1. 实验内容

用递归法走迷宫

1. 算法分析

输入初始地址后，依次检测四周是否有通路。若某方向有通路，递归调用maze处理新地址；若此方向没有通路，检测下一方向。所有方向检测完成后，判断地址栈是否为空，栈空，输出No path；栈不空，返回。

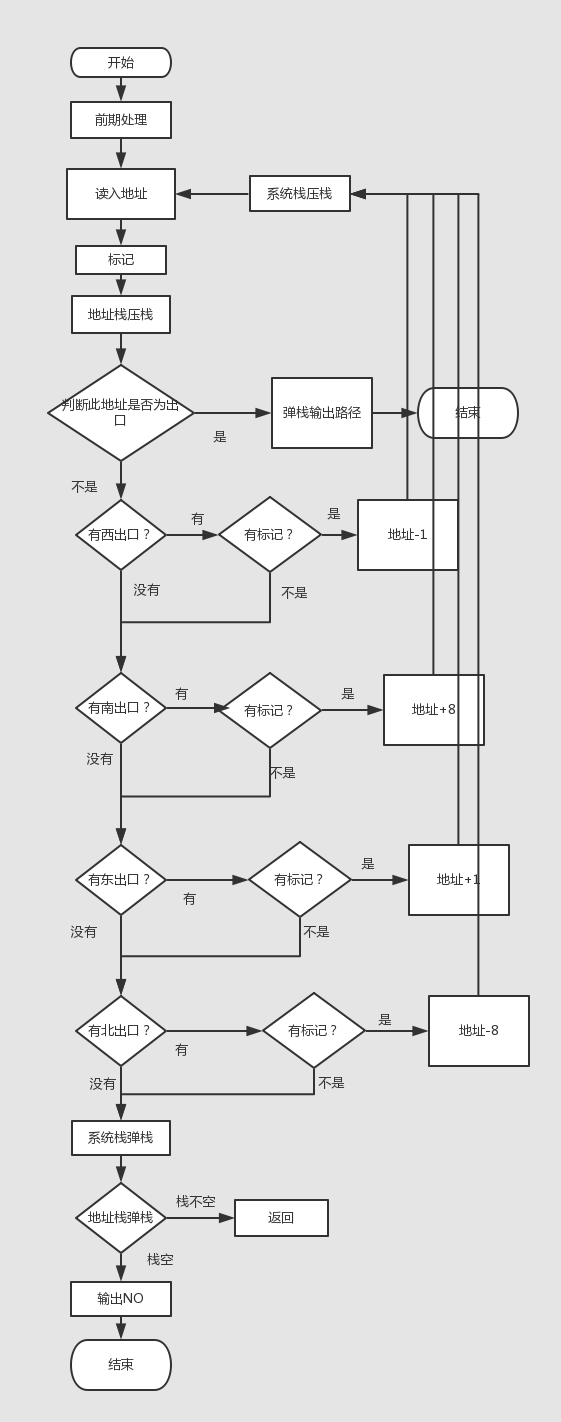
已走过的位置将地址的bit[14]标为1

建立地址栈，走到未标记地址，地址入栈；若此地址周围没有通路，弹栈。检测到出口，弹栈处理地址信息，直到栈空，输出通路。

建立系统栈，存储进入递归前系统的信息。进入递归前，系统信息压栈；跳出一层递归前，弹栈恢复系统信息

得出的通路数据，加ASCII码x0030后存储输出

1. 流程图



1. 代码和注释

;R1地址栈指针;指向上一个空位

;R2当前地址指针

;R6系统栈指针

.ORIG x3000

LD R6,SYSSTACK

LD R1,STACK

JSR GET\_ADDRESS

JSR MAZE

MAZE JSR MARK

JSR PUSH

JSR EXIT

JSR WESTDOOR

JSR SOUTHDOOR

JSR EASTDOOR

JSR NORTHDOOR

JSR POP

;RECOVER

ADD R6,R6,-1

LD R3,SYSBOTTOM

ADD R3,R3,R6

BRz L10

LDR R7,R6,0

ADD R6,R6,-1

LDR R2,R6,0

ADD R6,R6,-1

LDR R1,R6,0

;EMPTY

L10 LD R3,BOTTOM

ADD R3,R3,R1

BRp L5

LEA R0,NO

PUTS

HALT

L5 RET

GET\_ADDRESS LDI R3,ADDRESSI ;i;turn i,j into address

LDI R4,ADDRESSJ ;j

LD R2,ADDRESSI ;R2=x4000

ADD R2,R2,2

ADD R2,R3,R2 ;

