أسحه يعرابه



دستور کار آزمایشگاه معماری کامپیوتر



دانشكدهٔ مهندسی

گروه کامپيوتر

تهيه دننده :

محمدرضا فتاح

آزمایش اول

هدف : طرامی مسیر داده از ریزیردازندهٔ هشت بیتی

نمای کلی یک پردازنده هشت بیتی در شکل زیر (سمت چپ) نشان داده شده است بخشهایی که در داخل کادر خط چین قرار گرفته هستهٔ پردازنده را تشکیل می دهد و حافظهٔ برنامه و حافظهٔ موقت در خارج از آن قرار می گیرد . معمولاً CPU را در اینجا به دو بخش مسیر داده و



کنترل تقسیم می کنیم . در شکل سمت راست جزئیات مسیر داده آورده شده است . در آزمایش اول ، طراحی و پیاده سازی مسیر داده را به انجام خواهیم رساند و در آزمایشات بعدی بخش کنترل را به آن اضافه خواهیم نمود .

> جدول شمارهٔ ۱ لیست سیگنالهای موجود در مسیر داده را در چهار گروه مجزا نشان می دهد . کارکرد هر یک از سیگنالها در زیر توضیح داده شده است:

-۱ Muxsel -۱ سیگنال انتخاب مسیر دیتا که در این آزمایش فقط مسی است .

1	زمایش فقط مسیر باس دیتا و پورت ورودی فعال ا			5-ALU_Buf 6-Reg_Buf
اد	ست .		جدول ۱	
ז – ר	Wrei: فعال ساز نوشتن در مجموعهٔ رجیسترهای			
6	مومی . مجموعهٔ رجیستر دارای هشت رجیستر عمومی	ل است که شامل ی	ک خط ورودی و دو خط خروجی هش	ئىت بيتى ديتاست .

نوشتن در این مجموعه وابسته به کلاک است ولی خواندن مستقل از کلاک است .

۳- Wadd: خط آدرس و انتخاب رجیستر برای نوشتن .

سيختالهاي مسير داده									
سیگنال کنترل	سیگنال داده	سیگنال آدرس	ىيگنال سىيىتم						
1-Muxsel	1-Data_Bus	1-Aadd	1-Clock						
2-Wren	2-Input	2-Badd	2-RESET						
3-func_sel	3-Output	3-Wadd							
4-Cin									
5-ALU_Buf									
6-Reg_Buf									
	1 (1.)								

- ۴- Aadd و Badd : خطوط آدرس خواندن از رجیسترها برای هر یک از دو خروجی A و B . پس در این مجموعه رجیستر ، میتوان عمل خواندن همزمان از دو رجیستر را انجام داد .
 - -۵ Func_sel : خط انتخاب نوع عملیات جمع و یا تفریق برای ALU.
 - ۶- Cin : ورودی بیت نقلی برای عملیات جمع و تفریق
 - ALU_Buf و Reg_Buf : فعال ساز بافر خروجی ALU و خروجی A فایل رجیستر

اابتدا المانهای لازم برای بخش مسیر داده را طراحی کنید سپس با توجه به شکل مسیر داده در بالا ، المانها را به صورت مناسب بهم متصل کنید . با دادن پورتهای ورودی به تمامی سیگنالها مدار را برای شبیه سازی آماده کنید .

در مرحلهٔ دوم می خواهیم برنامه گفته شده در جدول ۴ را توسط این مدار اجرا کنیم . ابتدا مقادیر مناسب برای سیگنالها را در جداول زیر مشخص کنید سپس در محیط شبیه سازی عملکرد مدار را مشاهده نمایید .

	وضعيت سيگنالهاي كنترلي براي دستورات									
Reg_Buf Alu_Buf Cin Func_sel Wren Muxsel										
LDI	Rn,k	0	0	х	х	1	00			
ADD	Rn,Rm	0	1	0	1	1	11			
SUB	SUB Rn,Rm									
	جدول ۲									

وضعيت سيگنالهاي آدرس								
Wadd Badd Aadd								
LDI	Rn,k	n	х	х				
ADD	Rn,Rm	n	m	n				
SUB	Rn,Rm							
جدول ۳								

ات	دستور	، آدر س	یت سیگنال	وضعب	سیگنال دیتا				
		Wadd	Badd	Aadd	Input	Out			
LDI	R ₀ ,8	0	х	х	8	х			
LDI	R ₂ ,5								
ADD	R_0, R_2	0	2	0	х				
SUB	R_0, R_2								
	جدول ۴								

آزمایش دوم

هدف : طراعی بخش کنترل پردازنده

قسمت اول : آشکارساز کد دستورالعمل (Opcode)

در آزمایش قبل برای اجرای هر دستور ، مقدار دهی به سیگنالهای کنترلی را به صورت دستی انجام دادیم . در این آزمایش با طراحی مدار آشکار ساز کد وظیفه مقدار دهی به سیگنالهای کنترلی رل بهدهٔ این مدار می گذاریم . برای این منظور ابتدا به هر دستور یک شماره اختصاص می دهیم می خواهیم با دادن این کد به ورودی آشکار ساز ، مقادیر مناسب را برای سیگنالها دریافت کنیم .

پس برای این منظور نیاز به طراحی یک مدار منطقی برای دیکد کردن کدهای مربوطه داریم . ساده ترین روش برای پیاده سازی این مدار استفاده از حافظه ROM است . این حافظه باید دارای چهار ورودی و یک خروجی شانزده بیتی باشد . با این مدار می تواند شانزده دستور را برای پردازنده اختصاص داد . با توجه به شکل زیر ، آشکار ساز را بسازید و به مدار پردازنده اضافه کنید .

