وب سایت جامع الکترونیک ، برق و کامپیوتر

www.ir-micro.com

جزوه آزمایشگاه میکروپروسسور Z80

(قسمت اول)

شما هم می توانید مقالات ، آموزش ها و ... خود را برای ما ارسال کنید تا با نام خودتان در سایت قرار داده شود .

Hamed_2_0_0_5@Yahoo.com
Hamed@Ir-micro.com

جزوه راهنمای بورد کاربردی
APPLICATIONS BOARD

آینده پرواز

بنام خدا

فهرست مطالب

آزمایشهای APPLICATIONS BOARD

۱) مقدمه

اصولا از بورد کاربردی (Applications Board)A.B. برای آموزش چگونگی کاربرد میکروکامپیوتر در کنترل سیستمهای متفاوت استفاده میشود. تنها با استفاده از این یک بورد امکان انجام آزمایشهای متنوعی وجود دارد. این بورد مستقیما و از طریق یک کابل چهل سوراخه به میکروپروفسور MPF-1B متصل میشود در اینصورت MPF-1B باید مجهز به تراشه Z80 PIO که یک تراشه قابل برنامه ریزی است باشد. بورد کاربردی را میتوان به میکروکامپیوترهای MPF-1P، MPF-1/88 و یا هر کامپیوتر دیگری که دارای دو عدد باب ورودی - خروجی قابل برنامه ریزی ۸ بیتی باشد، وصل نمود.

A.B. توسط یک منبع تغذیه ۹V و ۰/۵A تامین میشود و شامل قسمتهای زیر میباشد :

*۸ عدد سوئیچ

*۸ عدد LED رنگی

*مبدل دیجیتال به آنالوگ ۸ بیتی

*مقایسه کننده که میتوان با استفاده از آن مبدل آنالوگ به دیجیتال نیز ساخت.

*یک نمایشگر ۱۰ قسمتی (BarGraph)

*یک موتور d.c.

*پنکه و اشعه مادون قرمز

*یک عدد هیتر

*سنسور حرارت

*سنسور شدت نور

*پتانسیومتر

*پروب لاجیکی

*کانکتور خارجی

که هرکدام از این قسمتهای A.B. ممکن است مستقلا در حالت روشن و یا خاموش سوئیچ شده و ترتیبی داده شده که بخشهای مختلف این سیستم بتوانند همزمان با هم کار کنند.

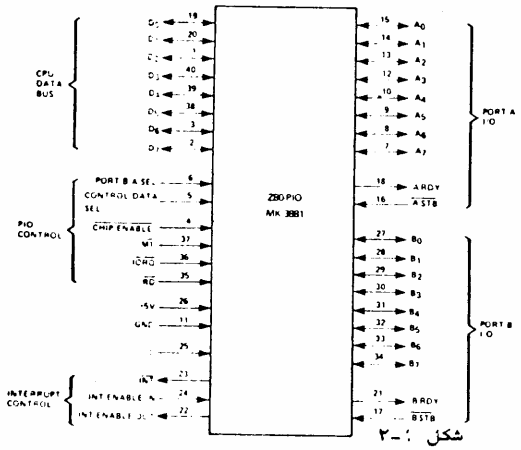
۲ - PIO (Parallel Input/Output Circuit)

برای استفاده از میکروپروفسور MPF-1B و وصل آن به A.B. لازم است یک آی سی چهل پایه Z80 PIO در سوکت چهل پایه بالای نمایشگرها در میکروپروفسور قرار داده شود، لذا آشنائی دانشجویان با PIO ضروری میباشد.

۱-۲)معرفی PIO

اصولا ارتباط میکروکامپیوترها با دنیای خارج خود توسط بابهای ورودی وخروجی صورت میگیرد ،بابهای ورودیوخروجی در واقع مدارهای رابطی هستند که برای اتصال میکروکامپیوتر به سیستم جنبی ،بین این دو قرار میگیرند.

PIO یک رابط قابل برنامه ریزی است که با برنامه ریزی مناسب میتوان از بابهای آن بعنوان بابهای ورودی وخروجی استفاده نمود .در اینجا شما آشنائی مختصری با PIO پیدا میکنید.



شکل ۱-۲

شکل ۱-۲:شمای کلی از سرهای تراشه PIO میباشد .این تراشه دارای هشت خط دوجهته اطلاعات و شش خط کنترل وهمچنین شامل دو باب ورودی - خروجی قابل برنامه ریزی B،A است که در چهار حالت زیر میتوانند قرار داده شوند.

حالت 0 :(خروجی) در اینمورت باب مورد نظر به صورت باب خروجی برنامه ریزی میشود.

حالت 1 :(ورودی) در اینحالت باب برنامه ریزی شده به صورت ورودی مورد استفاده قرار میگیرد و اطلاعات را میتوان توسط آن داخل میکروکامپیوتر وارد نمود.

حالت 2 :(دوجهته) در اینحالت باب مورد نظر بصورتی برنامه ریزی میشود که میتوان اطلاعات را در دوجهت ورودی وخروجی بین میکروکامپیوتر و سیستم جنبی رد و بدل کرد .تنها باب A باینمورت قابل برنامه ریزی است.

حالت 3 :(کنترل بیت) در اینحالت هریک از سرهای موجود بر روی باب مورد نظر برنامه ریزی را میتوان مستقیما بعنوان ورودی و یا خروجی تعریف نمود.

۲-۲)برنامه ریزی PIO

الف- برنامه ریزی حالت باب مورد نظر :

هریک از بابهای A و B توسط یک بایت کنترل قابل برنامه ریزی است ،فرمت کلی بایت مزبور

بصورت زیر است:

D7	D6	D5	D4	D3	D2	D1	D0
M	M	×	×	1	1	1	1

در اینحالت چهار بیت D0 تا D3 که برابر 1 است مشخص کننده این است که بایت یک بایت کنترل برای برنامه ریزی حالت باب میباشد و بیتهای M0 و M1 چهار حالت ممکن زیر را مشخص میکنند.

	M1	M0
برای حالت 0	0	0
برای حالت 1	0	1
برای حالت 2	1	0
برای حالت 3	1	1

درصورتیکه بایت کنترل فوق حالت 3 را برای باب مورد نظر انتخاب کند، بایت کنترل بعدی که وارد همان باب میشود بعنوان بایت معرف ورودی - خروجی تلقی میشود که بیتهای "یک" موجود در این بایت سرهای ورودی را مشخص میکند و بیتهای "صفر" مربوط به سرهای خروجی میباشند.

در میکروکامپیوتر MPF-1B، PIO بصورت زیر آدرس دهی میشود:

اطلاعات باب A	80H
اطلاعات باب B	81H
کنترل باب A	82H
کنترل باب B	83H

۳) طریقه وصل بورد کاربردی به کامپیوترهای مختلف

هرچند که اتصال بورد کاربردی به کامپیوتر بسیار راحت میباشد ولی بهرحال اینکار باید با دقت کافی انجام شود تا از هرگونه صدمه زدن به بورد و به کامپیوتر میزبان جلوگیری بعمل آید.

۱-۳) کامپیوتر MPF-1B

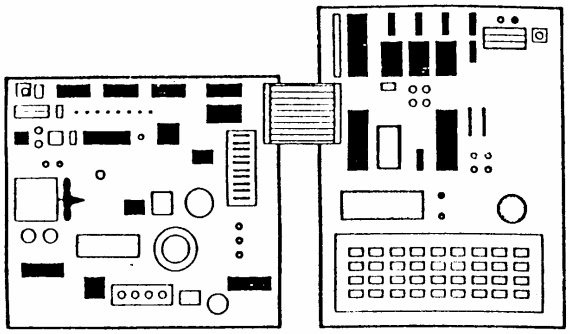
۱- درحالیکه هیچکدام از دو دستگاه (بورد کاربردی و میکروپروفسور) به برق متصل نیستند، کابل تخت چهل رشته ای را بین کانکتور بورد و کانکتور چهل پایه ای نزدیک به نمایشگرهای روی MPF-1B متصل کنید.

۲ - تغذیه MPF-1B را وصل کنید.

۳ - تغذیه بورد را وصل کنید.

۴ - ۸ عدد LED رنگی روی بورد روشن میشوند. البته هنگامیکه باب B از PIO بصورت خروجی

برنامه ریزی شود این LED ها خاموش میشوند.



شکل ۱-۳ :اتصال به MPF-1B

۲-۳) MPF-1P و MPF-1/88

قبل ازوصل بوردکاربردی به این دوکامپیوتر ،به یکبوردرابط(Interface Boarl) احتیاج میباشد ،بطوریکه توسط بوردرابط،دوباب۸بیتی قابل برنامه ریزی ایجادشده و آنگاه میتوان بوردکاربردی رابه کامپیوتروصل نمود.

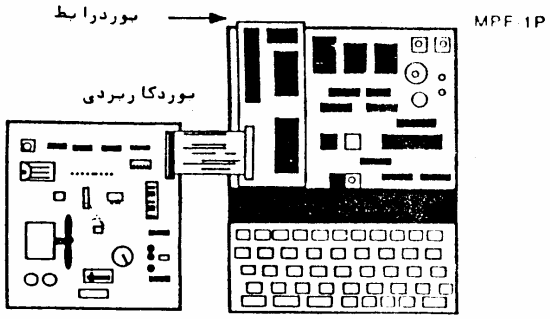
۱ - درحالیکه هیچ تغذیه ای وصل نیست،کابل چهل رشته ایتخت رابین بوردکاربردی وبورد رابط که شامل بابهای قابل برنامه ریزی استمتصل کنید.

۲ - بوردرابط رابه کامپیوترمیزبان متصل نمائید.

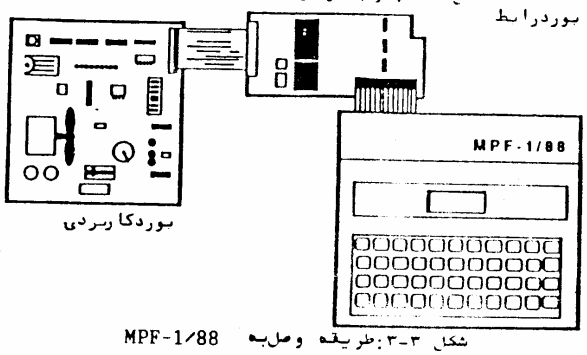
۳ - تغذیه کامپیوتربورد رابط را متصل کنید.

۴ - بوردکاربردی رابه برق وصل کرده وآنراروشن نمائید.

۵ - ۸ عدد LED رنگی روی بوردکاربردی روشن میشوندوهنگامیکه بابروی بورد رابط به صورتخروجی برنامه ریزی میشود،خاموش میگردند.



شکل ۳-۲:طریقه وصل به MPF-1P



شکل ۳-۳:طریقه وصل به MPF-1/88

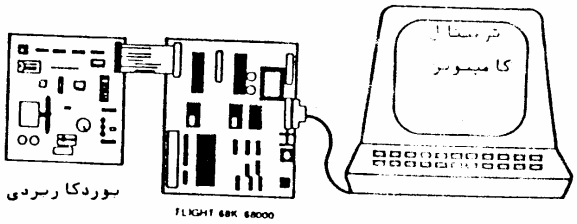
۳-۳) FLIGHT 68K

۱-درحالیکه هیچ تغذیه ای وصل نیست،کابل تخت چهل رشته ای را بین بوردکاربردی و کانکتور چهل پایه ای موجود در طرف چپ Flight 68K وصل کنید.

۲- Flight 68K را به برق وصل کرده و روشن کنید.

۳-بوردکاربردی را به برق وصل کرده و روشن کنید.

۴-۸عدد LED روی بورد روشن شده و بلافاصله با برنامه ریزی یا به صورت خروجی خاموش میشوند



شکل ۳-۴:طریقه وصل به Flight 68K

۴-۳)کامپیوتر BBC (مدلهای B و B+)

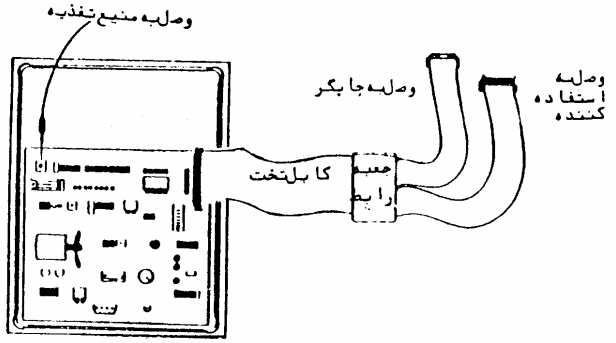
۱-درحالیکه دستگاه به برق وصل نیست،کابل تخت چهل رشته ای را بین بوردکاربردی وجعبه رابط (interface box) چهل پایه ای متصل کنید .

