

**مستند پروژه ریز پردازنده**

**عنوان پروژه:مدار تایمر دیجیتال**

**استاد مربوطه:  
خانم دکتر قزوینی**

**اعضا گروه:**

**حسنیه حقیقی**

**فاطمه السادات گلستانی**

**هدف از انجام پروژه**

هدف از این پروژه، طراحی یک مدار تایمر دیجیتال است که بتواند زمان را به صورت دقیقه و ثانیه نمایش دهد. تایمر باید به گونه‌ای باشد که با فشار دادن یک دکمه شروع به کار کند و زمان را از 0 تا 999 نمایش دهد.

**قطعات استفاده شده در نرم افزار** Proteus

* 7-Segment Display
* کاربرد: این قطعه برای نمایش ارقام دیجیتال (اعداد 0 تا 9) و حروف محدودی استفاده می‌شود. معمولاً در تجهیزات الکترونیکی برای نمایش اعداد مانند ساعت‌ها، کیلومترشمارها و دستگاه‌های اندازه‌گیری کاربرد دارد.
* Octal D-Type Transparent Latch
* کاربرد: این تراشه یک لچ 8 بیتی است که برای ذخیره موقت داده‌ها و انتقال آنها در مدارهای دیجیتال استفاده می‌شود. به طور گسترده‌ای در میکروکنترلرها و سیستم‌های دیجیتال برای تثبیت داده‌ها به کار می‌رود.
* Intel 8086 Microprocessor
* کاربرد: این پردازنده 16 بیتی اولین عضو خانواده پردازنده‌های x86 اینتل است و برای اجرای دستورات برنامه‌ها در کامپیوترها و سیستم‌های جاسازی شده استفاده می‌شود. در طراحی‌های آموزشی و پروژه‌های اولیه بسیار رایج است.
* Programmable Peripheral Interface
* کاربرد: این تراشه برای اتصال میکروپروسسورها به دستگاه‌های جانبی استفاده می‌شود. دارای سه پورت I/O قابل برنامه‌ریزی است که می‌توانند برای ورودی یا خروجی

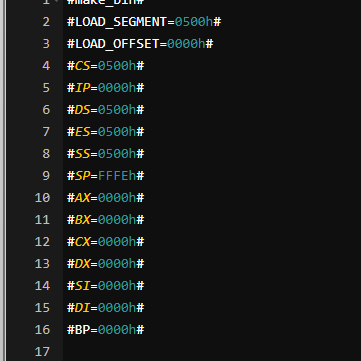
داده‌ها تنظیم شوند.

* Logic State Indicator
* کاربرد: این قطعه برای نمایش وضعیت منطقی (High یا Low) در یک مدار دیجیتال استفاده می‌شود. معمولاً در تحلیلگرهای منطقی و ابزارهای آموزشی کاربرد دارد.

8255 A

8255 مدار مجتمع واسط ورودی / خروجی برای تبادل همزمان داده ها با قابليت برنامه ریزی عملكرد می باشد. روش كار آن، موازی ناميده می شود.

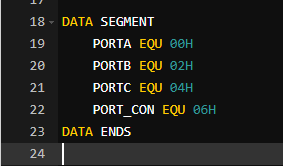
در اینجا به عنوان واسط برای اتصال خروجی های پردازنده به ورودی های سون سگمنت استفاده میکنیم.

****

**شرح کد پروژه:**

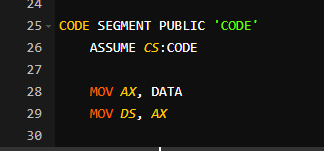
* **تنظیمات بارگذاری (Load)**

این بخش تنظیمات مربوط به بارگذاری فایل باینری در حافظه را مشخص می‌کند. این تنظیمات شامل آدرس بارگذاری، نقطه ورود و مقداردهی اولیه برای ثبات‌های مختلف است.

****

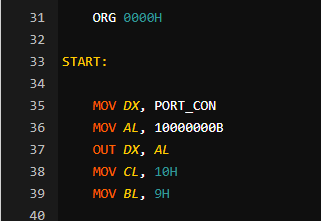
* **بخش داده‌ها (Data Segment)**

این بخش، آدرس‌های پورت‌های مختلف دستگاه 8255Aرا تعریف می‌کند. این آدرس‌ها با استفاده از دستور EQU تعیین شده‌اند.

* **بخش کد (Code Segment)**

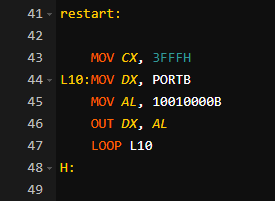
این بخش آغاز کد اجرایی است. دستور MOV برای انتقال داده بین ثبات‌ها و حافظه استفاده می‌شود.

