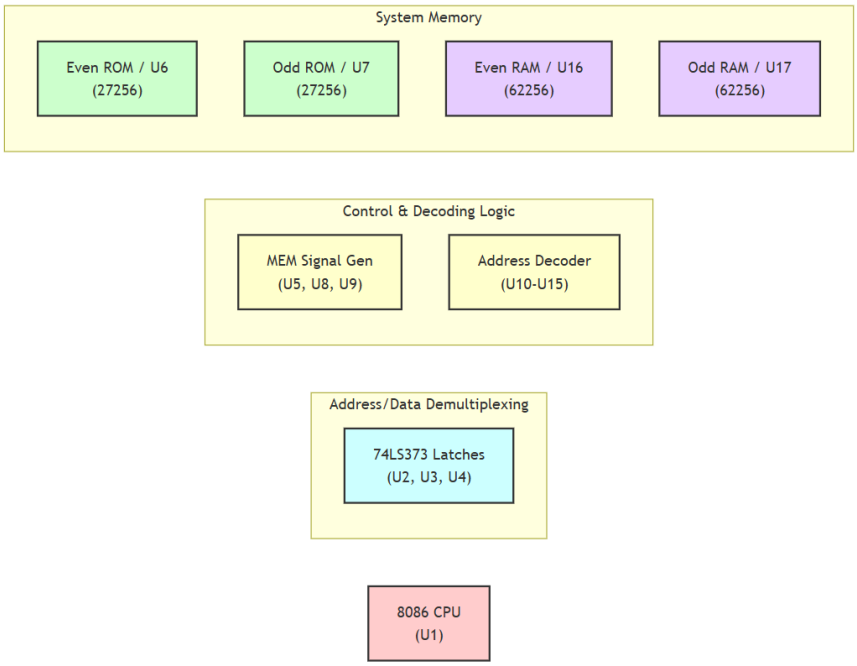
بسم الله الرحمن الرحيم

دیاگرام بلوکی سیستم طراحی‌شده

این سیستم بر پایه پردازنده ۱۶ بیتی ۸۰۸۶ در حالت حداقل طراحی شده است. معماری کلی سیستم شامل چهار بخش اصلی است:

۱. واحد پردازش مرکزی(CPU) : پردازنده ۸۰۸۶ که کنترل گذرگاه‌ها را بر عهده دارد
۲. واحد جداسازی(Demultiplexing) : شامل سه عدد آی‌سی لچ 74LS373برای تفکیک گذرگاه آدرس از گذرگاه داده
۳. مدار کنترل و رمزگشایی: گیت‌های منطقی برای تولید سیگنال‌های خواندن/نوشتن و انتخاب تراشه‌های حافظه
٤. حافظه‌ها: دو تراشه ROM (مدل 27256) برای ذخیره دستورالعمل‌ها و دو تراشه RAM (مدل 62256) برای ذخیره متغیرها و نتایج محاسبات اندازه هرکدام از این آی سی ها ۳۲ کیلوبایت است بنابراین در مجموع ۶۴ کیلوبایت حافظه رم و ۶۴ کیلوبایت حافظه رام داریم.



نحوه اتصال حافظه‌های RAM و ROM به پردازنده

در پردازنده ۸۰۸۶، پایه‌های AD0 تا AD15 بین آدرس و داده مشترک هستند. فرآیند اتصال به این شرح است:

- **تثبیت آدرس:** با استفاده از پالس ALE از سوی پردازنده، آدرس‌های تولید شده در سه تراشه لچ 74LS373 تثبیت (Latch) می‌شوند تا گذرگاه آدرس ۲۰ بیتی خالص (A0 تاA19) در کل سیکل در دسترس باشد.
- **گذرگاه داده:** برای پشتیبانی از معماری ۱۶ بیتی، حافظه‌ها به دو بانک زوج (Even) و فرد (Odd) تقسیم شده‌اند. ۸ بیت پایین گذرگاه داده (D0 تا D7) به بانک‌های زوج و ۸ بیت بالا (D8 تا D15) به بانک‌های فرد RAM و ROM متصل شده‌اند.
- **سیگنال‌های کنترلی:** سیگنال‌های خواندن (MEMR) و نوشتن (MEMW) با استفاده از ترکیب گیت‌های منطقی OR و NOT از روی پایه‌های RD ، WR و M/IO تولید شده و به پایه‌های فعال‌ساز حافظه‌ها (OE و WE) متصل گردیده‌اند.

