دیجیتال با استفاده از IPCoreهای شرکت Xilinx آزمایش ۷: آنالیز و عیبیابی طرح دیجیتال با استفاده نرم افزار ChipScop **آزمایش ۸**: طراحی و پیادهسازی پروتکل USART و RS232 € **آزمایش ۹**: طراحی و پیادهسازی پروتکل SPI و آشنایی با مبدلهای ADC **آزمایش ۱۰**: طراحی و پیادهسازی پروتکل I2C و آشنایی با سنسورهای دما **آزمایش ۱۱:** طراحی واحد کنترل دیجیتال یک سیستم انکوباتور آزمایش ۱۲: آشنایی با پردازندههای Embedded شده در FPGA (MicroBlaze) آزمایش ۱۳: طراحی و پیادهسازی پورت VGA برد DE2_115 **آزمایش ۱۴**: طراحی و پیاده سازی مبدلهای DAC و آشنایی با نحوه کارکرد آنها **آزمایش ۱۵:** آشنایی با برد Zybo و نرم افزار Vivado

مقدمه

زبان توصیف سخت افزار، زبانی است که سخت افزار سیستمهای دیجیتال را به فرم متنی توصیف مینماید. در واقع این زبان، صرفا یک زبان برنامه نویسی نیست، بلکه حول توصیف ساختارهای سخت افزاری و رفتار آنها بنا نهاده شده است. از زبان VHDL میتوان برای نمایش نمودارهای منطقی، عبارت بولی و دیگر مدارهای دیجیتال پیچیده استفاده کرد. این زبان به عنوان یک زبان مستندسازی برای نمایش و مستندکردن سیستمهای دیجیتال به کار میرود به نحوی که قابل خواندن به وسیله انسانها و کامپیوترها میباشد. این زبان به طور موثر و نیز به سادگی قابل ذخیره، بازیابی و پردازش به وسیله نرم افزار کامپیوتر است.

در پردازش HDLها دو کاربرد وجود دارد: الف- شبیه سازی منطقی، نمایشی از ساختار و رفتار یک سیستم منطقی دیجیتال به کمک یک شبیه ساز (مانند Modelsim) را تفسیر کرده و یک خروجی قابل خواندن مانند نمودار زمانی، تولید مینماید و بدین وسیله رفتار سخت افزار را قبل از ساخت پیشبینی مینماید و امکان تشخیص خطای عملیاتی در طراحی را بدون نیاز به خلق فیزیکی آن، فراهم میسازد.

ب- سنتز منطقی، فرایندی است که طی آن از قطعات و اتصال بین آنها به نام netlist درمدل سیستم دیجیتالی که در HDL توصیف شده است لیستی تهیه می گردد. netlist سطح گیت را می توان در ساخت یک مدار مجتمع یا طرح بورد مدار چاپی به کار برد. سنتز منطقی مشابه با کامپایل یک برنامه در زبان سطح بالاست. تفاوت در این است که در عوض تولید کد، یک بانک اطلاعاتی تولید می نماید که در آن دستورالعمل های ساخت یک قطعه سخت افزاری دیجیتال فیزیکی توصیف شده با کد HDL آمده است.

HDLهای انحصاری متعددی وجود دارند که به وسیله کمپانیها برای طراحی مدارهای مجتمع در صنعت ساخته شده اند. از این جمله میتوان به دو استاندارد VHDL و Verilog اشاره کرد که هر یک مزیتها و کاربردهای خاص خود را دارد.

دانشجویان در درسهای پیش نیاز آزمایشگاه، با یک زبان توصیف سخت افزار آشنا شدهاند. در این آزمایشگاه از زبان VHDL به منظور توصیف سختافزار استفاده خواهیم کرد که لازم است دانشجویانی که درس را اخذ نمودهاند مفاهیم و اصول زبان VHDL را مرور نمایند. از آنجا که سخت افزار مورد استفاده ما در این آزمایشگاه متعلق به شرکت Altera میباشد بنابراین برای پیاده سازی طرح بر روی FPGA موردنظر نیز از نرمافزارهای معرفی شده توسط این شرکت استفاده میکنیم که در این آزمایشگاه از نرم افزار aurus به منظور پیادهسازی طرح بر روی سخت افزار استفاده خواهیم کرد.

آزمایش ۱

آشنایی با برد DE2_115 شرکت Altera و ویژگیهای این برد

هدف از طراحی و عرضه بورد آموزشی و توسعه ALTERA DE2 عرضه یک ابزار ایدهآل برای نمونه سازی پیشرفته طرحهای FPGA در حوزههای مالتی مدیا، ذخیره و شبکه سازی داده هاست. تکنولوژی بکاررفته در طراحی این بورد به همراه ابزارهای CAD ارائه شده به همراه آن، افقهای جدیدی را در دامنه وسیعی از موضوعات بروی طراحان می گشاید. بورد دارای امکانات گستردهای است که در نهایت آنرا برای کاربردهای آموزشی در لابراتوارهای الکترونیک دانشگاهها ایده آل می سازد. این بورد برای پروژههای طراحی در موضوعات بروی طراحان می گشاید. بورد دارای امکانات گستردهای است که در نهایت آنرا برای کاربردهای آموزشی در لابراتوارهای الکترونیک دانشگاهها ایده آل می سازد. این بورد برای پروژههای طراحی در موضوعات مختلف حتی طراحی سیستمهای پیچیده دیجیتال نیز مناسب است. شرکت ALTERA میتوان به مطالب بسیار خوبی برای بورد 200 فراهم نموده است. از موارد ارائه شده توسط شرکت ALTERA میتوان به مطالب خودآموز و تمرینات برای لابراتوار های FPGA و دموهای تشریح شده اشاره نمود.

بورد ALTERA DE2 امکانات متعددی را برای کاربر در نظر گرفته است تا اجرای پروژههای مختلف در این بورد میسر گردد. قطعات بورد بگونهای انتخاب شده تا با المانهای استفادهشده در پروژههای شناخته شدهای که به تولید انبوه رسیدهاند مشابه باشد. پلتفرم DE2، کاربران و دانشجویان را قادر خواهد ساخت تا تمامی نکات کاربردی در طراحی پروژههای صنعتی را به خوبی درک نمایند.

قطعات و سخت افزارهای تعبیه شده در برد ALTERA DE2 به شرح زیر میباشد:

1_Altera Cyclone® IV 4CE115 FPGA device
3_USB Blaster for programming
5_Two 64MB SDRAM
7_SD Card socket
9_18 Slide switches
11_9 Green user LEDs
13_24-bit CD-quality audio CODEC
15_TV Decoder and TV-in connector
17_USB Host/Slave Controller
19_16x2 LCD module
21_2 SMA connectors for external clock
23_One 40-pin Expansion Header

- 2_Altera Serial Configuration device_EPCS64
 4_2MB SRAM
 6_8MB Flash memory
 8_4 Push-buttons
 10_18 Red user LEDs
 12_50MHz oscillator for clock sources
 14_VGA DAC with VGA-out connector
 16_2 Gigabit Ethernet PHY with RJ45
 18_RS-232 transceiver and 9-pin connector
- 20_PS/2 mouse/keyboard connector
- 22_21_IR Receiver
- 24_One High Speed Mezzanine Card



در ادامه به جزئیات بیشتری از هر بلاک می پردازیم.

1 **FPGA device**

- Cyclone IV EP4CE115F29 device
- 114,480 Les
- 432 M9K memory blocks
- 3,888 Kbits embedded memory
- 4 PLLs

3_Memory devices

- 128MB (32Mx32bit) SDRAM
- 2MB (1Mx16) SRAM
- 8MB (4Mx16) Flash with 8-bit mode
- 32Kb EEPROM

5 Clock

- Three 50MHz oscillator clock inputs
- SMA connectors

7 SD Card socket

• Provides SPI and 4-bit SD mode for SD • 16x2 LCD module

9_Connectors

- Two Ethernet 10/100/1000 Mbps ports
- High Speed Mezzanine Card (HSMC)
- Configurable I/O standards (voltage levels:3.3/2.5/1.8/1.5V)
- USB type A and B
- 40-pin expansion port
- VGA-out connector
- DB9 serial connector for RS-232 port with flow control
- PS/2 mouse/keyboard

2_FPGA configuration

- JTAG and AS mode configuration
- EPCS64 serial configuration device
- On-board USB Blaster circuitry

4 Switches and indicators

- 18 slide switches and 4 push-buttons
- 18 red and 9 green LEDs
- Eight 7-segment displays

6 Audio

- 24-bit encoder/decoder (CODEC)
- Line-in, line-out, and microphone

8_Display

آزمایش ۲

طراحی و پیادهسازی مدارهای ترکیبی و ترتیبی با استفاده از امکانات شماتیک

هدف

- آشنایی با مدارهای ترکیبی و ترتیبی، نحوه طراحی و توصیف سطح گیت
 - طراحی یک مدار با استفاده از فلیپ فلاپ
 - شبیهسازی طرح ایجادشده با استفاده از نرم افزار Quarus

شرح آزمایش

به عنوان اولین آزمایش جهت مرور مفاهیم مدارهای ترکیبی و ترتیبی (که در دروس معماری کامپیوتر و طراحی مدارهای خودکار دیجیتال توضیح داده شده است) قصد داریم با استفاده از فلیپ فلاپ نوع JK یک مدار ترتیبی را طراحی و طرح موردنظر را با استفاده از نرم افزار Quarus شبیهسازی کنیم. توضیح: بعد از توضیحات مراحل آزمایش نحوه شبیهسازی طرح با استفاده از نرمافزار Quarus توضیح داده

۱- مرحله اول

خواهدشد.

ابتدا با استفاده از جدول صحت فلیپ فلاپ JK ترتیب شمارش مدار زیر را به صورت تئوری بدست آورید.



شکل۳- شماتیک شمارنده خاص

۲-مرحله دوم

حال با کمک امکانات شماتیک نرم افزار Quarus مدار را در محیط نرم افزار طراحی کنید.

۳-مرحله سوم

در این قسمت از آزمایش شماتیک طراحی شده را شبیهسازی نموده و صحت عملکرد آن را با نتایج بدست آمده از مرحله اول مقایسه کنید.

خروجیهای مورد انتظار (برای تهیه گزارش کار)

- بیان دقیق مساله و بررسی جدول کارنو
- نکات کلیدی در نرم افزار Quarus با توجه به مطالب مطرح شده در کلاس
 - مفاهیم طراحی سنکرون و آسنکرون و نحوه پیادهسازی آنها
- مشکلاتی که در حین طراحی شماتیک و شبیهسازی با آنها مواجه شدید.

تمرين

با استفاده از فلیپ فلاپ T، شماتیک شمارنده زیر را طراحی کرده و صحت عملکرد آن را شبیهسازی کنید.



نحوه شبیهسازی طرح با استفاده از نرم افزار Quarus

Quartus II امکان شبیه سازی مدارات طراحی شده را برای شما فراهم می کند. در این قسمت، هم به منظور توضیح این ویژگی در نرم افزار Quartus و هم به منظور اطمینان حاصل کردن از درستی عملکرد طراحی، مدار خود را شبیه سازی می کنیم. به منوی New جاه مراجعه کرده و این بار به جای فایل های طراحی به بخش فایل های درستی سنجی/درستی یابی رفته و گزینه Vector Waveform File را انتخاب کنید.



شكل۴-گزينه طراحي شبيهسازي

بعد از انتخاب گزینه Vector Waveform File، پنجرهی مربوط به شکل موج باز می شود.



شکل ۵-پنجره شکل موجها

سپس گزینه Edit→Insert Node or Bus را همانند شکل زیر انتخاب کنید. بدین ترتیب که گرههای مورد نیاز در طراحی را انتخاب کرده و سیگنالها یا پورتهای ورودی/خروجی و یا هر موجودیتی را که خواستار حضور آن در شبیهسازی هستید را به شکل موج خود بیفزایید.

🔇 Quartı	is II - C:/Users	/mohsen/Des	ktop/Qu	artus/test -	test - [\	Wavefo	orm1.v	1
🖸 File	Edit View	Project As	signmer	nts Process	ing T	ools	Wind	c
0 🗃	⊮⊃ <u>U</u> ndo	Ctrl+	Z	⊂≃ test				
JJ Project Na	⊂≊ <u>R</u> edo	Ctrl+	Y		→ x	B		
Entity	X Cut	Ctrl+	x	LC Registers	Merr		gate.c))
🛆 Cyc	Depy	Ctrl+	c					٧
BDF	<u>P</u> aste		- • -	0 (0)	0		Аſ	
	🗙 <u>D</u> elete	D	el				Ð	
	Se <u>l</u> ect						Ì	-
	Find and	Replace	•					
	→ <u>G</u> o To	Ctrl+	G			# ¶	Si∳B	
						Vīī .	<u>×.</u>	
	Ins <u>e</u> rt		•	Insert Noc	le or <u>B</u> u	5		
	<u>V</u> alue			Insert Way	eform <u>l</u>	<u>D</u> ivider		
	G <u>r</u> ouping	1	•	Insert Cop	ied Noo	des		
	Time <u>B</u> ar		•	Insert <u>W</u> av	reform I	Interval	I	
	End Time	·						
Tasks	Grid Size		- E		∃ × ×			
Flow: Co	≜l Sort				•	<u>X?</u>)	XB	
Task 🖌	Group an	d Bus Bit <u>O</u> rd	er 🖡 🗋	Tin	ne 🛈 🛛	巺	≜↓	
	Propertie	5	-		00.01			
				00:	00:01			

شکل۶- گزینه Insert Node or Bus

بعد از انتخاب گزینه Insert Node or Bus پنجرهی همانند شکل زیر باز می شود.

Insert Node o	r Bus		×				
Name:	l		ОК				
Туре:	INPUT	•	Cancel				
Value type:	9-Level	-	Node Finder				
Radix:	Binary	•					
Bus width:	1						
Start index:	0						
Display gray code count as binary count							

شكل ٧- پنجره Insert Node or Bus

در این قسمت می توانید نام پین را در بخش Name بنویسید اما بهتر است از Node Finder استفاده نمایید.

amed: ×	▼ Filter: Pins: all	Custo	mize List	ОК
ook in: Igatel		💌 🗹 Include	subentities Stop	Cancel
odes Found:		Selected Nodes:		
Name	Assignments	Name	Assignments ¹	
a	Unassigned I	🗩 Igatela	Unassigned I	
▶b	Unassigned I	🕪 Igatelb	Unassigned I	
C	Unassigned I	🕩 gate c	Unassigned I	
Do 🕑	Unassigned (🐵 Igatelo	Unassigned (
		- 1		
	_	>		
		>>		
		<		
	_			
< III	•		•	

بر روی گزینه Node Finder کلیک کنید تا صفحهی همانند شکل زیر ایجاد شود.

شكل ٨- پنجره Node Finder

ابتدا از قسمت Filter گزینه Pind:all را انتخاب کنید. در ادامه بر روی گزینه List کلیک کنید تا تمام ورودی و خروجیهای مربوط به طرح را در قسمت Nodes Found برای شما نمایش دهد. سپس ورودی و خروجیهای مورد نظر را انتخاب، و روی علامت < کلیک کنید تا به قسمت Selected Nodes همانند شکل بالا اضافه شود. در ادامه با زدن گزینه OK شکل زیر نمایان می شود که در این قسمت می توانید تنظیماتی دیگر همانند انتخاب Radix

Insert Node of	r Bus		×
Name:	**Multiple Items**		OK
Туре:	**Multiple Items**	•	Cancel
Value type:	9-Level	-	Node Finder
Radix:	Binary	-	
Bus width:	1		
Start index:	0		
🗖 Display gr	ay code count as binary count		

شكل٩- پنجره Insert Node or Bus

بعد از انجام تنظیمات مربوطه و دلخواه گزینه OK را انتخاب کنید. در ادامه شکل زیر ظاهر می شود. همان گونه که مشاهده می کنید تمام سیگنال های ورودی و خروجی نمایان شدهاند اما شکل موجهای خروجی هنوز بصورت هاشور خورده باقی مانده که مقدار آن در حین شبیه سازی مشخص می شود.

rì Quartu	us II - C:	/Users/mol	Tools W	p/Quartus/test - test - [Waveform1.vwf*]		×
	Master	Time Bar:	20.25 ns	Pointer: 22.45 ns Interval: 2.2 ns	Start: End:	
ЪА ЖФ		Name	Value at 20.25 ns	0 ps 10.0 ns	20.0 ns 20.25 r	is
	0	а	B 0			
	■1	ь	BO			
apan 2.58	<u>∎</u> >2	c	BO			
XT 🗻	•🔊 3	0	BX		***************************************	*****
<u>e</u> 1.						
z Xw						
(C X (L)						
2 XB						
5. ∳1						
π. 4 τ						
	-		Þ	<		Þ
or Help, j	press F1				NUM	

شكل ١٠- پنجره شكل موجها

در ادامه فایل را در پوشه پروژه ذخیره نمائید. مدار طراحی شده میتواند به دو روش شبیهسازی شود. در روش سادهتر سیم های ارتباطی ایده ال و بدون تاخیر انتشار در نظر گرفته شده که به آن شبیهسازی عملکردی می گویند. در روش دوم که پیچیده میباشد، تمام تاخیر انتشارها محاسبه شده که به آن شبیهسازی زمانی گفته می شود. در روش دوف ما در این قسمت شبیه سازی عملکردی میباشد. برای اجرای این نوع شبیه سازی از قسمت میشود. هماند شکل زیر گزینه Simulator Tool را انتخاب نمایید.

Processing Tools Window	Help							
Stop Processing	Ctrl+Shift+C							
Start Compilation	Ctrl+L							
Analyze Current <u>F</u> ile								
St <u>a</u> rt	•							
Update Memory Initializati	on File							
Compilation <u>Report</u>	Ctrl+R							
Start Compilation and Simulation Ctrl+Shift+K								
Generate Functional Simul	ation Netlist							
🚬 Start Si <u>m</u> ulation	Ctrl+I							
Simulation Debug	►							
Simulation Report	Ctrl+Shift+R							
Compiler Tool								
Simulator Tool								
Classic Timing Analyzer To	lool							
💋 PowerPlay Power Analyzer	Tool							
V SSN Analyzer Tool								

شكل ۱۱ –انتخاب گزينه Simulator Tool

🕽 Simulator Tool							
Simulation mode: Functional Generate Functional Simulation Netlist							
Simulation input: test.vwf Add Multiple Files							
Simulation period Bun simulation until all vector stimuli are used							
C End simulation at: 100 ns							
- Simulation options Automatically add pins to simulation output waveforms Check outputs Waveform Comparison Settings							
Setup and hold time violation detection							
Glitch detection: 1.0 ns 💌							
Overwrite simulation input file with simulation results							
Generate Signal Activity File:							
Generate VCD File:							
0 %							
00:00:00							
🚬 Start 💿 Stop 😲 Open 🔮 Report							

بعد از انتخاب گزینه Simulator Tool شکلی همانند شکل زیر ظاهر میشود.

شكل١٢-تنظيم پنجره Simulator Tool

سپس در قسمت Simulation Mode گزینه Functional را انتخاب و گزینه Simulation Mode کمیک معدار دهید. Save Save کنید و به سیگنالهای ورودی مقدار دهید. Save کنید و سپس بر روی گزینه Start کلیک کنید و پس از اتمام با موفقیت بودن و بدون خطا میتوانید برای دیدن خروجی بر روی گزینه Report کلیک کنید.

R	Master	Time Bar:	5	i.125 ns	:	• •	Pointer:		1.0 ns		Interva	al:	-4.1	3 ns	9	Start:
A ₹		Name	Value at 5.13 ns	0 ps 5.12	10.0 ns 5 ns	20.0 ns	30.0 ns	40.0 ns	50.0 ns	60.0 ns	70.0 ns	80.0 ns	90.0 ns	100 _, 0 ns	110,0 ns	120.
<u> </u>	0<1	а	B 1													
	▶1	b	A 1						<u>, Г</u>							
đâ	₽ 2	с	A 1				-									
₩.,	@ 3	0	A 1													
→																
₽;; 2 ↓																

شكل١٣- پنجره شكل موجها و نتايج نهايي شبيه سازي

آزمایش ۳

طراحي يك مدارجمع كننده كامل

هدف

- یادآوری کدنویسی VHDL و استفاده از ساختار سلسه مراتبی
 - شبیهسازی طرح ایجاد شده با استفاده از نرم افزار Quarus
- نحوه پروگرم کردن طرح بر روی برد DE2 و استفاده از Switcheها و Ledهای تعبیه شده بر روی برد

شرح آزمایش

در این آزمایش قصد داریم با استفاده از طراحی یک مالتی پلکسر 1*2 یک مدار جمع کننده کامل طراحی کنیم.

در ادامه ضمن شبیه سازی طرح با استفاده از نرم افزار Quarus آنرا بر روی برد مربوطه آزمایشگاه پروگرم کرده و صحت عملکر آنرا با استفاده از سوئیچها و LEDهای تعبیه شده بر روی برد بررسی میکنیم.

توضیح: بعد از توضیحات مراحل آزمایش نحوه پروگرم کردن طرح بر روی برد DE2 توضیح داده خواهد شد.

۱- مرحله اول

در این مرحله از آزمایش ابتدا با استفاده از شماتیک زیر، برنامه یک مالتی پلکسر 1*2 را بنویسید.



۲- مرحله دوم

در ادامه آزمایش با استفاده از کدنویسی سلسه مراتبی و کد نوشته شده در مرحله اول، کد VHDL مدار جمع کننده زیر را بنویسید.



شكل١٥- شماتيك جمع كننده كامل

А	В	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

جدول ۱ - جدول صحت جمع کننده کامل

۳-مرحله سوم در این قسمت از آزمایش برنامه نوشته شده را شبیهسازی نموده و صحت عملکرد آن را با جدول مرحله دو مقایسه کنید.

۴-مرحله چهارم بعد از شبیهسازی و تایید درستی برنامه، آنرا بر روی برد مربوطه آزمایشگاه پروگرم کرده و صحت عملکرد آن را با استفاده از سوئیچها و LEDهای تعبیه شده بر روی برد بررسی کنید.

Using LEDs

There are 27 user-controllable LEDs on the DE2-115 board. Eighteen red LEDs are situated above the 18 Slide switches, and eight green LEDs are found above the push-button switches (the 9th green LED is in the middle of the 7-segment displays). Each LED is driven directly by a pin on the Cyclone IV E FPGA; driving its associated pin to a high logic level turns the LED on, and driving the pin low turns it off. **Figure 4-9** shows the connections between LEDs and Cyclone IV E FPGA.



Figure 4-9 Connections between the LEDs and Cyclone IV E FPGA

شکل۱۶- نحوه اتصال LEDهاای برد DE2

Using Push buttons and Switches

The DE2-115 board provides four push-button switches as shown in Figure 4-6. Each of these switches is debounced using a Schmitt Trigger circuit, as indicated in Figure 4-7. The four outputs called KEY0, KEY1, KEY2, and KEY3 of the Schmitt Trigger devices are connected directly to the Cyclone IV E FPGA. Each push-button switch provides a high logic level when it is not pressed, and provides a low logic level when depressed. Since the push-button switches are debounced, they are appropriate for using as clock or reset inputs in a circuit.





شکل۱۷- نحوه اتصال Switcheهای برد DE2

خروجی های مورد انتظار (برای تهیه گزارش کار)

- مفاهیم طراحی، توصیف و پیاده سازی طرح با VHDL
 - گزارش سنتز طرح
- مشکلاتی که در حین برنامه نویسی، شبیه سازی و پروگرم کردن با آنها مواجه شدید.

تمرين

با استفاده از طراحی یک دیکدر 8*3 یک جمع کننده کامل ساخته و صحت عملکرد آن را با استفاده از شبیهسازی بررسی نمایید.

نحوه پروگرم کردن برنامه بر روی برد DE2 شرکت Altera

پیادهسازی فرآیندی شامل ترجمه ،ترسیم ،جایابی و مسیریابی است. در این قسمت پیادهسازی روی برد DE2 نیادهسازی فرآیندی شامل ترجمه ،ترسیم ،جایابی و مسیریابی است. در این قسمت پیادهسازی روی برد Quartus تعریف انجام می شود. اما روند مشابهی برای برنامهریزی هر IFPGAای از شرکت ALTERA در Quartus تعریف شدهاست که در این قسمت به آن می پردازیم.

در ابتدای کار بایستی دستگاه مقصد را از حالت پیش فرض به دستگاه مقصد برد DE2 تغییر دهید. می توانید این کار را در ابتدای زمان تعریف پروژه انجام دهید و یا اگر از آن مرحله عبور کردهاید از طریق Assignment>Device و برای دستگاه Device این پنجره در قسمت خانواده IC و برای نوع دستگاه دستگاه دستگاه کنید و تایید را بزنید.

•
•
•
•
tible only 🔒
9-bit elements
_
•
-

شکل۱۸- پنجره انتخاب Device

درادامه فرآیند تخصیص پین انجام میشود. برای انتساب پینها از ابزار Assignment Editor استفاده میشود. بدین منظور از مسیر Pins + Pins را انتخاب و در لیست ارائه شده Pin را کلیک نموده و بعد NEW را که پررنگ شده، دوبار کلیک کنید. یک پنجره پایینرو ظاهر میشود. روی هر ورودی یا خروجی که میخواهید، کلیک کنید تا در جدول قرارگیرد. سپس با دوبار کلیک آن، پنجره دیگری باز میگردد.

