



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده مهندسی برق

مقطع  
کارشناسی

عنوان  
مسیریابی

درس  
معماری کامپیوتر

نگارش  
کسری خلفی  
9523038

استاد  
دکتر راعی

دی ماه 1397

## 1- الگوریتم :

الگوریتم کار بدین صورت است که چونکه تنها یک مسیر برای حرکت داریم در نتیجه لازم نیست که مسیر را ابتدا در جایی سیو کرده تا در صورت نرسیدن به انتها به نقطه ای بازگردد

الگوریتم کار بدین صورت است که ابتدا برای شروع حرکت برای شروع ادرس را به خانه ی دوم row2 منتقل میکنیم بدین صورت که ابتدا ادرس ای که به ادرس خانه های سطر ها اشاره میکند را اضافه میکنیم 4 تا که بریم به ادرس خانه ی row2 سپس به ادرس row2 یکی اضافه میکنیم تا به خانه ی rownum+2 برویم که همیشه یک است و ادرس شروع است سپس مسیر ها رو شروع میکنیم به چک کردن بدین صورت که اول سمت راست را چک میکنیم بدین صورت که به ادرسی که داریم بدونیم اضافه میکنیم سپس cmp میکنیم که ببینیم دو عدد برابرند یا نه در صورت برابر بودن ابتدا رجیستر دوباره branch میکنیم به rightcheeking که باز ببینیم به راست باید حرکت میکنیم یا نه و اگر برابر نبود مقدار اضافه کرده به ادرس را کم کرده و میرویم و پایین را چک میکنیم که یک یا نه بدین صورت که به ادرس به اندازه ی rownum اضافه میکنیم و با cmp چک میکنیم که پایین یک است یا خیر در صورت بودن branch میکنیم به rightcheeking که از اول چک کنیم مسیر را. اگر درست نبود مقدار rownum ای را که اضافه کرده بودیم را کم میکنیم و چپ را با leftcheeking چک میکنیم بدین ترتیب که از ادرس بدونیم کم میکنیم و با cmp یک بودن را چک میکنیم و در صورت بودن branch میکنیم دوباره به rightcheeking و در صورت نبودن ادرس کم کرده را اضافه کرده و بالا را با upcheeking میبینیم که یک یا نه بدین صورت که از ادرس به مقدار rownum کم میکنیم در صورت بودن میریم به right cheeking در صورت نبودن ادرس کم کرده را اضافه میکنیم و وارد لوپ بینهایت میکنیم که را

برای اینکه مسیر آمده را چک نکند پس از cmp به رجیستر r8 مقادیر 1 و 2 و 3 و 4 را به ترتیب برای rightcheeking و downcheeking و leftcheeking و upcheeking میدهیم و در اول هر کدام چک میکنیم که برای راست رونده 3 نباشد یعنی چون از چپ آمدیم چپ را چک نکن و اگر r8 برابر 3 شد branch کن به بعدی یعنی downcheeking و برای چپ مقدار r8 را با 1 چک کن و برای بالا با 2 و برای پایین با 4 چک کن r8 را و در صورت برابر بودن آن دستور را انجام نده

## 2- توضیح کد:

ابتدا آدرس rows را در رجیستر r0 میریزیم و خود rows را در رجیستر r6 میریزیم که برابر است با آدرس شروع اولین عدد و از r6 بدون کم میکنیم تا عدد اول را یک نام گذاری کنیم نه صفر سپس به r0 که آدرس rows بود 4 تا اضافه میکنیم که به آدرس بعدی بره و بشه آدرس rows2 را در مقدارش نشان بدهد و مقدارش را میریزیم توی r1 که بشه آدرس اولین آرایه ی row2 و به این آدرس بدون اضافه میکنیم تا بشه آدرس شروع سپس این آدرس را از آدرس اولین عدد که ریختیمش توی r6 کم میکنیم و این میشه عدد rownum+2 ام آرایه و میریزیمش توی r7 این مقدار رو میریزیم توی r13 که استک پوینترمونه سپس بدون کم میکنیم که بره جای خالی که بعدی ها را بتوانیم سیو کنیم حالا میریم ببینیم که به کدام طرف میره