اکنون اجرای برنامهٔ مربوط به آزمایش قبل را برای مدار جدید تکرار کنید .



قسمت دوم : مقدار دهی سیگنالهای آدرس و داده از طریق قسمت Oprand دستورالعمل

در آزمایش قبل برای اجرای هر دستور مقدار دهی به سیگنالهای کنترلی را از طریق آشکارساز کد انجام دادیم . اکنون می خواهیم سیگنالهای آدرس و داده نیز به صورت خودکار و از طریق قسمت Oprand دستورالعمل مقدار دهی شوند .

در صفحهٔ ۶ قالب هر یک از دستورات پردازنده مشخص شده است . هر دستورالعمل دارای یک قالب شانزده بیتی است که پنج بیت اول آن کد و شمارهٔ دستور و یازده بیت باقیمانده ، مبدأ و مقصد اطلاعات را برای هر دستور مشخص می کند .

به طور مثال به قالب دستورات زیر توجه کنید :

instruction	Opcode(5bit)	Operands(11bit)			operation
LDI Rn,k	00001	Rn		k	Rn=k
MOV R _n ,R _m	00010	Rn	Rm		R _n =R _m

در دستور LDI باید مقدار ثابت و عددی k در ثبات Rn ذخیره گردد . در قالب این دستور پنج بیت بالای آن برای کد دستور و سه بیت بعدی ، آدرس ثبات مقصد در بانک رجیستر را مشخص می کند . همچنین هشت بیت باقیمانده مقدار باینری عدد مورد نظر را در خود نگهدار می دارد .

در دستور MOV مقداری که در ثبات Rm وجود دارد باید در ثبات مقصد یعنی Rn ذخیره شود در این دستور نیز پنج بیت اول کد دستور ، سه بیت بعدی مقصد داده و سه بیت بعد از آن مبدأ اطلاعات را مشخص می کند . پنج بیت باقیمانده در این دستور کاربردی ندارند .

در آزمایش قبل پنج بیت بالای دستورالعمل (Opcode) را به ورودی مدار آشکار ساز کد دستورالعمل متصل نمودیم و سیگنالهای کنترلی سیستم را به خروجی آشکار ساز اتصال دادیم . در آزمایش امروز می خواهیم وضعیت سیگنالهای آدرس و دیتا را با استفاده از یازده بیت پایین دستورالعمل (Oprand) مشخص کنیم .

در آزمایش امروز ، سیستم فقط دارای دو پورت ورودی است یکی برای کلاک و دیگری یک پورت شانزده بیتی برای اِعمال دستورالعمل به سیستم. همچنین همانند قبل یک پورت خروجی برای نشان دادن وضعیت باس داده خواهیم داشت . بنابراین بقیهٔ پورتها باید حذف گردند و وضعیت سیگنالهای داده و آدرس از طریق Oprand مشخص گردد .اکنون مدار سیستم را به صورت مناسب تغییر دهید .

برای اجرای برنامهٔ داده شده در جدول زیر ابتدا کد ماشین برای هر دستور را بدست آورید و در جدول یادداشت کنید سپس در محیط شبیه سازی عملکرد سیستم را برای این برنامه مشاهده کنید .

٩	برنام	کد ماشین (16bit)
LDI	RO,8	
LDI	R2,5	
ADD	R0, R2	
SUB	R0,R2	
		جدول ۵

орс	Instruction	Cbus15	Cbu14	Cbu13	Cbus12	Cbus11	Cbus10	Cbus9	Cbus8	Cbus7	Cbus6	Cbus5	Cbus4	Cbus3	Cbus2	Cbus1	Cbus0
0	Fetch																
1	LDI R _n , k																
2	MOV R _n , R _m																
3	LDS Rn, k																
4	STS k, Rn																
5	ST R _n , R _m																
6	LD R _n , R _m																
7	ADD R _n , R _m																
8	SUB R _n , R _m																
9	JMP Lable																
10	JNZ R _n , Lable																
11	CALL Lable																
12	RET																
13																	
14																	
15																	
	جدول ۶ ریز دستورالعمل																

قالب دستورات پردازنده

instruction	Opcode(5bit)		Operands(11bit)	operation
LDI Rn,k	00001	Rn	k	Rn=k
MOV R _n , R _m	00010	Rn	R _m	R _n =R _m
LDS Rn,k	00011	Rn	k	Rn=Sram(K)
STS k,R _n	00100	Rn	k	Sram(k)=Rn
ST Rn,R _m	00101	Rn	Rm	Sram(Rm)=R _n
LD R _n ,R _m	00110	Rn	Rm	R _n =Sram(R _m)
ADD Rn,Rm	00111	Rn	Rm	Rn=Rn+Rm
SUB Rn,Rm	01000	Rn	Rm	Rn=Rn-Rm
JMP Lable	01001	000	Lable	PC=Lable
JNZ Lable	01010	R _n	Lable	lf(Rn!=0)PC=Lable
CALL Func	01011	000	Func	Stack←Pc , Pc←Func
RET	01100	000	Func	PC←STACK
[01101			
[01110			

آزمایش سوم

قسمت اول : ثبات دستورالعمل و انتفاب کنندهٔ مالت واکشی دستور (Fetch) و اجرا (Execute)

با توجه به ساختار کلی پردازنده (شکل ۱ در آزمایش اول) ، مدار دارای دو بخش کلی مسیر داده و کنترل است . در آزمایش اول مسیر داده را طراحی و پیاده سازی کردیم . در آزمایش دوم از بخش کنترل مدار ، آشکارساز کد دستور را به مدار اضافه نمودیم اکنون در این آزمایش و آزمایش بعد میخواهیم قسمت کنترل را کامل کنیم . قسمت کنترل پردازنده شامل ثبات دستورالعمل ، انتخاب کنندهٔ حالت اجرا (Execute)یا و وکشی دستور (Fetch) و همچنین آشکار ساز کد است . در شکل زیر نمای کلی این بخش آورده شده است .