🧭 A	ssignmen	t Editor					_ • •			
Ø	Category: All Pin & Timing + Logic Options									
	🕅 🕀 Information: The Assignment Editor is the interface for creating, editing, and viewing individual assignments, including pin assignments, in the Quartus II software. To creating									
8		From	То	Assignment Name	Value	Enabled				
TXT	1		iii>a	Location	PIN_30	Yes				
F	2		r∰ c	Location	PIN_12	Yes				
12	3	< <new>></new>	< <new>></new>	< <new>></new>						
-8										
ŧ										
900										
2										

شكل ۱۹- پنجره Assignment Editor

حالا هر پین واقعی روی FPGA را که میخواهید، انتخاب نمایید تا انتساب صورت گیرد. به همین ترتیب بقیه ورودی و خروجیها را به پینهای FPGA تخصیص دهید. اکنون با استفاده از مسیر Save این فایل تخصیص پینها را ذخیره نمایید. بار دیگر طرح خود را که پینهای واقعی و موردنظر شما را برای ورودی و خروجیها در نظر گرفته شده است کامپایل نمایید.

Location		I,IO Bank		I/O Standard		General Function	Spe
PIN_N26	-			LVTTL			
PIN_MZ4		I/O Bank 5	Row I C)	L/DS125p		-
PIN_M25		I/O Bank 5	Row I C)	LVDS125n		
PIN_N1		I/O Bank 2	Dedicate	ed Clock	CLK1, LVDS	CLKOn, Input	
PIN_N2		I/O Bank 2	Dedicate	ed Clock	alko, LVDS	CLKOp, Input	
PIN_N9		I/O Bank 2	Row I C)	L/DS31p		
PIN_N18		I/O Bank 5	Row I/C)	L/DS110p		
PIN_N20		I/O Bank 5	Row I C)	LVDS124p		
PIN_N23		I/O Bank 5	Row I/C)	LVDS126p, I	DPCLK7/DQS0R/CQ1R	
PIN_N24		I/O Bank 5	Row I C	>	LVDS126n		
PIN_N25		I/O Bank 5	Dedicate	ed Clock	alk4, LVDS	CLK2p, Input	
PIN_N26		I/O Bank 5	Dedicate	ed Clock	CLK5, LVDS	CLK2n, Input	
PIN_P1		I/O Bank 1	Dedicate	ed Clock	alks, LVDS	CLK1n, Input	
PIN_P2		I/O Bank 1	Dedicate	ed Clock	CLK2, LVDS	CLK1p, Input	
PIN_P3		I/O Bank 1	Row I C	>	LVDS26p, D	PCLK1/DQS1L/CQ1L#	
DIN D4		I/O Back 1	Row IC		LVDS26p		

شکل ۲۰- پنجره تخصیص Pinها

برای برنامهریزی دستگاه ابتدا سیمهای Power و اتصال به کامپیوتر برد را وصل کنید.



سپس از مسیر Tools -> Programmer پنجره برنامهریزی را باز کنید.

در ادامه صفحه برنامهریزی بر روی تنظیمات سخت افزار بروید و در بخش سختافزارهای موجود گزینه -USB

Blaster USB را انتخاب كنيد.

Hardware Settings JTAG Settings elect a programming hardware setup ardware setup applies only to the cur	to use when prog	ramming devices window.	s. This programming
Currently selected hardware: USB-E	Blaster [USB-0]		-
Hardware	Server	Port	Add Hardware
US8-Blaster	Local	US8-0	Remove Hardware
			Close

شكل٢٢- پنجره تنظيمات كابل

در ادامه باید فایل پروژه خود را به این قسمت اضافه کنید. بنابراین روی گزینه افزودن فایل کلیک کرده و در بخش فایل های خروجی دنبال فایلی با نام طراحی شما و با پسوند sof. بگردید و آن را باز کنید.

🐌 Programmer - C	:/Users/Parastoo/Desktop/FA/F	A - FA - [Chain2.cd	ŋ					
File Edit View	Processing Tools Window	Help 💎				Sea	rch altera.co	xm 🚯
Hardware Setu	DUSB-Blaster [USB-0] ISP to allow background programm	Mode: ing (for MAX II and M	JTAG MAX V devices)	•	Progress:			
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
Auto Detect	Select Programming File	astoo\Desktop\FA\o	utput files		- G	00		×
Add File Change File Change File Add Device Add Device	My Computer Parastoo	pof sof	waynot _ Mica					
	File name: FA.sof Files of type: Programming File	s (*.sof *.pof *.jam	*.jbc *.ekp *.jic)				Open Cancel	

شکل۲۳- پنجره انتخاب فایل Sof

سپس در صفحهی اصلی آن فایل آن را انتخاب و گزینهی Start را بزنید. اگر در روند برنامهریزی خطایی وجود نداشته باشد عملیات با موفقیت انجام خواهد شد.



شکل۲۴- برنامهریزی دستگاه

در این مرحله کار به اتمام میرسد. شما تا اینجا با عملیات پایهای در Quartus آشنا شدید و میتوانید سخت افزار خود را طراحی و روی برد DE2 آن را تست کنید.

آزمایش ۴

طراحی ضرب کننده و تکنیکهای برنامه نویسی VHDL

هدف

- آشنایی با الگوریتم ضرب کننده Booth و طراحی آن
- شبیهسازی طرح ایجاد شده با استفاده از نرم افزار Quarus
 - نکاتی برای نوشتن کدهای VHDL بزرگ

شرح آزمایش

در این آزمایش قصد داریم ابتدا با استفاده از ساختار ترتیبی کدهای VHDL یک نوع ضرب کننده به نام Booth طراحی کرده و در ادامه با استفاده از تکنیکهای برنامه نویسی VHDL کد نوشته شده رو از نظر سنتز کاری بهینهسازی کنیم.

امروزه نیاز به فیلترهای دیجیتال سریعتر و با توان مصرفی پایینتر در بسیاری از زمینهها به ویژه الگوریتمهای مخابراتی کاملاً مشهود است. ضربکننده دیجیتال یکی از عناصر اصلی فیلتر دیجیتال میباشد. بنابراین برای بهینهسازی یک فیلتر دیجیتال، باید ضربکننده آن بهینه باشد. الگوریتمهای زیادی برای بهینهسازی مربکنندهها ارائه شده است که ضعف اصلی این الگوریتمها در عدم توانایی برای ضرب اعداد منفی میباشد. الگوریتمهای زیادی برای میباسد. الگوریتمهای زیادی برای میبینهسازی مربکنندهها ارائه شده است که ضعف اصلی این الگوریتمها در عدم توانایی برای ضرب اعداد منفی میباشد. الگوریتمهای زیادی برای بهینهسازی الگوریتم بوث برای ضرب اعداد منفی میباشد. الگوریتمهای زیادی برای ضرب اعداد منفی میباشد. الگوریتم بوث برای ضرب دو عدد مکمل ۲ استفاده میگردد. اگر اعداد مورد نظر n بیتی باشند، آنگاه ثبات n بیتی X برای ذخیره مضروب استفاده میگردد و علاوه بر این دو ثبات n بیتی A و Q نیز مورد استفاده قرار میگیرد که مقدار اولیه آن برابر صفر خواهد شد. می میشود. همچنین فلیپ فلاپ I - Q با یکدیگر برابر باشند هیچ عمل جمع و تفریقی انجام نمیشود و در نهای میشود و در نهای می مورد و میرون که مقدار اولیه ثبات A برابر صفر در نظر گرفته میشود. همچنین فلیپ فلاپ I - Q نیز مورد استفاده قرار میگیرد که مقدار اولیه آن برابر صفر در نظر گرفته میشود. همچنین فلیپ فلاپ I - Q نیز مورد استفاده قرار میگیرد که مقدار اولیه آن برابر صفر در مورد. که مو در این ایگوریتم با توجه به الگویQ = Q میل جمع یا تفریق را انجام میدهد. در صورتی میک دو بیت Q = I - Q با یکدیگر برابر باشند هیچ عمل جمع و تفریقی انجام نمیشود و در نهایت در پایان هر میک دو بیت Q = I - Q با یکدیگر برابر باشند هیچ عمل جمع و تفریقی انجام نمیشود و در نهایت در پایان هر میک ثباتهای می و نوان یک ثبات واحد و به شکل I = Q = Q میک شرع و می نیز می انجام میده. در مورتی سیکل ثباتهای A و و فلیپ فلاپ I - Q به عنوان یک ثبات واحد و به شکل I = Q = Q شیفت پیدا میکند.

نحوه عملکرد مدار کنترلی با توجه به الگوی $Q_0 Q_{-1}$ در هر سیکل به صورت زیر میباشد.



شكل۲۵- فلوچارت ضربكننده Booth

ا ا اگر $Q_0 Q_{-1} = 0$ یا $Q_0 Q_{-1} = 0$ باشد آنگاه عمل جمع یا تفریق رخ نمیدهد و ثبات A $Q Q_{-1}$ به صورت $Q_0 Q_{-1} = 0$ به صورت حسابی شیفت داده می شود.

۲-اگر $Q_0 Q_{-1=01}$ با محتوای X جمع می گردد و نتیجه در A قرار می گیرد، یعنی عمل A انجام و سپس $Q_0 Q_{-1}$ شیفت داده می شود. A انجام و سپس $A Q Q_{-1}$

A=A-X اگر $Q_0 Q_{-1=10}$ باشد ابتدا ثبات X از ثبات A کسر می گردد و نتیجه در A قرار می گیرد، یعنی عمل A=A-X انجام و سپس $Q_0 Q_{-1}$ می شود.

تذکر: در شیفت حسابی به راست، بیت علامت (MSB) تغییر نمی *ک*ند. در نهایت پس از n سیکل نتیجه ضرب در ثبات AQ خواهد بود.

۱- مرحله اول

در این مرحله از آزمایش ابتدا برنامه VHDL طرح را با استفاده از دستورات ترتیبی بنویسید.

۲-مرحله دوم

در این قسمت از آزمایش برنامه نوشته شده را شبیهسازی نموده و صحت عملکرد آن را بررسی نمایید.

۳–مرحله سوم

بعد از شبیهسازی و تایید درستی برنامه، آنرا بر روی برد مربوطه آزمایشگاه پروگرم کرده و صحت عملکرد آن را با استفاده از سوئیچها و LEDهای تعبیه شده بر روی برد بررسی کنید.

حال در این قسمت از آزمایش قصد داریم در مورد تکنیکهای برنامه نویسی VHDL به منظور بهینهسازی سنتز کاری صحبت کنیم.

زبان VHDL در ابتدا به منظور شبیهسازی و مستندسازی طراحی شد. در اواخر دهه ۸۰ اولین سنتزکنندهها توانستند قسمتهایی از VHDL استاندارد را به Net List تبدیل کنند. در اوایل دهه ۹۰ این روند سرعت بیشتری یافت تا امروزه که سنتز در اغلب زبانهای توصیف سختافزار انجام میشود. VHDLروشهای مختلف توصیف را پشتیبانی میکند. این روشها از نظر اینکه تا چه حد به سختافزار نزدیک هستند طبقهبندی میشوند و به آنها سطوح خلاصه سازی Level Of Abstraction میگویند. سطوح مختلف خلاصه سازی همانطور که در شکل زیر دیده میشود، رفتار، جریان داده و ساختار میباشند.

Behavior	Performance specification	Ţ
Data flow	Sequential Descriptions State machines Register Transfers Selected assignments Arithmetic Operations	Levels of abstraction
Structure	Boolean equations Hierarchy Physical information	
		·

شکل۲۶- سطوح خلاصهسازی

بالاترین سطح در VHDL بیان رفتاری مدار است. در این سطح مدار بر اساس عملکرد در زمان، توصیف می شود. در سطح جریان داده توصیف مدار براساس چگونگی عبور اطلاعات از سیستم انجام می شود. از آن جا که رجیسترها قسمت عمده مدارات دیجیتال هستند اغلب به این سطح، منطق گذار رجیسترها RTL گفته می شود. پایین ترین سطح، سطح ساختار است که توصیف در آن با توجه به اجزای تشکیل دهنده مدار صورت می گیرد. در بعضی از کاربردهای سنتز، تقسیم بندی در چهار سطح tayout یا RTL و Behavioral انجام می شود. سنتز HEEE استاندارد در VHDL امکان پذیر نیست و تنها زیر مجموعه ای از دستورات توسط سنتز کنندهها پشتیبانی می شوند و از این نظر تفاوت هایی در انواع سنتز کننده ها وجود دارد. ورودی های یک سنتز کننده، فایل VHDL، فایل تکنولوژی و محدودیت ها می باشد و خروجی یک فایل گزارش و

یک Net list است. Net list یک شماتیک از گیتهاست و فایل گزارش نحوه انجام عملیات سنتز را شرح می-دهد.

الگوریتم پروسه سنتز را میتوان به صورت زیر نشان داد.



الف: محدوديتها

تمام سنتزکنندهها بهینهسازی را بر اساس محدودیتهایی که توسط طراح مشخص می شود انجام میدهند. این محدودیتها به سه گروه عمده تقسیم می شوند:

- ۱ قواعد طراحی
- ۲- محدودیتهای تایمینگ
 - ۳- محدودیتهای سطح

ترتیب هریک از محدودیتها به دلخواه قابل تغییر است اما سنتزکنندهها معمولا تقدم بالا را به صورت پیش فرض اعمال میکنند و قیدهای تایمینگ تقدم بالاتری نسبت به قیدهای سطح دارند. یک انتخاب دیگر تغییر تکنولوژی است که این روش نیز معمولا گران میباشد.

۱–قواعد طراحی

محدودیتهای طرح قیدهایی هستند که توسط کمپانیهای سازنده تعریف می شوند و هر سازنده محدودیتهای مختلفی را تعریف می کند. مانند fan out ،fall time ،rise time و ظرفیت اتصالات و.... اگر این محدودیتها اعمال نشوند طرح سنتز شده قابل ساخت نخواهد بود و تصحیح دستی برای طرحهای بزرگ غیرممکن می شود.

۲- محدودیتهای تایمینگ

به منظور بهینهسازی طرح از نظر تایمینگ موارد زیر باید برای سنتزکننده مشخص شود:

✓ کلاک ورودی و پریود آن
 ✓ تاخیر ورودی نسبت به کلاک
 ✓ تاخیر خروجی نسبت به کلاک
 ✓ تاخیر هر پین به پین دیگر
 ✓ تمام مسیرهای نادرست
 ۸ تمام می مسیرهای نادرست
 ۷ ولتاژ تغذیه
 ✓ محدوده دما
 ✓ مدروده دما

اگر با توجه به تابع تبدیل مدار بدانیم مثلا در شرایط عادی مسیری از ورودی a به خروجی b وجود ندارد باید آن را بصورت یک مسیر نادرست تعریف کنیم تا از بررسی غیرضروری آن توسط سنتزکننده جهت بهینهسازی اجتناب شود. خصوصا در طراحی برای FPGA لازم است مسیر نادرستهای موجود مشخص شود زیرا مسیریاب از این اطلاعات در تقدم بخشیدن به مسیرهای داده استفاده میکند.

۳-محدودیتهای سطح

پس از اجرای تمام محدودیتهای تایمینگ روی مدار قدم بعدی اعمال محدودیتهای مربوط به سطح مدار است. معمولا این محدودیت با معادل NAND مشخص می شود و اگر قیدی مشخص نشود سنتزکننده، بهینه سازی مدار از نظر سطح را اجرا نخواهد کرد.

ب: موارد مورد توجه در سنتز

۱-سنتزکنندههای پیشرفته الگوریتمهای مختلفی دارند و ممکن است نتایج بهینهسازی هر یک کاملا متفاوت باشد.

۲-اگر مساله تایمینگ یا سطح در ماشین حالت قرار دارند می توان آن را توسط State Machine Optimizer که در ساختار داخلی سنتزکننده ها قرار دارد بهینه کرد. این روش معمولا نتایج بهتری روی سطح و تایمینگ خواهد داد.

۳-اگر لازم باشد کد VHDL بازنویسی شود دو انتخاب اختیار طراح است:

✓ تغییر کد VHDL بدون تغییر تابعی آن

✓ تغییر تابعی کد VHDL

اغلب ممکن است لازم باشد کد VHDL با تغییر توابع آن دوباره بازنویسی شود. خصوصا اگرسنتزکننده قوی نباشد. چنین سنتزکنندهای با توجه به روشی که کد VHDL برای آن نوشته می شود سنتزهای متفاوتی خواهد داد. سنتزکنندههای پیشرفته به نوع نوشتن کد حساس نیستند اما وقتی طرح پیجیده می شود مانند دیگر سنتزکنندهها رفتار می کنند.

۴-تایمینگ در مدارهای سی ماسی ASIC و APGA تابعی از ولتاژ منبع تغذیه است. مـثلا اگـر ولتـاژ ۵٫۲۵ تـا ۴٫۷۵ ولت باشد تایمینگ مدار نسبت به وقتی که ولتاژ ۵٫۵ تا ۴٫۵ ولت است ۱۰ تا ۳۰ درصد بهبود مییابد. ۵-تایمینگ در مدارهای سی ماسی ASIC و APGA تابعی از دما میباشد. هرچه سیلیکون گرمتر شود گیتها کندتر میشوند. اگر دمای محیط ۲۵ درجه سانتیگراد را با دمـای ۷۰ درجـه سـانتیگراد مقایسـه کنـیم فـاکتور کارایی ۱٫۵ تا ۲ برابر تغییر خواهد داشت.

پ: تکنیکهای برنامه نویسی VHDL

طراحان و پیادهسازان حرفهای سیستمهای دیجیتال مبتنی بر FPGA، برای اینکه در اجرای طرحهای بزرگ دچار مشکل نشوند از چه تکنیکهایی استفاده میکنند؟ حتما شما هم به مشکلات زیادی در طراحی و پیادهسازی کدهای کوچک و متوسط برخوردهاید. مثلا:

- کد نوشته شده سنتز نمیشود.
- فركانس قابل اعمال به مدار، خيلي كمتر از مقدار مورد نظر است.
 - فضای اشغال شده در FPGA بیش از حد انتظار است.
- رفتار مدار پیادهسازی شده در FPGA با رفتار شبیهسازی متفاوت است.
 - زمان سنتز کد، بیش از حد زیاد است.
- ایراداتی در عملکرد مدار روی FPGA دیده میشود که به صورت منطقی قابل توجیه نیست. در این قسمت، شما را با تعدادی تکنیک کدنویسی آشنا میکنیم. این تکنیکها به شما کمک میکنند کدهایی

بنویسید که علی رغم حجم و پیچیدگی زیاد:

- به راحتی سنتز شوند.
- به بیشترین فرکانس کلاک قابل اعمال برسند.
- کمترین حجم منابع را در FPGA اشغال کنند.

تکنیک کدنویسی۱: همه پورتهای ورودی و خروجی را بافر کنید.

بافر کردن پورتهای ورودی به این معنی است که به ازای هر کدام از این پورتها یک سیگنال داخلی تعریف کنید و در ابتدای قسمت ترتیبی یعنی Process، پورتهای ورودی را به این سیگنالهای داخلی ارجاع دهید. پس از آن، دیگر در هیچ جای کد از پورتها استفاده نکنید و به جای آن از سیگنال داخلی متناظرش استفاده کنید.

بافر کردن پورت خروجی به این معنی است که به ازای هر کدام از این پورتها یک سیگنال داخلی تعریف کنید و به جای اینکه در Process به صورت مستقیم به خروجی مقدار بدهید، به این سیگنالهای داخلی ارجاع دهید. نهایتا در محیط Concurrent باید سیگنالهای داخلی را به پورتهای خروجی متناظر ارجاع دهید. بافر کردن پورتهای ورودی و خروجی به نرم افزار سنتز کمک میکند تا جانمایی و مسیریابی را بهتر انجام دهد و مدار حاصل دارای فرکانس کلاک قابل اعمال بیشتری خواهد بود.

تکنیک کدنویسی۲: همه کلاک های سیستم را به صورت دستی بافر کنید.

برای پیادهسازی مدارات ترتیبی، حتما نیاز به سیگنال کلاک دارید. این سیگنال معمولا توسط اسیلاتوری روی برد ساخته میشود و از طریق یکی از پایههای مخصوص FPGA وارد آن میشود. گاهی هم ممکن است بیش از یک سیگنال کلاک از بیرون به FPGA وارد شود. در هر صورت، همه این سیگنالهای کلاک ابتدا باید از یک بافر مخصوص به نام IBUFG عبور کنند. در بیشتر موارد، نرم افزار سنتز به طور خودکار سیگنالهای کلاک را تشخیص داده و آنها را از یک IBUFG عبور میدهد. اما در بعضی موارد ممکن است این اتفاق رخ ندهد. بنابراین، پیشنهاد میشود که طراح، تمام سیگنالهای کلاک را به صورت دستی از بافر IBUFG عبور دهد. برای این کار، کافیست در محیط کنارت زیر را بنویسید:

IBUFG_inst : IBUFG generic map (IBUF_LOW_PWR => TRUE IOSTANDARD => "DEFAULT") port map (O => Output_Clock, -- پايد خروجى بافرشده كلاک -- I => Input_Clock); پايه ورودى كلاک --

تکنیک کدنویسی ۳: از بلوک تولید کلاک استفاده کنید، حتی اگر نیاز به تغییر فرکانس ندارید. در FPGAها بلوکهای از پیش آمادهشدهای وجود دارند که به کمک آنها میتوانید فرکانس کلاک دریافتی از اسیلاتور روی برد را به مقدار دلخواه تغییر دهید. در بعضی از FPGAهای شرکت Xilinx نام این بلوک DCM و در بعضی MMCM است که در آزمایشهای بعدی با این واحد بیشتر آشنا میشوید. این بلوکها به جز اینکه فرکانس کلاک را تغییر میدهند، جیتر کلاک را تا حد زیادی بهبود میبخشند و حتی قابلیت تصحیح Duty فرکانس کلاک را تغییر میدهند، جیتر کلاک را تا حد زیادی بهبود میبخشند و حتی قابلیت تصحیح cycle سیگنال کلاک را هم دارند .وقتی از این بلوک استفاده میکنید، بافر BUFG هم به طور خودکار در مسیر سیگنال کلاک ورودی قرار داده میشود و دیگر نیازی به تعریف دستی ندارد. با توجه به قابلیتهای فوق، بهتر است حتی اگر فرکانس کلاک مورد نیاز شما همان فرکانس اسیلاتور روی برد است، از بلوک MCd استفاده کنید تا از مزایای آن در طرح خود استفاده کنید.

تکنیک کدنویسی ۴: همه سیستم را به صورت کاملا سنکرون طراحی کنید. در عمل، عمده مدارات سیستم شما به صورت ترتیبی پیادهسازی میشود. مدار ترتیبی مداری است که خروجی آن در هر لحظه به ورودیهای آن لحظه و ورودیهای لحظات قبل بستگی دارد. مدارات ترتیبی خود به دو نوع سنکرون و آسنکرون تقسیم بندی میشوند. مدارات سنکرون، مداراتی هستند که تغییر در خروجی یا حالات آن همزمان با یک سیگنال خارجی که کلاک نام دارد انجام میشود. در مدار ترتیبی آسنکرون، این تغییرات بدون توجه به وضعیت کلاک انجام میشود. برای پیادهسازی راحت تر مدارتان و دستیابی به سرعت کلاک بیشتر و جلوگیری از بروز مشکلات در هنگام سنتز کد، حتما تمام مدار را به صورت کاملا سنکرون پیادهسازی کنید. اما چگونه مداری را به کمک زبان VHDL به صورت سنکرون طراحی کنیم؟ برای این کار باید تمام توصیفات مداری را در ساختار Process و شرط لبه بالارونده کلاک نوشت.

تکنیک کدنویسی۵: از ساختارهای II و Case تودر تو و شرطهای پیچیده استفاده نکنید. معمولا در هنگام پیادهسازی یک مدار به دنبال رسیدن به حداکثر فرکانس قابل اعمال به سیستم هستیم. در FPGAها بر خلاف پردازندهها، فرکانس کلاک از پیش تعریف شدهای برای سیستم وجود ندارد. حداکثر فرکانس کلاک قابل اعمال به مدار پیادهسازی شده در FPGA به جز اینکه از چه FPGAای استفاده میکنید، به نحوه کدنویسی و طراحی شما بستگی دارد. یکی از مواردی که فرکانس کلاک قابل اعمال به مدار را به شدت کاهش می دهد، استفاده از ساختارهای شرطی II و Case به صورت تودرتو است. همچنین قرار دادن شرطهای پیچیده در این ساختارها نیز منجر به کاهش فرکانس کلاک قابل اعمال به مدار می شود.

بهتر است تا حد امکان، بیش از سه شرط تودرتو استفاده نشود. همچنین از شرطهای پیچیده مثل ترکیب انواع شرطهای مقایسهای با رجیسترهای بزرگ کمتر استفاده شود. اگر در هنگام پیادهسازی کمی به طرح بیشتر فکر کنید، همیشه راههای جایگزینی برای عدم استفاده از شرطهای تودرتو پیدا خواهید کرد. آخرین نکته در استفاده از ساختارهای شرطهای شرطی II از اولویت بیشتری برخوردار استفاده از ساختار ase به نوبه خود باعث افزایش حداکثر فرکانس است؛ چون نهایتا منجر به پیادهسازی مدار ساده تری می شود که این به نوبه خود باعث افزایش حداکثر فرکانس کلاک قابل اعمال به مدار خواهد شد.

تکنیک کدنویسی ۶: تولید لچهای ناخواسته لچهای تصادفی و ناخواسته از توصیفهای ناکامل سیگنالها یا متغیرها در بلوک if نتیجه میشوند. در مثال زیر یک لچ ناخواسته توسط address_strobe تولید میشود.

> Process(address,address_strobe) Begin If address_strobe='1' then Decode_signal<=address; End if; End process;

> > نتايج سنتز

Number of Slices: 0 out of 35840%IOB Flip Flops: 6Number of GCLKs:1 out of8

Process(address,address_strobe) Begin If address_strobe='1' then Decode_signal<=address; else Decode_signal<=false; Endif; end process;

نتايج سنتز

Number of Slices: 3 out of 35840%Number of 4 input LUTs: 6 out of 71680%

تکنیک کدنویسی۷: از حافظههای داخلی بلوکی و توزیع شده درون FPGA به نحو مناسب استفاده کنید.

