خلاصه درس

معماری کامپیوتر-فصل ۵

تهیه کننده و گردآورنده:

BORNA66

Power By WWW.Borna66.IR

WWW.PNU-CLUB.COM

باشگاه علمی و آموزشی دانشجویان پیام نور

ارایه رایگان جدیدترین اخبار و نمونه سوالات امتحانی و جزوات درسی و منابع ارشد کلیه رشته ها ی پیام نور

==========

WWW. COMPUTER-ENG.IR <-> وب سایت مهندسی کامپیوتر <-> WWW.BORNA66.IR

✓ سیکل دستورالعمل

برنامه ای که در واحد حافظه کامپیوتر ذخیره شده از دنباله ای از دستورالعمل ها تشکیل شده است . این برنامه طی سیکل های اجرایی دستورالعمل اجرا می شود . هر سیکل دستورالعمل نیز خود به مجموعه ای از سیکل های جزئی یا فازها تقسیم می شود . در کامپیوتر پایه هر سیکل دستور متشکل از فازهای زیر است .

1- برداشت یک دستور از حافظه(Fetch)
2- دیکد کردن دستور(Decode)
3- خواندن آدرس موثر
4- اجرای دستورالعمل (Execute)

پس از پایان مرحله 4 ، کنترل به مرحله 1 بازگشته و برداشت ، دیکد و اجرای دستور بعدی را انجام می دهد . این فرآیند به طور نامحدود ادامه می یابد مگر اینکه کامپیوتر با دستور HALT مواجه شود .

✓ برداشت و دیکد :

در آغاز شمارنده برنامه ، PC، با آدرس اولین دستورالعمل برنامه بار می شود . شمارنده SC پاک شده و در نتیجه سیگنال T_0 را تولید می نماید . پس از هر پالس ساعت ، SCیک واحد افزایش می یابد تا رشته سیگنال های T_0 ، T_1، T_2 و غیره متوالیا" تولید گردند .ریز عمل های مربوط به فاز برداشت و دیکد را می توان توسط عبارات انتقال ثبات زیر مشخص نمود .

