**Leakage数据提取与处理源代码**

1. **main.c**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<stdbool.h>

#include "fram.h"

#include "excel.h"

#include "leakage.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Copyright (c) 2020, longsys \*/

/\* All rights reserved. \*/

/\* Project Name: Leakage\_Test\*/

/\* Author: gaozhe\*/

/\* Date: 2020.03.26\*/

/\* Version:v1.0 \*/

/\* Abstract：Leakage的提取和处理主函数，实现各模块接口调用 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int mode\_num=0; //芯片模式标识符 LPDDR4 1CS(1) LPDDR4 2CS(2) LPDDR3 1CS(3) LPDDR3 2CS(4)

int pin\_num=66; //初始化默认 66 pins

int main()

{

int i;

int j;

int k;

char \*filename\_buf;

char dirname\_buf[50]={0};

struct Fileinfo dirinfo;

FILE \*fp\_log=NULL;

dirinfo=scan\_file(); //扫描文件夹目录，记录文件夹目录信息

for(i=0;i<dirinfo.dir\_sum;i++)

{

printf("\n 已识别文件夹 %s ！ \n",dirinfo.dir\_name[i]);

}

for(i=0;i<dirinfo.dir\_sum;i++)

{

for(j=1;j<=8;j++) //采集8个测试log文件 ,生成leakage二维结构体数组

{

strcpy(dirname\_buf,dirinfo.dir\_name[i]);

filename\_buf=select\_file(j); //选择log文本1~8

strcat(dirname\_buf,"/");

strcat(dirname\_buf,filename\_buf);

fp\_log=fopen(dirname\_buf,"r"); //打开log文本

if(fp\_log==NULL)

{

continue;

}

get\_leakage\_struct(fp\_log,leakage[j-1]); //获取leakage数据信息

fclose(fp\_log);

memset(dirname\_buf,0,50);

}

for(j=0;j<8;j++) //获取排序结构体二维数组

{

for(k=0;k<=leakage[0][0].sample\_num;k++)

{

leakage\_sort[j][k]=leakage[j][k];

}

}

get\_leakage\_sort(leakage\_sort); //将leakage\_sort结构体数组进行排序

write\_to\_excel(leakage,leakage\_sort,dirinfo.dir\_name[i],dirinfo.dir\_num[i]);

//将采集leakage数据 和 排序后leakage数据写到excel表格

for(j=0;j<8;j++) //清空排序结构体二维数组

{

for(k=0;k<=leakage[0][0].sample\_num;k++)

{

memset(&leakage\_sort[j][k],0,sizeof(leakage\_sort));

}

}

}

system("pause");

return 0;

}

1. **leakage.h**

#ifndef \_\_LEAKAGE\_H

#define \_\_LEAKAGE\_H

#include "fram.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Copyright (c) 2020, longsys \*/

/\* All rights reserved. \*/

/\* Project Name: Leakage\_Test\*/

/\* Author: gaozhe\*/

/\* Date: 2020.03.26\*/

/\* Version:v1.0 \*/

/\* Abstract：实现Leakage的模式识别，数据提取以及数据排序 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define LINE 1024 //文本一行字节数

#define LINE\_COUNT\_MAX 500 //最大读取文本行数

extern char\* select\_file(int num); //选择 log文件

extern struct Fileinfo scan\_file(); //扫描文件夹目录，返回目录信息结构体

extern void get\_leakage\_struct(FILE\* fp,struct Leakage\* leakage);

//实现Leakage模式识别，获取leakage数据信息

extern void get\_leakage\_sort(struct Leakage leakage\_sort[][150]);

//获取leakage\_sort排序数据信息

#endif

1. **leakage.c**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<stdbool.h>

#include "leakage.h"

#include "fram.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Copyright (c) 2020, longsys \*/

