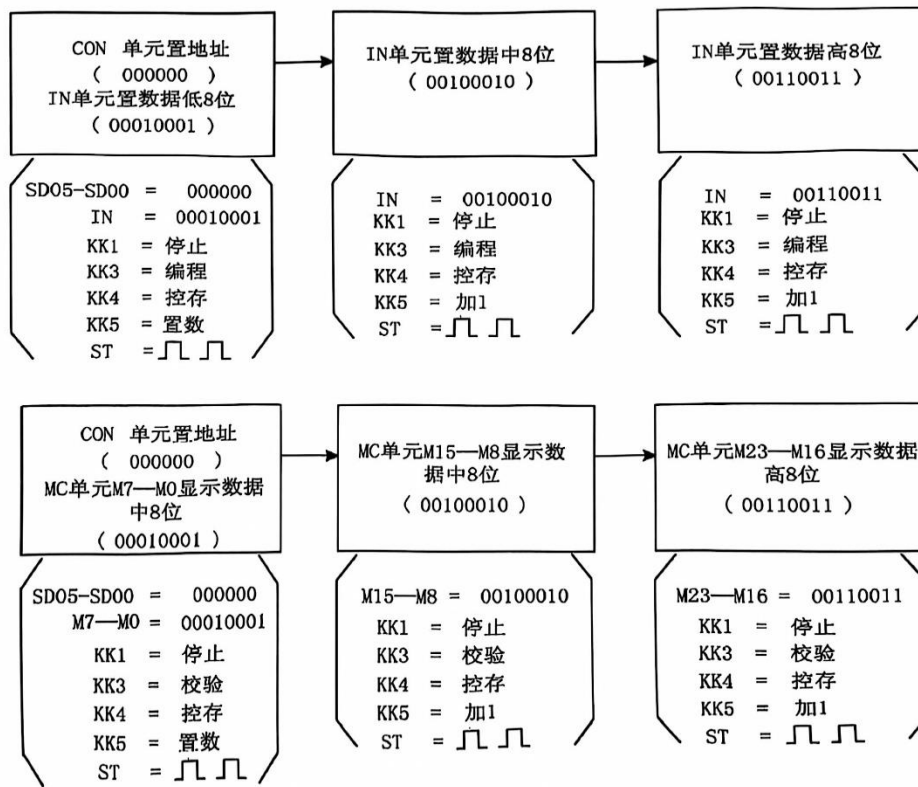


图 4-5-1 微程序控制器原理框图



23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	IOM	S3-S0	A字段	B字段	C字段	MA5-MA0

A字段				B字段				C字段			
14	13	12	选择	11	10	9	选择	8	7	6	选择
0	0	0	NOP	0	0	0	NOP	0	0	0	NOP
0	0	1	LDA	0	0	1	ALU_B	0	0	1	P<1>
0	1	0	LDB	0	1	0	RO_B	0	1	0	保留
0	1	1	LDRO	0	1	1	保留	0	1	1	保留
1	0	0	保留	1	0	0	保留	1	0	0	保留
1	0	1	保留	1	0	1	保留	1	0	1	保留
1	1	0	保留	1	1	0	保留	1	1	0	保留
1	1	1	LDIR	1	1	1	保留	1	1	1	保留

