



دانشگاه تهران - دانشکده‌ی فنی
دانشکده‌ی مهندسی برق و کامپیوتر

آزمون میان‌ترم درس معماری کامپیوتر

نیم‌سال دوم سال تحصیلی ۹۹-۰۰

مدرس : سعید صفری

تاریخ آزمون : ۲۵ اردیبهشت ۱۴۰۰

مدت زمان آزمون : ۷۲ ساعت

مشخصات اعضای گروه:

نام و نام خانوادگی:

شماره‌ی دانشجویی:

نام و نام خانوادگی:

شماره‌ی دانشجویی:

هدف از این آزمون طراحی و پیاده‌سازی چندمرحله‌ی (Muti-Cycle) یک پردازنده مبتنی بر استک است. این پردازنده دارای یک باس داده‌ی ۸ بیتی و باس آدرس ۵ بیتی است. حافظه دارای گنجایش 32×8 بیت (۳۲ کلمه‌ی ۸ بیتی) است، دستورات ۸ بیتی هستند و تمام opcode ها ۳ بیت طول دارند. پردازنده دارای یک استک (در داخل پردازنده) است که کلیه‌ی عملیات پردازنده مبتنی بر آن انجام می‌شود. به این صورت که مثلاً اجرای دستور add باعث می‌شود دو اپرند مورد نیاز ابتدا از روی استک داخلی pop شده و حاصل جمع به دست آمده بر روی استک push شود. جدول زیر دستورات این پردازنده را به همراه Opcode مربوطه نشان می‌دهد:

Instruction Mnemonic and Definition		Bits 7:5
ADD	-	000
SUB	-	001
AND	-	010
NOT	-	011
PUSH	Load Address	100
POP	Store Address	101
JMP	Jump Addressed	110
JZ	Load Direction	111

دستورات این پردازنده به سه دسته تقسیم شده است:

دستورات محاسباتی منطقی:

- ADD : محتویات دو خانه‌ی بالای استک pop شده حاصل جمع آن‌ها مجدداً روی استک push می‌شود.
- SUB : محتویات دو خانه‌ی بالای استک pop شده حاصل تفریق آن‌ها مجدداً روی استک push می‌شود.
- AND : محتویات دو خانه‌ی بالای استک pop شده AND آن‌ها مجدداً روی استک push می‌شود.
- NOT : محتویات خانه‌ی بالای استک pop شده NOT آن مجدداً روی استک push می‌شود.

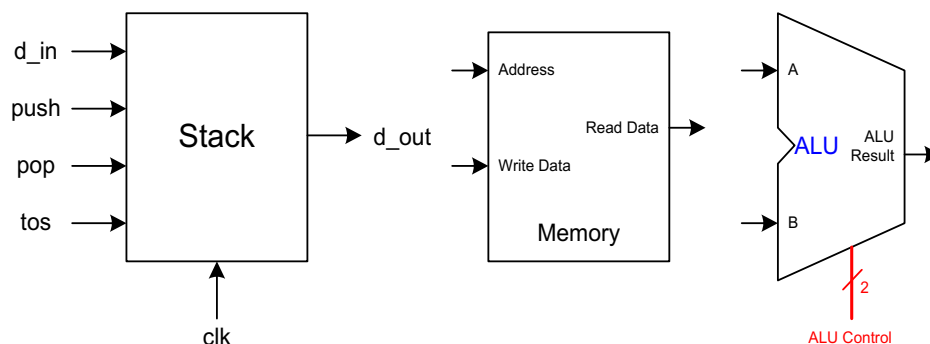
دستورات دسترسی به حافظه:

- PUSH : این دستور محتویات خانه‌ای از حافظه که توسط فیلد آدرس دستور مشخص شده است را روی استک push می‌کند.
- POP : این دستور محتویات خانه‌ی بالای استک را در خانه‌ای از حافظه که توسط فیلد آدرس دستور مشخص شده است pop می‌کند.

دستورات پرش:

- JMP : این دستور به خانه‌ای از حافظه که توسط فیلد آدرس دستور مشخص شده است پرش می‌کند.
- JZ : این دستور در صورتی که محتویات بالای استک صفر باشد، به خانه‌ای از حافظه که توسط فیلد آدرس دستور مشخص شده است پرش می‌کند.

برای طراحی این پردازنده component های اصلی زیر را در اختیار داریم.



نکته‌ی ۱: توجه داشته باشید که استک در داخل پردازنده قرار دارد و نقش رجیسترفایل را بازی می‌کند.

نکته‌ی ۲: ورودی‌های استک همه سنکرون با کلاک هستند. استک سه ورودی کنترلی دارد:

- **ورودی push:** در صورت فعال بودن این ورودی، با لبه‌ی بالارونده‌ی کلاک مقدار ۸ بیتی ورودی d_{in} در داخل استک push می‌شود و اشاره‌گر استک آپدیت می‌شود.

- **ورودی pop:** در صورت فعال بودن این ورودی، با لبه‌ی بالارونده‌ی کلاک مقدار بالای استک از روی استک pop می‌شود و روی خروجی ۸ بیتی d_out قرار می‌گیرد و اشاره‌گر استک آپدیت می‌شود.
- **ورودی tos:** در صورت فعال بودن این ورودی، با لبه‌ی بالارونده‌ی کلاک مقدار بالای استک روی خروجی ۸ بیتی d_out قرار می‌گیرد.

نکته‌ی ۳: برای طراحی این پردازنده می‌توانید از هر تعداد رجیستر یا مالتی‌پلکسر که نیاز دارید استفاده کنید.

مسیر داده

واحد كنترول