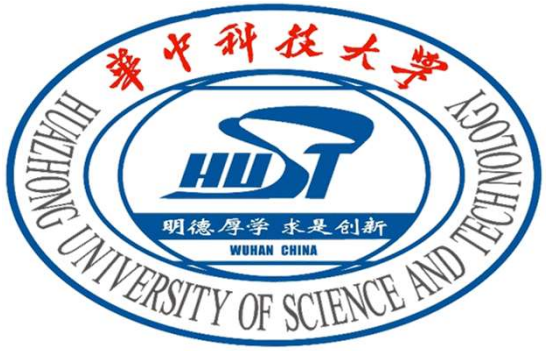
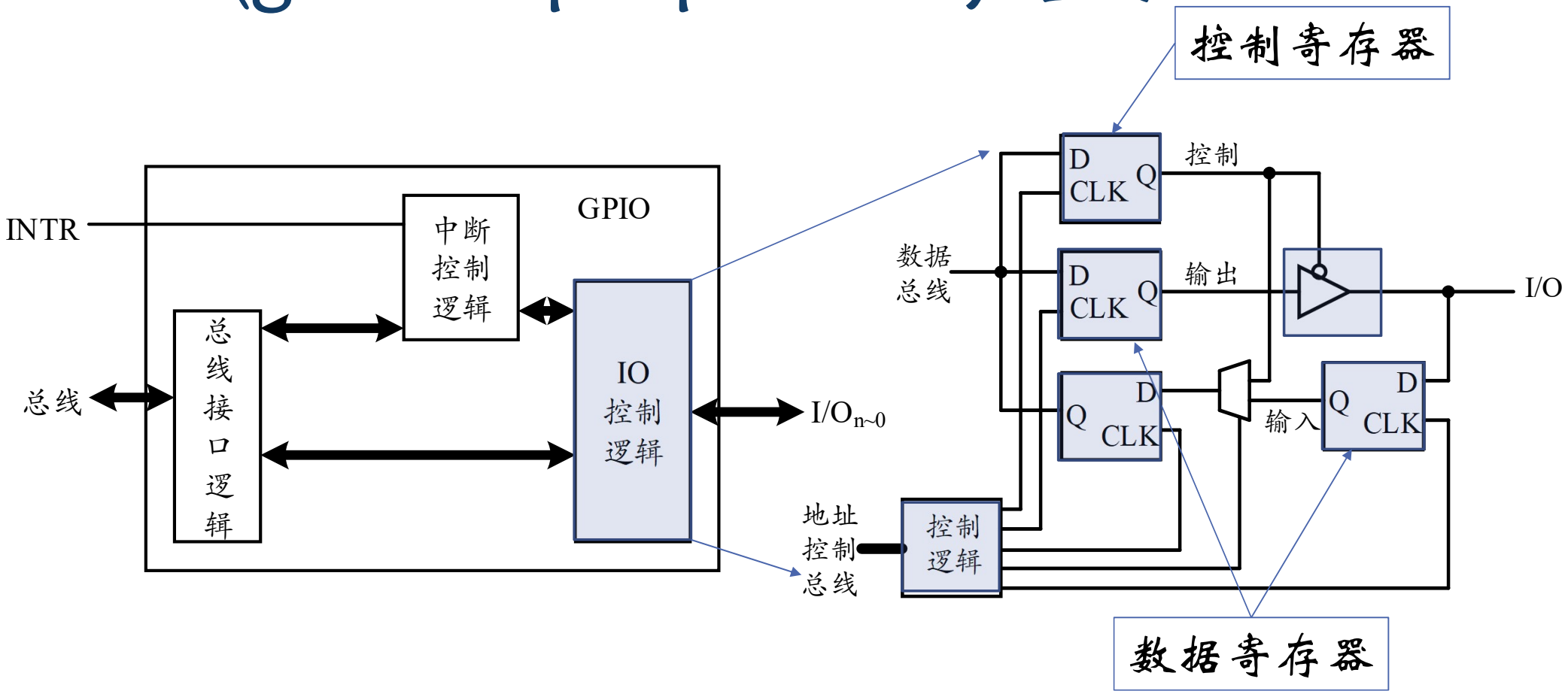
微机原理与接口技术