T_0 : AR ← PC

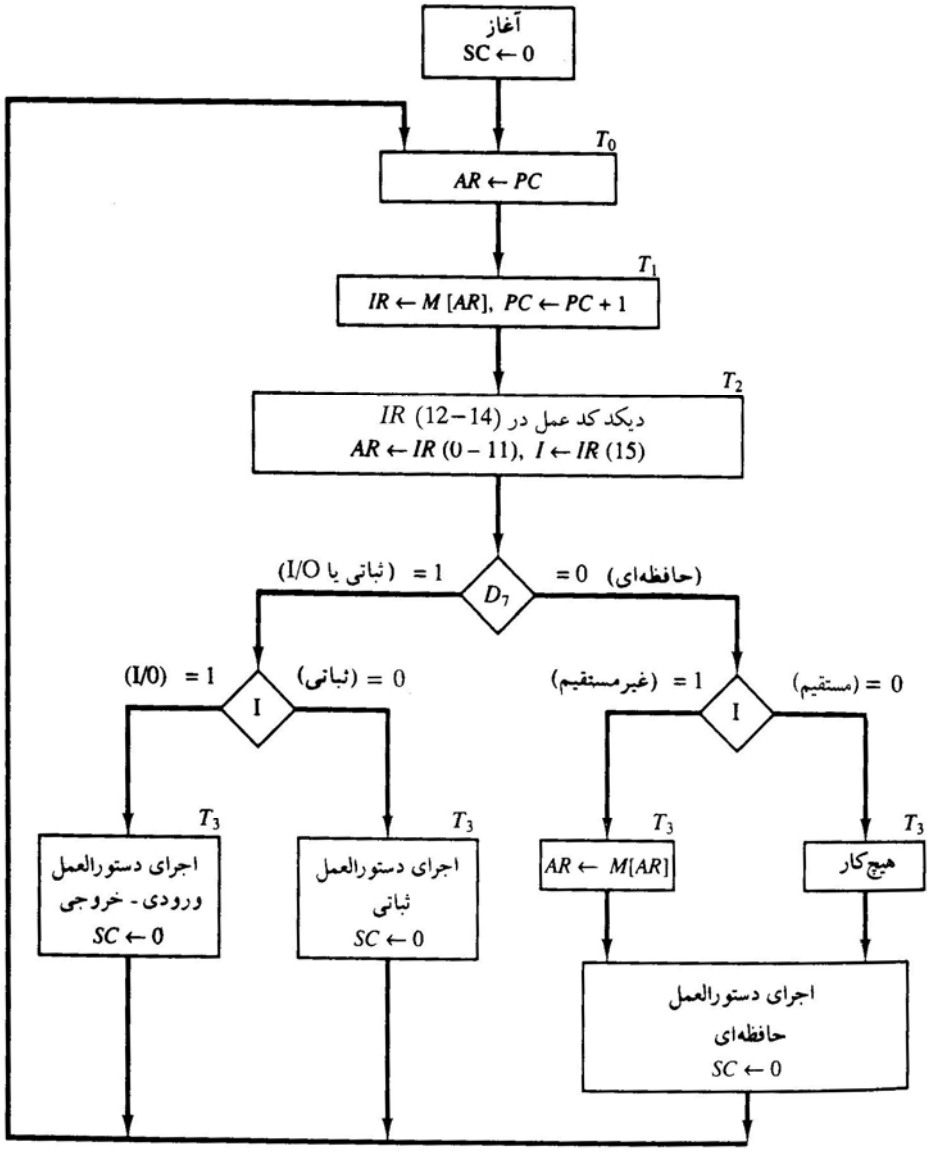
T_1 : IR ← ← $PC + 1$M[AR] , PC

T_2: D_0،.... D_7 ←دیکد ، AR ← IR(0-11) ، I ← IR(15)

WWW.PNU-CLUB.COM

✓ تعیین نوع دستورالعمل :

سیگنال زمانبندی بعدی که پس از دیکد شدن فعال است T_3 می باشد . در مدت زمان T_3 ، واحد کنترل نوع دستوری را که درست چند لحظه قبل خوانده شده معین می کند .



فلوچارت سیکل دستورالعمل (آرایش اولیه)

سه نوع دستورالعمل موجود به چهار مسیر فرعی جداگانه تقسیم می شوند . عمل انتخاب شده به وسیله سیگنال مرتبط با T_3 فعال می گردد که برای چهار مسیر توسط روابط زیر مشخص می شود .

$D'_7\ IT_3$: AR ← M[AR]

$D'_7 I' T_3$: هیچ کار

$D_7\ I' T_3$: اجرای دستور ارجاع به ثبات

$D_7\ IT_3$: اجرای یک دستور ورودی – خروجی

WWW.PNU-CLUB.COM

✓ دستورالعمل های ارجاع به ثبات :

این نوع دستورات وقتی که D_7=1 و I=0 است توسط کنترل قابل تشخیص اند. بیت های 0 تا 11 این نوع دستورات برای تعیین یکی از 12 دستورالعمل به کار می روند . این 12 بیت در HR(0-11) قرار دارند و در طول T_2 نیز بـه AR منتقـل شده اند . توابع کنترل و ریز عمل ها برای دستورات ارجاع به ثبات در جدول لیست شده اند .

