

۱-

(الف)

شماتیک در فایل پروتئوس ضمیمه شده است. برای اجتناب از شلوغی مدار توسط سیم ها، از terminal استفاده کرده ایم.

(ب)

در datasheet مربوط به این حافظه ها زمان های زیادی وجود دارند ولی نهایتا چند زمان خاص هستند که رعایت کردن آن ها در جوابدهی به این سؤال مد نظر قرار گرفته است

در نوشتن، ۳ زمان مهم وجود دارند:

۱- t_{DVWH} که بیانگر زمانی است که قبل از لبه بالارونده write خطوط داده حافظه باید مقدار valid شان را داشته باشند تا در

لبه بالارونده عمل نوشتن در حافظه به درستی انجام شود. در واقع به نوعی نقش setup time را بازی میکند.

۲- t_{WHDX} که بیانگر زمانی است که بعد از لبه بالازنده write خطوط داده حافظه باید مقدار valid شان را نگه دارند. نهایتا بعد از

آن مقدار valid شان میتوانند از دست بدهند.

۳- t_{WLWH} که در واقع همان عرض پالس نوشتن در حافظه است.

جزئیات بیشتر درباره اعداد و ارقام این زمان ها، در کد اسمبلی مربوط به قسمت د) در کامنت ها ضمیمه شده است.

در خواندن، ۳ زمان مهم وجود دارند:

۱- t_{AVQV} که بیانگر مدت زمانی است که باید از قرار دادن آدرس valid روی خطوط آدرس حافظه سپری شود تا داده valid در

خروجی حافظه دیده شود.

۲- t_{ELQV} که بیانگر مدت زمانی است که باید از فعال شدن $\text{Chip Enable} = \text{CE}$ سپری شود (دقت کنید بعلت active low بودن

این پایه، فعال شدن این پایه با 0 کردن آن انجام میگیرد) تا بتوان داده valid در خروجی حافظه مشاهده نمود.

۳- t_{GLQV} که بیانگر مدت زمانی است که باید از فعال شدن $\text{Output Enable} = \text{OE}$ سپری شود (دقت کنید بعلت active low بودن

این پایه، فعال شدن این پایه با 0 کردن آن انجام میگیرد) تا بتوان داده valid در خروجی حافظه مشاهده نمود.

البته کارمان در خواندن نسبت به نوشتن ساده تر است. همه زمان ها، زمانی هستند که باید قبل از مشاهده خروجی حافظه رعایت شوند. پس

با رعایت ماکزیمم این ۳ زمان میتوان هر سه ی این زمان ها را رعایت کرد. مطابق جدول ماکزیمم این ۳ زمان، چه برای حافظه M27C64A

و چه برای حافظه M48T02 برابر t_{AVQV} میباشد.

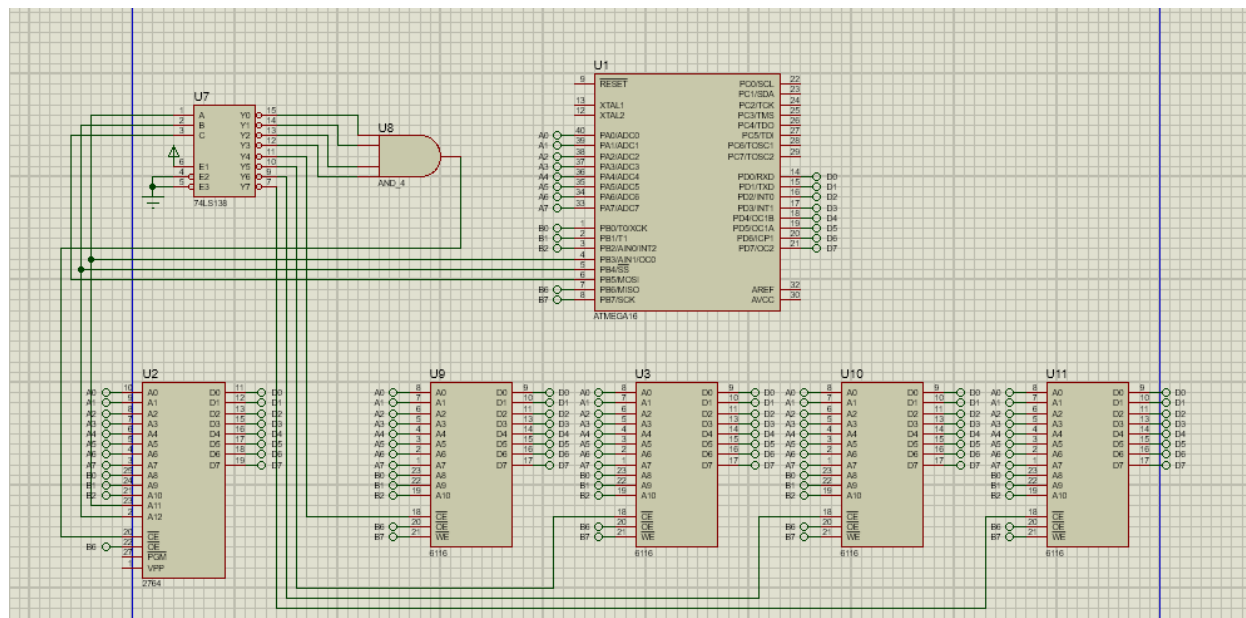
جزئیات بیشتر درباره اعداد و ارقام این زمان ها، در کد اسمبلی مربوط به قسمت د) در کامنت ها ضمیمه شده است.

(ج)

با توجه به این که تعداد حافظه ها ۵ تاست و این که این حافظه ها ورودی فعالسازشان را از خروجی های decoder میگیرند، پس

decoder حداقل باید ۵ پایه خروجی داشته باشد. برای همین از 74LS138 استفاده شده که یک decoder_3:8 میباشد. این decoder

۸ پایه خروجی دارد. ۴ پایه خروجی بالایی آن به کمک AND چهار ورودی به فعالساز^۱ EEPROM و ۴ پایه بعدی آن مستقیماً به ورودی فعالساز حافظه های SRAM^۲ وصل شده اند. جزئیات بیشتر در شکل زیر که از فایل پروتئوس گرفته شده قابل مشاهده است:



(د)

برنامه در فایل اسمبلی ضمیمه شده است. زیرروال MemRead_EEPROM1 کار خواندن از آدرس 1500H در حافظه EEPROM را انجام میدهد. زیرروال MemWrite_SRAM1 نیز کار نوشتن 90H در آدرس شماره 2500 را انجام می دهد که به SRAM اول تعلق دارد. برای اطمینان بیشتر از نوشته شدن درست، زیرروال MemRead_SRAM1 نیز نوشته شده و دقیقاً بعد از MemWrite_SRAM1 صدا زده شده است تا از همان آدرس، عمل خواندن را انجام دهد تا مطابقت نتیجه خوانده شده با نوشته شده محقق گردد.

-۲

ریزپردازنده مورد بررسی، ریزپردازنده RM48L952 از شرکت Texas Instruments میباشد. با مراجعه به datasheet مربوط به آن، معلوم شد این ریزپردازنده، حافظه های زیر را دارد که بصورت یکپارچه مورد استفاده قرار می گیرند:

- حافظه Program Flash با حجم 3MB
- حافظه RAM به همراه ECC با حجم 256KB
- حافظه flash به همراه ECC برای Emulated EEPROM

¹ M27C64A

² M48T02