

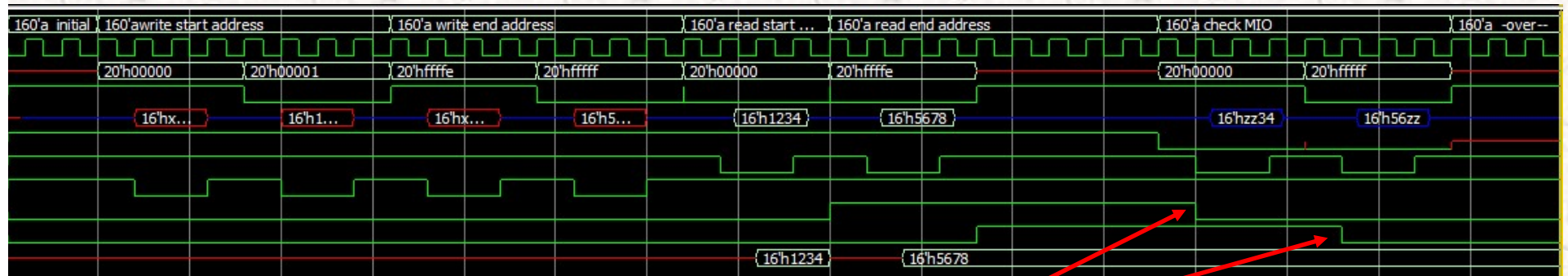
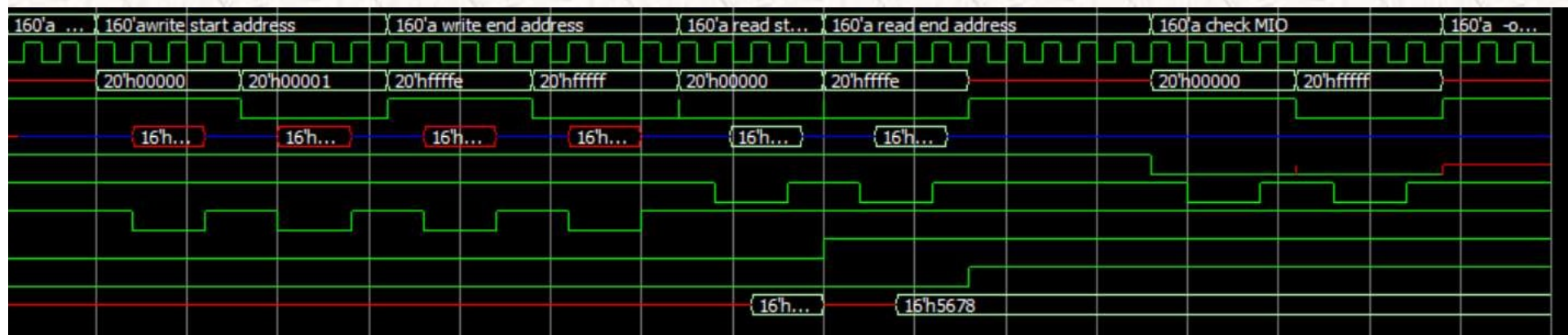
电路练习-存储体接入与总线波形

```
sram512KB sram512KB_1(  
    .D7_0( D15_0[7:0] ),  
    .A18_0( A19_0[19:1] ),  
    .CS_n( !MIO | A19_0[0] ),  
    .WR_n( WR_n ),  
    .RD_n( RD_n )  
);  
sram512KB sram512KB_2(  
    .D7_0( D15_0[15:8] ),  
    .A18_0( A19_0[19:1] ),  
    .CS_n( !MIO | BHE_n ),  
    .WR_n( WR_n ),  
    .RD_n( RD_n )  
);
```

// 偶地址存储体

// 奇地址存储体

仿真波形



检测到访问IO时也选中存储器，给出错误指示信号

第五章练习

- 设计目标：使用**8K*4bit**的存储器实现**48KB**的存储体，从物理地址**A0000H**开始
- 使用位扩展组成**8K*8bit**的存储器模块
- 每个存储体采用**3组 8K*8bit**的存储器模块进行字扩展，得到**24K*8bit**
- 加入**A0**和**/BHE**信号分别得到奇偶存储体各存储器模块的片选信号

偶地址存储体

```
sram8K4b sram8K4b_1(  
    .D3_0(D15_0[3:0]),  
    .A12_0(A19_0[13:1]),  
    .CS_n( CS0_n | A19_0[0]),  
    .WR_n(WR_n),  
    .RD_n(RD_n)  
);
```

```
sram8K4b sram8K4b_2(  
    .D3_0(D15_0[7:4]),  
    .A12_0(A19_0[13:1]),  
    .CS_n( CS0_n | A19_0[0]),  
    .WR_n(WR_n),  
    .RD_n(RD_n)  
);
```

```
sram8K4b sram8K4b_3(  
    .D3_0(D15_0[3:0]),  
    .A12_0(A19_0[13:1]),  
    .CS_n( CS1_n | A19_0[0]),  
    .WR_n(WR_n),  
    .RD_n(RD_n)  
);
```

```
sram8K4b sram8K4b_4(  
    .D3_0(D15_0[7:4]),  
    .A12_0(A19_0[13:1]),  
    .CS_n( CS1_n | A19_0[0]),  
    .WR_n(WR_n),  
    .RD_n(RD_n)  
);
```