اجرای دستورالعمل‌های ثباتی

$D_7I'T_3 = r$ مشترک در همه دستورالعملهای ثباتی			
$IR(i) = B_i$ بیتی در IR (0−11) که عمل را مشخص می‌کند			
	r:	$SC \leftarrow 0$	پاک کردن SC
CLA	rB_{11}:	$AC \leftarrow 0$	پاک کردن AC
CLE	rB_{10}:	$E \leftarrow 0$	پاک کردن E
CMA	rB_9:	$AC \leftarrow \overline{AC}$	متمم کردن AC
CME	rB_8:	$E \leftarrow \overline{E}$	متمم کردن E
CIR	rB_7:	$AC \leftarrow \text{shr } AC,\ AC(15) \leftarrow E,\ E \leftarrow AC(0)$	شیفت چرخشی به راست
CIL	rB_6:	$AC \leftarrow \text{shl } AC,\ AC(0) \leftarrow E,\ E \leftarrow AC(15)$	شیفت چرخشی به چپ
INC	rB_5:	$AC \leftarrow AC + 1$	افزایش AC
SPA	rB_4:	If $(AC(15) = 0)$ then $(PC \leftarrow PC + 1)$	گذر در صورت مثبت بودن
SNA	rB_3:	If $(AC(15) = 1)$ then $(PC \leftarrow PC + 1)$	گذر در صورت منفی بودن
SZA	rB_2:	If $(AC = 0)$ then $(PC \leftarrow PC + 1)$	گذر در صورت صفر بودن AC
SZE	rB_1:	If $(E = 0)$ then $(PC \leftarrow PC + 1)$	گذر در صورت صفر بودن E
HLT	rB_0:	$S \leftarrow 0$ S یک فلیپ‌فلاپ آغاز و پایان است	توقف کامپیوتر

جدول زیر هفت دستورالعمل ارجاع به حافظه را نشان می دهد . خروجی دیکد شده D_i برای i=0,1,2,3,4,5,6 متعلق به هر دستور نیز در جدول آورده شده است . آدرس موثر در ثبات AR و در مقطعT_2 به هنگام I=0 و یا در T_3 با I=1 جـای می گیرد .اجرای دستورهای حافظه ای با سیگنال زمانبندی T_4 شروع می شود . توصیف سـمبلیک هـر دسـتور هـم در جدول بر حسب علائم انتقال ثبات مشخص شده است . اجرای واقعی هر دستور در سیستم نیاز به یک رشـته ریـز عمـل دارد . دلیل این است که داده های ذخیره شده در حافظه نمی توانند مستقیما" پردازش شوند . این داده ها باید ابتـدا از یک حافظه خوانده شده و در ثباتی قرار گیرند که مدارهای منطقی بتوانند روی آنها عمل کنند . حـال مـا عملکـرد هـر دستور را توضیح داده و توابع کنترل و ریزعمل های لازم را برای اجرای آنها لیست می کنیم . فلوچـارتی کـه تمـام ایـن ریزاعمال را در اختیار می گذارد نیز در این بخش آمده است .

دستورالعمل‌های حافظه‌ای

سمبل	دیکدر عمل	توضیح سمبلیک
AND	D_0	$AC \leftarrow AC \wedge M[AR]$
ADD	D_1	$AC \leftarrow AC + M[AR],\quad E \leftarrow C_{out}$
LDA	D_2	$AC \leftarrow M[AR]$
STA	D_3	$M[AR] \leftarrow AC$
BUN	D_4	$PC \leftarrow AR$
BSA	D_5	$M[AR] \leftarrow PC,\quad PC \leftarrow AR + 1$
ISZ	D_6	$M[AR] \leftarrow M[AR] + 1$ If $M[AR] + 1 = 0$ then $PC \leftarrow PC + 1$

WWW.PNU-CLUB.COM

✓ AND با AC :

این دستورالعمل منطقی AND بر روی جفت بیت های AC و حافظه ای که توسط آدرس موثرش معین می شود را انجام می دهد. نتیجه عمل فوق در انتها به AC منتقل می شود . ریزعمل هایی که این دستورالعمل را اجرا می کنند عبارتند از:

D_0T_4 : DR ← M[AR]

D_0T_5 ← AC ∧ DR , SC ← 0

تابع کنترل این دستور خروجی دیکدر D_0 را به کار می برد زیرا این خروجی وقتی که عمل AND با کد 000 اجرا می شود فعال می گردد . دو سیگنال زمانبندی برای اجرای دستور فوق لازم است . لبه پالس مربوط به سیگنال T_4 ، عملوند را از حافظه به DR انتقال می دهد . لبه پالس بعدی یعنی T_5 ، نتیجه عمل AND بر روی AC وDR را به AC منتقل و SC را هم صفر می کند که این خود سبب واگذاری کنترل به سیگنال T_0 می گردد تا سیکل دستورالعمل جدیدی آغاز شود .