通用并行IO接口GPIO

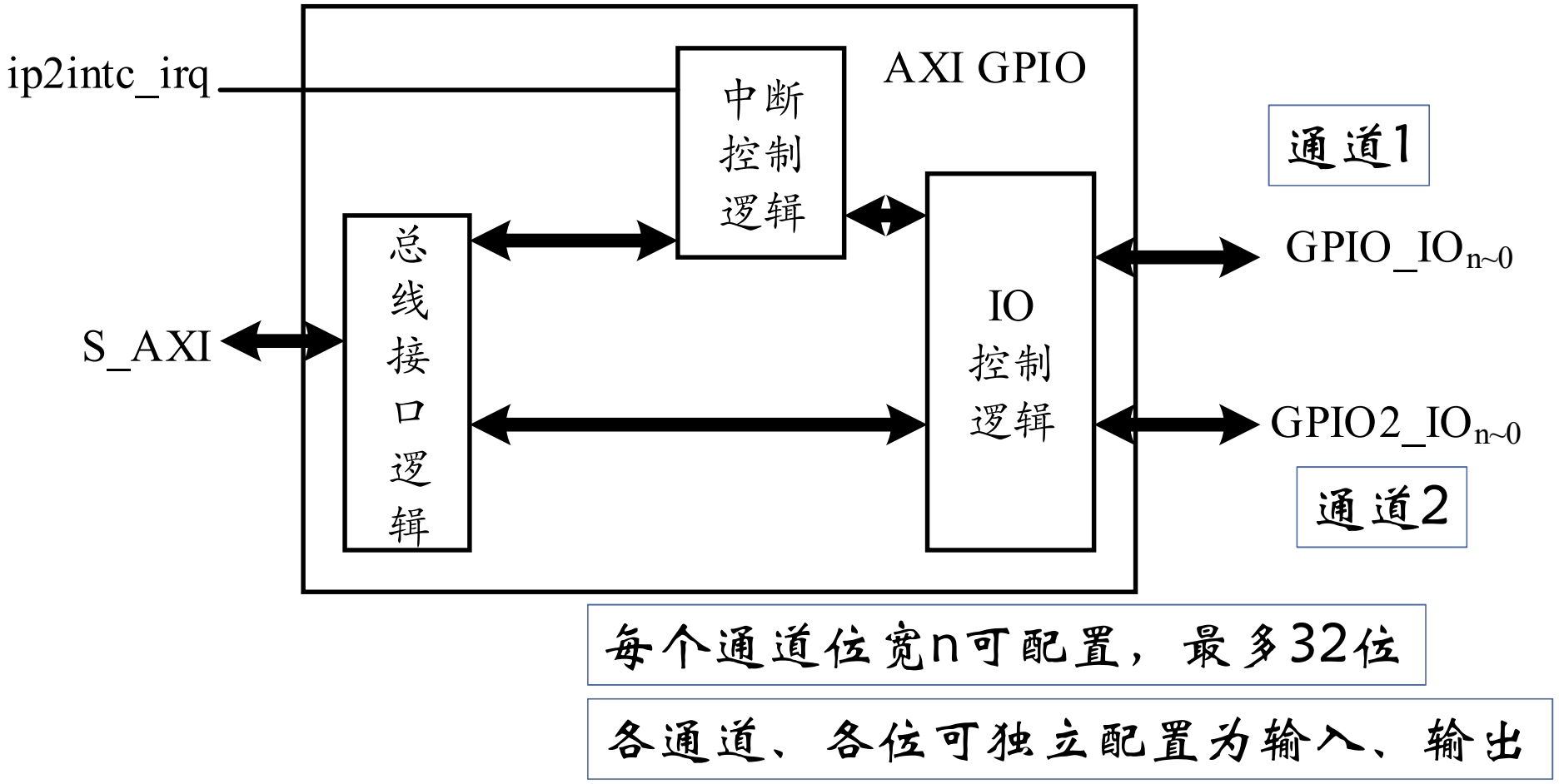
华中科技大学　左冬红



GPIO(general purpose IO)结构



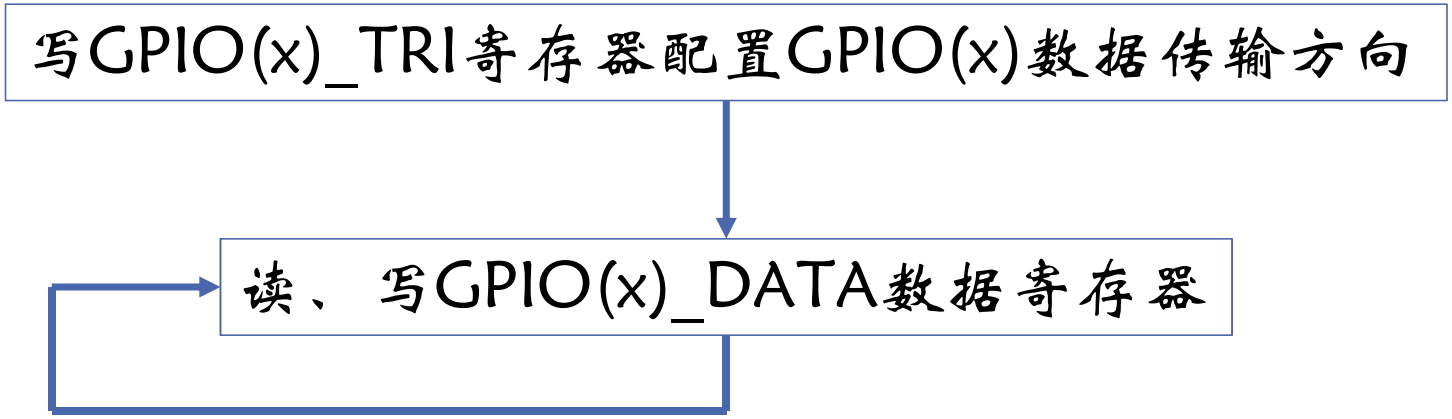
AXI GPIO


ip2intc_irq
中断
控制
逻辑
AXI GPIO
通道1
总
线
接
口
逻
辑
IO
控制
逻辑
GPIO_IO n~0
S_AXI
GPIO2_IO n~0
通道2
每个通道位宽n可配置，最多32位
各通道、各位可独立配置为输入、输出