/\* All rights reserved. \*/

/\* Project Name: Leakage\_Test\*/

/\* Author: gaozhe\*/

/\* Date: 2020.03.26\*/

/\* Version:v1.0 \*/

/\* Abstract：实现Leakage的模式识别及数据提取以及数据排序 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

extern int mode\_num;

extern int pin\_num;

/\*选择log文件\*/

char\* select\_file(int filename\_num)

{

char filename\_num\_buf[8];

static char filename\_buf[20];

sprintf(filename\_num\_buf,"%d",filename\_num); //文件名称识别

strcpy(filename\_buf,"TRLogDetailed\_MST1\_DUT0");

strcat(filename\_buf,filename\_num\_buf);

strcat(filename\_buf,".txt");

return filename\_buf;

}

/\*扫描文件夹目录\*/

struct Fileinfo scan\_file()

{

int i=0;

Int j=0;

int count=0;

int flag=0;

char path[100] = "./\*";

char \*p;

struct Fileinfo info;

struct \_finddata\_t fa; //文件目录信息结构体

long handle;

handle=\_findfirst(path,&fa);

do

{

if(fa.attrib == \_A\_ARCH||\_A\_HIDDEN||\_A\_RDONLY||\_A\_SUBDIR||\_A\_SYSTEM)

{

for(count=0;count<strlen(fa.name);count++)

{

if(fa.name[count]=='-')

{

flag++;

}

if(flag==3)

{

strcpy(info.dir\_name[i],fa.name);

p=strtok(fa.name,"-"); //样品编号赋值

info.dir\_num[i][0]=atoi(p);

for(j=1;j<4;j++)

{

p=strtok(NULL,"-");

info.dir\_num[i][j]=atoi(p);

}

i++;

break;

}

}

flag=0;

}

}while(\_findnext(handle,&fa) == 0); //成功找到时返回0

info.dir\_sum=i; //计算有效文件夹数量

\_findclose(handle);

return info;

}

/\*获取leakage结构体信息\*/

void get\_leakage\_struct(FILE\* fp,struct Leakage\* leakage)

{

int i=0;

int j=0;

int m=0;

int count=0;

int pin\_count=0;

int line\_count=0;

char buf[1024]; //读取每一行

char pin\_name\_buf[8]={0};

char channel\_name\_buf[2];

char temp\_buf[1];

char \*lp4=NULL;

char \*lp3=NULL;

char \*cs2=NULL;

char \*cs1=NULL;

bool flag\_mode=false; //模式识别标志位

bool lp4\_mode=false;

bool lp3\_mode=false;

bool cs2\_mode=false;

bool cs1\_mode=false;

bool flag\_pin=false; //全部引脚信息采集完成标志位

while(flag\_pin==false&&line\_count<LINE\_COUNT\_MAX)

{

fgets(buf,LINE,fp); //读取每一行数据

line\_count++; //记录已读取文本行数

if(mode\_num==0)

{

if(flag\_mode==false)

{

lp4=strstr(buf,"LPDDR4");

lp3=strstr(buf,"LPDDR3");

cs2=strstr(buf,"2 CS");

cs1=strstr(buf,"1 CS");

if(lp4!=NULL)

{

lp4\_mode=true;

lp4=NULL;

}

if(lp3!=NULL)

{

lp3\_mode=true;

lp3=NULL;

}

if(cs2!=NULL)

{

cs2\_mode=true;

cs2=NULL;

}

if(cs1!=NULL)

{

cs1\_mode=true;

cs1=NULL;

}

if(lp4\_mode&&cs1\_mode)

{

flag\_mode=true; //LPDDR4 1 CS

mode\_num=1; //模式 1

pin\_num=66; //66pins

printf("\n 已识别 LPDDR4 1 CS Mode ！ \n");

}

else if(lp4\_mode&&cs2\_mode)

{

flag\_mode=true; //LPDDR4 2 CS

mode\_num=2; //模式 2

pin\_num=68; //68pins

printf("\n 已识别 LPDDR4 2 CS Mode ！ \n");