✓ ADD با AC :

این دستور محتوای حافظه مشخص شده توسط آدرس موثر را با مقدار AS جمع می کند. حاصل جمع به AC و رقم نقلی خروجی C_{out} به فلیپ فلاپ E (بیت گسترش انباره "Extended AC")منتقل می شود . ریزاعمال لازم برای این دستور عبارت است از :

D_1T_4 : DR ← M[AR]

D_1T_5 :←EAC ← $AC + DR$, C_{out} , SC ← 0

مشابه با دو دستور قبل دو سیگنال زمانی T_4 وT_5 به کار رفته اند ولی خروجی D_1 دیکدر به جای D_0 که در عمل AND به کار رفت فعال است . پس از برداشت و دیکد دستور تنها یک خروجی دیکدر فعال خواهد بود و این خروجی رشته ریزعمل هایی که واحد کنترل هنگام اجرای یک دستورالعمل حافظه ای دنبال می کند را تعیین می نماید .

✓ LDA : بار کردن AC

این دستور محتوای حافظه ای که توسط آدرس موثرش مشخص شده را به AC منتقل می کند .

D_2T_4 : DR ← M[AR]

D_2T_5 :AC ← DR , SC ← 0

با باز نگری مجدد شکل ثبات های کامپیوتر پایه متصل به یک گذرگاه مشترک ملاحظه می شود که مسیر مستقیمی از گذرگاه به AC وجود ندارد . مدار جمع کننده و منطقی ، اطلاعات را از DR دریافت می کند که قابل انتقال به AC است . بنابراین لازم است تا ابتدا کلمه حافظه به DR فراخوانده شده و سپس از DR به AC منتقل شود . دلیل عدم ارتباط گذرگاه به AC ، تاخیری است که در مدارهای جمع کننده و منطقی وجود دارد . در این جا فرض بر این است که زمان لازم برای خواندن از حافظه و انتقال به گذرگاه و مدارهای جمع کننده و منطقی بیش از یک پالس ساعت است . با عدم اتصال گذرگاه به ورودی های AC می توانیم زمان یک سیکل ساعت را برای هر ریزعمل حفظ کنیم.

WWW.PNU-CLUB.COM

✓ STA : ذخیره کردن AC در حافظه :

این دستور محتوای AC را در حافظه ای که با آدرس موثرش مشخص شده ذخیره مـی نمایـد . چـون خروجـی AC بـه گذرگاه وصل است و ورودی حافظه هم به گذرگاه متصل شده ، می توانیم این دستور را با یک ریزعمل انجام دهیم .

D_3T_4: M[AR] ← AC , SC ← 0

✓ *BUN* : انشعاب بدون شرط

این دستور برنامه را به دستورالعملی که توسط آدرس موثرش مشخص شده منتقل می نماید . به خاطر داشته باشید که *PC* آدرس دستوری که در سیکل بعدی خوانده می شود در خود نگه می دارد.*PC* در زمان T_1 افزایش می یابد تا برای دستور بعدی در برنامه آماده باشد . دستورالعمل BUN به برنامه نویس امکان می دهد تا دستوری را در خارج از رشته متوالی برنامه مشخص کند و در این حالت گوییم برنامه بدون شرط انشعاب یافته است . این دستورالعمل با یک ریزعمل اجرا می شود .

D_4T_4:PC ← AR , SC ← 0

آدرس موثر از AR و از طریق گذرگاه مشترک به PC منتقل می شود. با پاک شدن SC ، کنترل به T_0 منتقل می گردد . سپس دستور بعدی توسط مقدار جدید در PC از حافظه برداشت و اجرا می گردد.

✓ BSA : انشعاب با ذخیره آدرس برگشت :

این دستور برای انشعاب به بخشی از برنامه که زیرروال (Subroutine) یا رویه خوانده می شود مفید است .هر وقت این دستور اجرا شود BSA آدرس دستور بعدی را که در PC موجود است در حافظه ای که توسط آدرس موثر معین می شود ذخیره می نماید . سپس آدرس موثر به علاوه 1 به PC منتقل می شود تا به عنوان آدرس اولین دستور در زیر روال مورد استفاده قرار گیرد . این عمل در جدول دستورالعمل های حافظه ای همراه با عبارت انتقال ثبات در زیر آمده است .

