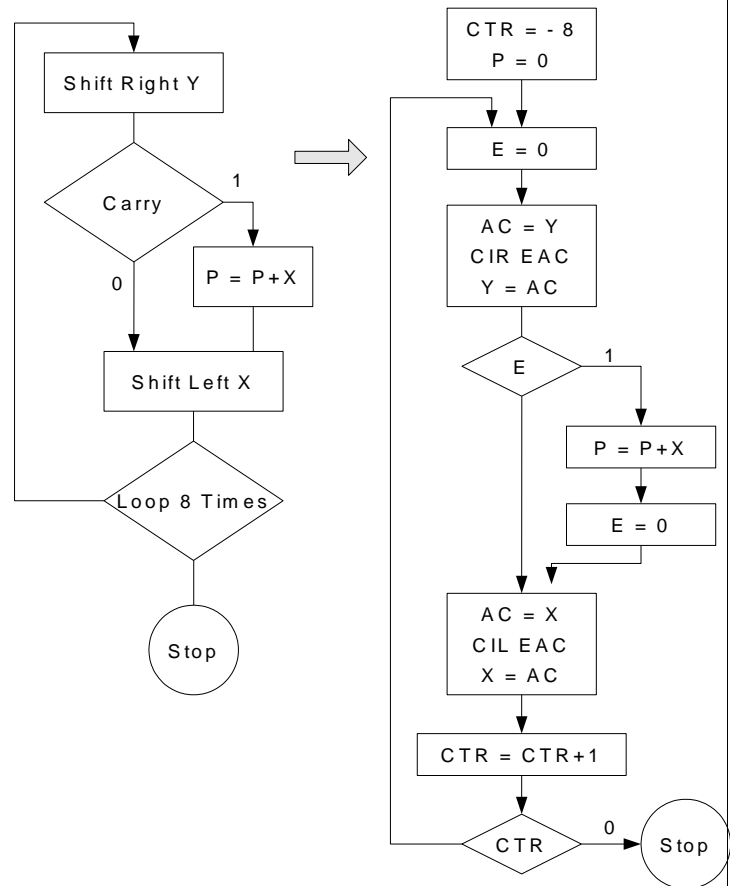


۳- برنامه ضرب دو عدد (P=X*Y) (جدول ۶-۱۴ کتاب)



```

    LOP,   ORG 100
           CLE
           LDA Y
           CIR           شیفت به راست
           STA Y
           SZE           چک کردن بیت نقلی شیفت
           BUN ONE
           BUN ZRO
    ONE,   LDA X
           ADD P
           STA P
           CLE
    ZRO,   LDA X
           CIL
           STA X
           ISZ CTR       افزایش شمارنده
           BUN LOP       اگر شمارنده صفر نباشد حلقه تکرار شود
           HLT
    CTR,   DEC -8
    X,     HEX 000F     عدد اول
    Y,     HEX 000B     عدد دوم
    P,     HEX 0        محل ذخیره حاصل ضرب
           END
    
```

۱- برنامه جمع اعضای یک آرایه (جدول ۶-۱۳ کتاب)

نکات:

- استفاده از مود غیر مستقیم برای داشتن اشاره گر
- نحوه ایجاد حلقه با دستور ISZ

1	ORG 100	شروع برنامه از آدرس ۱۰۰
2	LDA ADS	خواندن آدرس شروع لیست اعداد
3	STA PTR	ذخیره در اشاره گر
4	LDA NBR	خواندن تعداد اعدادی که باید جمع بزند
5	STA CTR	ذخیره در شمارنده
6	CLA	
7	LOP, ADD PTR I	مود غیرمستقیم برای داشتن اشاره گر
8	ISZ PTR	افزایش اشاره گر
9	ISZ CTR	افزایش شمارنده
10	BUN LOP	تکرار حلقه در صورت صفر نشدن شمارنده
11	STA SUM	
12	HLT	انتهای کار برنامه
13	ADS, HEX 150	آدرس شروع لیست اعداد
14	PTR, HEX 0	اشاره گر
15	NBR, DEC -100	تعداد اعداد (به صورت منفی)
16	CTR, HEX 0	شمارنده
17	SUM, HEX 0	محل ذخیره جمع نهایی
18	ORG 150	
19	DEC 75	عدد اول
.	.	
.	.	
118	DEC 23	عدد آخر
119	END	

۲- برنامه جمع دو عدد ۳۲ بیتی (C=A+B) (جدول ۶-۱۵ کتاب)

نکات:

- ابتدا ۱۶ بیت کم ارزش جمع زده می شود
- سپس ۱۶ بیت پر ارزش جمع زده می شود و بیت نقلی جمع قبل نیز به آن اضافه می شود