表 4-5-1 微指令格式

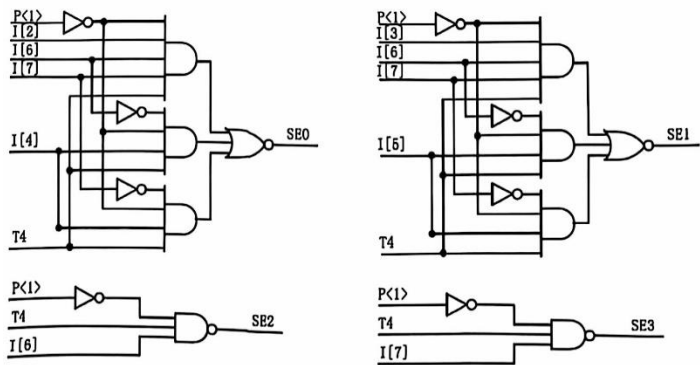


图 4-5-3 指令译码原理

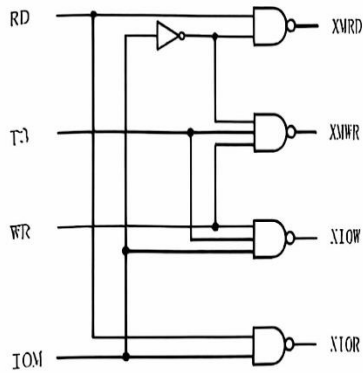


图 4-5-4 读写控制逻辑

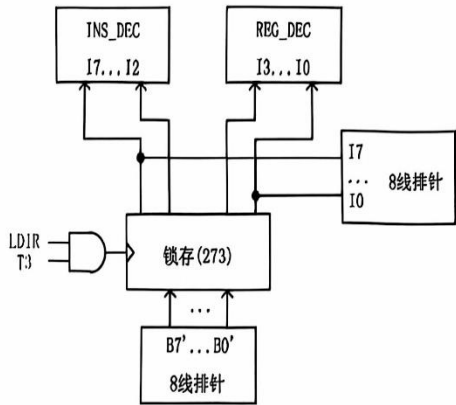


图 4-5-5 IR 单元的原理图

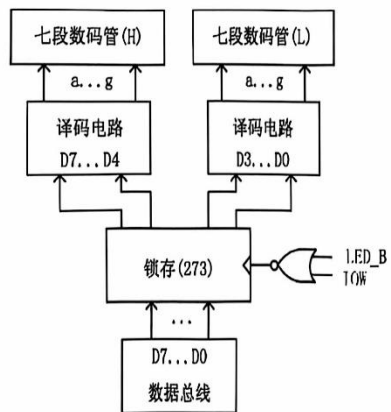


图 4-5-6 OUT 单元的原理图

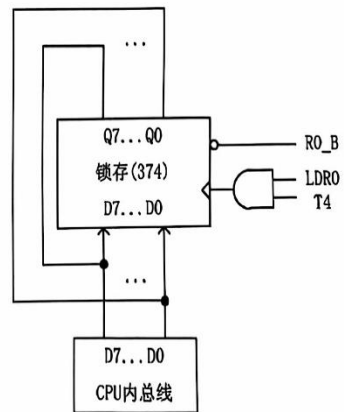


图 4-5-7R0 单元原理

其中 MA5...MA0 为 6 位的后续微地址, A、B、C 为三个译码字段, 分别由三个控制位译码出多位。C 字段中的 P<1>为测试字位。其功能是根据机器指令及相应微代码进行译码, 使微程序转入相应的微地址入口, 从而实现指令的识别, 并实现微程序的分支, 本系统上的指令译码原理如图 4-5-3 所示, 图中 I7...I2 为指令寄存器的第 7...2 位输出, SE5...SE0 为微控器单元微地址锁存器的强置端输出, 指令译码逻辑在 IR 单元的 INS_DEC (GAL20V8) 中实现。

