

数字逻辑与处理器基础实验 实验报告

实验二：反应速度测试仪的设计

无 13 管思源 2021012702

实验目的

- 掌握简单的组合和时序电路综合设计方法
- 掌握数码管动态显示方法
- 掌握 Verilog 层次化建模方法

设计方案

分两步实现指导书要求的功能：先将任意 4 位数字显示在数码管上，再设计顶层模块实现任务流程和计时功能。

任意 4 位数字的显示

设计思路

根据课上提示，我们以 1ms 的间隔循环遍历 4 个数码管，依次设置 `sel[3:0]`、`seg[6:0]` 和 `dp`，即可实现动态显示。

我们创建 `display_number` 模块，输入为

- `clk`：时钟信号（使用系统时钟）
- `num[13:0]`：待显示的 4 位数字（0至9999）
- `dpt`：小数点位置（最左侧为 0，最右侧为 3）

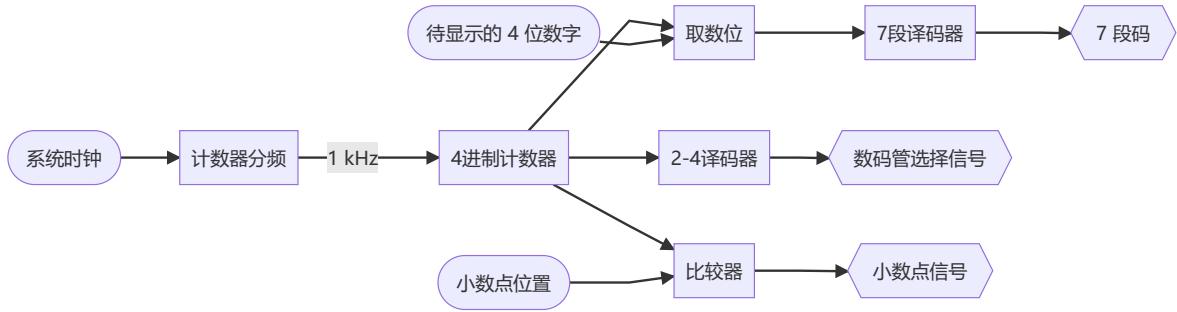
输出为

- `sel[3:0]`：数码管选择信号
- `seg[6:0]`：7 段码
- `dp`：小数点信号

模块内需要实现以下功能

- 1ms 为周期的时钟，即 100000 倍计数分频器（系统时钟为 100MHz）
- 4 个数码管的循环遍历，即一个 4 进制计数器（包括数码管选择信号和小数点信号的设置）
- 4 位数字到每位数字7段码的转换，可以使用取模操作和实验一提供的 `BCD7` 模块

原理框图



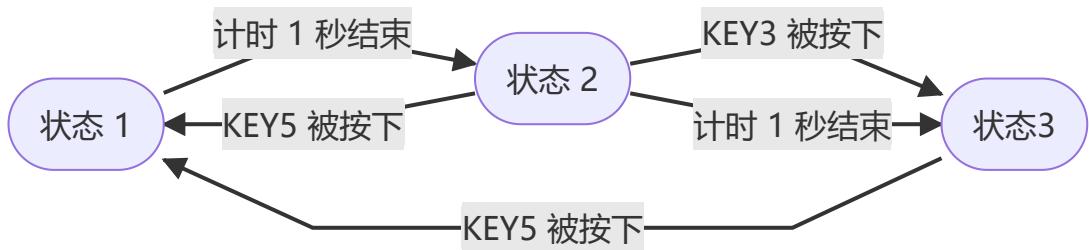
顶层模块的设计

设计思路

指导书描述了系统的三个状态：

1. 系统启动后或按下 KEY5 复位后计时 1 秒
2. LED 亮，等待 1 秒 KEY3 被按下
3. LED 灭，显示 LED 亮到 KEY3 被按下的时间（反应时间）

