

计算机组成与结构专题实验

实验报告

第三次 算术逻辑运算单元 ALU

一、 实验目的

探究运算器的数据传输路径。

验证运算器的组合逻辑功能。

深入理解算术逻辑运算中加法和减法的工作机制。

验证实验台在 8 位运算中加法、减法、逻辑与以及直通功能的正确性。

根据给定的数据，完成若干指定的算术运算和逻辑运算任务。

二、 实验原理

在计算机的中央处理器（CPU）中，算术逻辑单元（ALU）是核心组件之一，负责执行算术运算（如加法和减法）以及逻辑运算（如“与”、“或”、“非”）。对于一个 8 位的算术逻辑单元，其基本功能可以通过由标准逻辑器件 74LS181（4 位）组合而成的电路结构来实现。这种电路结构的功能可以用硬件描述语言（HDL）进行表述。表 4-1 详细列出了 ALU 的基本算术和逻辑功能。

表 4-1 ALU 的运算功能

选择端 S3 S2 S1 S0	M=H 逻辑操作	M=L 算术操作	
	逻辑功能	Cn=L (无进位)	Cn=H (有进位)
0 0 0 0	$F = A$	$F = A$	$F = A$ 加1
0 0 0 1	$F = \overline{A+B}$	$F = A+B$	$F = (A+B)$ 加1
0 0 1 0	$F = AB$	$F = A+\overline{B}$	$F = A+B+1$
0 0 1 1	$F = 0$	$F = \text{减1 (2的补码)}$	$F = 0$
0 1 0 0	$F = \overline{AB}$	$F = A$ 加 \overline{AB}	$F = A$ 加 \overline{AB} 加1
0 1 0 1	$F = \overline{B}$	$F = (A+B)$ 加 \overline{AB}	$F = (A+B)$ 加 $\overline{AB}+1$
0 1 1 0	$F = A \oplus B$	$F = A$ 减 B	$F = A$ 减 B 减1
0 1 1 1	$F = \overline{AB}$	$F = A+\overline{B}$	$F = (A+\overline{B})$ 减1
1 0 0 0	$F = A+B$	$F = A$ 加 AB	$F = A$ 加 AB 加1
1 0 0 1	$F = \overline{A \oplus B}$	$F = A$ 加 B	$F = A$ 加 B 加1
1 0 1 0	$F = \overline{B}$	$F = (A+\overline{B})$ 加 AB	$F = (A+\overline{B})$ 加 AB 加1
1 0 1 1	$F = AB$	$F = AB$	$F = AB$ 减1
1 1 0 0	$F = 1$	$F = A$ 加 A^*	$F = A$ 加 A 加1
1 1 0 1	$F = A+\overline{B}$	$F = (A+B)$ 加 A	$F = (A+B)$ 加 A 加1
1 1 1 0	$F = A+B$	$F = (A+\overline{B})$ 加 A	$F = (A+\overline{B})$ 加 A 加1
1 1 1 1	$F = A$	$F = A$	$F = A$ 减1

ALU181 是基于 74LS181 功能使用 VHDL 硬件描述语言设计而成的 8 位运算器。参与运算的两个 8 位数据分别为 $A[7..0]$ 和 $B[7..0]$ 。运算模式由 $S[3..0]$ 的 16 种不同组合决定， $S[3..0]$ 的值由一个 4 位二进制计数器 LPM_COUNTER 产生，其计数时钟为 $Sc1k$ 。此外， $M=0$ 表示选择算术运算， $M=1$ 表示选择逻辑运算； CN 为低位进位输入； $F[7..0]$ 为运算结果输出； $C0$ 为运算后的高位进位输出。两个 8 位数据 A 和 B 通过总线 $IN[7..0]$ ，分别经两个 74373 电平锁存器锁存后输入 ALU181。ALU181 的功能如表 4-1 所示。

ALU 的 Verilog 程序如下：

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY ALU181 IS
PORT (
S : IN STD_LOGIC_VECTOR(3 DOWNTO 0 );
A : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
B : IN STD_LOGIC_VECTOR(7 DOWNTO 0);

```