M[AR] ← PC , PC ← AR+1

یک مثال عددی که نشان دهنده چگونگی به کار رفتن این دستور است در شکل زیر دیده می شود .فرض بر این است که دستور BSA در حافظه 20 قرار گرفته باشد . بیت 1 برابر 0 و بخش آدرس دستور 135 تصور می شود . پس از فازهای برداشت و دیکد ، PC حاوی 21 خواهد بود که آدرس دستور بعدی در برنامه است (این همان آدرس بازگشتی است) AR حاوی آدرس موثر 135 است . این مطلب در قسمت (الف) شکل ملاحظه می گردد . دستورالعمل BSA عمل عددی زیر را اجرا می نماید .

M[135] ← 21 , PC ← 135+1=136

نتیجه این عمل در بخش (ب) از شکل مزبور دیده می شود . آدرس بازگشت 21 در حافظه 135 ذخیره شده و کنترل از 136 زیرروال به کار خود ادامه می دهد . بازگشت به برنامه اصلی (در آدرس 21) توسط دستور BUN که در انتهای زیر روال قراردارد انجام می شود . وقتی این دستور اجرا شود کنترل آدرس موثر را که در مکان 135 قرار دارد و همان آدرس

WWW.PNU-CLUB.COM

21 است می خواند . پس از اجرای BUN ، آدرس 21 به PC منتقل می شود و در نتیجه دستور واقع در آدرس بازگشت اجرا خواهد شد .

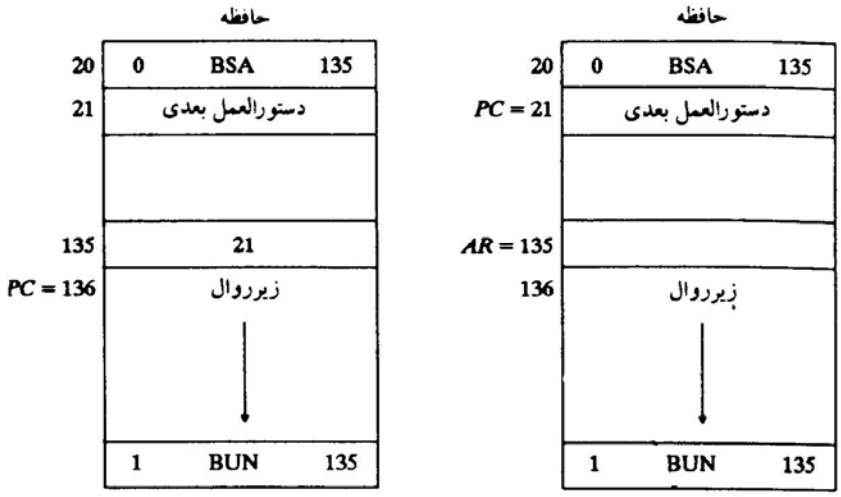
دستورالعمل BSA عملی را انجام می دهد که معمولا به آن فراخوانی (Call) زیرروال می گویید . دستور BUN در انتهای زیرروال عملی را که به آن بازگشت از زیرروال گویند انجام می دهد . در بسیاری از کامپیوترهای تجاری آدرس برگشت از زیرروال در یکی از ثبات های پردازشگر یا در بخشی از حافظه به نام پشته (Stack) ذخیره می شود .

امکان اجرای دستور BSA در یک پالس ساعت ، در صورتی که از سیستم گذرگاه کامپیوتر پایه استفاده کنیم ، میسر نیست . برای استفاده صحیح از حافظه و گذرگاه ، دستور BSA باید با دو ریزعمل زیر اجرا شود

D_5T_4: $M[AR] \leftarrow PC$, $AR \leftarrow$ AR+1

D_5T_5: $PC \leftarrow AR$, $SC \leftarrow 0$

سیگنال زمان بندی T_4 یک عمل نوشتن در حافظه را آغاز می کند و طی آن محتوای PC را روی گذرگاه قرار می دهد ، و ورودی INR مربوط به AR را فعال می کند. نوشتن در حافظه و افزایش AR تا لبه پالس بعدی تکمیل می گردد . در T_5 گذرگاه برای انتقال AR به PC مورد استفاده قرار می گیرد .