}

else if(lp3\_mode&&cs1\_mode)

{

flag\_mode=true; //LPDDR3 1 CS

mode\_num=3; //模式 3

pin\_num=118; //118pins

printf("\n 已识别 LPDDR3 1 CS Mode ！ \n");

}

else if(lp3\_mode&&cs2\_mode)

{

flag\_mode=true; //LPDDR3 2 CS

mode\_num=4; //模式 4

pin\_num=122; //122pins

printf("\n 已识别 LPDDR3 2 CS Mode ！ \n");

}

else

{

flag\_mode=false; //非法样品类型

mode\_num=0; //模式 0

}

}

}

if(mode\_num==1||mode\_num==2) //LPDDR4 1/2 CS模式数据识别与提取

{

for(i=0;i<100;i++)

{

if(buf[i]=='\_'&&buf[i+2]==' '&&buf[30]=='L') //有效信息判断条件

{

pin\_count++;

if(pin\_count==pin\_num)

flag\_pin=true;

for(j=0;j<i;j++)

{

memset(temp\_buf,0,1);

sprintf(temp\_buf,"%c",buf[j]); //生成引脚名称

strcat(pin\_name\_buf,temp\_buf);

memset(temp\_buf,0,1);

}

sprintf(channel\_name\_buf,"%c",buf[i+1]); //生成通道名

struct Leakage \*temp\_leakage

= someprocess\_1(buf,pin\_name\_buf,channel\_name\_buf);

memcpy(&leakage[++count],temp\_leakage,sizeof(leakage[++count]));

//Leakage结构体赋值

if(atof(leakage[1].value)<=0.20)

//判断是否有效数据，无效则清空结构体并退出循环

{

memset(&leakage[1].pin\_name,0,10);

memset(&leakage[1].channel,0,10);

memset(&leakage[1].type,0,10);

memset(&leakage[1].value,0,10);

goto Label1;

}

leakage[0].sample\_num=count; //计算结构体数量

memset(pin\_name\_buf,0,10);

memset(channel\_name\_buf,0,2);

}

}

memset(buf,0,1024);

}

else if(mode\_num==3||mode\_num==4)

//LPDDR3 1 CS模式数据识别与提取

{

for(i=0;i<100;i++)

{

if(buf[i]!=' '&&buf[i+1]==' '&&buf[i+2]==' '&&buf[30]=='L')

//有效信息判断条件

{

pin\_count++;

if(pin\_count==pin\_num)

flag\_pin=true;

for(j=0;j<=i;j++)

{

memset(temp\_buf,0,1);

sprintf(temp\_buf,"%c",buf[j]); //生成引脚名称

strcat(pin\_name\_buf,temp\_buf);

memset(temp\_buf,0,1);

}

struct Leakage \*temp\_leakage

= someprocess\_2(buf,pin\_name\_buf);

memcpy(&leakage[++count],temp\_leakage,sizeof(leakage[++count]));

//Leakage结构体赋值

if(atof(leakage[1].value)<=0.20)

//判断是否有效数据，无效则清空结构体并退出循环

{

memset(&leakage[1].pin\_name,0,10);

memset(&leakage[1].type,0,10);

memset(&leakage[1].value,0,10);

goto Label1;

}

leakage[0].sample\_num=count; //计算结构体数量

memset(pin\_name\_buf,0,10);

memset(channel\_name\_buf,0,2);

}

}

memset(buf,0,1024);

}

else

{

;

}

}

Label1: //识别为无效数据，直接跳转退出文本信息采集

;

}

/\*排序，生成新二维结构体数组\*/

void get\_leakage\_sort(struct Leakage leakage\_sort[][150])

{

int count;

int leakage\_num;

int i;

int j;

float k;

float m;

struct Leakage temp\_leakage;

leakage\_num=leakage\_sort[0][0].sample\_num;

for(count=0;count<8;count++)