۲-جعبه رابط را ازطریق کانکتور ۲۶پایه ای به باب چاپگر مربوطه کامپیوتر BBC متصل کنید .

۳- ارتباطات را چک کنید .

۴-کامپیوتر BBC را روشن کنید .

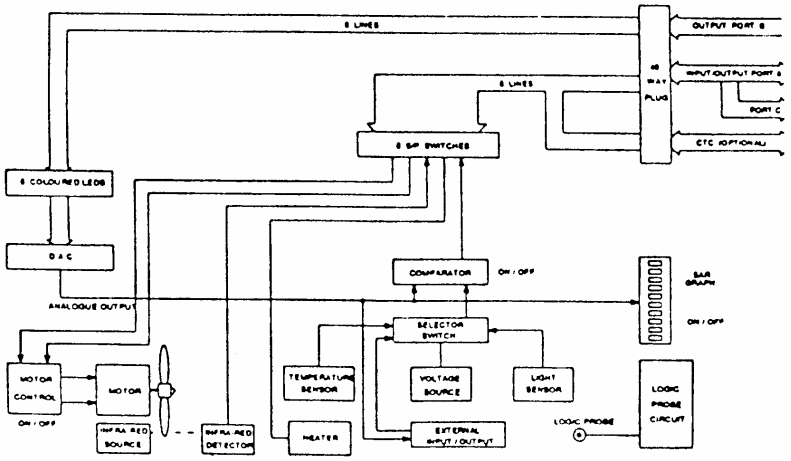
۵-بوردکاربردی را به برق وصل نمائید.



شکل ۵-۳ طریقه اتصال به BBC

۴-قسمتهای مختلف .A.B

همانطور که در بخش PIO توضیح داده شد دو باب A و B در PIO میتوانند به صورتهای متفاوتی برنامه ریزی شوند منتهی برای استفاده آنها در .A.B باب A در PIO به هر دو صورت ورودی و خروجی استفاده میشود وبسته به نوع آزمایش میتوان به یکی از این دو صورت و یا کنترل بیت برنامه - ریزی شود و باب B همیشه به صورت خروجی برنامه ریزی میشود .سوئیچهای مینیاتوری که در بالای بورد قرار دارند و به باب A متصل میباشند در مواقعیکه استفاده نمیشوند بهتر است در وضعیت یک قرار داده شوند.



شکل ۱-۴:بلوک دیاگرام Applications Board

۱-۴)سوئیچها و LED ها

هنگامیکه موتور ،هیتر (Heater)، ADC و Bar Graph خاموش باشند ،بصورت یک سیستم ورودی -خروجی ۸بیتی ساده عمل میکند ،سوئیچها به باب A از PIO میکروکامپیوتر متصل میباشند ودر صورتیکه در رضعیت یک باشند یک "۱"منطقی و اگر در حالت صفر باشند " 0 "منطقی را ایجاد می - کند .از این سوئیچها برای کنترل دستی موتور و هیتر میتوان استفاده نمود .در صورتیکه موتور روشن باشد ،میتوان توسط سوئیچهای ۶ و ۷ آنرا کنترل نمود .در صورتیکه این بیتها هر دو صفر و یا هر دو یک باشند موتور متوقف میشود و اگر متفاوت باشند موتور حرکت میکند .

وضعیت موتور	بیت ۶	بیت ۷
متوقف	۰	۰
چرخش معکوس	۱	۰
چرخش مستقیم	۰	۱
متوقف	۱	۱

همچنین هیتر را میتوان توسط سوئیچ ۵ کنترل نمود بطوریکه اگر هیتر روشن باشد و این سوئیچ ON باشد هیتر روشن میماند و اگر OFF باشد هیتر خاموش خواهد شد .

وضعیت هیتر	بیت ۵
خاموش	۰
روشن	۱

اگر باب A که متصل به سوئیچها است در حالت اولیه ای قرار داده شود که سه بیت ۵ و ۶ و ۷ بصورت خروجی برنامه ریزی شوند (کنترل بیت)،حال سیستم میکروکامپیوتر میتواند مستقیما موتور و هیتر را کنترل نماید در چنین مواقعی باید سوئیچها را در وضعیت ۱ قرار داد .

۸ عدد LED رنگی که درکنار سوئیچها در بالای بورد قرار دارند به ۸بیت از باب خروجی PIO-B در میکروکامپیوتر متصل میباشد و هنگامیکه "۱" منطقی به آنها فرستاده شود روشن و اگر "0" فرستاده شود خاموش میشوند.

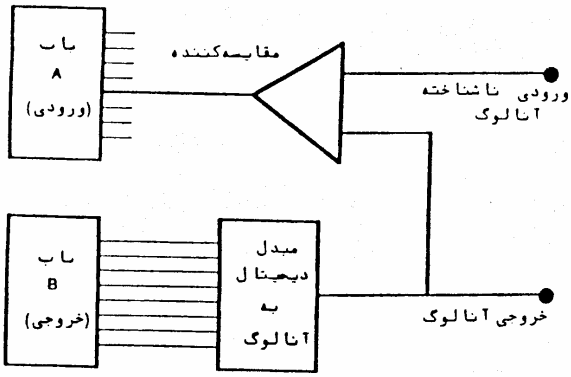
LED ها خاموش "0" منطقی

LED ها روشن "۱" منطقی

توسط این LED ها میتوان اطلاعات باینری که از باب خروجی B می‌آید را مشاهده نمود. همچنین در مواقعیکه ولتاژ آنالوگی را به دیجیتال تبدیل میکنیم، ولتاژ دیجیتال شده را میتوان روی این LED ها مشاهده نمود.

۴-۲)خروجی آنالوگ و Bar-Gragh

باب خروجی B از طریق ۸عدد LED رنگی مستقیما به یک مبدل دیجیتال به آنالوگ متصل میشود. ترتیبی داده شده که خروجی آنالوگ این مبدل در رنج 0.00 تا 2.55 ولت بتواند قرار بگیرد. یک تغییر وضعیت در بیت با ارزش کمتر ورودی دیجیتال DAC در ولتاژ آنالوگ خروجی DAC به اندازه ۱۰میلی ولت تغییرات ایجاد میکند. ولتاژ آنالوگ تولید شده بوسیله DAC در سه قسمت دیگر از A.B. بکار میرود، ابتدا به کانکتور خارجی ولتاژ آنالوگ خروجی مرتبط میشود که این کانکتور در لبه پائین بورد قرار دارد. دوم، خروجی DAC به مقایسه کننده ای وصل میشود که باعث میشود سیستم رابتوان، در مواقعیکه نرم افزار مناسبی نوشته شده است، بعنوان یک مبدل آنالوگ به دیجیتال مورد استفاده قرارداد. سوم، این خروجی آنالوگ از طریق یک سوئیچ ON OFF به Bar-Graph متصل میشود. توسط Bar-Graph میتوان سریعا خروجی آنالوگ DAC را امتحان نمود بدینصورت که به ازای هر 17H که به خروجی باب B یا در واقع LED ها اضافه شود یک قسمت از Bar Graph روشن میشود.



شکل ۲-۴) بلوک دیاگرام مبدل آنالوگ به دیجیتال

ولتاژ آنالوگ تولید شده توسط DAC بوسیله مقایسه کننده ای با یک ولتاژ ورودی نامشخص مقایسه میشود و خروجی مقایسه کننده مستقیما به بیت سوم از باب ورودی A (البته در صورتیکه سوئیچ ADC روشن باشد) متصل میشود . اگر خروجی DAC نسبت به ولتاژ نامشخص ورودی پائین تر باشد بیت ۳ ، "۱" منطقی شده و اگر خروجی DAC بالاتر از ولتاژ نامشخص باشد ، بیت ۳ برابر 0 منطقی خواهد شد . برای استفاده از مبدل آنالوگ به دیجیتال لازم است که بیت ۳ از سوئیچها به صورت خاموش یعنی در حالت یک باشد .

۳-۴) ورودی آنالوگ

ورودی آنالوگ نامشخصی که به مقایسه کننده وارد میشود و قبلا در مورد آن صحبت شد همانطور که از شکل (۱-۴) مشخص است از طریق یک سوئیچ سلکتور سه حالته ، از یکی از قسمتهای زیر تامین میشود :

۱) سنسور حرارت

۲) منبع ولتاژ

۳) سنسور شدت نور

سلکتور این سوئیچ براحتی به طرف راست و یا چپ میلغزد و ورودی مورد نظر را انتخاب میکند .

- سنسور حرارت در واقع یک دیود نیمه هادی میباشد که روی سطح یک مقاومت پروانه سوار شده است و یک ولتاژ خطی متناسب با حرارت تولید میکند ، بطوریکه به ازای هر ۱ درجه سانتیگراد ۲ میلی ولت ولتاژ افزایش میباشد و در صفر درجه 0.0 ولت است .

منبع ولتاژ : یک پتانسیومتر ساده میباشد ، بطوریکه یک ولتاژ آنالوگ در رنج صفر تا ۲/۵۵ ولت تولید میکند و در بسیاری از آزمایشها استفاده میشود .

- سنسور شدت نور یک مقاومت نوری است که در رنج وسیعی از شدت نور ، زیاد میشود . ولتاژی که این سنسور تولید میکند بطور غیر خطی با افزایش شدت نور ، زیاد میشود . ولتاژ ایجاد شده در رنج ۰/۰۲ ولت تا حدود ۱/۸ ولت میباشد .

ورودی هیتر یک ورودی آنالوگ نیست اما میتوان مستقیما برای کنترل سنسور حرارت از آن استفاده نمود و بدینصورت یک کنترل مدار بسته حرارت بوجود آورد . وقتی که هیتر روشن میشود میتواند مستقیما بوسیله کامپیوتر و یا از طریق بیت ۵ سوئیچها (وصل به باب A) کنترل شود .

((توجه هیتر خیلی داغ است ، لطفا به آن دست نزنید))

حرارت سطح هیتر میتواند تا حدود ۷۵ درجه سانتیگراد بالا برود و بنابراین بهتر است که به آن دست نزنید . هنگامیکه هیتر روشن است LED قرمز کنار آن نیز روشن میباشد .

توجه داشته باشید که سنسور حرارت را توسط پنکه متصل به موتور میتوان خنک نمود.

۴-۴)کنترل موتور و فیدبک

توسط بیتهای ۰و۷ از بانک A میتوان موتور d.c کوچکی را که روی بورد قرار دارد کنترل نمود. قبل از حرکت کردن موتور، سوئیچ موتور باید روشن باشد و بیتهای ۰و۷ برای کنترل حرکت موتور بصورت مستقیم و یا عکس و یا توقف موتور، استفاده میشوند. موتور میتواند با سرعت بالائی بخ خد و این سرعت قابل کنترل نیز میباشد.

یک پنکه به شافت موتور متصل است که علاوه بر خنک کردن سنسور حرارت، در هنگام چرخش باعث قطع مکرر اشعه مادون قرمز شده و در نتیجه هر بار که یک پره اشعه را قطع میکند. یک پالس منفی به بیت ۴ از بانک A میفرستد و بدینوسیله میتوان تعداد دور موتور را شمارش و کنترل نمود.

۵-۴)پروب لاجیکی

سیگنالهای موجود در یک سیستم میکروکامپیوتری از سه حالت زیر را میتوانند داشته باشند:

LOW -" 0 " منطقی	۰/۸ V	تا	صفر
امپدانس بالا	۲/۴ V	تا	۰/۸
HIGH -" 1 "منطقی	۵ V	تا	۲/۴

پروب لاجیکی برای سنجش ولتاژ هر نقطه از مدار طراحی شده است و سه LED مربوط به پروب لاجیکی به صورتهای HIGH و LOW و PULSE نامگذاری شده اند.

روشن شدن HIGH LED مبین یک ولتاژ high (بالای ۲/۴ ولت) میباشد که در واقع با " ۱ " منطقی مطابقت میکند و روشن شدن LOW LED یک ولتاژ پائین یعنی زیر ۰/۸ ولت را نشان میدهد و در واقع صفر منطقی را نمایش میدهد .