据此我们设计有限状态机，以下是状态转移图：



有限状态机（即顶层模块）的输入为

- `clock`：时钟信号（使用系统时钟）
- `key3`：KEY3 按键信号
- `key5`：KEY5 按键信号

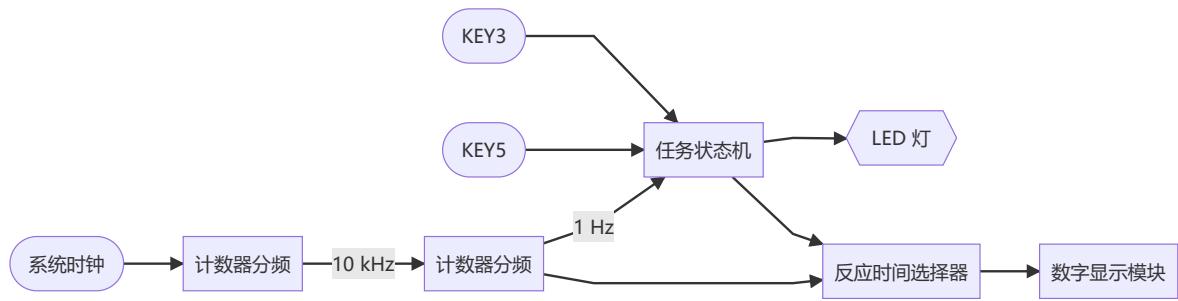
输出为

- `led[0]`：LED 灯信号
- 4 位数码管的显示信号

对于顶层模块，我们需要实现以下功能

- 0.1ms 为周期的时钟，用于测量反应时间，可以使用系统时钟进行 10000 倍计数分频
- 1s 的计时器供状态转移使用，可以在 0.1ms 时钟基础上进行 1000 倍计数分频
- 任务状态机，实现状态转移（包括按键检测和 LED 显示）
- 四位反应时间的显示，可以使用前面设计的 `display_number` 模块

原理框图



注意事项

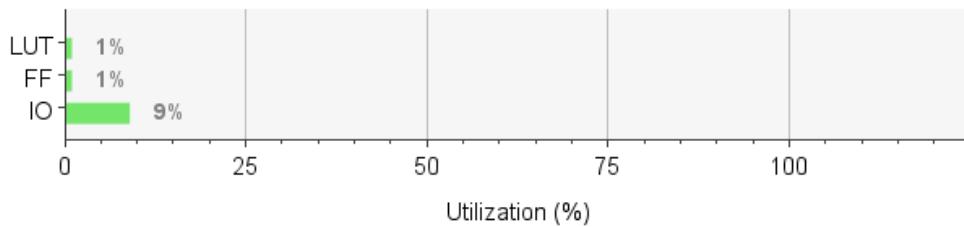
- 计数器不使用取模操作（消耗硬件资源较多），而是判断到达了计数值后重置
- 注意数码管的左右顺序，sel 低位对应左侧数码管，对应要取数字的千位
- 在状态 3 中，我们需要保留反应时间，因此需要来自状态机的控制信号使计数器不再累加

综合结果与分析

硬件资源使用情况

Summary

Resource	Utilization	Available	Utilization %
LUT	193	20800	0.93
FF	59	41600	0.14
IO	23	250	9.20



Hierarchy							
Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	Bonded IOB (250)	BUFGCTRL (32)	
top	193	59	73	193	23	1	
d (display_number)	174	28	58	174	0	0	

如图所示，使用了 193 个 LUT 和 59 个寄存器。其中，数字显示模块使用了 174 个 LUT 和 28 个寄存器，顶层模块自身使用了 19 个 LUT 和 31 个寄存器。