{

for(i=1;i<leakage\_num+1;i++) //冒泡排序法

{

for(j=1;j<leakage\_num-i+1;j++)

{

k=atof(leakage\_sort[count][j].value);

m=atof(leakage\_sort[count][j+1].value);

if(k<m)

{

temp\_leakage=leakage\_sort[count][j];

leakage\_sort[count][j]=leakage\_sort[count][j+1];

leakage\_sort[count][j+1]=temp\_leakage;

}

}

}

}

}

1. **fram.h**

#ifndef \_\_FRAM\_H

#define \_\_FRAM\_H

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Copyright (c) 2020, longsys \*/

/\* All rights reserved. \*/

/\* Project Name: Leakage\_Test\*/

/\* Author: gaozhe\*/

/\* Date: 2020.03.26\*/

/\* Version:v1.0 \*/

/\* Abstract：声明Leakage数据类型结构体和结构体赋值函数 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

struct Leakage //声明Leakage引脚信息结构体

{

int sample\_num;

char pin\_name[10];

char channel[10];

char type[10];

char value[10];

};

struct Fileinfo //声明文件目录信息结构体

{

char dir\_name[30][50]; //存储目录信息

int dir\_num[30][4]; //存储样品标号信息

int dir\_sum; //计算目录数

};

extern struct Leakage leakage[8][150];

//定义leakage数据采集二维结构体数组

extern struct Leakage leakage\_sort[8][150];

//定义leakage数据排序二维结构体数组

extern struct Leakage\*

someprocess\_1(char \*buf,char \*pin\_name\_buf,char \*channel\_name\_buf);

extern struct Leakage\* someprocess\_2(char \*buf,char \*pin\_name\_buf);

#endif

1. **fram.c**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include "leakage.h"

#include "fram.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Copyright (c) 2020, longsys \*/

/\* All rights reserved. \*/

/\* Project Name: Leakage\_Test\*/

/\* Author: gaozhe\*/

/\* Date: 2020.03.26\*/

/\* Version:v1.0 \*/

/\* Abstract：定义Leakage数据类型结构体和结构体赋值函数 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

struct Leakage leakage[8][150]; //定义Leakage结构体二维数组

struct Leakage leakage\_sort[8][150]; //定义Leakage排序结构体二维数组

struct Leakage\* someprocess\_1

(char \*buf,char \*pin\_name\_buf,char \*channel\_name\_buf)

//LPDDR4 1/2 CS 结构体赋值

{

struct Leakage info;

memset(&info,0,sizeof(info));

memcpy(&info.pin\_name,pin\_name\_buf,8);

memcpy(&info.channel,channel\_name\_buf,2);

memcpy(&info.type,buf+35,3);

memcpy(&info.value,buf+55,7);

return &info;

}

struct Leakage\* someprocess\_2(char \*buf,char \*pin\_name\_buf)

//LPDDR3 1/2 CS 结构体赋值

{

struct Leakage info;

memset(&info,0,sizeof(info));

memcpy(&info.pin\_name,pin\_name\_buf,8);

memcpy(&info.type,buf+35,3);

memcpy(&info.value,buf+55,7);

return &info;

}

1. **excel.h**

#ifndef \_\_EXCEL\_H

#define \_\_EXCEL\_H

#include <stdio.h>

#include "fram.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Copyright (c) 2020, longsys \*/

/\* All rights reserved. \*/

/\* Project Name: Leakage\_Test\*/

/\* Author: gaozhe\*/

/\* Date: 2020.03.26\*/

/\* Version:v1.0 \*/

/\* Abstract：Excel数据打印函数声明 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

extern void showtitle\_1\_mode1(SheetHandle sheet,int \*num);

//打印 模式1/2 表格标题栏 1

extern void showtitle\_2\_mode1(SheetHandle sheet,int \*num);

//打印 模式1/2 表格标题栏 2

extern void showtitle\_1\_mode3(SheetHandle sheet,int \*num);