* **تنظیم اولیه (Initialization)**

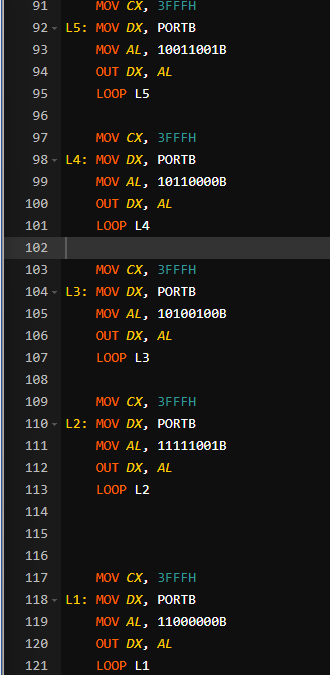
****این بخش تنظیمات اولیه پورت کنترل را انجام می‌دهد و تمام پورت‌ها را به عنوان خروجی تنظیم می‌کند. همچنین مقادیر اولیه برای ثبات‌های CL و BL تنظیم می‌شود.

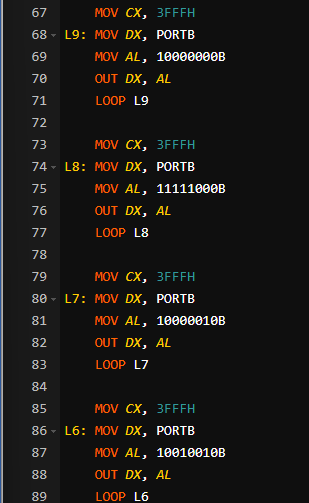
آدرس 10000000 برای مود ساده قطعه 8255 نوشته شده است.

* **شروع حلقه اصلی (Main Loop Start)**

****این بخش از کد، یک حلقه را آغاز می‌کند که مقدار BL را از 9 شروع کرده و در هر دور حلقه کاهش می‌دهد.

* **مراحل مختلف نمایش اعداد (Displaying Numbers)**

****

****

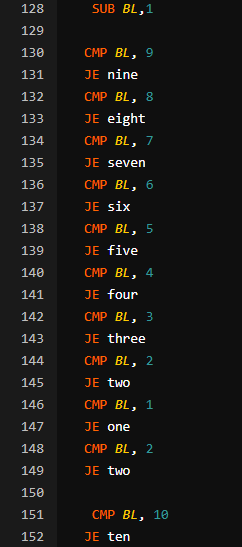
کد زیر، مراحل مختلف برای نمایش اعداد 9 تا 0روی پورت‌ها را شامل می‌شود. هر بخش از کد مقدار باینری مربوط به هر عدد را به پورت ارسال می‌کند و سپس حلقه‌ای را برای تاخیر اجرا می‌کند.

