

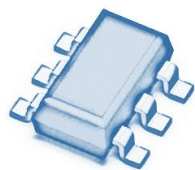


دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

۱۳۹۸/۳/۱

پاسخ تکلیف ۶ – انتقال ثبات ها و دستورات کامپیوتر پایه

معماری کامپیوتر



(۱) کامپیوتر پایه دارای ۶۴ ثبات و یک واحد حافظه اصلی با 512K کلمه ۳۲ بیتی است. بیت‌های انتخاب گذرگاه، ثبات‌های PC، AR، DR و AC را بنویسید.

۶۴ ثبات و یک واحد حافظه نیاز به ۷ بیت برای انتخاب باس دارد.

۲۱۹ خانه حافظه داریم. بنابراین ۱۹ بیت برای آدرسی دهی حافظه نیاز است. ثبات‌های آدرس دهی باید ۱۹ بیتی باشند.

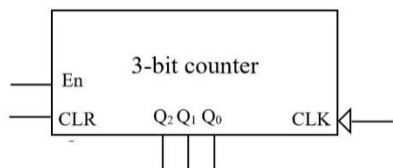
کلمه‌ها ۳۲ بیتی اند، ثبات‌هایی که با داده‌ها کار می‌کنند باید ۳۲ بیتی باشند.

Bus Select : 7 bits

PC, AR: 19 bits

DR, AC: 32 bits

(۲) بوسیله یک شمارنده باینری سه بیتی با بازنشانی همگام و با استفاده از اجزای زیر یک شمارنده طراحی کنید که دنباله‌ی زیر را بشمارد :



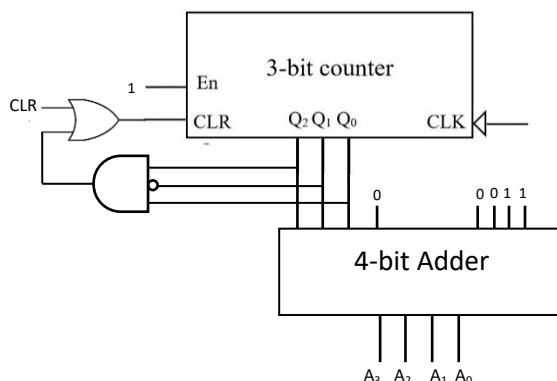
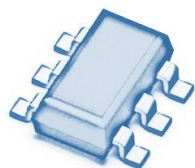
$3 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 11 \rightarrow 13 \rightarrow 3 \rightarrow 5$

در طرح خود می‌توانید از جمع‌کننده، ضرب‌کننده، شیفت‌دهنده و مدارهای منطقی پایه استفاده کنید.

ابتدا رابطه دنباله را بدست می‌آوریم.

$$2 \times 0 + 3 \rightarrow 2 \times 1 + 3 \rightarrow 2 \times 2 + 3 \rightarrow 2 \times 3 + 3 \rightarrow 2 \times 4 + 3 \rightarrow 2 \times 5 + 3 \rightarrow \dots$$

این رابطه به صورت $2x + 3$ است. پس در هر کلاک با دو برابر کردن (شیفت به چپ) و جمع با عدد سه (۰۰۱۱)، عددهای دنباله به دست می‌آیند. وقتی در شمارنده به عدد ۵ رسیدیم ($\overline{Q_2}Q_1\overline{Q_0}$) ($2 \times 5 + 3 = 13$) شمارنده را reset می‌کنیم تا دوباره شمارش اعداد را از ابتدا تکرار کند.



(۳) رجیستر R مقدار اولیه‌ی ۱۰۰۱۱۰۰ را دارد.

الف) این رجیستر پس از یک شیفت محاسباتی از راست و یک شیفت چرخشی از چپ، سپس یک شیفت منطقی از راست، چه مقداری به خود می‌گیرد؟ (مرحله به مرحله بیان کنید)

۱۱۰۰۱۱۰۰	<i>ashr</i>	پس از یک شیفت محاسباتی از راست :
۱۰۰۱۱۰۰۱	<i>cil</i>	پس از یک شیفت چرخشی از چپ :
۰۱۰۰۱۱۰۰	<i>shr</i>	پس از یک شیفت منطقی از راست :

ب) اگر بر روی رجیستر با همان مقدار اولیه خود یک شیفت محاسباتی از چپ اعمال شود، چه مقداری به خود می‌گیرد؟ آیا سرریز رخ می‌دهد؟

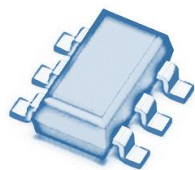
در حالت اولیه اگر یک شیفت محاسباتی از چپ داشته باشیم دچار سرریز می‌شود زیرا بیت علامت حفظ نمی‌شود.