در این مدار ابتدا کد ماشین دستورالعمل روی ورودی ثبات دستورالعمل قرار داده می شود برای بارگذاری ورودی در داخل ثبات ، باید پایه های Load_reg و LoakEn در حالت فعال قرار گیرند با آمدن لبهٔ بالا روندهٔ کلاک ، مقدار ورودی در ثبات ذخیره می گردد . ذخیره شدن کد دستور در ثبات مربوطه ، قسمتی از عملیات واکشی دستور (Fetch) است که در یک کلاک انجام می شود . در این حالت تمام سیگنالهای کنترلی بخش مسیر داده ، باید در حالت فیر قرار گیرند .

در پردازنده برای اجرای هر دستور دو حالت واکشی و اجرا بدنبال هم اتفاق میافتد . قسمت مقسم دو از بخش کنترل ، وظیفهٔ انتخاب این دو حالت را به عهده دارد . خروجی این مدار به خط انتخاب یک مالتی پالکسر دو به یک متصل است . در کلاک اول خروجی مدار مقسم دو صفر است در نتیجه خط انتخاب مالتی پلکسر صفر خواهد شد و در این حالت مقدار صفر که در اصل کد عملیات Fetch است روی ورودی آشکار ساز کد قرار خواهد گرفت . با این توصیفات برای تولید مقادیر سیگنالهای کنترلی جهت عملیات Fetch ، مقادیر مربوطه باید در آدرس صفر حافظهٔ آشکار ساز آشکار ساز کد قرار خواهد گرفت . با این توصیفات برای تولید مقادیر سیگنالهای کنترلی جهت عملیات Fetch ، مقادیر مربوطه باید در آدرس صفر حافظهٔ آشکار ساز آشکار ساز مقاد گرفت . با این توصیفات برای تولید مقادیر سیگنالهای کنترلی جهت عملیات Fetch ، مقادیر مربوطه باید در آدرس صفر حافظهٔ آشکارساز کد نوشته شود . در کلاک دوم خروجی مقسم دو به مقدار یک تغییر می کند در نتیجه مقدار Bocode دستور فعلی روی خروجی مالتی بازی برای از کرد فروجی مقدار می گرد در نتیجه مقدار مالت وردی آشکار ماز آشکارساز کد نوشته شود . در کلاک دوم خروجی مقسم دو به مقدار یک تغییر می کند در نتیجه مقدار Bocode دستور فعلی روی خروجی مالتی پلازم بذکر مالتی پلکسر و در نتیجه ورودی آشکار ساز کد قرار می گیرد و متعاقب آن سیگنالهای کنترلی برای اجرای آن دستور تولید می شود . لازم بذکر است در هنگام چرخهٔ اجرا سیگنالهای کنترلی قست در هنگام چرخهٔ اجرا سیگنالهای کنترلی قست در هنگام مالتی در هنگام پرخهٔ اجرا سیگنالهای کنترلی قسمت مالتی .

اکنون با اضافه کردن المانهای مناسب به مدار پردازنده ، اجرای برنامهٔ داده شده در جدول ۵ را برای این آزمایش تکرار کنید .

قسمت دوم : مافظهٔ برنامه و شمارندهٔ برنامه (Program Counter)

در یک پردازنده برای اجرای یک برنامه ، ابتدا کدهای زبان ماشین برنامه در یک حافظه که در کنار پردازنده قرار می گیرد ذخیره می شود . خطوط آدرس این حافظه توسط شمارندهٔ داخلی cpu به نام شمارندهٔ برنامه (PC) مقدار دهی می گردد . مقدار شمارنده با واکشی هر دستور (Fetch) افزایش می یابد . خروجی حافظهٔ برنامه نیز به ورودی ثبات دستورالعمل بخش کنترل متصل می شود .



تکلیف : برای اجرای دستور JMP Lable ، سخت افزار مدار را بصورت مناسب تغییر دهید . سپس برنامهٔ زیر را اجرا نمایید:

نامه	برن	کد ماشین (16bit)
LDI	RO , 3	
LDI	R2,1	
Loop:		
ADD	R0, R2	
JMP	Loop	
		جدول ۷



قسمت دوم: قابلیت تعریف یشته در مافظهٔ موقت (SRAM)

برای اجرای دستوراتی مثل PUSH و یا POP و همچنین دستور CALL و RET به وجود پشته (Stack) در کنار پردازنده نیاز داریم . معمولاً پشته در فضای حافظهٔ موقت SRAM تعریف میگردد . برای آدرس دهی و مدیریت این فضا پردازنده نیاز به یک اشاره گر پشته دارد.