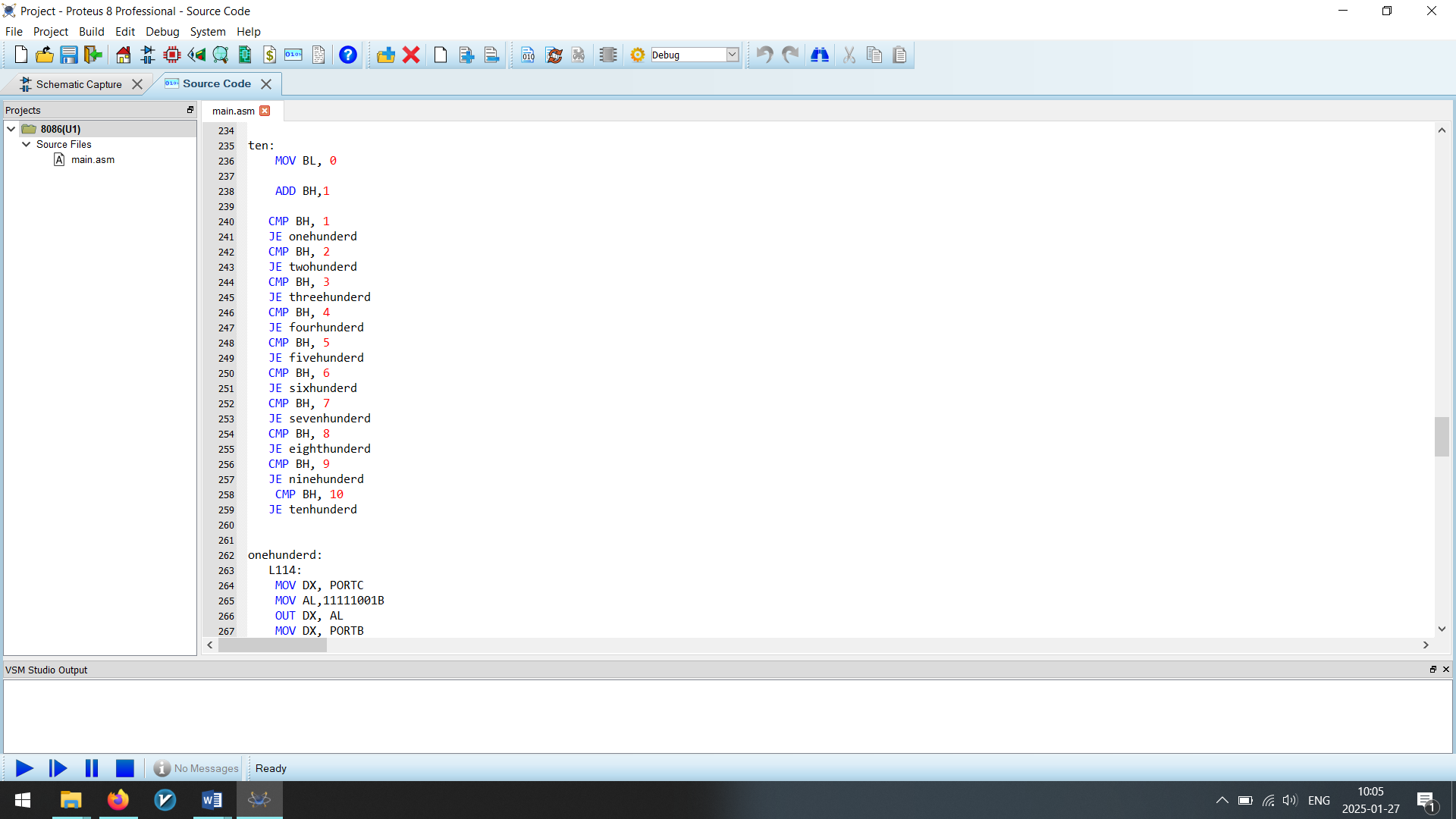
* **تغییر مقادیر (Updating Values)**

این بخش از کد، مقدار BL را افزایش می‌دهد و سپس بر اساس مقدار BL، به برچسب متناظر پرش می‌کند تا مقدار مناسب برای نمایش انتخاب شود.

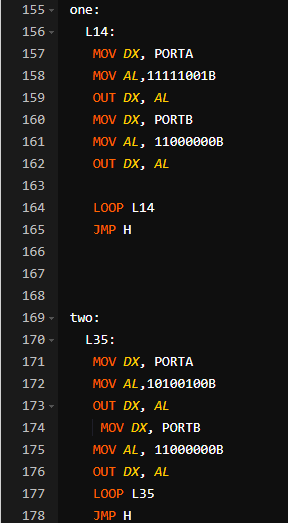
با این سری دستورات ما میتوانیم اعداد دو رقمی و سه رقمی داشته باشیم.