PULSE LED در مواقعیکه یک موج وجود دارد یعنی ولتاژ با سرعت صفر و یک میشود خاموش و روشن میگردد .

۵)کاربرد کنترلی

۵-۱)کنترل سرعت

کنترل سرعت چرخش ماشینها دارای بیشترین کاربرد در سیستمهای کنترل صنعتی میباشد و همانطورکه درهمه سیستمهای کنترلی ،وجود فیدبک ضروری است در Applications Board نیز فیدبک از وضعیت موتور توسط پنکه ایکه پره های آن در یکدور چرخش اشعه مادون قرمز را به تعداد پره ها قطع نموده ،گرفته میشود .

در اینجا کنترل به دو روش میتواند صورت بگیرد :

۱- زمان یکدور چرخش اندازه گیری بشود .

۲- تعداد چرخش در یک واحد زمانی اندازه گیری شود .

در روش اول،کنترل سرعت خیلی خوب میتواند انجام بگیرد ،زیرا زمان یکدور چرخش بسیار کوچک است .(البته کنترل مناسب با نرم افزار صورت میگیرد .)

درروش دوم ،که تعداد چرخش در یک واحد زمانی شمرده میشود ،کنترل سرعت مشکل میباشد ولی در عوض اندازه گیری سرعت موتور بسیار راحت است و این روش در مواقعی بیشتر استفاده میشود که اندازه گیری سرعت از کنترل آن مهمتر باشد .

۵-۲)کنترل حرارت

در هر جائیکه در یک پروسس صنعتی تغییرات درجه حرارت لازم باشد ،فیدبک نیز برای کنترل سیستم بکار میرود ،برای مثال درکارخانه های مواد غذائی ،پروسس های شیمیائی ،ساخت ترموستات سیکلهای حرارتی در لوازم الکتریکی ،همه مجهز به سیستمهای کنترل حرارت میباشد .

در A.B فیدبک چیپ وضعیتی با توجه به وجود هیتر و پنکه و توانسیو سرخطی حرارت موجود میباشد .

مثلا با استفاده از صفحه کلید میتوان حداکثر درجه حرارت لازم را وارد نمود و در فواصل زمانی معینی درجه حرارت هیتر را روی نمایشگر ،مشاهده نمود و هرگاه درجه حرارت به حداکثر رسید آلارم مخصوصی ایجاد شود .

۶)مشخصات مداری بورد کاربردی

۱ - ۵ عددسوئیچ تکی:برای خاموش وروشن شدن قسمتهای مختلف موتور و ۱۰ LED) Bar Graph قسمتی)وفیدبک موتور (F-MOTOR) وهیتر (Heater) و ADC

ولت۵+ ۱ منطقی
ولت۰/۴ 0 منطقی

۲-۸عدد LED رنگی :

۱ منطقی LED روشن

قابل تطبیق با TTL

0 منطقی LED خاموش

۳- مبدل دیجیتال به آنالوگ۸بیتی :

این مبدل ولتاژی آنالوگ دررنج صفرتا۲/۵۵تولیدمیکند.

ورودی باینری	خروجی آنالوگ
00000000	0.00 V
11111111	2.55 V

۱۰۰ns مدت زمان جواب گرفتن

۰/۱ % درصدخطا

۴- مقایسه کننده ولتاژ

- ماکزیمم تفاوت ولتاژ ورودی :۳۰ ولت
- ماکزیمم ولتاژورودی نسبت به زمین وبرعکس :۱۵ ولت
- بهره مداربار : ۱۰۶ dB
- ولتاژ OFF SET ورودی دیفرانسیال :۲ میلی ولت
- خروجی قابل تطبیق باTTL ومتصل به بیت سوم از PIO وکانال یکازورودی تریگر CTC .

۵ - موتور DC

- کنترل شده ازطریق بیتهای ۶و۷باب A مربوط به PIO

۶ - پنکه واشعه مادون قرمز

هنگامی که سرراه اشعه مانعی وجودندارد :یک منطقی به بیت چهارم ازباب A (PIO)وکانال صفراز CTC

هنگامیکه مانع سرراه اشعه قراربگیرد:0 منطقی به بیت چهارم ازباب A (PIO)وکانال صفر از CTC

۷ - هیتر

حداکثر تولید گرما : ۱ وات

ولتاژ هیتر از : ۱/۲ ولت (خاموش) تا ۲/۲۴ ولت پس از یک ساعت روشن ماندن میرسد.

۸ - حسگر شدت نور

- تولید ولتاژی از ۰/۰۲ ولت (تاریک کامل) تا ۱/۸ ولت (روشن)

- هرچه شدت نور کمتر مقاومت فتوسل بیشتر میشود.

۹ - پتانسیومتر

تولید ولتاژی از صفر تا ۲/۵۵ ولت

۱۰- پروب لاجیکی

HIGH LED :روشن میشود	ولتاژ پروب بیش از ۳/۴ ولت
LOW LED : روشن میشود	ولتاژ پروب کمتر از ۰/۸ ولت
PULSE LED :چشمک میزند (به مدت ۰/۳ ثانیه)	ولتاژ پروب تغییر حالت میدهد

۷) وضعیت بابهای کامپیوتر میزبان

همه کامپیوترهائی که میتوانیم بورد کاربردی را به آنها متصل نمائیم دارای تعدادی بابهای ورودی - خروجی موازی بصورت زیر می باشند :

کامپیوتر	باب A (ورودی/خروجی)	باب B (خروجی)	باب A (کنترل)	باب B (کنترل)
MPF-1B	80 H	81 H	82 H	83 H
MPF-1P	A0 H	A1 H	A2 H	A3 H
MPF-1/88	FFA0 H	FFA1 H	FFA2 H	FFA3 H
FLT 68K	$008000 H	800800013	800800005	00800007
BBC	FE60(user)	FE61(printer)	FE62	FE63

برای کامپیوترهائی که دارای آی سی CTC میباشند،

بیتهای ۳ و ۴ باب A بصورت زیر متصل میشوند :

بیت ۴ از باب A : به کانال صفر CTC

بیت ۳ از باب A : به کانال یک CTC

کانکتور چهل پایه

باب A		
بیت	جهت	کار
۷	ورودی	بیت ۷ برابر یک برای گردش دستی موتور بطور مستقیم

۶	"	بیت۶برابریک‌برای گردش‌دستی موتوربطورمعکوس
۵	"	بیت۵ برابریک‌برای روشن کردن هیتر
۴	"	هنگام روشن بودن موتور - پالس‌منفی
۳	"	هنگام روشن بودن ADC -خروجی مقایسه‌کننده
۲	"	بیت۲ ازباب A
۱	"	بیت۱ ازباب A
۰	"	بیت‌صفرازباب‌A

یا

۷	خروجی	کنترل موتورازطریق میکروکامپیوتردرجهت‌مستقیم
۶	"	کنترل موتورازطریق میکروکامپیوتردرجهت‌معکوس
۵	"	کنترل هیترازطریق میکروکامپیوتر
۴	ورودی	پالس‌منفی موتور
۳	"	خروجی‌مقایسه‌کننده‌هنگام روشن بودن ADC
۲	ورودی-خروجی	ـــــ
۱	"	ـــــ
۰	"	ـــــ

باب B

۷	خروجی	LED ۷	ورودیهای مبدل‌دیجیتال‌به آنالوگ
۶	"	LED ۶	
۵	"	LED ۵	
۴	"	LED ۴	
۳	"	LED ۳	
۲	"	LED ۲	
۱	"	LED ۱	
۰	"	LED ۰	

پایه های J1

شماره پایه	سیگنال	شماره پایه	سیگنال
1	NC	21	LED 0
2	NC	22	LED 1
3	NC	23	LED 2
4	NC	24	LED 3
5	NC	25	LED 4
6	NC	26	LED 5
7	MFWD/S7	27	LED 6
8	MREV/S6	28	LED 7
9	HTR/S5	29	NC
10	REVPUL/S4	30	GND
11	GND	31	REVCNT
12	ADC/S3	32	ADCCMP
13	S2	33	NC
14	S1	34	NC
15	S0	35	NC
16	NC	36	NC
17	NC	37	NC
18	NC	38	NC
19	NC	39	NC
20	NC	40	NC

آزمایشهای

APPLICATIONS BOARD

مقدمه ای بر سیستم کامپیوتر

هدف اصلی از این آزمایشها معرفی میکروپروفسور و بورد کاربردی آن به شما میباشد. کلیه کارهائی که شما باید انجام دهید بطور کامل و با جزئیات شرح داده شده است. بطوریکه برای یادآوری میتوانید براحتی به آنها مراجعه کنید. اگر برای اولین بار است که میخواهید با کامپیوتر میکروپروفسور کار کنید سعی کنید با دقت زیادی پیش بروید. سیستم کاملی که شما با آن کار میکنید دارای توانائیهای متنوعی میباشد بطوریکه آزمایشهای متفاوتی را در سطوح مختلف میتوان با آن انجام داد.

پس از انجام آزمایشهای موجود در این کتاب ایده بسیار خوبی از کار بورد کاربردی (Applications Board) خواهید داشت.

شروع کار

۱) برای شروع کار احتیاج به وسایل های زیر دارید:

- میکروپروفسور
- بورد کاربردی
- منبع تغذیه ۹ ولت و ۰/۵ آمپر
- کابل تخت چهل سوزنه

توجه کنید که میکروپروفسور باید شامل تراشه Z80 PIO باشد.

۲) - در میکروپروفسور را بیرون آورده و در طرف چپ آن بورد را قرار دهید.

- با دقت یکی از کابلهای تخت را به کانکتور چهل پایه موجود روی برد و کانکتور چهل پایه نزدیک تراشه PIO متصل کنید. (به شکل ۱-۳) توجه کنید.

- منبع تغذیه بورد و همچنین منبع تغذیه میکروپروفسور را متصل کنید.

۳) با [illegible] در بورد سایشگرها از ست را است به چپ انتقال پیدا میکند و همزمان با [illegible] ۸ عدد چراغ رنگی مربوطه روی بورد روشن میشود.

توجه کنید که در هنگام کار صحیح و طبیعی دستگاه، رادیاتور روی بورد نزدیک سوکت تغذیه داغ است و درجه حرارتی برابر با ۷۰ درجه سانتیگراد را دارا میباشد. لذا از دست زدن به آن بپرهیزید.

۴) همیشه برای برگشتن به وضعیت ابتدائی سیستم، کافی است که کلید RESET را فشار دهید.

آزمایش اول
وارد کردن برنامه

مفاهیم کلی

در این آزمایش چگونگی واردکردن یک برنامه در حافظه به شما معرفی میشود، همچنین اجرا و اصلاح برنامه موجود در حافظه را خواهید گرفت. سافر از اینکه شما همه دستورات میکروپروسسور را میشناسید، از توضیح اضافی در این مورد خودداری میشود.

برنامه اول

برنامه اول را هرگاه که کامپیوتر را روشن میکنید وارد حافظه کنید و سپس اجرا نمائید. توسط این برنامه PIO موجود روی میکروپروفسور طراحی میشود و بدون آن تعداد کمی از آزمایشهای شما جواب میگیرد.

آدرس	اطلاعات	دستورات
1C00	3E	LD A،FFH
1C01	FF	
1C02	D3	OUT (82H)،A
1C03	82	
1C04	D3	OUT (82H)،A
1C05	82	
1C06	D3	OUT (83H)،A
1C07	83	
1C08	3E	LD A،00
1C09	00	
1C0A	D3	OUT (83H)،A
1C0B	83	
1C0C	C3	JP 0000
1C0D	00	
1C0E	00	

اگر تاکنون با میکروپروفسور کار نکرده اید قسمتهای زیر را به دقت انجام دهید:

الف) وارد کردن برنامه

برای واردکردن برنامه اول به ترتیب زیر عمل کنید :

۱- کلیدهای [ADDR] وسپس [1]،[C]،[0]،[0] را به ترتیب از راست به چپ فشار دهید .

آدرس 1C00 درحافظه RAM ، یعنی جائی که برنامه اول قرار میگیرد می باشد .

نمایشگرها را به صورت 1.C.0.0. مشاهده می کنید .

