گزارش تحلیلی­ دستورکار دهم.

امیرحسین ادواری ۹۸۲۴۳۰۰۴ – زهرا حیدری ۹۸۲۴۳۰۲۰

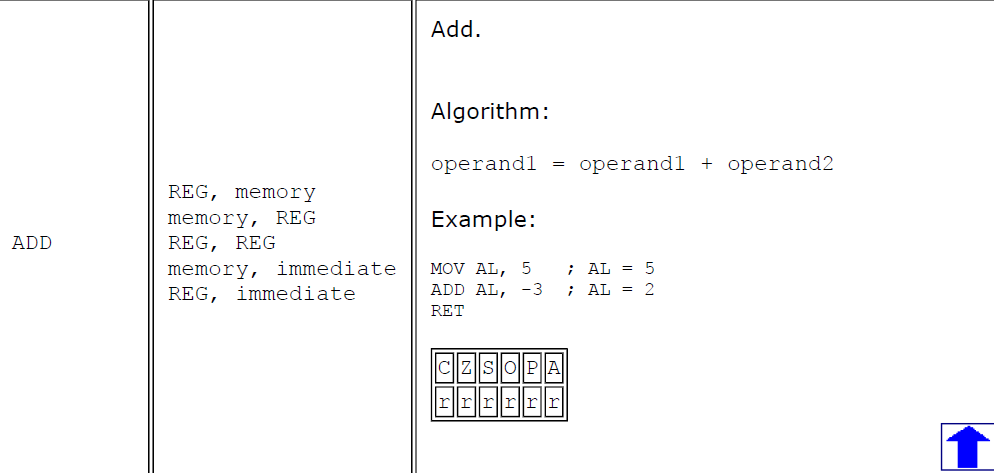
سوال اول)‌

سوال دوم)

الف) در این دستور میان عملوند اول و دوم ویرگول (کاما)‌ قرار ندارد.

ب) در این دستور تلاش شده تا نیمه­بالای CX در خود CX‌ قرار گیرد، جدا از آنکه به­طور منطقی ایراد دارد، ریختن مقدار ۸ بیتی در ۱۶ بیتی مولد خطاست.

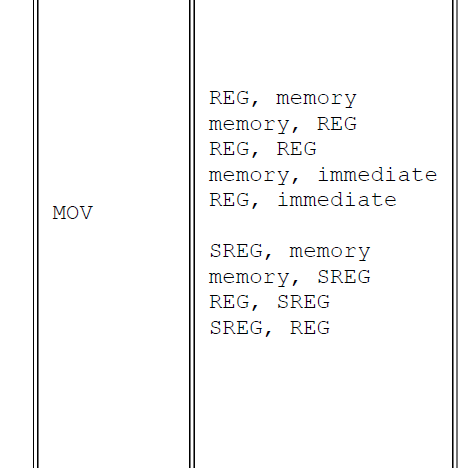
پ) در این دستور عملوند مقصد یک عدد (immediate) است (dest یک لیترال عددی است) و مشخصا جنبه حافظه­ای ندارد.



در این رفرنس مربوط به دستورات اسمبلی نیز این حالت برای این دستور وجود ندارد. (عملوند اول یا بایستی حافظه باشد و یا رجیستر)‌

ت) این دستور صرفا یک عملوند میپذیرد!‌ کاراکتر نقطه در انتهای دستور نیز اضافی­است و مولد خطا.

ث) در این دستور نیز مثل مورد پ عملوند اول نمی­تواند immediate باشد (جنبه حافظه­ای ندارد) شکل زیر حالات مجاز استفاده از دستور را نشان می­دهد.



ج) دستوری به­نام MOVE در اسمبلی 8086 وجود ندارد.

چ) این دستور نیز طبق تصویری که در مورد پ آمده است (و حالات مختلف عملوندهای دستور جمع را نشان می­دهد). نامعتبر است، عملوند اول نمی­تواند immediate باشد.

ح) در این دستور سعی شده تا مقدار ۸ بیتی در ۱۶ بیت ریخته شود که مولد خطاست.

خ) معکوس حالت قبل، در این دستور سعی شده یک مقدار (رجیستر) ۱۶ بیتی در ۸ بیتی ریخته شود که واضحا مولد خطاست.