روش رمزگشایی آدرس و نگاشت حافظه

در این مدار از روش رمزگشایی جزئی استفاده شده است. این روش پیچیدگی سخت‌افزاری را کاهش داده و همزمان از تداخل حافظه‌ها جلوگیری می‌کند.

- **تفکیک ROM و RAM :** پرارزش‌ترین بیت آدرس یعنی خط A19 برای انتخاب نوع حافظه به کار رفته است. با استفاده از گیت‌های منطقی، اگر مقدار A19 برابر با 1 باشد، تراشه‌های ROM و در صورت 0 بودن، تراشه‌های RAM انتخاب می‌شوند.
- **مدیریت بانک‌های ۱۶ بیتی:** در کنار خط A19 ، از سیگنال خط A0 برای فعال‌سازی تراشه‌های زوج (Even) و از سیگنال خروجی BHE برای فعال‌سازی تراشه‌های فرد (Odd) استفاده شده است.

- **جدول تخصیص فضای آدرس به RAM وROM :**

بلوک حافظه	وضعیت بیت A19	آدرس شروع (Hex)	آدرس پایان (Hex)
RAM	0	00000H	7FFFFH
ROM	1	80000H	FFFFFH

الف) محاسبه آدرس‌های ROM شرط انتخاب(A19 = 1):

- **آدرس شروع**: با قرار دادن بیت A19 روی 1 و صفر کردن ۱۹ بیت باقی‌مانده، پایین‌ترین آدرس استخراج می‌شود:

(0000) (0000) (0000) (0000) (1000) در مبنای باینری و 80000H در مبنای هگزادسیمال

- **آدرس پایان**: با قرار دادن بیت A19 روی 1 و یک کردن ۱۹ بیت باقی‌مانده، بالاترین آدرس استخراج می‌شود:

(1111) (1111) (1111) (1111) (1111) در مبنای باینری و FFFFFH در مبنای هگزادسیمال

ب) محاسبه آدرس‌های RAM شرط انتخاب(A19 = 0) :

- **آدرس شروع:** با قرار دادن بیت A19 روی 0 و صفر کردن بقیه بیت‌ها:

(0000) (0000) (0000) (0000) (0000) در مبنای باینری و 00000H در مبنای هگزادسیمال

- **آدرس پایان:** با قرار دادن بیت A19 روی 0 و یک کردن بقیه بیت‌ها:

(1111) (1111) (1111) (1111) (0111) در مبنای باینری و 7FFFFH در مبنای هگزادسیمال

پردازنده ۸۰۸۶ پس از دریافت سیگنال Reset، واکشی دستورات را از آدرس FFFF0H آغاز می‌کند. با توجه به جدول بالا، این آدرس به درستی در محدوده فضای آدرس‌دهی سخت‌افزاری ROM قرار گرفته است.

مدار کلاک و ریست پردازنده

- **مدار کلاک:** برای تامین پالس ساعت سیستم، از یک منبع پالس دیجیتال (Clock Generator) با فرکانس پایدار ۵ مگاهرتز متصل به پایه ۱۹ (CLK) استفاده شده است.
- **مدار ریست:** برای تامین سیگنال RESET از یک(Logic State) متصل به پایه ۲۱ استفاده شد. این logis state یک پالس Active-High را برای راه‌اندازی اولیه و پاک کردن ثبات‌های پردازنده تامین می‌کند.