(۴) ثبات در ابتدا مقادیر اولیه زیر را دارند :

AR	BR	CR	DR
11110010	11111111	10111001	11101010

بررسی کنید بعد از انجام هر یک از ریز عملگرهای متوالی زیر چه مقداری در هر کدام از این ثبات‌ها قرار می‌گیرد.

$AR \leftarrow AR + BR$	\rightarrow add BR to AR : 11110001
$CR \leftarrow CR \wedge DR$	\rightarrow and CR to DR : 10101000
$BR \leftarrow BR + 1$	\rightarrow increment BR : 00000000
$AR \leftarrow AR - CR$	\rightarrow subtract CR from AR : 00000000



۵) سیکل برداشت و آدرس دهی غیرمستقیم (Indirect) و اجرای دستورات BUN, BSZ, ISZ و INP را بنویسید. برای هر کدام از دستورات بالا در حد یک جمله توضیح دهید که چه کاری انجام می‌دهد و چه کاربردی دارد.

سیکل برداشت و آدرس دهی غیرمستقیم (Indirect): واکنشی دستور و بدست آوردن آدرس موثر

$$\begin{aligned} T_0: AR &\leftarrow PC \\ T_1: IR &\leftarrow M[AR], PC \leftarrow PC + 1 \\ T_2: &\text{Decode } IR(12 - 14), AR \leftarrow IR(0 - 11), I \leftarrow IR(15) \\ D_7IT_3: &AR \leftarrow M[AR] \end{aligned}$$

BUN (Branch Unconditional): این دستور بدون هیچ شرطی برنامه را به آدرس دستوری می‌برد که آدرس موثر آن مشخص شده باشد. (همانند goto) و مقدار PC را به آدرس مورد نظر تغییر می‌دهد.

$$D_5T_4: PC \leftarrow AR, SC \leftarrow 0$$

BSA (Branch and Save Return Address) این دستور به یک بخش از برنامه رجوع می‌کند. مکان فعلی اجرای برنامه را در آدرس مؤثری که با خود دارد ذخیره می‌کند، سپس PC را یکی بعد از آن آدرس موثر قرار می‌دهد. این دستور همانند call است و برای فراخوانی زیرروال و توابع کاربرد دارد.

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

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

$AR \leftarrow PC$ آدرس اولین دستوری است که بعد از بازگشت از تابع باید اجرا شود.

$AR + 1$ آدرس دستوری است که باید بعد از پرش اجرا شود (آدرس اولین دستور تابع زیرروال)

ISZ (Increment and skip if zero): این دستور برای افزایش یک واحدی کلمه ای است که آدرس موثر آن داده شده است.

مقدار کلمه مشخص شده توسط آدرس موثر در حافظه را یکی اضافه کرده و اگر برابر صفر شد، آنگاه دستور بعدی اجرا نمی‌شود و به PC یک واحد اضافه می‌شود. (می‌توان برای شمارنده حلقه استفاده شود). نمی‌توان به طور مستقیم مقدار یک خانه از حافظه را اضافه کرد، ابتدا باید در رجیستر DR ریخته شود، مقدار آن را یک واحد زیاد کرد و سپس مقدار اضافه شده را در خانه حافظه مورد نظر نوشت.

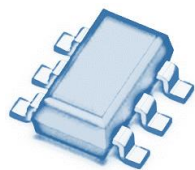
$$D_6T_4: DR \leftarrow M[AR]$$

$$D_6T_5: DR \leftarrow DR + 1$$

$$D_6T_6: M[AR] \leftarrow DR, \text{if } DR = 0 \text{ then } PC \leftarrow PC + 1, SC \leftarrow 0$$

SNA (Skip if AC is Negative): اگر مقدار ذخیره شده در AC یک مقدار منفی باشد، دستور بعد از آن اجرا نمی‌شود (به دو دستور بعد پرش می‌کند) و مقدار PC یک واحد زیاد می‌شود. اما اگر مقدار $AC > 0$ مثبت باشد همان دستور بعد اجرا می‌شود. برای دستورات شرطی کاربرد دارد.