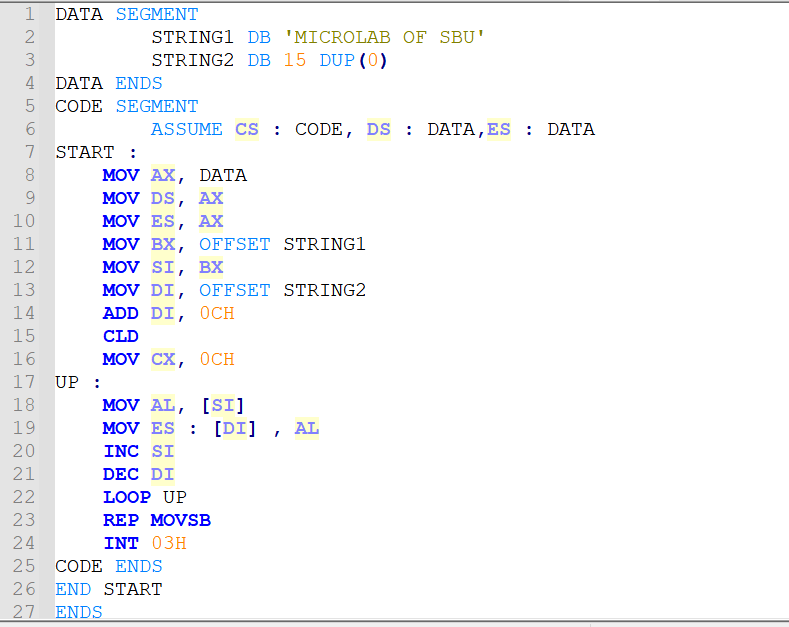
د) در این دستور سعی شده که یک مقدار ۸ بیتی‌ (AL) با یک immediate ۱۶ بیتی جمع شود که یک خطاست و این امکان موجود نیست.

ذ) طبق تصویر مورد ث که حالت مختلف عملوندها را در دستور MOV نشان می­دهد، این دستور نامعتبر است چراکه عملوند اول نمیتواند immediate باشد.

ر) عملوند اول دستور IN صرفا می­تواند AX یا AL باشد.

سوال سوم)‌

این برنامه به­طور کلی ۱۲ کاراکتر اول رشته STRING1 را به صورت معکوس در رشته STRING2 قرار می­دهد.



1. در ابتدا مقادیر ES و DS با آدرس Data مقدار دهی می­شوند. (خطوط ۹ و ۱۰)
2. بعد آدرس شروع STRING1‌در BX و پس از آن BX در SI قرار می­گیرد (خطوط ۱۱و ۱۲)
3. مشابه مرحله قبل مقدار STRING2 در DI قرار میگیرد، سپس DI با مقدار ۱۲ جمع می­شود تا به ۱۲امین بایت در STRING2 اشاره کند (قرار است کاراکتر اول STRING1 در کاراکتر ۱۲ام STRING2 قرار گیرد و با تکرار این روال رشته معکوس شود)‌ (خطوط ۱۳و۱۴)
4. سپس direction flag در خط ۱۵ام clear‌ می­شود (DF=0)
5. بعد در مقدار CX که عملا گام حلقه است، مقدار ۱۲ قرار داده می­شود (خط ۱۶)
6. در این مرحله در یک حلقه مرتبا مقدار بایتی که SI در حال حاضر به آن اشاره می­کند را در AL قرار داده (خط۱۸) و سپس این AL را در بایتی در حافظه که DI در حال حاضر به آن اشاره می­کند، می­نویسیم (خط ۱۹)‌
7. سپس در انتهای حلقه مقدار SI که عملا اشاره­گر به STRING1 است را increment‌ کرده تا به کاراکتر بعدی اشاره کند (خط ۲۰) و نیز مقدار DI‌ که عملا اشاره­گر به STRING2‌ است را decrement‌ کرده تا به بایت پیش از بایت فعلی اشاره­کند (طبق کارکردی که شرح داده شد، بایتهای STRING1 از ابتدا به ترتیب از انتها در STRING2 کپی می­شوند)‌
8. دستور LOOP نیز کارکرد حلقه را فراهم می­کند. در هرمرحله CX را یک واحد کم کرده و به ابتدای حلقه بازمی­گردد. (خط ۲۲)
9. این دستور به تعداد CX دستوران MOV را تکرار کرده و مقادیر SI و DI را با توجه به اینکه DF=1 است increment میکند.