.....略

奇地址存储体

```
sram8K4b sram8K4b_1(  
    .D3_0(D15_0[11:8]),  
    .A12_0(A19_0[13:1]),  
    .CS_n( CS0_n | BHE_n ),  
    .WR_n(WR_n),  
    .RD_n(RD_n)  
);
```

```
sram8K4b sram8K4b_2(  
    .D3_0(D15_0[15:12]),  
    .A12_0(A19_0[13:1]),  
    .CS_n( CS0_n | BHE_n ),  
    .WR_n(WR_n),  
    .RD_n(RD_n)  
);
```

```
sram8K4b sram8K4b_3(  
    .D3_0(D15_0[11:8]),  
    .A12_0(A19_0[13:1]),  
    .CS_n( CS1_n | BHE_n ),  
    .WR_n(WR_n),  
    .RD_n(RD_n)  
);
```

```
sram8K4b sram8K4b_4(  
    .D3_0(D15_0[15:12]),  
    .A12_0(A19_0[13:1]),  
    .CS_n( CS1_n | BHE_n ),  
    .WR_n(WR_n),  
    .RD_n(RD_n)  
);
```

.....略

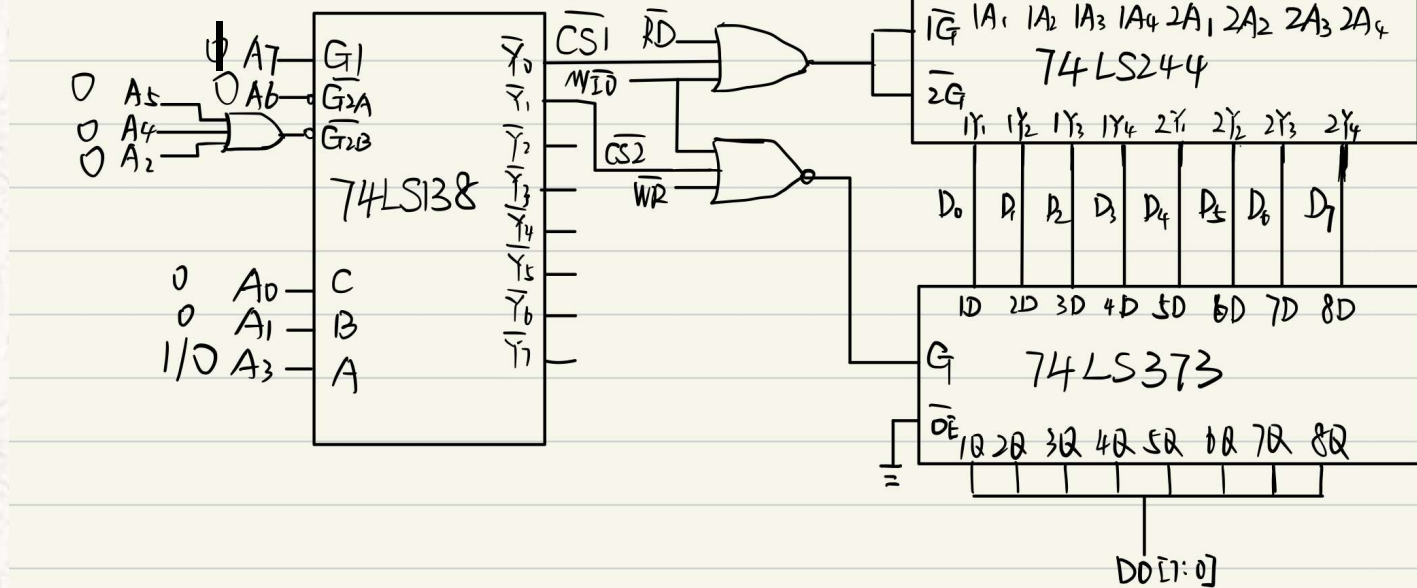
译码逻辑

```
Is138 Is138_1(  
    .G1(MIO & A19_0[19]),  
    .G2A_n(A19_0[18]),  
    .G2B_n(!A19_0[17]),  
    .C(A19_0[16]),  
    .B(A19_0[15]),  
    .A(A19_0[14]),  
    .Y0_n(CS0_n),  
    .Y1_n(CS1_n),  
    .Y2_n(CS2_n),  
    .Y3_n(),  
    .Y4_n(),  
    .Y5_n(),  
    .Y6_n(),  
    .Y7_n()  
);
```