//打印 模式 3 表格标题栏 1

extern void showtitle\_2\_mode3(SheetHandle sheet,int \*num);

//打印 模式 3 表格标题栏 2

extern void write\_to\_excel

(struct Leakage leakage[][150],struct Leakage leakage\_sort[][150],

char \*dirname\_buf,int \*dir\_num);

//生成Excel表格数据

#endif

1. **excel.c**

#include <stdio.h>

#include <string.h>

#include <conio.h>

#include "include\_c/libxl.h"

#include "excel.h"

#include "fram.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Copyright (c) 2020, longsys \*/

/\* All rights reserved. \*/

/\* Project Name: Leakage\_Test\*/

/\* Author: gaozhe\*/

/\* Date: 2020.03.26\*/

/\* Version:v1.0 \*/

/\* Abstract：Excel数据打印函数定义 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

extern int mode\_num; //模式标识符在外部文件中定义

/\*打印 模式1/2 表格标题栏 1\*/

void showtitle\_1\_mode1(SheetHandle sheet,int \*num)

{

int i;

char buf[10]="样品#";

char buf\_name[10]={0};

char buf\_num[10]={0};

xlSheetWriteStrA(sheet, 1, 1, "通道a", NULL);

xlSheetWriteStrA(sheet, 1, 7, "通道b", NULL);

xlSheetWriteStrA(sheet, 2, 0, "引脚", NULL);

xlSheetWriteStrA(sheet, 2, 1, "类型", NULL);

xlSheetWriteStrA(sheet, 2, 7, "类型", NULL);

for(i=0;i<4;i++)

{

strcpy(buf\_name,buf);

sprintf(buf\_num,"%d",num[i]);

strcat(buf\_name,buf\_num);

xlSheetWriteStrA(sheet, 2, i+2, buf\_name, NULL);

xlSheetWriteStrA(sheet, 2, i+8, buf\_name, NULL);

memset(buf\_name,0,strlen(buf\_name));

}

}

/\*打印 模式1/2 表格标题栏 2\*/

void showtitle\_2\_mode1(SheetHandle sheet,int \*num)

{

int i;

char buf[10]="样品#";

char buf\_name[10]={0};

char buf\_num[10]={0};

for(i=0;i<4;i++)

{

xlSheetWriteStrA(sheet, 0, 1+7\*i, "通道a", NULL);

xlSheetWriteStrA(sheet, 0, 3+7\*i, "通道b", NULL);

xlSheetWriteStrA(sheet, 1, 0+7\*i, "引脚", NULL);

xlSheetWriteStrA(sheet, 1, 1+7\*i, "类型", NULL);

xlSheetWriteStrA(sheet, 1, 2+7\*i, "Leakage", NULL);

xlSheetWriteStrA(sheet, 1, 3+7\*i, "引脚", NULL);

xlSheetWriteStrA(sheet, 1, 4+7\*i, "类型", NULL);

xlSheetWriteStrA(sheet, 1, 5+7\*i, "Leakage", NULL);

}

for(i=0;i<4;i++)

{

strcpy(buf\_name,buf);

sprintf(buf\_num,"%d",num[i]);

strcat(buf\_name,buf\_num);

xlSheetWriteStrA(sheet, 0, 0+7\*i, buf\_name, NULL);

memset(buf\_name,0,strlen(buf\_name));

}

}

/\*打印 模式3 表格标题栏 1\*/

void showtitle\_1\_mode3(SheetHandle sheet,int \*num)

{

int i;

char buf[10]="样品#";

char buf\_name[10]={0};

char buf\_num[10]={0};

xlSheetWriteStrA(sheet, 1, 0, "引脚", NULL);

xlSheetWriteStrA(sheet, 1, 1, "类型", NULL);

for(i=0;i<4;i++)

{

strcpy(buf\_name,buf);

sprintf(buf\_num,"%d",num[i]);

strcat(buf\_name,buf\_num);

xlSheetWriteStrA(sheet, 1, i+2, buf\_name, NULL);

memset(buf\_name,0,strlen(buf\_name));