درون FPGAها دو نوع حافظه RAM وجود دارد که بنا به نیازتان میتوانید از آنها استفاده کنید: حافظههای بلوکی یا Block RAM و حافظه های توزیع شده یا Distributed RAM حافظههای بلوکی، منابع مستقلی هستند در اندازه معمولا ۱۸ کیلوبیت و حافظه توزیع شده در حقیقت همان TULهای موجود در FPGA هستند که با سیم بندی مناسبی تبدیل به یک حافظه توزیع شده در حقیقت همان TULهای موجود در FPGA هستند داشته باشید یعنی مناسبی تبدیل به یک حافظه توزیع شده در حقیقت همان TULهای موجود در FPGA هستند در اندازه معمولا ۱۸ کیلوبیت و حافظه توزیع شده در حقیقت همان TULهای موجود در FPGA هستند که با سیم بندی مناسبی تبدیل به یک حافظه نیاز دارید. اگر برای پیاده سازی چنین حافظهای از یک حافظه بلوکی استفاده کنید، ۲۳۰ بیت از این حافظه نیاز دارید. اگر برای پیاده سازی چنین حافظهای از یک حافظه بلوکی استفاده کنید، ۳۲۰ بیت از این حافظه استفاده میشود و باقی آن بدون استفاده میماند. بنابراین، در این حالت بهتر است از حافظه توزیع شده استفاده میشود و باقی آن بدون استفاده میماند. بنابراین، در این حالت بهتر است از حافظه توزیع شده استفاده کنید. برعکس حالت فوق، اگر برای پیادهسازی یک حافظه بلوکی از حافظه توزیع شده استفاده کنید. برای یاده سازی یک حافظه بلوکی از حافظه توزیع شده استفاده کنید. برعکس حالت فوق، اگر برای پیادهسازی یک حافظه در لیل بیتی از حافظه توزیع شده و سیم بندی از حافظه باید به کمک TULها ساخته شود و هر TULها برای ساخت این حافظه اشغال خواهند شد و سیم بندی از حافظه چند ده بیت گنجایش دارد، با توجه به اینکه این نوع حافظه بای ساخت این حافظه اشغال خواهند شد و سیم بندی از حافظه بود و سیم آزادی در کنار آنها بسیار مفصلی بین آنها برقرار خواهد شد. این سیم بندی زیاد باعث خواهد شد حتی بعضی از TULهای استفاده نیار مفصلی بین آنها برقار خواهد شد. این سیم بندی زیاد باعث خواهد شد حتی بعضی از TULهای وجود ندارد. با توجه به دو مثالی که ذکر شد، بهتر است برای حافظههای خیلی کوچک، خصوصا اگر به تعداد وربدی در در حر به آنها نیاز دارید، از حافظههای توزیع شده TULها استفاده کنید و اگر به حافظههای بزرگ نیاز وربدی حام از حافظههای باز کی سیر و سیم آزادی دو گری به معداد در در حر به آنها نیاز دارید، از حافظههای توزیع شده TULها استفاده کنید و اگر به حافظههای بزرگ نیاز زیادی در حر حر به آنها نیاز دارید، از
تكنيك كدنويسي ٨: تست High Impedance

مثال زیر از نوع داده Sig استفاده می کند تا شرایط را برای وقتی که سیگنال Sig شناور است مشخص مثال زیر از نوع داده If-Then یک سلول منطقی را مشخص می کند که Fan in راه انداز خودش را چک کرده است. نتیجه آن True است اگر راهانداز H یا L باشد. چنین المانی در سنتز وجود ندارد بنابراین مقایسه همواره False بوده و دستورات درون If هیچگاه اجرا نمی شوند و تست z در ورودی- ها بی معنی بوده و قابل سنتز نمی باشد.

Process(sig) Begin If sig='Z' then data<="000001"; End if; end process;

نتايج سنتز

Number of Slices: 0 out of 3584 0% Number of Ios : 7 و همانظور که مشاهده میکنید نرم افزار هشداری مبنی بر اینکه خروجی شما مقدار نمیگیرد به ما نشان میدهد.

WARNING:Xst:1306 - Output <data> is never assigned.

خروجیهای مورد انتظار (برای تهیه گزارش کار)

- مفاهیم طراحی، توصیف و پیاده سازی طرح با VHDL
 - گزارش شبیهسازی و سنتز طرح
 - This with the set of the set o
- مشکلاتی که در حین برنامه نویسی، شبیه سازی و پروگرم کردن با آنها مواجه شدید.

تمرين

با توجه به مطالب گفته شده در مورد بهینهسازی کدهای VHDL برنامه الگوریتم ضرب BOOTH را بازنویسی نموده و ضمن شبیهسازی و صحت عملکرد آن، دو برنامه را از نظر سنتز کاری با هم مقایسه نمایید.

آزمایش ۵

طراحی و پیادهسازی ساعت RTC

هدف

- طراحی یک ساعت دقیق و ایجاد تایمر با استفاده از کلاک داخلی برد DE2
 - شبیهسازی طرح ایجاد شده با استفاده از نرم افزار Quarus
 - استفاده از Segmentهای تعبیه شده بر روی برد برای نمایش ساعت

شرح آزمایش

عبارت RTC از سر کلمههای Real Time Clock گرفته شده است که به سیستمهایی که برای تولید زمان دقیق طراحی می شوند، اطلاق می شود. بیشترین کاربرد RTC در ساعتها، تایمرها و به طور کلی دستگاههایی که به صورت مرتبط با زمان سروکار دارند می باشد.

زمانهایی که توسط RTC تولید می شود به صورت مقادیر رند و دقیق می باشد. RTC عموما به دو روش ایجاد می شود: توسط آی سی های ساعت مجهز به RTC و توسط میکرو کنترلر با کمک تایمر.

نمونهای از آی سیهای ساعت DS1307 و PCF8583 میباشند که علاوه بر تولید مقدارهای صدم ثانیه، دهم ثانیه و ... مجهز به رم داخلی، آلارم روزانه و هفتگی و همچنین قابلیت شمارش زمان توسط خود آیسی میباشند. این آیسیها با پروتکل ارتباط سریال I2C میتوانند به کنترلر متصل شوند.

در روشی که توسط میکروکنترلر پیاده می شود از تایمرهای مجهز به قابلیت استفاده از اسیلاتور توان پایین استفاده می شود.

در این آزمایش بعد از طراحی ساعت RTC با استفاده از کدنویسی VHDL آن را بر روی Segmentهای تعبیه شده بر روی برد نمایش میدهیم.

۱- مرحله اول در این مرحله از آزمایش با سون سگمنت آشنا می شویم. سون سگمنتها قطعاتی هستند که در آنها از ۸ عدد LED استفاده شده است. V LED عدد مربوط به نمایش می باشد و یکی دیگر از آنها مربوط به ممیز است. سون سگمنتها چون از LED تشکیل شده و LED ها ساختاری چون آند و کاتد دارند لذا آنها را به دو دسته آند مشترک و کاتد مشترک دسته بندی می کنند.

در این دسته از سون سگمنتها، کاتد تمامی LEDهای موجود در سون سگمنت به یکدیگر متصل شده است. به این معنا که برای روشن شدن هرکدام از LED های a تا g کافی است که پایه مشترک (پایه کاتد مشترک) رو به صفر (زمین) وصل کنیم و برای روشن شدن هر سگمنت به پایه ی متناظر اون سگمنت یک (مثبت) بدیم.

الف-كاتد مشترك

ب-آند مشترک : در این دسته از سون سگمنتها، آند تمامی LEDهای موجود در سون سگمنت به یکدیگر متصل شده است. به این معنا که برای روشن شدن هرکدام از LEDهای a تا g کافی است که پایه مشترک (پایه آند مشترک) رو به مثبت (یک منطقی) وصل کنیم و برای روشن شدن هر سگمنت به پایه ی متناظر اون سگمنت صفر (زمین) بدیم.



۲- مرحله دوم

ابتدا ماژول یک 7Segment که بر روی برد DE2 قرار دارد را با استفاده از ارتباط بین پایهها و دیودهای A تا G طراحی کنید.

		Segr	7 Segment Display Output				
а	b	С	d	е	f	g	
1	1	1	1	1	1	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	0	1	9

جدول۲- جدول صحت سون سگمنت

Using the 7segment Displays

The DE2-115 Board has eight 7-segment displays. These displays are arranged into two pairs and a group of four, behaving the intent of displaying numbers of various sizes. As indicated in the schematic in **Figure 4-10**, the seven segments (common anode) are connected to pins on Cyclone IV E FPGA. Applying a low logic level to a segment will light it up and applying a high logic level turns it off.

Each segment in a display is identified by an index from 0 to 6, with the positions given in **Figure 4-10**. **Table 4-4** shows the assignments of FPGA pins to the 7-segment displays.



Figure 4-10 Connections between the 7-segment display HEX0 and Cyclone IV E FPGA شکل۲۹- اتصال پایههای سون سگمنت برد DE2

۳ – مرحله سوم در این قسمت از آزمایش با استفاده از ساختار ترتیبی کدهای VHDL و فرکانس 50MHZ تعبیه شده بر روی برد یک تایمر ایجاد کرده و با استفاده از آن تایمر ساعت مورد نظر را طراحی کنید.

Clock Circuitry

The DE2-115 board includes one oscillator that produces 50 MHz clock signal. A clock buffer is used to distribute 50 MHz clock signal with low jitter to FPGA. The distributing clock signals are connected to the FPGA that are used for clocking the user logic. The board also includes two SMA connectors which can be used to connect an external clock source to the board or to drive a clock signal out through the SMA connector. In addition, all these clock inputs are connected to the phase locked loops (PLL) clock input pins of the FPGA to allow users to use these clocks as a source clock for the PLL circuit.

The clock distribution on the DE2-115 board is shown in Figure 4-11. The associated pin assignments for clock inputs to FPGA I/O pins are listed in Table 4-5.



۴- مرحله چهارم
حال با استفاده از طراحی 7Segment در مرحله اول، اعداد ثانیه، دقیقه و ساعت را به 7Segmentهای برد
متصل کرده و با پروگرم کردن برنامه آنها را بر روی برد نمایش دهید.

خروجیهای مورد انتظار (برای تهیه گزارش کار)

- مفاهیم طراحی، توصیف و پیاده سازی طرح با VHDL
 - گزارش شبیهسازی و سنتز طرح
- مشکلاتی که در حین برنامه نویسی، شبیه سازی و پروگرم کردن با آنها مواجه شدید.

تمرين

با استفاده از طراحی ساعت RTC، تاریخ را نیز برای آن در نظر گرفته و آنرا طراحی کنید و بر روی Segmentهای برد نمایش دهید.

آزمایش ۶

پیادهسازی مدارهای دیجیتال با استفاده از IPCoreهای شرکت Xilinx

هدف

- آشنایی با مفهوم IPCoreها
- ISE نحوه ایجاد Core DCM با ویژگیها و مشخصات خاص در نرم افزار
 - آشنایی با Coreهای نرمافزاری و سخت افزاری.

شرح آزمایش

در این آزمایش با هستههای سختافزاری و نرمافزاری و نحوه تولید آنها آشنا می شوید. شرکت Xilinx تعدادی هسته آماده با عملکردهای پرکاربرد را ایجاد کرده و امکانی را در نرم افزار ISE به وجود آورده که به کاربر اجازه استفاده از این هستهها را می دهد. در واقع کاربر قادر است با استفاده از بخش تولیدکننده هسته، هسته مورد نظر را انتخاب و پارامترهای آن را مطابق میل خود تنظیم کند. سپس هسته مورد نظر به صورت خودکار تولید شده و می توان از آن در طراحی اصلی استفاده نمود.

یک دسته از FPGAها از قبل و به صورت انحصاری توسط شرکتهای سازندهی FPGA با توجه به ساختار داخلی آن، طراحی و عرضه می شود مثلا برای استفاده از حافظهی داخلی یا میکروکنترلر تعبیه شده در FPGA می توان از COREهای مخصوص آن استفاده کرد. در اینجا کد VHDL واقعی وجود ندارد و در واقع Core به عنوان یک wrapper برای آن بلوک سخت افزاری می باشد.

دستهی دوم، برنامههای عمومیای هستند که به زبان توصیف سخت افزار نوشته و سنتز شدهاند و فایل سنتز شدهی آن عرضه میشود. مثلا برای ارتباط وسایل جانبی با FPGA میتوان از این دسته COREها استفاده نمود. در اینجا کد VHDL واقعا وجود دارد ولی در دسترس نمیباشد. (ممکن است شرکتهای سازندهی FPGA این دسته از COREها را هم تولید نمایید.) iter این دسته از Component را هم تولید نمایید.) iter iter iter iter iter iter iter میتوان در بدنه پروژههای بزرگتر به عنوان Component استفاده کرد. و یا به schematic یک Schematic جدید در طراحی شماتیک بکار برد به شرطی که فایل سنتز شده و پورت های I/O مربوط به این Component را در اختیار داشته باشیم. iter iter iter iter iter اینکه از چه Coreهای میتوانیم استفاده کنیم بستگی به FPGA ای دارد که انتخاب میکنیم. برای FPGA موجود در آزمایشگاه (XC3S400) Scoreهای قابل استفاده را میتوانید در نرمافزار بینید.

۱- مرحله اول

در این مرحله ما قصد داریم با یکی از هستههای XC3S400 که DCM نام دارد آشنا شویم. در FPGAها بلوکهای از پیش آمادهای وجود دارند که به کمک آنها میتوانید فرکانس کلاک دریافتی از اسیلاتور روی برد را به مقدار دلخواه تغییر دهید. در بعضی از FPGAهای شرکت Xilinx نام این بلوک DCM و دربعضیMMCM است. ۴ عدد DCM درون تراشه XC3S400 وجود دارد که ما میتوانیم از آنها به صورت سختافزاری استفاده کنیم. این بلوکها به جز اینکه فرکانس کلاک را تغییر می دهند جیتر کلاک را تا حد زیادی بهبود می بخشند و حتی قابلیت تصحیح duty cycle کلاک را هم دارند. وقتی از این بلوک استفاده می کنید، بافر iBUFG هم به طور خودکار در مسیر سیگنال کلاک ورودی قرار داده می شود و دیگر نیازی به تعریف دستی ندارد.

با توجه به قابلیتهای فوق، بهتر است حتی اگر فرکانس کلاک مورد نیاز شما همان فرکانس اسیلاتور روی برد است، از بلوک DCM استفاده کنید .

شکل زیر بلوک دیاگرام DCM را نشان میدهد.



شکل ۳۱– بلوک دیاگرام DCM

۲- مرحله دوم

اولین قدم در روند به کارگیری هسته ایجاد پروژه است. در واقع در ابتدا شما نیاز دارید یک پروژه جدید بسازید. سپس فایل بلوک اصلی خود را از نوع HDL ایجاد کنید تا بعد از ایجاد هستهی DCM از آن در فایل اصلی نمونه بگیرید. توجه داشته باشید حتی اگر هسته تنها فایل پروژه باشد نمیتواند بلوک اصلی باشد. بنابرین حتما نیاز به یک فایل اصلی برای نمونه گیری از هسته دارید.

برای اضافه کردن هسته به پروژه در پنجرهی طراحی روی Device راست کلیک کرده و New Source را انتخاب کنید.

File	Edit Vi	ew Project	Source Proc	ess Tools	Window	0
Desig	n	-1	the star end p	1	++□ &	×
-#	View:	Implementation	n 🔿 🎆 Simulat	ion		
E	Hierarchy					^
ò	🖻 🎆 xc	15400 5	ource			
	Th	e vi	ource opy of Source		files to	
21	Pr	ojec Manua	al Compile Orde	er		
2	Us	e: 🕨 Impler	me <mark>nt Top Mod</mark> u	ile		
		File/Pa	th Display	•		
		Expand Collap	d All se All		project. file to ject.	~
>	No Pro	Find		Ctrl+F		•
	No single	des 🖸 Design	Properties			
	÷- 🎾	Design Utilities				

در پنجره New Source Wizard ، گزینه (Core Generator & Architecture Wizard) IP را انتخاب کرده و

، کنید.	کلیک	Next	ر روى	مناسب ہ	نام	انتخاب	پس
---------	------	------	-------	---------	-----	--------	----

New Source Wizard Select Source Type Select source type, file name and its location.	>
 IP (CORE Generator & Architecture Wizard) Schematic User Document Verilog Module Verilog Test Fixture VHDL Module VHDL Library VHDL Dackage VHDL Test Bench Embedded Processor 	Elle name: DCM50T075 Logation: E:\Hamid\VHDL_Prj\Desktop\DCM50T075\ipcore_dir
More Info	Next > Cancel

شکل۳۳- پنجره New Source

پس از این مرحله لیستی از تمام هستههای موجود در ISE نشان داده می شود. نمایش هستهها به دو صورت است. در برگهی نخست این هستهها طبقه بندی شدهاند و شما یا از قبل باید بدانید هستهی مورد نظرتان در کدام دسته بوده و یا از نوع هسته تشخیص دهید که امکان حضور آن در کدام یک از دستههای مربوط می باشد. اما در برگه دوم هستهها با نامشان لیست شدهاند. در قسمت پایین تر امکان جستجوی IPها توسط نامشان وجود دارد.

View by Function	View by <u>N</u> ame		
Name	*	Version	^
🖨 📂 Sp	artan-3		
***	Cascading in Series with Two DCMs	13.1	heat
****	Clock Forwarding / Board Deskew (DCM)	13.1	
	Clock Switching with Two DCMs	13.1	
	Single DCM	13.1	
E Sp	artan-3E, Spartan-3A Board Deskew with an Internal Deskew (DCM-SP)	13.1	~
<		>	
Search IP Catalog: do	m	Cjear	
			12

شكل٣٢- ينجره New Source Wizard 1

همانطور که در شکل زیر مشخص است در برگه دوم تعدادی از هستهها خاکستری نشان داده شدهاند. این هستهها در این نسخه از ISE یعنی نسخه WebPack غیرفعال بوده و در صورت نیاز به آنها باید از نسخههای دارای مجوز و قابل خریداری ISE استفاده کرد.

Select IP	
Create Coregen or Architecture Wizard IP Core.	
View by Function View by Name	
Name	Version AXI4 AXI4-Stre
 32-bit Initiator/ larget for PCI 32-bit Initiator/Target for PCI (Spartan-6 only) 	4.14
32-bit Initiator/Target for PCI (Virtex-5/7-Series)	4.18
3GPP LTE Channel Estimator	1.1 AXI4-Stree
3GPP LTE MIMO Decoder	1.0 2.0 AYM_Strever
Search IP Catalog:	Clear
All IP versions	Only IP compatible with chosen part
Please select IP	
More Info	Next Cancel
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید.	Next Cancel شکل۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard	Next Cancel شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد ×
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard Summary Project Navigator will create a new skeleton source with the follo	Next Cancel شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد ×
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard Project Navigator will create a new skeleton source with the follo	Next Cancel شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد × wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard Summary Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50T075\ipcore_d Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50T075\ipcore_d	<u>Next</u> <u>Cancel</u> شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد × wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50T075\ipcore_d Source Type: IP (CORE Generator & Architecture Wizard)	Next Cancel شکل ۳۵- پنجرہ این مرحلہ پنجرہ خلاصہی طراحی نشان داد × wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard Summary Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50T075\ipcore_d Source Type: IP (CORE Generator & Architecture Wizard) Source Name: DCM50T075.xaw Core Type: Single DCM; Version: 13.1	Next Cancel شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد × wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard Summary Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50T075\ipcore_d Source Type: IP (CORE Generator & Architecture Wizard) Source Name: DCM50T075.xaw Core Type: Single DCM; Version: 13.1	Next Cancel شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard Summary Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50T075\ipcore_d Source Type: IP (CORE Generator & Architecture Wizard) Source Name: DCM50T075.xaw Core Type: Single DCM; Version: 13.1	مکل ۲۵- پنجره شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد × wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard New Source Wizard Summary Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50TO75\ipcore_d Source Type: IP (CORE Generator & Architecture Wizard) Source Name: DCM50TO75.xaw Core Type: Single DCM; Version: 13.1	Next Cancel شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد « wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard New Source Wizard Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50T075\ipcore_d Source Type: IP (CORE Generator & Architecture Wizard) Source Name: DCM50T075.xaw Core Type: Single DCM; Version: 13.1	Next Cancel شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard New Source Wizard Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50T075\ipcore_d Source Type: IP (CORE Generator & Architecture Wizard) Source Name: DCM50T075.xaw Core Type: Single DCM; Version: 13.1	مکل ۲۵- پنجره شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد × wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard New Source Wizard Summary Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50TO75\ipcore_d Source Type: IP (CORE Generator & Architecture Wizard) Source Name: DCM50TO75.xaw Core Type: Single DCM; Version: 13.1	<u>Next</u> Cancel شکل۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard Summary Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50T075\ipcore_d Source Type: IP (CORE Generator & Architecture Wizard) Source Name: DCM50T075.xaw Core Type: Single DCM; Version: 13.1	مکل ۲۵- پنجره شکل ۲۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد × wing specifications.
More Info New Source Wizard 2 ه می شود. آن را مطالعه و گزینه Finish را بزنید. New Source Wizard • Summary Project Navigator will create a new skeleton source with the follo Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\DCM50TO75\ipcore_d Source Type: IP (CORE Generator & Architecture Wizard) Source Name: DCM50T075.xaw Core Type: Single DCM; Version: 13.1	مکل ۲۵- پنجره شکل ۳۵- پنجره این مرحله پنجره خلاصهی طراحی نشان داد wing specifications.

نوع هسته و پیچیدگی آن ممکن است تا چند دقیقه طول بکشد.

۲ – مرحله دوم
بعد از آن که هسته تولید شد حال نوبت به تنظیم پارامترهای هسته میرسد.
در این قسمت می توان سورس ایجاد شده به زبان مورد نظر و ابزار سنتز و همچنین Device مورد نظر را انتخاب
کرد. مانند شکل زیر عمل کرده و بر روی OK کلیک کنید.

💸 Xilinx Architecture W	izard - Setup	×
XAW File: E:\Hamid\VHDL_Prj\Desk Output File Type	top\DCM50T075\ipcore_dir\.\DCM50T075.xaw	
VHDL	◯ Verilog	
Synthesis Tool		
Part	<u>×</u>	
xc3s400-5pq208		Select
	OK Cancel	

شکل۳۷- پنجره Summary

در پنجره بعدی نمایی از بلوک DCM را مشاهده خواهید کرد که می توان تنظیمات برای فرکانس مورد نظر را اعمال کرد.

💸 Xilinx Clocking Wizard - General Setup	×
CLKIN CLKFB	CLK0 CLK90 CLK90 CLK180 CLK270 CLK270 CLK2X CLK2X CLK7X CLK7X CLKFX CLKFX CLKFX CLKFX CLKFX CLKFX CLKFX CLKFX CLKFX CLKFX CLK50 CLK50 CLK2X CLX2X C
Input Clock Frequency	hase Shift Гуре: <mark>NONE ✓</mark> ∕alue: 0 ≎
CLKIN Source	Feedback Source
External O Internal O Single Differential	External Internal None Single Differential
Use Duty Cycle Correction	

شکل ۳۸- پنجره General Setup واحد DCM

پینهای خروجی CLK180، CLK90 و CLK270 برای تولید کلاک خروجی به ترتیب با اختلاف فاز 90، 180 و 270 درجه با کلاک ورودی استفاده می شود. از طرفی پین CLKFX برای تولید فرکانس دلخواه با PLL به کار می رود که ما در این پروژه از این پین استفاده می کنیم.

Input Clock Frequency: کلاک اسیلاتور را در این قسمت وارد می کنیم. Phase Shift: اگر بخواهیم کلاک خروجی مورد نظر با اختلاف فاز نسبت به کلاک ورودی تولید شود در این قسمت تنظیمات مربوط را انجام میدهیم. CIKIN Source: چون کلاک از طریق اسیلاتور تولید می شود Input Clock را انتخاب می کنیم. Feedback Value: اگر 11 انتخاب شود ورودی فیدبک را از CLK0 و اگر 2x انتخاب شود ورودی فیدبک را از CLK2X می گیرد. در صفحه بعد در قسمت Clock Buffer Setting در صورتی که بخواهیم از Global Bufferها در تمام خروجیها استفاده شود از گزینه اول استفاده می کنیم. در صورتی که بخواهیم فقط خروجیهای مورد نظر را بافر کنیم از گزینه دوم استفاده می کنیم.

ck Buffer Settings		
Use Global Buffers for all selecte	d clock outputs	
) Customize buffers		
Input 10	Input 11	View/Edit Buffer
CLKO		Global Buffer
CLK2X		Global Buffer

شكل ۳۹- پنجره Clock Buffers واحد

در ادامه بر روی Next کلیک می کنیم. در مرحله بعد در قسمت Use Output Frequency مقدار فرکانس خروجی مورد نیاز کلاک خود را وارد نمایید. همچنین با استفاده از گزینه Calculate می توان ضرایب ضرب و تقسیم را نیز محاسبه کرد.

0.0 1400	le	Fin (MHz)	Fo	ut (MHz)	
Low		1.000 - 280.000 18		8.000 - 210.000	
High		1.000 - 280.000	210.0	00 - 280.000	
Use output frequency Use output frequency Use Multiply (M) ar M 4 0	ncy • 50 MHz • MHz • MHz nd Divide (D) values D 1	Ons ≎			
enerated Output	D	Output Freq (MHz)	Period Jitter (unit interval)	Period Jitter (pk-to-pk ns	
0	0	0	0	0	

شكل ۴۰- پنجره Clock Frequency واحد DCM

در ادامه بر روی Next کلیک نمایید.