数字显示模块使用的 LUT 占绝大部分，这是因为数字取位和 7 段译码的逻辑较为复杂；而顶层模块自身使用的寄存器更多，这是由于两个计数分频器用到的位数较大。

静态时序分析结果

一开始，我先进行了综合，然后在综合设计中添加时钟进行静态时序分析。以下是我得到的结果：

Design Timing Summary									
Setup			Hold			Pulse Width			
Worst Negative Slack (WNS):	-2.515 ns		Worst Hold Slack (WHS):	0.232 ns		Worst Pulse Width Slack (WPWS):	4.500 ns		
Total Negative Slack (TNS):	-6.956 ns		Total Hold Slack (THS):	0.000 ns		Total Pulse Width Negative Slack (TPWS):	0.000 ns		
Number of Failing Endpoints:	4		Number of Failing Endpoints:	0		Number of Failing Endpoints:	0		
Total Number of Endpoints:	118		Total Number of Endpoints:	118		Total Number of Endpoints:	60		
Timing constraints are not met.									
Intra-Clock Paths - clock - Setup									
Name	Slack ^ 1	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement
↳ Path 1	-2.515	16	151	FSM_onehot_state_reg[0]/C	d/digit_reg[3]/D	12.520	3.677	8.843	10.000
↳ Path 2	-2.412	16	151	FSM_onehot_state_reg[0]/C	d/digit_reg[2]/D	12.375	3.677	8.698	10.000
↳ Path 3	-2.027	16	151	FSM_onehot_state_reg[0]/C	d/digit_reg[1]/D	12.089	3.677	8.412	10.000
↳ Path 4	-0.002	13	151	FSM_onehot_state_reg[0]/C	d/digit_reg[0]/D	10.017	3.362	6.655	10.000
↳ Path 5	5.108	4	14	count_reg[4]/C	num_reg[10]/R	4.472	0.853	3.619	10.000
↳ Path 6	5.108	4	14	count_reg[4]/C	num_reg[11]/R	4.472	0.853	3.619	10.000
↳ Path 7	5.108	4	14	count_reg[4]/C	num_reg[12]/R	4.472	0.853	3.619	10.000
↳ Path 8	5.108	4	14	count_reg[4]/C	num_reg[9]/R	4.472	0.853	3.619	10.000
↳ Path 9	5.119	4	14	count_reg[4]/C	num_reg[1]/R	4.461	0.853	3.608	10.000
↳ Path 10	5.119	4	14	count_reg[4]/C	num_reg[2]/R	4.461	0.853	3.608	10.000