}

}

/\*打印 模式3 表格标题栏 2\*/

void showtitle\_2\_mode3(SheetHandle sheet,int \*num)

{

int i;

char buf[10]="样品#";

char buf\_name[10]={0};

char buf\_num[10]={0};

for(i=0;i<4;i++)

{

xlSheetWriteStrA(sheet, 1, 0+4\*i, "引脚", NULL);

xlSheetWriteStrA(sheet, 1, 1+4\*i, "类型", NULL);

xlSheetWriteStrA(sheet, 1, 2+4\*i, "Leakage", NULL);

}

for(i=0;i<4;i++)

{

strcpy(buf\_name,buf);

sprintf(buf\_num,"%d",num[i]);

strcat(buf\_name,buf\_num);

xlSheetWriteStrA(sheet, 0, 0+4\*i, buf\_name, NULL);

memset(buf\_name,0,strlen(buf\_name));

}

}

/\*Excel写函数，生成表格数据\*/

void write\_to\_excel(struct Leakage leakage[][150],

struct Leakage leakage\_sort[][150],char \*dirname\_buf,int \*dir\_num)

{

int i;

BookHandle book = xlCreateBook();

xlBookSetKey(book,"hello","windows-21202e0009c9e10465be636ca9k1lchd");

if(book)

{

SheetHandle sheet1;

SheetHandle sheet2;

sheet1 = xlBookAddSheet(book, "leakage info", 0);

if(sheet1)

{

if(mode\_num==1||mode\_num==2)

{

if(mode\_num==1)

{

xlSheetWriteStrA(sheet1, 0, 0, "LPDDR4", NULL);

//表格开头打印芯片信息

xlSheetWriteStrA(sheet1, 0, 1, "1 CS", NULL);

}

else

{

xlSheetWriteStrA(sheet1, 0, 0, "LPDDR4", NULL);

xlSheetWriteStrA(sheet1, 0, 1, "2 CS", NULL);

}

showtitle\_1\_mode1(sheet1,dir\_num); //LP4型号表格1标题栏

for(i=1;i<=leakage[0][0].sample\_num;i++)

{

if(i%2==1) //打印 out类型数据信息

{

xlSheetWriteStrA(sheet1, (i+5)/2, 0, leakage[0][i].pin\_name, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 1, leakage[0][i].type, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 2, leakage[1][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 3, leakage[3][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 4, leakage[5][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 5, leakage[7][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 7, leakage[0][1].type, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 8, leakage[0][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 9, leakage[2][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 10, leakage[4][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 11, leakage[6][i].value, NULL);

}

}

for(i=1;i<=leakage[0][0].sample\_num;i++)

{

if(i%2==0) //打印 in类型数据信息

{

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

0, leakage[0][i].pin\_name, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

1, leakage[0][i].type, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

2, leakage[1][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

3, leakage[3][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

4, leakage[5][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

5, leakage[7][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

7, leakage[0][1].type, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

8, leakage[0][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

9, leakage[2][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

10, leakage[4][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

11, leakage[6][i].value, NULL);

}

}

}

else if(mode\_num==3||mode\_num==4)

{

if(mode\_num==3)

{

xlSheetWriteStrA(sheet1, 0, 0, "LPDDR3", NULL);

xlSheetWriteStrA(sheet1, 0, 1, "1 CS", NULL);

}

else

{

xlSheetWriteStrA(sheet1, 0, 0, "LPDDR3", NULL);

xlSheetWriteStrA(sheet1, 0, 1, "2 CS", NULL);

}

showtitle\_1\_mode3(sheet1,dir\_num); //LP3型号表格1标题栏

for(i=1;i<=leakage[0][0].sample\_num;i++)