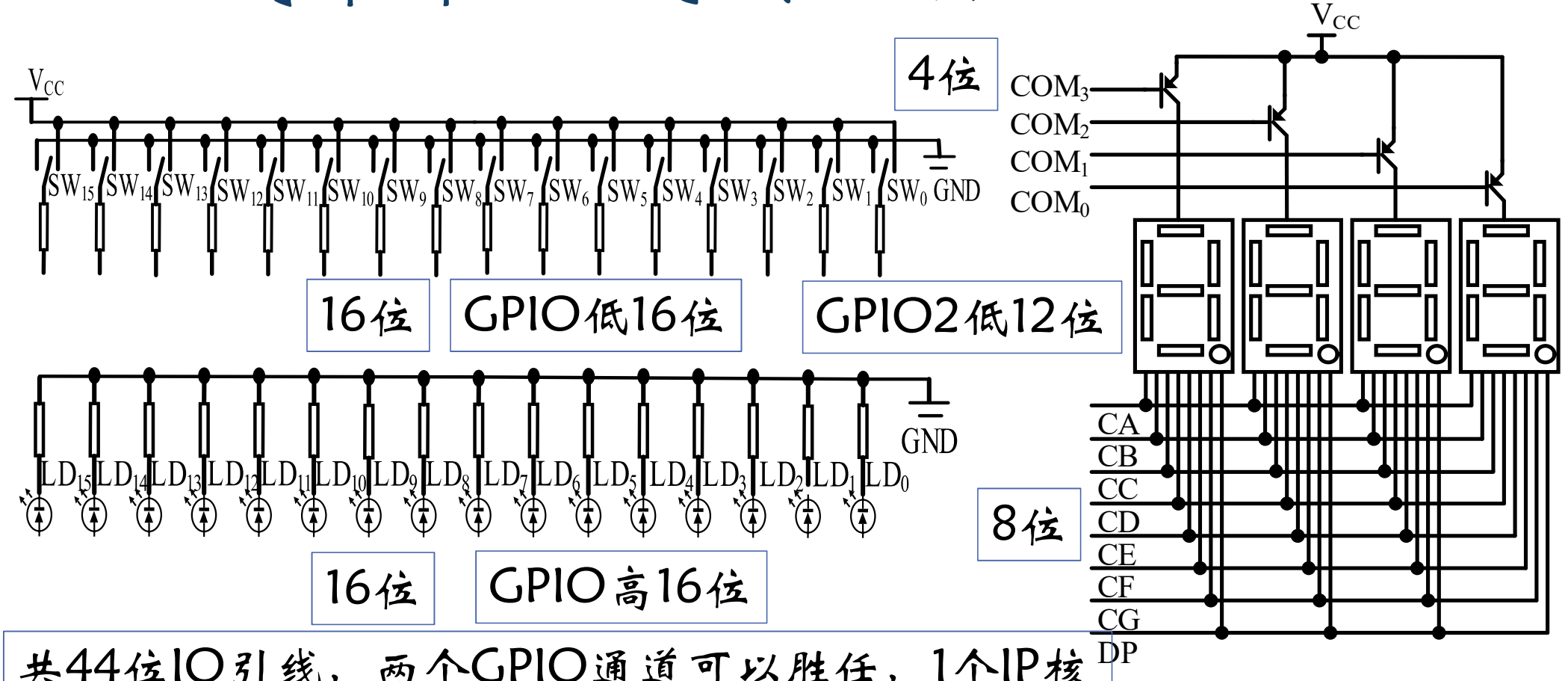
GPIO寄存器存储空间映射

寄存器名称	偏移地址	初始值	含义
GPIO_DATA	0x0	0	32位宽，通道GPIO_IO数据寄存器，D_i对应$GPIO_IO_i$
GPIO_TRI	0x4	0xffffffff	32位宽，通道GPIO_IO控制寄存器，D_i对应$GPIO_IO_i$的传输方向控制。1-输入；0-输出。
GPIO2_DATA	0x8	0	32位宽，通道GPIO2_IO数据寄存器，D_i对应$GPIO2_IO_i$
GPIO2_TRI	0xc	0xffffffff	32位宽，通道GPIO2_IO控制寄存器，D_i对应$GPIO2_IO_i$的传输方向控制。1-输入；0-输出。