بعد از عدد 99 ما از BH استفاده

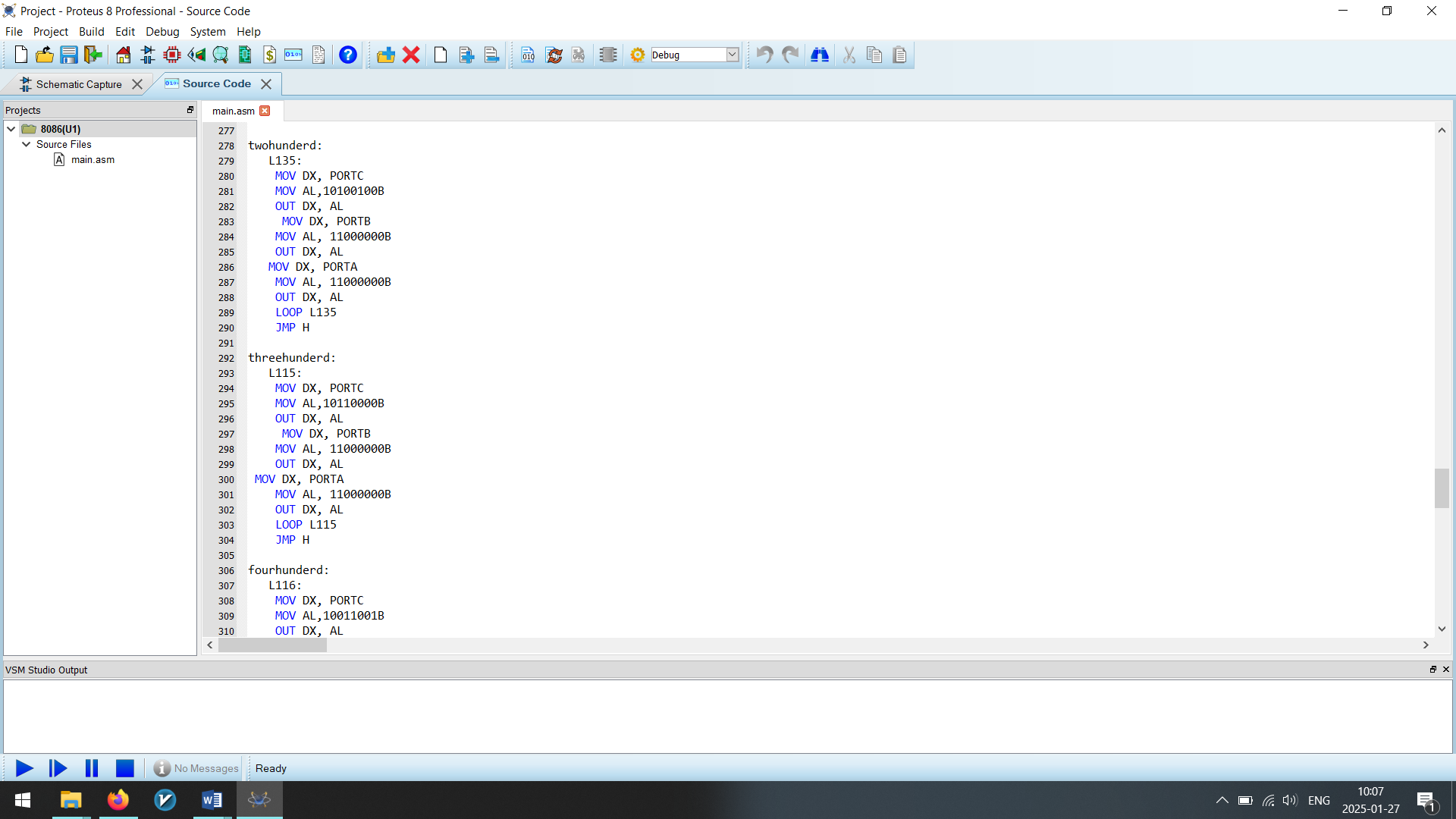
****می کنیم برای نشان دادن اعداد 3 رقمی. این بخش از کد مقدار BL را به 10 برمی‌گرداند و به شروع حلقه restart پرش می‌کند.

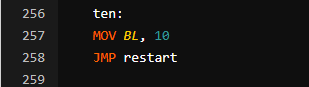
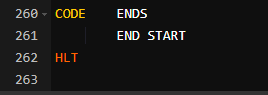


* **برچسب‌های نمایش اعداد (Number Display Labels)**

این بخش‌ها شامل کدهایی هستند که مقدار باینری متناظر هر عدد را به پورت‌های PORTA و PORTB ارسال می‌کنند.

این روند نشان داده شده ادامه پیدا می کند تا عدد99. پس از ان BL صفر خواهد شد و همین روند را برای اعداد 3 رقمی و قطعه سون سگمنت سوم تکرار خواهد شد.

