



طراحی کامپیوتری سیستم های دیجیتال

(سال تحصیلی ۹۸-۹۷، نیمسال اول)

پروژه پایانی: طراحی، شبیه سازی و سنتز FSM

۱-۱- طراحی پردازنده پایه

پردازنده در نظر گرفته شده از لحاظ ساختاری مشابه پردازنده معرفی شده در کتاب مانو^۱ است که در تمرین قبل با آن آشنایی پیدا کردید. پردازنده مدنظر از یک واحد کنترل و یک بخش موسوم به جریان داده تشکیل شده است که این دو بخش با واحد حافظه و بانک رجیستری در ارتباط هستند.

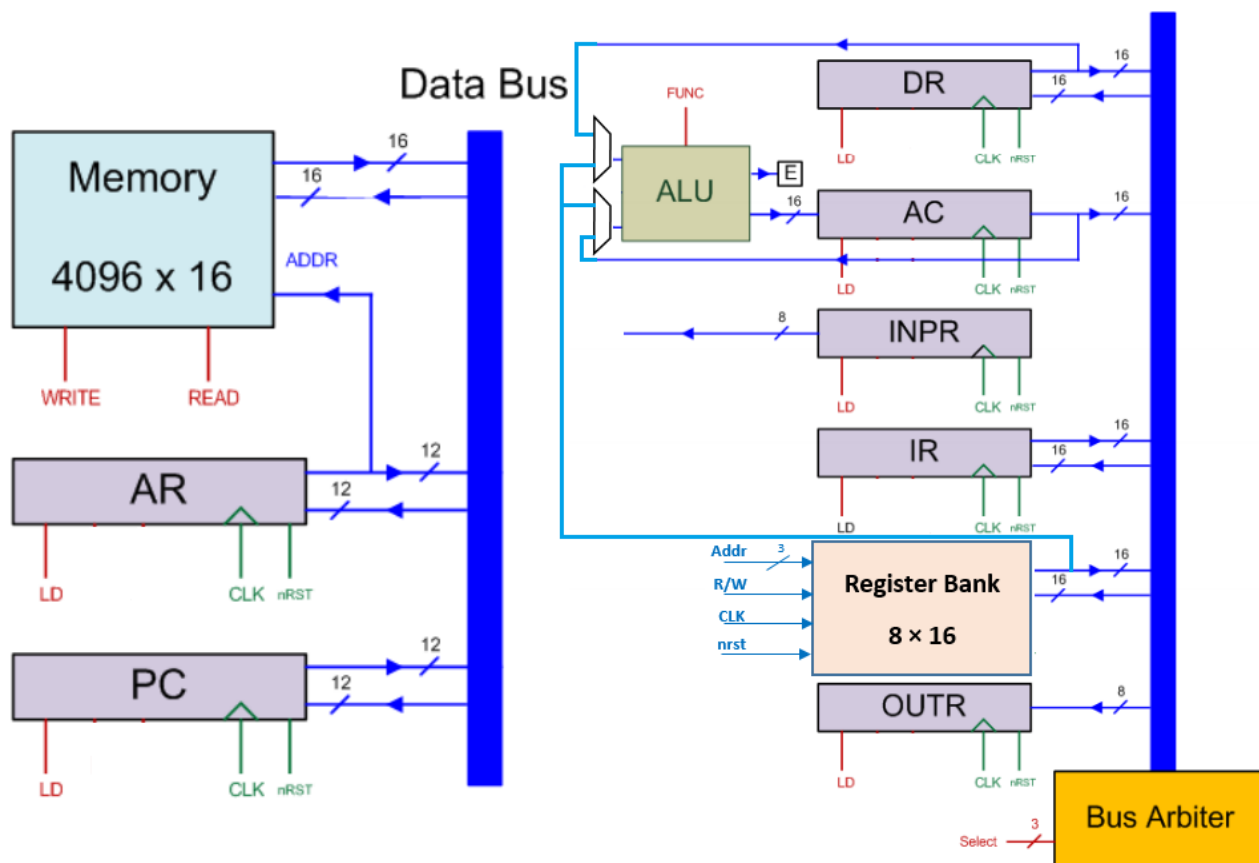
۱-۱-۱- واحد جریان داده (Data Path):

این قسمت متشکل از یک ALU ۱۶ بیتی، رجیسترها، واحد حافظه و یک گذرگاه داده می باشد که از طریق آن قسمت های مختلف به یکدیگر متصل می گردند.