توضیح عملکرد برنامه اسمبلی

برنامه اسمبلی با هدف تست کامل گذرگاه‌های سیستم (خواندن از حافظه، انجام عملیات حسابی و نوشتن مجدد در حافظه) طراحی شده است. ابتدا ثبات DS با مقدار 0000H مقداردهی می‌شود تا پردازنده بتواند به آدرس‌های پایه RAM دسترسی داشته باشد. برنامه دو بایت داده (که قبلاً در حافظه قرار گرفته‌اند) را از آدرس‌های زوج (0000H) و فرد (0001H) خوانده و وارد ثبات‌های پردازنده می‌کند. سپس دستور ADD برای جمع این دو عدد اجرا شده و در نهایت، نتیجه این جمع با یک دستور MOV در آدرس بعدی (0002H) RAM ذخیره می‌گردد. در پایان، پردازنده با دستور HLT در یک حلقه بی‌نهایت متوقف می‌شود.

(کامنت در کدبرنامه وجود دارد)

فرآیند مرحله‌به‌مرحله انجام کار و شبیه‌سازی

۱. **نحوه نوشتن برنامه اسمبلی و هدف آن:** برنامه با هدف خواندن، محاسبه و ذخیره اطلاعات نوشته شد. بخش مهم کد، استفاده از دستور ORG 0FFF0H در آخر برنامه بود. از آنجایی که پردازنده ۸۰۸۶ پس از بیدار شدن به این آدرس مراجعه می‌کند، یک دستور پرش بلند (JMP 0F000H:0000H) در این قسمت قرار دادیم تا روند اجرای برنامه به صورت خودکار به ابتدای حافظه رام منتقل شود.

۲. **نحوه اسمبلی کردن برنامه و نوع فایل خروجی:**کد اسمبلی در نرم افرار EMU8086 کامپایل شد و خروجی آن به صورت یک فایل باینری(bin.) درآمد. سپس با استفاده از یک اسکریپت پایتون، این فایل باینری به دو فایل مجزا (rom_even.bin) برای بایت‌های کم‌ارزش و (rom_odd.bin) برای بایت‌های پرارزش تقسیم شد تا با سخت‌افزار ۱۶ بیتی مدار همخوانی داشته باشد.

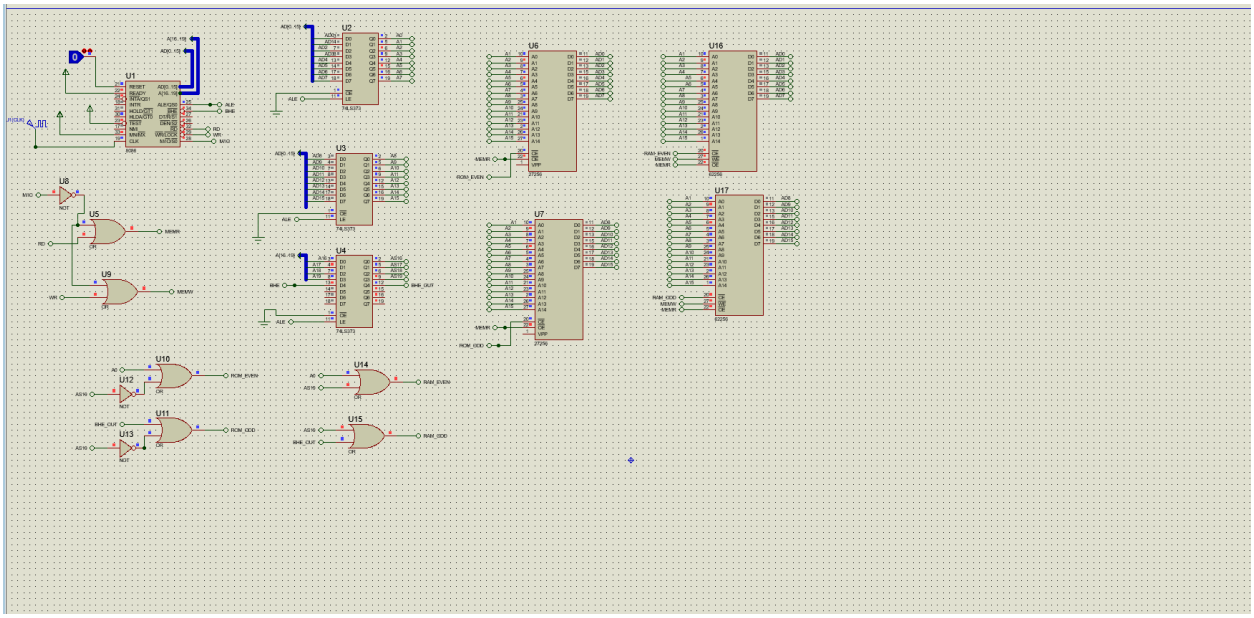
۳. **نحوه قرار دادن فایل برنامه در حافظه ROM در محیطProteus :** در پروتئوس، فایل باینری بایت‌های زوج در تراشه ROM زوج و فایل باینری بایت‌های فرد را در تراشه ROM فرد بارگذاری کردیم. در تنظیمات هر دو آی‌سی، پارامتر آدرس شروع خواندن فایل (File Base Address) روی 0x0000 تنظیم شد تا کدها دقیقاً از اولین بایت فایل‌های آپلودشده خوانده بشوند.