{

if(i%2==1) //打印 out类型数据信息

{

xlSheetWriteStrA(sheet1, (i+5)/2, 0, leakage[0][i].pin\_name, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 1, leakage[0][i].type, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 2, leakage[0][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 3, leakage[2][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 4, leakage[4][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+5)/2, 5, leakage[6][i].value, NULL);

}

}

for(i=1;i<=leakage[0][0].sample\_num;i++)

{

if(i%2==0) //打印 in类型数据信息

{

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

0, leakage[0][i].pin\_name, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

1, leakage[0][i].type, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

2, leakage[0][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

3, leakage[2][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

4, leakage[4][i].value, NULL);

xlSheetWriteStrA(sheet1, (i+leakage[0][0].sample\_num+4)/2,

5, leakage[6][i].value, NULL);

}

}

}

else

;

}

sheet2 = xlBookAddSheet(book,"leakage sort",0);

if(sheet2)

{

if(mode\_num==1||mode\_num==2)

{

showtitle\_2\_mode1(sheet2,dir\_num); //LP4型号表格2标题栏

for(i=1;i<=leakage[0][0].sample\_num;i++)

//打印已排序leakage数据信息

{

xlSheetWriteStrA(sheet2, i+1, 0, leakage\_sort[1][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 1, leakage\_sort[1][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 2, leakage\_sort[1][i].value, NULL);

xlSheetWriteStrA(sheet2, i+1, 3, leakage\_sort[0][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 4, leakage\_sort[0][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 5, leakage\_sort[0][i].value, NULL);

xlSheetWriteStrA(sheet2, i+1, 7, leakage\_sort[3][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 8, leakage\_sort[3][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 9, leakage\_sort[3][i].value, NULL);

xlSheetWriteStrA(sheet2, i+1, 10, leakage\_sort[2][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 11, leakage\_sort[2][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 12, leakage\_sort[2][i].value, NULL);

xlSheetWriteStrA(sheet2, i+1, 14, leakage\_sort[5][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 15, leakage\_sort[5][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 16, leakage\_sort[5][i].value, NULL);

xlSheetWriteStrA(sheet2, i+1, 17, leakage\_sort[4][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 18, leakage\_sort[4][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 19, leakage\_sort[4][i].value, NULL);

xlSheetWriteStrA(sheet2, i+1, 21, leakage\_sort[7][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 22, leakage\_sort[7][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 23, leakage\_sort[7][i].value, NULL);

xlSheetWriteStrA(sheet2, i+1, 24, leakage\_sort[6][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 25, leakage\_sort[6][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 26, leakage\_sort[6][i].value, NULL);

}

}

else if(mode\_num==3||mode\_num==4)

{

showtitle\_2\_mode3(sheet2,dir\_num); //LP3型号表格2标题栏

for(i=1;i<=leakage[0][0].sample\_num;i++)

//打印已排序leakage数据信息

{

xlSheetWriteStrA(sheet2, i+1, 0, leakage\_sort[0][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 1, leakage\_sort[0][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 2, leakage\_sort[0][i].value, NULL);

xlSheetWriteStrA(sheet2, i+1, 4, leakage\_sort[2][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 5, leakage\_sort[2][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 6, leakage\_sort[2][i].value, NULL);

xlSheetWriteStrA(sheet2, i+1, 8, leakage\_sort[4][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 9, leakage\_sort[4][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 10, leakage\_sort[4][i].value, NULL);

xlSheetWriteStrA(sheet2, i+1, 12, leakage\_sort[6][i].pin\_name, NULL);

xlSheetWriteStrA(sheet2, i+1, 13, leakage\_sort[6][i].type, NULL);

xlSheetWriteStrA(sheet2, i+1, 14, leakage\_sort[6][i].value, NULL);

}

}

else

;

}

strcat(dirname\_buf,"/Leakage\_Test.xls");

if(xlBookSave(book,dirname\_buf))

printf("\n 已创建Excel表格 %s \n",dirname\_buf);

xlBookRelease(book);

}

}