۲- کلید [DATA] را فشار دهید ، در این صورت دو نقطه روشن به سمت راست نمایشگرها منتقل می شود .

۳- کلیدهای [3] وسپس [E] را فشار دهید ، این در واقع اولین بایت از برنامه میباشد .

۴- کلید [+] را فشار دهید روی نمایشگرها آدرس بعدی یعنی 1C01 را مشاهده میکنید .

۵- حال به همین صورت ادامه دهید یعنی با فشار دادن کلید [+] و ظاهر شدن آدرس جدید میتوانید اطلاعات مربوطه را وارد حافظه کنید .

در صورتیکه اطلاعاتی را اشتباه وارد کرده اید میتوانید با فشار دادن کلید [-] و رفتن به آدرس قبلی آنرا تصحیح نمائید .

۶- قبل از اجرای برنامه خود سعی کنید آنرا یکبار امتحان نمائید .

ب) اجرا کردن یک برنامه

۱- اگر برنامه در حافظه قرار دارد با فشار دادن کلیدهای زیر به ترتیب از چپ به راست میتوانید آنرا به اجرا درآورید : [ADDR] [1] [C] [0] [0] [GO]

۲- چراغهای رنگی روی بورد کاربردی خاموش می شوند و روی میکروپروفسور پیام [illegible] را مشاهده میکنید .

حال برنامه دوم را به عنوان تمرین وارد حافظه نمائید :

برنامه دوم

در برنامه زیر اطلاعات از باب ورودی 80H وارد شده و یک واحد به آن اضافه میگردد ، سپس نتیجه از طریق باب خروجی 81H نمایش داده میشود :

آدرس	اطلاعات	دستورات
1800	DB	IN A,(80H)
1801	80	
1802	C6	ADD A,01
1803	01	

1804.....D3.....OUT (81H)،A

1805.....81

1806.....C3.....JP 1800H

1807.....00

1808.....18

پس از وارد کردن برنامه در حافظه ،به طریقی که قبلا توضیح داده شد ، آنرا اجرا نمائید .

وضعیت سوئیچهای بابت ورودی 80H را عوض کنید و نتیجه را روی بابت 81H مشاهده کنید . عدد باینری موجود روی بابت خروجی باید یک واحد بیش از عدد باینری ورودی باشد .

ج)تصحیح کردن یک برنامه

در صورتیکه مجبور باشید برنامه خود را تصحیح کنید میتوانید بصورت زیر عمل کنید :

۱)کلید [ADDR] را فشار دهید و آدرس مورد نظر را وارد کنید . سپس کلید [DATA] را فشار داده و اطلاعات صحیح را وارد کنید . کلید [+] را برای ادامه فشار دهید .

۲)برنامه دوم را به صورتی تصحیح کنید که اطلاعات ورودی به جای جمع شدن با عدد [با عدد 07 جمع شود . حال برنامه اصلاح شده را اجرا کنید .

۳)در این مرحله ، برنامه را طوری تصحیح نمائید که به جای جمع اطلاعات ورودی با 07 ، تفریق انجام بگیرد . نتیجه به چه صورت است ؟

تمرین :

برنامه سوم

آدرس	اطلاعات	دستورات
1800.....	DB80.....	IN A،(80H)
1802.....	FE0F.....	CP 0FH
1804.....	CA0E18.....	JP Z،180EH
1807.....	3EFF.....	LD A،FFH
1809.....	D381.....	OUT (81H)،A
180B.....	C30018.....	JP 1800H
180E.....	3E00.....	LD A،00H
1810.....	D381.....	OUT (81H)،A

```
1812.....C30018.....JP 1800H
```

۱)برنامه سوم را واردحافظه کرده واجرانمائید.

۲)آیامیتوانیدبگوئیداین برنامه چه عملی انجام میدهد؟

۳)سعی کنیدباتعویض اطلاعات ورودی حالتهای مختلف را آزمایش نمائید.

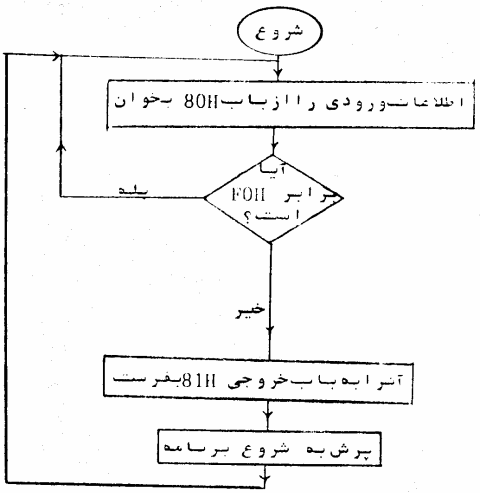
آزمایش دوم
تصمیم گیری

مفاهیم کلی

یکی از دلایل اصلی اهمیت کامپیوتر در دنیای جدید، توانائی کامپیوتر به ساده کردن کارها و تصمیم گیری برنامه ریزی شده می باشد. تصمیم گیری پس از یک عمل ریاضی و یا منطقی با اجرای یک دستور شرطی و توسط میکروپروسسور انجام میشود. در واقع بر حسب اینکه اتفاق مورد نظر افتاده باشد یا نه، یک انتخاب صحیح و ساده انجام می شود.

تمرین

برنامه ای بنویسید که اطلاعات ورودی را از باب 80H خوانده و اگر برابر FOH باشد، آنرا در نظر نگیرد فرضا به شروع کار یعنی آنجائی که اطلاعات وارد میشود برگردد و برای هر اطلاعات دیگری آنرا به باب خروجی 81H بفرستد.



در این مثال، تصمیم گیری پس از مقایسه اطلاعات ورودی با عدد FOH صورت میگیرد و سپس اگر نتیجه کار صفر باشد پرش انجام می گیرد.

آدرس	کدهگزادسیمال	دستورات حفظی
1800.....	DB80.....	IN A,(80H)

```
1802........FEF0..............CP F0H
1804........CA0018............JP Z·1800H
1807........D381..............OUT (81H)·A
1809........C30018............JP 1800H
```

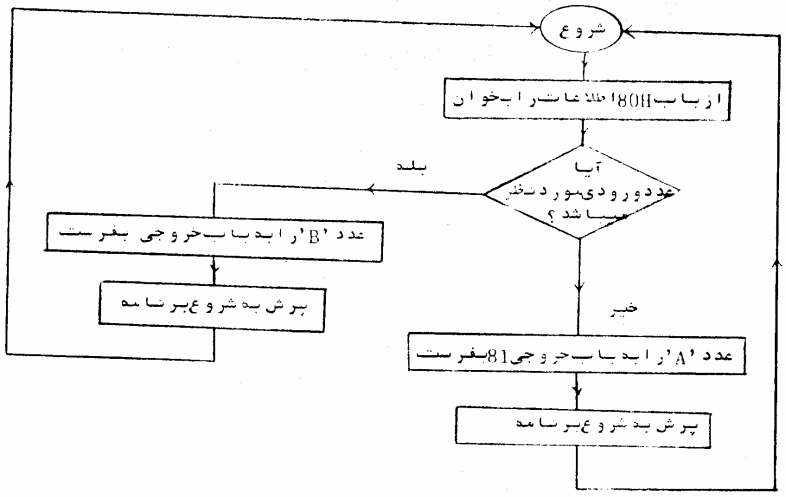
۱)برنامه را وارد حافظه نمائید و آنرا اجرا کنید. سپس به سوالات زیر پاسخ دهید:

الف- هنگامیکه اطلاعات ورودی F0H است چه اتفاقی می افتد؟

ب - چطور می توانید از خروج عدد 33H روی باب خروجی جلوگیری کنید؟

ج - اگر بخواهید فقط عدد 33H روی باب خروجی ظاهر شود، برنامه را چگونه اصلاح می کنید؟

بعضی اوقات با وجود یک شرط، یکسری اطلاعات و در صورت عدم وجود شرط، اطلاعات دیگری به باب خروجی فرستاده می شود. فلوچارت زیر این منظور را به خوبی نشان میدهد:



۲)به سوالات زیر پاسخ دهید:

الف- برنامه ای بنویسید که ستد اولا اجرا شده و کارهای زیر انجام دهد:

اگر اطلاعات ورودی از باب 80H برابر 07 است عدد 'AAH' را به باب خروجی 81H بفرستد در غیر اینصورت عدد '55H' به باب خروجی 81H فرستاده شود.

ب- برای یک دروازه NAND هشت به یکسید انید که هر ترکیبی از ورودیها باعث ایجاد یک '1'

منطقی درخروجی NAND میشود مگر درموقعیکه همه ورودیها '1' باشند. دراین صورت خروجی '0' میباشد. حال شما برنامه ای بنویسید که بتوانید با استفاده از سوئیچهای پاب 80 به عنوان ورودی و یکی از چراغهای پاب 81 به عنوان خروجی، یک دروازه NAND هشت به یک را شبیه سازی نمائید.

ج - برنامه ای بنویسید که یک دروازه NOR هشت به یک را شبیه سازی کند.

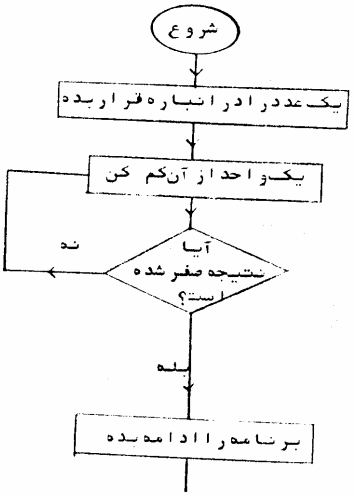
اگر بخواهید برنامه ب را تبدیل به NOR کنید چگونه آنرا اصلاح می کنید؟

آزمایش سوم
حلقه‌های تاخیر

استفاده از حافظه برای ذخیره کردن اطلاعات

مفاهیم کلی

شاید بیشترین هنر کامپیوتر ، بالابودن سرعت آن ، یعنی انجام میلیونها کار در یک ثانیه ، باشد . ولی گاهی اوقات بخصوص وقتی که انسان به عنوان یک سیستم جانبی کنار کامپیوتر قرار میگیرد این سرعت فوق العاده زیاد اشکال ایجاد کرده و باید سعی شود تا بطریقی کامپیوتر را کندتر نمائیم . حلقه های تاخیر اغلب به این منظور استفاده می‌شوند . یعنی با استفاده از این حلقه ها بطور خیلی ساده کامپیوتر برای یک مدت زمان مشخص منتظر می‌ماند ، معمولا در این مواقع کامپیوتر شمارش معکوسی را از یک عدد بزرگ به صفر آغاز می‌نماید . فلوچارت زیر یک چنین تاخیری را ایجاد می‌نماید :



بزرگترین عددی که می‌توان در انباره قرار داد عدد FFH می‌باشد و یا در واقع ۲۵۵ دسیمال ، چونکه کامپیوتر خیلی سریع کار میکند . بزرگترین تاخیری که با این روش ایجاد می‌شود حدود ۲ میلی‌ثانیه می‌باشد و آن موقعی است که عدد 00 را در انباره قرار دهیم . چرا ؟

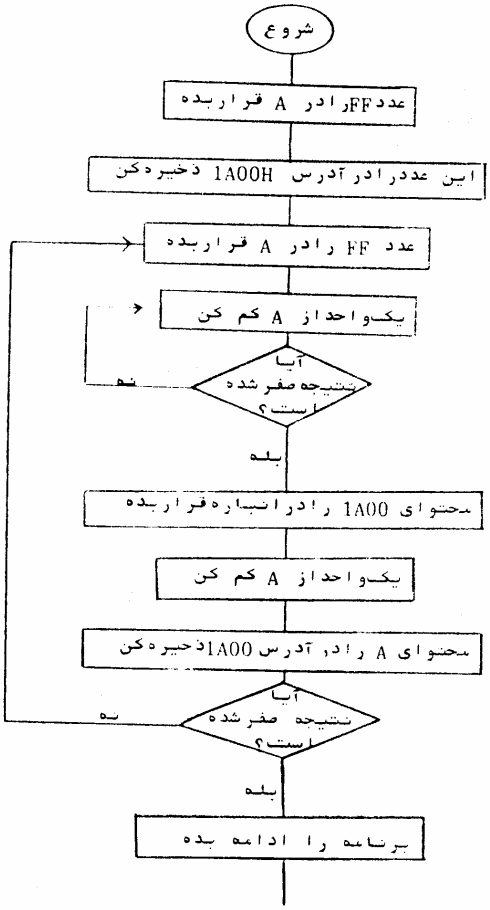
برنامه زیر به منظور ایجاد تاخیر نوشته شده و از آدرس 1820H شروع می‌شود . اگر برنامه را از آدرس دیگری شروع می‌کنید دقت کنید که آدرس پرش آنرا نیز بتحو مناسبی تغییر دهید :

آدرس	کدهگزادسیمال	دستورات
1820	3EFF	LD A,FFH
1822	3D	DEC A
1823	C22218	JP NZ,1822H

توجه کنید که قبل از دستور JP NZ,1822 هیچ مقایسه ای لازم نیست زیرا دستور DEC A خودش روی پرچم ZERO بطور مناسب اثر میگذارد.