واحد حساب و منطق (ALU) در این پردازنده وظیفه اجرای دستوراتی از جدول (۱-۱) را بر عهده دارد که نیاز به عملیات محسباتی و یا منطقی دارند. بنابراین تعداد دستوراتی که نیاز به اجرا در ALU دارند براساس تشخیص طراح و با توجه به دستورات موجود در جدول (۱-۱)، مشخص می شود. این واحد به یک رجیستر E که برای نگهداری مقدار نقلی خروجی در نظر گرفته شده است نیز متصل می باشد. رجیسترها در این پردازنده علاوه بر پایه های nrst و LD دارای یک پایه INR نیز می باشند که با فعال شدن آن مقدار رجیستر مربوطه یک واحد زیاد خواهد شد. برای پردازنده مورد نظر یک بانک رجیستری، متشکل از ۸ رجیستر نیز در نظر گرفته شده است که مطابق شکل (۱-۱) به دیگر واحدها متصل می گردد. بانک رجیستری دارای یک پایه R/W می باشد که صفر بودن آن به معنی خواندن از یک رجیستر است و در صورتی که نیاز به نوشتن درون یک رجیستر از بانک باشد، باید این پایه یک شود.

^۱ Mano, M. M. Computer System Architecture, 1982.

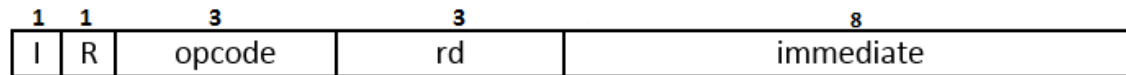
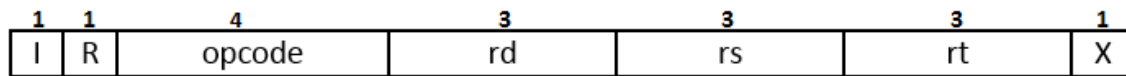
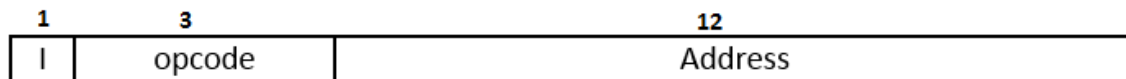
حافظه استفاده شده در این ساختار دارای ۴۰۹۶ خانه حافظه می‌باشد که از خانه ۰ تا ۲۰۴۷ مربوط به ذخیره‌سازی دستورات و از خانه ۲۰۴۸ تا ۴۰۹۵ مربوط به ذخیره داده‌هاست. خواندن و نوشتن در این حافظه با پایه‌های RE و WR کنترل می‌گردد. شکل زیر نحوه اتصال واحدها در بخش جریان داده را مشخص می‌نماید.



شکل (۱-۱) بخش‌های مختلف واحد جریان داده

۱-۲-۱- دستورات پردازنده

ساختار دستورات در این پردازنده مطابق شکل (۲-۱) می‌باشد. همانطور که مشاهده می‌شود، در این پردازنده سه نوع دستور وجود دارد که نوع اول مشابه ساختار دستورات تمرین قبل و در برگرنده بخش نخست جدول دستورات (memory instruction) می‌باشد. نوع دوم دستورات کار با رجیسترها می‌باشد، که علاوه بر opcode دارای ۳ بخش rd, rs, rt نیز هست، که مشخص کننده رجیسترهایی از بانک رجیستر است که باید روی آن‌ها عملیاتی انجام گیرد (قابل ذکر است که تمامی دستورات نوع ۲ نیاز به استفاده از این بخش‌ها ندارند). نوع سوم دستورات که شامل دو دستور می‌باشد مربوط به دستورات کار با مقادیر immediate است که علاوه بر opcode دارای دو بخش rd و immediate نیز هست. Immediate یک مقدار ۸ بیتی است که در عملیات مربوطه مورد استفاده قرار می‌گیرد. براساس شکل (۲-۱) مقدار "I" در ساختار دستورات جدا کننده دستور نوع ۱ از دستورات نوع ۲ و ۳ است که صفر بودن آن نشان دهنده دستورات نوع ۱ است. به منظور تشخیص دستورات نوع ۲ از ۳ نیز از مقدار "R" استفاده می‌شود که صفر بودن آن نشان دهنده دستورات نوع ۲ است.