第六章设计

- 设计温度输入接口和控制输出接口，端口地址分别为**80H**和**88H**。

80H: 1000 0000 B
88H: 1000 1000 B



错误示例

```
ls244 ls244_1(  
    .p1A1_4(tempreature[7:4]),  
    .p2A1_4(tempreature[3:0]),  
    .p1G_n(CS1_n | RD_n | MIO),  
    .p2G_n(CS1_n | RD_n | MIO),  
    .p1Y1_4(D15_0[7:4]),  
    .p2Y1_4(D15_0[3:0])  
);
```

```
ls373 ls373_1(  
    .DI(D15_0[7:0]),  
    .OE_n(CS2_n | WR_n | MIO),  
    .G(1'b1),  
    .DO(DO[7:0])  
);
```

// IO选通逻辑

```
ls138 ls138_1(  
    .G1(A19_0[7]),  
    .G2A_n(A19_0[6]),  
    .G2B_n(A19_0[5] | A19_0[4] | A19_0[2]),  
    .C(A19_0[1]),  
    .B(A19_0[0]),  
    .A(A19_0[3]),  
    .Y0_n(CS1_n),  
    .Y1_n(CS2_n),  
    .Y2_n(),  
    .Y3_n(),  
    .Y4_n(),  
    .Y5_n(),  
    .Y6_n(),  
    .Y7_n()  
);
```


第七章练习

- 8253地址分配
- 使用触发器实现2分频电路
- 使用8253级联实现分频和计时

// 8253地址31H, 33H, 35H, 37H

ls138 ls138_1(

.G1(A19_0[5]), // 声明D触发器

.G2A_n(A19_0[7]), Dflop Dflop_1(

.G2B_n(A19_0[6]), .clk(clk_4M),

.C(MIO), .reset_n(~reset),

.B(A19_0[4]), .D(tt),

.A(A19_0[3]), .Q(CLK_2M),

.Y0_n(), .Q_n(tt)

.Y1_n(),);

.Y2_n(CS_8253),

.Y3_n(),

.Y4_n(),

.Y5_n(),

.Y6_n(),

.Y7_n()

);

intel8253 intel8253(

.D7_0(D15_0[15:8]),

.WR_n(WR_n),

.RD_n(RD_n),

.CS_n(CS_8253),

.A0(A19_0[1]),

.A1(A19_0[2]),

.CLK0(CLK_2M),

.GATE0(1'b1),

.OUT0(CLK_1K),

.CLK1(CLK_1K),

.GATE1(1'b1),

.OUT1(CLK_1),

.CLK2(CLK_1),

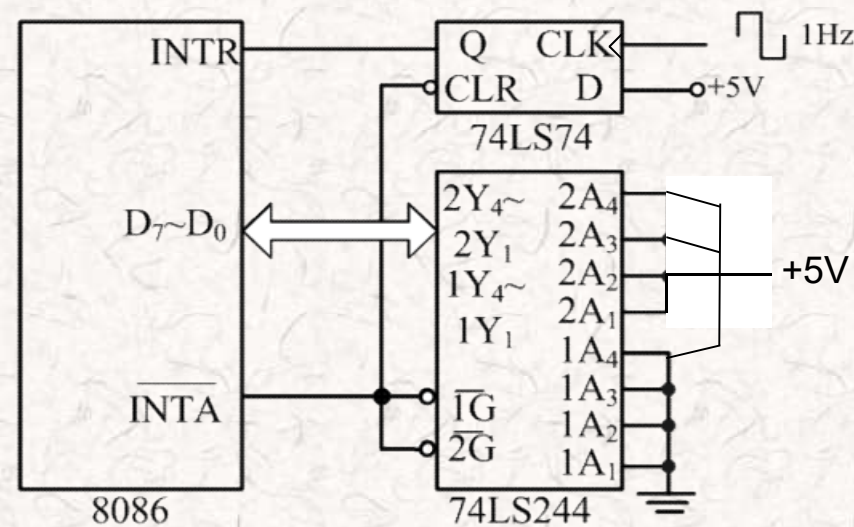
.GATE2(1'b1),

.OUT2(t_3s)

);

第八章练习

- 发起中断请求，在中断应答周期发送中断类型号**30H**，并清除中断请求
- 为该中断设置中断向量**1000H：2000H**




```

ls244 ls244_1(
    .p1A1_4( 4'b0011 ),
    .p2A1_4( 4'b0 ),
    .p1G_n( INTA_n),
    .p2G_n( INTA_n),
    .p1Y1_4(D15_0[7:4] ),
    .p2Y1_4(D15_0[3:0] )
);

```

// 声明D触发器

```

Dflop Dfolp_1(
    .clk(INT_request),
    .reset_n(INTA_n),
    .D(1'b1),
    .Q(INTR),
    .Q_n()
);

```

//-----以下需要自行填写，模拟编程，为
 类型号30H的中断设置中断向量1000H:
 2000H-----

```

label=" set int vector ";
writeMEM( 20'h30*4, 16'h2000 ,1'b1 );
writeMEM( 20'h30*4+2, 16'h1000 ,1'b1 );

```

//-----自行填写模拟编程结束·-----