从图 4-5-2 中也可以看出, 微控器产生的控制信号比表 4-5-1 中的要多, 这是因为实验的不同, 所需的控制信号也不一样, 本实验只用到部分的控制信号。

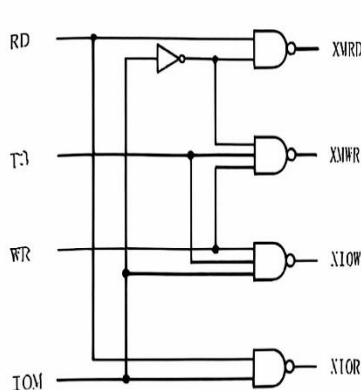


图 4-5-4 读写控制逻辑

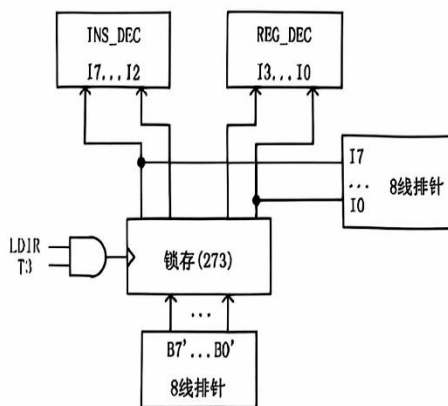


图 4-5-5 IR 单元的原理图

本实验除了用到指令寄存器 (IR) 和通用寄存器 R0 外, 还要用到 IN 和 OUT 单元, 从微控器出来的信号中只有 IOM、WR 和 RD 三个信号, 所以对这两个单元的读写信号还应先经过译码, 其译码原理如图 4-5-4 所示。IR 单元的原理图如图 4-5-5 所示, R0 单元原理如图 4-5-7 所示, IN 单元的原理图见图 2-18 所示, OUT 单元的原理图见图 4-5-6 所示。

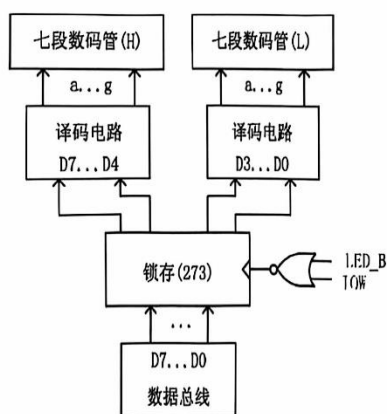


图 4-5-6 OUT 单元的原理图

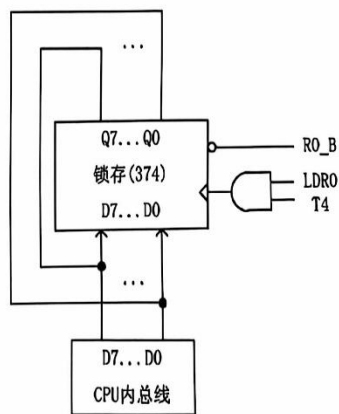


图 4-5-7 R0 单元原理图

本实验安排了四条机器指令, 分别为 ADD (0000 0000)、IN (0010 0000)、OUT (0011 0000) 和 HLT (0101 0000), 括号中为各指令的二进制代码, 指令格式如下:

LDRI 根据 10, 11 → 10=1 11=1 时 LDRI 有效
 RS.B 根据 12, 13 → 12=0 13=0 RS.B 有效

本实验在前一个实验的基础上增加了三个部件，一是 PC（程序计数器），另一个是 AR（地址寄存器），还有就是 MEM（主存），因而在微指令中应增加相应的控制位，其微指令格式如表 4-6-1 所示。

表 4-6-1 微指令格式

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	10M	S3-S0	A'字段	B'字段	C'字段	MA5-MA0