$$D_7IT_3IR(3): \text{if } AC(15) = 1 \text{ then } PC \leftarrow PC + 1, SC \leftarrow 0$$



INP (Input Character): مقدار یک کاراکتر 8 بیتی را که از ورودی خوانده و در رجیستر INPR قرار دارد را در ۸ بیت کم ارزش AC ذخیره می کند و مقدار flag ورودی را صفر می کند (به معنای دریافت اطلاعات و آماده برای دریافت داده جدید).

$$D_7IT_3IR(11): AC(0 - 7) \leftarrow INPR, FGI \leftarrow 0, SC \leftarrow 0$$

۶) محتوای تعدادی از خانه های حافظه کامپیوتر مبنا مطابق زیر می باشد. تعیین نمایید که پس از توقف:

آدرس	حافظه
0	4004F
1	F881H
2	777FH
3	FFFEH
4	2001H
5	0002H
6	3003H
7	3008H
8	6003H
9	4004H
10	7001H

الف) چند دستورالعمل اجرا می شود؟

۶ دستور

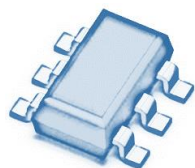
ب) چند مراجعه به حافظه انجام می شود (مجموع خواندن و نوشتن)

در کل ۱۰ دستور: ۶ دستور برای خواندن دستورات و ۲ دستور برای خواندن از حافظه و دو دستور برای نوشتن در حافظه

ج) محتوای AC, DR, PC, AR, IR و E را پس از توقف تعیین نمایید.

$$AC=7001, \quad DR=777F, \quad IR=7001, \quad AR=001, \quad PC=009, \quad E=0$$

آدرس	حافظه	دستور	
0	4004F	BUN 4	$PC \leftarrow 4$
1	F881H		
2	777FH		
3	FFFEH 7001H		
4	2001H	LDA 1	$PC \leftarrow 5, AR \leftarrow 1$ $AC \leftarrow M[AR] \% AC \leftarrow F881H$
5	0002H	AND 2	$PC \leftarrow 6, AR \leftarrow 2$ $DR \leftarrow M[AR] \% DR : 777FH$ $AC \leftarrow AC \text{ and } DR \% AC \leftarrow (F881H \text{ and } 777FH) = 7001H$
6	3003H	STA 3	$PC \leftarrow 7, AR \leftarrow 3$ $M[3] \leftarrow AC \% M[AR] = 7001H$
7	3008H	STA 8	$PC \leftarrow 8, AR \leftarrow 8$ $M[8] \leftarrow AC \% M[8] \leftarrow 7001H$
8	6003H 7001H	HLT	$PC \leftarrow 9$ Halt computer
9	4004H		
10	7001H		



(۷) مجموعه دستورات روبرو را که در خانه‌های $M[0]$ تا $M[16]$ قرار دارد در نظر بگیرید.

0	BUN 6	6	LDA 1	12	LDA 2
1		7	CMA	13	ADD 4
2		8	STA 3	14	BUN 9
3		9	ISZ 3	15	STA 6
4		10	BUN 12	16	HLT
5		11	BUN 15		

الف) این دستورات چه عملی را انجام می‌دهند؟

مقدار خانه آدر 1 حافظه را مکمل می‌کند و در خانه 3 حافظه ذخیره می‌کند. اگر این مقدار 1- بود، \overline{AC} را در خانه ششم حافظه ذخیره می‌کند و در غیر این صورت در یک حلقه تا زمانی که مقدار محتوای خانه ۳ برابر صفر شود، مجموع خانه دوم و چهارم را در

AC ذخیره می‌کند. و زمانی که مقدار $M[3]$ برای صفر شد این مقدار را در خانه ۶ ذخیره می‌کند. در هر دو صورت مقدار خانه سوم در پایان برابر صفر می‌شود. حلقه جمع $(M[2]+M[4])$ به تعداد $M[1]$ بار اجرا می‌شود یعنی $M[1]$ شمارنده حلقه است.