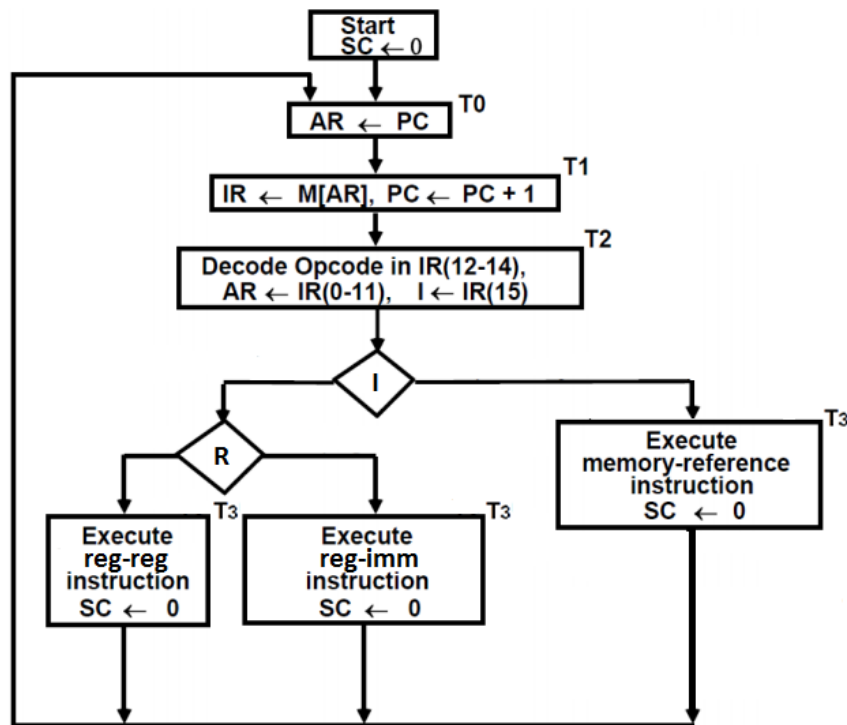
شکل (۲-۱) ساختار دستورات

جدول (۱-۱) مجموعه دستورالعملها

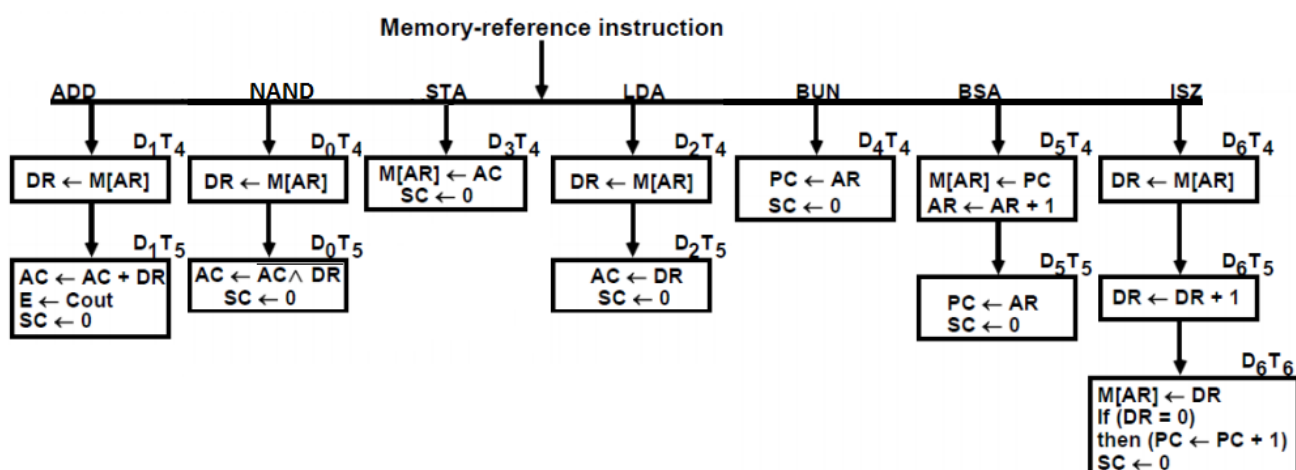
memory Instruction	I	Instr code		Description
ADD	0	000		ADD memory word to AC
NAND	0	001		NAND memory word to AC
NOT	0	010		NOT memory word to AC
LDA	0	000		Load AC from memory
STA	0	001		Store content of AC into memory
BUN	0	010		Branch unconditionally
BSA	0	011		Branch and save return address
ISZ	0	111		Increment
Register instruction	I	R	Instr code	Description
ADDR	1	0	0000	Add registers
SHR	1	0	0001	Right arithmetic shift AC
SHL	1	0	0010	Left arithmetic shift AC
CIR	1	0	0011	Circulate right AC and E
CIL	1	0	0100	Circulate left AC and E
INC	1	0	0101	Increment AC
CLA	1	0	0110	Clear AC
CMA	1	0	0111	Complement AC
CLE	1	0	1000	Clear E
CME	1	0	1001	Complement E
SZA	1	0	1010	Skip next instr. If AC is zero
SZE	1	0	1011	Skip next instr. If E is zero
INP	1	0	1100	Input Char to AC
OUT	1	0	1101	Output Char from AC
Immediate instruction	I	R	Instr code	Description
LDI	1	1	000	Load immediate to a register
ADDI	1	1	001	Add immediate