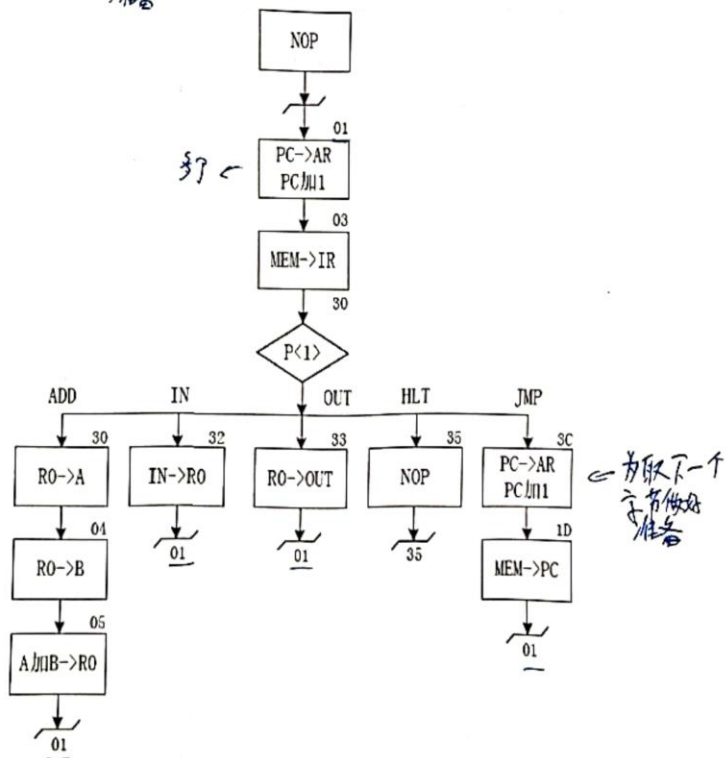
A'字段				B'字段				C'字段			
14	13	12	选择	11	10	9	选择	8	7	6	选择
0	0	0	NOP	0	0	0	NOP	0	0	0	NOP
0	0	1	LDA	0	0	1	ALU_B	0	0	1	P<1>
0	1	0	LDB	0	1	0	RO_B	0	1	0	保留
0	1	1	LDRO	0	1	1	保留	0	1	1	保留
1	0	0	保留	1	0	0	保留	1	0	0	保留
1	0	1	LOAD	1	0	1	保留	1	0	1	LDPC
1	1	0	LDAR	1	1	0	PC_B	1	1	0	保留
1	1	1	LDIR	1	1	1	保留	1	1	1	保留

LOAD=1
 LDPC=1
 当 T3 到来 PC+1

没选中数就空

每次开始运行前按 CLR 键将 PC、AR 清零
 PC 有自动加 1 功能，为下一条取指令做准备

图 4-6-4 简单模型机微程序流程图



系统涉及到的微程序流程图 5-1-4 所示，当拟定“取指”微指令时，该微指令的判别测试字段为

P<1>测试。指令译码原理见图 4-5-3 所示，由于“取指”微指令是所有微程序都使用的公用微指令，因

LDPC=1, T3 脉冲到来: PC-B=0, PC 内容送入 AR
 ① LOAD=1 → PC+1
 ② LDPC=0 → PC 内容
 LDPC=0 时, T3 脉冲到来