LDA AL	
ADD BL	جمع نیمه های کم ارزش
STA CL	ذخیره در نیمه کم ارزش پاسخ
CLA	
CIL	رقم نقلی را به آکومولاتور منتقل می کند
ADD AH	جمع نیمه پر ارزش با نقلی جمع قبل
ADD BH	جمع نیمه های پر ارزش
STA CH	ذخیره در نیمه پر ارزش پاسخ
HLT	
AL, HEX 1A2B	
AH, HEX 123C	
BL, HEX 234D	
BH, HEX AB32	
CL, -	
CH, -	

<p>۴- زیربرنامه و مبادله پارامترها با آن</p> <p>- مقدار بازگشتی را معمولا با آکومولاتور باز می گردانیم</p> <p>- پارامتر های دریافتی را اگر یکی باشد با آکومولاتور به زیر برنامه می دهیم</p> <p>مثال: زیر برنامه ای که یک عدد را بر هشت تقسیم کند:</p> <pre> DIV8 Return address CLE CIR CLE CIR CLE CIR BUN DIV8 I </pre>	<p>۶- ورودی/خروجی به روش Polling</p> <p>برنامه زیر پس از انتظار برای ورودی عمل ورودی انجام می دهد:</p> <pre> CIF, SKI چک کردن آماده بودن ورودی BUN CIF INP انجام ورودی در صورت آماده بودن STA CHR HLT CHR, -- </pre> <p>برنامه زیر پس از انتظار برای خروجی عمل خروجی انجام می دهد:</p> <pre> LDA CHR COF, SKO چک کردن آماده بودن خروجی BUN COF OUT انجام خروجی در صورت آماده بودن HLT CHR, HEX 0057 کد اسکی یک کاراکتر </pre>
<p>۵- تبادل پارامتر با زیر برنامه وقتی بیش از یک پارامتر داریم</p> <p>- اگر بیش از یک پارامتر داشته باشیم سایر پارامتر ها با روشی به نام Parameter Linkage منتقل می نماییم</p> <p>مثال: زیر برنامه ای که دو مقدار را با هم OR می کند (جدول ۶-۱۷)</p> <p>نکات:</p> <p>- یکی از پارامترها از طریق AC ارسال شده</p> <p>- پارامتر دیگر از طریق Parameter Linkage</p> <p>- در این روش پارامتر مورد نظر را بعد از دستور BSA قرار می دهیم</p> <p>- یعنی داده در لابلای کد</p> <p>- در سابروتین با یک دسترسی غیر مستقیم می توان به داده دسترسی یافت (چون آدرس خانه بعد از BSA را به عنوان آدرس بازگشت در اختیار داریم)</p> <p>- باید حواسمان باشد که قبل از بازگشت از زیربرنامه آدرس بازگشت را به تعداد پارامترهای لینک شده افزایش دهیم</p> <pre> ORG 200 LDA X خواندن پارامتر اول در آکومولاتور BSA OR پرش به سابروتین HEX 3AF6 پارامتر دوم بعد از دستور پرش STA Y ذخیره مقدار بازگشتی از سابروتین HLT X, HEX 7B95 Y, HEX 0 OR, HEX 0 محل ذخیره آدرس بازگشت از سابروتین CMA شروع کد سابروتین STA TMP LDA OR I دسترسی به پارامتر لینک شده CMA AND TMP CMA ISZ OR افزایش آدرس بازگشت BUN OR I بازگشت از سابروتین TMP, HEX 0 END </pre>	<p>۷- روتین سرویس دهی وقفه</p> <p>اگر ورودی آماده باشد کاراکتر ورودی را خوانده در بافر ورودی ذخیره می کند</p> <p>اگر خروجی حاضر باشد کاراکتر خروجی را خوانده در بافر خروجی ذخیره می کند</p> <pre> SRV Return Address STA SAC ذخیره آکومولاتور CIR STA SE ذخیره فلگ محاسباتی SKI چک کردن آماده بودن ورودی BUN NXT INP انجام ورودی در صورت آماده بودن STA PT1 I ISZ PT1 افزایش اشاره گر بافر ورودی NXT, SKO چک کردن آماده بودن خروجی BUN EXT LDA PT2 I OUT انجام خروجی در صورت آماده بودن ISZ PT2 افزایش اشاره گر بافر خروجی EXT LDA SE CIL بازیابی فلگ محاسباتی LDA SAC بازیابی آکومولاتور ION فعال سازی مجدد اینترپت BUN SRV I بازگشت از وقفه SAC, SE, PT1, PT2, </pre> <p>نکته: مشکل روش فوق این است که مرتبا اشاره گر بافرها افزایش می یابد. بهتر است که بافرها به صورت صف حلقوی طراحی شوند و توابعی (زیربرنامه هایی) جهت چک کردن خالی و پر بودن بافرها هم نوشته شود.</p>