(ب) حافظه و PC پس از اجرا

(الف) حافظه، PC ، AR در زمان T_4

✓ ISZ : افزایش و گذر اگر نتیجه صفر

این دستور کلمه ای که توسط آدرس موثر مشخص می شود را یک واحد افزایش می دهد و اگر مقدار افزایش یافته 0 شود ، PC را یک واحد افزایش می دهد. برنامه نویس معمولا یک عدد منفی را (به فرم متمم 2) در حافظه قرار می دهد. این عدد منفی مرتبا افزایش می یابد و نهایتا به صفر می رسد . در این لحظه PC یک واحد افزایش و سپس به حافظه بازگردانده شود . این کار با ریزعمل های زیر انجام می شود :

D_6T_4 : $DR \leftarrow$ M[AR]

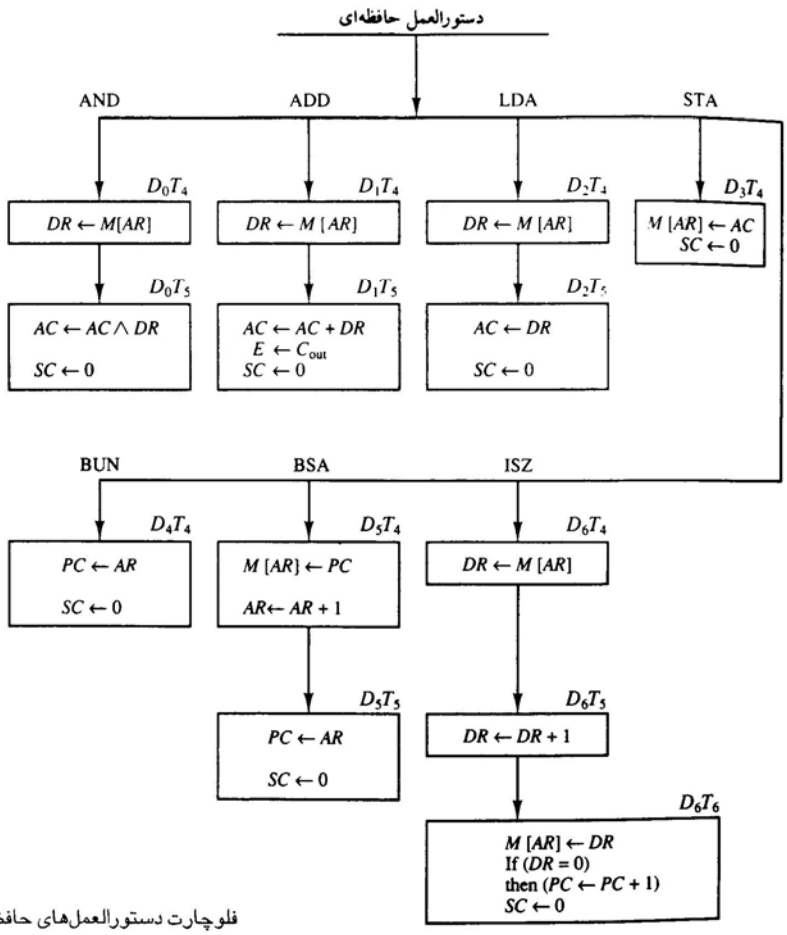
WWW.PNU-CLUB.COM

$D_6T_5 : DR \leftarrow DR + 1$

$D_6T_6 : PC \leftarrow DR$ *if (DR=0) then* $(PC \leftarrow PC + 1)$, $SC \leftarrow 0$

✓ فلوچارت کنترل

فلوچارت نمایش اجرای ریزعمل های مربوط به هفتدستورالعمل ارجاع به حافظه در شکل فلوچارت دستورالعمل های حافظه ای نشان داده شده است . تابع کنترل در بالای هر چهارگوشه آمده است . ریزعمل های اجراشده در T_4 ، T_5 و T_6 به مقدار کد عمل بستگی دارد . این مطلب در فلوچارت توسط شش مسیر مختلف مشخص شده که هریک از آنها پس از دیکد شدن دستورالعمل توسط کنترل انتخاب می شوند . شمارنده برنامه SC در هر حالت پس از آخرین سیگنال زمانبندی، 0 می شود . این امر باعث انتقال کنترل به سیگنال زمانبندی T_0 برای آغاز سیکل دستورالعمل بعدی می گردد.



