



دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

مشخصات پروژه

عنوان درس : طراحی و معماری کامپیوتر

مستندات پروژه ۲

(assembler simulator)

بهنوش بهیانی

استاد : دکتر رضایی

خرداد ماه ۱۴۰۱

شرح پروژه :

در این پروژه قرار است دستوراتی که در پروژه ی اول به کد ماشین تبدیل کرده ایم را اجرا کرده و بعد از اجرای هر دستور محتویات رجیستر ها و حافظه را به کاربر نمایش دهیم. در حقیقت قرار است برنامه ای بسازیم تا مانند QtSpim عمل کند.

توابع و کلاس ها:

Struct inst (۱)

این کلاس وظیفه نگه داری نام، نوع، opcode ، شماره ی رجیستر های مبدا و مقصد و همچنین مقدار offset ، کد ماشین و کد ماشین هگز را نگهداری میکند.

Struct registers (۲)

محتویات داخل هر رجیستر و وضعیت استفاده از آن رجیستر با کمک این کلاس نگه داری میشود.

Struct memory (۳)

آدرس و محتویات هر آدرس حافظه با کمک این ساختار میتوان نگه داری کرد.

Func dec_to_hex() (۴)

این تابع کد ماشین را به فرم هگز آن تبدیل میکند که برای تعیین نوع دستور و رجیستر های مبدا و مقصد آن راحت تر باشیم.

Func hex_to_dec() (۵)

این تابع وظیفه دارد مقادیر offset و imm و را به مقدار دسیمال آن تبدیل میکند.

Func main() (۶)

- ابتدا فایل ورودی یا همان mc. را میخوانیم.
- سپس مقادیر را در آرایه ای از دستور العمل ها نگه داری میکنیم.
- یک حلقه به اندازه ی تعداد لاین ها تعریف میکنیم و هر دستور را decode کرده و مقادیر Hex[] , offset , rs , rd , name , type , opcode را بدست می آوریم.

- بعد از اینکه تمام دستورات را دیکود کردیم دستورات و آدرس آنها (شماره خط) را در بخش حافظه هم اضافه میکنیم .
- حال در بخش اصلی از $pc=0$ دستورات را شروع به خواندن و اجرایی کردن میکنیم و توجه داشته باشید که باید یک آرایه ۱۶ تایی از کلاس رجیستر ها نیز قبلا تعریف کرده باشیم و مقادیر اولیه تمام رجیستر ها به غیر از رجیستر صفر را منفی یک بدهیم. هر دستور را که اجرا کردیم گاهی مقادیر رجیستر ها و گاهی مقدار pc تغییر میکند که باید در هر مرحله از اجرا آنها را نمایش دهیم و بدین ترتیب دستورات را تارسیدن به دستور `halt` اجرا میکنیم.