



فاز اول پروژہ:

در این مرحله از پروژه می‌بایست در محیط شماتیک نرم‌افزار Quartus، پردازنده‌ای با مشخصات زیر را پیاده‌سازی کنید.

- فرمت و مجموعه دستورالعمل‌های محاسباتی و منطقی:

Opcode						Destination Reg					Source Reg 1					Source Reg 2					S/R Amount					Reserved					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

<i>Opcode</i>	<i>Instruction</i>	<i>Description</i>
000001	ADD	$DST \leftarrow SRC1 + SRC2$
000010	SUB	$DST \leftarrow SRC1 - SRC2$
000011	MUL	$DST \leftarrow SRC1 \times SRC2$
000100	DIV	$DST \leftarrow SRC1 \div SRC2$
000101	MOD	$DST \leftarrow SRC1 \% SRC2$
000110	MAX	$DST \leftarrow \text{MAX}(SRC1, SRC2)$
000111	MIN	$DST \leftarrow \text{MIN}(SRC1, SRC2)$
001000	NOT	$DST \leftarrow \sim SRC1$
001001	NAND	$DST \leftarrow SRC1 \sim \& SRC2$
001010	XNOR	$DST \leftarrow SRC1 \sim \wedge SRC2$
001011	SHL	$DST \leftarrow SRC1 \ll \text{S/R AMOUNT}$
001100	SHRL	$DST \leftarrow SRC1 \gg \text{S/R AMOUNT}$
001101	ROL	$DST \leftarrow \text{ROTATE LEFT}(SRC1, \text{S/R AMOUNT})$
001110	ROR	$DST \leftarrow \text{ROTATE RIGHT}(SRC1, \text{S/R AMOUNT})$
001111	SLT	$SRC1 < SRC2: DST = 1$

- **فرمت و مجموعه دستورالعمل‌های با عملکرد صریح (Immediate):**

Opcode						Destination Reg					Source Reg					Immediate Data															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

<i>Opcod</i>	<i>Instruction</i>	<i>Description</i>
010000	LDI	DST[15:0] ← IMM
010001	LUI	DST[31:16] ← IMM
010010	ADDI	DST ← SRC + SIGN EXTEND (IMM)
010011	SUBI	DST ← SRC – SIGN EXTEND (IMM)
010100	MULI	DST ← SRC × SIGN EXTEND (IMM)
010101	DIVI	DST ← SRC ÷ SIGN EXTEND (IMM)
010110	NANDI	DST ← SRC ~& SIGN EXTEND (IMM)
010111	XNORI	DST ← SRC ~^ SIGN EXTEND (IMM)

- فرمت و مجموعه دستورالعمل‌های دسترسی به حافظه:

Opcode						Value Reg (VR)					Address Reg (AR)					Offset															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

<i>Opcode</i>	<i>Instruction</i>	<i>Description</i>
011000	LW	$VR \leftarrow \text{MEM} [\$AR+ \text{SIGN EXTEND (Offset)}]$
011001	SW	$\text{MEM} [\$AR+ \text{SIGN EXTEND (Offset)}] \leftarrow VR$
011010	LB	$VR[7:0] \leftarrow \text{MEM} [\$AR+ \text{SIGN EXTEND (Offset)}]$
011011	SB	$\text{MEM} [\$AR+ \text{SIGN EXTEND (Offset)}] \leftarrow VR[7:0]$

- فرمت و مجموعه دستورالعمل‌های پرش و توقف:

Opcode						Reg 1					Reg 2					Address															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

<i>Opcode</i>	<i>Instruction</i>	<i>Description</i>
011100	JMP	PC \leftarrow PC[31:18] Address “00”
011101	JR	PC \leftarrow \$Reg1
011110	BEQ	REG1 == REG2: PC \leftarrow PC + SIGN EXTEND (Address “00”)
011111	BLT	REG1 < REG2: PC \leftarrow PC + SIGN EXTEND (Address “00”)
000000	HLT	STOP PC

توضیحات:

- طول کلمه در این معماری ۳۲ بیت است.
- تعداد ثبات‌های عمومی ۳۲ است.
- کوچکترین واحد دسترسی به حافظه بایت بوده و فضای آدرس دهی از صفر تا ۲^{۳۲}-۱ است. اما می‌توانید برای تست و سنتز مدار خود در Quartus، سائز حافظه را کوچکتر در نظر بگیرید.
- دستورالعمل‌ها به صورت خط لوله^۱ اجرا می‌شوند.
- نمایش اعداد در این معماری به صورت مکمل ۲ است.
- برای طراحی هر یک از اجزاء مورد نیاز می‌توانید در صورت نیاز از بلوک‌های اساسی در قسمت Wizard استفاده کنید.
- تمام ثبات‌ها (اعم از PC، IR و ...) می‌بایست یک درگاه ریست ناهمگام^۲ داشته باشند که به سیگنال ریست پردازنده متصل باشند (با یک شدن این سیگنال محتوای این ثبات‌ها صفر می‌شود).
- در دستورالعمل‌های BLT و SLT مقایسه مقدار دو ثبات به صورت بی‌علامت^۳ انجام می‌شود.
- انواع مخاطراتی^۴ که ممکن است در پردازنده‌ی طراحی شده رخ دهد را گزارش کرده و برای رفع آن‌ها در صورت نیاز ساختار پردازنده را تغییر دهید.
- پردازنده‌ی خود را توسط برنامه‌ای که تمام دستورالعمل‌های نام برده در آن وجود داشته باشد، بیازمایید.
- پس از اطمینان از صحت اجرای تمامی دستورالعمل‌ها، کد ماشین محاسبه جمله n ام دنباله فیبوناچی را داخل حافظه قرار دهید و نتیجه اجرای برنامه روی پردازنده و زمان اجرا بر اساس تعداد سیکل را گزارش کنید.

فاز دوم پروژه:

در این مرحله، می‌بایست از بین پروژه‌های **الف** تا **د** یکی را به دلخواه انتخاب کرده و کمک پردازنده‌ی متناظر با آن را در محیط شماتیک نرم‌افزار Quartus پیاده سازی کنید. سپس، باید پردازنده‌ی کمکی خود را به پردازنده‌ی اصلی که در فاز قبل طراحی کرده‌اید، متصل کنید تا پردازنده‌ی اصلی بتواند از این واحد برای اجرای دستورالعمل‌های کمکی استفاده کند. جزئیات طراحی کمک پردازنده، نحوه‌ی اتصال پردازنده‌ها و طراحی ساختار دستورالعمل‌ها برای کمک پردازنده بر عهده دانشجویان است. توجه

¹ Pipeline

² Asynchronous

³ Unsigned

⁴ Hazard

داشته باشید که ساختار دستورالعمل‌های کمک پردازنده باید مطابق با فرمت ساختار دستورالعمل‌های پردازنده‌ی اصلی باشد. همچنین، برای مدیریت ارتباط بین پردازنده‌ی اصلی و پردازنده‌ی کمکی می‌توانید به تعداد مورد نیاز، دستور به مجموعه دستورالعمل‌های پردازنده‌ی اصلی اضافه کنید. درنهایت، باید مشخص گردد که هر یک از عملیات خواسته شده در چند سیکل انجام می‌شود.

الف) کمک پردازنده برای محاسبات برداری: این کمک پردازنده باید قادر به انجام دستورالعمل‌های جمع، تفریق، ضرب

نقطه‌ای، ضرب عدد در بردار، تقسیم بردار بر عدد، نرمال کردن و محاسبه‌ی اندازه روی بردارها به صورت predicate باشد (یعنی عملیات برداری روی زیرمجموعه‌ای از عناصر بردار که توسط ماسک⁵ مشخص می‌شود انجام شود). همچنین، می‌توانید در صورت نیاز دستورالعمل‌های دسترسی به حافظه، کنترلی و پرشی برای کمک پردازنده در نظر بگیرید. کمک پردازنده باید بتواند یک بردار را از حافظه خوانده و آن را در حافظه ذخیره نماید. پس از اتمام مراحل طراحی، کمک پردازنده‌ی خود را توسط برنامه‌ای که تمام دستورالعمل‌های نام برده در آن وجود داشته باشد، بیازمایید. پس از اطمینان از صحت اجرای تمامی دستورالعمل‌ها، با استفاده از دستورات اسمبلی برنامه‌ای بنویسید که در آن دو ماتریس $n \times n$ را از ورودی گرفته و با استفاده از تابع predicate عناصر صفر آن‌ها را تبدیل به یک کرده و سپس دو ماتریس را در هم ضرب کند.