فلوچارت دستورالعمل‌های حافظه‌ای

✓ رودی ، خروجی (Input,Output) و وقفه(Interrupt)

WWW.PNU-CLUB.COM

یک کامپیوتر مادامی که با محیط خارج خود ارتباط برقرار نکرده است نمی تواند مفید واقع شود . دستورات و داده های ذخیره شده در حافظه باید از برخی وسایل خارجی متصل به ورودی گرفته شوند . نتایج محاسبات هم باید از طریق نوعی وسیله خروجی به کاربر انتقال یابد . کامپیوتر های تجاری مجهز به انواع وسایل ورودی و خروجی هستند .

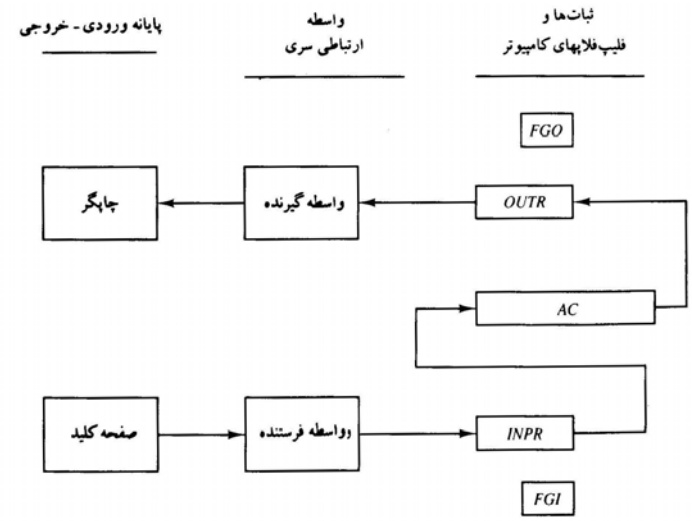
WWW.PNU-CLUB.COM

✓ آرایش ورودی ، خروجی

پایانه ، اطلاعات سری را ارسال و دریافت می کند . هر یک از کمیت های اطلاعاتی حاوی یک کدالفبا عددی هشت بیتی است . اطلاعات سری وارده از صفحه کلید به داخل INPR شیفت داده می شود . اطلاعات سری چاپگر در OUTR ذخیره می گردد . این دو ثبات با یک مدار واسط وسیله جانبی به طور سری و با AC ارتباط موازی دارند . آرایش ورودی،خروجی در شکل زیر دیده می شود . مدار واسط وسیله ارسال کننده (Transmitter Interface) (فرستنده) اطلاعات سری را از صفحه کلید دریافت و به INPR می فرستد . مدار واسط دریافت کننده (Reciever Interface) ، اطلاعات را از OUTR دریافت و به چاپگر ارسال می دارد.

ثبات INPR از هشت بیت تشکیل و اطلاعات الفباعددی را در خود نگه می دارد . پرچم یک بیتی FGI یک فلیپ فلاپ کنترل است و این پرچم هر وقت اطلاعات جدیدی در وسیله ورودی موجود باشد برابر 1 می شود و وقتی کامپیوتر این اطلاعات را پذیرفت برابر 0 می گردد . این پرچم برای هم زمان کردن سرعت جریان اطلاعات بین وسیله ورودی و کامپیوتر لازم است . فرآیند انتقال اطلاعات به قرار زیر است.