可以看到建立时间不满足时钟要求，有四条路径的总延时超过了 10 ns。我们进一步查看延时最大的一条路径，如下图所示：

Path 1 - timing_1						
Data Path						
Delay Type	Incr (ns)	Path ...	Location	Cell Pin	Cell	Nelist Resources
<u>FDRE (Prop_fdre_C_Q)</u>	(r) 0.433	4.997	Site: SLICE_X56Y51	Q	FSM_onehot_state_reg[0] (FDRE)	FSM_onehot_state_reg[0]/Q d/digit_reg[1]_0
net (fo=15, routed)	1.391	6.389			i__1_carry_1_i_11 (LUT4)	d/i1_carry_1_i_11/2 d/i1_carry_1_i_11/O d/i1_carry_1_i_11_n_0
<u>LUT4 (Prop_lut4_I2_O)</u>	(r) 0.105	6.494	Site: SLICE_X63Y48	I2	i__1_carry_1_i_11 (LUT4)	d/i1_carry_1_i_11_n_0
net (fo=2, routed)	0.940	7.433			i__1_carry_1_i_4 (LUT6)	d/i1_carry_1_i_4/I5 d/i1_carry_1_i_4/O
<u>LUT6 (Prop_lut6_I5_O)</u>	(r) 0.105	7.538	Site: SLICE_X65Y48	I5	i__1_carry_1_i_4 (LUT6)	d/i1_carry_1_i_4/O d/i1_carry_1_i_4/I5
net (fo=2, routed)	0.777	8.315			i__1_carry_1_i_8 (LUT6)	d/i1_carry_1_i_8/I0 d/i1_carry_1_i_8/O
<u>LUT6 (Prop_lut6_I0_O)</u>	(r) 0.105	8.420	Site: SLICE_X62Y50	I0	i__1_carry_1_i_8 (LUT6)	d/i1_carry_1_i_8/I0 d/i1_carry_1_i_8/O
net (fo=1, routed)	0.000	8.420			i1_carry_1_i_8_n_0	d/i1_carry_1_i_8_n_0
			Site: SLICE_X62Y50	S[0]	digit1_inferred_0/i__1_carry_1 (CARRY4)	d/digit1_inferred_0/i__1_carry_1/S[0]
<u>CARRY4 (Prop_c_4_S[0_CO[3])</u>	(r) 0.440	8.860	Site: SLICE_X62Y50	CO[3]	digit1_inferred_0/i__1_carry_1 (CARRY4)	d/digit1_inferred_0/i__1_carry_1/CO[3]
net (fo=1, routed)	0.000	8.860			digit1_inferred_0/i__1_carry_1_n_0	d/digit1_inferred_0/i__1_carry_1_n_0
<u>CARRY4 (Prop_carry4_CI_O[2])</u>	(r) 0.200	9.060	Site: SLICE_X62Y51	CI	digit1_inferred_0/i__1_carry_2 (CARRY4)	d/digit1_inferred_0/i__1_carry_2/CI
net (fo=15, routed)	0.973	10.033			digit1_inferred_0/i__1_carry_2 (CARRY4)	d/digit1_inferred_0/i__1_carry_2/O[2]
<u>LUT3 (Prop_lut3_I2_O)</u>	(r) 0.253	10.286	Site: SLICE_X61Y52	I2	i__50_carry_i_1 (LUT3)	d/i50_carry_i_1/I2 d/i50_carry_i_1/O
net (fo=1, routed)	0.414	10.700			i__50_carry_i_1 (LUT3)	d/i50_carry_i_1_n_0
<u>CARRY4 (Prop_c_4_DI[3_CO[3])</u>	(r) 0.318	11.018	Site: SLICE_X63Y52	DI[3]	digit1_inferred_0/i__50_carry (CARRY4)	d/digit1_inferred_0/i__50_carry/DI[3]
net (fo=1, routed)	0.000	11.018			digit1_inferred_0/i__50_carry (CARRY4)	d/digit1_inferred_0/i__50_carry/CO[3]
<u>CARRY4 (Prop_carry4_CI_O[1])</u>	(r) 0.265	11.283	Site: SLICE_X63Y53	CI	digit1_inferred_0/i__50_carry_0 (CARRY4)	d/digit1_inferred_0/i__50_carry_0/CI
net (fo=3, routed)	0.898	12.181			digit1_inferred_0/i__50_carry_0 (CARRY4)	d/digit1_inferred_0/i__50_carry_0/O[1]
<u>LUT5 (Prop_lut5_I2_O)</u>	(r) 0.250	12.431	Site: SLICE_X63Y50	O	i__74_carry_0_i_5 (LUT5)	d/i74_carry_0_i_5/I2 d/i74_carry_0_i_5/O
net (fo=1, routed)	0.000	12.431			i__74_carry_0_i_5 (LUT5)	d/i74_carry_0_i_5_n_0
<u>CARRY4 (Prop_c_4_S[3_CO[3])</u>	(r) 0.332	12.763	Site: SLICE_X63Y50	S[3]	digit1_inferred_0/i__74_carry_0 (CARRY4)	d/digit1_inferred_0/i__74_carry_0/S[3]
net (fo=1, routed)	0.000	12.763			digit1_inferred_0/i__74_carry_0 (CARRY4)	d/digit1_inferred_0/i__74_carry_0/CO[3]
<u>CARRY4 (Prop_carry4_CI_O[1])</u>	(r) 0.190	12.953	Site: SLICE_X63Y51	CI	digit1_inferred_0/i__74_carry_1 (CARRY4)	d/digit1_inferred_0/i__74_carry_1/CI
net (fo=5, routed)	0.546	13.498			digit1_inferred_0/i__74_carry_1 (CARRY4)	d/digit1_inferred_0/i__74_carry_1/O[2]
<u>LUT4 (Prop_lut4_I3_O)</u>	(r) 0.261	13.759	Site: SLICE_X61Y52	I3	digit[0]_i_3 (LUT4)	d/digit[0]_i_3/I3 d/digit[0]_i_3/O
net (fo=4, routed)	0.776	14.535			digit[0]_i_3 (LUT4)	d/digit[0]_i_3_n_0
<u>LUT6 (Prop_lut6_I2_O)</u>	(r) 0.105	14.640	Site: SLICE_X61Y53	I2	digit[3]_i_35 (LUT6)	d/digit[3]_i_35/I2 d/digit[3]_i_35/O
net (fo=3, routed)	0.485	15.125			digit[3]_i_35 (LUT6)	d/digit[3]_i_35_n_0
<u>LUT6 (Prop_lut6_I5_O)</u>	(r) 0.105	15.230	Site: SLICE_X60Y54	I5	digit[3]_i_17 (LUT6)	d/digit[3]_i_17/I5 d/digit[3]_i_17/O
net (fo=3, routed)	0.669	15.899			digit[3]_i_17 (LUT6)	d/digit[3]_i_17_n_0
<u>LUT6 (Prop_lut6_I0_O)</u>	(r) 0.105	16.004	Site: SLICE_X60Y55	I0	digit[3]_i_17 (LUT6)	d/digit[3]_i_17/I0 d/digit[3]_i_17/O
net (fo=1, routed)	0.975	16.979			digit[3]_i_17 (LUT6)	d/digit[3]_i_17_n_0
<u>LUT6 (Prop_lut6_I5_O)</u>	(r) 0.105	17.084	Site: SLICE_X64Y49	I5	digit[3]_i_1 (LUT6)	d/digit[3]_i_1/I5 d/digit[3]_i_1/O
net (fo=1, routed)	0.000	17.084			digit[3]_i_1 (LUT6)	d/digit[3]_i_1_n_0
<u>FDRE</u>			Site: SLICE_X64Y49	D	digit_reg[3] (FDRE)	d/digit_reg[3]/D
<u>Arrival Time</u>		17.084				