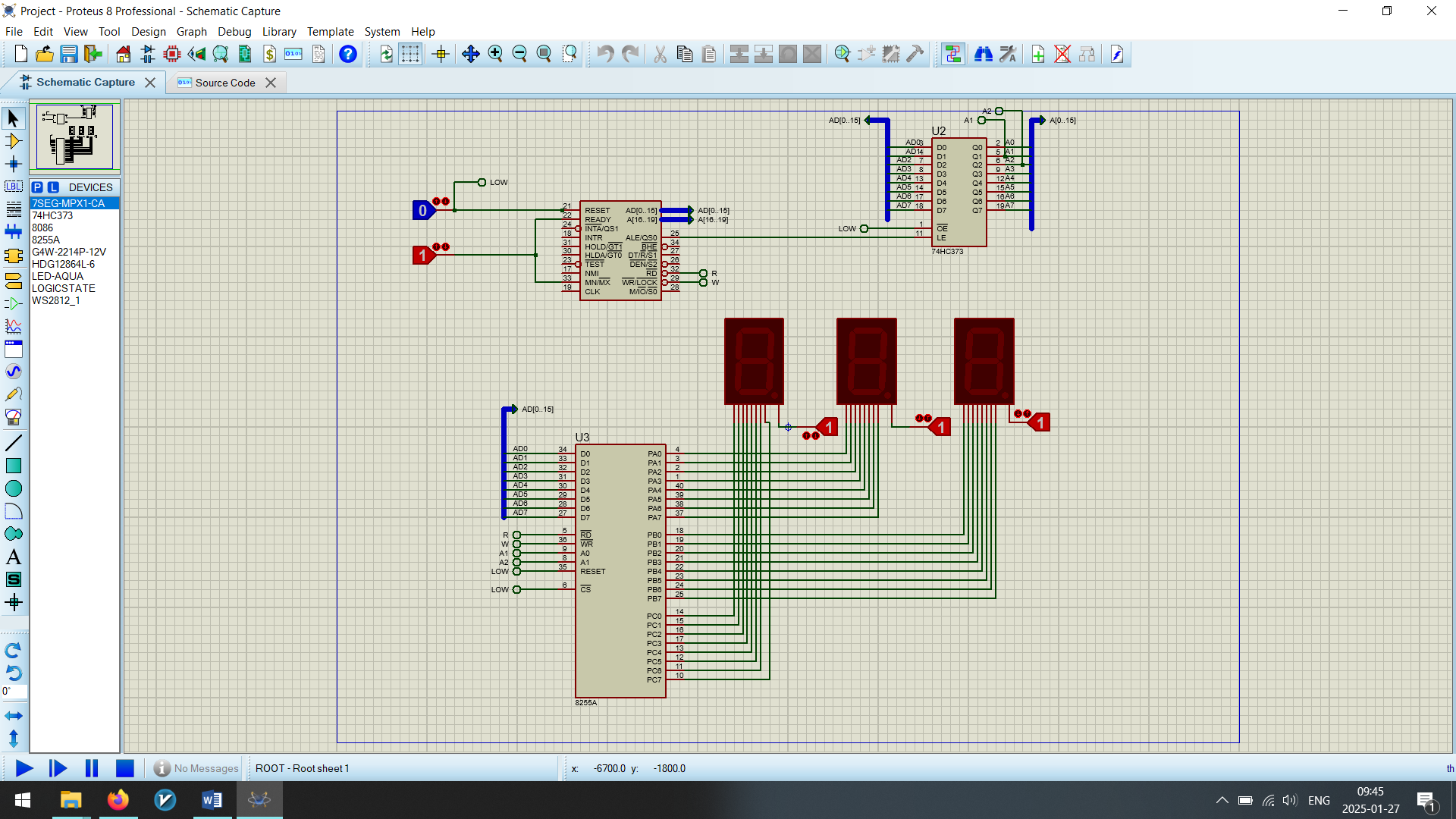


* **بازگشت به شروع (Restart)**
* **دستور توقف (Halt)**

در پایان، دستور HLT برنامه را متوقف می‌کند.

**شرح پروژه:**

برای شبیه‌سازی این کد در پروتیوس، ابتدا مدار مناسب را میسازیم که شامل میکروکنترلر 8086 یا معادل آن و پورت 8255Aباشد. سپس، پورت‌ها و کنترلرهای لازم را به درستی تنظیم می کنیم.



1. باز کردن پروتیوس:

پروتیوس را باز میکنیم و یک پروژه جدید ایجاد میکنیم.

2. اضافه کردن میکروکنترلر 8086:

به کتابخانه‌ی قطعات میرویم و میکروکنترلر 8086 را جستجو کرده و به شبیه‌سازی خود اضافه میکنیم.

3. اضافه کردن پورت8255A:

پورت 8255A را نیز از کتابخانه‌ی قطعات پیدا کرده و به شبیه‌سازی اضافه میکنیم.

4. اتصال پورت‌ها به میکروکنترلر:

پورت‌ها و کنترلرها را به صورت زیر به میکروکنترلر متصل میکنیم:

- PA (پورت A) به پورت‌های خروجی مناسب در میکروکنترلر.

- PB (پورت B) به پورت‌های خروجی مناسب در میکروکنترلر.

- PC (پورت C) به پورت‌های خروجی مناسب در میکروکنترلر.

- CONTROL به پورت کنترل مناسب در میکروکنترلر.

5. اتصال تغذیه و زمین:

همه‌ی قطعات به منبع تغذیه و زمین (GND) متصل میکنیم.

6. اضافه کردن سون سگمنت ها برای نمایش اعداد

7. پیکربندی قطعات:

هر پورت (A, B, C) و پورت کنترل 8255A را مطابق با برنامه اسمبلی پیکربندی میکنیم.

8. بارگذاری برنامه اسمبلی:

- ابتدا کد اسمبلی خود را کامپایل کرده و فایل BIN تولید میکنیم.

- در پروتیوس، روی میکروکنترلر کلیک میکنیم و فایل BIN را بارگذاری میکنیم.

9. شبیه‌سازی و مشاهده خروجی:

- شبیه‌سازی را اجرا میکنیم و مشاهده میکنیم که سون سگمنت ها به ترتیب اعداد را نمایش می‌دهند.

**عملکرد کلی مدار:**

این مدار به عنوان یک سیستم مبتنی بر پردازنده طراحی شده است که داده‌ها را دریافت کرده و آن‌ها را روی نمایشگرهای 7-segment نمایش می‌دهد. مراحل کلی عملکرد مدار به شرح زیر است:

**.1 دریافت داده از پردازنده:**

* پردازنده: پردازنده 8086 یا مشابه آن، وظیفه پردازش داده و ارسال آدرس و داده به قطعات جانبی را بر عهده دارد.
  + خطوط AD0-AD15 همزمان هم داده و هم آدرس را انتقال می‌دهند.
  + خطوط A16-A19 برای انتقال بیت‌های بالای آدرس استفاده می‌شوند.
  + سیگنال‌های RD (خواندن) و WR (نوشتن) وظیفه کنترل جهت تبادل داده‌ها را دارند.