### Rightchecking:

همانطور که در بالا توضیح داده شد اول چک میکنیم که r8 برابر 3 است یا نه یعنی داره از چپ میاید یا نه ک اگر امد دیگه چپ را چک نکنیم و میریم downchecking و اگر نبود ابتدا مقدار را در r5 ذخیره میکنیم و به آدرس بدون اضافه میکنیم و مقدارش را میریزیم توی r4 و یک cmp میزنیم در صورت eq بودن r8 را یک میکنیم و اینکه آرایه ی چندمه رو با تفاضل از r6 میریزیم تو r7 سپس r7 را میریزیم توی استک پوینتر r13 و یکی از ان کم میکنیم که خانه ی خالی شود و دوباره میریم به ابتدای rightchecking در صورت eq نبودن از r1 مقدار اضافه شده را کم میکنیم سپس میریم پایین را چک کنیم با

### downchecking

### Downchecking:

R8 را چک میکنیم که ببینیم 4 است یا نه در صورت 4 بودن پایین را چک نمیکنیم و میریم بعدی را چک میکنیم اگر نبود مقدار داخل آدرس را میریزیم در r5 و به آدرس به ازای rownum اضافه میکنیم که بره آدرس سطر پایین و مقدار داخل آدرس را میریزیم در r4 و cmp میزنیم بین دو تا رجیستر 4 و 5 در صورت برابر بود r8 را 2 میکنیم و اینکه عدد چندم است را با تفریق از آدرس خانه ی اول به اضافه ی یک به دست میاوریم و در r7 ریخته و r7 را در داخل رجیستر 13 میریزیم و آدرسش رو یکی کم میکنیم که به خانه ی خالی اشاره کنه و

میریم به **rightcheking** و در صورت برابر نبودن مقدار اضافه شده به ادرس را کم میکنیم تا به ادرس قبلی برگردیم و برنچ میکنیم به **leftcheking**

**Leftcheking:**

ابتدا **r8** را با **1** مقایسه میکنیم اگر برابر بودند دیگر چپ را چک نمیکند و میره برای چک کردن بالا ولی اگر برابر نبود مقدار داخل ادرس را در **r5** میریزیم و از ادرس ای که توی **r1** است یدونه کم میکنیم که بشه ادرس قبلی یا همان چپی و مقدار داخلش را میریزیم داخل **r4** و رجیستر های **4** و **5** را باهم مقایسه میکنیم در صورت برابری **r8** را برابر **3** قرار میدهیم و ادرسی که در **r1** است را از ادرس خانه ی اول کم کرده و به اضافه ی یک میکنیم که بشود خانه ی چندم آرایه و در رجیستر **13** میریزیمش و از **r13** یدونه کم میکنیم که بره به خانه ی خالی و میریم به **rightcheking** اما اگر برابر نبودند مقدار کم شده را اضافه میکنیم و تا بشه ادرس قبلی عه و میریم به **upcheking**

**Upcheking:**

اول میبینیم که **r13** برابر **2** است یا نه در صورت برابری دیگر **upcheking** اجرا نمیشود و برمیگردیم به چک کردن راست در صورت برابر نبودن مقدار داخل **r1** را در **r5** میریزیم و از ادرسی که داریم به مقدار **rownum** کم میکنیم تا بره به ادرس بالایی و در **r1** میریزیم ادرس رو دوباره سپس مقدار این ادرس را در **r4** میریزیم و **cmp** میزنیم بین دو تا رجیستر **4** و **5** که در صورت برابری مقدار ادرس را از **r6** کم میکنه که میشه ادرس را از خانه ی اول کم میکنه و به علاوه ی یک میکنه تا بشه ادرس آرایه ای که هست بر حسب **rownum** و این مقدار رو میریزیم داخل **r13** که استک پوینتره و از **r13** یدونه کم میکنیم که بره به خانه ی خالی و در صورتی که **r4** و **r5** برابر نبودند مقدار کم شده از رجیستر **1** رو که برابر **rownum** بود را اضافه میکنیم به ادرس امان که همان رجیستر **1** عه و چون همه ی مسیر ها را چک کردیم در آخر **upcheking** و به مسیر دیگه ای نمیتوانیم برویم در نتیجه به آخر مسیر رسیده ایم و وارد لوپ بینهایت میشویم و یعنی به خانه ی آخر رسیده ایم