۱-۱-۳- واحد کنترل

واحد کنترل در یک پردازنده متشکل از یک ماشین حالت می‌باشد که در هر حالت از آن سیگنال‌های کنترلی خاصی به منظور استفاده در قسمت جریان داده، تولید می‌گردد. ماشین حالت پردازنده مورد نظر شامل سه حالت اصلی Fetch، Decode و Execute می‌باشد که در یک حلقه در زمان‌های T_1 ، T_2 و T_3 آغاز می‌شوند. حلقه اجرای این پردازنده در شکل (۱-۳) قابل مشاهده است. همانطور که مشخص است علاوه بر سه مرحله فوق یک مرحله اولیه در زمان T_0 نیز وجود دارد. همچنین باید به این نکته توجه داشت که تعداد سیکل مورد نیاز برای انجام مرحله Execute (شروع در زمان T_3) وابسته به دستور در حال اجراست که از یک تا سه سیکل مطابق شکل (۱-۴) قابل تغییر است.



شکل (۱-۳) روال اجرای دستورات

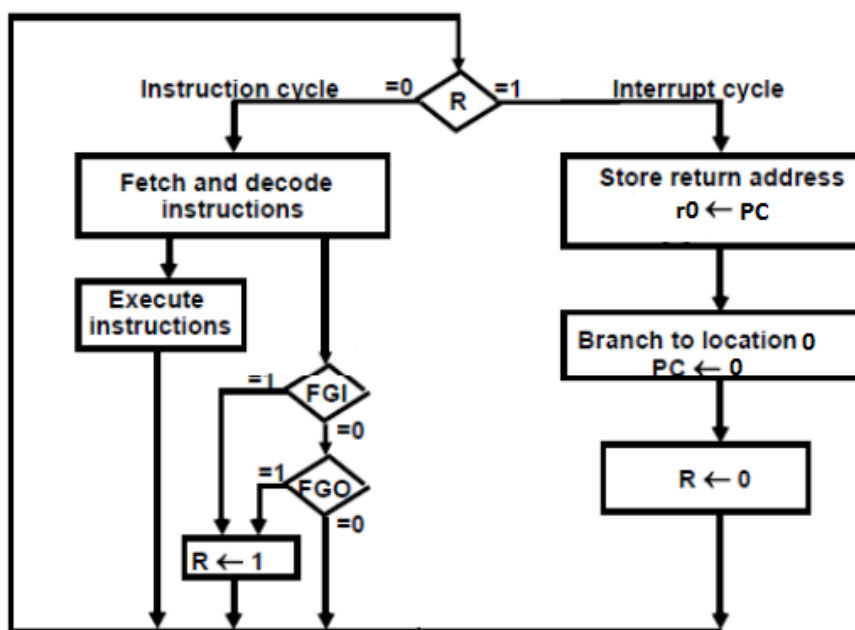


شکل (۱-۴) مرحله اجرای دستورات نوع اول

عملکرد پردازنده برای اجرای دستورات به این صورت است که واحد کنترل بایستی دو دسته سیگنال کنترلی برای واحد جریان داده تولید نماید. دسته اول سیگنال‌های زمانی T_0 تا T_6 می باشد که مشخص کننده حالت پردازنده می باشند. دسته دوم سیگنال‌های کنترلی استخراج شده از مرحله decode می باشد که مهمترین آن‌ها سیگنال‌های مشخص کننده دستورات پردازنده می باشند. در هر سیکل کامل اجرا، یکی از این سیگنال‌ها که مطابق با دستور بارگذاری شده درون پردازنده است فعال می گردد. پس از تولید این سیگنال‌ها واحد جریان داده براساس مقادیر آن‌ها تصمیم به فعال سازی سیگنال‌های کنترلی حافظه و رجیسترها و همچنین مقدار دهی ورودی‌های ALU و کنترل کننده گذرگاه داده می گیرد.

۱-۱-۴- وقفه

