به نام خدا

گزارش کار آزمایشگاه ریزپردازنده

آزمایش ۱۰

مدرس : مهندس بی طالبی

تارا برقیان

مهرشاد سعادتی نیا

نیم سال اول ۰۱-۱۴۰۰

فهرست

مقدمه :

در این تمرین با برنامه نویسی زبان اسمبلی ۸۰۸۶ و بستن یک کامپیوتر با اجزای جانبی آشنا شدیم.

پاسخ سوالات تحلیلی:

- با استفاده از DB DD DW میتوان مقدار حافظه مورد نیاز را تعیین کرد و با EQU میتوان ان قسمت دلخواه را نام گذاری کرد مثلا : PORT_A EQU 060H

- اگر فقط از nand ها استفاده کنیم میتوان ورودی گیت را مستقیما به مقادیر ادرس یا not آن وصل کرد و با تاخیر کمتری دارد از طرفی پیچیدگی آن از یک بلوک ماژول بیشتر است پس هزینه ساخت بیشتر میشود البته تراشه های بیشتری هم برای اتصال و نگه داری نیاز است.

  از طرفی اگر از decoder استفاده کنیم مدار ساده تر میشود (درست است که شاید درون ماژول پیچیده باشد اما ما با ان درگیر نمیشودیم.) البته برای ورودی های دیکودر از nand ممکن است استفاده کنیم که این خودش باعث اضافه شدن قطعات میشود و مسیر پیموده شده طولانی میشود(تاخیر بیشتر) ولی وقتی میخواهیم یک قطعه بزرگ بسازیم سادگی و قابلیت اسکیل برایمان بسیار مهم میشود و در مقایسه با nand خالی این روش بهتری است. همچنین اگر دیکودر برای سیستم های مربوط به حافظه استفاده شود گیت های کمتری نیاز دارد.

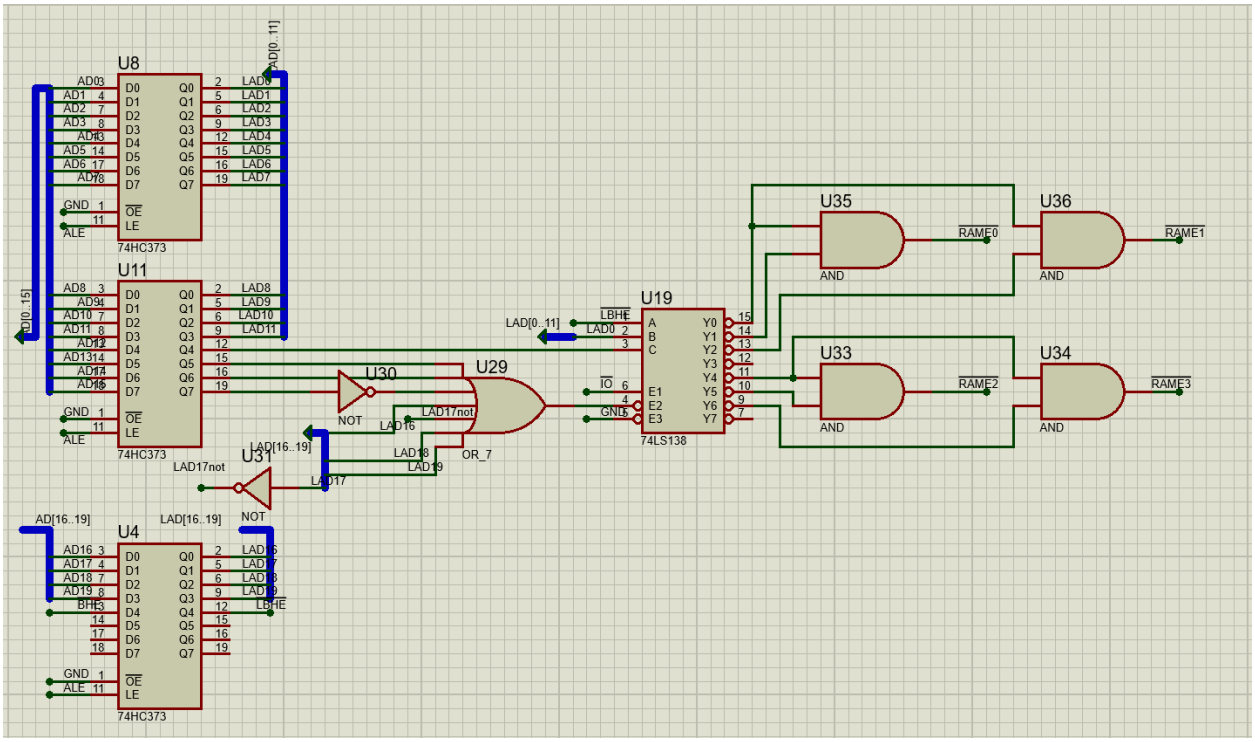
- یکی از مهم ترین تفاوت های این ۲ پردازنده دیتا باس آنها است که در ۸۰۸۶ ، ۱۶ بیت و در ۸۰۸۸، ۸ بیتی است.
  سیگنال M/IO در ۸۰۸۸ به صورت عکس میباشد و برای استفاده از آن باید با not معکوس شود.
  سیگنال bank high enable (bhe) در ۸۰۸۷ نام دیگری دار : SSO یا status signal
  در کل اکثر تغییرات مربوط به طول باس داده و واحد های درگیر با آن است.