ኛ Xilinx Clocking Wizard - Summary			>
Feature Summary:			
A single DCM_SP configured			
iles To Be Generated:			
File Directory: E:\Hamid\VHDL_9tj\Desktop\as\pcore_dir ArchWiz file: dc.xaw HDL file: dc.vhd UCF template file: dc_arwz.ucf			
Block Attributes:			
Attributes for DCM_SP_bikname = DCM_SP_INST CLK_FEEDBACK = 1X CLKDV_DIVIDE = 2 CLKFX_MULTIPLY = 2 CLKFX_MULTIPLY = 3 CLKIN_PERIOD = 20000 CLKIN_PERIOD = 20000 CLKIN_PERIOD = 0000 CLKIN_PERIOD = SYSTEM_SYNCHRONOUS DESKEW_ADJUST = SYSTEM_SYNCHRONOUS DESKEW_ADJUST = SYSTEM_SYNCHRONOUS DESKEW_ADJUST = SYSTEM_SYNCHRONOUS DESKEW_JDIUST = SYSTEM_SYNCHRONOUS DESKEW_JDIUST = SYSTEM_SYNCHRONOUS DESKEW_JDIUST = SYSTEM_SYNCHRONOUS DESKEW_JDIUST = SYSTEM_SYNCHRONOUS DESKEW_JDIUST = SYSTEM_SYNCHRONOUS DESKEW_JDIUST = SYSTEM_SYNCHRONOUS DIL_FREQUENCY_MODE = LOW DUTY_CYCLE_CORRECTION = TRUE FACTORY_JF = 16hC080 FRACTORY_JF = ALSE			
Show all modifiable attributes			
Show only the modifiable attributes whose values differ from the	e default		
	< Back	Finish	Cancel

شکل ۴۱- پنجره Summary واحد DCM

با کلیک بر روی IPCore ،Finish مورد نظر ایجاد شده و میتوانید از آن استفاده کنید.

۳- مرحله سوم

حال در نرم افزار با انتخاب DCM ساخته شده و در قسمت Process با انتخاب DCM ساخته شده و در قسمت Roceas با انتخاب Create Schematic Symbol ساخته شماتیک مورد نظر ساخته می شود و می توانیم حتی در محیط شماتیک نیز از آن استفاده کنیم.



شكلSchematic -۴۳ واحد DCM

۴- مرحله چهارم

در این مرحله قصد داریم Core ایجاد شده را شبیهسازی کنیم.

برای ایجاد فایل شبیه سازی دوباره بر روی Device راست کلیک کرده و New Source را انتخاب میکنیم.



در پنجره Wizard گزینه VHDL Test Bench را انتخاب کرده و پس از نوشتن نام فایل مناسب بر روی Next

	للیک میکنیم.
New Source Wizard Select Source Type Select source type, file name and its location.	×
 BMM File ChipScope Definition and Connection File Implementation Constraints File IP (CORE Generator & Architecture Wizard) MEM File Schematic User Document Verilog Module Verilog Test Fixture VHDL Module VHDL Vackage VHDL Test Bench Embedded Processor 	File name: TestDCM Location: E:\Hamid\VHDL_Prj\Desktop\TestDCM
More Info	✓ Add to project Next > Cancel

شکل۴۵- پنجره New Source

در پنجره بعدی DCM را انتخاب کرده و بر روی Next کلیک کنید.

> New Source Wizard	×
←Associate Source Select a source with which to associate the new source.	
DCM50T075	
More Info < Back Next > Cance شکل ۴۵- انتخاب واحد DCM برای ایجاد Test Bench شکل ۲۵- انتخاب واحد Test کارک کرده تا فادل تریزی دیچ یا تماید.	
> New Source Wizard	تر چ <u>دیت</u> بر روی ×
Summary Project Navigator will create a new skeleton source with the following specifications.	
Add to Project: Yes Source Directory: E:\Hamid\VHDL_Prj\Desktop\TestDCM Source Type: VHDL Test Bench Source Name: TestDCM.vhd Association: DCM50TO75	
More Info < Back Finish Cano	cel

شکل۴۶- پنجره Summary ایجاد Test Bench

در صفحه ادیتور کد مانند شکل زیر عمل کرده و مقدار CLKIN_IN_Period را برابر 20ns قرار داده تا فرکانس کلاک ورودی ۵۰ مگاهرتز شود. همچنین قسمتهای مشخص شده را پاک نمایید.



شکل ۴۷- کد Test Bench

پس از انجام دادن مراحل فوق با دابل کلیک کردن بر روی Simulate Behavioral Model تست بنچ را شبیه سازی می کنیم.



					230.000 hs					
Name	Value	0 ns	100 ns	200 n		300 ns	400 ns	500 ns	600 ns	700 n
🗓 dkin_in	1			П						
ᠾ clkfx_out	1				າມາມມ		ການບາກ		ոսսոս	UL
🕼 clkin_ibufg_out	1			Г			IIIII	luuuu		Л
🕼 clk0_out	1			П						Л
U locked_out	0									
🕼 clkin_in_period	20000 ps					20000 ps				

شکل ۴۸- محیط شبیه سازی ISE

خروجی های مورد انتظار (برای تهیه گزارش کار)

- ارائه مفهوم Coreهای نرم افزاری و سخت افزاری
 - گزارش شبیهسازی و سنتز طرح
- مشکلاتی که در حین برنامهنویسی، شبیهسازی و پروگرم کردن با آنها مواجه شدید.

تمرين

فراخوانی و انجام تنظیمات یکی از Coreهای کاربردی موجود در کتابخانه نرمافزار ISE (برای هر گروه یک Core و Core توسط مدرس تعیین می شود)

۱-شمارنده BCD

۲-حافظه FIFO

۳-حافظه LIFO

Multiplier -۴

۵-حافظه RAM

آزمایش ۷

آنالیز و عیبیابی طرح دیجیتال با استفاده نرم افزار ChipScop

هدف

- آشنایی با روشهای تست طراحیها
- آشنایی با ChipScop و ویژگیهای نرمافزار Xilinx
 - نحوه تست طراحي با استفاده از ChipScop

شرح آزمایش

یکی از چالشهای یک طراح دیجیتال با FPGA مسأله تست مدار دیجیتال پیادهسازی شده است. دلیل بنیادین چنین چالشی مربوط به تفاوت پردازندهها و تراشههای FPGA است. همانطور که میدانید، به هنگام کار با پردازندهها با سختافزاری از پیش آماده شده سروکار داریم که دستورات مشخصی را به ترتیب اجرا میکند. در هر لحظه از زمان فقط یکی از این دستورات اجرا می شود. بنابراین عیبیابی آن نسبتاً آسان است؛ چون می توان برنامه را خط به خط اجرا و دنبال کرد و همزمان نتایج متغیرهای مختلف را بررسی نموده تا ایرادات احتمالی آشکار شوند. اما در مورد FPGAها، از آن جایی که در حال توصیف یک سختافزار دیجیتال هستیم و ماهیت سختافزار به گونهای است که ممکن است دهها و صدها عملیات مختلف را به طور موازی انجام دهد، و همچنین دستوراتی که خط به خط و به ترتیب اجرا شوند وجود ندارد، نمی توان به آسانی عیبیابی یک برنامه کامپیوتری، اتفاقات را دنبال کرد. همین مسأله باعث می شود عیبیابی در طراحی با FPGA نسبتاً دشوار و کند انجام شود. اولین ابزار کمکی برای عیبیابی، نرمافزارهای شبیهسازی است که به کمک آن میتوان شبیهسازی منطقی و زمانی را انجام داد. به کمک این شبیهسازها می توان سیگنالهای دلخواه در طرح را مانیتور کرد. هنگامی که رفتار یک سیگنال یا رجیستر بر مبنای نتایجی که شبیهساز ارائه می کند مطابق انتظار طراح نیست، ممکن است بخشهای مختلفی در کد توصیف سختافزاری، دلیل خطا باشد. بنابراین طراح باید با تمرکز زیاد و بررسی دقیق کد، سعی در یافتن اشتباهات احتمالی کند. بسیاری از خطاها وقتی طرح بسیار بزرگ شده است اتفاق می افتد. همچنین خطاهایی وجود دارند که پس از شبیهسازی طولانی ممکن است خود را نشان دهند. باتوجه به اینکه عمل شبیهسازی خصوصاً برای طرحهای بزرگ یا برای بازههای زمانی طولانی، بسیار زمانبر است، در عمل کارایی شبیهسازها محدود می شوند. برای رفع این مشکل، شرکت Xilinx نرمافزاری را طراحی کرده است که به کمک رابط Jtag موجود در تراشههای FPGA، این امکان را به طراح میدهد که مقادیر رجیسترهای مورد نظر خود را بعد از پیکرهبندی FPGA و به هنگام کار آن مانیتور کند. این نرمافزار که Chipscope نام دارد، به همراه مجموعه نرم افزار ISE Design Suite به طراحان ارائه میشود. یکی از محاسن استفاده از نرمافزار Chipscope سرعت بسیار زیاد عیبیابی است؛ زیرا به جای شبیهسازی، عملاً کد مورد نظر درون FPGA قرار دارد که سرعت کار آن بسیار زیاد است. همچنین در این حالت، مواردی که امکان شبیهسازی آن وجود ندارد نیز در نظر گرفته میشود و نتایج در شرایطی کاملاً واقعی محاسبه میشوند. در این آزمایش قصد داریم به کمک آزمایش قبل و استفاده از واحد DCM سیگنال کلاک Z0MHZ را به یک سیگنال کلاک 40MHZ تبدیل کنیم و نتایج حاصل از آن را با استفاده از نرم افزار Chipscope مشاهده کنیم.

۱- مرحله اول

در این مرحله از آزمایش ابتدا همانند آنچه در آزمایش قبل توضیح داده شد با استفاده از واحد DCM و کلاک 20MHZ تعبیه شده بر روی برد FPGA یک کلاک 40MHZ ایجاد کنید.

۲- مرحله دوم

در این مرحله قصد داریم به نحوه تولید Core مربوط به نرم افزار Chipscope بپردازیم. ابتدا بر روی خانواده FPGA مورد نظر کلیک راست کرده و گزینه New Source را انتخاب می کنیم.



شکل۴۹- گزینه New Source

در ادامه با تعیین نام مناسب و انتخاب گزینه ChipScope با همان آدرس پیش فرض با انتخاب گزینه NEXT

New Source Wizard		×
Select Source Type Select source type, file name and its location.		
BMM File ChipScope Definition and Connection File Implementation Constraints File IP (CORE Generator & Architecture Wizard) MEM File Schematic User Document Verilog Module VHDL Module VHDL Library VHDL Test Bench Embedded Processor	Eile name: chip Logation: C: \Users\Mohsen\Desktop\DCm\DCM	
Mara Tafa	Add to project	Cancel

شکل ۵۰- پنجره New Source

همانطور که مشاهده میکنید core مورد نظر ایجاد شده و خلاصه ای از آن را همانند شکل زیر به ما نشان

New Source Wizar	d		×
Summary Project Navigato	r will create a new skeleton source w	ith the following specifications.	
\dd to Project: Yes Source Directory: C:\ Source Type: ChipSco Source Name: chip.cd	Jsers (Mohsen \Desktop \DCm \DCM pe Definition and Connection File c		
Association: sevense	ment		
More Info		< <u>B</u> ack <u>F</u> inish	Cancel
	ايجاد Core ChipScope	شکل۵۱- پنجره Summary	

۳- مرحله سوم

همانظور که در شکل زیر مشاهده می کنید Core مورد نظر ایجاد شده است. حال در این مرحله از آزمایش قصد داریم به نحوه تنظیم این Core بپردازیم.



شكل ۵۲- نحوه تنظيم Core ChipScope

ابتدا بر روی core چیپ اسکوپ دابل کلیک کرده تا صفحهای همانند شکل زیر ایجاد شود.

ChipScope Pro Core Ins	erter [chip.cdc]	
<u>File E</u> dit <u>H</u> elp		
		8
	ICON	Select Integrated Controller Options
CON	Parameters	
	No ICON parameters exist for the selected device family.	
Core Utilization	-	
LUT Count: 97 FF Count: 28		
BRAM Count: 0		
	< <u>P</u> revious Ne <u>x</u> t >	New ILA Unit New ATC2 Unit
Messages ERROR UNITO: Unconnect ERROR UNITO: Unconnect	ed clock net ed trigger net(s)	
•		

شکل۵۳- پنجره تنظیم۱ نرم افزار ChipScope

در ادامه با انتخاب گزینه ICON بر روی Next کلیک کرده و به مرحله بعد میرویم.

در این قسمت با انتخاب یک پورت برای Number of Input Trigger Ports و عرض دو، برای Trigger Width روی Number of Input Trigger Ports کلیک کرده و به مرحله بعد میرویم.

	ILA Trigger Parameters Capture Parameters Net Connections	Select Integrated Logic Analyzer Options
	Trigger Parameters Capture Parameters Net Connections	
OU. ILA		
	Trigger Input and Match Unit Settings	
	Number of Input Trigger Ports: 1	Number of Match Units Used: 1
pre Utilization	TRIGO: Trigger Width: 2	Match Type: Basic w/edges
	# Match Units: 1	Bit Values: 0, 1, X, R, F, B, N
	Counter Width: Disabled	▼ Functions: =, <>
F Count: 205 BRAM Count: 1	Trigger Condition Settings ✓ Enable Trigger Sequencer	Max Number of Sequencer Levels:
	Storage Qualification Condition Settings	
	✓ Enable Storage Qualification	
	< <u>P</u> revious Ne <u>x</u> t >	Remove Unit
sages		
COR UNITU: Unconnected tri	igger net(s)	

شکل۵۴- پنجره تنظیم۲ نرم افزار ChipScope

در مرحله بعد تعداد Sample مورد نظر از سیگنال را می توانیم انتخاب کنیم که آن را بر روی ۵۱۲ گذاشته و با زدن گزینه Next به مرحله بعد می رویم.

ChipScope Pro Core Inser	ter [chip.cdc]
<u>File Edit H</u> elp	
	8
DEVICE	ILA Select Integrated Logic Analyzer Options
	Tringer Parameters Canture Parameters Net Connections
UO: ILA	Cantra Setting
	Sample On Rising Clock Edge
	Data Depth: 512 🔽 Samples
Core Utilization	Triang Back Unod to Date
	Ingger Ports Used As Data
	Include TRIG0 Port (width=2)
LUT Count: 333	
FF Count: 205	
BRAM Count: 1	
· · · · · · · · · · · · · · · · · · ·	
	< Previous Ne <u>x</u> t> Remove Unit
Маералар	
ERROR UNITO: Unconnected	d trigger net(s)
show SignalBrowserDialog	=
4	
•	

شکل۵۵- پنجره تنظیم۳ نرم افزار ChipScope

در این مرحله باید ورودی و خروجیهای که قصد داریم وضعیت آنها را مشاهده کنیم به پورتهای مورد نظر در Core چیپ اسکوپ متصل نماییم. ابتدا بر روی گزینه Modify Connections کلیک کنید.

ChipScope Pro Core Inse	erter [chip.cdc]	×
<u>File Edit H</u> elp		
		?
DEVICE	ILA Select Integrated Logic Analyzer Opti	ons
U0: ILA	Trigger Parameters Capture Parameters Net Connections	
	Net Connections	
Core Utilization		
LUT Count: 333		
EE County 205		
PP Count. 205		
BRAM Count: 1		
		_
	Modify Connections	
	< Previous Return to Project Navigator Remove	Unit
	2	
ERROR UNITO: Unconnecto	ed trigger net(s)	
		_
show SignalBrowserDialog	g	E
4		

شکل۵۶- پنجره تنظیم۴ نرم افزار ChipScope

هنگامی که بر روی گزینه Modify Connections کلیک میکنیم صفحهای همانند شکل زیر ایجاد می شود که تمام ورودی و خروجی های کد مورد نظر در آن قسمت وجود دارد.

Calect Net					
Structure / Nets					Net Selections
— / [dcmvhdl]					Clock Signals Trigger/Data Signals Channel CH:0
∢ Net Name	Pattern:			Filter	
NetNerse			Dese Ture		
NetName	Source Instance	Source Component	Base Type		
NO	XST_GND	GND	GND	^	
CO/CLK0_BUF	CO/DCM_INST	DCM	DCM		
CU/CERFX_BUF		DUM	DUM		
clkbuf OBUE	CO/CLKIN IBUEG INST	IBUEC	IBUEG		
clkfx_OBUE	CO/CLKEX BUEG INST	BUEG	BUEG		
clkin	demybdl	demyhdi	PORT		
•				<u>×</u>	CP0 Make Connections Move Nets Up Remove Connections Move Nets Down
		C	K Canc	el	

شکل۵۲- پنجره Select Net نرم افزار ChipScope

در سمت راست این پنچره باید ورودی و خروجیهای مورد نظر را به پورتهای تعیین شده انتساب دهیم که در قسمت Clock Signals کلاک اصلی به سیستم را از سمت چپ انتخاب کرده و گزینه Make Connections را میزنیم تا کلاک اصلی سیستم به این پورت اختصاص داده شود.

Net Selections	
Clock Signals	Trigger/Data Signals
Channel	
CH:0 /c0/CLK0	D_BUF
CP0	
	I
Make Connection	Move Nets Up
Remove Connectio	Move Nets Down

شکل۵۸- پنجره انتساب سیگنال Clock نرم افزار ChipScope

در ادامه از قسمت Trigger/data signals برای بقیه ورودی و خروجیهای که قصد دیدن تغییرات آنها را داریم عملیات انتساب را انجام میدهیم.

Net Sele	ctions			
Clock S	ignals	Trigge	er/Data Signals	
CH:0	/clkbuf O	BUF		
CH:1	/clkfx_O	BUF		
ТРО				
Make C	onnections	s	Move Nets Up	
Remove	Connection	ns	Move Nets Down	

شکل۱۵۹- پنجره انتساب سیگنال I/O نرم افزار ChipScope

بعد از انتساب ورودی و خروجیهای مورد نظر به پورتها بر روی OK کلیک کنید تا به صفحه قبل باز گردد. همانطور که در شکل زیر مشاهده می کند Clock Port و Trigger Port که در قسمت قبل قرمز رنگ بود به رنگی مشکلی در آمده که به معنای این است که پورتهای خود را درست انتخاب کردیم. در ادامه بر روی گزینه Return To Project Navigator کلیک کرده تا تنظیمات انجام شده، بر روی Core چیپ اسکوپ اعمال شود.

ChipScope Pro Core Ins	erter [chip.cdc]	- C X
<u>F</u> ile <u>E</u> dit <u>H</u> elp		
		8
	ILA Select Integrated Logic	Analyzer Options
	Trigger Parameters Capture Parameters Net Connections	
UU: ILA	► Net Connections	
Core Utilization	CLOCK PORT CLOCK PORT TRIGGER PORTS	
	Modify Connections	
	Previous Return to Project Navigator	Remove Unit
Messages		
show SignalBrowserDialo show SignalBrowserDialo	g g	
•		•

شكل ۶۰- پنجره تنظيم ۴ نرم افزار ChipScope

۴- مرحله چهارم

در ادامه آزمایش بر روی گزینه Generate Programming File کلیک کرده تا فایل bit. مورد نظر برای پروگرم کرده FPGA ساخته شود.



شکل ۶۱- گزینه Generate Programming File

حال با استفاده از نرم افزار IMPact برنامه خود را که شامل Core چیپ اسکوپ میباشد بر روی FPGA پروگرم نمایید.



شکل ۶۲- گزینه انتخاب نرمافزار IMPact

۵- مرحله پنجم
حال در این مرحله از آزمایش قصد داریم با استفاده از نرمافزار ChipScope و Core مورد نظری که ساختیم
عملکرد و تغییرات سیگنالهای آن را در داخل FPGA مورد بررسی قرار دهیم.

ابتدا بر روی گزینه Analyze Design Using ChipScope کلیک کنید.



شكل ۶۳- گزينه انتخاب Analyze Design Using ChipScope

بعد از انتخاب گزینه Analyze Design Using ChipScope نرم افزار شناسایی کابل را انجام میدهد که بعد از شناسایی، شکلی همانند شکل زیر ایجاد میشود.

Ew Project A Revealed A Reve A Revealed A Re	~
G	nipscope Pro
	-
	ChipScope Pro Analyzer
F0: ChipScope Pro Analyzer Version: 13.2 0.61xd (8 OMMAND: open_cable	ChipScope Pro Analyzer JTAG Chain Device Order Index Name Device Name IR Length Device IDCODE USERCODE OMyDevice0 System ACE CF 8 0a001093 IMvDevice1 XC6SIX45T 6 34028093
F0: ChipScope Pro Analyzer Version: 13.2 0.61xd (B DMMAND: open_cable F0: Started ChipScope host (localhost:50001) F0: Successfully opened connection to server. localh F0: Successfully opened Xilinx Platform USB Cable F0: Cable: Platform Cable USB, Port: USB21, Speed: 3	ChipScope Pro Analyzer ITAG Chain Device Order Index Name Device Name IR Length Device IDCODE USERCODE OMyDevice0 System ACE_CF 8 Da001093 IMyDevice1 XC6SLX45T 6 34028093 OK Cancel Read USERCODEs
FD: ChipScope Pro Analyzer Version: 13.2 0.61xd (B OMMAND: open_cable FD: Started ChipScope host (localhost:50001) FD: Successfully opened connection to server: localh FD: Successfully opened Xilimx Platform USB Cable FD: Cable: Platform Cable USB, Port: USB21, Speed: 3	ChipScope Pro Analyzer TAG Chain Device Order Index Name Device Name IR Length Device IDCODE USERCODE OMyDevice0 System ACE CF 8 0a001093 1MyDevice1 xC6SLX45T 6 34028093 OK Cancel Read USERCODEs

بعد از شناسایی کابل پنجره زیر باز میشود که با انتخاب گزینه Start 💌 میتوانید عملکرد سیگنالها را مشاهده نمایید.



خروجیهای مورد انتظار (برای تهیه گزارش کار)

- ارائه مفاهیم تکمیلی در رابطه با این آزمایش
 - گزارش شبیهسازی و سنتز طرح
- مطالعه و پیشنهاد روشها و نرم افزارهای دیگر برای عیب یابی FPGA
- مشکلاتی که در حین برنامه نویسی، شبیه سازی و پروگرم کردن و نرم افزار چیپ اسکوپ با آنها مواجه شدید.

تمرين

فراخوانی و انجام تنظیمات یکی از Coreهای کاربردی موجود در کتابخانه نرم افزار و تست طرح با استفاده از نرمافزار چیپ اسکوپ (برای هر گروه یک Core توسط مدرس تعیین می شود)

RAM -شمارنده BCD -۲ BCD -۳ FIFO -۳ حافظه Multiplier -۴ LIFO - حافظه ۲۰

آزمایش ۸

طراحی و پیادهسازی پروتکل USART و USART و

هدف

- انواع روشهای انتقال سریال
- آشنایی با پروتکل USART
- بررسی نحوه ارتباط سریال FPGA از طریق پورت RS232 و تبادل اطلاعات بین این دو

شرح آزمایش

انتقال اطلاعات را میتوان به دو روش کلی، انتقال به روش موازی و انتقال به روش سریال تقسیم بندی نمود. در روش موازی، داده ها بصورت موازی و به تعداد بیت موجود، به خطوط (سیم) نیاز دارند تا ارتباط ایجاد شود. به طور مثال برای انتقال دادهی یک بایتی (۸بیت) به ۸ خط نیاز داریم که به این ترتیب دادهها نسبت به روش سریال، سریعتر منتقل میشوند ولی عیبهایی هم دارند که در اثر طولانی بودن مسیر و به تعداد زیاد خطوط، نویز سیمها بیشتر میشود و همچنین سیمهایی که در مجاور هم هستن، ظرفیت خازنی شان هم بیشتر شده که این کار باعث تغییر شکل پالسها میشود.

در روش سریال انتقال داده معمولا توسط یک خط (سیم) انجام می شود که این تک سیمه بودن با این که سرعت انتقال داده رو پایین می آورد ولی باعث کاهش نویز و کاهش هزینه سیمهای اضافی شده که اینکار هم برای فواصل دور این سبک انتقال رو بهینه می کند.

به طور كلى سه روش ارتباط بين فرستنده و گيرنده قابل بررسي ميباشد:

۱-روش یک طرفه (Simple): در این روش، اطلاعات فقط در یک جهت انتقال می یابند.

۲-روش نیمه دو طرفه (Half Duplex): در این روش، اطلاعات می تواند در دو جهت انتقال یابد. اما در هرلحظه فقط یک جهت امکان پذیر است.

۳-روش دو طرفه (Full Duplex): در این روش اطلاعات در یک لحظه در هر دو جهت انتقال مییابد. دو روش کلی برای انتقال سریال وجود دارد که عبارتند از: ۱-انتقال سریال به روش سنکرون یا همزمان: در این روش، انتقال اطلاعات بین دو سیستم، به ازای هر پالس ساعت، یک بیت ارسال میشود. بطوری که فرستنده همزمان با ارسال اطلاعات، پالس ساعت را نیز ارسال میکند و گیرنده به کمک پالس ساعت که از فرستنده ارسال شده، خود را تنظیم و همزمان میکند تا از روی خط داده، اطلاعات را به درستی بردارد. ۲-انتقال سریال به روش آسنکرون یا غیر همزمان:

در این روش، یک کاراکتر همراه با بیت توازن، یک بیت پایانی و یک بیت شروع جمعا به تعداد ۱۰ بیت فرستاده می شود. لازم به ذکر است که تمام بیت های حاوی اطلاعات مفید نیستند!