GPIO程序控制流程



GPIO简单并行数字外设接口电路示例



4位

16位 GPIO低16位 GPIO2低12位

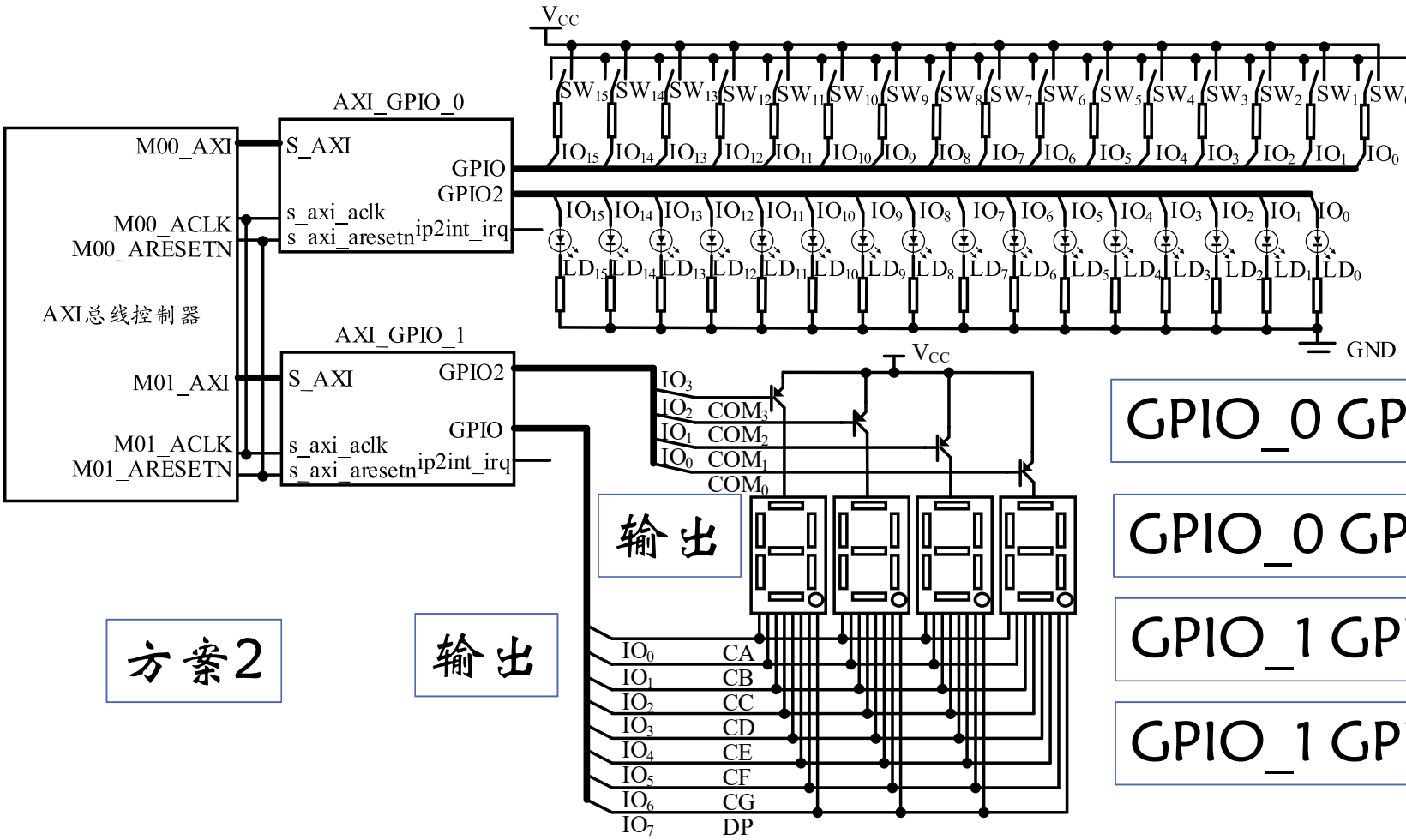
16位 GPIO高16位

8位

共44位IO引线，两个GPIO通道可以胜任，1个IP核

不同外设采用不同GPIO通道，2个IP核

GPIO简单并行数字外设接口电路



GPIO_0 GPIO_TRI 0xffff

GPIO_0 GPIO2_TRI 0x0

GPIO_1 GPIO_TRI 0x0

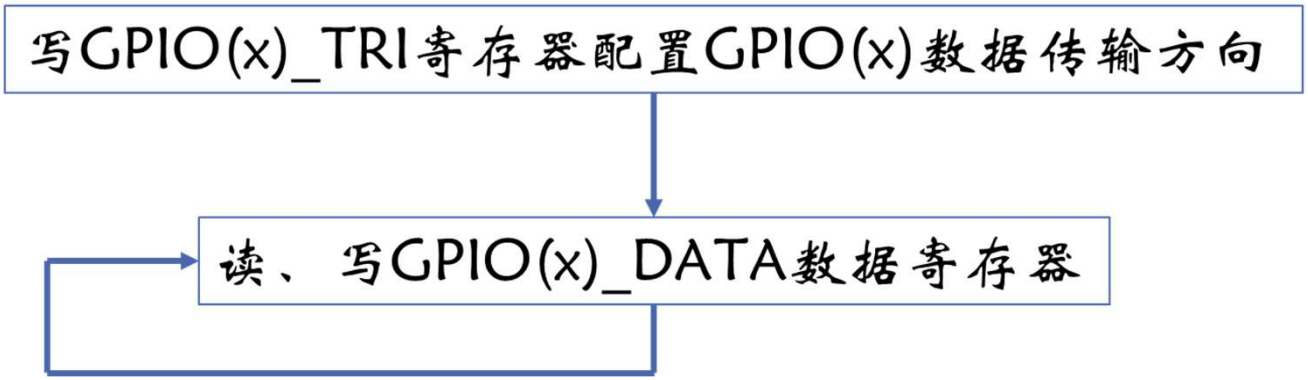
GPIO_1 GPIO2_TRI 0x0

GPIO存储空间映射

AXI_GPIO_0接口基地址为0x4000 0000，
AXI_GPIO_1接口基地址为0x4001 0000

GPIO	寄存器	地址	有效数据位	读写控制
AXI_GPIO_0	GPIO_DATA	0x4000 0000	$D_{15\sim0}$	读
	GPIO_TRI	0x4000 0004	$D_{15\sim0}$	写
	GPIO2_DATA	0x4000 0008	$D_{15\sim0}$	写
	GPIO2_TRI	0x4000 000c	$D_{15\sim0}$	写
AXI_GPIO_1	GPIO_DATA	0x4001 0000	$D_{7\sim0}$	写
	GPIO_TRI	0x4001 0004	$D_{7\sim0}$	写
	GPIO2_DATA	0x4001 0008	$D_{3\sim0}$	写
	GPIO2_TRI	0x4001 000c	$D_{3\sim0}$	写