واضح است که یک تاخیر طولانی را می توان با ایجاد حلقه های تاخیر داخل یکدیگر بوجود آورد، برای مثال میتوان با تکرار ۲۵۵ بار تاخیر ۲ میلی ثانیه ، تاخیر نیم ثانیه را ایجاد نمود.

به فلوچارت زیر توجه کنید. چونکه در حین برنامه از انباره استفاده می شود برای حفظ محتوای آن عددی که در آن قرار دارد را در آدرس 1A00H حافظه نیز نگهداری می کنیم.



تمرین

۱)فلوچارتی رارسم کنیدوطبق آن برنامه‌ای بنویسید کدکارهای زیررا انجام دهد:

الف- اطلاعاتوورودی را ازباب 80H واردکند.

ب- اطلاعاتراادرآدرس 1B00 ذخیره کند.

ج - تاخیری به اندازه ۰/۵ ثانیه ایجادکند.

د - اطلاعاتموجود درآدرس 1B00 را دراختیاره قرار دهد.

هـ - اطلاعاتراه به باب 81H بفرستد.

و - به شروع برنامه پرش کند.

در حین اجرای برنامه چه اتفاقی می افتد؟

۲)برنامه‌ای بنویسید که پس ازاجراشدن،چراغهای بابخروجی 81H بافرکانس 1HZ(نیم ثانیه روشن ونیم ثانیه خاموش)چشمکبزنند.

بجای اینکه همه‌چراغهاباهم خاموش‌و روشن شوند برنامه‌راطوری تصحیح کنید که:

الف- چراغهای فرد باهم وچراغهای زوج‌باهم روشن وخاموش‌شوند.(باهمان فرکانس)

ب- چهارچراغ دستراستوچهارچراغ دستچپ‌روشن وخاموش‌شوند.

آزمایش چهارم
استفاده از پروب لاجیکی

مفاهیم کلی

سیگنالهای موجود در یک سیستم میکروکامپیوتری می‌توانند در یکی از سه حالت زیر باشند :

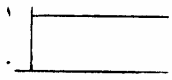
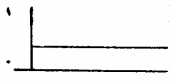
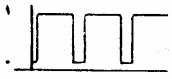
پائین (Low) <------ صفر منطقی <---- ولت ۰/۸ تا صفر

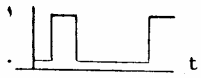
(Tri-state) امپدانس بالا <---- ولت ۲/۴ تا ۰/۸

بالا (High) <------ یک منطقی <---- ولت ۵ تا ۲/۴

پروب لاجیکی برای سنجش ولتاژ هر نقطه از مدار طراحی شده است. به LED قرمز مربوط به پروب لاجیکی به صورتهای HIGH ، LOW و PULSE نامگذاری شده اند. روشن شدن HIGH LED مبین یک ولتاژ بالای ۲/۴ ولت می‌باشد که در واقع با یک منطقی مطابقت می‌کند و روشن شدن LOW LED یک ولتاژ پائین یعنی زیر ۰/۸ ولت را نشان می‌دهد و در واقع صفر منطقی را نشان میدهد.

PULSE LED در مواقعی که یک موج وجود دارد، یعنی ولتاژ با سرعت صفر و یک میشود. خاموش و روشن میگردد. در دیاگرامهای زیر، کار پروب لاجیکی بهتر شرح داده می شود :

	شکل موج	نورهای پروب	توضیحات
۱)		HIGH ● LOW O PULSE O	HIGH روشن، یک منطقی و پایدار
۲)		HIGH O LOW ● PULSE O	LOW روشن، صفر منطقی و پایدار
۳)		HIGH O LOW O PULSE O	همه خاموش و خروجی امپدانس بالا
۴)		HIGH ● LOW O PULSE ✸	HIGH روشن و پایدار، PULSE چشمک میزند
۵)		HIGH O LOW ● PULSE ✸	LOW روشن و پایدار، PULSE چشمک میزند

۶) 

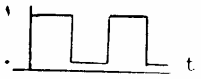
HIGH ⊖
LOW ●
PULSE ✸

HIGH سیمه روشن ، LOW روشن ، PULSE چشمک زن

۷) 

HIGH ●
LOW ⊖
PULSE ✸

HIGH روشن ، LOW سیمه روشن ، PULSE چشمک زن

۸) 

HIGH ⊖
LOW ⊖
PULSE ✸

HIGH سیمه روشن ، LOW سیمه روشن ، PULSE چشمک زن

بطور کلی اندازه روشنائی نورهای HIGH و LOW دلالت بر این دارد که شکل موج فرستاده شده در کدام حالت است. نور پالس با همان نرخ تغییر شکل موج چشمک میزند و سرعت فرستاده شدن موج مشکلی ایجاد نمی‌کند و حتی به بیش از ۳ پالس در ثانیه می‌تواند برسد.

توسط پروب لاجیکی شکل موجهای روی مسیر عمومی خطوط اطلاعات و آدرس CPU را میتوان امتحان و آزمایش نمود.

تمرین

الف) امتحان کردن پروب لاجیکی

در این قسمت آزمایش، شما چگونگی امتحان کردن پروب لاجیکی را با آزمایش کردن یک شکل موج شناخته شده خواهید آموخت:

۱- بورد کاربردی را به میکروکامپیوتر وصل کنید و هر دو را روشن نمائید.

۲- پروب لاجیکی را به سر کنتور مربوطه متصل کنید.

۳- با قرار دادن پروب بر روی سرهای تغذیه ولتاژ ۵ ولت مدار مجتمع در قسمت پائین بورد، می‌توانید نورهای HIGH و LOW را سریعا آزمایش کنید.

۴- حال سرهای تغذیه وزمین Z80 یعنی سرهای شماره ۱۱ و ۲۹ را امتحان کنید.

۵- سر ۶ از Z80 را امتحان نمائید. این سر ورودی CLOCK مربوط به Z80 می‌باشد. می‌بینید که هر دو نور HIGH و LOW روشن هستند. (البته ممکن است یکی از آنها پر نورتر باشد) و نور پالس چشمک میزند. شکل موج CLOCK یک موج مربعی با فرکانس بالا میباشد.

۶- خطوط آدرس و اطلاعات Z80 را امتحان کنید.

- هنگامیکه پیام [illegible] روی نمایشگرها است کدام یک از خطوط مسیر عمومی آدرس پالس فعالی را نشان نمی دهد؟

۷ - بالاخره سر ۲۶ ورودی RESET را امتحان کنید،توجه کنیدکه وقتی کلید [RESET] زده میشودچه اتفاقی می‌افتد ؟

ب)فعال کردن خطوط با اجرای برنامه

فعال شدن خطوط آدرس ،اطلاعات و یا کنترل در میکروکامپیوتر بستگی به برنامه ای دارد که در سیستم اجرا می شود.برای مثال اگر فقط آدرس مورد استفاده در برنامه 1800،1801،1802 باشد، خطوط A13 ، A14 و A15 همیشه بصورت صفر منطقی (در حین اجرای برنامه)می‌باشند .سیگنالهای کنترل تولیدشده نیز همیشه به اجرای دستورات استفاده شده بستگی دارند.در Z80 چهار سیگنال کنترل اصلی وجود دارد :

$\overline{MREQ}$: (Memory Request) - برای تقاضای حافظه

$\overline{IORQ}$: (Input/Output) - برای تقاضای بابهای ورودی / خروجی

$\overline{RD}$: (Read) - برای خواندن

$\overline{WR}$: (Write)- برای نوشتن

وجود خط روی این اسامی دلالت براین دارد که حالت فعال شدن آنها فعال صفر می‌باشد.به این معنی که در شرایط معمولی در حالت 'یک' منطقی قرار دارند و هنگام مثلا خواندن دستور $\overline{RD}$ بصورت 'صفر' منطقی میباشند .

اگر برنامه ای شامل هیچ دستوری که کار نوشتن WRITE را ایجاب می‌کند .نباشد این خط کنترلی در همه زمانها باید به صورت 'یک' منطقی باقی بماند .

برنامه اول

آدرس	کدهگزادسیمال	دستورات
1800.....	C30018.....	JP 1800H

۱ - در برنامه اول فقط یک دستور پرش دارد که پرش به خودش می‌کند .کد و آدرس ها را بصورت باینری بنویسید تا بروشنی وضعیت صفر و یک بودن خطوط آدرس و اطلاعات را در هر بایت مشاهده نمائید .

توجه کنید که خطوط آدرس و اطلاعات هیچکدام در وضعیت منطقی مشابهی هستند و هیچکدام فعال نیستند .

۲ - برنامه را وارد و اجرا کنید .

۳ - حال یک لیست از خطوط آدرس ،اطلاعات، $\overline{MREQ}$ ،$\overline{IORQ}$ ،$\overline{RD}$ و $\overline{WR}$ بنویسید .

۴ - با استفاده از پروب لاجیکی حالت هر کدام از خطوط لیست شده را امتحان کنید و در لیست

کنار هم بنویسید. آیا این حالتها با آنچه که انتظار داشتید مطابقت دارد؟

هر اختلافی بصورت زیر توضیح داده می شود:

a)خطوط مسیر عمومی آدرس A0 تا A6 برای آدرسهای دیگری که حافظه های دینامیکی را تعیین دستورات، تازه (REFRESH) می کنند. استفاده می شود.

b)خطوط آدرس در حالت امپدانس بالا (T-stste) در قسمت پائین (Low) T-state قرار می گیرند.

c)خطوط اطلاعات در حالت امپدانس بالا خود را به طرف بالا (High) می کشند.

پس از این برنامه اول را اجرا کنید. حالت خطوط کنترلی را در لیست خود مشخص کنید.

یک برنامه ساده ای بنویسید و آزمایش کنید که:

وقتی دستور HALT اجرا میشود روی مسیر عمومی چه اتفاقی می افتد؟

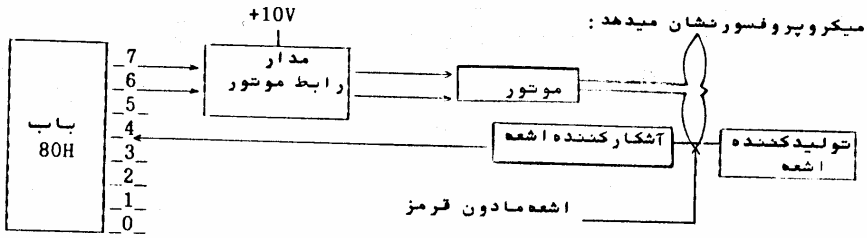
وقتی کلید RESET را فشار می دهید. چه اتفاقی می افتد؟

آزمایش پنجم

موتور

مفاهیم کلی

دیاگرام زیر چگونگی ارتباط موتور و مدار اطراف را بطور بلوک دیاگرامی با میکروپروفسور نشان میدهد :



توجه کنید که همه ارتباطات مدار موتور با باب 80H میباشد. این باب خود نیز به ۸ سوئیچ ورودی متصل میشود که از طریق این سوئیچها میتوان موتور را بصورت دستی کنترل نمود ولی اگر موتور تحت کنترل کامپیوتر باشد، این سوئیچها باید همگی در وضعیت ۱ باشند.

باب 80H یک باب قابل برنامه ریزی می باشد، به این معنی که هر بیت میتواند بصورت ورودی و یا خروجی برنامه ریزی شود. در این آزمایش بیت ۶ و ۷ بصورت خروجی مورد استفاده قرار میگیرند. در این مواقع با فرستادن بایتهای خاصی به باب 80H ، موتور را میتوان توسط کامپیوتر کنترل نمود. توجه داشته باشید که فقط بیتهای ۶ و ۷ به صورت خروجی عمل میکنند هرچند که یک بایت کامل به باب 80H فرستاده شود و بیتهای صفر تا ۵ میتوانند هرچه باشند اگرچه معمولا آنها را صفر در نظر میگیرند.

