



در این تمرین از شما خواسته شده است که یک پردازنده مبتنی بر استک را به صورت Multi-Cycle طراحی کنید. این پردازنده دارای یک باس داده‌ی ۸ بیتی و باس آدرس ۵ بیتی است. حافظه دارای گنجایش 8×32 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.

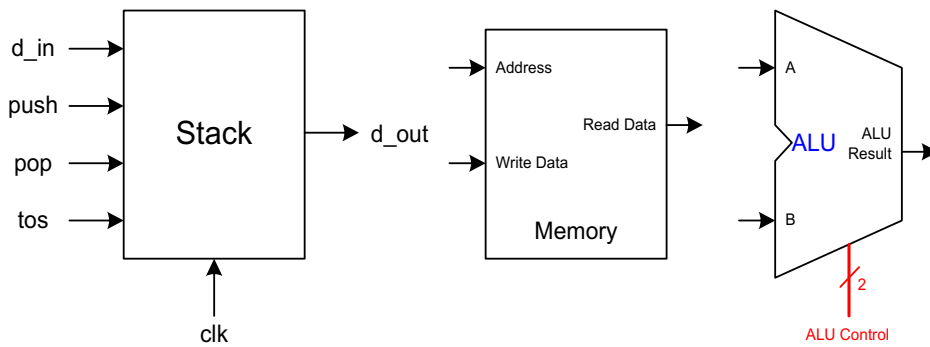
دستورات دسترسی به حافظه:

- 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.



مسیر داده و واحد کنترل این پردازنده را به صورت چند مرحله‌ای طراحی کنید. برای تست پردازنده‌ای که طراحی کردید، ابتدا برنامه‌ی زیر را به زبان اسمبلی بنویسید، سپس آن را به زبان ماشین تبدیل کنید و آن را در آدرس ۰ حافظه‌ی دستور لود کرده و اجرا کنید. بدیهی است که باید در حافظه‌ی داده، داده‌های مناسب را قرار دهید تا نتیجه اجرای برنامه شما مشخص شود.

برنامه‌ی تست: برنامه‌ای بنویسید که حاصل جمع عناصر یک آرایه‌ی ۵ عنصری با آدرس شروع ۲۵ را به دست آورد.