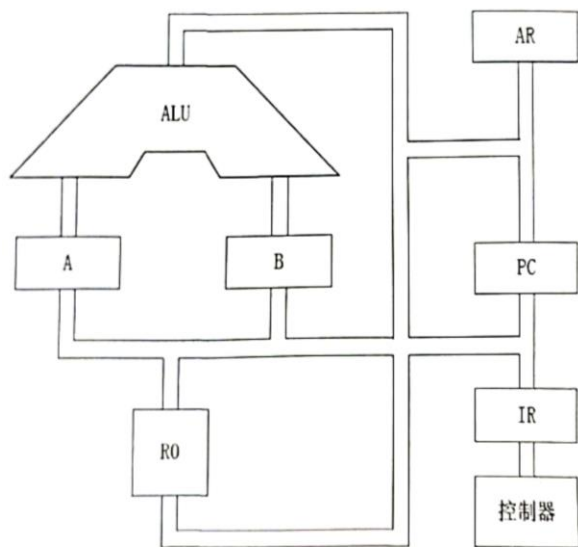


图 4-6-1 基本 CPU 构成原理图

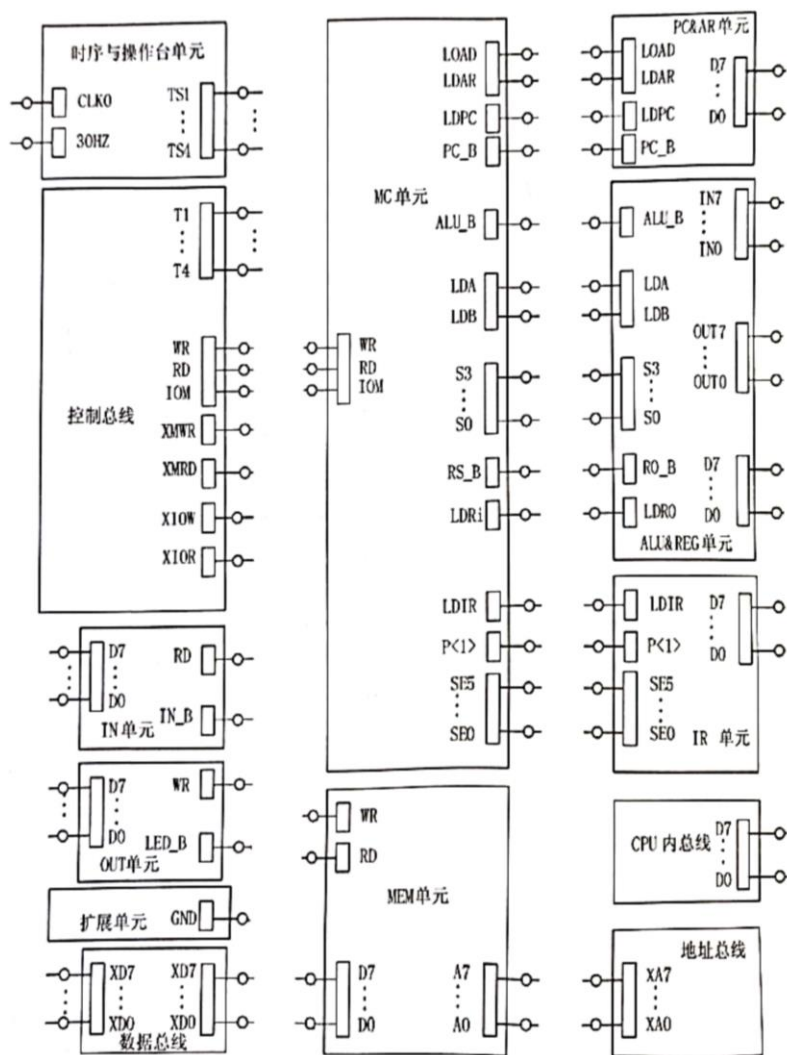


图 4-6-5 实验接线图

表 4-6-1 微指令格式

23	22	21	20	19	18-15	14-12	11-9	8-6	5-0
M23	M22	WR	RD	10M	S3-S0	A'字段	B'字段	C'字段	MA5-MA0

A'字段				B'字段				C'字段			
14	13	12	选择	11	10	9	选择	8	7	6	选择
0	0	0	NOP	0	0	0	NOP	0	0	0	NOP
0	0	1	LDA	0	0	1	ALU_B	0	0	1	P<1>
0	1	0	LDB	0	1	0	RO_B	0	1	0	保留
0	1	1	LDRO	0	1	1	保留	0	1	1	保留
1	0	0	保留	1	0	0	保留	1	0	0	保留
1	0	1	LOAD	1	0	1	保留	1	0	1	LDPC
1	1	0	LDAR	1	1	0	PC_B	1	1	0	保留
1	1	1	LDIR	1	1	1	保留	1	1	1	保留