در ابتدا به پرچم ورودی FGI مقدار 0 داده می شود . وقتی کلیدی روی صفحه کلید فشار داده می شود ، یک کد الفباعددی هشت بیتی به طور سری به INPR شیفت پیدا کرده و پرچم FGI برابر 1 می گردد. مادامی که این پرچم 1 است محتوای INPR قابل تغییر به وسیله کلید دیگر نیست . کامپیوتر بیت پرچم را چک می نماید ، اگر پرچم 1 باشد ، اطلاعات از INPR به طور موازی به AC منتقل و FGI پاک می شود . به محض پاک شدن پرچم ، اطلاعات جدید حاصل از کلید جدید می تواند وارد INPR گردد . ثبات خروجی OUTR به طور مشابه کار می کند ولی جهت انتقال اطلاعات معکوس است . در آغاز پرچم خروجی FGO برابر 1 است . کامپیوتر این پرچم را چک می کند ، اگر برابر 1 باشد اطلاعات از AC به طور موازی به OUTR منتقل و FGO پاک می شود . وسیله جانبی خروجی اطلاعات کد شده را پذیرفته و کاراکتر مربوطه را چاپ و در پایان FGO را 1 می کند . کامپیوتر مادامی که FGO برابر 0 است اطلاعاتی را در OUTR ذخیره نمی کند زیرا صفر بودن FGO نشانه این است که خروجی در حال چاپ کاراکتر است .



آرایش ورودی- خروجی

WWW.PNU-CLUB.COM

✓ دستورات ورودی خروجی :

دستورات ورودی خروجی برای تبادل اطلاعات از ثبات AC ، و چک بیت های پرچم برای کنترل وقفه مورد نیازند. دستورات ورودی خروجی دارای کد عملیاتی 1111 بوده و هنگامی که $D_7 = 1$ و I= 1 باشد توسط کنترل قابل تشخیص اند. بیت های دیگر معین کننده دستورالعمل خاص مورد نظر هستند.توابع کنترل و ریزعمل های ورودی خروجی در جدول زیر لیست شده اند.

دستورالعمل‌های ورودی- خروجی

$D_7IT_3 = p$ (مشترک در همه دستورالعمل‌های ورودی - خروجی)
$IR(i) = B_i$ [بیتی در IR(0−11) که دستورالعمل را مشخص می‌کند]

	p:	$SC \leftarrow 0$	پاک کردن SC
INP	pB_{11}:	$AC(0\text{–}7) \leftarrow INPR, \quad FGI \leftarrow 0$	دریافت کاراکتر ورودی
OUT	pB_{10}:	$OUTR \leftarrow AC(0\text{–}7), \quad FGO \leftarrow 0$	ارسال کاراکتر خروجی
SKI	pB_9:	If $(FGI = 1)$ then $(PC \leftarrow PC + 1)$	گذر بر اساس پرچم ورودی
SKO	pB_8:	If $(FGO = 1)$ then $(PC \leftarrow PC + 1)$	گذر بر اساس پرچم خروجی
ION	pB_7:	$IEN \leftarrow 1$	روشن کردن فعال‌ساز وقفه
IOF	pB_6:	$IEN \leftarrow 0$	خاموش کردن فعال‌ساز وقفه

WWW.PNU-CLUB.COM

باشگاه دانشجویان دانشگاه پیام نور

www.pnu-club.com

- ارایه جدیدترین **اخبار و اطلاعیه های** مختلف مرتبط با دانشگاه پیام نور
- ارایه جدیدترین نمونه **سوالات امتحانی** در تمامی رشته های پیام نور بصورت رایگان
- بانک جامع نمونه سوالات در چندین نیمسال برای تمامی رشته ها **بصورت رایگان**
- ارایه جدیدترین **مقالات و مطالب علمی مرتبط با رشته های** تحصیلی پیام نور
- بحث و گفتگو و تبادل نظر **و پاسخ به سوالات** مختلف پیام نور و رشته های تحصیلی پیام نور
- ارایه جدید ترین **جزوات و منابع کارشناسی و کارشناسی ارشد** رشته های تحصیلی پیام نور
- ارایه جدیدترین **نرم افزار های کامپیوتری** و مباحث گوناگون آموزشی و علمی و سرگرمی و.....
- و ارایه هر آنچه که مرتبط با پیام نور و سایر موضوعات **تخصصی و عمومی مجاز و قابل بحث**

تهیه کننده و گردآورنده:

BORNA66

Power By WWW.Borna66.IR