برای ذخیرهٔ یک کمیت در پشته ، ابتدا مقدار مورد نظر در آدرسی که اشاره گر به آن اشاره می کند ذخیره شده سپس به مقدار اشاره گر یکی اضافه میشود و در مرحلهٔ برداشتن مقدار از پشته ، ابتدا از مقدار اشاره گر یکی کم شده سپس مقدار از پشته برداشته می شود . ساختن اشاره گر پشته توسط ویزارد نرم افزار کمی مشکل است بنابراین این المان بصورت آماده در اختیار شما قرار می گیرد .

		1			يشته	د اشاره گر ې	جدول عملكر	<u> </u>
Polk	n			PSET	clock	Push	POP	out
	Ppush	<u> </u>	1	1	Х	Х	Х	FF (Hex)
[0"2			2	0	0→1	0	1	Pout + 1
- Pset 5	Ррор		3	0	0→1	1	0	Pout - 1
		J	4	0	0→1	0	0	No change
برنامه	(16	کد ماشین (bit		برات	RET تغي	، CALL و	ی دستورات	ايجاد قابليت اجرا
LDI R0,8		,			جرا نماييد :	ده شده را ا-	س برنامهٔ دا	دار ایجاد کنید سپ
LDI R3,5								
CALL Func								
Loop:								
JMP Loop								
Func:								
ADD R0, R2			•••••					
RET								
	جدول ۹							
						جرا کنيد :	دول ۱۳ را ا	امهٔ داده شده در ج
برنامه	(16	کد ماشین (bit						
LDI RO, 8								
CALL Func								
Loop:								
JMP Loop								
Func:								
PUSH RO								
POP R2								
ADD R0, R2								
RET								





راهنمای نرم افزار QUARTUS

۱



دانشکدهٔ برق و کامپیوتر آزمایشگاه مدار منطقی محمّدرضا فتّاح

راهنمای نرم افزار آشنایی با نرم افزار کوارتوس (Quartus)

مقدمه

شرکت ALTRA که در زمینهٔ ساخت آی سی های برنامه پذیر مانند FPGA فعالیت می کند نرم افزار کوارتوس را برای طراحی مدارهای منطقی و پیاده سازی آن روی آی سی های برنامه پذیر در اختیار کاربران قرار داده است . در این نرم افزار با روشهای مختلف از جمله برنامه نویسی متنی و یا روش شماتیک می توان عمل طراحی و شبیه سازی و در نهایت سنتز مدار را انجام داد.

در این متن روش طراحی توسط زبان وریلاگ وشبیه سازی و سپس پیاده سازی آن را با استفاده از این نرم افزار توضیح خواهیم داد.

۱– ايم*ا*د پروژه

در این نرم افزار هر پروژه شامل چندین فایل از جمله فایل اصلی برنامه و فایلهای فرعی دیگر است برای جلوگیری از پراکندگی فایلها ، حتماً سعی کنید در ابتدا یک پوشه با نام مناسب بسازید . توجه داشته باشید که نام پوشه و یا فایلها حتماً با استفاده از حروف اصلی لاتین و بدون استفاده از کارکترهایی مانند & و غیره باشد . در غیر این صورت هنگام کامپایل برنامه با خطا مواجه خواهید شد . البته ایجاد پوشه را می توانید هنگام ساخت پروژه نیز انجام دهید.

قصد ما در این جا طراحی و پیاده سازی مدار تمام جمع کنندهٔ یک بیتی است بنابراین نام فایل را FullAdder_1bit انتخاب کنید که مربوط به آزمایش اول دستور کار (بخش دوم) است . برای ایجاد پروژه مراحل زیر را دنبال کنید:

نرم افزار را اجرا کنید و طبق شکل (سمت چپ)از منوی File گزینه New Project Wizard را انتخاب کنید.

Quartus II 1 AX-PLUS IFile Sit View Project Assignments Pro New Ctrl+N Open Ctrl+F Corvert Max+PLUS II Project Save Project Class Brainet	Next	New Project Wizard: Directory, Name. Top-Level Entity [page 1 of 5] What is the wacking directory for this project? C:Ubers/veza/Desktop/TestProject Uhat is the name of this project? C:TestProject Uhat is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file. Fulldade_Ibil Use Existing Project Settings
		< Back Next > Finish Cancel

توجه داشته باشید در این مرحله فقط پروژهٔ معرفی شده ایجاد می شود ولی فایل مربوطه با نام داده شده در مرحلهٔ ۳ ، بعداً باید با همین نام ساخته شود. نوع و فرمت این فایل (متنی یا شماتیک) در هنگام ساخت مشخص می شود توجه داشته باشید که این فایل اصطلاحاً TopLevel نامیده می شود. اگر در پروژه ساخته شده فایلی با این نام وجود نداشته باشد در هنگام کامپایل ، پیغام خطای عدم وجود فایل TopLevel داده خواهد شد. کلید Next را بزنید تا پنجرهٔ زیر (سمت چپ) ظاهر شود.



در این مرحله آی سی برنامه پذیری که قصد داریم طرح نهایی مدار خود را روی آن پیاده سازی کنیم انتخاب میکنیم. بورد آموزشی آزمایشگاه مدارهای منطقی که آزمایشهای خود را با آن انجام میدهیم مبتنی بر آی سی با نام ALTRA آست. این المان یک آی سی برنامه پذیر CPLD ساخت شرکت ALTRA است . حروف EP در ابتدای نام ، نشانهٔ این است که حافظهٔ برنامه پذیر این آی سی از نوع EPROM است . حروف MAX7000 نشان دهندهٔ خانوادهٔ این آی سی یعنی MAX7000S و عدد 128 نشانگر ظرفیت مدارات داخلی این المان بر حسب MacroCell است (۱۲۸ ماکروسل). عدد84 تعداد پایه های آی سی را نشان میدهد. تعدادی از این پایه ها مربوط به تغذیه آی سی و تعدادی دیگر مربوط به برنامه ریزی و باقیمانده می توانند بعنوان ورودی و یا خروجی (I/O) استفاده گردند.