**.2 تفکیک آدرس و داده با استفاده از (با استفاده از 74HC373):**

* از آنجا که خطوط AD0-AD15 در پردازنده به صورت دوگانه (multiplexed) استفاده می‌شوند:
  + سیگنال ALE (Address Latch Enable): برای جداسازی بخش آدرس از داده فعال می‌شود.
  + 74HC373: آدرس‌های بیت پایین (AD0-AD7) را ذخیره می‌کند و آن‌ها را به صورت ثابت (Latched) به خطوط A0-A7 ارسال می‌کند.
  + این کار باعث می‌شود که پس از آدرس‌دهی، خطوط AD0-AD15 فقط برای داده‌ها استفاده شوند.

**.3ارسال داده به A8255 :**

* پردازنده از طریق خطوط داده (AD0-AD7) و سیگنال‌های کنترلی RD)، WR، (CSبا قطعه A8255 ارتباط برقرار می‌کند.
* وظیفه A8255:
  + به عنوان یک واسط ورودی/خروجی قابل برنامه‌ریزی عمل می‌کند.
  + پردازنده می‌تواند داده‌ها را به یکی از پورت‌های A، B، یا C ارسال کند.
  + پورت‌های A، B، و C به ترتیب به سه نمایشگر 7-segment متصل هستند.

**.4 کنترل نمایشگرهای :7-Segment**

* داده‌هایی که از طریق پردازنده به پورت‌های A 8255 ارسال می‌شوند، به نمایشگرهای 7-segment فرستاده می‌شوند.
* هر نمایشگر به صورت جداگانه به یک پورت از A 8255 متصل است:
  + پورت A: نمایشگر اول (سمت راست).
  + پورت B: نمایشگر دوم (وسط).
  + پورت C: نمایشگر سوم (سمت چپ).
* داده‌ای که به هر پورت ارسال می‌شود، به صورت الکترونیکی الگوی روشن و خاموش شدن LEDهای نمایشگر را تعیین می‌کند.
  + مثلاً اگر داده 0x3Fارسال شود، عدد "0" روی نمایشگر نمایش داده می‌شود.

**.5چرخه عملیات پردازنده:**

* پردازنده با استفاده از ترکیب سیگنال‌های آدرس و داده:
  1. آدرس دستگاه مورد نظر در اینجا A) 8255( را مشخص می‌کند.
  2. داده‌ای که باید نمایش داده شود را روی خطوط داده قرار می‌دهد.
  3. سیگنال WR را فعال می‌کند تا داده به پورت مناسبA8255 نوشته شود.
* پس از ارسال داده به A8255، این داده مستقیماً به نمایشگر مربوطه منتقل می‌شود و عدد مورد نظر نمایش داده می‌شود.