7 6 5 4 3 2 1 0	
0 0 × × × × × ×	متوقف
1 0 × × × × × ×	مستقیم
0 1 × × × × × ×	معکوس
1 1 × × × × × ×	متوقف

در برنامه زیر باب A در حالت اولیه خاصی قرار میگیرد و موتور با یک دستور ساده کنترل میشود :

برنامه اول

```
آدرس      کدهگز          دستورات
1800.....3EFF.....INIT:LD A،FFH
1802.....D382.....OUT (82H)،A
1804.....3E3F.....LD A،3FH
1806.....D382.....OUT (82H)،A
1808.....3E80..... FWD:OUT (80H)،A
180A.....D380.....OUT (80)،A
180C.....76.....HALT
```

در این برنامه پس از اجرا ،توسط چهار دستور اول ،باب 80H درحالت اولیه ای مناسب برای کنترل موتور قرار میگیرد و پس از اجرای سه دستور آخر موتور در جهت مستقیم شروع به چرخش میکند .

تمرین

الف- کنترل ساده موتور

۱)ابتدا مطمئن شوید که همه سوئیچهای باب 80H در وضعیت یک هستند و در ضمن همه سوئیچهای موجود روی بورد به جز سوئیچ موتور خاموش (OFF) میباشند . (سوئیچ موتور روشن)

۲)برنامه اول را وارد حافظه کرده و اجرا کنید .

موتور با سرعت زیاد شروع به چرخش میکند .

۳)برای متوقف کردن موتور میتوانید سوئیچ کنترل موتور را در حالت (OFF) قرار دهید .

برای برگشتن به برنامه مونیتور کلید [MONI] را فشار دهید .

۴)برنامه اول را طوری تغییر دهید که پس از اجرا شدن موتور بایستد .

ابتدا مطمئن شوید که سوئیچ موتور در حالت (ON) است .

۵)کد کنترل موتور را بنحوی تغییر دهید که پس از اجرای برنامه موتور بصورت معکوس بچرخد .

۶)توجه کنید که قسمت برنامه ریزی باب در برنامه ،فقط کافیست دفعه اول اجرا شود و در بقیه موارد کنترل موتور احتیاجی به برنامه ریزی باب نمیباشد .

- به برنامه اول نگاه کنید در آدرس 1808 کدامیک از بایتها و کدام بیت است که باعث چرخش مستقیم موتور میشود ؟

- برای چرخش معکوس موتور از چه بایتی باید استفاده نمود ؟

ب- بوجود آوردن یک تاخیر زمانی

در اغلب مواقع لازم است که میکروپروسسور کار مشخصی را در یک مدت زمانی کاملا ثابت و معلوم

انجام دهد .برای اینکار با استفاده از یک دستور شروع میکروپروسسور را وادار به شمارش حلقه - های تاخیر میکنیم .این روش در آزمایشهای قبلی توضیح داده شده است ولی در این آزمایش کمی ساده تر شده است.

به برنامه زیر توجه کنید :

```
        LD B,00
LOOP :  DEC B
        JP NZ,LOOP
```

این برنامه ساده شمارشی را شروع کرده و باعث ایجاد حدود ۲ میلی ثانیه تاخیر میشود .مدت زمان دقیق با محاسبه تعداد حلقه ها (در اینجا ۲۵۶) و کل زمانی که اجرای هر حلقه طول میکشد ،بدست می آید . اصولا مدت زمان اجرای هر حلقه بستگی به تعداد سیکلهای زمانی (T-States) هر دستور دارد فرضا در مثال بالا :

```
LD B,00      - 7   T-states
DEC B        - 4   T-states
JP NZ,LOOP   -10   T-states
```

بنابراین در این برنامه : T-states ۷+۲۵۶(۴+۱۰)=۳۵۹۱

با توجه به اینکه فرکانس پالس ساعت (clock rate) در میکروپروسسور ۱/۷۹ میباشد پس هر سیکل زمانی .۵۶ نانوثانیه طول میکشد و کل زمان برای اجرای یکبار حلقه برابر :

۳۵۹۱ × ۵۶.=۲/۰۱ میلی ثانیه

برای ایجاد تاخیرهای زمانی طولانی تر میتوان از یک زوج ثبات به عنوان شمارنده حلقه استفاده نمود .فقط باید به این امر توجه شود که دستور DEC که یک واحد از ثبات کم میکند برای زوج ثباتها روی پرچم اثر نمی گذارد ولی با استفاده از روشی مثل زیر میتوان این مشکل را برطرف نمود :

```
        LD BC,0000
LOOP :  DEC BC
        LD A,B
        OR C
        JP NP,LOOP
```

با استفاده از دستورات سوم وچهارم درصورتیکه محتوای BC به صفر برسد بر پرچم Z اثر گذاشته

واز طبقه حارج میشود. تعداد سیکلهای زمانی (T-states) دستورات بالا به اینصورت میباشد:

LD BC،0000	-10
DEC BC	- 6
LD A،B	- 4
OR C	- 4
JP NZ،LOOP	-10

۱) اندازه تاخیری که این برنامه بوجود می‌آورد را محاسبه کنید و سپس با اجرای برنامه دوم آنرا امتحان نمائید.

برنامه دوم

آدرس	کدهگز	دستورات
1800	3EFF	INIT:LD A،FFH
1802	D382	OUT (82H)،A
1804	3E3F	LD A،3FH
1806	D382	OUT (82H)،A
1808	3E80	FWD:LD A،80H
180A	D380	OUT (80H)،A
180C	010000	LD BC،0000
180F	0B	DEL:DEC BC
1810	78	LD A،B
1811	B1	OR C
1812	C20F18	JP NZ،DEL
1815	3E00	LD A،00H
1817	D380	OUT (80H)،A
1819	76	HALT

۲) برنامه دوم را وارد حافظه کرده و آنرا اجرا نمائید. آیا به اندازه مدت زمانی که انتظار دارید این برنامه انجام میشود؟ اگر اینطور نیست آدرس دستور پرش را امتحان کنید.

۳) با استفاده از یک دستور XOR میتوانید بایت مربوط به چرخش موتور را هر بار عوض کنید. با استفاده از چه دستوری میتوانید بیتهای ۰ و ۷ را بر عکس کنید بدون اینکه روی بیتهای صفر تا ۵ اثری

بگذارد؟

برنامه دوم رابه نحوی اصلاح کنیدکه :

a)باب 80H رادرحالت اولیه خاص کارموتورقراردهد.

b)موتور راوادارابه چرخش بطورمستقیم نماید.

c)برای یک مدت زمانی منتظربماند (چرخش ادامه پیداکند)

d)جهت موتوربرعکس شود

e)به بند c بپرد

توجه کنیدکه شمابایدبایت کنترل موتورراد رشباتی که دربرنامه تاخیر استفاده نمیشود،ذخیره نمائید.

تمرین

پس از اطمینان ازصحت کارموتور،میتوانیدتمرینهای زیررا انجام دهید:

۱)برنامه ای بنویسیدکه با اجرای آن ،موتوردرهرجهت۹ثانیه بچرخد.

۲)برنامه ای بنویسیدکه با اجرای آن،موتوررابتوان توسط بیتهای0،1 ازباب80H کنترل نمود.

توجه : برای انجام اینکارمیتوانیداطلاعات ورودی راازباب 80H واردکنیدوآنرابه طرف راست انتقال دهیدوسپس آنرابرای کنترل موتوراستفاده نمائید.

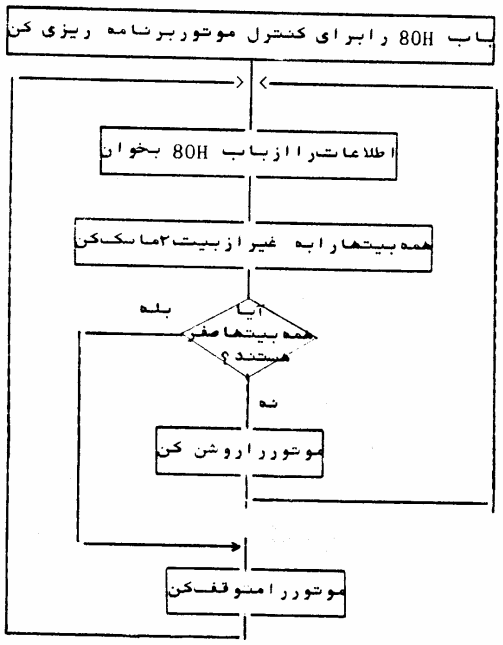
آزمایش ششم

ماسک‌کردن بیت‌وتست آن

مفاهیم کلی

برای‌یک‌سیستم‌مبتنی بر میکروپروسسور Z80 اطلاعاتی‌که ازباب‌ورودی وارد میشود ،۸بیت است‌وگاهی‌اوقات‌یک‌ویا‌تعدادی‌ازاین ۸بیت‌مهم ومورد‌نظر میباشندوبقیه باید‌درنظرگرفته نشوند .این پروسس‌ماسک‌کردن بیتهای غیرلازم نامیده میشودوبا استفاده ازدستورات‌منطقی میتوان اینکار راانجام‌داد .به روش‌مشابه میتوان یک‌ویا‌تعدادبیشتری ازبیتهای ورودی راامتحان‌کردوازصفرویایک‌بودنشان مطمئن شد .ماسک‌کردن بیتهامعمولابادستور AND برای صفرکردن بیتهای غیرلازم وبادستور OR برای یک‌کردن بیتهای غیرلازم انجام‌میگیرد .دراین موارد ابتدا کامپیوتر باید منتظر بماند تا اینکه اطلاعات‌ورودی برسدوسپس ماسک‌موردنظر را انجام دهد .

درفلوچارت‌زیر فقط وقتی موتورحرکت‌میکندکه بیت۲ ازباب 80H برابر یک‌منطقی باشد :



برای فلوچارت‌بالامیتوان برنامه‌زیر را نوشت :

برنامه اول

آدرس	کدهگز	دستورات
1800	3EFF	INIT:LD A،FFH
1802	D382	OUT (82H)،A
1804	3E3F	LD A،3FH
1806	D382	OUT (82H)،A
1808	DB80	START:IN A،(80H)
180A	E604	AND 04H
180C	CA1618	JP Z،STOP
180F	3E80	FWD:LD A،80H
1811	D380	OUT(80H)،A
1813	C30818	JP START
1816	3E00	STOP:LD A،00H
1818	D380	OUT (80H)،A
181A	C30818	JP START

تمرین

الف- تست‌کردن برای اطمینان ازوضعیت‌ورودی

۱)به‌جزسوئیچ‌موتورکه‌درحالت‌روشن‌قرارمیدهید،بقیه سوئیچهای بوردرادرحالت‌خاموش‌قرار دهید.

۲)برنامه‌اول رادرآدرس 1800 واردکنیدوسپس اجرانمائید.درصورتیکه بیت۲ازسوئیچ باب 80H برابر یک‌منطقی باشدموتورباید‌شروع به چرخیدن کند،باصفرویک‌کردن بیت۲ ببینیدچه اتفاقی می افتد،آیاصفرویک‌کردن بیتهای دیگرسوییچ درچرخش‌موتوراثری دارد؟

۳)برنامه خود راطوری تصحیح کنیدکه موتوربه جای کنترل بابیت۲،بابیت۳کنترل شود.

۴)برنامه‌رابنحوی تغییردهیدکه صفربودن بیت۳،باعث‌چرخش‌موتورویک‌بودن‌آن،باعث‌توقف موتورشود.باتعویض یک‌بایت‌میتوانیداینکاررا انجام دهید.