روال اجرای فوق، روال عادی اجرای یک دستور در پردازنده است. اگر در ابتدای سیکل اجرا پردازنده تشخیص وقفه دهد باید روال اجرای دستورات را تغییر داده و روتین مربوط به وقفه را اجرا نماید. نحوه تشخیص وقفه توسط پردازنده در شکل (۵-۱) نشان داده شده است. همانطور که مشاهده می کنیم، به منظور کنترل عملیات وقفه نیاز به بررسی مقادیر موجود در سه فلیپ فلاپ R ، FGI و FGO می باشد. هنگامیکه یک وسیله خارجی قصد ارسال داده‌ای به پردازنده را داشت، باید مقدار FGI را برابر "1" کرده و داده مورد نظر خود را در رجیستر $INPR$ قرار دهد و برعکس اگر پردازنده قصد ارسال داده‌ای به خارج را داشته باشد، باید مقدار FGO را بررسی کند، اگر مقدار FGO برابر "1" بود، داده مورد نظر را در رجیستر $OUTR$ قرار می دهد. براساس این توضیحات یک شدن مقادیر FGI و FGO همواره توسط وسایل خارجی و صفر شدن آن‌ها توسط پردازنده انجام می شود. مطابق شکل (۵-۱) پس از مرحله Decode، اگر هر کدام از مقادیر FGI و یا FGO برابر یک باشد مقدار R برابر یک می شود و در سیکل بعدی اجرا روال سرویس دهی به وقفه اجرا می شود.



شکل (۵-۱) روال انجام وقفه

روال سرویس دهی به وقفه در این پردازنده به این شکل است که، ابتدا مقدار موجود در PC (return address) درون رجیستر $r0$ موجود در بانک رجیستری ذخیره می گردد و سپس مقدار PC برابر صفر قرار داده می شود. مکان صفر حافظه در این پردازنده همواره حاوی یک دستور پرش (BUN) به محل ذخیره روتین پاسخ دهی به وقفه است که در نتیجه موجب می شود در سیکل بعدی اجرا، این روتین از ابتدا شروع به اجرا شود. آخرین دستور از روتین پاسخ دهی به وقفه همواره باید آدرس ذخیره شده در رجیستر $r0$ را به درون PC منتقل کند تا از سیکل بعدی روال عادی دستورات پیگیری شود.

۱-۱-۵- نحوه اجرای دستورات

نحوه اجرای دستورات حافظه‌ای در شکل (۴-۱) نشان داده شد. در جدول (۲-۱) نحوه اجرای سایر دستورات قابل مشاهده است. همانطور که مشخص است دستورات بدون عملوند همگی عملیات خود را بر روی رجیستر AC انجام می‌دهند.