图 4-6-4 简单模型机微程序流程图

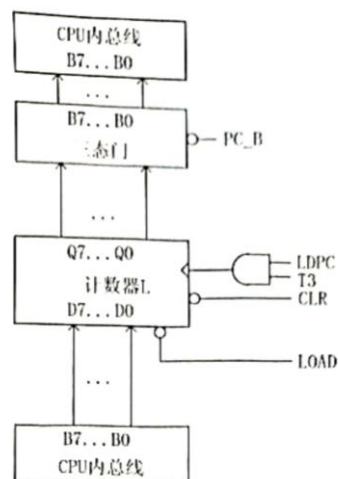
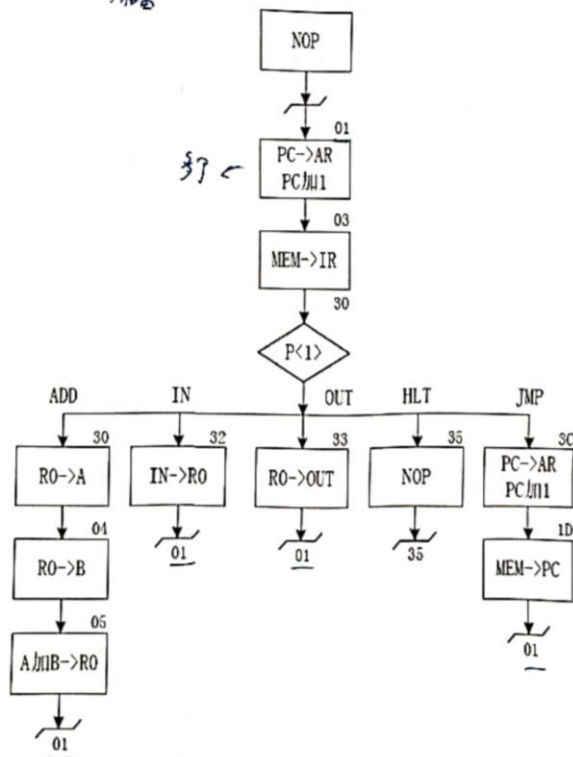


