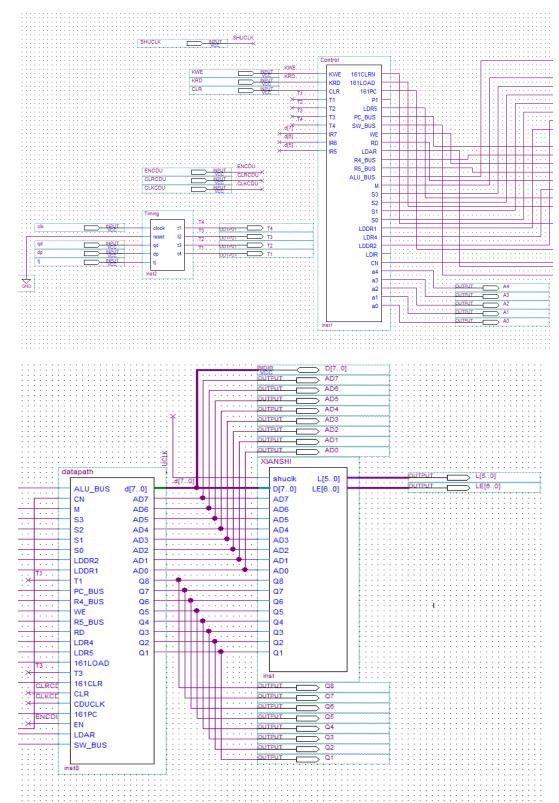
一、模型机1

1. 实验目标任务:

将数据通路, 时序电路和微程序控制器整合成为一个整体, 即模型机, 完成结构上的整合和时许上的整合, 即将各个模块整合, 让各控制信号的触发与时序节拍信号相结合, 在制定的节拍完成制定的操作。

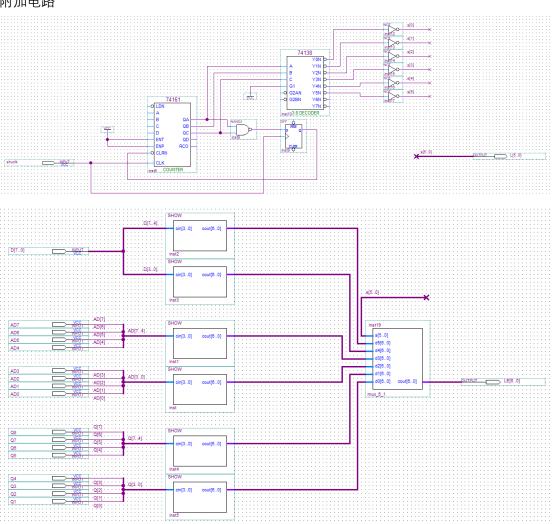
2. 顶层电路



图一为微程序控制器和时序电路部分,左下角为时序电路,产生各个节拍,右侧即微程序控制器,即之前微控模块所做,发出控制信号,产生下地址。最左侧的输入部分上方为显示模块的时钟,下侧的 3 个输入为之后数据通路的写入模块所需。图二为数据通路与显示模块部分,左侧为数据通路,右侧为显示模块,由于无电路板,于是将数据通路导出的总线数据,AR 存放的地址,以及 R5 保存的数据 Q 单独引出,便于调试。数据通路的运算器部分加入了 2 块 4 位 2 进制计数器级联,作为写入模块。

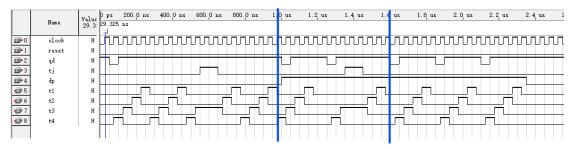
在调试过程中,功能仿真正确,但时序仿真时,发现 ALU 模块无法及时的将总线上的数据给入 DR1 和 DR2,因此,将 DR1 和 DR2 的节拍设置为 T1,保证能够及时写入数据,时序仿真正确。

3. 附加电路



数码管显示部分包括两个部分,上图用于产生位选信号,下图用于产生段选信号,上图通过一个模 5 计数器来在六个周期产生六个不同的值,通过 3-8 译码器产生 6 个不同的译码,作为位选信号,控制 6 个数码管的显示;下图通过将输入的信号译出 0-F16 个数字,产生段选信号,接入 6 选一多路复用器,选择端接入上图位选信号,最终输出当前位选信号的显示。