۲- ایماد فایل و کامپایل برنامه

تا این مرحله چگونگی ایجاد یک پروژه به پایان رسید اکنون می خواهیم یک فایل متنی برای نوشتن و طراحی یک جمع کنندهٔ کامل یک بیتی را شروع کنیم. طبق شکل زیر(سمت چپ) از منوی File گزینهٔ New را انتخاب و در پنجرهٔ باز شده گزینه Verilog HDL را انتخاب نمایید.

Quartus II - C:/Users/reza/Desktop/TestProject/TestProject MAX+PLUS II File Edit View Project Assignments New Ctrl+N Open Ctrl+O Cose Ctrl+F4 Project Navigato Ertity MAX7000 Ertity MAX7000 MAX7000 Sopen Project Ctrl+J Convert MAX+PLUS II Project	ایجاد فایل شماتیک ایجاد فایل متنی وریلاگ ایجاد فایل متنی VHDL	New	2
FullA → FullA			OK Cancel

اکنون فایل متنی وریلاگ ایجاد شد با انتخاب گزینهٔ Save از منوی File فایل ایجاد شده را با نامی که قبلاً در مرحلهٔ ایجاد پروژه برای فایل TopLevel انتخاب کرده بودید ذخیره نمایید .توجه داشته باشید اگر مراحل بالا را بدرستی انجام داده باشید نام فایل پیش فرض بصورت خودکار همان نام مورد نظر و در مسیر پوشه ای که ساخته اید خواهد بود.

با ساخت فایل متنی اکنون می خواهیم طراحی و شبیه سازی و سنتز مدار جمع کنندهٔ کامل را شروع کنیم.ابتدا برنامهٔ مدار را در فایل ایجاد شده با زبان وریلاگ بنویسید(می توانید آنرا از داخل دستور کار کپی کنید) .



حال می توانیم برنامه را کامپایل کرده از خطاهای احتمالی مطلع شده وآنرا برطرف کنیم .برای این منظور از هر یک از گزینه های مشخص شده در شکل بعد(سمت چپ) می توانید استفاده کنید.

signments Processing) Tools Window Help	\land				
FullAddert U Stop Processing Ctrl+Shift+C					
Update Memory Initialization File Compilation Report Ctrl+R Start Compilation & Simulation Ctrl+Shift+K	2, A, B, Cin);	Analysis & Synthesis	Fitter	Assembler 89 %	Timing Analyzer
Generate Functional Simulation Netlist					
10 Simulation Report Ctrl+Shift+R	>		Full Compilation - A	Assembler	
Timing Analyzer Tool PowerPlay Power Analyzer Tool		▶ Start	🔠 Stop		Report

پیامهای خطا و اخطار بصورت شکل زیر در پنجرهٔ پیغامها که در زیر پنجرهٔ فایل قرار دارد ظاهر خواهد شد .



با دوبار کلیک روی پیغام خطای احتمالی محل آن در فایل برنامه مشخص می شود .

س_ شبیہ سازی

در مرحلهٔ بعد با استفاده از ابزار شبیه سازی درستی عملکرد مدار را بررسی میکنیم . برای این منظور از منوی File گزینهٔ New را انتخاب کنید. در پنجره ظاهر شده و از برگهٔ OtherFile گزینهٔ Vector Waveform را گزینش نمایید.



نام بیش فرض فایل 📥 Waveform1.vwf Master Time Bar: 20.25 ns Pointer: 150 ps Interval: -20.1 ns Start 10.0 ns 20.0 ns 0 ps Value at 20.25 ns Name 20.25 ns ናጉ نوار زمان بندى ينجره ينجرة شكل موج سيگنال ليست سيگنالها + + 111

فایل شبیه سازی مانند شکل زیر ایجاد می شود:

اکنون باید سیگنالهای مدار را به فایل شبیه سازی وارد کنیم برای این منظور در پنجرهٔ لیست سیگنال، دوبار کلیک کرده و سپس مانند شکل زیر (سمت راست)مراحل مختلف را به ترتیب شماره های ذکر شده انجام دهید .

FullAdder_1bit.	/wf				Node Finder 1 2 4
faster Time Bar:	20.25 ns	Pointer:	150 ps Interval:	-20.1 ns	Named: File: Pins: all D Customize List D OK
Name	Value at 20.25 ns	Insert Node or Bus	10.0 ns	20.0 ns	Look in: FullAder 158 Cance
المركبة	₽ ₽	Name: INPUT Type: INPUT Value type: 9Level Radic: Binary Bus width: 1 Start index: 0	v 2 Ca v Node	DK	Nodes Found: Selected Nodes: Name Assignme PA Unassign PB Unassign PCin Unassign Coul Unassign PSUM Unassign SUM Unassign V FullAdder_1bitSUM Unassign Sum Unassign Sum PSUM Unassign V FullAdder_1bitSUM Unassign Sum V Sum
		Display gray code cour	t as binary count		< <u> </u>

حال سیگنالهای ورودی و خروجی مدار به فایل شبیه سازی اضافه شده است. در مرحلهٔ بعد ابتدا بازهٔ زمانی شبیه سازی و همچنین بازهٔ زمانی کل از شکل زیر و برای رازی و مرای می می می از شکل زیر و برای زمانی هر حالت از شکل مناییم . برای بازهٔ زمان کل از شکل زیر و برای زمانی هر حالت از شکل مفحهٔ بعد استفاده کنید .



