

مدارهای منطقی برنامه پذیر (FPGA)	دانشگاه صنعتی امیر کبیر	دانشکده مهندسی برق
دکتر پورفرد	تاریخ بارگزاری در سامانه: ۱۴۰۲/۱/۳۰	تاریخ تحویل: ۱۴۰۲/۱/۱۵

تکلیف اجباری شماره ۱ (معادل میان ترم): پیاده سازی و شبیه سازی پردازنده

چند سیکلی Multi cycle

هدف از انجام این آزمایش آشنایی با مسیر داده و کنترلر در پردازنده MIPS و پیاده سازی توصیفی آن به صورت multi-cycle در زبان توصیف سخت افزار VHDL و تست آن توسط یک برنامه ی ساده با زبان اسمبلی MIPS است.

پیش از آزمایش:

- ۱- برای انجام آزمایش لازم است مطالب پیوست در رابطه با پردازنده MIPS از دانشگاه صنعتی شریف (آقای دکتر موحدین) در مورد مسیر داده و کنترلر برای ساختار multi-cycle به دقت مطالعه شوند.
- ۲- برای پیاده سازی، باید دستورات به صورت مناسب تقسیم بندی شوند و عملیاتی که در هر پالس ساعت باید انجام شود، مشخص گردند.

دستوراتی که باید پیاده سازی شوند به شرح زیر شامل دو بخش اند: اول دستورات ساده تری که در درس معماری کامپیوتر شبیه آن در پردازنده RISC-V توضیح داده شده بود و دوم دستورات تکمیلی که لازم است در این آزمایش به دسته اول اضافه شوند و در زیر با فونت پررنگ مشخص شده اند:

R format: add, sub, addu, subu, and, or, xor, nor, slt, sltu, **jr, jalr, multu, mfhrl, mflo**
I format: beq, bne, lw, sw, addi, addiu, slti, sltiu, andi, ori, xori, lui
J format: **j, jal**

- تذکر ۱: از تولید exception در عملیات جمع و تفریق اعداد علامتدار صرفنظر کنید.
- تذکر ۲: دستورات بایستی در کمترین تعداد پالس ساعت ممکن اجرا شوند.
- تذکر ۳: مشخصات زمانی عناصر مورد استفاده به شرح زیر است:
- تذکر ۴: برای تست کد خود یک فایل تست تهیه کنید و فلسفه و شیوه تهیه آنرا توضیح دهید.
- تذکر ۵: قبل از انجام پروژه ویدئوهای آموزشی بنده در رابطه با VHDL را ببینید و از اسلایدهای دانشگاه GMU در کانال درس برای یادگیری بهتر زبان VHDL می توانید استفاده کنید.
- تذکر ۶: اسلایدهای دکتر موحدین از دانشگاه صنعتی شریف در رابطه با معماری پردازنده MIPS در بخش پردازنده تک سیکلی و چند سیکلی و موضوع FSM را حتما از کانال درس ملاحظه کنید.

Memory:

Address to Read-Data propagation delay: 7 ns

Write to Read/Write access time: 2 ns

Write-Data setup time: 0.1 ns

مدارهای منطقی برنامه پذیر (FPGA)	دانشگاه صنعتی امیر کبیر	دانشکده مهندسی برق
دکتر پورفرد	تاریخ بارگزاری در سامانه: ۱۴۰۲/۱/۳۰	تاریخ تحویل: ۱۴۰۲/۱/۱۵

Write is controlled by positive edge of clock

Register File:

Read-Register to Read-Data propagation delay: 2 ns

Write-Register & Write-Data setup time: 0.1 ns

Registers:

Clock to Q delay: 0.1 ns

Input setup time: 0.1 ns

ALU: Inputs to Outputs propagation delay: 2 ns

Multiplexers: Inputs to Output propagation delay: 0.1 ns

Clock Period: 2.5 ns

کلیه فایل‌های نرم افزار ISE (ترجیحا در این بستر پیاده سازی انجام شود) یا Vivado خود را در پوشه های جداگانه به همراه فیلم توضیحات کد و توضیحات فایل تستی که طراحی کردید (فیلم برداری از دسکتاپ کامپیوتر) را در سامانه بارگزاری کنید. علاوه بر این توضیحات کد خود به زبان VHDL, Assembly را در فایل Word, PDF ای که ارسال می کنید، توضیح دهید. توجه کنید که صورت سوال را در فایل ارسالی خود مجددا تایپ کنید و اگر هر قسمت از سوال را در یک بخش از فایل توضیح می دهید، صورت سوال آن بخش را در قسمت مربوطه تایپ کنید. کلیه فایلها را RAR کرده و در قالب یک پوشه واحد با شماره دانشجویی خود و شماره تکلیف به فرمت 88123456, 88123457#HW1 ارسال کنید. همچنین نقش هر یک از اعضای گروه در انجام بخش های مختلف پروژه را در فایل ارسالی مشخص کنید.

op(31:26)								
28–26	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
31–29								
0(000)	R-format	Bltz/gez	jump	jump & link	branch eq	branch ne	blez	bgtz
1(001)	add immediate	addiu	set less than imm.	set less than imm. unsigned	andi	ori	xori	load upper immediate
2(010)	TLB	FlPt						
3(011)								
4(100)	load byte	load half	lwl	load word	load byte unsigned	load half unsigned	lwr	
5(101)	store byte	store half	swl	store word			swr	
6(110)	load linked word	lwcl						
7(111)	store cond. word	swcl						
op(31:26)=010000 (TLB), rs(25:21)								
23–21	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
25–24								
0(00)	mfc0		cfc0		mtc0		ctc0	
1(01)								
2(10)								
3(11)								
op(31:26)=000000 (R-format), funct(5:0)								
2–0	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
5–3								
0(000)	shift left logical		shift right logical	sra	sllv		srlv	srav
1(001)	jump register	jralr			syscall	break		
2(010)	mfhi	mthi	mflo	mtlo				
3(011)	mult	multu	div	divu				
4(100)	add	addu	subtract	subu	and	or	xor	not or (nor)
5(101)			set l.t.	set l.t. unsigned				
6(110)								
7(111)								

FIGURE 2.19 MIPS instruction encoding. This notation gives the value of a field by row and by column. For example, the top portion of the figure shows **load word** in row number 4 (100_{two} for bits 31–29 of the instruction) and column number 3 (011_{two} for bits 28–26 of the instruction), so the corresponding value of the op field (bits 31–26) is 100011_{two}. Underscore means the field is used elsewhere. For example, R-format in row 0 and column 0 (op = 000000_{two}) is defined in the bottom part of the figure. Hence, **subtract** in row 4 and column 2 of the bottom section means that the funct field (bits 5–0) of the instruction is 100010_{two} and the op field (bits 31–26) is 000000_{two}. The floating point value in row 2, column 1 is defined in Figure 3.18 in Chapter 3. Bltz/gez is the opcode for four instructions found in Appendix A: bltz, bgez, bltzal, and bgezal. This chapter describes instructions given in full name using color, while Chapter 3 describes instructions given in mnemonics using color. Appendix A covers all instructions.