****

**قطعات استفاده شده در نرم افزار Proteus**

* **.1 پردازنده (Microprocessor)**
* پردازنده‌ای که در این مدار استفاده شده است (احتمالاً 8086 یا مشابه آن) دارای پایه‌های زیر است:
* **پایه‌ها و عملکرد آن‌ها:**
* **AD0-AD15 (Address/Data)**
* این پایه‌ها به صورت دوگانه (Multiplexed) برای انتقال آدرس و داده استفاده می‌شوند.
* **در مرحله اول:** پردازنده آدرس را روی این پایه‌ها قرار می‌دهد.
* **در مرحله دوم:** این پایه‌ها برای ارسال یا دریافت داده استفاده می‌شوند.
* **A16-A19 (Address High)**
* برای انتقال بیت‌های بالایی آدرس (بیت‌های 16 تا 19) استفاده می‌شوند.
* **ALE (Address Latch Enable)**
* این سیگنال برای قفل کردن (Latch) آدرس در قطعاتی مثل 74HC373 استفاده می‌شود. این پایه تفکیک آدرس از داده‌ها را ممکن می‌سازد.
* **RD (Read)**
* فعال (LOW) می‌شود تا داده‌ها از حافظه یا یک دستگاه جانبی خوانده شوند.
* **WR (Write)**
* فعال (LOW) می‌شود تا داده‌ها به حافظه یا دستگاه جانبی نوشته شوند.
* **RESET**
* فعال (HIGH) می‌شود تا پردازنده به حالت اولیه بازگردد.
* **CLK (Clock)**
* فرکانس ساعت برای هم‌زمان‌سازی عملیات پردازنده.
* **READY**
* اگر این پایه HIGH باشد، پردازنده منتظر می‌ماند تا دستگاه خارجی آماده شود.
* **TEST**
* برای کاربردهای خاص تست استفاده می‌شود.
* **NMI (Non-Maskable Interrupt)**
* یک وقفه غیرقابل ماسک که در اولویت بالا پردازش می‌شود.
* **INTR (Interrupt Request)**
* برای درخواست وقفه قابل ماسک از پردازنده.
* **INTA (Interrupt Acknowledge)**
* پاسخ پردازنده به یک درخواست وقفه.
* **DEN (Data Enable)**
* فعال کردن خطوط داده.
* **DT/R (Data Transmit/Receive)**
* تعیین جهت تبادل داده.
* **M/IO (Memory/Input Output)**
* تعیین نوع دسترسی (حافظه یا ورودی/خروجی).
* **.2آدرس (Latch 74HC373)**
* این قطعه یک لچ 8 بیتی است که برای ذخیره بیت‌های پایین آدرس استفاده می‌شود.
* **پایه‌ها و عملکرد آن‌ها:**
* **D0-D7 (Data Inputs)**
* داده‌ها (یا آدرس‌ها) از پردازنده دریافت می‌شوند (AD0 تا AD7).
* **Q0-Q7 (Data Outputs)**
* خروجی بیت‌های latched شده (A0 تا A7).
* **LE (Latch Enable)**
* وقتی فعال (HIGH) شود، داده‌ها روی ورودی D0-D7 در خروجی Q0-Q7 قفل می‌شوند.
* **OE (Output Enable)**
* وقتی فعال (LOW) شود، خروجی‌ها Q0-Q7 فعال می‌شوند.
* **8255A (Programmable Peripheral Interface).3**
* این قطعه برای ارتباط پردازنده با دستگاه‌های جانبی استفاده می‌شود. دارای 3 پورت A، B، و C است.
* **پایه‌ها و عملکرد آن‌ها:**
* **PA0-PA7 (Port A)**
* برای ارسال یا دریافت داده از دستگاه‌های جانبی استفاده می‌شود.
* **PB0-PB7 (Port B)**
* مشابه پورت A.
* **PC0-PC7 (Port C)**
* مشابه پورت‌های A و B.
* **D0-D7 (Data Bus)**
* برای انتقال داده بین پردازنده و این قطعه.
* **RD (Read)**
* وقتی فعال (LOW) باشد، داده از یکی از پورت‌ها خوانده می‌شود.
* **WR (Write)**
* وقتی فعال (LOW) باشد، داده به یکی از پورت‌ها نوشته می‌شود.
* **A0، A1**
* برای انتخاب پورت یا رجیستر کنترل:
* **00** پورت A
* **01** پورت B
* **10** پورت C
* **11** رجیستر کنترل
* **CS (Chip Select)**
* وقتی فعال (LOW) باشد، این قطعه برای پردازنده قابل دسترسی است.
* **RESET**
* این پایه برای تنظیم مجدد (Reset) رجیسترها استفاده می‌شود.

**4. نمایشگرهای 7-segment**

* سه عدد نمایشگر 7-segment برای نمایش اعداد طراحی شده‌اند.
* **اتصال‌ها و عملکرد آن‌ها:**
* هر نمایشگر 7-segment دارای 8 پایه برای کنترل 7 LED (برای نمایش اعداد) و یک نقطه اعشاری (Decimal Point) است.
* این پایه‌ها به پورت‌های A،B و C از قطعه A8255 متصل شده‌اند.
* داده‌ای که به پورت‌های A،B یا C ارسال می‌شود، الگوی روشن و خاموش شدن LEDهای نمایشگر را تعیین می‌کند.



**شرح پروژه:**

برای شبیه‌سازی این کد در پروتیوس، ابتدا مدار مناسب را میسازیم که شامل میکروکنترلر 8086 یا معادل آن و پورت 8255Aباشد. سپس، پورت‌ها و کنترلرهای لازم را به درستی تنظیم می کنیم.

* 1. **باز کردن نرم‌افزار پروتئوس**
* نرم‌افزار **Proteus** را باز کرده و از منوی "New Project" یک پروژه جدید ایجاد می‌کنیم.
* یک نام برای پروژه انتخاب کرده و پوشه ذخیره‌سازی آن را تعیین می‌کنیم.
* در این مرحله، تنظیمات اولیه مانند نوع پروژه (Blank Project) و تنظیمات منبع تغذیه انجام می‌شود.

**2. اضافه کردن میکروکنترلر 8086:**

* از قسمت **Library** یا کتابخانه قطعات، عبارت "8086" را جستجو می‌کنیم.
* پردازنده 8086 را انتخاب کرده و به صفحه طراحی شبیه‌سازی اضافه می‌کنیم.
* میکروکنترلر 8086 به عنوان قلب مدار عمل می‌کند و وظیفه پردازش اطلاعات و مدیریت جریان داده را بر عهده دارد.