آدرس	دستور	توضیحات
0	BUN 6	$PC \leftarrow 6, goto 6$
6	LDA 1	$PC \leftarrow 7, AC \leftarrow M[1]$
7	CMA	$PC \leftarrow 8, AC \leftarrow \overline{AC}$
8	STA 3	$PC \leftarrow 9, M[3] \leftarrow AC$
9	ISZ 3	$PC \leftarrow 10, DR \leftarrow M[3]$ $DR \leftarrow DR + 1, M[3] \leftarrow DR$ $if (DR == 0) then PC \leftarrow 11$
10	BUN 12	$PC \leftarrow 12$
11	BUN 15	$PC \leftarrow 15$
12	LDA 2	$PC \leftarrow 13, AC \leftarrow M[1]$
13	ADD 4	$PC \leftarrow 13, DR \leftarrow M[4], AC \leftarrow AC + DR$
14	BUN 9	$PC \leftarrow 9$
15	STA 6	$PC \leftarrow 16, M[6] \leftarrow AC$
16	HLT	$PC \leftarrow 17, Halt computer$

$AC \leftarrow M[1]$

$AC \leftarrow \overline{AC}$

$M[3] \leftarrow AC$

repeat:

 If $((M[3] + 1) == 0)$

$M[6] \leftarrow AC$

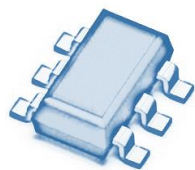
 HLT

 Else:

$AC \leftarrow M[2]$

$AC \leftarrow AC + M[4]$

 goto repeat



ب) برای خانه های حافظه $M[1]$ تا $M[5]$ به دلخواه خود مقادیر اولیه ای انتخاب نمایید. در پایان الگوریتم، مقدار این خانه ها را تعیین کنید.

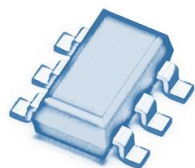
مقدار		آدرس
0001	0001	1
0011	0011	2
0000	0100	3
0101	0101	4
0110	0110	5
1000		6

ج) چند بار به حافظه رجوع کرده ایم؟

دستورات LDA و STA و ADD هر کدام یک مراجعه به حافظه دارند و دستور ISZ دو دسترسی به حافظه دارد. برای مقادیر دلخواه شرط if دو بار چک شده است و یکبار بدنه if و یکبار بدنه else اجرا شده است پس در کل ۱۳ دستور اجرا می شود. ۲ دستور load، دو دستور STA، یک دستور ADD و دو دستور ISZ اجرا شده است. پس در کل $2 \times 2 + 1 + 2 + 2 + 13 = 22$ دستور حافظه اجرا شده است. به ازای هر بار تکرار حلقه یک دستور ISZ، یک دستور ADD و یک دستور LDA به دستورات فوق اضافه می شود.

د) مقدار AC در پایان الگوریتم چیست؟

مقدار AC برابر با مقدار حافظه $M[6]$ خواهد بود. اگر $M[1] = 0$ آنگاه مقدار $AC = \overline{M[1]}$ در مرحله اول شرط if برقرار می شود (در غیر این صورت مقدار AC برابر با $M[2] + M[4]$ یا مقدار حافظه $M[6]$ می شود. برای مفروضات بخش ب مقدار AC برابر با ۱۰۰۰ می شود

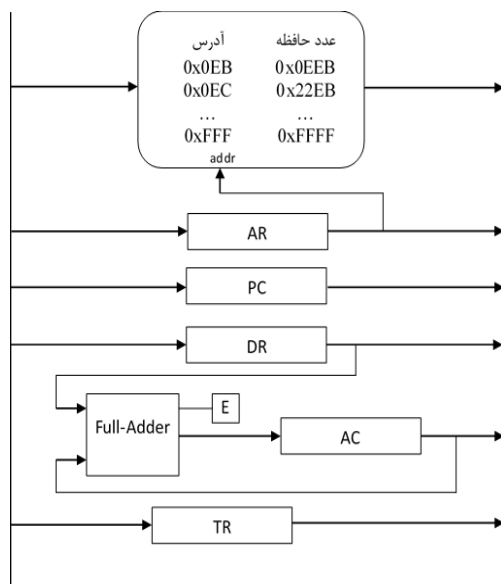


(۸) دیاگرام باس زیر را در نظر بگیرید.

الگوریتمی بر حسب RTL بنویسید که در پایان، دو عدد 0xBEEB و 0x22EB را با هم جمع کرده در رجیستر TR ذخیره کند. سپس در صورتی که حاصل جمع دارای رقم نقلی carry بود، رجیستر PC مقدار 0xFFFF و در غیر این صورت، مقدار 0x000 را بگیرد. تعداد تایمینگ‌ها کمینه باشد.