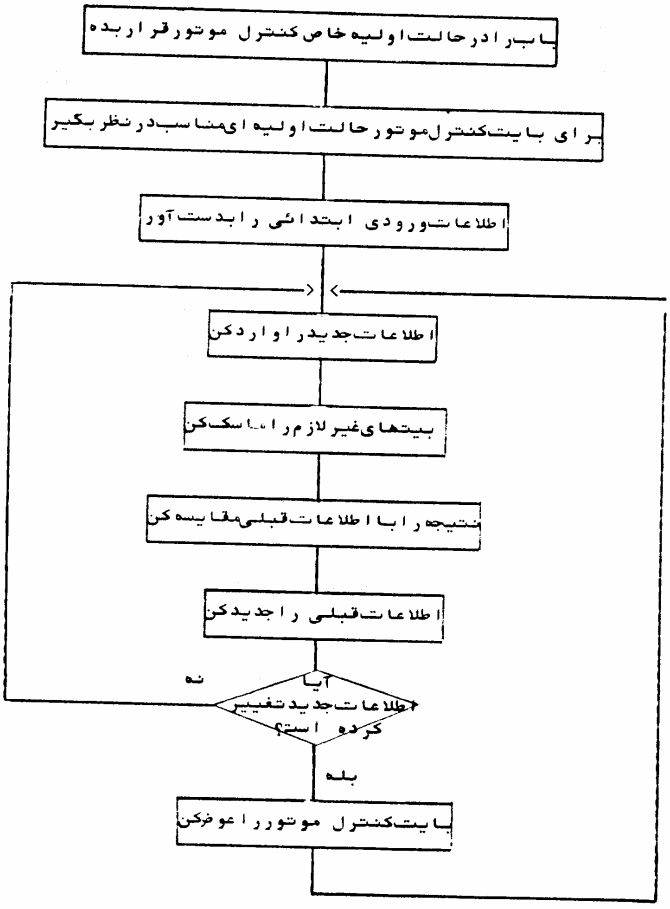
۵)اگربخواهیدیک‌بودن بیت۲یایک‌بودن بیت۳باعث‌چرخش‌موتورشود،برنامه‌راچگونه اصلاح

میکنید؟فرض کنید بخواهید صفر بودن هر دو بیت۲و۳موتور را متوقف کند.

۶)برنامه را به گونه ای اصلاح کنید که یک بودن بیت۲و یک بودن بیت۳موتور را بچرخاند.و اگر بیت۲یا بیت۳صفر است موتور متوقف شود.

ب-انتظار برای تغییر ورودی

اگر اطلاعاتی که از باب ورودی وارد میشود بطور پیوسته با اطلاعات قبلی که از همین باب آمده، مقایسه شود میتوان تشخیص داد که آیا اطلاعات عوض شده اند یا نه .درفلوچارت زیر اینکار انجام میشود و هرگاه اطلاعات ورودی عوض شوند موتور تغییر وضعیت میدهد:



دربرنامه زیر از شبات B به عنوان محل ذخیره اطلاعات قبلی و از شبات C برای ذخیره بایت کنترل

موتور استفاده شده است. در این برنامه موتور در صورتی تغییر وضعیت میدهد که یکی از بیتهای 0-3 از باب 80H عوض شده باشند.

برنامه دوم

```
1800.....3EFF.....INIT:LD A،FFH
1802.....D382.....OUT (82)H،A
1804.....3E3F.....LD A،3FH
1806.....D382.....OUT (82H)،A
1808.....0E80.....LD c،80h
180A.....DB80.....GETDAT:IN A،(80H)
180C.....E60F.....AND 0FH
180E.....47.....LD B،A
180F.....DB80.....LOOP:IN A،(80H)
1811.....E60F.....AND 0FH
1813.....B8.....CP B
1814.....47.....LD B،A
1815.....C21B18.....JP NZ،MOTA
1818.....C30F18.....JP LOOP
181B.....79.....MOTA:LD A،C
181C.....EE80.....×OR 80H
181E.....4F.....LD C،A
181F.....D380.....OUT (80H)،A
1821.....C30F18.....JP LOOP
```

حال کارهای زیر را انجام دهید:

۱) برنامه دوم را وارد حافظه کنید.

۲) مطمئن شوید که کلید کنترل موتور روشن است و سپس یکی از بیتهای 0-3 روی باب 80H را تغییر دهید. بقیه بیتها روی باب را عوض کنید، اگر آنطور که انتظار دارید پیش نمی رود، برنامه وارد

شده درحافظه را امتحان کنید .

سپس به سوالات زیر پاسخ دهید :

۳)دربرنامه دوم آیا وجود دستور LD C،80H لازم است؟بدون این دستور چه اتفاقی میافتد؟

۴)دربرنامه دوم آیا وجود دستورات زیر ضروری است؟

```
GETDAT:IN A،(80)H
       AND 0FH
       LD B،A
```

اگر این دستورات را حذف کنید چه اتفاقی میافتد؟

۵)چرا ثبات B دربرنامه دوم ،قبل ازدستور پرش JP NZ،MOTA بایدجدید شود؟

تمرین

تمرینهای زیر براساس برنامه دوم داده شده اند .شما میتوانید کامپیوتر خود را دقیقا برای این کارها برنامه ریزی کنید .

۶)برنامه ای بنویسید که درصورتیکه اطلاعات زیر درباب 80H قرار بگیرد ،عدد F0H را به باب خروجی 81H بفرستد :

صفر منطقی ـ بیت۳

یک " ـ بیت۴

صفر " ـ بیت۵

یک " ـ بیت۶

ودر غیر اینصورت عدد 0FH به خروجی فرستاده شود .

بقیه بیتهای ورودی میتوانند صفر و یا یک باشند .

برنامه را از آدرس 1800H شروع کنید .

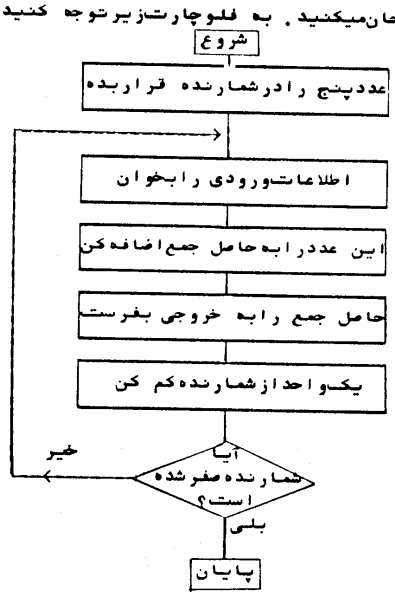
۷)برنامه ای بنویسید که هربار که اطلاعات ورودی از باب 80H تغییر میکند یک واحد به عدد باب خروجی 81H اضافه کند .این برنامه را از آدرس 1800 شروع کنید .

آزمایش هفتم

جمع کردن داده های ورودی

مفاهیم کلی

میتوان برنامه ای نوشت که پنج عدد باینری وارد شده از طریق سوئیچها را با یکدیگر جمع کرده ونتیجه را روی چراغهای رنگی نمایش دهد. واردکردن یک بایت اطلاعات از باب ورودی کار ساده - ای میباشد زیرا با یکبار استفاده از دستور IN میتوان اینکار را انجام داد. ولی در مواقعیکه لازم است پنج بایت متفاوت پشت سرهم وارد شوند این مشکل ممکن است پیش بیاید که در فاصله زمانی که کامپیوتر میخواهد بایت جدید را بخواند، این بایت آماده نشده باشد. در این صورت باید برنامه را بطریقی نوشت که کامپیوتر نیز به اندازه اپراتوریکه توسط سوئیچها اطلاعات را وارد میکند، سرعتش پائین بیاید. مثلا اگر یک تاخیر طولانی بین دستورات IN در برنامه قرار داده شود، اپراتور فرصت تعویض اطلاعات ورودی را بدست خواهد آورد، وحتی اپراتور میتواند با استفاده از یکی از کلیدهای موجود روی صفحه کلید، جدید شدن و آماده شدن داده های ورودی را به میکروپروسسور اعلام کند. و بالاخره یکی از بیتهای سوییچ ورودی، مثلا بیت هفتم، را میتوان به عنوان ورودی (STROBE) انتخاب نمود، و با فعال نمودن این بیت به میکروپروسسور نشان داد که اطلاعات جدید روی هفت بیت دیگر آماده خوانده شدن میباشد. در این برنامه شما برنامه را به هر سه روش نوشته و امتحان میکنید. به فلوچارت زیر توجه کنید.



چونکه درشمارنده عددپنج قرارداده شده استپس پنج بایت ازداده های ورودی بایکدیگر جمع میشوند .درضمن به دلیل اینکه دستور OUT درداخل حلقه است نتیجه جمع هر بایت ورودی باحاصل جمع قبلی روی چراغهای رنگی نمایش داده میشود .

برنامه اول

دراین برنامه ازثبات B به عنوان شمارنده استفاده میشود .زمان تاخیری حدود .1ثانیه که دربرنامه قرارداده شده است.این فرصت را به استفاده کننده میدهدکه بتواند عددموجودروی باب ورودی را عوض کند .درثباتهای E وD ،C عددمربوط به تاخیر قراردارد وثبات H مجموع هر جمع جاری رادرخودجای میدهد .دراین آزمایش فرض شده است که سرریز رخ ندهد .

آدرس	کدهگز	دستورات
1800	0605	SRART:LD B،05
1802	2600	LD H،00
1804	0E0A	DELAY:LD C،0A
1806	110000	DEL:LD DE،0000
1809	1B	DEL1:DEC DE
180A	7A	LD A،D
180B	B3	OR E
180C	C20918	JP NZ،DEL1
180F	0D	DEC C
1810	C20618	JP NZ،DEL
1813	DB80	MAIN:IN A،(80H)
1815	84	ADD A،H
1816	67	LD H،A
1817	D381	OUT (81H)،A
1819	05	DEC B
181A	C20418	JP NZ،DELAY

تمرین:

الف)حلقه باتاخیرطولانی

۱- برنامه اول را وارد حافظه کنید. مطمئن شوید که همه سوئیچهای روی بورد در حالت خاموش هستند.

۲- عدد 03H را روی باب 80H قرار دهید و برنامه را اجرا کنید.

تقریبا پس از ۵۰ ثانیه، چراغ HALT روشن میشود و عدد 0FH روی چراغهای رنگی نمایش داده میشود. اگر جواب صحیح را دریافت نکرده اید، برنامه را تست کنید.

۳- حال سعی کنید اعداد زیر را بعنوان ورودی وارد کنید:

30،1F،2A،05،81 جمع

نتیجه چیست؟

۴- عدد مربوط به تاخیر را عوض کنید بطوریکه بین هر دو داده ورودی ۳ ثانیه تاخیر ایجاد کند.

حال به سوالات زیر پاسخ دهید:

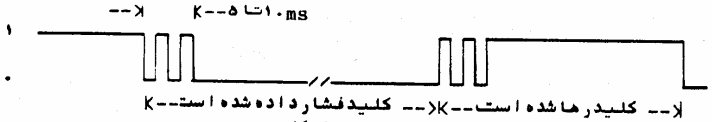
a) برای اینکه بخواهید بین سرعت اجرای برنامه و توانائی وارد کردن اطلاعات و ورودی هماهنگی رضایت بخشی را بوجود بیاورید، چند ثانیه تاخیر لازم است؟

b) مزیت و نقصان این روش وارد کردن اطلاعات به میکروکامپیوتر را بروشنی توضیح دهید.

ب) انتظار برای فشار داده شدن کلید:

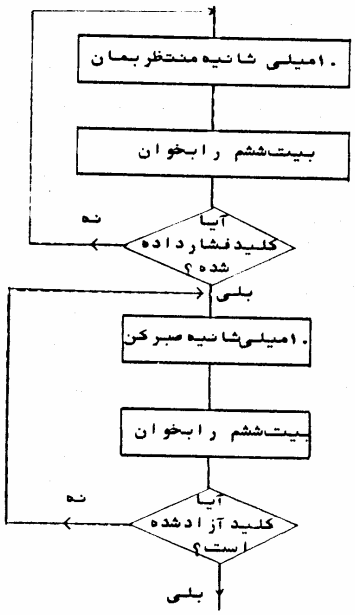
برای حذف بعضی از محدودیتهای روش قبلی، میتوان با استفاده از یکی از کلیدهای صفحه کلید آماده شدن اطلاعات روی سوئیچها را با فرستادن سیگنالی به میکروپروسسور خبر داد. با این روش تاخیر طولانی برداشته شده و میتوان اطلاعات را با هر سرعتی که مورد نیاز استفاده کننده است وارد نمود و برای اینکار احتیاج به یک ورودی میباشد.