در حقیقت فقط هشت بیت حاوی این اطلاعات میباشند، بیت توازن نشان میدهد که آیا اطلاعات بطور صحیح دریافت شده یا خیر. بیتهای شروع و پایان، بجای سیگنال ساعت که به همراه اطلاعات ارسال نمی شود، بکار گرفته می شوند. در عوض فرض می شود که اطلاعات با فرکانسی در محدوده ۱۰٪ فرکانس از پیش تعریف شده ارسال می شود.

هنگام دریافت اطلاعات، در مد آسنکرون به دلیل آنکه هیچ پالس ساعتی با اطلاعات فرستاده نمی شود، باید با دقت خاص دریافت نمود. برای این کار، اطلاعات را با یک بیت شروع و به دنبال آن اطلاعات اصلی، بیت توازن و در انتها بیت پایانی ارسال می شوند. گیرنده خود را با جستجو برای یک بیت شروع همزمان می کند.



شکل۶۶- قالب دریافت اطلاعات سریال

برای این کار از رشته اطلاعات نمونهبرداری شده تا یک منطق صفر به دست آید، بعد از آن نیمی از زمان یک بیت سپری شده تا مجددا از آن نمونه برداری شود. اطلاعات نمونه برداری شده در یک رجیستر انتقالی جمع شده تا به صورت یک بایت اطلاعات موازی درآید. سپس با بررسی بیت توازن، بررسی خطاها و بیت پایانی آزمایش می گردد. اگر بیت پایانی موجود باشد، اطلاعات قابل قبول است و در غیر این صورت یک خطای قالب بندی نشانداده خواهد شد. خطای قالب بندی معمولا در صورتی اتفاق می افتد که اطلاعات با نرخ انتقال (Baud Rate اشتباه دریافت گردد.

در این آزمایش قصد داریم تا با استفاده از پروتکل USART ارتباطی بین FPGA و کامپیوتر از طریق پورت RS232 برقرار کنیم.

۱- مرحله اول

پروتكلRS232چيست؟

پروتکل RS232 یکی از پروتکلهای قدیمی اما در عین حال، پرکاربرد در صنایع مختلف است. در این مرحله از آزمایش، نحوه پیادهسازی پروتکل RS232 را شرح میدهیم و در مورد ملزومات پیادهسازی بخشهای مختلف آن توضیح داده می شود.

وقتی با FPGA کار می کنید، حتما نیاز خواهید داشت با دنیای بیرون آن ارتباط برقرار کنید. مثلا دستورات یا پارامترهایی را از یک کامپیوتر یا یک برد دیگر به FPGA ارسال کنید. یا بعضی از نتایج محاسبه شده در FPGA را به کامپیوتر، برای نمایش یا انجام محاسبات مورد نیاز ارسال کنید. برای این کار، نیاز به یک پروتکل ارتباطی دارید. یکی از پروتکلهای ارتباطی که از گذشته مورد استفاده بوده و هنوز هم در خیلی از موارد میتواند انتخاب بهینه باشد، پروتکل S232 است. به کمک این پروتکل میتوان با سرعتهایی در حد چند ده کیلوبیت بر ثانیه و به صورت پکتهایی شامل هشت بیت، دیتا را منتقل نمود. ارتباط سریال S232 برای انتقال دادهها در مسیرهای با طول متوسط و سرعت کم از گذشته مورد استفاده قرار می گرفته است. در حال حاضر هم بسیاری از وسایل جانبی در سیستمهای کنترلی و یا پردازشی از این پروتکل ساده اما کارآمد برای انتقال دادهها در مسیرهای با طول متوسط و سرعت کم از گذشته مورد استفاده قرار می گرفته است. در حال حاضر هم بسیاری از وسایل جانبی در سیستمهای کنترلی و یا پردازشی از این پروتکل ساده اما کارآمد برای انتقال دادهها در مسیرهای با طول متوسط و سرعت کم از گذشته مورد استفاده قرار می گرفته است. در حال حاضر هم بسیاری از وسایل جانبی در سیستمهای کنترلی و یا پردازشی از این پروتکل ساده اما کارآمد برای در یک ارتباط دوطرفه سریال، دو وسیله مطابق شکل زیر به تبادل دادهها می پردازند. در هر سمت، یک فرستنده و یک گیرنده S2322 وجود دارد. بین دو وسیله مورد نظر، حداقل سه سیم برای ارتباط وجود دارد. یکی برای انتقال داده از وسیله شماره ۱ به شماره ۲ و دیگری ارتباط در جهت عکس را فراهم میکند. یک سیم هم به



شکل۶۷- نحوه سیم بندی فرستنده و گیرنده ارتباط سریال

اصولا یکی از این دو وسیله Data Terminal Equipment یا DTE و دیگری Data Circuit Terminating یا DCE میتوان از DCE میتوان از DCE میتواند یک پورت کامپیوتر باشد و به عنوان مثالی برای DCE میتوان از مودم، چاپگر، حافظههای بیرونی و دیگر وسایل جانبی نام برد.

پورتهای سریال RS232 در دو اندازه ۹ پین و ۲۵ پین وجود دارند. پورت در اندازه ۹ پین که بیشتر مورد استفاده قرار می گیرد در شکل زیر نشان داده شده است.



شکل ۶۸- نحوه اتصال RS232 به برد DE2

در این پورت ۹ پین که از دید DCE نشان داده شده است، پایه شماره ۲ برای ارسال دیتا Tx و پایه شماره ۳ برای دریافت دیتا Rx استفاده می شود. پایه شماره ۵ هم به عنوان پایه زمین GND استفاده شده است. اگر پورت ۹ پین را در سمت DTE در نظر بگیریم، جای پین Tx و پین Rx عوض می شوند و به ترتیب پین شماره ۲ و ۳ خواهند بود.

۲- مرحله دوم

موضوعی که آگاهی از آن برای طراحی و پیادهسازی این بلوک مورد نیاز است، فرمت پکت ارسال در پروتکل RS232 است .شکل زیر، یک پکت سریال RS232 را نشان میدهد. هر پکت یا بسته داده در ارتباط RS232 یک بایت را منتقل میکند.


همانطور که در شکل بالا دیده می شود، زمانی که داده ای ارسال نمی شود، باید سطح منطقی یک روی خط ارسال قرار بگیرد. پس از شروع ارسال یک پکت، پکتی با مشخصات فوق باید در خروجی قرار بگیرد. این پکت از ۱۱ بیت تشکیل شده است. توجه کنید که هر پکت، RS232 یک بایت دیتا را منتقل می کند.

بیت اول که مقدار آن صفر است Start Bit نامیده می شود و مشخص کننده شروع پکت است. هشت بیت مربوط به دیتا در ادامه قرار می گیرند. بیت های دیتا به ترتیب از سبک به سنگین ارسال می شوند. در ادامه، بیت پریتی Parity Bit قرار می گیرد و در آخر، بیت Stop Bit که همواره یک است قرار گرفته و پایان یک پکت را مشخص می کند.

بیت پریتی

مقدار بیت پریتی برابر XOR همه بیتهای داده ورودی هستند. از این بیت به هنگام دریافت پکت در گیرنده و برای بررسی صحت داده دریافتی استفاده میشود. برای این کار، با توجه به اینکه مقدار این بیت برابر XOR تمام هشت بیت داده ورودی است، درگیرنده و پس از دریافت بایت موردنظر، یک بار دیگر بیت پریتی بایت دریافتی را حساب نموده و آن را با بیت پریتی دریافتی مقایسه میکنند. در صورت مغایر بودن این دو مقدار، بایت دریافتی دارای اعتبار نخواهد بود و باید نادیده گرفته شود.

سرعت ارسال

آخرین موردی که در ساخت سیگنال پکت RS232 باید به آن توجه کنید، موضوع عرض هر بیت یا سرعت ارسال است. در پروتکل RS232 هرکدام از بیتهای دیتا، Stop ،Start و پریتی دارای عرض زمانی مشخصی هستند. به عبارت دیگر، فرکانس ارسال بیتها در پروتکل RS232 مقادیر مشخصی هستند که باید از جانب طراح ماجول ارسال و دریافت RS232 مورد توجه قرار گیرد. این فرکانس که در اصطلاح به آن baud rate گفته میشود، دارای چند مقدار مشخص است. بعضی از این

مقادیر عبارتند از: ۱۱۵۲۰۰، ۵۷۶۰۰، ۳۸۴۰۰، ۹۶۰۰ بیت برثانیه پس پیادهساز ماجول فرستنده RS232 باید این موضوع را در نظر داشته باشد و عرض تمام بیتها را بر مبنای

پس پیادشار تا جوان ترسینده 2022، بینا این موضوع را در نظر داشته باسه و فرص نمام بیکند را بر مبتای baud rate مورد نظر طراحی کند.

۳ – **مرحله سوم** همانطور که در بخش قبلی هم اشاره کردیم، یک ماجول کامل دارای دو بخش فرستنده و گیرنده است. این دو بخش می توانند به طور همزمان کار کنند. بنابراین، ارتباط RS232 یک ارتباط Full Duplex است. در این مرحله از آزمایش ماژول فرستنده و گیرنده سریال با نرخ ارسال ۹۶۰۰ را طراحی کنید و ضمن شبیهسازی صحت عملکرد آن را بررسی نمایید. **۴– مرحله چهارم** ضمن پروگرم کردن برنامه بر روی FPGA صحت عملکرد آن را با استفاده از نرم افزار Hyperterminal بررسی نمایید.

نکتها: Hyperterminal نرم افزاری همراه همه نسخههای سیستم عامل Windows است که می تواند به عنوان ترمینال ارتباط سریال استفاده شود.

نکته ۲: در طراحی ماژول فرستنده و گیرنده تمام کد را به صورت سنکرون با یک کلاک بنویسید. به عبارت دیگر، Process در کد شما باید فقط حساس به کلاک سیستم باشد و نه سیگنال دیگری. همچنین تمام کد باید داخل عبارت شرط لبه بالارونده کلاک نوشته شود. اگر مجبور به استفاده از کلاک با فرکانس دیگری هستید، بهتر است در صورت امکان این کلاک ضریب صحیح از کلاک اصلی سیستم باشد. تا جای ممکن از ساختارهای شرطی تو در تو اجتناب کنید و سعی کنید بیش از سه لایه عبارت شرطی تو در تو در ماجول نداشته باشید. همچنین تا جای ممکن از شرطهای ساده استفاده کنید.

خروجیهای مورد انتظار (برای تهیه گزارش کار)

- ارائه مفاهیم تکمیلی در رابطه با این آزمایش
 - گزارش شبیهسازی و سنتز طرح
- مطالعه و پیشنهاد روشها و نرم افزارهای دیگر جهت صحت عملکرد ارتباط
- مشکلاتی که در حین برنامهنویسی، شبیهسازی و پروگرم کردن با آنها مواجه شدید.

تمرين

ماژول فرستنده و گیرنده را برای ارسال و دریافت یک کاراکتر ۸ بیتی با استفاده از IPcoreهای نرمافزار Quarus با نرخ ارسال ۵۷۶۰۰ طراحی کنید و همچینن درصد ضریب خطا آن را بدست آورید.

آزمایش ۹

طراحی و پیادهسازی پروتکل SPI و آشنایی با مبدل های ADC

هدف

- آشنایی با پروتکل SPI
- آشنایی با مبدل ADC0838
- بررسی نحوه ارتباط مبدلهای ADC با FPGA از طریق پروتکل SPI

شرح آزمایش

مبدلهای آنالوگ به دیجیتال امروزه به عنوان پل ارتباطی دنیای واقعی کمیتهای آنالوگ و دنیای پردازش دیجیتال نقش کلیدی در بسیاری از سیستمهای الکترونیکی، مخابراتی و کنترلی را ایفا میکنند. اهمیت اصلی مبدل آنالوگ به دیجیتال در این است که رایانهها فقط قادر به تشخیص صفر و یک هستند. بنابرین باید دنبال راهی برای تبدیل کمیتهای آنالوگ به دیجیتال باشیم که این کار به روش نمونهبرداری انجام میشود. برای این کار در لحظههای مجزای زمانی مقادیری (نمونههایی) برداشته میشود که این نمونهها یک سطح را مشخص میکنند. سرعت نمونه برداری (تعداد نمونهها در واحد زمان) مشخص کننده دقت نمایش یک سطح به صورت تابع ناپیوسته زمانی میباشد.

در این آزمایش هدف دریافت اطلاعات از چهار کانال آنالوگ توسط یک برد FPGA میباشد، که برای دریافت این اطلاعات از مبدل آنالوگ به دیجیتال ADC0838 که یک آیسی هشت کاناله میباشد استفاده کردهایم. پروتکل ارتباطی که برای دریافت اطلاعات در این پروژه به کار گرفته شده است SPI میباشد که FPGA در اینجا نقش Master و مبدل ADC0838 به عنوان Slave به کار رفته است.

۱- مرحله اول

در شکل زیر بلوک دیاگرام SPI نشان داده شده است.



شکل ۷۰- بلوک دیاگرام پروتکل SPI

همانطور که مشاهده می کنید، سیستم دارای دو بخش Master و Slave است. در بخش Master سیستم دارای یک شیفت رجیستر هشت بیتی و مولد پالس ساعت و بخش Slave فقط شامل یک شیفت رجیستر هشت بیتی است. این دو بخش بوسیله سه سیم (غیر از GND) بهم متصل می شوند. Master in slave out) MISO خط انتقال اطلاعات از Slave به Slave Slave Slave انتقال اطلاعات از Master به Slave Slave in) MOSI (master out slave in) MOSI توليد مى شود. (spi clock) SCK خط انتقال پالس ساعت که توسط Master توليد مى شود. (slave select) SS (فعال با سطح پايين)

آغازکننده ارتباط همیشه Master است و فقط Master میتواند شروع به انتقال کند و Slave باید منتظر دریافت اطلاعات بماند. واحد تولید پالس ساعت در Master وظیفه ایجاد پالس ساعت برای شیفت رجیسترهای Master و Slave (از طریق پایه SCK) را جهت همزمانسازی به عهده دارد. باتوجه به شکل قبل شیفت رجیسترهای Master و NISO به صورت هشت بیتی بوده و از طریق خطوط MISO و MOSI بصورت چرخشی بهم متصل شدهاند و بدین معنی است که اگر دادهای از Master به سمت Slave فرستاده شود، میتوان درهمان زمان و درجهت عکس، دادهای از Slave به سمت Master به سمت Slave و در کل، این دوشیفت رجیستر هشت بیتی تشکیل یک شیفت رجیستر شانزده بیتی را میدهند. با نوشتن یک بایت در رجیستر داده SPDR (SPDR) مورد مولدپالس ساعت Slave میش می می داده ای از Slave بالس نمود، و در کل، این دوشیفت رجیستر هشت میدهد. پس از انتقال یک بایت ازاطلاعات، مولد پالس ساعت اوزار، هشت بیت داده را به سمت Slave، شیفت

مد Slave

درمد Slave پین SS بعنوان ورودی شناخته می شود. اگر این پین در وضعیت Low قرار گیرد، SPI فعال شده و MISO بصورت خروجی توسط کاربر تعریف و بقیه پینهای SPI ورودی می باشند. به منظور غیر فعال نمودن SPI می توان پین SS را High نمود، دراین حالت همهی پینهای SPI ورودی شده و SPI غیرفعال خواهد شد. از این قابلیت می توان به منظور بکار گیری یک Master و چند Slave استفاده کرد.

مد Master

در این مد کاربر میتواند جهت پین SS را تعیین نماید. اگر SS بعنوان ورودی تعریف شود بایستی آن را از خارج High نمود تا در عملکرد Master مشکلی پیش نیاید. اگر این پین از خارج Low شود رابط SPI را به عنوان فرمانی برای رفتن به مد Slave تلقی کرده و وارد مد Slave شده و شروع به دریافت داده می کند.

۲- مرحله دوم

در این مرحله از آزمایش قصد داریم با یکی از مبدلهای ADC به نام ADC0838 آشنا شویم. سری ADC0831 مبدلهای A/D هشت بیتی با یک سریال I/O و مولتی پلکسرهای ورودی قابل تنظیم، تا هشت کانال هستند. سریال I/O برای این که با سریال اطلاعات استاندارد تعویضی NSC MICROWIRE و ارتباط راحت با خانوادهی پردازنده های COPS مطابقت پیداکند پیکربندی شده است و میتواند با شیفت رجیسترهای استاندارد یا میکروپروگرمها اتصال برقرار کند. مولتی پلکسرهای دو، چهار، یا هشت کاناله نرم افزار طراحی شده، برای ورودی های متغیر یا تک پایانه درست همانند کانال علامت هستند. ولتاژ مرجع ورودی میتواند تغییر بکند تا اجازهی انکود کردن هر صفحهی ولتاژ کوچکتر تا رزولوشن کامل هشت بیت را بدهد.

ویژگیهای ADC 0838

✓ تطابق NSC MICROWIRE واسط مستقیم با پردازندههای خانواده COPS
 ✓ ارتباط آسان با تمام پردازندهها یا راه اندازهای Stand-alone
 ✓ راه انداز نسبی با ولتاژ مرجع پنج ولت
 ✓ نیاز نداشتن به تنظیمات مورد نیاز کامل
 ✓ انتخاب مولتی پلکسر دو، چهار یا هشت کاناله با آدرس منطقی
 ✓ تنظیم کنندهی انحرافی که اجازهی عمل کردن با منابع ولتاژ بالا را میدهد
 ✓ محدودهی ورودی ۰ تا ۵ ولت با منبع تغذیهی ۵ ولتی
 ✓ سازگار با ورودی /خروجی داده سریال دیجیتال
 ✓ سازگار با ورودی/خروجی TTL/MOS
 ✓ دستورالعمل کاربردی
 ✓ آدرس دهی مولتی پلکسرها

چینش پایههای این مبدل به شکل زیر است:



شکل ۷۱- چینش پایههای ADC 0838

یک ترکیب مشخص ورودی بین آدرسدهی متوالی MUX، مقدم برآغاز بحث قرار گرفته است. آدرس MUX انتخاب میکند کدام یک از ورودیهای آنالوگ باید فعال شوند و آیا این ورودی تک پایانه یا متغیر است. در حالت متغیر همچنین قطب کانالها را مشخص میکند. ورودیهای متغیر با جفتهای کانال مجاور محدود شدهاند. برای مثال کانال ۰ و کانال ۱ شاید به عنوان یک جفت متغیر انتخاب شوند ولی کانال ۰ یا ۱ نمیتوانند به صورت مجزا با هیچ کانال دیگری کار کنند. به علاوه برای انتخاب کد متغیر علامت نیز شاید انتخاب شود. کانال ۰ شاید به عنوان ورودی + انتخاب شود و کانال ۱ به عنوان ورودی منفی یا بالعکس.

این برنامهریزی به بهترین حالت توسط کدهای آدرسدهی MUX در جدولهای زیر برای ADC0838 نشان داده شده است.

	MUX Ac	dress				Analo	g Sing	le-En	ded Cl	nannel	#	
SGL/	ODD/	SEL	ECT	0	1	2	3	4	5	6	7	COM
DIF	SIGN	1	0									
1	0	0	0	+								-
1	0	0	1			+						-
1	0	1	0					+				-
1	0	1	1							+		-
1	1	0	0		+							-
1	1	0	1				+					-
1	1	1	0						+			-
1	1	1	1								+	-

Single-Ended MUX Mode

جدول۳- آدرس دهی Single-Ended MUX

Differential MUX Mode

MUX Address				Analog Differential Channel-Pair #							
SGL/	ODD/	SEL	ECT	(0 1		1	2		3	
DIF	SIGN	1	0	0	1	2	3	4	5	6	7
0	0	0	0	+	-						
0	0	0	1			+	-				
0	0	1	0					+	-		
0	0	1	1							+	-
0	1	0	0	-	+						
0	1	0	1			-	+				
0	1	1	0					-	+		
0	1	1	1							-	+

جدول۴- آدرس دهی Differential MUX

آدرس MUX بوسیلهی خط DI به مبدل انتقال پیدا می کند. چون ADC0831 فقط یک کانال ورودی متغیر با یک علامت قطبی ثابت را شامل می شود به آدرس دهی نیازی ندارد.

یک خصوصیت مهم این مبدلها لینک سریال با پردازنده ی کنترل خود است. استفاده از یک فرمت ارتباط سریالی دو پیشرفت بارز سیستم را ارائه می دهد. آن اجازه می دهد تا توابع بیشتری در پکیج مبدل با عدم افزایش اندازه ی پکیج شامل بشوند و می توانند ارسال سیگنالهای سطح پایین آنالوگ را بوسیله ی موقعیت یابی مبدل درست در حسگر آنالوگ از بین ببرد. ارسال پارازیت بالا اطلاعات دیجیتال را در مقابل پردازنده ی میزبان ایمن می سازد.

برای درک عملکرد این مبدل بهتر است به دیاگرام زمانبندی شکل زیر مراجعه کنیم.



شكل ۷۲- دياگرام زمانبندي ADC0838

۱-یک تبدیل با اولین Active low خط CS راهاندازی می شود. این خط باید برای کل تبدیل Active low با است.

۲-یک کلاک بوسیلهی پردازنده ساخته میشود. (اگر دائما فراهم نشده باشد)

۳-در هر لبه بالارونده کلاک، وضعیت داده در خط داخل ثبات شیفت آدرس ماکس ثبت می شود. بیت شروع در واقع اولین یک منطقی است که در روی خط ظاهر می شود. (تمام صفرهای قبل از اولین یک نادیده گرفته می شوند) پیرو بیت شروع، مبدل انتظار دارد تا بیت های بعدی دوم تا چهارم کلمه تخصیص ماکس باشند.

۴-وقتی که بیت شروع به محل شروع ثبات ماکس شیفت داده بشود، کانال ورودی تخصیص داده می شود. یک وقفه به مدت ۵۰٪ کلاک، به منظور اجازه به کانال ماکس انتخاب شده به صورت خودکار برای قرار گرفتن درج می شود. خط وضعیت SAR در این زمان فعال می شود تا مخابره کند که یک تبدیل در حال حاظر در حال وقوع است و خط DI از کار انداخته شده است.

۵-اکنون خط DO از وضعیت سه حالته بیرون میآید و یک صفر آماده را برای یک دوره کلاک در زمان راه اندازی ماکس آماده میکند.

۶-زمانی که تبدیل شروع می شود، خروجی مقایسه کننده SAR که نشان می دهد آیا ورودی آنالوگ بزرگتر یا کوچکتر از هر ولتاژ متوالی از مقاومت داخلی است، درخط DO از هر لبه پایین رونده کلاک ظاهر می شود. این داده نتیجه تبدیلی است که بیرون شیفت داده می شود و می تواند بلافاصله توسط پردازنده خوانده شود. ۲-بعد از هشت کلاک تبدیل کامل می شود. خط وضعیت SAR به جهت نمایش این چرخه بعد از ۵۰٪ کلاک در آینده به حالت LOW باز می گردد. ۸-همه ثباتهای داخلی وقتی که خط CS در حالت high باشد پاک می شوند. اگر تبدیل دیگری تقاضا شود CS باید توسط اطلاعات آدرس حالت high را به wol انتقال بدهد. خطوط ID و OD می توانند با یکدیگر گره زده شوند و به واسطه یک پردازنده تک بیتی دوسویه با یک سیم، کنترل شوند. این امکان دارد زیرا ID تنها در مدت وقفه آدرس دهی ماکس دیده می شود. در حالی که خط OD هنوز در وضعیت امپدانس بالا

۲- مرحله دوم طبق دیاگرام زمان بندی ADC0838 نحوه عملکرد آنرا را بر اساس پروتکل SPI طراحی کنید.

۳ – مرحله سوم طرح ایجاد شده را شبیهسازی کرده و آنرا با دیاگرام زمان بندی ADC0838 مقایسه کنید.

خروجیهای مورد انتظار (برای تهیه گزارش کار)

- ارائه مفاهیم تکمیلی در رابطه با این آزمایش
 - گزارش شبیهسازی و سنتز طرح
- مشکلاتی که در حین برنامهنویسی، شبیهسازی و پروگرم کردن با آنها مواجه شدید.

تمرين

در مورد ۵ نوع مبدل آنالوگ به دیجیتال که بر اساس پروتکل SPI کار می کنند و نحوه عملکرد آنها گزارشی ارائه دهید.

آزمایش ۱۰

طراحی و پیادهسازی پروتکل I2C و آشنایی با سنسورهای دما

هدف

- آشنایی با پروتکل I2C
- آشنایی با سنسور دما LM75
- بررسی نحوه ارتباط سنسور دما با FPGA از طریق پروتکل I2C

شرح آزمایش

سنسورهای دما یکی از پرکاربردترین تجهیزات و سنسورهای صنعت میباشند که تقریبا تمام مهندسینی که در زمینه ابزاردقیق و اتوماسیون صنعتی فعالییت میکنند باید این سنسورها را بشناسند، نقاط قوت و ضعف آنها را بدانند و با راه اندازی و اتصال آنها به نمایشگر، ترانسمیتر و سیستمهای کنترل آشنا باشند. سنسورهای دما در تمام صنایع اعم از صنایع نفت، گاز، پتروشیمی، ریخته گری، غذایی، شیمیایی، آجر و سرامیک و صنایعهای کوچکتر، کاربرد بسیار فراوانی دارد. از موارد مهمی که باید به آن اشاره کرد شناخت در حوزه سنسورها میباشد که زمانی یک شخص قدرت انتخاب سنسور مناسب برای یک پروسه را دارد که آشنایی و تجربه بسیار خوبی در مورد سنسورهای دما و روشهای اندازه گیری دما داشته باشد.