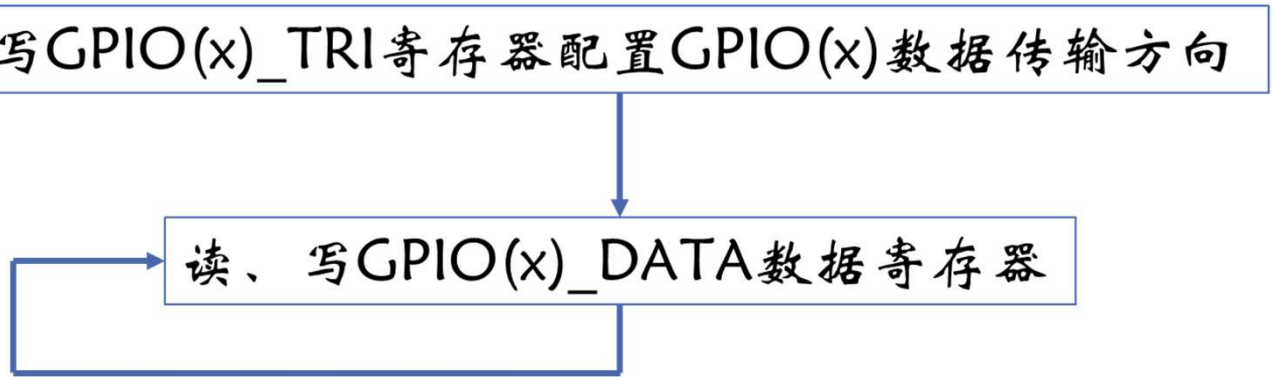
GPIO简单并行数字外设功能需求




```
unsigned short key;
Xil_Out16(0x40000004,0xffff);
key = Xil_In16(0x40000000)&0xffff;
```