这条路径从 state 寄存器出发，经过多层 LUT，最后到达 digit 寄存器。可以看出，路径中大部分延时和 LUT 都用于计算 digit，这与耗时的数字取位操作相符。

在我尝试优化逻辑达到时钟要求的过程中，我（为了节省时间）在硬件约束文件中添加了时钟约束，并无意间用没有任何改动的代码进行综合。

结果是出乎意料的（如下图所示）：

Design Timing Summary

Setup	Hold	Pulse Width	
Worst Negative Slack (WNS):	0.288 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	118	Total Number of Endpoints:	60

All user specified timing constraints are met.

Name	Slack ^ 1	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement
Path 1	0.288	14	151	FSM_onehot_state_reg[0]/C	d/digit_reg[3]/D	9.642	3.467	6.175	10.0
Path 2	0.302	14	151	FSM_onehot_state_reg[0]/C	d/digit_reg[1]/D	9.589	3.467	6.122	10.0
Path 3	0.526	14	151	FSM_onehot_state_reg[0]/C	d/digit_reg[2]/D	9.364	3.467	5.897	10.0
Path 4	1.959	11	151	FSM_onehot_state_reg[0]/C	d/digit_reg[0]/D	7.971	3.574	4.397	10.0
Path 5	5.021	3	24	num_reg[9]/C	num_reg[13]/R	4.564	0.875	3.689	10.0
Path 6	5.283	3	24	num_reg[9]/C	num_reg[10]/R	4.329	0.875	3.454	10.0
Path 7	5.283	3	24	num_reg[9]/C	num_reg[11]/R	4.329	0.875	3.454	10.0
Path 8	5.283	3	24	num_reg[9]/C	num_reg[12]/R	4.329	0.875	3.454	10.0
Path 9	5.283	3	24	num_reg[9]/C	num_reg[9]/R	4.329	0.875	3.454	10.0
Path 10	5.416	3	24	num_reg[9]/C	num_reg[5]/R	4.169	0.875	3.294	10.0

综合后竟满足了时钟要求！

对比在硬件约束文件中添加时钟约束前后、关键路径的线延时和逻辑延时的差异，我们发现在添加时钟约束后逻辑延时并没有明显变化，而线延时却从 9 ns 左右下降到了 6 ns 左右。因此，我认为这是因为 Vivado 在综合时会根据时钟约束来优化布局布线，来尽可能满足时钟要求。

老师在课上提到可以使用 4 个 10 进制计数器串联来避免取位操作，这的确是很好的设计/优化思路。但是，一方面出于模块泛用性和调试便利的考虑，我在设计方案时决定将 4 位数字显示功能模块化，希望能接受任意的 4 位整数；另一方面因为设计已经满足了时钟要求，所以没有采用这种优化办法。

关键代码及文件清单

代码文件（按模块例化结构）

- `top.v`：顶层模块，包含任务状态机和毫秒计时器
 - `display_number.v`：数字显示模块，将 4 位数字显示在数码管上
 - `BCD7.v`：BCD 码转 7 段码模块，同实验一提供的代码

其他文件

- `constraint.xdc`：硬件约束文件
- `top.bit`：生成的比特流文件，可直接烧写到开发板
- `img/*.png`：综合结果截图