در پروسه کنترل، ثبت، اندازه گیری، و نمایش حرارت یک سیستم یا شئ اختلاف بسیاری زیادی بین مفاهیم "سنسور حرارت" و "اندازه گیری حرارت" وجود دارد. یک دماسنج جیوهای معمولی میتواند به آسانی برای اندازه گیری دمای اتاق، یک مایع و ... مورد استفاده قرار گیرد، در حالیکه از آن نمیتوان برای ثبت و کنترل دمای محیط یا شئ مورد اندازه گیری استفاده نمود. متقابلاً یک سنسور گرما نمیتواند برای نشان دادن دمای محیطی که در آن قرار گرفته شده است به تنهایی مورد استفاده قرار گیرد. سنسورهای حرارت را میتوان بطور کلی به دو گروه تماسی و غیرتماسی تقسیم کرد.

سنسور تماسی یا Contact Sensor برای اندازه گیری دمای محیط در واقع دمای خودش را اندازه گیری می کند. با تماس این سنسور به شئ تحت کنترل یا قرار گرفتن آن در محیط تحت اندازه گیری، تعادل گرمایی بین سنسور و محیط ایجاد می شود. در این حالت جریان گرما یا Heat Flow بین محیط و سنسور وجود ندارد. در سنسورهای حرارتی غیرتماسی آنچه سنجیده میشود توان حرارتی مادون قرمز یا نوری متصاعد شدهای است که از یک سطح (یا جسم) با مساحت (یا حجم) مشخص یا قابل محاسبه دریافت میگردد. علاوه بر این، روشهای پیشرفته ترموگرافی با تصویربرداری از اجسام و تجزیه و تحلیل تصاویر دریافتی که قادر به اندازه گیری دقیق دمای اجزای مختلف جسم است نیز در زمره سنسورهای حرارتی غیرتماسی قرار میگیرد. سنسورهای حرارتی تماسی تنوع و فراوانی بسیار بیشتری نسبت به نوع غیرتماسی دارند. این سنسورها شامل: انواع ترموکوپلها TC، مقاومتهای RTD و RTP، ترمیستورها بی متالها، ترمومترهای شیشهای، ترموولها، و انواع نیمه هادی شامل دیود، ترانزیستور و آی سیهای اندازه گیری و کنترل دما هستند. ترموسنسورهای غیرتماسی نیز شامل ترمومترهای مادون قرمز و لیزری، تصویربرداری حرارتی و انواع طیف سنجهای نوری است. بطورکلی این دسته از سنسورهای حرارتی بر مبنای قابلیت طیف منتشرشده اندازه گیری را صورت میدهای نوری است. بکارگیری این گروه از سنسورها در صنعت به فراگیری TDRها و ترموکوپلها نرسیده است اما کارآیی غیرقابل این دسته از سنسورهای حرارتی بر مبنای قابلیت طیف منتشرشده اندازه گیری را صورت میدهد. هرچند هنوز بکارگیری این گروه از سنسورها در صنعت به فراگیری TDRها و ترموکوپلها نرسیده است اما کارآیی غیرقابل میرود.

در ادامه آزمایش قصد آن داریم تا با معرفی یکی از انواع سنسورهای حرارتی به مطالعه ساختار و بررسی نحوه پیاده سازی آن بپردازیم.

۱- مرحله اول

در این مرحله از آزمایش قبل از معرفی سنسور دما قصد داریم با پروتکل I2C آشنا شویم. در دنیای صنعتی امروز، انتقال دیتا به عنوان یکی از مهمترین بخشهای فرایندهای کنترلی شناخته شده است و طراحان در تلاشاند تا پروتکلهای ارتباطی جدید با ایمنی و سرعت بالا طراحی نمایند. یکی از این پروتکلهای معروف، I2C است که اولین بار توسط شرکت فیلییس در دهه ۱۹۸۰ جهت ارتباط

یعی از این پروعی این از میروغان علیه است که به دلیل سادگی و سرعت عمل بالای آن مورد توجه سایر دستگاههای TV با پردازشگر ابداع شده است که به دلیل سادگی و سرعت عمل بالای آن مورد توجه سایر تولیدکنندگان نیز قرار گرفت.

ارتباط به طریق پروتکل I2C فقط به دو سیم نیاز دارد. این دو سیم در بهترین حالت توانایی آدرس دهی ۱۰۲۴ وسیله را دارند. این پروتکل برعکس SPI میتواند چند Master داشته باشد. اما آنها قادر به صحبت با یکدیگر نیستند و از یک باس برای ارتباط با slaveها استفاده میکنند.

وسایل متصل به I2C میتوانند با فرکانس 100kHz تا 400kHz با هم ارتباط برقرار کنند که کمتر از سریال Uart و SPI است. در این پروتکل بعد از ارسال هر ۸ بیت داده یک بیت ACK یا ACK نیز فرستاده میشود. سختافزار مورد نیاز برای استفاده از I2C پیچیده تر از SPI ولی سادهتر از سریال Uart است و به لحاظ نرم افزاری به سادگی قابل اجرا است. هر باس I2C از دو سیگنال تشکیل شده است: SCL که کلاک سیگنال است و SDA که داده سیگنال است. کلاک توسط Master تولید می شود. برعکس Uart و SPI در ایورهای باس در I2C از نوع Open Drain هستند. یعنی می توانند خط سیگنال را low کنند اما توانایی high کردن آن را ندارند. بنابراین امکان آسیب دیدن در ایور وسایل زمانی که یکی بخواهد سیگنال را high و دیگری بخواهد آن را wol کند وجود ندارد و اتلاف توان هم کاهش می یابد. هر سیگنال با مقاومت پول آپ شده است تا در صورتی که هیچ وسیله ای نخواهد آن را wol کند، خودبه خود high شود. مقاومت هول آپ معمولا ۴٫۷کیلو هستد و در صورت نیاز کاهش می یابند. این پروتکل در فواصل کوتاه ۲ تا ۳ متری قابل استفاده است اما برای فواصل زیاد و یا سیستمی با تعداد زیادی وسیله مرتبط به باس، باید مقدار پول آپ را کاهش داد.

با توجه به اینکه درایورها سیگنال را high نمی کنند امکان اتصال وسایل با سطح ولتاژ ورودی و خروجی متفاوت وجود دارد. مثلا اگر ولتاژ تغذیه دو وسیله با هم تفاوت داشته باشند مقاومت پول آپ را به ولتاژ تغذیه پایین تر وصل می کنیم. البته این کار تنها زمانی مقدور است که این ولتاژ به اندازه کافی برای high کردن ورودی دستگاه دیگر بالا باشد. در غیر این صورت حتما باید از یک مدار یا قطعه تغییر سطح یا level shifter استفاده نمود.

پروتکل پیام از دو قاب تشکیل میشود: قاب آدرس که Master در آن مشخص میکند که پیام به کدام Slave ارسال میشود و قاب (یا قابهای) داده که به صورت ۸ بیتی تبادل میشوند. وقتی که SCL پایین میشود داده روی SDA قرار می *گ*یرد و پس از بالا شدن SCL نمونه برداری میشود.



شکل ۷۳- دیاگرام زمانبندی پروتکل I2C

شرایط شروع برای آغاز یک قاب آدرس، دستگاه Master پایه SCL را بالا و SDA را پایین نگه میدارد. این باعث می شود همه slaveها متوجه شوند که قرار است یک داده تبادل شود. اگر دو تا Master به طور همزمان بخواهند کنترل باس را در اختیار بگیرند دستگاهی که زودتر SDA را پایین نگه دارد کنترل باس را در اختیار می گیرد. قاب آدرس ۷ بیتی ابتدا پرارزشترین بیت ارسال میشود و سپس بیت خواندن و نوشتن ارسال میشود که مشخص می کند قرار است داده خوانده یا نوشته شود که بیت ۰ به معنای نوشتن و بیت ۱ به معنای خواندن است. بیت نهم این قاب بیت ACK و یا NACK است. وقتی هر هشت بیت قاب فرستاده شدند، دستگاه گیرنده کنترل SDA را به عهده می گیرد. اگر گیرنده پیش از نهمین پالس کلاک خط SDA را پایین نکشد به این معنی است که گیرنده داده را دریافت نکرده یا نمیداند چگونه آن را تجزیه و تحلیل کند. در این صورت فرایند متوقف میشود و بر عهده تعده می آمید که در این باره تصمیم گیری کند.



قاب های داده وقتی قاب آدرس فرستاده شد، ارسال داده میتواند آغاز شود. Master پالسهای کلاک را با فواصل زمانی مشخص تولید می کند و داده با توجه به وضعیت بیت R/W توسط Master و یا Slave روی خط SDA قرار می گیرد. تعداد قابهای داده دلخواه است و رجیستر داخلی بیشتر دستگاههای Slave به طور خود کار افزایش می یابد.

شرایط توقف وقتی همه قابهای داده فرستاده شدند، Master شرایط توقف را تولید می کند. به این طریق که ابتدا SCL از ۰ به ۱ میرود و در همین وضعیت میماند و SDA نیز از ۰ به ۱ میرود.

۲- مرحله دوم

بعد از آشنایی با پروتکل I2C در این مرحله از آزمایش قصد داریم با سنسور حرارتی IM75 آشنا شویم. IM75 یک سنسور حرارتی همراه با مبدل آنالوگ به دیجیتال (یا به بیانی تغییرات دلتا – سیگما) و دارای آشکارساز دیجیتالی افزایش بیش از حد دما ازطریق رابط I2C میباشد. این سنسور قادر به خواندن دما در کسری از ثانیه است، IM75 میتواند به عنوان کلید حرارتی مستقل نیز عمل کند. برای چنین کاری میتوان بیشینه و کمینهی دما را برای تغییر وضعیت کلید برنامهریزی کرد وقتی دمای اندازی گیری شده از محدوده بیشینه و کمینهی دما را برای تغییر وضعیت کلید برنامهریزی کرد وقتی دمای اندازی گیری شده از محدوده برای کامپیوتر یا میکروکنترلر به کار برد. هنگام روشن شدن مدار، محدودههای بالایی و پایینی حرارتی بر روی ماک درجه و ۸۰ درجه تنظیم میشود. این سنسور به صورت SMT و برای کار با ولتاژ تغذیه ۳٫۳ تا ۵ ولت ساخته شده است و می تواند دمای موردنظر را با تفکیکپذیری نیم درجه از ۵۵ – تا ۱۲۵ + درجه سانتیگراد اندازه گیری کند. یکی از دیگر محاسن این سنسور آن است که جریان کاری لازم در حدود ۲۰۵ میکروآمپر است که این جریان در بعضی موارد به یک میلی آمپر هم میرسد و در حالت خاموش در حدود یک میکرو آمپر است

درشکل زیر پایههای سنسور را مشاهده میکنید.



PIN	NAME	FUNCTION
1	SDA	Serial-Data Input/Output Line. Open drain. Connect SDA to a pullup resistor.
2	SCL	Serial Clock Input. Open drain. Connect SCL to a pullup resistor.
3	OS	Overtemperature Shutdown Output. Open drain. Connect OS to a pullup resistor.
4	GND	Ground
5	A2	2-Wire Interface Address Input. Connect A2 to GND or +Vs to set the desired I ² C bus address. Do not leave unconnected (see Table 1).
6	A1	2-Wire Interface Address Input. Connect A1 to GND or $+V_S$ to set the desired I ² C bus address. Do not leave unconnected (see Table 1).
7	AO	2-Wire Interface Address Input. Connect A0 to GND or +V _S to set the desired I ² C bus address. Do not leave unconnected (see Table 1).
8	+Vs	Positive Supply Voltage Input. Bypass to GND with a 0.1µF bypass capacitor.

جدول ۵- توضیح عملکرد پایههای سنسور LM75

این سنسور ۸ پایه دارد. پایههای ۴ و ۸ پایههای تغذیه سنسور هستند و بنا به مدل سنسور از ۳٫۳ تا ۵ ولت تغییر میکنند در ضمن برای تغذیه حتما یک خازن ۰٫۱ میکرو فاراد باید به پایه ۸ وصل شود.

پایه شماره ۳ میتواند هم به عنوان یک وقفه برای پروسسور استفاده شود هم میتواند محدوده بالایی را مشخص کند. یعنی این که وقتی دما به محدوده بالایی (۲۵ درجه) رسید این سنسور این پایه را صفر می کند (این پایه AO (ACTIVE LOW) است. پایه شماره ۱ و ۲ یعنی SDA و SCL پایههای ارتباط I2C سنسور هستند که اغلب به میکروکنترلری که قادر به ارتباطات اینترفیس باشد وصل میشوند. پایههای ۵ و ۶ و ۷ این سه پایه یعنی AO و 1 A و A2 این سه پایه آدرسهای اینترفیس سنسور هستند. در پروتکل Master ، I2C باری ارتباط با Slave آنها را با یک اسم میشناسد که ۷ بیت اول نام سنسور است و بیت ۸ به سنسور اعلام میکند که توسط این سه پایه تعریف میشود که بهتر است هر سه صفر یا یک باشند. همانطور که گفته شد بعد از ارتباط با سنسور و صدا کردن آن توسط بیت هشتم وضعیت سنسور را تعیین میکنیم یعنی به سنسور اعلام میکنیم که اطلاعات بفرستد یا اطلاعات بگیرد که ما این وضعیت را ۱ قرار داده و سنسور را در حالت فرستنده قرار میدهیم تا میزان دما را برای ما بفرستد

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
1	0	0	1	A2	A1	AO	R/W

LM75	سنسور	آدرس	رجيستر	شکل ۷۶-
------	-------	------	--------	---------

دیاگرام زمان،ندی سنسور برای read کردن رجیستر وضعیت به صورت زیر می،اشد.



شکل ۷۷- دیاگرام زمانبندی سنسور LM75

۳- مرحله سوم

بعد از آشنایی با سنسور دما در این مرحله از آزمایش طبق دیاگرام زمان بندی LM75 نحوه عملکرد آنرا را بر اساس پروتکل I2C طراحی کنید. در ادامه طرح ایجاد شده را شبیهسازی کرده و آنرا با دیاگرام زمان بندی LM75 مقایسه کنید.

خروجی های مورد انتظار (برای تهیه گزارش کار)

- ارائه مفاهیم تکمیلی در رابطه با این آزمایش
 - گزارش شبیهسازی و سنتز طرح
- مشکلاتی که در حین برنامهنویسی، شبیهسازی و پروگرم کردن با آنها مواجه شدید.

تمرين

در مورد یکی دیگر از سنسورهای پرکاربرد دما به نام DS18B20 و نحوه عملکرد و پیادهسازی آن گزارشی ارائه دهید.

آزمایش ۱۱

طراحي واحد كنترل ديجيتال يك سيستم انكوباتور

هدف

- آشنایی با واحد کنترل دیجیتال
- آشنایی یا سیستمهای انکوباتور
- نحوه طراحي واحد كنترل سيستمهاي انكوباتور

شرح آزمایش

انکوباتور چیست؟

انکوباتور (به انگلیسی Incubator) ابزاری است آزمایشگاهی که در آزمایشگاههای بیولوژی برای کشت و رشد نمونههای زنده مانند سلولها یا میکروبها به کار میرود. این وسیله با کنترل رطوبت، دما، میزان اکسیژن و دیاکسید کربن شرایطی مناسب برای رشد سازوارههای زنده فراهم میکند. انکوباتور یکی از از ابزار های مهم در آزمایشهای میکروبیولوژی، زیست شناسی سلولی و... به حساب می آید. این وسیله توسط پزشک کودکان فیلیپینی فیدل موندو اختراع شده است.همچنین انکوباتور وسیله، دستگاه یا محلی است که گرمای لازم و مناسب تولید جوجه از تخم مرغ فراهم میکند. همچنین به وسیلهای میگویند که با ایجاد گرما، زمینه را برای ادامه حیات نوزادان زودرس پس از تولد فراهم میآورد. در ادبیات کارآفرینی، انکوباتورها یا مراکز رشد، مراکزی هستند که برای پرورش یا ایجاد کسب و کارهای کوچک ایجاد میشوند. انکوباتورها طرح هایی مبتنی بر نوآوری هستند و از کادر مدیریتی کوچکی تشکیل میشوند. این مراکز دارای مکان فیزیکی و تسهیلات مشترک میباشند.

Incubator = دستگاهی که گرما را حفظ می کند/ نگه می دارد.

در این آزمایش قصد داریم واحد کنترل دیجیتال یک سیستم انکوباتور (Incubator) را طراحی و پیادهسازی کنیم.

۱- مرحله اول

در این سیستم مطابق شکل زیر یک حسگر دما، یک واحد خنک کننده (Cooler) مجهز به پنکه (Fan) و یک واحد گرم کننده (Cooler) مجهز به پنکه (Fan) و یک واحد گرم کننده (Heater) وجود دارد.



شکل ۷۸- نمای کلی یک سیستم انکوباتور

حسگر دما، دمای محفظه را که میان 10- تا 60+ درجه سانتیگراد متغیر است میخواند و در قالب یک عدد ۸ بیتی به سیستم شما تحویل میدهد. دما هر دقیقه یک بار از حسگر دریافت می شود و براساس آن واحد کنترل دیجیتال تصمیم میگیرد که:

۱-چگونه واحدهای گرمکننده و سردکننده را فعال و غیرفعال کند.
۲-چگونه در صورت فعال بودن واحد سرد کننده دور Fan آن را تنظیم کند.
روش کار واحد کنترل دیجیتال توسط دو نمودار حالت زیر توصیف شده است: در این نمودار حالت برچسبهای مربوط به یالها نشان دهنده تغییرات دمایی هستند که باعث تغییر حالت در واحد کنترل دیجیتال شده و باعث واکنش سیستم به شکل روشن و خاموش شدن Heater و یا تغییر دور Fan مربوط به Cooler می موند.



شكل ٧٩- نمودار حالات سيستم انكوباتور

دقت کنید که در نمودارهای حالت فوق نمودار حالت سمت راست زمانی فعال می شود (از حالت out خود خارج می شود) که نمودار حالت سمت چپ در حالت S2 که در آن Cooler روشن است قرار گرفته باشد و اگر نمودار out سمت چپ در حالت S2 که در آن cooler روشن است قرار گرفته باشد و اگر نمودار حالت سمت چپ در حالت S2 خود نباشد نمودار حالت سمت راست غیر فعال می شود (یعنی وارد حالت out to حود می شود) که نمودار حالت و S2 خود نباشد نمودار حالت سمت راست غیر فعال می شود (یعنی وارد حالت out حود می شود) که نمودار حالت S2 خود نباشد نمودار حالت سمت راست غیر فعال می شود (یعنی وارد حالت out to حود می شود) چون معنی ندارد که وقتی Cooler خاموش است دور آن تنظیم گردد. در این آزمایش هدف طراحی واحد کنترل دیجیتال است و قسمتهای حسگر، Heater و roler وجود خارجی ندارند. لذا در آزمایشگاه باید دانشجویان عددی را که مثلاً توسط حسگر دما خوانده می شود خود با استفاده از امکانات بورد FPGA به مدار اعمال کنند و واکنش سیستم و همچنین دور را به جای آنکه Heater و Cooler واقعی روشن و خاموش شدن LED می می می می دور با به جای آنکه Heater و S2 دوش وا

۲- مرحله دوم

حال که با یک نوع سیستم انکوباتور آشنا شدید نحوه عملکرد آن را طراحی کنید و ضمن شبیهسازی صحت عملکرد آن را بررسی نمایید.

۳- مرحله سوم

طرح را بر روی برد مربوطه آزمایشگاه پروگرم کرده و با استفاده از امکانات برد صحت عملکرد آن را بررسی کنید.

خروجیهای مورد انتظار (برای تهیه گزارش کار)

- ارائه مفاهیم تکمیلی در رابطه با این آزمایش
 - گزارش شبیهسازی و سنتز طرح
- مشکلاتی که در حین برنامهنویسی، شبیهسازی و پروگرم کردن با آنها مواجه شدید.

تمرين

در مورد سیستم مدیریت ساختمان یا BMS گزارشی ارائه دهید و یک نوع سیستم BMS را نیز طراحی کنید.

آزمایش ۱۲

آشنایی با پردازنده های Embedded شده در FPGA (MicroBlaze)

هدف

- آشنایی با معماری و کاربرد پردازندههای MicroBlaze
- نحوه ایجاد پردازندههای تعبیه شده در FPGA با استفاده از نرم افزار ISE
 - پیادہسازی محیطہای Embedded
 - استفاده از نرم افزار SDK

شرح آزمایش

با وجود این که تراشههای FPGA برای پیادهسازی الگوریتمهای موازی بسیار مناسب هستند، الگوریتمهای ترتیبی و بالاخص الگوریتمهایی که نیاز به قدرت پردازش بالایی ندارند، ارجح است بخاطر صرفهجویی در زمان و تسهیل امر پیادهسازی، به صورت برنامهای نرمافزاری روی یک ریزپردازنده پیادهسازی شوند. سیستمهای پیچیده ی امروزی، نیاز به حافظهای با حجم زیاد، ریزپردازنده ای پُرسرعت، یک پردازنده ی سیگنال دیجیتال، و چندین نوع واسط سیستمی به منظور ارتباط با تراشهها، سیستمها و بوردهای دیگر دارند. بکارگیری

سیستمی متشکل از پردازنده و حافظهی خارجی، منجر به کاهش کارائی سیستم و افزایش هزینهها می گردد. چنانچه این دو (پردازنده و حافظهی خارجی)، با هم یکپارچه شوند، آنگاه با کاهش مصرف توان و تسهیل پیچیدگی امر طراحی و مسائل یکپارچهسازی و اتصالات، میتوانیم روی ساخت یک سیستم سریعتر فکر کنیم و تنگناهای کارائی را برطرف نمائیم. همچنین ادغام قابلیتهای پردازشی در FPGAهای موجود بر بوردهایی که در حال حاضر برای کارهای کنترلی استفاده میشوند، میتواند عملکرد سیستم را افزایش داده و هزینه یکل سیستم (هزینه طراحی، یکپارچهسازی، ساخت و ...) را کاهش دهد.

FPGAهای ارائه شده توسط شرکت Xilinx برای سالهاست که به یک سختافزار مهم برای واسطهای کنترلی و سیستمی تبدیل شده است، اما چنانچه فاقد قابلیتهای پردازشی باشند، این مورد مهم، ممکن است بسته به کاربرد هدف، به وسیلهی پردازندههای خارجی برآورده گردد.

بسیاری از شرکتها، تمام تلاششان را روی گسترش مفهموم «سیستم روی تراشه» متمرکز کردهاند تا با افزودن مجموعهای از سختافزارهای جدید، قابلیتهای اضافیای را بر روی یک تک تراشه ایجاد نمایند. این تراشهها، یک رامحل بسیار ارزنده در دورنمای رقابت برای عرضهی تولیدات نوین و کاهش زمان مورد نیاز برای ارائهی آنها به بازار مصرف، میباشد. در سالهای اخیر و به دنبال کاهش هزینههای ساخت و افزایش تعداد PCOreها، فناوری FPGA در بازارهای گوناگون به یک سختافزار مطلوب تبدیل شده است. این فنآوری، با کمک روشهای بهینهسازی معماری، فناوریهای جدید ساخت و افزودن هستههای پردازشی جاسازی شدهی سخت و نرم، تکمیل شده است. امروزه امکان پیادهسازی و استفاده از میکروکنترلرها با استفاده از SoftCoreها قابلیتی مهم است. با استفاده از این PCOreها می توان نیازمندیهای کنترلی را در کنار بخش پردازشی برطرف نمود. میکروبلیز و NIOS هستههای میکروپروسسوری هستند که به صورت تمام سفارشی به ترتیب بر روی FPGAهای شرکت Xilinx و Alter قابل پیاده سازی هستند.

در این قسمت قصد آشنایی با میکروبلیز (MicroBlaze) را داریم.

باید بدانیم این IPCore در تمامی تراشههای شرکت Xilinx قابل پیادهسازی بوده و با توجه به امکانات لازم میتوان آن را به صورت سفارشی مورد استفاده قرار داد. میکروکنترلرهایی که با استفاده از میکروبلیز پیادهسازی میشوند به دلیل استفاده از گیتهای درون FPGA در مقایسه با میکروکنترلرهای موجود در بازار از سرعت پایینتری برخوردار خواهند بود. ولی میتوان انتظار داشت در حد میکروکنترلرهای رده پایین بازار همچون سری AVR عمل کنند. در FPGAهای سری Spartan، پروتکل ارتباطی بر اساس PLB بوده و از سرعت بالایی برخوردار نیست، در مقابل برای تمام سریهای دیگر نظیر Kintex و Xintex با استفاده از ارتباط IX درون تراشه سرعت بالاتری بین بخشهای پردازشی و MicroBlaze برقرار خواهد بود.

- پیادہسازی سخت افزار در محیط EDK درون
 - پیادہسازی نرم افزار بوسیله SDK

در این آزمایش، هدف ما طراحی و ایجاد یک میکروبلیز (Soft Core) به عنوان Embedded Processor میباشد که قصد داریم آنرا درون FPGA طراحی کرده و توسط پروتکل UART یک پیغام را از طرف PC توسط این میکروبلیز دریافت و پاسخ دهیم.

> ۱- مرحله اول در این مرحله از آزمایش قصد داریم یک میکروبلیز را درون FPGA طراحی کنیم.

برای این منظور مطابق شکل زیر ابتدا یک پروژه جدید با نام MicroBlaze_HUT ایجاد کنید و تراشه XC3S400-PQ208 را برای آن انتخاب کنید تا در گام های بعدی، میکروبلیز را درون آن طراحی نمایید.