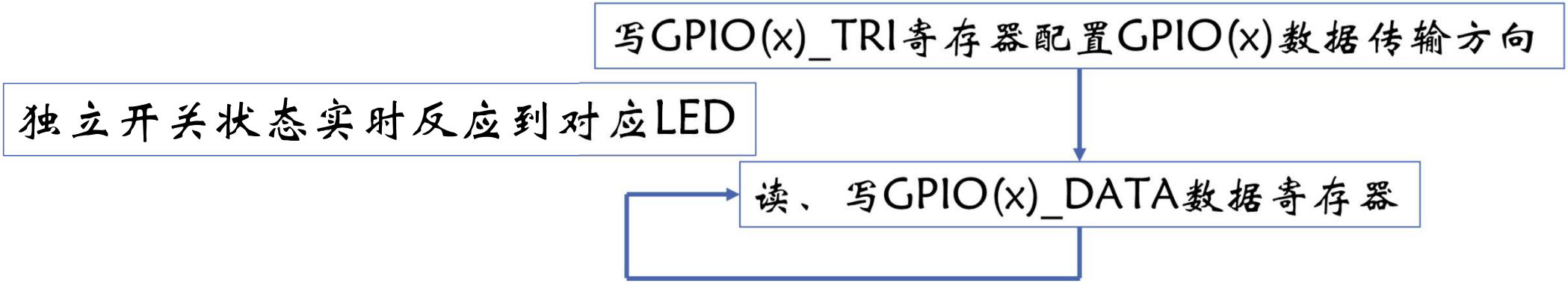
GPIO简单并行数字外设功能需求

16个LED间隔点亮




```
Xil_Out16(0x4000000c,0x0);
Xil_Out16(0x40000008,0x5555);
```

GPIO简单并行数字外设功能需求

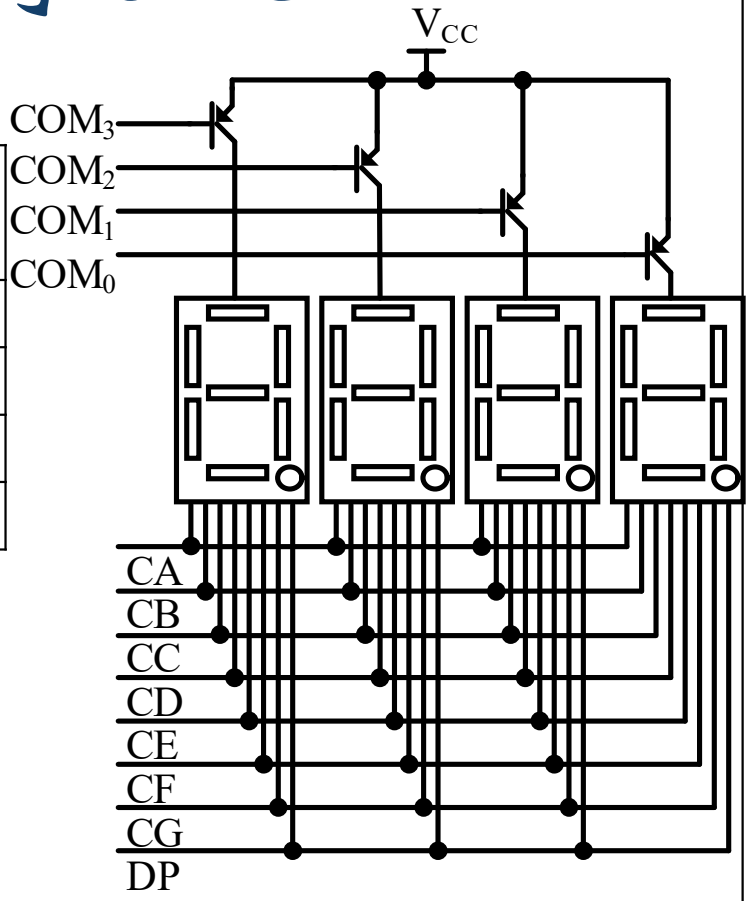



```
Xil_Out16(0x40000004,0xffff);//设置AXI_GPIO_0通道GPIO输入
Xil_Out16(0x4000000c,0x0);//设置AXI_GPIO_0通道GPIO2输出
while (1)
Xil_Out16(0x40000008, Xil_In16(0x40000000)&0xffff);
```

控制4位七段数码管显示数字0~3

数字	IO_7 (DP)	IO_6 (CG)	IO_5 (CF)	IO_4 (CE)	IO_3 (CD)	IO_2 (CC)	IO_1 (CB)	IO_0 (CA)	段码
0	1	1	0	0	0	0	0	0	0xc0
1	1	1	1	1	1	0	0	1	0xf9
2	1	0	1	0	0	1	0	0	0xa4
3	1	0	1	1	0	0	0	0	0xb0

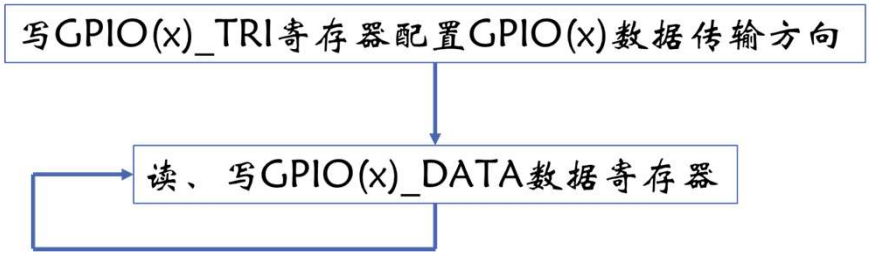
数码管控制信号	IO_3 (COM_3)	IO_2 (COM_2)	IO_1 (COM_1)	IO_0 (COM_0)	位码
COM_3	0	1	1	1	0x7
COM_2	1	0	1	1	0xb
COM_1	1	1	0	1	0xd
COM_0	1	1	1	0	0xe



GPIO简单并行数字外设功能需求

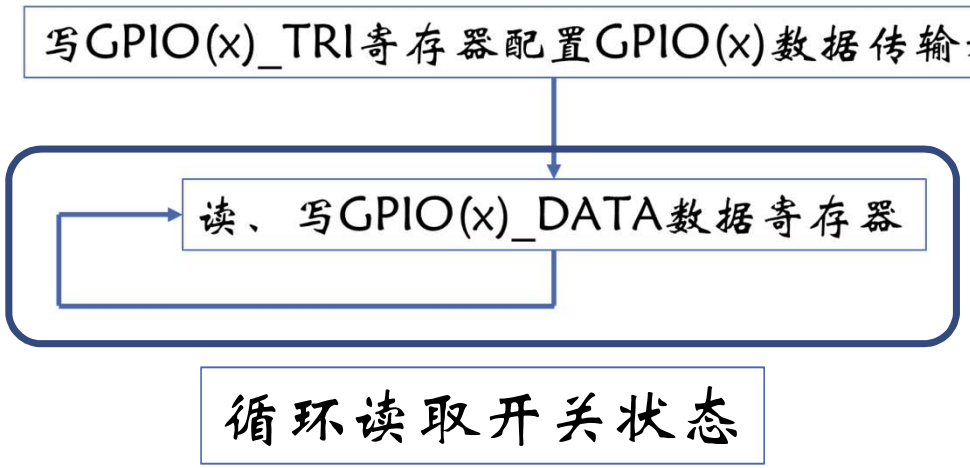
```
unsigned char segcode[8]={0xc0,0xf9,0xa4,0xb0};
unsigned char pos=0xf7;
Xil_Out8(0x40010004,0x0);
Xil_Out8(0x4001000c,0x0);
while(1){              //循环扫描
        for(i=0;i<4;i++) //4位扫描一遍
        {
            Xil_Out8(0x40010000,segcode[i]);//
            Xil_Out8(0x40010008,pos); //
            for(j=0;j<10000;j++); //
            pos=pos>>1;//
        }
        pos=0xf7;  //
    }
```