**3. اضافه کردن پورت A8255:**

* از کتابخانه قطعات، عبارت "A8255 " را جستجو کرده و به شبیه‌سازی اضافه می‌کنیم.
* 8255A یک **پورت ورودی/خروجی قابل برنامه‌ریزی** (PPI) است که برای ارتباط میکروکنترلر با دستگاه‌های جانبی مانند نمایشگرها یا کلیدها استفاده می‌شود.
* این قطعه سه پورت اصلی A)، B، (C و یک پورت کنترلی (Control Port) دارد.

**.4 اتصال پورت‌ها به میکروکنترلر:**

* **اتصالات ضروری بین میکروکنترلر و پورت 8255A:**
  + خطوط **AD0 تا AD7** از 8086 را به پایه‌های **D0 تا D7** در A8255 متصل می‌کنیم. این خطوط برای انتقال داده استفاده می‌شوند.
  + سیگنال‌های کنترلی:
    - **WR (Write)** برای نوشتن داده به A8255 متصل می‌شود.
    - **RD (Read)** برای خواندن داده از A8255 متصل می‌شود.
    - **A0 و A1** به پایه‌های آدرس در A8255 متصل می‌شوند تا پورت‌های مختلف A)، B، (Cیا Control انتخاب شوند.
    - **CS (Chip Select)** برای فعال‌سازی A8255 به سیگنال کنترلی میکروکنترلر متصل می‌شود.
  + پورت‌های **PA، PB و PC** از 8255A را به نمایشگرهای 7-segment متصل می‌کنیم.

**.5 اتصال تغذیه و زمین:**

* همه قطعات 8086، A8255 و نمایشگرها به خطوط تغذیه و زمین **VCC)** و (**GND** متصل می‌شوند.
* از یک منبع تغذیه 5 ولت برای تأمین انرژی مدار استفاده می‌کنیم.

**.6اضافه کردن نمایشگرهای 7-Segment**

* از کتابخانه قطعات، عبارت "7-Segment Display" را جستجو کرده و سه عدد نمایشگر 7-segment به مدار اضافه می‌کنیم.
* هر نمایشگر به یکی از پورت‌های 8255A متصل می‌شود:
  + **پورت A (PA0-PA7)** نمایشگر اول (سمت راست).
  + **پورت B (PB0-PB7)** نمایشگر دوم (وسط).
  + **پورت C (PC0-PC7)** نمایشگر سوم (سمت چپ).
* پایه‌های نمایشگر به ترتیب به پایه‌های داده پورت‌های A8255 متصل می‌شوند.

**.7 پیکربندی قطعات:**

* برای اینکه A8255 به درستی عمل کند، نیاز به پیکربندی رجیستر کنترلی داریم:
  + با ارسال یک دستور کنترلی به A8255 (از طریق پورت کنترل)، تعیین می‌کنیم که هر پورت به صورت ورودی یا خروجی عمل کند.
  + در این پروژه، هر سه پورت A)، B ، (Cبه صورت **خروجی** تنظیم می‌شوند تا داده‌ها را به نمایشگرها ارسال کنند.
  + رجیستر کنترلی را مطابق با **مد Mode 0** برنامه‌ریزی می‌کنیم.

**.8 نوشتن و بارگذاری برنامه اسمبلی:**

* برنامه‌ای به زبان اسمبلی برای 8086 می‌نویسیم که عملکرد زیر را انجام دهد:
  + مقادیر عددی را تولید کند و آن‌ها را به ترتیب به پورت‌های A، B و C ارسال کند.
  + برنامه شامل دستورات زیر است:
    - **MOV:** برای انتقال داده به پورت‌ها.
    - **OUT:** برای ارسال داده به پورت‌های خروجی.
    - **LOOP:** برای تکرار نمایش داده‌ها.
* برنامه را با استفاده از اسمبلر مانند (emu8086) کامپایل کرده و فایل باینری (.BIN) تولید می‌کنیم.
* فایل BIN را در میکروکنترلر 8086 بارگذاری می‌کنیم:
  + روی 8086 کلیک کرده و از قسمت تنظیمات، فایل BIN تولید شده را انتخاب می‌کنیم.

**.9اجرای شبیه‌سازی و مشاهده خروجی:**

* شبیه‌سازی را اجرا می‌کنیم تا عملکرد مدار بررسی شود.
* پردازنده، داده‌ها را به پورت‌های A، B و C ارسال می‌کند.
* نمایشگرهای 7-segment داده‌های عددی را به ترتیب نمایش می‌دهند.
  + برای مثال: اگر پردازنده عدد 123 تولید کند، نمایشگرها به ترتیب "1"، "2"، و "3" را نشان می‌دهند.