New Project	Wizard	×
Create New Pro	ect and type. ons, and comment for the project	
Name:	MicroBlaze_HUT	
Location:	C:\Users\mohsen\Desktop\New folder\MicroBlaze_HUT	
Working Directory: Description:	C:\Users\mohsen\Desktop\New folder\MicroBlaze_HUT	
Select the type of to Top-level source typ HDL	p-level source for the project	
More Info		Next Cancel

شکل ۸۰- پنجره ایجاد پروژه در نرمافزار Xilinx

حال بر روی نام تراشه راست کلیک کرده و New Source را میزنیم و در پنجره ویزارد باز شده گزینه Embedded Processor را انتخاب کرده و نام فایل را انتخاب می کنیم و گزینه Next را میزنیم.

VHDL Library VHDL Package VHDL Test Bench Embedded Processor

شکل ۸۱- ینجره New Source Wizard در نرمافزار Xilinx

با انتخاب گزینه Next نرم افزار XPS باز می شود. از آنجایی که یک پروژه خالی ایجاد کردهایم در گام اول از ما سوال می کند که آیا مایلیم از ویزارد BSB برای ایجاد این پروسسور تعبیه شده استفاده کنیم. ما Yesرا انتخاب می کنیم.



۲- مرحله دوم

در این مرحله از آزمایش قصد داریم تا تنظیمات واحد UART را انجام دهیم. صفحه ویزارد (BSB) Base system Builder بصورت زیر باز میشود که ابزارهای لازم برای ساخت یک پردازنده تعبیه شده رادر اختیار ما قرار میدهد.

ISE Project Navigator (O.40d) - D:\Temp\M ∑ File Edit View Project Source P D P ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	Base System Builder	(1			8 ×
Design ↔ □	r (O.40d) - D.YtempVM Base System Builder Hurring Hur				
Hierarchy MicroBlaze_HUT C MicroBlaze_HUT C X:3400-4pq208 C MicroBlaze_HUT C M	Welcome to the Base System Bui This tool leads you through the steps n Select One of the Following: I would like to create a new desig I would like to load an existing .bs	ider lecessary for creating an embedd n sb settings file (saved from a prev	led system. rious session)		Browse
No Processes Running Processes: Embedded_HUT Design Summary/Reports Design Utilities Manage Processor Design (XPS Generate Top HDL Source Export Hardware Design To SDI					

شکل ۸۳- پنجره Wizard 1 نرم افزار XPS

در اولین صفحه تنظیمات، به نرم افزار می گوییم که ما در حال ایجاد یک سیستم برای یک برد سفارشی با تراشه XC3S400PQ208-4 هستیم. همچنین ریست کلی سیستم را میتوانیم Active High یا Active Low انتخاب کنیم که ما در اینجا حالت پیش فرض آن را انتخاب کردهایم.

	Board	System	Processor	Peripheral	Cache	Summ
Board Selection						
Select a target develop	ment board.					
Board						
spartan3 xc3s400 pq20 Please use the custom b I would like to creat Board Information Architecture spartan3	8 -4 poard option. te a system for a custom b Device vc3s400	oard	Package pq208	¥	Speed Grade	
Use Stepping						
Reset Polarity Active L	ow					

در گام بعدی تک پروسسوری یا دو پروسسوری بودن سیستم را انتخاب می کنیم که ما بنا بر شرایط ساده مساله همان تک پروسسوری را انتخاب می کنیم و Next را میزنیم.

Welcome	Board	System	Processor	Peripheral	Cache	Sum
System Configuration Configure your system.						
@ 5	Single-Processor Sys	stem		O Dual-Process	or System	
Select this option to creat Wizard will let you config some major configuration	te a design with a si are the processor, the parameters for the	ngle processor. This he peripheral set and peripherals.	Select this opt will let you con accessible to t two processor	ion to create a design w figure the types of the he two processors and t s.	ith two processors, processors, the peri the peripherals share	This Wizard pherals ed by the
Processor 1		1	Proces	Isor 1 Process	GPIO	
	Processor 1 F	Peripherals		Shared Mailbo	Peripherals Mutex	
	RS232 GI	PIO		Process	tor 2 Perioherals	
			Proces	sor 2 DDR	EMAC	



در گام بعدی مطابق شکل زیر، فرکانس کلاک مرجع پروسسور را 50MHz، نوع پروسسور را میکروبلیز، فرکانس کلاک سیستم را 66.67MHz و مقدار حافظه محلی را 8KB انتخاب میکنیم که همین مقدار حافظه برای این پروژه کافیست. در آخر هم میتوانیم Floating Point را فعال کنیم اگرچه نیازی به آن نداریم.

Welcome	Board	System	Processor	Peripheral	Cache	Sum
Processor Configuratio	'n					
Configure the processor(s)).					
	0					
Reference Oock Frequence	sy (50.00					
Processor 1 Configuration						
Processor Type	MicroBlaze					0
System Clock Frequency	66.67					* M
Local Memory	8 ×25					
Debug Interface	a ka					
	10.000					

در این مرحله، مطابق شکل زیر می توانیم امکانات جانبی میکرو خود را انتخاب کنیم که ما در این پروژه ، UARTرا به میکرو اضافه می کنیم. در پروژه دیگر می توانید اترنت، حافظهها، SPI و ... را به میکروپروسسور خود اضافه نمایید.

Welcome	Board	System	Processor	Peripheral	Cache	Summ
Peripheral Configuration To add a peripheral, drag it f	n from the "Available F	eripherals" to the p	rocessor peripheral list.	To change a core paramet	er, click on the perip	bheral.
Available Peripherals Add Device Expor	t Import.		Processor 1 (Micro	Blaze) Peripherals		Select All
Desigh and Names			Core		Paramete	r
Peripheral Names			dimb cntir			
IO Devices			Core		Imb_bram	n_if_cntlr
Imb_bram_if_cnt	Ir		ilmb_cntlr			
xps_bram_if_cntl	r		Core: Imb_b	oram_if_cntlr		
xps_timebase_wo	lt					
xps_timer						
Add IO Devices for Gene	eric Board		— ×			
Select on IO device or extend	and an an an a three in	an unun deuelennen				
board.	rhai memory that is	on your developme	nt			
IO Interface Type						
EMC			-			
EMC						
ETHERNET						
GPIO						
MEMORY						
PCI32	/					
PCI_ARBITER						
SYSACE						
TEMAC						
UART -						
	S.	$\nabla \mathbf{PS} = [\mathbf{h} \cdot \mathbf{I} - \mathbf{N}]$	Vizard 1 . ~ .	NV 15 m		
	1	۷ درم افرار ۱۵	/- پنجره + Izaiu	سكل ٢٠		
			11			

حال در پنجره بعدی باید مشخصات UART را تعیین کنیم که ما در اینجا Baud Rate را 9600 و تعداد بیت داده را همان 8 وبدون Parity در نظر می گیریم.

Peripheral Configuration b add a peripheral, drag it from the "Available Peripherals" to the processor peripheral list. To change a core parameter, click on the peripheral. Available Peripherals Add Device Export Import Processor 1 (MicroBlaze) Peripherals Peripheral Names Processor 1 (MicroBlaze) Peripherals Internal Peripherals Core Imb_bram_if_cntlr Psc232 xps_timebase_wdt 8 xps_timer 9600 Data Bits 8 Parity None Use Interrupt Imb_cntlr Core: Imb_bram_if_cntlr Core: Imb_bram_if_cntlr xps_timedase_wdt Add >	Welcome	Board	System	Processor	Peripheral	Cache	Summa
Add Device Export Import Processor 1 (MicroBlaze) Peripherals Select Al Peripheral Names IO Devices Parameter RS232 xps_uartlite Internal Peripherals RS232 xps_uartlite Baud Rate 9600 Import Data Bits 8 Parity None Vise Interrupt Import Use Interrupt Import Add > Add > Famoura Famoura	eripheral Configuration o add a peripheral, drag it from	the "Available Per	ripherals" to the pr	rocessor peripheral list. 1	ο change a core paramet	er, click on the periphe	ral.
Peripheral Names Core Parameter ID Devices RS232 xps_uartlite Internal Peripherals Baud Rate 9600 Imb_bram_if_cntlr ys_timebase_wdt Baud Rate 9600 Data Bits 8 Parity None Use Interrupt Interrupt Interrupt dImb_cntlr Core: Imb_bram_if_cntlr Core: Imb_bram_if_cntlr	Add Device Export	Import		Processor 1 (MicroE	laze) Peripherals		Select All
IO Devices RS232 Internal Peripherals Imb_bram_if_cntlr xps_bram_if_cntlr 9600 xps_timebase_wdt 9600 xps_timer 8 Parity None Use Interrupt Imb_cntlr Core: Imb_bram_if_cntlr Core: Imb_bram_if_cntlr Add > Parnue	Peripheral Names			Core		Parameter	
Internal Peripherals Imb_bram_if_cntlr Imb_bram_if_cntlr yss_bram_if_cntlr xps_timebase_wdt 9600 xps_timer Data Bits Parity None Use Interrupt Imb_cntlr Core: Imb_bram_if_cntlr Core: Imb_bram_if_cntlr Add > Parous	IO Devices			RS232			
Imb_bram_if_cntlr perity xps_time Parity None Use Interrupt Imb_cntlr Core: Imb_bram_if_cntlr Core: Imb_bram_if_cntlr Core: Imb_bram_if_cntlr	- Internal Peripherals			RS232		xps_uartlite	-
Aps_brain_in_chui xps_timebase_wdt xps_timer Data Bits Parity Use Interrupt dImb_cntlr Core: Imb_bram_if_cntlr iImb_cntlr Core: Imb_bram_if_cntlr	Imb_bram_if_cntlr			Baud Rate		9600	
xps_timer Parity None Use Interrupt Imb_cntlr Core: Imb_bram_if_cntlr Imb_cntlr Core: Imb_bram_if_cntlr Core: Imb_bram_if_cntlr	xps_bran_ii_cnur			Data Bits		8	
Use Interrupt dlmb_cntlr Core: Imb_bram_if_cntlr iImb_cntlr Core: Imb_bram_if_cntlr Core: Imb_bram_if_cntlr	xps_timer			Parity		None	-
Add >				Use Interrupt			
Add >				dimb_cntir	am if onth		
Add >				ilmb_cntlr	ann_n_critit		
Add >				Core: Imb_br	am_if_cntlr		
- Bampura			Add >				
			- Demove				

شکل ۸۸- پنجره Wizard 5 نرم افزار XPS

خلاصه نتایج عملیات بالا در شکل زیر نشان داده شده است. به عنوان مثال Block RAMها از آدرس صفر تا 1FFF قرار دارند که همان 8KB یا به عبارتی 2 به توان 13بایت میباشد.

	Board	System	Processor	Peripheral	Cache	Summ
Summary Below is the summary of the	e system you are cr	eating.				
ystem Summary						
Core Name	Instance Name	Base Address	High Address			
Processor 1 xps_uatlite Imb_bram_if_cntlr Imb_bram_if_cntlr Imb_bram_if_cntlr	microblaze_0 R5232 r dlmb_cntlr r ilmb_cntlr	0x84000000 0x00000000 0x00000000	0x8400FFFF 0x00001FFF 0x00001FFF			
ile Location	aze_HUT\Embedd aze_HUT\Embedd aze_HUT\Embedd	ed_HUT\Embedd ed_HUT\Embedd ed_HUT\data\Em	led_HUT.xmp led_HUT.mhs ibedded_HUT.ucf			
ile Location Deverall D:\Temp\MicroBla D:\Temp\MicroBla D:\Temp\MicroBla D:\Temp\MicroBla D:\Temp\MicroBla D:\Temp\MicroBla	aze_HUT\Embedd aze_HUT\Embedd aze_HUT\Embedd aze_HUT\Embedd aze_HUT\Embedd aze_HUT\Embedd	ed_HUT\Embedd ed_HUT\Embedd ed_HUT\data\Em ed_HUT\etc\fast ed_HUT\etc\bitg	led_HUT.xmp led_HUT.mhs bedded_HUT.ucf _runtime.opt rnload.cmd en.ut			
ile Location - Overall - D:\Temp\MicroBla - D:\Temp\MicroBla - D:\Temp\MicroBla - D:\Temp\MicroBla - D:\Temp\MicroBla	aze_HUT\Embedd aze_HUT\Embedd aze_HUT\Embedd aze_HUT\Embedd aze_HUT\Embedd aze_HUT\Embedd	ed_HUT\Embedd ed_HUT\Embedd ed_HUT\Embedd ed_HUT\data\Em ed_HUT\etc\fast_ ed_HUT\etc\bitg XPS نرم افزار	led_HUT.xmp led_HUT.mhs ibedded_HUT.ucf _runtime.opt nload.cmd en.ut Wizard 6 بنجره -/	شکل ۱۹		

در پنجره بعدی برای ساخت این پروسسور از ما میخواهد که چک کنیم فایل ucf موجود باشد، و از شماره پایه JTAG باشیم که ما تایید میکنیم.



شکل ۹۰- تایید فایل ucf

در نتیجه، فایلهای پروژه ساخت پروسسور، رابط های BUS، پورت ها و آدرسها نمایش داده می شود.



شکل ۹۱- پنجره فایلهای پروژه

در پایان این مرحله نرم افزار XPS را می بندیم.

۳- مرحله سوم

در این مرحله از آزمایش در نرم افزار ISE مطابق شکل زیر بر روی Generate Top HDL Source دو بار کلیک میکنیم تا فایل HDL میکروبلیز ساخته شود.

Design ↔ □ ♂ ×	03	Design Overview			En	bedded HUT	Project Status		
View: Vi	0	- D IOB Properties	Project File:	Mic	croBlaze_HUT.xise		Parser Errors:		N
Hierarchy		- D Module Level Utilization	Module Name:	Em	bedded_HUT		Implementati	on State:	N
MicroBlaze_HUT		Iming Constraints Pinout Report	Target Device:	xc	3s400-4pq208		• Errors:	8	
Embedded_HUT (Embedded_HU	S	Clock Report	Product Version:	ISE	E 13.1		• Warnir	igs:	
	(inter	- Can Static Timing	Design Goal:	Ba	lanced		• Routin	g Results:	
	38	Platgen Messages	Design Strategy:	Xili	nx Default (unlock	ed)	• Timing	Constraints:	1
(2)	100	- D Libgen Messages	Environment:				• Final T	iming Score:	
		Simgen Messages							
		XPS Reports				XPS Reports			
		- Platgen Log File	Report Name	Generat	bed	A S REPORTS	Frrors	Warnings	Infos
* III +	ŧ.	Simgen Log File	Plateen Log File						
Running: Generate Top HDL Source	П	- D BitInit Log File	Libgen Log File						-
Processes: Embedded_HUT		System Log File	Simpen Log File						-
Design Summary/Reports	17	Parser Messages	BitInit Log File						-
Design Utilities	V	Design Properties	System Log File	2014 18:	ىيە مارىر. 26 50:48	جفاد ش	-		-
- Compare Processor Design (APS)	-	Ontional Design Summary Contents							
Export Hardware Design To SDK		- Show Clock Report	-				-50		
		Show Failing Constraints			D	etailed Repor	ts		
		Show Warnings	Report Name		Status	Generated	Errors	Warnings	Inf
		Show Errors	Synthesis Report						
			Translation Report						
			Map Report						
Start 🕫 Design 🚺 Files 🌓 Libraries	Σ	Design Summary	1						

شکل ۹۲- گزینه Generate Top HDL Source

سپس از دو گزینه شکل زیر بر روی یکی دو بار کلیک میکنیم تا سخت افزار طراحی شده به نرم افزار SDK منتقل شده و منتقل شده و منتقل شود. تفاوت این دو گزینه در این است که در یکی فایل بیتی طرح نیز به نرم افزار SDK منتقل شده و در دیگری منتقل نمی شود و در پایان پروژه فایل بیتی ساخته می شود.

			-	-					
	File	Edit	View	Project	Source	Process	Tools	Window	Lay
1	1 🦻		SP 22	1 × 10		10 Cil	>>	PPP	0 20
Desi	gn							+ 🗆 🗗 🗡	3
	View	: 🕥 🗄	Impler	mentation	Sir	nulation			► Ξ
5	Hier	archy							
6	1		icroBlaze 3s400-4p	≥_HUT					N2
00		- "H	Ent Emt	bedded_H	UT_top - S	STRUCTUR	E (Embe	dded_HU1	
6			Em	bedded_H	HUT_i - Em	bedded_H	UT (Emb	edded_H	12 ·····
(23)									12
									24
									20
									20
	-				111			1	0
	62	No Pr	ocesses R	lunning				Y	0
TT.	Pro	esses	Embeda	ded_HUT_	i - Embed	ded_HUT		/	
要红	(t)	20	Design	Utilities					
	1	20	Manage	Processo	Design (XPS)	_	-	
		2	Export F	Hardware	Design To	SDK witho	ut Bitstre	am	
	L	65	Export H	lardware	Design To	SDK with E	Bitstream	1	
							~		
								12	
Incert	Chan			172		1 the sector of			
	Star		Design			Libraries			
Erro									

شکل ۹۳- گزینه انتقال طرح به نرم افزار SDK

با این کار نرم افزار SDK به طور اتوماتیک توسط ISE فراخوانی شده و در اولین گام آدرس فضای کاری را از ما سوال می کند. گزینه OK را انتخاب می کنیم.



شکل ۹۴- پنجره فراخوانی نرمافزار SDK توسط ISE

محيط كار SDK بصورت زير باز مى شود كه تا ساخته شدن كتابخانه ها صبر مى كنيم.



شکل ۹۵- پنجره محیط کار نرمافزار SDK

مطابق شکل زیر، بر روی Xilinx Board Support Package کلیک می کنیم تا بتوانیم کتابخانهها، پلت فرمهای سخت افزاری دلخواه و سیستم عامل مورد نیاز را انتخاب کنیم.

C/C++ - Xilinx SDK	-			
ile Edit Source Refactor Navigate	Search Run Proj	ect Xilinx Tools W	indow Help	
	🏥 🔀 🕺 🗃 🗧	• 🖻 • 🞯 •	≪ ▼ 🛞 ▼	😭 📴 C/C++
Xilinx C Project		• * - • -		
Xilinx C++ Project				🗄 Out 🛛 🛞 Ma 📄 🗖
Xilinx Hardware Platform Specifica	tion			An outline is not available.
Xilinx Board Support Package				
Project				
Source Folder				
😂 Folder				
C Source File				
Header File				
File from Template				
G Class				
Cther	Ctrl+N			
	💽 Problems 📢	🔄 Tasks 🖳 Console		oerties 🖉 Terminal 🛛 🗖 🗖
	C-Build [standal	one_bsp_0]	₽ <u>↑</u> 🔄	
	"Compiling	uartlite"		
	"Compiling	bram"		
	"Compiling	cpu"		
	Running exe	cs_generate.	ies'	
		·		-
Xil	inx Board Suppo	نزينه ort Package	شکل ۹۶- گ	
	تاثيب حجر	ها با	15)	بحرميان شدم تنمل التيافي
مان است) را التخاب و سیستم	تراشه بورد خود	، همان مربوط به	ا موجود (که	لجره بار شده لتها پلك قرم
			1 11	
نات پایه پردازنده را فراهم می	دسترسی به امد	طح پایین است و	م افزاری سه	Standalof (که یک لایه نر
				_
				ب می گنیم.
New Board Support Package P	oject			
linx Board Support Pack	age Project			n
reate a Board Support Package.	2			
roject name: standalone_bsp_	1			
I lise default location				
acation: Cilleard SDC)waster	ace\standalons	en 1		Province
C:\USers\SPC\WOrksp	ace\standalohe_b	shTr		Browse
Choose file system: d	efault 👻			

شکل ۹۲- پنجره New Board Support Package

Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.

Finish

Cancel

Target Hardware

xilkernel

?

Hardware Platform: hw_platform_0 CPU: hw_platform_0

Board Support Package OS

نوع OS و نسخه آن در پنجره زیر قابل تغییر است. همچنین میتوانیم هر یک از کتابخانه های موجود (مانند TCP/IP و...)را به پروژه اضافه کنیم که برای این پروژه نیازی به این کتابخانهها نیست.

ontrol various setti	ackage Settings	ort Package.	
Overview standalone drivers cpu	standalone_bsp_1 OS Type: stando OS Version: 3.01.4 Target Hardware Hardware Specifica Processor:	alone	Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.
	Supported Librarie Check the box ne	s dt to the libraries voi	u want included in your Board Support Package. You can configure the library in the navigator on the left.
	Supported Librarie Check the box new Name	s et to the libraries you Version	u want included in your Board Support Package. You can configure the library in the navigator on the left.
	Supported Librarie Check the box new Name wip130	s et to the libraries you Version 3.00.a	u want included in your Board Support Package. You can configure the library in the navigator on the left. Description IwIP TCP/IP Stack library: IwIP v1.3.0, Xilinx adapter v3.00.a
	Supported Librarie Check the box new Name wip130 wiffatfs	s kt to the libraries you Version 3.00.a 1.00.a	u want included in your Board Support Package.You can configure the library in the navigator on the left. Description MuP TCP/IP Stack library: IwIP v1.3.0, Xilinx adapter v3.00.a Provides read/write routines to access files stored on a FAT16/32 file system. Requires Syste
	Supported Librarie Check the box new Name wip130 difafs kilflash	s t to the libraries you Version 3.00.a 1.00.a 2.02.a	u want included in your Board Support Package.You can configure the library in the navigator on the left. Description MuP TCP/IP Stack library: IwIP v1.3.0, Xilinx adapter v3.00.a Provides read/write routines to access files stored on a FAT16/32 file system. Requires Syste Xilinx Flash library for Intel/AMD CFI compliant parallel flash
	Supported Librarie Check the box new Name wip130 diffafs diffash dilfash	s Version 3.00.a 1.00.a 2.02.a 2.01.a	u want included in your Board Support Package.You can configure the library in the navigator on the left. Description IwiP TCP/IP Stack library: IwIP v1.3.0, Xilinx adapter v3.00.a Provides read/write routines to access files stored on a FAT16/32 file system. Requires Syste Xilinx Flash library for Intel/AMD CFI compliant parallel flash Xilinx In-system and Serial Flash Library
	Supported Librarie Check the box new Name wip130 difaffs difaffs difash difaf	s t to the libraries you 3.00.a 1.00.a 2.02.a 2.01.a 1.00.a	u want included in your Board Support Package.You can configure the library in the navigator on the left. Description Image: Image

شکل ۵۸- پنجره Board Support Package Setting

حال که کتابخانههای مورد نظر را انتخاب کردیم، در گام بعدی از قسمت نشان داده شده در شکل زیر گزینه Xilinx C Project را انتخاب می کنیم (البته می توانیم ++C را هم انتخاب کنیم).



شکل ۹۹- گزینه ایجاد فایل C

از آنجا که هدف، ارسال و دریافت یک پیغام ساده از طریق UART است بنابراین یک قالب ساده Hello World از آنجا که هدف، ارسال و دریافت یک پیغام ساده از طریق UART است بنابراین یک قالب ساده (را انتخاب می کنیم. در پروژههای بعدی خود می توانید برنامه تست حافظه (Memory Tests) و یا ... را انتخاب کنید.

l ew Xilinx C Proje Create a managed ma	ect ake application project. Choose from one of the sample applications.
Project name: hello	_world_1
Use default locati	ion
Location: C:\Users\!	SPC\workspace\hello_world_1 Browse
Choose file	e system: default 👻
Target Hardware	
Hardware Platform:	hw_platform_0
Processor:	microblaze_0
Select Project Temp	plate
Dhrystone	Description
Empty Application	Let's say 'Hello World' in C.
Meno Wond IwiP Echo Server Memory Tests Peripheral Tests SREC Bootloader Xilkernel POSIX The	reads Demo
	ی توانید هر کدام از این قالب
	های آماده را انتخاب کنید

شکل ۱۰۰- انتخاب Template برای Uart

بعد از انتخاب گزینه Finish پنجره زیر باز می شود که نرم افزار، سیستم را طراحی می کند. همانطور که می بینید از قسمت نشان داده شده فایل برنامه C را باز کنید. این برنامه یک پیغام ساده را بر روی پورت متصل به خود ارسال می کند.



شكل ۱۰۱- پنجره فايل C پروژه Uart

خروجیهای مورد انتظار (برای تهیه گزارش کار)

- ارائه مفاهیم تکمیلی در رابطه با این آزمایش
- بررسی صحت عملکرد برنامه C با استفاده از یک نوع نرم افزار ترمینال
 - مشکلاتی که در حین کار با نرم افزار SDK مواجه شدید.

تمرين

با استفاده از پردازنده MicroBlaze پروتکل SPI را راهاندازی کنید.

آزمایش ۱۳

طراحی و پیادهسازی پورت VGA برد DE2_115

هدف

- کاربرد پردازش تصویر در علوم مهندسی
 - آشنایی با پورت VGA
- بررسی نحوه طراحی و پیادهسازی VGA

شرح آزمایش

بدون شک یکی از مهمترین و پرکاربردترین موضوعات موجود در علوم مهندسی، علوم پایه و پزشکی، پردازش تصاویر دیجیتال است. در حال حاضر پردازش تصویر به یک موضوع تحقیقاتی در بین دانشجویان چه در داخل و چه در خارج از کشور تبدیل شده است. ابزار همواره به عنوان یکی از فاکتورهای مهم در پردازش تصویر تلقی میشود. کامپیوترها، بردهای مبتنی بر سیستم عامل (برد رزپریپای Raspberry pi) از متداول ترین ابزارهای موجود در زمینه پردازش تصویر هستند. FPGA به خاطر قدرت پردازشی بسیار بالا، همواره به عنوان یک تراشه قدرتمند در حوزه پردازش تصویر مطرح بوده است.