توضیح کد:

ابتدا از قسمت دیکد شروع می کنم و با توجه به ایسنکه آدرس های ممووری از 0x28000 تا 0x28FFFF می باشد مدار را باید طوری طراحی کنیم که در صورت دسترسی به این حافظه ها بانک حافظه ی مربوطه را فعال کند.

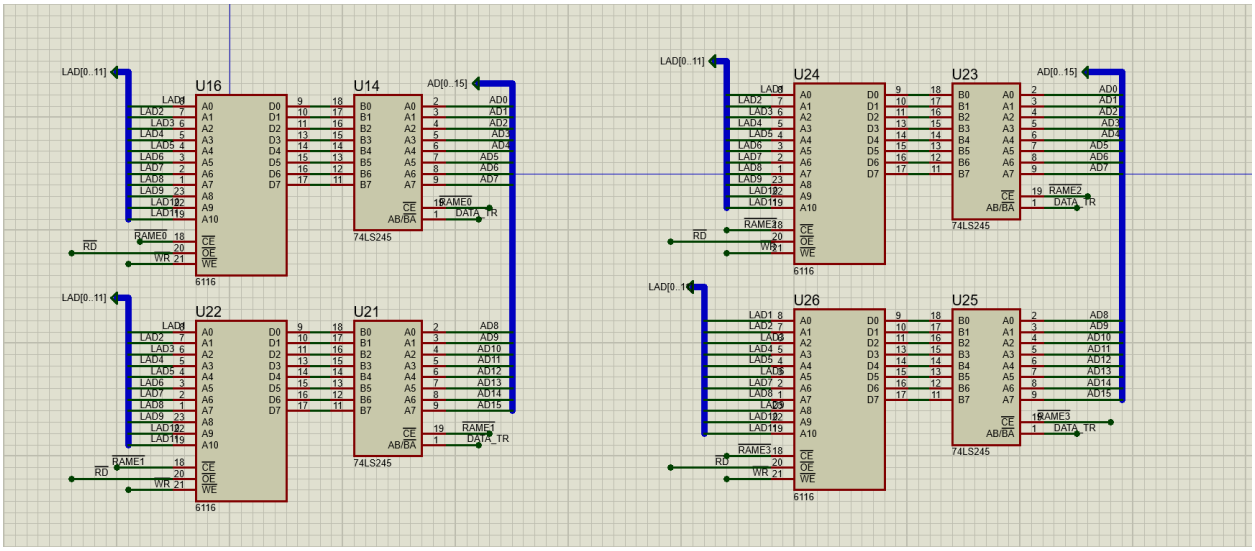
برای دریافت آدرس های باس های حافظه از latch 74HC373 اسفتاده می کنیم. با خروجی آنرا برای دیکد آدرس استفاده می کنیم.

مدار دیکد را در شکل زیر مشاهده می کنیم که شیوه ی کار آن تقریباً مشخص است و با کمک BHE و A0 (LAD0) می توانیم بانک های زوج و فرد را انتخاب کنیم. و not LAD17 و ۱۵ به یک OR میفرستیم و خروجی آنرا به E2 میدهد.

