```c
 * switches.c

#include "supportFiles/leds.h" //needed to access the LED functions
#include "switches.h"
#include "xparameters.h"       //needed to access base address of GPIOs
#include "xil_io.h"            //includes the low-level Xilinx functions needed for reading and writing to GPIOs

#define SWITCHES_GPIO_BASE_ADDRESS XPAR_SLIDE_SWITCHES_BASEADDR //base address from xparameters.h
#define SWITCHES_DATA_OFFSET 0                                  //value based on register documentation provided by Xilinx for GPIO_DATA
#define SWITCHES_TRI_OFFSET 4                                   //value based on register documentation provided by Xilinx for GPIO_TRI
#define SWITCHES_VALUE 0xF                                      //value to be written to GPIO_TRI to make sure it behaves correctly
#define SWITCHES_ALL_ON 0xF                                     //value to verify that all switches are on
#define SUCCESSFUL_SWITCHES_INIT 0                              //value passed in to leds_init function
#define LEDS_ALL_OFF 0x0                                        //value used to turn off all LEDs after all 4 switches are turned on

//helper function to read from GPIOs
int32_t switches_readGpioRegister(int32_t offset){
    return Xil_In32(SWITCHES_GPIO_BASE_ADDRESS + offset); //using low-level Xilinx call
}

//helper function to write to GPIOs
void switches_writeGpioRegister(int32_t offset, int32_t value){
    Xil_Out32(SWITCHES_GPIO_BASE_ADDRESS + offset, value); //low-level Xilinx call
}

//Initializes the SWITCHES driver software and hardware. Returns one of the STATUS values defined above.
int32_t switches_init(){
    switches_writeGpioRegister(SWITCHES_TRI_OFFSET, SWITCHES_VALUE);          //writing only to GPIO_TRI
        if(switches_readGpioRegister(SWITCHES_TRI_OFFSET) == SWITCHES_VALUE){   //reads from GPIO_TRI to make sure the data was correctly written to it
            return SWITCHES_INIT_STATUS_OK;
        }
//GPIO_DATA doesn't need to be written to in order to behave correctly

        return SWITCHES_INIT_STATUS_FAIL;
}

//Returns the current value of all 4 switches as the lower 4 bits of the returned value.
//bit3 = SW3, bit2 = SW2, bit1 = SW1, bit0 = SW0.
int32_t switches_read(){
    return switches_readGpioRegister(SWITCHES_DATA_OFFSET) & SWITCHES_VALUE; //need to bit-mask in order to get the last 4 bits to work with
}

void switches_runTest(){
    leds_init(SUCCESSFUL_SWITCHES_INIT);

```

```
50      int32_t readInVal = 0;      //variable that will contain the values read from
  GPIO_DATA
51      int32_t oldVal = 0;         //variable to store the former value read from
  GPIO_DATA
52
53      //runs until all 4 switches are slid upward
54      while(readInVal != SWITCHES_ALL_ON){
55
56          readInVal = switches_read(); //get values of switches
57
58          if(oldVal != readInVal){        //check to make sure new value was read in
  before doing anything
59              leds_write(readInVal);      //writes current values of the LEDs
60          }
61          oldVal = readInVal;          //store the value just used to check it against
  the one about to read in
62      }
63      leds_write(LEDS_ALL_OFF);    //clear LEDs when all the switches are turned on
64  }
65
66
67
```