در تمام آزمایشها زمان شبیه سازی را یک میلی ثانیه (1ms) انتخاب کنید.

1 Fite Edit View Project ≫ Undo End Time ⇔ Pode	Assignments Ctrl+Z 1b							
X Cut Copy	Ctrl+X Ctrl+C	Grid Size		Cin	A	В	Cout	SUM
M712 💼 Paste Dit Paste Special Repeat Paste	Ctrl+V ter	Base grid on C Clock settings:		0 0 0	0 0 1	0 1 0		
X Delete	Del - C	Period: 100 US	>	0 1 1	0 0 1	1 0 1 0		
Grid Size ★ Sort Properties	-	OK Cancel		1	1	1		
	اا دالت	انتخاب بازهٔ زمانی برای هر -		نده	نمع كنا	مدار ج	, حالات	جدول

برای انتخاب بازهٔ زمانی هر حالت به تعداد حالات موجود در جدول درستی مدار نگاه می کنیم و بازهٔ زمانی کل را به تعداد حالات تقسیم می کنیم.تعداد حالات جدول عدد هشت است که برای جلوگیری از حاصل تقسیم با مقدار اعشاری آنرا ۱۰ حالت فرض می کنیم . پس زمان هر بازه برای این مدار ۱۰۰ میکرو ثانیه (100us) خواهد بود. برای قرار گرفتن همهٔ حالات در پنجرهٔ نمایش سیگنال ،از کلیدهای Ctrl+W استفاده نمایید . اکنون زمان آن رسیده است که مقادیر ورودی را برای همهٔ حالتها طبق جدول درستی مشخص نماییم به طور مثال ورودی B را در نظر بگیرید . این سیگنال در حالت اول دارای مقدار صفر و در حالت بعدی دارای مقدار یک است.همچنین مقدار پیش فرض سیگنال در حالت اول دارای مقدار صفر و در حالت بعدی دارای مقدار یک است.همچنین مقدار پیش فرض سیگنال در شروع مقدار صفر است برای اِعمال مقدار یک به حالت دوم، نشانگر ماوس را به ابتدای قسمت دوم سیگنال B برده کلید چپ ماوس را فشار دهید و همزمان ماوس را به سمت راست و تا انتهای حالت دوم بکشید(مانند شکل زیر). این بازهٔ زمانی بصورت رنگی در می آید اکنون با انتخاب کلید آل از نوار ابزار سمت چپ صفحه ، مقدار سیگنال به مقدار یک تغییر میکند. برای بقیهٔ حالات و سیگنالهای ورودی این عمل را تکرار کنید .



Τ		Value at	0 ps	100,0 us	200,0 us	300,0 us	400,0 us	500,0 us	600,0 us	700,0 us	800 <mark>,0</mark> u
	Name	20.25 ns	20.25 ns								
*	Cin	B 0									
>	A	BO									
*	В	B 0									
	Cout	BX		*********	***********	*****	**********	**********	**********	**********	******
4	SUM	BX		****	******	******	******	******	******	***********	*******
٦											

بعد از اینکه مقادیر ورودی را مشخص کردید فایل را ذخیره نمایید . نام پیش فرض پیشنهادی که همان نام فایل اصلی ولی با پسوند متفاوت است را تایید نمایید . برای آغاز شبیه سازی مراحل نشان داده شده در شکل بعد را اجرا کنید .

	🕞 Simulator Tool
Processing Dools Window Help	Simulation mode: Timing Generate Functional Simulation Netlist
udd 🕮 Stop Processing Ctrl+Shift+C 🔯 🕨 🗞 🛼	Simulation input: C:\Users\reza\Desktop\TestProject\FullAdder_1bit.vwf
Start Compilation Ctrl+L	☐ Simulation period
Analyze Current Eile	Run simulation until all vector stimuli are used
Start • UI FullAdd	C End simulation at: 100 ns 🖃
Update Memory Initialization File	
Compilation Report Ctrl+R	Simulation options
Start Compilation & Simulation Ctrl+Shift+K	Automatically add pins to simulation output waveforms
M Generate Functional Simulation Netlict 10.45 us	Check outputs Waveform Comparison Settings
Start Simulation Ctrl+I 200.0 us	Setup and hold time violation detection
Simulation Debug	Glitch detection: 1.0 ns 💌
_ 🕀 Simulation Report Ctrl+Shift+R	Overwrite simulation input file with simulation results
	Generate Signal óctivitu File:
I P Simulator Tool	
Pl Timing Analyzer Lool	
PowerPlay Power Analyzer Tool	100 %
	💽 Start 🗋 💿 Stop 🛛 🕸 Open 🛛 🧶 Report

نتيجه بصورت زير خواهد بود:



اگر بخواهیم خروجی را به صورت عددی مشاهده کنیم مراحل را طبق شکل زیر و به ترتیب انجام دهید:





۲- پیادہ سازی مدار

با توجه به اطمینان از درستی عملکرد مدار اکنون می خواهیم طرح خود را روی آی سی برنامه پذیر پیاده سازی و عملکرد مدار را بصورت واقعی مشاهده کنیم. برای این منظور ابتدا باید عمل اختصاص دهی پایه های آی سی به ورودی و خروجیهای مدار را انجام دهیم .با توجه به شکل زیر و به ترتیب شماره ها مراحل ذکر شده را انجام دهید .