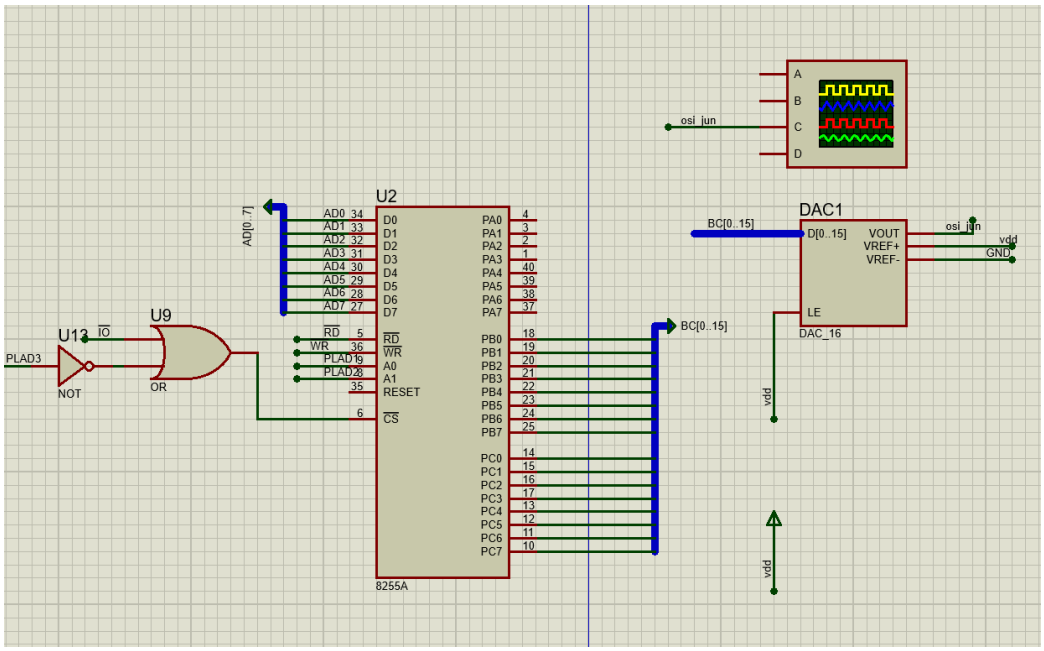


در شکل زیر دو جفت بانک زوج و فرد حافظه را مشاهد می کنیم که به کمک مدار دیکد قسمت قبلی انتخاب می شود.

از چیپ های 74LS245 هم برای ارتباط دو طرفه مموری استفاده می شود.



شکل زیر هم قسمت ارتباط با DAC را نشان می دهد که سیگنال سینوسی تولید شده در میکرو را به کمک چیپ 8255 به مبدل دیجیتا به آنالوگ میدهد و خروجی تولید شده را برای نمایش به اسیلوسکوپ وصل می کنیم.



از پورت های B, C چیپ برای انتقال داده ۱۶ بیتی به مبدل دیجیتال استفاده کردیم و در داخل کد هم تنظیمات را به شکل زیر انجام دادیم.

```
PORTA EQU 060H
PORTB EQU 062H
PORTC EQU 064H
CON_REG EQU 066H
```

همانطور که در صورت سوال گفته شده آدرس اولین پورت (A) را با کمک EQU تعریف می کنیم و باقی پروت ها را هم با کمک آن اِدرسشان مشخص میشود.

سپس در کد می توانیم مقادیری که می خواهیم را در آدرس های مربوطه بنویسیم تا به DAC داده شود.

ابتدا از موج مثلثی شروع کردیم تا از کارکرد سیستم اطمینان حاصل کنیم اما تفاوت چندانی با سینوسی ندارد و با کمی تغییر می توانیم آنرا تبدیل به سینوس کنیم.

ابتدا آدرس کنترل رجیستر را در DX قرار میدهیم. سپس مقدار1000_0000 را در AL قرار بعد با کمک دستور OUT آنرا در آدرس پورت کنترل رجیستر قرار میدهیم.

```
MAIN          PROC FAR
    MOV AX, @DATA
    MOV DS, AX
PART_1 :
    MOV DX,CON_REG
    MOV AL, 80H             ; SET THE CONTROL WORD = 1000 0000
    OUT DX, AL
```

در ادامه نیاز هست که مقادیر مورد نیاز برای تولید موج مثلثی را در رجیستر AX بنویسیم تا بتوانیم با انتقال آن DX در پورت خروجی OUT کنیم. که کد آنرا مشاهده می کنید. توضیحات کد در کامنت ها قابل مشاهده است.

```
UP:
    CALL OUTPUT
    INC AX ; To raise wave from 0V to 5V increment AL
    CMP AX,0000H

    JNZ UP ; Jump UP till rising edge is reached i.e. 5V

    MOV AX,00FFH
UP1:
    CALL OUTPUT
    DEC AX ; To fall wave from 5V to 0V decrement AL
    CMP AX,00FFH
    JNZ UP1 ; Jump UP till falling edge is reached i.e. 0V
    JMP BEGIN

OUTPUT:
    OUT DX,AX
    CALL DELAY
    INT 21H
```