4. 时序电路仿真图+功能说明



第一部分连续运行到停机,qd 从 1 到 0 开始运行,当 tj 等于 1 时实现停机 第二部分连续到单步到停机,qd 从 1 到 0 开始运行,当 dp 等于 1 时为单步运行 状态,之后 tj 等于 1 实现停机

第三部分单步调试, dp 等于 1, 每当 qd 从 1 到 0 时, 仅发出一个周期的节拍信号, 实现单步运行。

5. Mif 文件结果

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	20	0D	C0	0E	40	10	60	10
08	E0	0F	80	A0	18	55	8A	F0
10		00		00	00	00	_	00
	00		00				00	
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00
38	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00
48	00	00	00	00	00	00	00	00
50	00	00	00	00	00	00	00	00
58	00	00	00	00	00	00	00	00
60	00	00	00	00	00	00	00	00
68	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00
78	00	00	00	00	00	00	00	00
80	00	00	00	00	00	00	00	00
88	00	00	00	00	00	00	00	00
90	00	00	00	00	00	00	00	00
98	00	00	00	00	00	00	00	00
a0	00	00	00	00	00	00	00	00
a8	00	00	00	00	00	00	00	00
b0	00	00	00	00	00	00	00	00
b8	00	00	00	00	00	00	00	00
c0	00	00	00	00	00	00	00	00
c8	00	00	00	00	00	00	00	00
d0	00	00	00	00	00	00	00	00
d8	00	00	00	00	00	00	00	00
e0	00	00	00	00	00	00	00	00
e8	00	00	00	00	00	00	00	00
f0	00	00	00	00	00	00	00	00
				1		1		

运行后会在 10H 写入结果:D0

二、模型机 2

1. LDA:10H STA:40H OUT:50H COM:02H

JMP:12H

ADD:42H

AND:52H

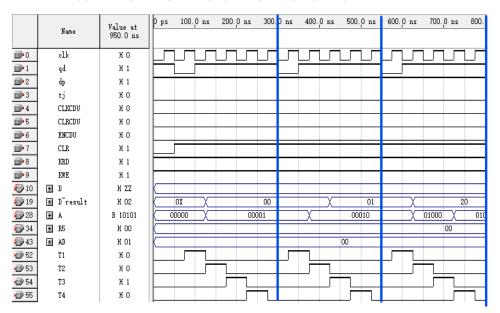
2. 截图:

mif 文件

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	20	19	C0	1B	40	0C	60	0C
08	Α0	10	00	00	00	00	00	00
10	20	1A	80	C0	18	E0	0C	80
18	01	02	03	04	00	00	00	00
20	00	00	00	00	00	00	00	00

仅截取所用部分, 其余单元均为 00

所进行的指令: LDA ADD STA->0CH OUT->0CH JMP->10H LDA COM ADD AND COM 说明:数码管显示采用六位显示,依次为 BUS,AR,R5 的数据。



为取指令部分两个周期, 分别完成 PC->AR,PC+1 和 RAM->IR,由于此后均 有该步骤,因此之后不予 赘述

使用的 6 位数码管依次显示: 000000 010000

200000

	37	Value at	700.0 ns 800.0 ns	900.0 ns	1.0 us	1.1 _, us	1.2 us 1.	3 us 1.	4 us 1.5 us
	Name	200.0 ns							
₽ 0	clk	но							
i 1	qd	Н 1						╙┸	
■ 2	dp	Н 1							
■ 3	tj	н о							
1 4	CLKCDU	н о							
₱ 5	CLRCDU	н о							
№ 6	ENCDU	н о							
→ 7	CLR	Н 1							
№ 8	KRD	Н 1							
₽ 9	KWE	Н 1							
10	₩ D	H ZZ				ZZ			
19	₩ D~result	н оо	20	X 01 X	02		19		02
⊚ 28	 A	B 00001	0100C 01001		10101		10110	\square X	00001
	⊞ R5	Н 00			()			02
a 43 €	# AD	н оо	- 00	\square X	0		Х	19	9
⊚ 52	T1	н о							
⊚ 53	T2	н о							
₫ 54	T3	Н 1			\neg				
⊚ 55	T4	н о							
						' '		•	•

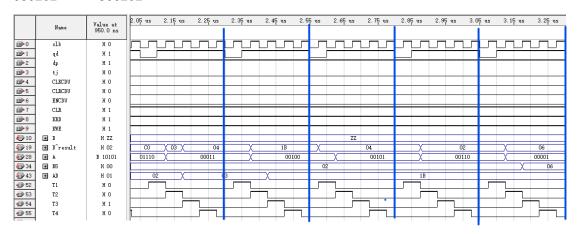
完成 LDA 指令,第一部分将地址打入 AR,第二部分从 RAM 里读出存放 B 元素的地址给 AR,第三部分取出 B 元素给入 R5