ب) کمک پردازنده برای اعداد مختلط: این کمک پردازنده باید قادر به انجام دستورات جمع، تفریق، ضرب، تقسیم، مقایسه،

معکوس، مزدوج و تبدیل نمایش قطبی به نمایی و یا بالعکس روی اعداد مختلط باشد. همچنین، می‌توانید در صورت نیاز دستورالعمل‌های دسترسی به حافظه، کنترلی و پرشی برای کمک پردازنده در نظر بگیرید. کمک پردازنده باید بتواند یک عدد مختلط را به هر یک از دو فرمت قطبی و یا نمایی از حافظه خوانده و آن را در حافظه ذخیره نماید. پس از اتمام مراحل طراحی، کمک پردازنده‌ی خود را توسط برنامه‌ای که تمام دستورالعمل‌های نام برده در آن وجود داشته باشد، بیازمایید. پس از اطمینان از صحت اجرای تمامی دستورالعمل‌ها، با استفاده از دستورات اسمبلی برنامه‌ای بنویسید که در آن دو عدد مختلط به فرمت قطبی را از ورودی گرفته و مزدوج حاصل تقسیم را به فرمت نمایی ذخیره کند.

ج) کمک پردازنده برای اعداد ممیز شناور: این کمک پردازنده باید قادر به انجام دستورات جمع، تفریق، ضرب، تقسیم،

مقایسه، معکوس و گرد کردن (به نزدیکترین عدد صحیح) روی اعداد ممیز شناور با دقت ساده⁶ براساس استاندارد IEEE-754 باشد. همچنین، می‌توانید در صورت نیاز دستورالعمل‌های دسترسی به حافظه، کنترلی و پرشی برای کمک پردازنده در نظر

⁵ Mask

⁶ Single Precision

بگیرید. همانند موارد استثناء در نظر گرفته شده در استاندارد IEEE^۷، برای هر یک از موارد زیر باید در خروجی سیگنالی وجود داشته باشد که آن ها را گزارش کند. این موارد عبارتند از:

- Division by zero
- QNaN (quiet not a number)
- SNaN (signaling not a number)
- Inexact
- Underflow
- Overflow

پردازنده باید بتواند یک عدد ممیز شناور را از حافظه خوانده و آن را در حافظه ذخیره نماید. پس از اتمام مراحل طراحی، کمک پردازنده‌ی خود را توسط برنامه‌ای که تمام دستورالعمل‌های نام برده در آن وجود داشته باشد، بیازمایید. پس از اطمینان از صحت اجرای تمامی دستورالعمل‌ها، با استفاده از دستورات اسمبلی برنامه‌ای بنویسید که در آن دو عدد ممیز شناور را از ورودی گرفته، عدد بزرگتر را بر عدد کوچکتر تقسیم کرده و نتیجه‌ی گرد شده را ذخیره کند.

(د) کمک پردازنده برای چند جمله‌ای‌ها: این کمک پردازنده باید قادر به انجام دستورات جمع، تفریق، ضرب، تقسیم، مشتق و محاسبه‌ی مقدار در یک نقطه روی چند جمله‌ای‌ها باشد. همچنین، می‌توانید در صورت نیاز دستورالعمل‌های دسترسی به حافظه، کنترلی و پرشی برای کمک پردازنده در نظر بگیرید. پردازنده باید بتواند یک چند جمله‌ای را از حافظه خوانده و آن را در حافظه ذخیره نماید. پس از اتمام مراحل طراحی، کمک پردازنده‌ی خود را توسط برنامه‌ای که تمام دستورالعمل‌های نام برده در آن وجود داشته باشد، بیازمایید. پس از اتمام مراحل طراحی، با استفاده از دستورات اسمبلی برنامه‌ای بنویسید که در آن یک چند جمله‌ای از ورودی گرفته و ریشه‌ی آن را با استفاده از روش نیوتن-رافسون محاسبه و ذخیره کند.

در صورت تشخیص تقلب، نمره صفر برای تقلب دهنده و گیرنده لحاظ خواهد شد.

این پروژه را در قالب گروه‌های ۴ نفره می‌توانید انجام دهید.

⁷ IEEE Standard for Floating-Point Arithmetic," in *IEEE Std 754-2008*, vol., no., pp.1-70, 29 Aug. 2008, doi: 10.1109/IEEESTD.2008.4610935.