实验目的：实现中断控制逻辑——按一次切换一种状态。

实验设计：在复位中初始化相关寄存器，并利用循环等待中断到来。在中断向量表之后编写中断服务程序，在中断服务程序中（只用一个中断不需要判断中断来源，两个及以上则需要）先释放中断标志位，再执行用户想要的功能即反转LD4。

注意：释放中断标志位时，写一清零。

代码及其注释：

①找到向main主函数跳转的命令并删除。

②初始化相关寄存器

;RCC.IOPENR initialize

LDR R1, =0x40021000 ;Base address #4002 1000

MOVS R2, #0x34 ;Address offset #0x34

MOVS R0, #0x05 ;Set value #0x0000 0005

STR R0, [R1,R2]

;GOIOC.MODER initialize

LDR R1, =0x50000800 ;Base address #5000 0800

MOVS R2, #0x00 ;Address offset #0x00

LDR R0, =0xF3FFFFFF ;Set value #0xF3FF FFFF

STR R0, [R1,R2]

;EXTI.EXTICR4 initialize

LDR R1, =0x40021800 ;Base address #4002 1800

MOVS R2, #0x6C ;Address offset #0x6C

LDR R0, =0x00000200 ;Set value #0x0000 0200

STR R0, [R1,R2]

;EXTI.IMR1 initialize

;Base address #4002 1800,don't need to modify it

MOVS R2, #0x80 ;Address offset #0x80

LDR R0, =0xFFF82000 ;Set value #0xFFF8 2000

STR R0, [R1,R2]

;EXTI.EMR1 initialize

;Base address #4002 1800,don't need to modify it

MOVS R2, #0x84 ;Address offset #0x84

;Set value #0xFFF8 2000,don't need to modify it

STR R0, [R1,R2]

;EXTI.FTSR1 initialize

;Base address #4002 1800,don't need to modify it

MOVS R2, #0x04 ;Address offset #0x04

LDR R0, =0x00002000 ;Set value #0x0000 2000

STR R0, [R1,R2]

;GPIOA.MODER initialize

LDR R1, =0x50000000 ;Base address #50000000

MOVS R2, #0x00 ;Address offset #0x00

LDR R0, =0xEBFFF7FF ;Set value #0xEBFFF7FF

STR R0, [R1,R2]

;NVIC.ISER initialize

LDR R1, =0xE000E100

MOVS R2, #0x00

MOVS R0, #0x80

STR R0, [R1,R2]

③设置LD4灯的初始状态

LDR R1, =0x50000000 ;light

MOVS R2, #0x14

MOVS R0, #0x20

STR R0,[R1,R2]

④循环等待中断到来

LEBEL5

B LEBEL5

⑤添加中断服务程序

EXTI4\_15\_IRQHandler PROC

EXPORT EXTI4\_15\_IRQHandler [WEAK]

PUSH {LR}

;清除中断标志位————写一清零

LDR R1, =0x40021800;Base address #4002 1800

MOVS R2, #0x010 ;Address offset #0x6C

LDR R0, =0x00002000;Set value #0x0000 2000

STR R0, [R1,R2]

LDR R1, =0x50000000

MOVS R2, #0x14

LDR R0, [R1,R2] ;GPIOA.ODR

CMP R0, #0x20

BEQ LEBEL1 ;如果亮跳转到LEBEL1

MOVS R0, #0x20 ;不亮则变亮

STR R0, [R1,R2]

B LEBEL ;跳出中断

LEBEL1

MOVS R0,#0x00 ;变灭

STR R0, [R1,R2]

LEBEL

POP {PC}

ENDP