### 3-تصاویری از نتایج:

مسیر دلخواهی که من زدم اینه:

```
rownum      DCD      12
rows         DCD      row1,row2,row3,row4,row5,row6,row7,row8,row9,row10,row11,row12
row1         DCB      0,0,0,0,0,0,0,0,0,0,0,0,0
row2         DCB      0,1,1,1,1,1,0,0,0,0,0,0,0
row3         DCB      0,0,0,0,0,1,0,0,0,0,0,0,0
row4         DCB      0,1,1,1,0,1,0,0,0,0,0,0,0
row5         DCB      0,1,0,1,1,1,0,0,0,0,0,0,0
row6         DCB      0,1,0,0,0,0,0,0,0,0,0,0,0
row7         DCB      0,1,1,1,1,1,1,1,1,1,0,0,0
row8         DCB      0,0,0,0,0,0,0,0,0,0,1,0,0
row9         DCB      0,0,0,0,0,0,0,0,0,0,1,0,0
row10        DCB      0,0,0,0,0,0,0,0,0,0,1,0,0
row11        DCB      0,0,0,0,0,0,0,0,0,0,1,1,0
row12        DCB      0,0,0,0,0,0,0,0,0,0,0,0,0
```

و اما مقداری که در r13 ذخیره کردم بر حسب ارایه بدین صورت هستند:

Address:	0x20000f00
0x20000f88:	000 000 000 000
0x20000f8c:	000 000 000 000
0x20000f90:	131 000 000 000
0x20000f94:	130 000 000 000
0x20000f98:	118 000 000 000
0x20000f9c:	106 000 000 000
0x20000fa0:	094 000 000 000
0x20000fa4:	082 000 000 000
0x20000fa8:	081 000 000 000
0x20000fac:	080 000 000 000
0x20000fb0:	079 000 000 000
0x20000fb4:	078 000 000 000
0x20000fb8:	077 000 000 000
0x20000fbc:	076 000 000 000
0x20000fc0:	075 000 000 000
0x20000fc4:	074 000 000 000
0x20000fc8:	062 000 000 000
0x20000fcc:	050 000 000 000
0x20000fd0:	038 000 000 000
0x20000fd4:	039 000 000 000
0x20000fd8:	040 000 000 000
0x20000fdc:	052 000 000 000
0x20000fe0:	053 000 000 000
0x20000fe4:	054 000 000 000
0x20000fe8:	042 000 000 000
0x20000fec:	030 000 000 000
0x20000ff0:	018 000 000 000
0x20000ff4:	017 000 000 000
0x20000ff8:	016 000 000 000
0x20000ffc:	015 000 000 000
0x20001000:	014 000 000 000
0x20001004:	000 000 000 000
0x20001008:	000 000 000 000
0x2000100c:	000 000 000 000

مقدار نهایی رجیستر ها هم در اخر بدین صورت هستند:

Register	Value
<b>Core</b>	
R0	0x080000CC
R1	0x0800017A
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000001
R6	0x080000F7
R7	0x00000083
R8	0x00000001
R9	0x0000000C
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000F8C
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x080000BE
xPSR	0x81000000
+ Banked	
+ System	
- Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	952
Sec	0.00003808

Project Registers

