

## 实验二 模型机中组合部件的实现（一）

### 一、实验目的

1. 了解简易模型机的内部结构和工作原理。
2. 熟悉译码器、运算器的工作原理。
3. 分析模型机的功能，设计指令译码器。
4. 分析模型机的功能，设计 ALU。

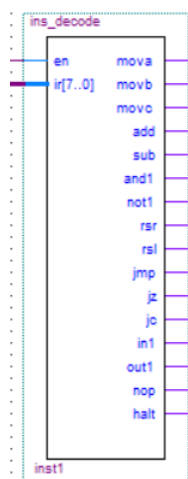
### 二、实验背景

#### 1. 指令译码器

指令译码器是根据指令系统表中的指令编码，对输入的 8 位指令进行解析，判定是哪条指令，则对应指令的输出为 1，否则输出为 0。

表 1 指令系统表

汇编符号	功能	编码
MOV R1, R2	$(R2) \rightarrow R1$	1100 R1 R2
MOV M, R2	$(R2) \rightarrow (C)$	1100 11 R2
MOV R1, M	$((C)) \rightarrow R1$	1100 R1 11
ADD R1, R2	$(R1) + (R2) \rightarrow R1$	1001 R1 R2
SUB R1, R2	$(R1) - (R2) \rightarrow R1$	0110 R1 R2
AND R1, R2	$(R1) \& (R2) \rightarrow R1$	1011 R1 R2
NOT R1	$\neg (R1) \rightarrow R1$	0101 R1 XX
RSR R1	$(R1)$ 循环右移一位 $\rightarrow R1$	1010 R1 00
RSL R1	$(R1)$ 循环左移一位 $\rightarrow R1$	1010 R1 11
JMP add	$add \rightarrow PC$	0011 00 00, address
JZ add	结果为 0 时 $add \rightarrow PC$	0011 00 01, address
JC add	结果有进位时 $add \rightarrow PC$	0011 00 10, address
IN R1	(开关 7-0) $\rightarrow R1$	0010 R1 XX
OUT R2	$(R2) \rightarrow$ 发光二极管 7-0	0100 XX R2
NOP		0111 00 00
HALT	停机	1000 00 00



指令译码器的输入输出引脚如上图所示。**en** 为使能信号，**ir[7..0]** 是 8 位指令编码，输出是对应的 16 条指令。引脚之间的相互关系如下表所示：

表 2 指令译码器引脚关系

en	ir[7..0]	16 个输出信号
1	8 位的指令编码	指令编码对应的指令输出为 1，其它输出为 0
0	8 位的指令编码	不管 ir 为何值，16 个输出全为 0

## 2. ALU

算术逻辑运算类指令：

ADD R1, R2

SUB R1, R2

AND R1, R2

NOT R1

这类指令的执行过程为：

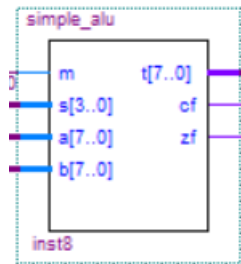
由 R2 的编码通过 RAA1、RAA0 从通用寄存器组 A 口读出 R2 的内容,由 R1 的编码通过 RWBA1、RWBA0 从通用寄存器组 B 口读出 R1 的内容，在 S3~S0 和 M 的控制下，实现运算，经移位逻辑送入总线 BUS；由 /WE 控制和 R1 的编码选择 RWBA1、RWBA0，将 BUS 上的数据写入通用寄存器 R1。其中 ADD 和 SUB 指令影响状态位 Cf 和 Zf。

指令具体功能如下：

汇编符号	功能	编码
ADD R1, R2	$(R1) + (R2) \rightarrow R1$	1001 R1 R2
SUB R1, R2	$(R1) - (R2) \rightarrow R1$	0110 R1 R2

AND R1, R2	$(R1) \& (R2) \rightarrow R1$	1011 R1 R2
NOT R1	$\neg (R1) \rightarrow R1$	0101 R1 XX

ALU 除了要完成 ADD、SUB、AND、NOT 运算外，还需在 MOVA、MOVB、RSR、RSL 和 OUT 五条指令执行时，提供将数据传送至总线的数据通路。ALU 模块的输入输出引脚如下图所示：



其中 m 和 s[3..0] 是控制信号，控制 a[7..0] 和 b[7..0] 输入的数据进行什么操作，并将产生的结果输出到 t[7..0]、cf 和 zf。各引脚间的相互关系如下表所示：

表 3 ALU 引脚关系

m	s[3..0]	t[7..0]	cf	zf
1	1001	$t=a+b$	有进位, cf=1 无进位, cf=0	和为零, zf=1 和不为零, zf=0
1	0110	$t=b-a$	有借位, cf=1 无借位, cf=0	差为零, zf=1 差不为零, zf=0
1	1011	$t=a\&b$	不影响	不影响
1	0101	$t=\neg b$ (注: b 相反)	不影响	不影响
0	1010	$t=b$	不影响	不影响
0	1100 或 0100	$t=a$	不影响	不影响

### 三、实验内容

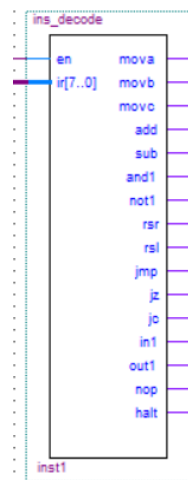
1. 用 VERILOG 语言设计指令译码器；
2. 用 VERILOG 语言设计 ALU。

### 四、实验要求

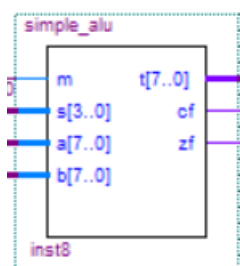
1. 完成学习通平台的实验作业。
2. 采用 VERILOG 语言设计逻辑电路，再利用波形编辑区进行仿真验证，以此验证电路的正确性。
3. 在 Tool 下用 netlist viewer 查看 RTL viewer，了解语句描述对应的 RTL 视图。

4. 最后撰写实验报告，提交至学习通平台，做得好的同学将在学习通平台分享设计。

5. 指令译码器的文件名为ins\_decode.v，接口信号名称必须严格按照下图命名（区分大小写），文件名、接口信号名称不合要求将会判为0分。



6. ALU的文件名为simple.v，接口信号名称必须按照下图命名(区分大小写)，文件名、接口信号名称不合要求将会判为0分。



## 五、思考题

1. 指令译码器必须要 16 个输出吗？可否将一些输出合并，哪些可以合并，为什么？

2. ALU 中的 `S[3..0]` 控制信号是来自哪里或者说与什么信息相同？

3、为何 `S[3..0]` 等于 1100 时将输入 `a` 传给 `t`，`S[3..0]` 等于 1010 或 0100 时将输入 `b` 传给 `t`？