4位七段数码管显示数字0~3

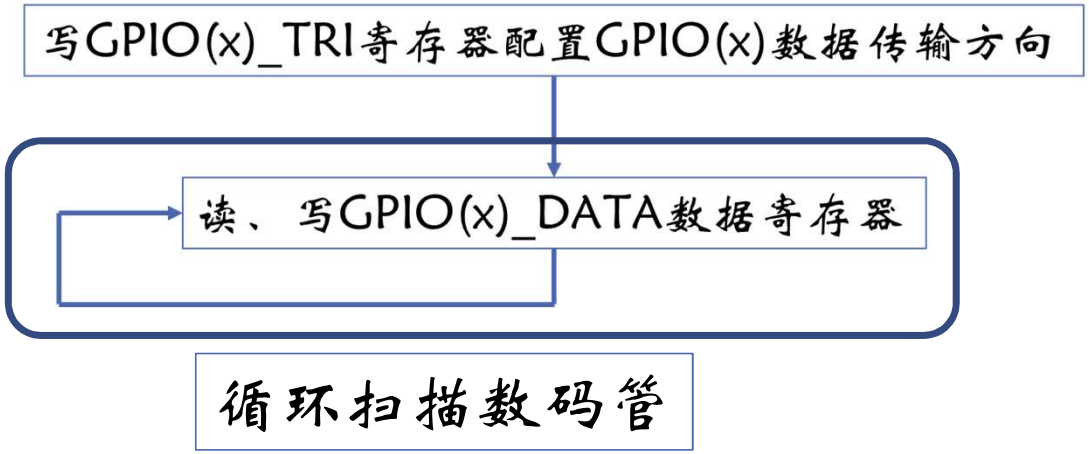


GPIO简单并行数字外设功能需求

同一控制程序将各个独立开关SW15~0状态实时反应到对应LED LD15~0上，同时将16位独立开关SW15~0表示的二进制数以十六进制形式显示在4位七段数码管上

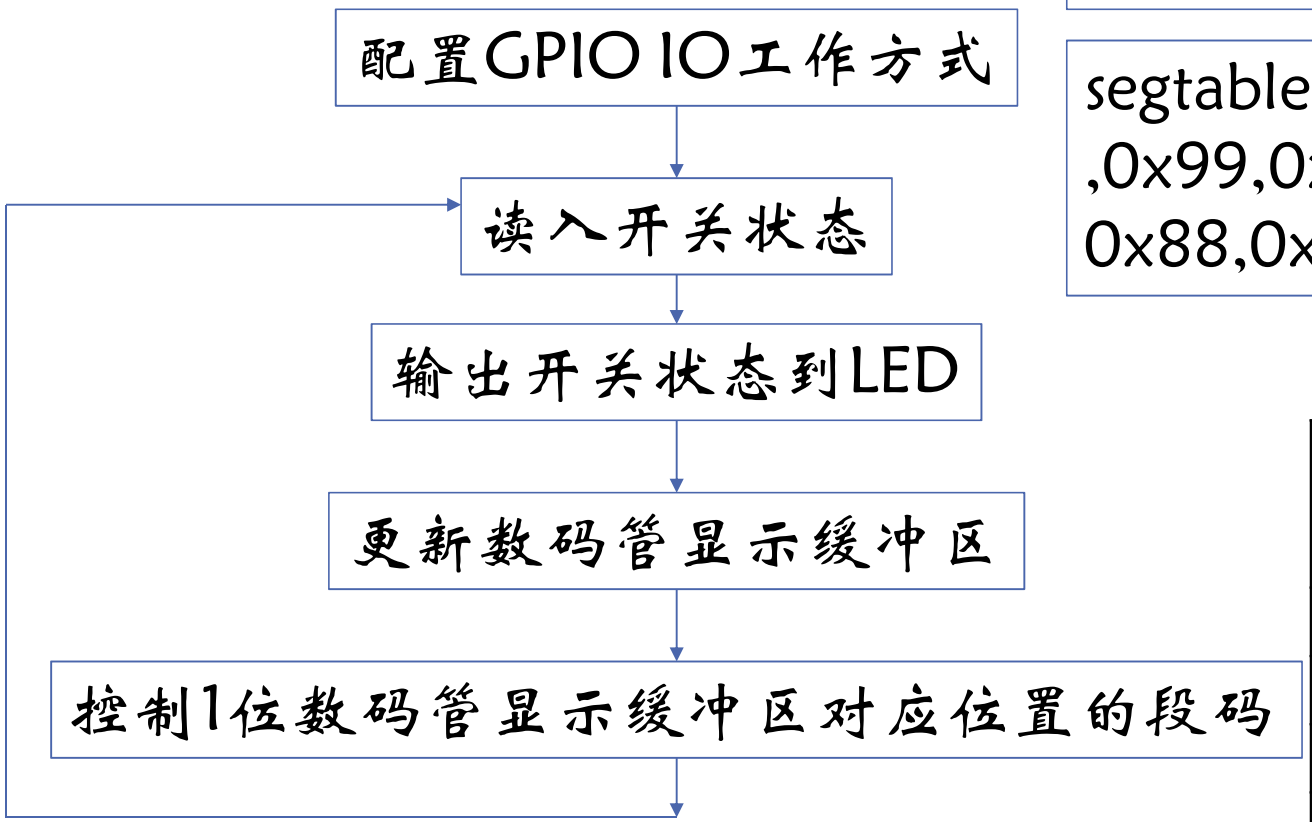


循环读取开关状态



循环扫描数码管

程序流程图(主)