ADD R2,R3,R2

ADD R2,R3,R2

ADD R2,R3,R2

ADD R2,R3,R2

ADD R2,R3,R2

ADD R2,R3,R2

ADD R2,R3,R2

ADD R2,R2,R4

RET

PUSH STR R2,R1,0

ADD R1,R1,1

RET

POP ADD R1,R1,-1

RET

EXIT LD R3,MASK\_EXIT ;is this address the exit

LDR R4,R2,0

AND R4,R3,R4

BRnz L6

JSR TRANSRESULT

L6 RET

STORE STR R1,R6,0

ADD R6,R6,1

STR R2,R6,0

ADD R6,R6,1

STR R7,R6,0

ADD R6,R6,1

RET

MARK LD R3,MASK\_MARK

NOT R3,R3

LDR R4,R2,0

NOT R4,R4

AND R3,R4,R3

NOT R3,R3

STR R3,R2,0

RET

WESTDOOR LD R3,MASK\_WEST

LDR R4,R2,0

AND R3,R4,R3

BRz L11

LD R3,MASK\_MARK

ADD R2,R2,-1

LDR R4,R2,0

AND R3,R3,R4

BRp L1

JSR STORE

JSR MAZE

L1 ADD R2,R2,1

L11 RET

SOUTHDOOR LD R3,MASK\_SOUTH

LDR R4,R2,0

AND R3,R4,R3

BRz L12

LD R3,MASK\_MARK

ADD R2,R2,+8

LDR R4,R2,0

AND R3,R3,R4

BRp L2

JSR STORE

JSR MAZE

L2 ADD R2,R2,-8

L12 RET

EASTDOOR LD R3,MASK\_EAST

LDR R4,R2,0

AND R3,R4,R3

BRz L13

LD R3,MASK\_MARK

ADD R2,R2,1

LDR R4,R2,0

AND R3,R3,R4

BRp L3

JSR STORE

JSR MAZE

L3 ADD R2,R2,-1

L13 RET

NORTHDOOR LD R3,MASK\_NORTH

LDR R4,R2,0

AND R3,R4,R3

BRz L14

LD R3,MASK\_MARK

ADD R2,R2,-8

LDR R4,R2,0

AND R3,R3,R4

BRp L4

JSR STORE

JSR MAZE

L4 ADD R2,R2,8

L14 RET

TRANSRESULT LD R4,ADDRESSI ;将结果转换，并存到x7000，并输出

ADD R4,R4,2

NOT R4,R4

ADD R4,R4,1 ;减去起始地址

LD R5,STACK ;栈指针

LD R6,RESULT

L8 LDR R3,R5,0 ;栈指针内容

BRz L9

AND R2,R2,0

AND R1,R1,0

ADD R3,R4,R3

L7 ADD R3,R3,-8

BRzp L20

ADD R2,R3,8

BRnzp L21

L20 ADD R1,R1,1

BRnzp L7

L21 LD R0,ASCII

ADD R1,R1,R0

STR R1,R6,0

ADD R6,R6,1

ADD R2,R2,R0

STR R2,R6,0

ADD R6,R6,1

LD R1,SPACE

STR R1,R6,0

ADD R6,R6,1

ADD R5,R5,1

BRnzp L8

L9 LD R0,RESULT

PUTS

HALT

ASCII .FILL x0030

SPACE .FILL x0020

BOTTOM .FILL x9FFF

SYSBOTTOM .FILL xB001

ADDRESSI .FILL x4000

ADDRESSJ .FILL x4001

STACK .FILL x6000

SYSSTACK .FILL x5000

RESULT .FILL x7000

MASK\_MARK .FILL x4000

MASK\_WEST .FILL x0001

MASK\_SOUTH .FILL x0002

MASK\_EAST .FILL x0004

MASK\_NORTH .FILL x0008

MASK\_EXIT .FILL x0010

NO .STRINGZ "No path"

.END

1. 运行结果