○ مقدار اولیه رجیسترها به صورت زیر می‌باشد:

PC	AR	AC	DR	TR
0x0EB	0xFFFF	0xFFFF	0xFFFF	0xFFFF



○ در پاسخ خود به محدودیت‌های باس داده شده و نیز مسیرهای قابل استفاده توجه نمایید.

○ همه رجیسترها توانایی inc, clr, Load و dec را دارند و تأخیر full-adder قابل صرف نظر کردن است.

○ حافظه قابلیت read و write آسنکرون را دارا می‌باشد. مقدار نهایی AC، DR و AR هر چیزی می‌تواند باشد.

$$T_0: AR \leftarrow PC$$

$$T_1: DR \leftarrow M[AR]$$

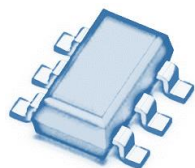
$$T_2: AC \leftarrow DR, AR \leftarrow AR + 1$$

$$T_3: DR \leftarrow M[AR]$$

$$T_4: AC \leftarrow AC + DR$$

$$T_5: TR \leftarrow AC, PC \leftarrow 0$$

$$T_6: \text{if } (E = 1) PC \leftarrow PC - 1$$



۹) در کامپیوتر پایه می‌خواهیم دستورات ارجاع به حافظه را با مجموعه دستورات جدول روبرو جایگزین نماییم. RTL اجرای هر یک از این دستورات را از T4 به بعد مشخص نمایید. توجه کنید که قابلیت XOR به ALU کامپیوتر پایه اضافه شده است، ولی عملیات تفریق ندارد. برای انجام عملیات تفریق از مکمل ۲ استفاده کنید.

دستور	نشان سمبلیک	کد عمل	سمبل
$D_0T_4: DR \leftarrow M[EA]$ $D_0T_5: AC \leftarrow AC \oplus DR, SC \leftarrow 0$	$AC \leftarrow AC \oplus M[EA]$	000	XOR
$D_1T_4: DR \leftarrow M[EA]$ $D_1T_5: AC \leftarrow AC + DR$ $D_1T_6: M[EA] \leftarrow AC, SC \leftarrow 0$	$M[EA] \leftarrow AC + M[EA]$	001	ADM
$D_2T_4: DR \leftarrow M[EA]$ $D_2T_5: AC \leftarrow DR, DR \leftarrow AC$ $D_2T_6: AC \leftarrow \overline{AC}$ $D_2T_7: AC \leftarrow AC + 1$ $D_2T_8: AC \leftarrow AC + DR, SC \leftarrow 0$	$AC \leftarrow AC - M[EA]$	010	SUB
$D_3T_4: DR \leftarrow M[EA]$ $D_3T_5: M[EA] \leftarrow AC, AC \leftarrow DR, SC \leftarrow 0$	$AC \leftarrow M[EA], M[EA] \leftarrow AC$	011	XCH
$D_4T_4: DR \leftarrow M[EA]$ $D_4T_5: TR \leftarrow AC, AC \leftarrow DR \oplus AC$ $D_4T_6: if (AC = 0) then (PC \leftarrow PC + 1), AC \leftarrow TR, SC \leftarrow 0$	$if (M[EA] == AC) then$ $PC \leftarrow PC + 1$	100	SEQ
$D_5T_4: if (AC(15) = 0 and AC \neq 0)$ $then (PC \leftarrow AR), SC \leftarrow 0$	$if (AC > 0) then (PC \leftarrow M[EA])$	101	BPA

جدول دستورات کامپیوتر پایه

Symbol	Hexadecimal code		Description
	<i>I</i> = 0	<i>I</i> = 1	
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load memory word to AC
STA	3xxx	Bxxx	Store content of AC in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear AC
CLE	7400		Clear E
CMA	7200		Complement AC
CME	7100		Complement E
CIR	7080		Circulate right AC and E
CIL	7040		Circulate left AC and E
INC	7020		Increment AC
SPA	7010		Skip next instruction if AC positive
SNA	7008		Skip next instruction if AC negative
SZA	7004		Skip next instruction if AC zero
SZE	7002		Skip next instruction if E is 0
HLT	7001		Halt computer
INP	F800		Input character to AC
OUT	F400		Output character from AC
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off