امروزه، ربات ناوبری در کنار عدم برخود با مانع بر مبنای سیستم بینایی به یک چالش اساسی و مهم در صنعت رباتیک تبدیل شده است. برای حل این مسئله، میبایست سرعت دریافت، پردازش و تصمیم گیری ربات بستر پویا را افزایش داد، لذا در طی چند سال گذشته تحقیقات بسیاری در این زمینه با هدف افزایش قدرت نرمافزاری و سخت افزاری ربات بستر پویا با استفاده از تکنولوژیهای پیشرفته صورت گرفته است. هدف اصلی این سیستم، طراحی، ساخت و پیادهسازی یک هسته پردازش تصویر قدرتمند بر روی FPGA با رویکرد افزایش سرعت دریافت و استخراج اطلاعات مورد نیاز برای سیستم بینایی ربات بستر پویای ماویکرد افزایش و سخت افزاری ربات بستر پویا با استفاده از تکنولوژیهای پیشرفته صورت گرفته است. هدف اصلی این سیستم، طراحی، ساخت و پیادهسازی یک هسته پردازش تصویر قدرتمند بر روی NTACO با رویکرد افزایش سرعت دریافت و استخراج اطلاعات مورد نیاز برای سیستم بینایی ربات بستر پویای NTACO به صورت زمان واقعی و با قابلیت ایجاد یک معماری انحصاری می اشد. ویژگیهایی همچون پیادهسازی سخت افزاری، انعطاف پذیری، معماری انحصاری، سرعت وزن پایین به همراه عدم نویزپذیری، این امکان را را برای این میکاد و زن پایین به همراه عدم نویزپذیری، این امکان را برای این سیستم فرای این سیستم فرای به پردازش تصویر قدر پرای سیان پی رات بستر پویای NTACO به صورت زمان واقعی و با قابلیت ایجاد یک معماری انحصاری می اشد. ویژگیهایی همچون پیادهسازی سختافرازی، انعطاف پذیری، معماری انحصاری، سرعت پردازش بالا در کنار قیمت و وز پایین به همراه عدم نویزپذیری، این امکان را برای این سیستم فراهم می کند تا از آن در سیستمهای مکاترونیکی که نیاز به پردازش تصویر زمان واقعی در این استفاده شود. یکی از مزیتهای اصلی در این هسته پردازش تصویر ایجاد پلی بین الگوریتمهای دارند، نیز استفاده شود. یکی از مزیتهای اصلی در این هستم بران می به سراه می ای به می از می و وانی پاین ند.
پردازش تصویر و سختافزار است تا بتوان از آن به صورت کاربردی و مستقیم در صنایع مختلف ، پروژه های تحقیقاتی و پایان نامه های دانشگاهی استفاده نمود.

The DE2-115 board includes a 15-pin D-SUB connector for VGA output. The VGA synchronization signals are provided directly from the Cyclone IV E FPGA, and the Analog Devices ADV7123 triple 10-bit high-speed video DAC (only the higher 8-bits are used) is used to produce the analog data signals (red, green, and blue). It could support the SXGA standard (1280*1024) with a bandwidth of 100MHz. Figure 4-21 gives the associated schematic.



Figure 4-21 Connections between FPGA and VGA DE2 شکل ۱۰۲- نحوه اتصال یورت VGA به برد

The timing specification for VGA synchronization and RGB (red, green, blue) data can be found on various educational website (for example, search for "VGA signal timing"). **Figure 4-22** illustrates the basic timing requirements for each row (horizontal) that is displayed on a VGA monitor. An active-low pulse of specific duration (time (a) in the figure) is applied to the horizontal synchronization (hsync) input of the monitor, which signifies the end of one row of data and the start of the next. The data (RGB) output to the monitor must be off (driven to 0 V) for a time period called the back porch (b) after the hsync pulse occurs, which is followed by the display interval (c). During the data display interval the RGB data drives each pixel in turn across the row being displayed. Finally, there is a time period called the front porch (d) where the RGB signals must again be off before the next hsync pulse can occur. The timing of the vertical synchronization (vsync) is the similar as shown in **Figure 4-22**, except that a vsync pulse signifies the end of one frame and the start of the next, and the data refers to the set of rows in the frame (horizontal timing). **Table 4-14** and **Table 4-15** show different resolutions and durations of time periods a, b, c, and d for both horizontal and vertical timing.

Detailed information for using the ADV7123 video DAC is available in its datasheet, which can be found on the manufacturer's website, or in the DE2_115_datasheets\VIDEO-DAC folder on the DE2-115 System CD. The pin assignments between the Cyclone IV E FPGA and the ADV7123 are listed in **Table 4-16**. An example of code that drives a VGA display is described in Sections 6.2 and 6.3.



Figure 4-22 VGA horizontal timing specification

شکل ۱۰۳- دیاگرام زمانبندی پورت VGA

	Table 4-14 VGA Horizontal Timing Specification										
VGA mode	VGA mode			Horizontal Timing Spec							
Configuration	Resolution(HxV)	a(us)	b(us)	c(us)	d(us)	Pixel clock(MHz)					
VGA(60Hz)	640x480	3.8	1.9	25.4	0.6	25					
VGA(85Hz)	640x480	1.6	2.2	17.8	1.6	36					
SVGA(60Hz)	800x600	3.2	2.2	20	1	40					
SVGA(75Hz)	800x600	1.6	3.2	16.2	0.3	49					
SVGA(85Hz)	800x600	1.1	2.7	14.2	0.6	56					
XGA(60Hz)	1024x768	2.1	2.5	15.8	0.4	65					
XGA(70Hz)	1024x768	1.8	1.9	13.7	0.3	75					
XGA(85Hz)	1024x768	1.0	2.2	10.8	0.5	95					
1280x1024(60Hz)	1280x1024	1.0	2.3	11.9	0.4	108					

جدول ۴- مشخصات زمان بندی پورت VGA

VGA mode	Vertical Timing Spec						
Configuration	Resolution(HxV)		a(lines) b(lines)		d(lines)	Pixel clock(MHz)	
VGA(60Hz)	640x480	2	33	480	10	25	
VGA(85Hz)	640x480	3	25	480	1	36	
SVGA(60Hz)	800x600	4	23	600	1	40	
SVGA(75Hz)	800x600	3	21	600	1	49	
SVGA(85Hz)	800x600	3	27	600	1	56	
XGA(60Hz)	1024x768	6	29	768	3	65	
XGA(70Hz)	1024x768	6	29	768	3	75	
XGA(85Hz)	1024x768	3	36	768	1	95	
1280x1024(60Hz)	1280x1024	3	38	1024	1	108	

 Table 4-15
 VGA Vertical Timing Specification

Table 4-16Pin Assignments for ADV7123

Signal Name	FPGA Pin No.	Description	I/O Standard
VGA_R[0]	PIN_E12	VGA Red[0]	3.3V
VGA_R[1]	PIN_E11	VGA Red[1]	3.3V
VGA_R[2]	PIN_D10	VGA Red[2]	3.3V
VGA_R[3]	PIN_F12	VGA Red[3]	3.3V
VGA_R[4]	PIN_G10	VGA Red[4]	3.3V
VGA_R[5]	PIN_J12	VGA Red[5]	3.3V
VGA_R[6]	PIN_H8	VGA Red[6]	3.3V
VGA_R[7]	PIN_H10	VGA Red[7]	3.3V
VGA_G[0]	PIN_G8	VGA Green[0]	3.3V
VGA_G[1]	PIN_G11	VGA Green[1]	3.3V
VGA_G[2]	PIN_F8	VGA Green[2]	3.3V
VGA_G[3]	PIN_H12	VGA Green[3]	3.3V
VGA_G[4]	PIN_C8	VGA Green[4]	3.3V
VGA_G[5]	PIN_B8	VGA Green[5]	3.3V
VGA_G[6]	PIN_F10	VGA Green[6]	3.3V
VGA_G[7]	PIN_C9	VGA Green[7]	3.3V
VGA_B[0]	PIN_B10	VGA Blue[0]	3.3V
VGA_B[1]	PIN_A10	VGA Blue[1]	3.3V
VGA_B[2]	PIN_C11	VGA Blue[2]	3.3V
VGA_B[3]	PIN_B11	VGA Blue[3]	3.3V
VGA_B[4]	PIN_A11	VGA Blue[4]	3.3V
VGA_B[5]	PIN_C12	VGA Blue[5]	3.3V
VGA_B[6]	PIN_D11	VGA Blue[6]	3.3V
VGA_B[7]	PIN_D12	VGA Blue[7]	3.3V
VGA_CLK	PIN_A12	VGA Clock	3.3V
VGA_BLANK_N	PIN_F11	VGA BLANK	3.3V
VGA_HS	PIN_G13	VGA H_SYNC	3.3V
VGA_VS	PIN_C13	VGA V_SYNC	3.3V
VGA_SYNC_N	PIN_C10	VGA SYNC	3.3V

جدول ۷- تعریف پایههای پورت VGA

۱- مرحله اول

در این مرحله از آزمایش به بررسی آیسی ADV7123 که رابطی بین FPGA و VGA میباشد میپردازیم. **نکته:** برای بررسی این آی سی به دیتاشیت آن مراجعه نمایید.

۲- مرحله دوم

در این مرحله از آزمایش مطابق کدی که در اختیارتان قرار داده می شود نحوه اتصال VGA به FPGA را طراحی کنید.

خروجیهای مورد انتظار (برای تهیه گزارش کار)

ارائه مفاهیم تکمیلی در رابطه با این آزمایش

- بررسی صحت عملکرد طرح با استفاده از شبیه سازی آن
- بررسی صحت عملکرد طرح با پروگرم کردن برنامه بر روی برد
- مشکلاتی که در حین برنامهنویسی، شبیهسازی و پروگرم کردن با آنها مواجه شدید.

تمرين

با استفاده از طرح خود عبارت "Sadjad University of Technology" را بر روی مانیتور کامپیوتر نمایش دهید.

آزمایش ۱۴

طراحی و پیادهسازی مبدلهای DAC و آشنایی با نحوه کارکرد آنها

هدف

- آشنایی با مبدل DCA5744
- نحوه راه اندازی و پیاده سازی مبدل DCA5744
- بررسی نحوه ارتباط مبدلهای DAC با FPGA از طریق پروتکل SPI

شرح آزمایش

در بسیاری از پروسههای کنترلی جهت بستن حلقه کنترل و برای صدور فرامین آنالوگ جهت محرکها و کنترلرهای مختلف در خارج از سیستم جمع آوری اطلاعات و کنترل ، به فرامین آنالوگ به صورت ولتاژ و یا جریان نیاز میباشد. در سیستمهای دیجیتال امروزی این کار توسط مبدلهای دیجیتال به آنالوگ DAC انجام میشود و همانند مبدلهای ADC دارای دقت، سرعت و مشخصات متفاوتی میباشند. وقتی توسط اسکنر از یک تصویر اسکن میگیریم،کاری که اسکنر انجام میدهد یک تبدیل آنالوگ به دیجیتال است: اسکنر را ضبط میکنیم یا از یک تجزیه کنده ولای دول را به اطلاعات دیجیتال تدیل میکند. وقتی صدایمان را ضبط میکنیم یا از یک تجزیه کننده VOIP روی کامپیوتر استفاده میکنیم، از یک مبدل آنالوگ به دیجیتال برای تبدیل صدایمان استفاده میکنیم. وقتی یک سیدی صوتی در استودیو ضبط میشود باز مهم تبدیل آنالوگ به دیجیتال اتفاق میافتد، نتیجهی تبدیل صدا به اعداد دیجیتال بر روی دیسک ذخیره میشود. هر موقع که سیگنال آنالوگ دوباره مورد نیاز باشد، مبدل دیجیتال به آنالوگ مود نیاز است. وقتی یک می دی صوتی را اجرا میکنیم کاری که سی دی خوان انجام میدهد خواندن اطلاعات دیجیتال بر روی دیسک ذخیره می تبدیل آنالوگ به دیجیتال اتفاق میافتد، نتیجهی تبدیل صدا به اعداد دیجیتال بر روی دیسک ذخیره می میشود. هر موقع که سیگنال آنالوگ دوباره مورد نیاز باشد، مبدل دیجیتال به آنالوگ مورد نیاز است. وقتی یک روی دیسک و تبدیل دوبارهی آنها به سیگنال آنالوگ است، به این ترتیب ما می توانیم موسیقی را بشنویم. اما چرا دیجیتال؟ زنهاست. از آنجایی که سیگنال های آنالوگ میتوانند هر مقداری فرض شوند، نویز بخشی

نویر و اعتساس یکی از انهاست. از انجایی که سیکنالهای انانوک می واند هر مقداری فرض سوند، نویز بخسی از سیگنال اصلی تفسیر می شود. در طرف دیگر، سیستمهای دیجیتال، فقط می توانند دو عدد را درک کنند، صفر و یک. هر چیز دیگر غیر از این دو حذف میشود. به همین خاطر وقتی یک سی دی صوتی را گوش می کنیم هیچ اغتشاش مزاحمی را نمی شنویم، حتی اگر قبلا هزاران مرتبه آنرا اجرا کرده باشیم. برتری دیگر سیستمهای دیجیتال در برابر آنالوگ تواناییشان در فشردهسازی اطلاعات است. از آنجایی که دیجیتال شده ی یک سیگنال آنالوگ تنها دسته ای از اعدادند، این اعداد می توانند فشرده و متراکم شوند. فشرده سازی می تواند جهت صرفه جویی در فضای ذخیره سازی یا پهنای باند بکار رود. به منظور پوشش دادن مجموعه نسبتا وسیعی از نیازهای سیستمهای صنعتی، مبدل های متنوع DAC طراحی و ساخته شده است که در این آزمایش به بررسی و نحوه راهاندازی یک نوع از این مبدل ها می پردازیم. در این مرحله از آزمایش قصد داریم با مبدل دیجیتال به آنالوگ AD5744 آشنا شویم. در این مرحله از آزمایش قصد داریم با مبدل دیجیتال به آنالوگ AD5744 آشنا شویم. می ملکوجی این مبدل بین VIL تا 2004 است.

AD5744 قابلیت فراهم کردن تقویت کننده خروجی یکپارچه، بافر مرجع، کنترل جریان توان بالا و پایین اختصاصی را دارد. همچنین دارای پورت های I/O دیجیتال، برنامهریز از طریق رابط سریال و یک سنسور دمای آنالوگ است.

اگر بخواهیم به برخی از ویژگیهای این مبدل بپردازیم میتوانیم به موارد زیر اشاره کنیم: ۱-توانایی تبدیل ۱۴ بیت دیجیتال به آنالوگ ۲-محدوده خروجی قابل برنامهریزی 10۷± ۳-زمان تنظیم حداکثر ۱۰میکروثانیه ۴-پینهای کنترلی خروجی منطقی ۵-محدوده دمای بین ۴۰- سانتی گراد تا ۸۵+ سانتی گراد

GENERAL DESCRIPTION

The AD5744R is a quad, 14-bit, serial input, bipolar voltage output DAC that operates from supply voltages of ±11.4 V to ±16.5 V. Nominal full-scale output range is ±10 V. The AD5744R provides integrated output amplifiers, reference buffers, and proprietary power-up/power-down control circuitry. The part also features a digital I/O port, programmed via the serial interface, and an analog temperature sensor. The part incorporates digital offset and gain adjust registers per channel. The AD5744R is a high performance converter that provides guaranteed monotonicity, integral nonlinearity (INL) of ± 1 LSB, low noise, and 10 µs settling time. The AD5744R includes an onchip 5 V reference with a reference temperature coefficient of 10 ppm/°C maximum. During power-up when the supply voltages are changing, VOUTx is clamped to 0 V via a low impedance path.

The AD5744R is based on the *i*CMOS* technology platform, which is designed for analog systems designers within industrial/instrumentation equipment OEMs who need high performance ICs at higher voltage levels. *i*CMOS enables the development of analog ICs capable of 30 V and operation at ±15 V supplies, while allowing reductions in power consumption and package size, coupled with increased ac and dc performance.

The AD5744R uses a serial interface that operates at clock rates of up to 30 MHz and is compatible with DSP and microcontroller interface standards. Double buffering allows the simultaneous updating of all DACs. The input coding is programmable to either twos complement or offset binary formats. The asynchronous clear function clears all DATA registers to either bipolar zero or zero scale, depending on the coding used. The AD5744R is ideal for both closed-loop servo control and open-loop control applications. The AD5744R is available in a 32-lead TQFP and offers guaranteed specifications over the -40°C to +85°C industrial temperature range (see Figure 1 for the functional block diagram).



پیکربندی پینهای مبدل AD5744 در شکل زیر نشان داده شده است

شکل ۱۰۴- پیکربندی پینهای مبدل AD5744

همچنین توضیحی در مورد عملکرد پینهای مبدل را میتوانید در جدول زیر مشاهده کنید.

Pin No.	Mnemonic	Description
1	SYNC	Active Low Input. This is the frame synchronization signal for the serial interface. While SYNC is low, data is transferred
		in on the falling edge of SCLK.
2	SCLK	Serial Clock Input. Data is clocked into the shift register on the falling edge of SCLK. This operates at clock
		speeds of up to 30 MHz.
3	SDIN	Serial Data Input. Data must be valid on the falling edge of SCLK.
4	SDO	Serial Data Output. This pin is used to clock data from the serial register in daisy-chain or readback mode.
5	CLR	Negative Edge Triggered Input. ¹ Asserting this pin sets the data register to 0x0000.
6	LDAC	Load DAC. This logic input is used to update the data register and, consequently, the analog outputs. When tied permanently low, the addressed data register is updated on the rising edge of SYNC. If LDAC is held high during
		the write cycle, the DAC input register is updated, but the output update is held off until the falling edge of LDAC. In this mode, all analog outputs can be updated simultaneously on the falling edge of LDAC. The LDAC pin must not be left unconnected.
7,8	D0, D1	Digital I/O Port. D0 and D1 form a digital I/O port. The user can set up these pins as inputs or outputs that are configurable and readable over the serial interface. When configured as inputs, these pins have weak internal pull-ups to DV _{cc} . When programmed as outputs, D0 and D1 are referenced by DV _{cc} and DGND.
9	RSTOUT	Reset Logic Output. This is the output from the on-chip voltage monitor used in the reset circuit. If desired, it can be used to control other system components.
10	RSTIN	Reset Logic Input. This input allows external access to the internal reset logic. Applying a Logic 0 to this input clamps the DAC outputs to 0 V. In normal operation, RSTIN should be tied to Logic 1. Register values remain unchanged.
11	DGND	Digital Ground Pin.
12	DVcc	Digital Supply Pin. Voltage ranges from 2.7 V to 5.25 V.
13, 31	AVDD	Positive Analog Supply Pins. Voltage ranges from 11.4 V to 16.5 V.
14	PGND	Ground Reference Point for Analog Circuitry.
15, 30	AVss	Negative Analog Supply Pins. Voltage ranges from –11.4 V to –16.5 V.
16	ISCC	This pin is used in association with an optional external resistor to AGND to program the short-circuit current of the output amplifiers. Refer to the Design Features section for more information.
17	AGNDD	Ground Reference Pin for DAC D Output Amplifier.
18	VOUTD	Analog Output Voltage of DAC D. Buffered output with a nominal full-scale output range of ± 10 V. The output amplifier is capable of directly driving a 10 k Ω , 200 pF load.
19	VOUTC	Analog Output Voltage of DAC C. Buffered output with a nominal full-scale output range of ± 10 V. The output amplifier is capable of directly driving a 10 k Ω , 200 pF load.
20	AGNDC	Ground Reference Pin for DAC C Output Amplifier.
21	AGNDB	Ground Reference Pin for DAC B Output Amplifier.
22	VOUTB	Analog Output Voltage of DAC B. Buffered output with a nominal full-scale output range of ±10 V. The output amplifier is capable of directly driving a 10 kQ. 200 pF load.

Pin No.	Mnemonic	Description
23	VOUTA	Analog Output Voltage of DAC A. Buffered output with a nominal full-scale output range of ± 10 V. The output amplifier is capable of directly driving a 10 k Ω , 200 pF load.
24	AGNDA	Ground Reference Pin for DAC A Output Amplifier.
25	REFAB	External Reference Voltage Input for Channel A and Channel B. The reference input range is 1 V to 7 V, and it programs the full-scale output voltage. $V_{REFIN} = 5 V$ for specified performance.
26	REFCD	External Reference Voltage Input for Channel C and Channel D. The reference input range is 1 V to 7 V, and it programs the full-scale output voltage. $V_{REFIN} = 5 V$ for specified performance.
27	REFOUT	Reference Output. This is the reference output from the internal voltage reference. The internal reference is $5 V \pm 3 \text{ mV}$ at 25° C, with a reference temperature coefficient of 10 ppm/°C.
28	REFGND	Reference Ground Return for the Reference Generator and Buffers.
29	TEMP	This pin provides an output voltage proportional to temperature. The output voltage is 1.47 V typical at 25°C die temperature; variation with temperature is 5 mV/°C.
32	BIN/2sCOMP	This pin determines the DAC coding. This pin should be hardwired to either DV _{cc} or DGND. When hardwired to DV _{cc} , input coding is offset binary (see Table 7). When hardwired to DGND, input coding is twos complement (see Table 8).

جدول ۸- عملکرد پینهای مبدل AD5744

فرمت شيفت رجيستر ورودى

LSB

شیفت رجیستر ورودی شامل ۲۴ بیت میباشد که دادههای این شیفت رجیستر از قسمت MSB با کنترل کلاک ورودی، به مبدل وارد میشوند. شیفت رجیستر شامل یک بیت برای read یا write میباشد. همچنین یک بیت صفر همواره در بین بیست و سوم این رجیستر باید set شود. ۳ بیت برای انتخاب رجیستر و ۳ بیت برای آدرسدهی کانالها مورد استفاده قرار گرفتهاست. از طرفی این رجیستر ۱۶ بیت داده دیجیتال را نیز شامل میشود.

فرمت این شیفت رجیستر ورودی به صورت شکل زیر میباشد.

	-	-	-
		-	-
		•	
		•	•
			-
		-	
			_

DB23	DB22	DB21	DB20	DB19	DB18	DB17	DB16	DB15 to DB0
R/W	0	REG2	REG1	REG0	A2	A1	A0	Data

شكل ۱۰۵- فرمت شيفت رجيستر ورودى مبدل AD5744

همچینن توضیح عملکرد بیتهای این شیفت رجیستر در جدول زیر توضیح داده شده است.

Table 10. Input Shift Register Bit Function Descriptions

Register Bit	Description	Description								
R/W	Indicates a read from or a write to the addressed register									
REG2, REG1, REG0	Used in asso register, gain	Used in association with the address bits, determines if a read or write operation is to the data register, offset register, gain register, or function register.								
	REG2	REG1	REGO	Function						
	0	0	0	Function register						
	0	1	Data register							
	0	0 1 1 Coarse gain register								
	1	0	0	Fine gain register						
A2, A1, A0	Decodes the DAC channels									
	A2	A1	A0	Channel Address						
	0	0	0	DAC A						
	0	0	1	DAC B						
	0	1	0	DAC C						
	0	0 1 1 DAC D		DAC D						
	1	0	0	All DACs						
Data	Data bits		•	·						

جدول ۹- عملکرد بیتهای شیفت رجیستر ورودی مبدل AD5744

فرمت data رجیستر Data رجیستر به وسیله سه بیت REG با تنظیم 010 آدرسدهی می شود. همچنین کانال آنالوگ مورد نظر با انتخاب سه بیت A0 و A1 و A2 تعیین می گردد. از طرفی ۱۴ بیت داده مورد نظر در مکان های DB2 تا DB15 قرار گرفته اند که تو بیت آخر آن نیز دلخواه می باشد

Table 13. Programming the Data Register

REG2	REG1	REG0	A2	A1	A0	DB15 to DB2	DB1	DB0
0	1	0	D	AC addre	ess	14-bit DAC data	Х	Х

شكل ۱۰۶-فرمت Data رجيستر مبدل AD5744

فرمت function رجيستر

به دلیل اینکه ما قصد داریم خروجی آنالوگ خود را در حالت متمم ۲ نیز مشاهده کنیم باید قبل از واردکردن دادههای دیجیتال به آیسی، رجیستر clear را آدرسدهی کنیم که آدرسدهی آن درمبدل به صورت زیر میباشد.

REG2	REG1	REGO	A2	A1	AO	DB15 to DB6	DB5	DB4	DB3	DB2	DB1	DBO
0	0	0	0	0	0		NOP, data = don't care					
0	0	0	0	0	1	Don't care	Local ground offset adjust	D1 direction	D1 value	D0 direction	D0 value	SDO disable
0	0	0	1	0	0	Clear, data = don't care						
0	0	0	1	0	1		Load, data = don't care					

شكل ۱۰۷- فرمت function رجيستر مبدل AD5744

یک خصوصیت مهم این مبدل لینک سریال با پردازندهی کنترل خود است. که برای درک عملکرد این مبدل بهتر است به دیاگرام زمانبندی در شکل زیر مراجعه کنیم.



شکل ۱۰۸ - دیاگرام زمانبندی مبدل AD5744

۲- مرحله دوم

طبق دیاگرام زمان بندی AD5744 نحوه عملکرد آنرا را بر اساس پروتکل SPI طراحی کنید.

۳- مرحله سوم طرح ایجاد شده را شبیه سازی کرده و آنرا با دیاگرام زمان بندی AD5744 مقایسه کنید.

خروجی های مورد انتظار (برای تهیه گزارش کار)

- ارائه مفاهیم تکمیلی در رابطه با این آزمایش
 - گزارش شبیهسازی و سنتز طرح
- مشکلاتی که در حین برنامهنویسی، شبیهسازی و پروگرم کردن با آنها مواجه شدید.

تمرين

در مورد ۵ نوع مبدل دیجیتال به آنالوگ که بر اساس پروتکل SPI کار می کند و نحوه عملکرد آنها گزارشی ارائه دهید.