از منوی Assignments گزینهٔ Pins را انتخاب کنید . پنجره ای حاوی آی سی مورد نظر باز می شود.توجه داشته باشید پایه هایی که بصورت دایره هستند می توانند بعنوان پایه های ورودی و یا خروجی تعریف گردند.برای هماهنگی با بورد آموزشی آزمایشگاه پایه های لبهٔ پایین آی سی را برای ورودیها و پایه های لبهٔ کنار سمت راست را برای خروجیها در نظر بگیرید.

برای اختصاص دهی پایه ابتدا روی پایهٔ مورد نظر دو بار کلیک کنید. با باز شدن پنجرهٔ Pin Properties و با کلیک روی منوی باز شدنی روی اسم پایهٔ مورد نظر خود کلیک کرده تا عمل اختصاص دهی انجام شود. برای بقیهٔ پایه های مدار این کار را انجام دهید.

برای کامل شدن عملیات یک بار دیگر عمل کامپایل برنامه را انجام دهید . اکنون می توانید مدار خود را روی آی سی پروگرام نمایید . مانند شکل زیر از منوی Max+Plus گزینهٔ Programmer را انتخاب کنید .

yartus II - C:/Users/reza/De	esktop/TestProject/	TestProject - FullAdder1bit - [Pin I	Planner]	LINE ADDR. Manual House
MAX+PLUSI File Edit	View Project A	ssignments Processing Tools	Window Help	\land
\Lambda Hierarchy Display	NO N	FullAdder1bit	- 🐹 🖉 🏈 🥮	💿 🕨 🔊 🔈 🔊 💌
💫 <u>G</u> raphic Editor				<u> </u>
🕵 Sym <u>b</u> ol Editor				
S Text Editor		ચે FullAdder_1bit.v		FullAdder_1bit.vwf
🕵 Waveform Editor	Macr (±	Groups	<u>×</u>	
Korplan Editor	2	Named: ×	Groups	• 0 0
<u>C</u> ompiler		Node Name	Direc	* 0
Simulator		< <new node="">></new>		13 🔊
Timing Analyzer				14 🚺
P rogrammer	₽ ≞			15 0
A Message Processor				16 0

بعد از باز شدن پنجرهٔ جدید ابتدا باید نوع پروگرامر خود را انتخاب کنید . مطابق شکل بعد عمل نمایید.

dder_1bit	tv 1	🖸 FullAdo	der_1 bit.vwf		5	🍠 Pin Pl	anner				
lware Setu	No Hardware								Мо	de: JTA	G
ole real-time	ISP to allow background prog	gramming (for MAX II devi	ces)			6					
rt	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-	xamine	Security Bit	Erase	ISP CLAM
	FullAdder1bit.pof	EPM7128SL84	001E0030	0000FFFF							
d File ange ve Fi	Select a programming hardwar hardware setup applies only to Currently selected hardware: – Available hardware items: –	e selup to use when prog the current programmer v	gramming devices. vindow.	This programming	Add Ha Po	Hardwar rdware ty rt:	re Pe By	teBlasterM\ T1	3 / or ByteBla	ster II	
dDe	Hardware	ServerF		Remove Hardware.	Ba	ud rate: rver name					

اکنون روی کلید Start کلیک کنید . عملیات برنامه ریزی شروع می شود .

₹	🏴 Start		Device	Checksum	Userco
	Stop	FullAdder1bit.pof	EPM7128SL84	001E0030	0000FF
	Auto Detect				



راهنمای نصب نرم افزار QUARTUS

۱



دانشکدهٔ برق و کامپیوتر آزمایشگاه مدار منطقی محمّدرضا فتّاح





:٨



؛٩





روش دیگری نیز در انتهای این متن برای بدست آوردن آدرس مک آورده شده است . ۴- از پوشهٔ CRACK فایل license را با ویرایشگر Notepad باز کنید و سپس طبق شکل زیر ازمنوی EDIT گزینهٔ Replace را انتخاب کرده و مراحل جایگزینی را انجام دهید. سپس فایل را ذخیره نمایید.

Copyright	Windows [] (c) 2009	Jersion 6.1.7601] licrosoft Corporation. All rights reserve	ed. 🔒
C:\Users\I	LogicLab≻g	tmac	
Physical A	Address	Transport Name	
54-04-A6-9		\Device\Tcpip_{C5749037-5E69-4710-B791-29	92C486DFØ88>
license - N	otepad		
File Edit F	ormat View	Help	
FEATURE	Features uartus_dev SSUER=NiTR FFFFFFFF Replace	aloper alterad 2020 12 permanent uncounted DUS HOSTID-"CHANGEME" SIGN="FFFF FFFF FFFF FFF FFFF FFFF FFFF FFFF	d \ F FFFF FFFF FFFF \ FF FFFF" FF FFFF" FF FFFF FF
FEATURE	Find what: Replace with.	"CHANGEME" Find Next 5404a693bd90 Replace Replace FF FFF Replace All FF FFF	FF FFFF" ed \ F FFFF FFFF FFFF \ FF FFFF \ FF FFFF \
FEATURE	Match cas	e Cancel Ited \ FFFFF FFFFFFFFFF	F FFFF FFFF FFFF \ FF FFFF \ FF FFFF"
FEATURE			

۵- نرم افزار را اجرا کنید و مراحل بعدی را مطابق شکلهای زیر دنبال کنید:



۵

۶- انتخاب فایل license و سپس انتخاب کلید OK .