图 4-6-2 程序计数器 (PC) 原理图

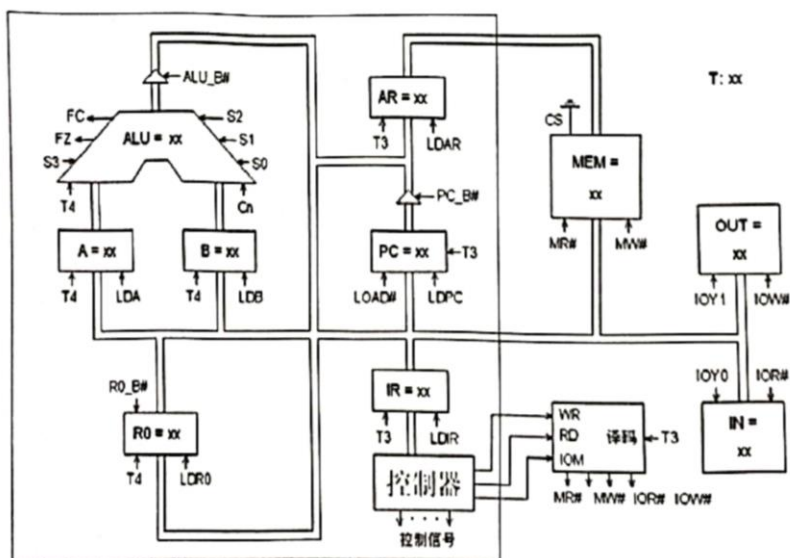


图 4-6-3 数据通路图

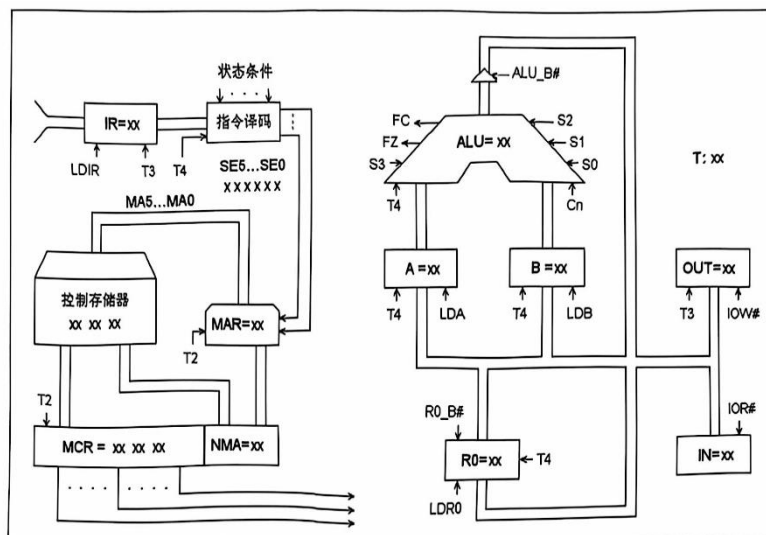


图 4-5-8 数据通路图

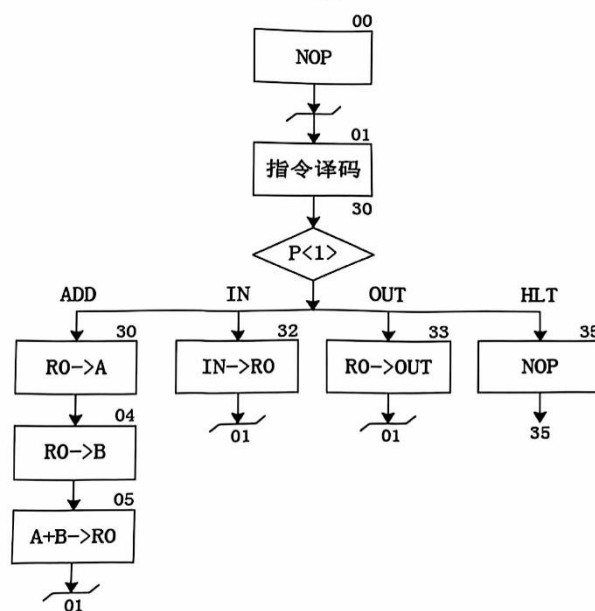


图 4-5-9 微程序流程图

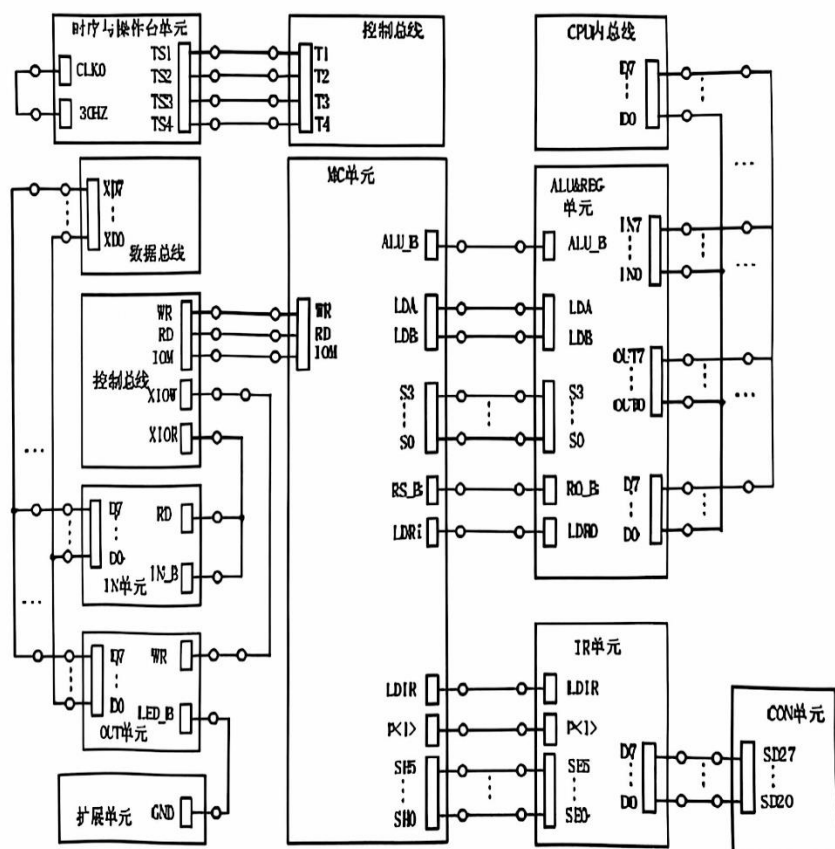


图 4-5-10 实验线路连接图