使用的6位数码管依次显示: 020100 191900 021902

重复取指令过程

数码管显示为:

030202 C00202



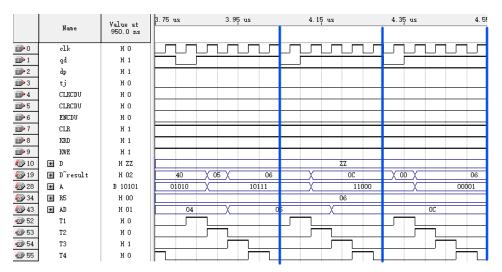
完成 ADD 指令, 实现 B 加 D

各个阶段数码管显示为:

040203 1B021B 04021B 02021B 06061B

重复取指令过程数码管显示为:

050604 400604



完成 STA 指令,将 R5 的数据送入 0C 地址单元

数码管显示为:

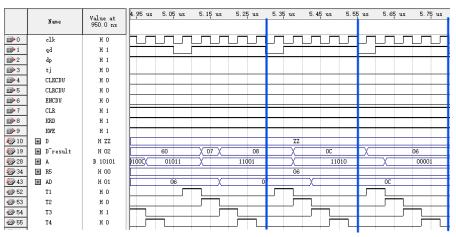
060605

0C060C 06060C

重复取指令过程

数码管显示为:

070606 600606



完成 OUT 指令, 将 0C 地 址单元的数据取出输出到 总线

数码管显示为:

080607

0C060C

06060C

重复取指令过程

数码管显示为:

090608 A00608

	Name	Value at 950.0 ns	6.25 us 6.35 us 6.45 us 6.55 us 6.65 us 6.75 us
→ 0	clk	но	
■ 1	qd	Н 1	
₽ 2	$^{\mathrm{dp}}$	Н 1	
■ 3	tj	но	
□ 4	CLKCDU	но	
1 5	CLRCDU	но	
№ 6	ENCDU	но	
→ 7	CLR	Н 1	
№ 8	KRD	Н 1	
₽ 9	KWE	Н 1	
10	∓ D	H ZZ	ZZ
19	⊞ D~result	H 02	AO X 09 X OA X 10
⊚ 28	∓ Å	B 10101	01101 X 11100 X 00001
⊚ 34	⊞ R5	н оо	06
⊚ 43	∓ AD	H 01	08 09
⊚ 52	T1	но	
⊚ 53	T2	но	
€ 54	T3	Н 1	
⊕ 55	T4	но	

完成 JMP 指令, 跳转到下 一段指令 10H 处

数码管显示为:

0A0609

100609

重复取指令过程

数码管显示为:

110610 200610

	Name	Value at 950.0 ns	7.3 us	7.4 _, us	7.5 _, us	7.6 us	7. 7 _, us	7.8	us 7.5	jus i	3.0 _, us
-	11		II		. — –					. — .	
<u></u> 0	clk	но		\underline{L}	\sqcup \sqcup	$\vdash \vdash \vdash$		<u> </u>	ᅵ ᆜ	\sqcup \sqcup	<u> </u>
<u>i</u> 1	qd	Н 1									
₽ 2	dp	Н 1									
1 3 ⋅ 3	tj	но									
→ 4	CLKCDV	но									
₽ 5	CLRCDV	но									
₽ 6	ENCDU	но									
7	CLR	Н 1									
№ 8	KRD	Н 1									
₽ 9	KWE	Н 1									
10	₩ D	H ZZ					ZZ				
a 19	∄ D~result	н 02	20	X 11 X	12	\Box X	1 A		X	- 03	3
₽ 28	± A	B 10101	01001	\square X	10101	\square X	1	0110			00001
⊚ 34	 R 5	н оо				06					03
⊕ 43	⊞ AD	H 01	10	X		1				1 A	
⊚ 52	T1	но							JTL		
⊚ 53	T2	но									
⊚ 54	T3	Н 1					шП				
⊚ 55	T4	но									\lnot

完成第二个 LDA 指令,将 C元素输入到R5

数码管显示为:

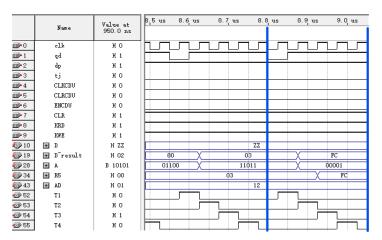
120611

1A061A

03031A

重复取指令过程 数码管显示为:

130312 800312



完成 COM 指令, R5 的数据 取反送入 R5, 实现 NOT C

数码管显示为:

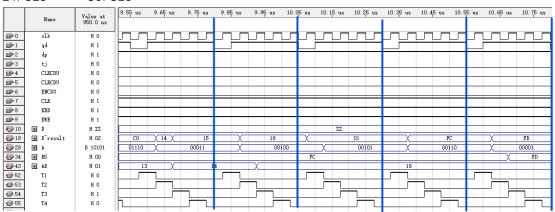
030312

FCFC12

重复取指令过程

数码管显示为:

14FC13 C0FC13



完成 ADD 指令, 实现 NOT (C) 加 A

数码管显示为:

15FC14 18FC18

01FC18

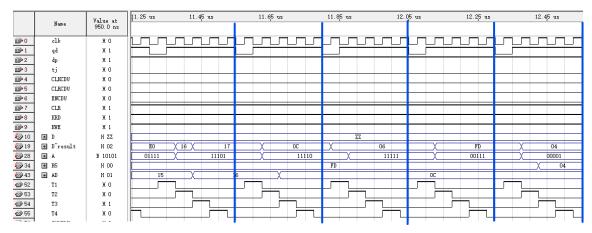
FCFD18

FDFD18

重复取指令过程

数码管显示为:

16FD15 E0FD15



完成 AND 指令,实现 R5 与 0C 地址单元的数据相与,实现(NOT(C)加 A)AND(B 加 D)数码管显示为:

重复取指令过程

数码管显示为:

180417 800417

	Name	Value at 950.0 ns	13,05 us 13.15 us 13.25 us 13.35 us 13.45 us 13.5
₽ 0	clk	но	
■ 1	qd	Н 1	
■ 2	$d_{\mathbf{p}}$	Н 1	
■ 3	tj	но	
₩ 4	CLKCDU	но	
₽ 5	CLRCDU	но	
№ 6	ENCDU	но	
₽ 7	CLR	Н 1	
№ 8	KRD	н 1	
■ 9	KWE	Н 1	
10	∓ D	H ZZ	ZZ
19	⊞ D~result	н 02	80 X 04 X FB
⊚ 28	 A	B 10101	01100 11011 00001
⊚ 34	⊞ R5	н оо	04 FB
⊚ 43	⊞ AD	н 01	17
⊚ 52	T1	но	
⊚ 53	T2	но	
€ 54	T3	н 1	
⊚ 55	T4	но	

完成 COM 指令, R5 的数据 取反送入 R5, 实现 NOT (NOT(C)加 A)AND(B 加 D)

数码管显示为: 040417 FBFB17

结束,完成 NOT (NOT(C)加 A)AND(B 加 D)复合运算。

3.

- a) 依据发出的 4 个节拍信号完成的任务来进行划分
- b) 通过在定义信号时对其进行初始化,顺序写入各个地址单元的值
- c) 第一句是从 ram 中取出 ar 的地址对应的单元的值送入 bus_reg_t2, 起到了从 ram 中读的作用, 第二句是将 r5 寄存器的值送入 ram 中 ar 的地址对应的单元中, 起到了向 ram 中写的作用。
- 4. 通过用多个进程来实现,每个进程对应一个时钟信号,如 Microcomputer.vhd 中,便用 4 个进程 ct1 到 ct4 来对应了 4 个时钟信号 t1 到 t4。每个进程中只含有一个且未被使用过的时钟信号,在该进程中,完成当前时钟信号有效情况下需要完成的任务。

5. 总结:

有一说一,这两次的实验与之前的运算器存储器等相比,难度确实增大许多,虽然有实验引导部分,但是涵盖的知识点,需要用到的东西比原先多了不止一倍,仅仅是先阅读相关内容是远远不够的,我真正地理解了实验的部分是在不了解的情况下,先去实现这个实验,然后在实现过程中,才对控制器和模型机有了初步的了解,实践出真知。之后通过模型机的时许节拍信号的分配,完成各个部分的整合,才对这一整个实验有了整体的认识,熟悉的了解各个部分的作用,什么时候工作的,怎样工作的,于是在之后的调试环节,也就很得心应手了。不过实验过程中难免会有困难,比如模型机时序仿真无法得到正确结果什么的,这时候与同学交流往往可以得到很好的解决。我感觉,动手+交流,是本次实验较为成功的关键。