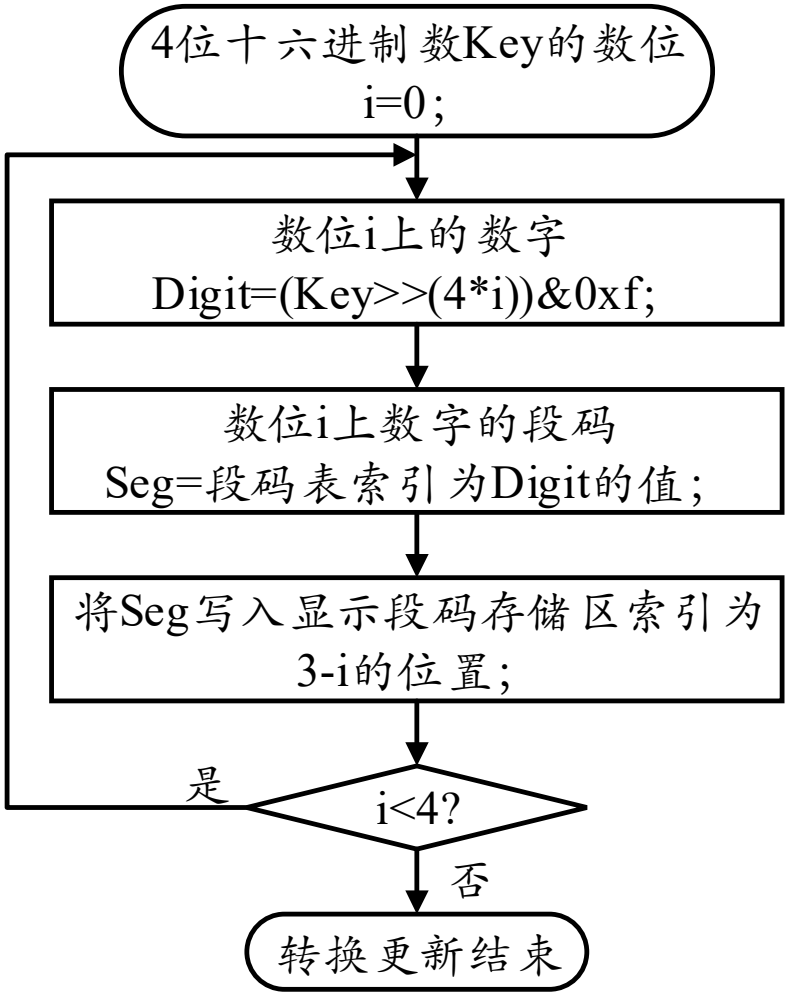
segcode[8]={0xc0,0xf9,0xa4,0xb0}

Sw=0x5678

segtable[16]={0xc0,0xf9,0xa4,0xb0
,0x99,0x92,0x82,0xf8,0x80,0x90,
0x88,0x83,0xc6,0xa1,0x86,0x8e,}

	(i)		(3-i)	4*i
数字	数位索引	段码	显示缓冲区索引	右移二进制数位
5	3	0x92	0	3*4
6	2	0x82	1	2*4
7	1	0xf8	2	1*4
8	0	0x80	3	0*4

程序流程图(子)-更新显示缓冲区



segcode[8]={0xc0,0xf9,0xa4,0xb0}

Sw=0x5678

segtable[16]={0xc0,0xf9,0xa4,0xb0
,0x99,0x92,0x82,0xf8,0x80,0x90,
0x88,0x83,0xc6,0xa1,0x86,0x8e,}

(i) (3-i) 4*i

数字	右移二进制数位	数位索引	显示缓冲区索引
5	3*4	3	0
6	2*4	2	1
7	1*4	1	2
8	0*4	0	3

数码管实时显示开关值的程序段

```
char segtable[16]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,\
        0x80,0x90,0x88,0x83,0xc6,0xa1,0x86,0x8e,};//段码表
char segcode[4]={0xc0,0xc0,0xc0,0xc0};//显示缓冲区
short poscode[4]={0xf7,0xfb,0xfd,0xfe};//位码表
Xil_Out8(0x40010004,0x0);
Xil_Out8(0x4001000c,0x0);
Xil_Out16(0x40000004,0xffff);
Xil_Out16(0x4000000c,0x0);
```

```
while(1)
  {
     for(int i=0;i<4;i++)
      {
     short Key=Xil_In16(0x40000000);//
     Xil_Out16(0x40000008, Key);//
     for(int digit_index=0;digit_index<4;digit_index++)//
     segcode[3- digit_index]=segtable[(Key >>(4*digit_index))&0xf];
     Xil_Out8(0x40010000,segcode[i]);//
     Xil_Out8(0x40010008,poscode[i]); //
     for(int j=0;j<10000;j++);//延时控制
     }
     }
```

小结

- GPIO特征
  - 输入缓冲
  - 输出锁存
  - IO方向可编程配置，且各位独立控制
- GPIO编程
  - 写控制寄存器控制输入、输出方向
  - 读写数据寄存器实现输入、输出

下一讲：外设控制器EPC，自学，大字节序实例