× Customize General Toolbars Commands Tcl Look & Feel Choose the preferred look and feel for the Quartus II software. You can fully customize the Quartus II software regardless of what you choose here. Click Apply and restart the Quartus II software for any change to take effect. You can also click Apply without changing the selection to reset to the factory defaults for the selected look and feel. C Quartus II Apply Quick menus • Off Quartus II menu: MAX+PLUS II menu: Left • Reset All ОК Cancel

اکنون می توانید از نرم افزار استفاده کنید.

 $-\gamma$

نحوه به دست آورن آدرس مک (آدرس فیزیکی) سیستم

کامپیوترهای شخصی و لپ تاپ ها برای به دست آوردن مک آدرس در ویندوز کافی است از قسمت اتصالات شبکه (در مورد اتصالات کابلی local area connection و در خصوص اتصالات بی سیم (wireless connection) ، اتصال مورد نظر خود را انتخاب و پنجره وضعیت(status) را باز کنید . پس از آن با زدن دکمه جزییات (details) پنجره جزییات باز شده و مقدار درج شده در مقابل physical Address همان آدرس مک کارت شبکه شما می باشد.که در شکل نیز مشخص شده است. ویندوز XP:



Local Area Connection Status	?
ieneral Support	
Internet Protocol (TCP/IP)	
Address Type:	Assigned by DHCP
IP Address:	172.31.85.25
Subnet Mask:	255.255.248.0
Default Gateway:	172.31.80.1
	Details
Repair	
	Close

N	etwork Connection Detai	is	? 🗙
	Network Connection <u>D</u> etails:		
	Property	Value	
	Physical Address	00-40-05-43-BA-C6	
	IP Address Subpot Maak	192.168.0.1 255.255.255.0	_
	Default Gateway	200.200.200.0	
	DNS Server		
	WINS Server		
			ose

ويندوز 7 , 8 :



2	Network and Sharing Center	r –	×
🛞 🍥 🝷 🛉 👱 > Control Pane	el → All Control Panel Items → Network and Sharing Cen	ter v 🖒 Search Control Panel	Q
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>T</u> ools <u>H</u> elp			
Control Panel Home	View your basic network information and set	et up connections	
Change adapter settings	View your active networks		_
Change advanced sharing settings	Wireless_science Public network	Access type: No Internet access Connections: Ethernet	
	Wireless2_arshadi Public network	Access type: No Internet access Connections: d Wi-Fi (Wireless2_arshadi)	
	Unidentified network Public network	Access type: No network access Connections: VMware Network Adapter VMnet1 VMware Network Adapter VMnet8	
Change your networking settings			
See also	Set up a new connection or network Set up a broadband, dial-up, or VPN connection	on; or set up a router or access point.	
HomeGroup	Troubleshoot problems		
Internet Options	Diagnose and repair network problems, or get	troubleshooting information.	
Windows Firewall			
Windows Mobile Device Center			

General		
Connection —		
IPv4 Connectivity:		Internet
IPv6 Connectivity:		No Internet access
Media State:		Enabled
Duration:		01:15:12
Speed:		10.0 Mbps
D <u>e</u> tails		
D <u>e</u> tails Activity ——		
D <u>e</u> tails	Sent —	Received
Details Activity Bytes:	Sent — 10,107,215	— Received 37,020,480

Connection-specific DN Description Realtek PCIe GBE Fa Physical Address 50-E5-49-31-F0-29 DHCP Enabled	amilv Controller
Description Realtek PCIe GBE Fa Physical Address 50-E5-49-31-F0-29 DHCP Enabled No	amily Controller
Physical Address 50-E5-49-31-F0-29	
HCP Enabled	
Pv4 Address 172.16.30.31	
Pv4 Subnet Mask 255.255.252.0	
Pv4 Default Gateway 172.16.30.1	
Pv4 DNS Servers 172.16.2.30	
172.16.2.31	
Pv4 WINS Server	
letBIOS over Tcpip En Yes	
ink-local IPv6 Address fe80::54a0:3907:c4f1	:e596%11
Pv6 Default Gateway	
Pv6 DNS Server	

همچنین میتوانید از دستور getmac در (cmd) command prompt برای به دست آوردن آدرس فیزیکی کلیه کارت های شبکه سیستم استفاده نمایید. در مورد سایر سیستم عامل ها مانند لینوکس و ios نیز میتوانید از روشهای مشابه استفاده کنید. به عنوان نمونه در لینوکس میتوانید از دستور ifconfig استفاده کرده و اطلاعات مربوط به پارامتر Hwaddr که همان آدرس مک سیستم است را مشاهده نمایید.

تلفنهای هوشمند و تبلت ها

در خصوص انواع تلفنهای هوشمند و تبلت ها با توجه به تنوع سیستم عامل ، روشهای مختلفی برای به دست آوردن آدرس مک وجود دارد که به عنوان نمونه برای اندروید با مراجعه به قسمت تنظیمات (setting) ، بخش سیستم(system) ، درباره دستگاه(about device) ، موقعیت(status) ، آدرس wifi mac ، میتوانید آدرس مک دستگاه خود را مشاهده نمایید.