در میکروپروفسور، صفحه کلید به باب ورودی شماره 00 وصل شده است و کلید USER مستقیما به بیت ششم از باب ورودی 00 متصل میباشد پس توسط این کلید میتوان آن سیگنال مورد نظر را ساخت وقتی که کلید USER آزاد باشد، بیت ششم یک منطقی و اگر فشار داده شود، این بیت صفر منطقی خواهد شد. هنگام پرش از صفر به یک و یا برعکس، در اثر لرزش کلید سیگنال تولید شده به سرعت صفر و یک میشود و اگر این مسئله مورد توجه قرار نگیرد باعث ایجاد اشتباه در تشخیص صفر و یا یک بودن بیت ششم خواهد شد. برای رفع این اشکال میتوان پس از هر بار زده شدن کلید و یا هنگام رها شدن کلید، مدت زمانی حدود ۵ تا ۱۰ میلی ثانیه صبر نمود و مجددا وضعیت را چک کرد (debouncing).



شکل ۱-۷: لرزش کلید

فلوچارت زیر برای رفع مشکل لرزش کلید مناسب میباشد :



به برنامه دوم توجه کنید :

برنامه دوم

آدرس	کدهگز	دستورات
1800	0605	START:LD B,5
1802	2600	LD H,00
1804	110005	DEL10M:LD DE,0500H
1807	1B	DEL10:DEC DE
1808	7A	LD A,D
1809	B3	OR E
180A	C20718	JP NZ,DEL10
180D	DB00	GETKEY:IN A,(00)
180F	E640	AND 40H
1811	C20418	JP NZ,DEL10M
1814	110005	DE10M:LD DE,0500H
1817	1B	DE10:DEC DE

```
1819.....B3.............OR E
181A.....C21718.........JP NZ,DE10
181D.....DB00.....GETKEY:IN A,(00)
181F.....E640...........AND 40H
1821.....CA1418.........JP Z,DE10M
1824.....DB80.....MAIN:IN A,(80H)
1826.....84.............ADD A,H
1827.....67.............LD H,A
1828.....D381...........OUT (81H),A
182A.....05.............DEC B
182B.....C20418.........JP NZ,DEL10
182E.....76.............HALT
```

حال کارهای زیر انجام دهید:

۱- برنامه دوم را وارد حافظه میکروپروفسور کنید.

اعداد 1، 2، 3، 4، 5 را با یکدیگر جمع کنید.

آیا این روش وارد اطلاعات، آسانتر از روش قبلی نیست؟

۲- اگر تاخیر مربوط به لرزش کلید را کوتاهتر انتخاب کنید، (مثلا یک میلی ثانیه) و یا آنرا حذف کنید چه اتفاقی میافتد؟

ج) انتظار برای پالس استراب (STROBE)

در صورتیکه صفحه کلیدی روی میکروکامپیوتر شما موجود نبود برای فرستادن سیگنال میتوانستید از روش دیگری استفاده کنید.

باینصورت که شما میتوانید یکی از بیتهای باب ورودی را به عنوان بیت استراب استفاده کنید. در چنین مواقعی طبیعی است که این بیت به عنوان بیت حاوی داده ورودی به حساب نمی آید و در واقع برای اطلاعات ورودی بقیه هفت بیت در نظر گرفته میشود. برای نمونه میتوان بیت هفتم را به عنوان بیت استراب استفاده کرد. این بیت دقیقا رفتار کلید USER را خواهد داشت. هنگامیکه بیت هفتم از صفر منطقی تبدیل به یک منطقی میشود، کامپیوتر داده ورودی را میخواند.

برنامه ای که از بیت هفتم باب 80H به عنوان بیت استراب استفاده میکند همان برنامه دوم قسمت (ب) میباشد با این تفاوت که آدرس باب ورودی مربوط به سیگنال عوض شده و برای داده

ورودی بیت‌هفتم‌درنظرگرفته‌نمیشود.

۱- برنامه‌دوم مربوط به‌قسمت(ب)رابدین صورت‌تصحیح نمائید:

ازبیت‌هفتم به عنوان‌پالس استراب‌نموده و آنرا ازمحاسبات‌حذف‌نمائید.

۲- اعدادهگزادسیمال زیررابایکدیگرجمع کنید: 41،03،1A،0F،2C

بعدازاینکه‌هرکدام‌ازاین‌اعدادراروی سوئیچهای باب80H قراردادیدبیت‌هفتم آنرا ابتدا 0

وسپس‌تبدیل به یک‌منطقی نمائید.

۳- برنامه راطوری تصحیح کنید که‌یکی‌ازهشت‌بیت‌ورودی به عنوان استراب‌طراحی شود.

آیاهیچ مشکلی بوجودنمیاید؟

۴- برنامه‌راطوری تصحیح کنیدعمل خواندن وجمع کردن وقتی انجام شودکه سوئیچ ازحالت

یک‌به 0 منطقی قرارمیگیرد.

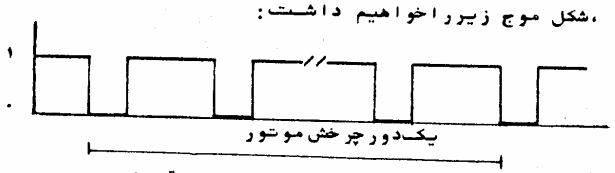
آزمایش هشتم

رویدادشمار

مفاهیم کلی

یک تصور کلی که از کامپیوتر وجود دارد این است که کامپیوتر اصولا منتظر رویدادی میماند و به محض اتفاق افتادن آن عکس العمل مناسب را نشان میدهد، و یا تعداد دفعاتی که آن رویداد رخ میدهد را شمارش میکند.

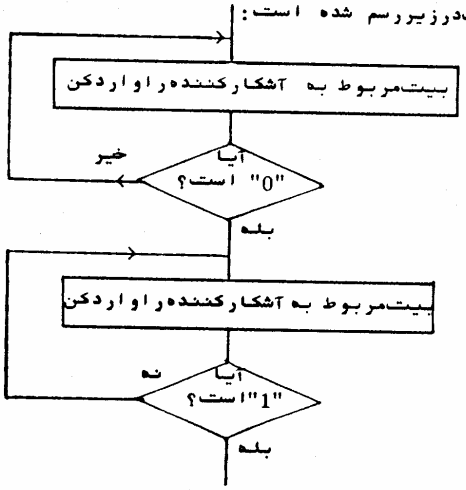
در این آزمایش پنکه n پره ای متصل به شافت موتور روی بورد هنگام چرخش موتور، با قطع اشعه مادون قرمز "0" منطقی و در مواقعیکه اشعه قطع نمیشود "1" منطقی ایجاد میکند. بنابراین همانطور که پره های چرخنده با هر یک دور چرخش موتور و در واقع یک دور چرخش پنکه و در واقع n بار قطع اشعه، شکل موج زیر را خواهیم داشت:



شکل ۱-۸)خروجی آشکارکننده اشعه مادون قرمز

برای اینکه بتوانیم عبور یک پره را متوجه شویم لازم است نرم افزاری مناسب نوشته شود و بدین صورت است که باید CPU منتظر بماند تا اینکه "0" منطقی ایجاد شود و سپس منتظر بماند تا اینکه "1" منطقی ایجاد شود و آنگاه متوجه میشود که یک پره اشعه را قطع نموده است شکل موج خروجی آشکارکننده به بیت چهار از باب A متصل است.

یک فلوچارت مناسب در زیر رسم شده است:



این فلوچارت برای آشکارکردن عبور یک پره است، درصورتیکه میتوان در برنامه اصلی ، با صدا کردن این برنامه ، چرخش یک دور پنکه (یک دور موتور) که برابر با n بار قطع اشعه توسط پره میباشد را مشخص نمود.

برنامه اول

در این برنامه درثبات B تعداد دفعات قطع اشعه قرار دارد:

```
REV:LD B,n
LOOP1:IN A,(80H)
      AND 10H
      JP NZ,LOOP1
LOOP0:IN A,(80H)
      AND 10H
      JP Z,LOOP0
      DEC B
      JP NZ,LOOP1
      RET
```

تمرین

زیربرنامه های زیر را بنویسید:

۱- زیربرنامه ای بنویسید که PIO را برای کنترل موتور در حالت اولیه ای مناسب قرار دهد. این زیربرنامه را MINI بنامید و آنرا در آدرس 1800H قرار دهید.

۲- زیربرنامه ای به نام FWD بنویسید که باعث چرخیدن موتور در جهت مستقیم شود. این زیربرنامه را در آدرس 1810H قرار دهید.

۳- زیربرنامه REV را در آدرس 1820H برای چرخش معکوس موتور بنویسید.

۴- زیربرنامه STOP را برای توقف موتور نوشته و در آدرس 1830H قرار دهید.

۵- زیربرنامه ای بنویسید که ایجاد تاخیر یک ثانیه کند و آنرا در آدرس 1840H قرار دهید.

۶- زیربرنامه ای بنویسید که با استفاده از برنامه اول ، یک دور چرخش موتور را بشمارد. این زیربرنامه را در آدرس 1850H وارد کنید.

شما میتوانید زیربرنامه های بالا را به دفعاتی که نیاز دارید در برنامه های اصلی خود فرا بخوانید. که اینکار را با استفاده از دستور CALL انجام میدهید.

برای مثال برنامه‌دوم را درنظربگیرید:

برنامه‌دوم

بافرض اینکه‌زیربرنامه‌های اتاغرانوشته‌باشیدبرنامه‌دوم را ملاحظه‌کنید:

```
MAIN:CALL 1800H   حالت‌اولیه‌به باب‌میدهد
LOOP:CALL 1810H   چرخش‌مستقیم موتور
     CALL 1850H   ادورچرخش
     CALL 1830H   توقف‌موتور
     CALL 1840H   اثانیه تاخیر
     CALL 1820H   چرخش‌معکوس‌موتور
     CALL 1850H   ادورچرخش
     CALL 1830H   توقف
     CALL 1840H   اثانیه تاخیر
     JP LOOP      تکرار
```

۷- برنامه‌دوم رابصورت‌کدماشین بنویسیدواز آدرس 1870H واردحافظه‌کنید.

۸- مطمئن شویدکه بیتهای‌عو۷ازسوئیچهای باب80H درحالت"1"منطقی میباشندوکلیدکنترل موتوروکلیدفیدبک‌موتورروشن است.

حال برنامه‌را اجراکنید.توجه‌داشته باشیدکه برنامه‌اصلی از آدرس 1870H شروع میشود.اگر پیام SYS-SP روی نمایشگرها ظاهرشد،یکباردیگرزیربرنامه راتست‌کنیدومطمئن شویدکه همه آنهابادستور RET خاتمه میابند.

تمرین :

۱- برنامه‌دوم‌راطوری تصحیح کنیدکه‌موتوردرهرجهت۵ دوربچرخد.برای اینکارلزومی ندارد که زیربرنامه یک‌دورچرخش‌را۵بارفرابخوانیدبلکه باقراردادن یک‌حلقه‌شمارش‌میتوانید اینکاررا انجام دهید.به‌هرحال‌دقت‌کنیدکه شمازیک‌ثبات‌به عنوان شمارنده‌بایداستفاده نمائید.

۲-اغلب‌اوقات‌لازم است‌که اعمال متفاوت،نتایج متفاوتی رانیزبه‌باربیاورند.برنامه حالت۱ راطوری اصلاح کنیدکه‌موتوربه تعداد دوری که عددموجودروی بیتهای 0-3 از سوئیچهای باب 80H نشان میدهد،بچرخد.این بدین معنی است‌که موتورمیتواندازا 0 تا

تا 15 دور در هر وضعیت بچرند. توجه کنید که اگر عدد صفر را روی این چهار بیت دادید، موتور نباید هیچ حرکتی بکند.

۳- همچنین شما میتوانید با قرار دادن عدد مربوط به تعداد دور مورد نظرتان در ثباتی مشخص، موتور را وادار به چرخش به همان تعداد دور نمائید. برای مثال ثبات E را در نظر بگیرید:

```
LD E,05
CALL 1870H      برنامه شمارش متغیر
```

حال برنامه اصلی را بصورتهای زیر مجددا بنویسید:

- ۱۰ دور چرخش مستقیم و سپس توقف برای یک ثانیه
- ۳ " معکوس " "
- ۲۰ " مستقیم " "
- ۶ " معکوس " "

گرد آوری وتنظیم :

حامد مظاهری

hamed@ir-micro.com