جدول (۲-۱) نحوه اجرای دستورات نوع دوم و سوم

Instruction	Symbolic Description
ADDR	$rd \leftarrow rs + rt$
SHR	$AC \leftarrow shr\ AC$
SHL	$AC \leftarrow shl\ AC$
CIR	$AC \leftarrow shr\ AC, AC(15) \leftarrow E, E \leftarrow AC(0)$
CIL	$AC \leftarrow shl\ AC, AC(0) \leftarrow E, E \leftarrow AC(15)$
INC	$AC \leftarrow AC + 1$
CLA	$AC \leftarrow 0$
CMA	$AC \leftarrow AC'$
CLE	$E \leftarrow 0$
CME	$E \leftarrow E'$
SZA	If $(AC = 0)$ then $(PC \leftarrow PC + 1)$
SZE	If $(E = 0)$ then $(PC \leftarrow PC + 1)$
INP	$AC(7-0) \leftarrow INPR, FGI \leftarrow 0$
OUT	$OUTR \leftarrow AC(7-0), FGO \leftarrow 0$
LDI	$rd \leftarrow imm$
ADDI	$rd \leftarrow rd + imm$

۱-۲- شبیه‌سازی

به منظور شبیه‌سازی طرح، نیاز است که دستورات مربوط به الگوریتم ضرب دو عدد ۸ بیتی با استفاده از جمع و شیف، درون حافظه قرار گیرد. بنابراین درستی طرح با مشخص کردن دو عدد ۸ بیتی به عنوان عملوندهای ضرب و سنجش مقدار نهایی ضرب که باید درون رجیستر ۲۷ قرار گیرد انجام خواهد شد.

۱-۳- سنتز

نیاز است طرح خود را با استفاده از یکی از ابزارهای سنتز شرکت Xilinx سنتز نموده و گزارش‌های سنتز طرح را اعلام نمایید.

۴-۱- بخش‌های امتیازی

در این قسمت سه بخش به عنوان بخش‌های امتیازی در نظر گرفته شده است که انجام هر بخش به میزان ۱۰٪ از نمره نهایی پروژه، نمره اضافی خواهد داشت.

۱- **شبیه‌سازی وقفه:** در حالت عادی قسمت مربوط به وقفه باید پیاده‌سازی شود اما نیازی به شبیه‌سازی برای این قسمت نیست. به منظور دریافت نمره امتیازی این قسمت باید روالی را برای شبیه‌سازی و اثبات درستی روال سرویس‌دهی به وقفه در نظر بگیرید.

۲- **خواندن از فایل:** نوشتن کد باینری دستورات در یک فایل و انتقال دستورات از فایل به قسمت مربوط به دستورات درون حافظه به جای مقدار دهی مستقیم حافظه.

۳- **شبیه‌سازی با استفاده از حلقه:** به منظور نوشتن برنامه ضرب دو عدد با استفاده از الگوریتم جمع و شیفت نیاز است تا دستورات مربوط به هر مرحله از الگوریتم، یک به یک درون حافظه قرار گیرند. در صورتیکه روال اجرای این الگوریتم با کمک یک حلقه که با استفاده از دستورات همین پردازنده نوشته شده است، انجام شود، نمره امتیازی محاسبه خواهد شد.

۵-۱- نکات فنی:

- ساختار پیاده‌سازی شده باید شامل واحدهای مجزای رجیستر، حافظه، گذرگاه داده باشد که همگی در یک واحد بزرگتر (top module) نمونه‌گیری شده باشند. top module برنامه باید شامل قسمت‌های نمونه‌گیری شده به همراه processهای مجزا مربوط به قسمت‌های واحد کنترل و واحد جریان داده باشد. با توجه به ساختار ماشین حالت واحد کنترل، مشخص است که process مربوط به این قسمت خود به دو یا سه process مجزا تقسیم می‌گردد. همانطور که از قبل می‌دانید در این ساختار تنها process ترتیبی واحد کنترل دارای ورودی کلاک در لیست حساسیت خود می‌باشد.
- مقادیر داده‌ها در پردازنده باید از نوع مکمل دو علامت‌دار در نظر گرفته شوند.
- نام سیگنال‌ها دقیقاً به همان صورتی که صورت تمرین تعریف شده است، استفاده شود.
- در صورت وجود حالتی که در صورت پروژه به آن اشاره نشده است، یک فرض مناسب برای آن حالت در نظر بگیرید. توضیحات مربوط به فرضیات موجود در طرح را در گزارش خود ذکر نمایید.
- گزارش طرح باید شامل توضیح کامل کد ها و تصاویر مربوط به شکل موج‌های خروجی حاصل از مرحله شبیه‌سازی باشد.

موفق باشید