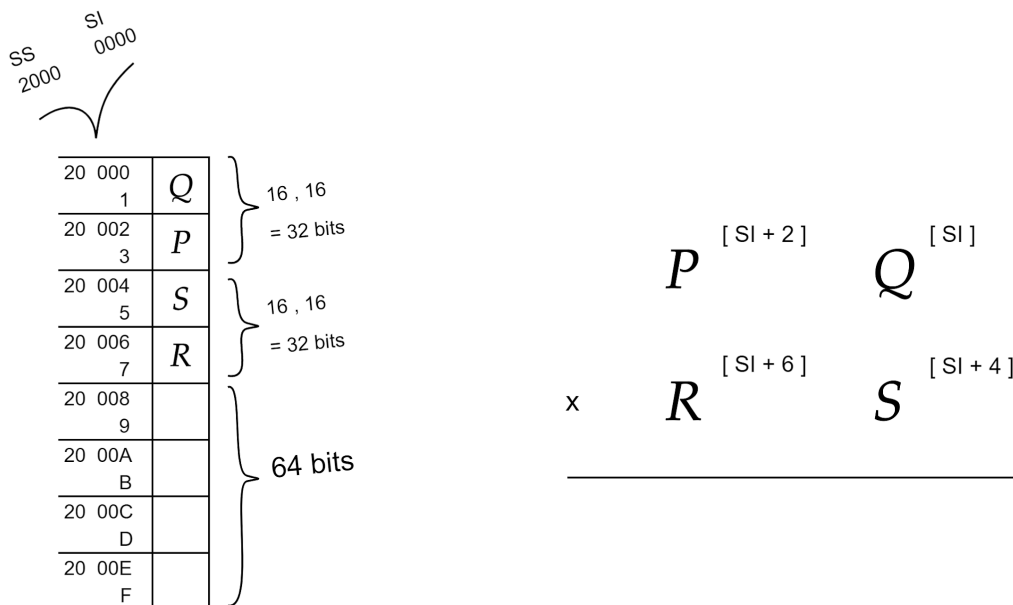


گزارش پیاده‌سازی تمرین دوم

متین زیودار - 97243037

سوال 1

در این سوال همانند ضرب دوران دبیرستان تنها با کمی تفاوت عمل می‌کنیم.



1. S را در Q ضرب می‌کنیم، سپس در خانه اول و دوم می‌نویسیم (زیرا حاصل ضرب، 2 دابل‌ورد است).
2. S را در P ضرب می‌کنیم و در خانه دوم و سوم می‌نویسیم (با مقادیر قبلی همراه با carry جمع می‌کنیم).
3. R را در Q ضرب می‌کنیم و در خانه دوم و سوم می‌نویسیم (با مقادیر قبلی همراه با carry جمع می‌کنیم).
4. R را در P ضرب می‌کنیم و در خانه سوم و چهارم می‌نویسیم (با مقادیر قبلی همراه با carry جمع می‌کنیم).

برای اجرای برنامه می‌بایستی درون کد دو عدد را درون data segment در NUMS نوشته و سپس پس از اجرای برنامه، مقادیر را در خانه RES مشاهده کنیم. (ذخیره‌سازی همانند تمامی پردازنده‌های 8086 به صورت little-endian در نظر گرفته شده‌است)

سوال 2

در این سوال عدد N از طریق کنسول دریافت می‌شود و سپس اعداد با ایندکس چاپ می‌شوند.

طریقه نوشتن بدین صورت بود که ابتدا کد را در جاوا نوشته و سپس آن را به اسمبلی تبدیل کردیم. از عدد 2 شروع می‌کنیم و بررسی می‌کنیم که آیا N بر 2 بخش پذیر است یا نه. به همین ترتیب اعداد را افزایش می‌دهیم تا به خود عدد برسیم. (از آنجایی که رادیکال بلد نبودم، حلقه بررسی باقی‌مانده تا خود عدد ادامه می‌دهد. برای همین بهینه نیست)

سوال 3

بخش الف)

در این سوال عدد BCD از طریق کنسول دریافت می‌شود. این تابع از example های emu8086 برداشته شده است. طریقه عملکرد بدین گونه است که پس از گرفتن هر کاراکتر ابتدا آن را چاپ می‌کند و سپس چک می‌کند که بین 0 تا 9 باشد، سپس عدد قبلی را ضربدر 10 می‌کند و با مقدار عددی کاراکتر جمع می‌کند (زیرا عدد به صورت ascii ورودی گرفته شده بود). اگر overflow رخ دهد، رقم آخر وارد شده را پاک می‌کند. تا زمانی که کاربر enter وارد کند ادامه پیدا می‌کند. کاربر می‌تواند با فشردن backspace آخرین رقم خود را نیز پاک کند که روشش برعکس اضافه کردن رقم است. عدد Binary نیز همانند بالا است با این تفاوت که هر سری عدد قبلی ضربدر 2 می‌شود. (مبنای دو) هر دو عدد وارد شده توسط کاربر بایستی نهایتاً یک رجیستر فضا اشغال کنند. (16 بیت) در نهایت توسط نمونیک DIV، مقدار باقی‌مانده محاسبه می‌گردد.

بخش ب)

در این سوال گرفتن ورودی و محاسبه خروجی همانند سوال قبل می‌باشد با این تفاوت که عدد دوم نیز BCD در نظر گرفته شده است.

سوال 4

ابتدا از کاربر در کنسول تعداد خانه‌های آرایه پرسیده می‌شود (حداکثر 100 مقدار فضا در حافظه در نظر گرفته شده است که قابل تغییر است و اگر کاربر بیشتر از 100 هم بخواهد وارد کند، از آنجایی که بعد از آرایه فضای دیگری گرفته نشده است به مشکل نمی‌خوریم باز) سپس در هر خط به صورت BCD مقادیر آرایه ورودی گرفته می‌شود و نوشته می‌شود در حافظه. پس از گرفتن تمام ورودی‌ها از خانه اول آرایه شروع می‌کنیم و خانه به خانه جلو می‌رویم و اگر عدد فرد بود، آن را به مجموع اضافه می‌کنیم. در نهایت مجموع را چاپ می‌کنیم.

سوال 5

در این سوال عدد N از طریق کنسول دریافت می‌شود. پاس کردن پارامتر با استفاده از رجیستر AX صورت می‌گیرد. در هر مرحله برای آنکه مقدار i را نگه داریم که در $factorial(i - 1)$ ضرب کنیم، مقادیر رجیستر AX را روی استک پوش می‌کنیم و سپس پس از فراخوانی بازگشتی، آن را پاپ می‌کنیم.