٤. **نحوه راه‌اندازی شبیه‌سازی و اجرای برنامه توسط پردازنده:** شبیه‌سازی مدار در پروتئوس آغاز شد. برای راه‌اندازی پردازنده، کلید منطقی متصل به پایه RESET برای یک لحظه در وضعیت 1 و سپس مجدداً در وضعیت 0 قرار گرفت تا یک لبه پایین‌رونده ایجاد شود. پردازنده با دریافت این شوک بیدار شده و اجرای دستورات واکشی‌شده از رام را آغاز نمود.

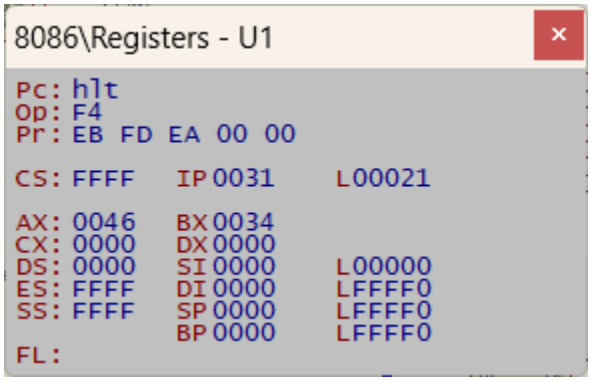
۵. **صبور بودن در اجرای شبیه سازی:** شبیه‌سازی مدار به صورت زمان‌واقعی (Real-Time) اجرا نمی‌شود. بنابراین پس از اعمال سیگنال ریست، نباید بلافاصله دکمه توقف (Pause) را فشار دهیم بلکه لازم است چند ثانیه به نرم‌افزار زمان بدهیم تا پردازنده بتواند سیکل های لازم را طی کرده و فرآیند نوشتن اطلاعات در RAM را اتمام کند. پس از گذشت این مکث کوتاه است که می‌توانیم مدار را متوقف کرده و نتایج ثبت‌شده را در حافظه مشاهده کنیم.

٦. **مشاهده نتیجه اجرای برنامه در حافظه RAM در محیط شبیه‌سازی:** پس از اجرای کامل برنامه رسیدن به دستور(HLT) ، شبیه‌سازی متوقف گردید. با باز کردن پنجره محتویات حافظه مربوط به تراشه‌هایRAM ، مشاهده شد که حاصل‌جمع محاسباتی پردازنده با موفقیت در آدرس مشخص‌شده

ذخیره شده است. همچنین بررسی پنجره 8086 Registers نشان داد که مقادیر در ثبات‌های مربوطه به درستی پردازش شده‌اند.



تصویر ۲: نمای کلی از مدار طراحی شده در حال اجرا (شامل اتصالات پردازنده، لچ‌ها و حافظه‌ها)



تصویر ۱: وضعیت ثبات‌های پردازنده پس از اجرای برنامه (پنجره ۸۰۸۶ Registers)