```

F : OUT STD_LOGIC_VECTOR(7 DOWNT0 0);
M : IN STD_LOGIC;
CN : IN STD_LOGIC;
CO : OUT STD_LOGIC );
END ALU181;
ARCHITECTURE behav OF ALU181 IS
SIGNAL A9 : STD_LOGIC_VECTOR(8 DOWNT0 0);
SIGNAL B9 : STD_LOGIC_VECTOR(8 DOWNT0 0);
SIGNAL F9 : STD_LOGIC_VECTOR(8 DOWNT0 0);
BEGIN
A9 <= '0' & A ; B9 <= '0' & B ;
PROCESS(M,CN,A9,B9)
BEGIN
CASE S IS
WHEN "0000" => IF M='0' THEN F9<=A9 + CN ; ELSE F9<=NOT A9; END IF;
WHEN "0001" => IF M='0' THEN F9<=(A9 or B9) + CN ; ELSE F9<=NOT(A9 OR
B9); END IF;-56-
WHEN "0010" => IF M='0' THEN F9<=(A9 or (NOT B9))+ CN ; ELSE F9<=(NOT
A9) AND B9; END IF;
WHEN "0011" => IF M='0' THEN F9<= "000000000" - CN ; ELSE F9<="000000000";
END IF;
WHEN "0100" => IF M='0' THEN F9<=A9+(A9 AND NOT B9)+ CN; ELSE F9<=NOT
(A9 AND B9); END IF;
WHEN "0101" => IF M='0' THEN F9<=(A9 or B9)+(A9 AND NOT B9)+CN; ELSE
F9<=NOT B9; END IF;
WHEN "0110" => IF M='0' THEN F9<=(A9 - B9) - CN ; ELSE F9<=A9 XOR B9;
END IF;
WHEN "0111" => IF M='0' THEN F9<=(A9 or (NOT B9)) - CN; ELSE F9<=A9 and
(NOT B9); END IF;
WHEN "1000" => IF M='0' THEN F9<=A9 + (A9 AND B9)+CN; ELSE F9<=(NOT A9)and
B9; END IF;
WHEN "1001" => IF M='0' THEN F9<=A9 + B9 + CN ; ELSE F9<=NOT(A9 XOR B9);
END IF;
WHEN "1010" => IF M='0' THEN F9<=(A9 or(NOT B9))+(A9 AND B9)+CN; ELSE
F9<=B9; END IF;
WHEN "1011" => IF M='0' THEN F9<=(A9 AND B9)- CN ; ELSE F9<=A9 AND B9;
END IF;
WHEN "1100" => IF M='0' THEN F9<=(A9 + A9) + CN ; ELSE F9<= "000000001";
END IF;
WHEN "1101" => IF M='0' THEN F9<=(A9 or B9) + A9 + CN ; ELSE F9<=A9 OR
(NOT B9); END IF;
WHEN "1110" => IF M='0' THEN F9<=((A9 or (NOT B9)) +A9) + CN ; ELSE F9<=A9
OR B9; END IF;
WHEN "1111" => IF M='0' THEN F9<=A9 - CN ; ELSE F9<=A9 ; END IF;

```

```
WHEN OTHERS => F9<= "000000000" ;  
END CASE;  
END PROCESS;
```

三、 实验任务

1. 设计 ALU 元件

在 Quartus II 软件环境中，创建 VHDL 文件，并输入算术逻辑单元（ALU）的 VHDL 代码。编译该 VHDL 文件，并将其封装为一个可调用的原理图元件，以便后续使用。

2. 建立顶层文件工程

采用图形化方式创建顶层文件工程。根据电路图，从 Quartus II 的基本元件库中调取所需元件，并将其放置在图形编辑窗口中进行连线。同时，添加输入输出引脚，并将设计好的原理图文件保存为 ALU.bdf。将该文件设置为当前工程文件，后续操作均基于此工程文件进行。

3. 器件选择与引脚锁定

根据芯片型号及康芯提供的引脚表，完成引脚分配。具体步骤为：先编译工程，然后进行引脚锁定，最后再次编译以确认锁定信息并生成芯片编程所需的文件

4. 引脚锁定

根据电路模式 0 的要求，对各信号进行引脚锁定：

数据输入端 IN 锁定为 PI08~PI018（通过按键 1 和 2 输入数据 A[7..0] 和 B[7..0]）。

A0_B1 选择信号（用于选择 A 或 B）锁定为 PI03（通过按键 3 控制）。

进位输入 CN 锁定为 PI06（通过按键 7 控制）。

运算模式选择信号 M 锁定为 PI07（通过按键 8 控制）。

时钟信号 Sclk 锁定为 PI05（通过按键 6 控制）。

计算结果 F 输出锁定为数码管 5 和 6（PI032~PI039）

数据 A[7..0] 输出锁定为数码管 1 和 2（PI016~PI023）。

数据 B[7..0] 输出锁定为数码管 3 和 4（PI024~PI031）。

计算后的进位输出 CN4 锁定为数码管 7 的最低位 PI040。

运算模式选择信号 S[3..0] 输出锁定为数码管 8（PI044~PI047）

5. 芯片编程

打开编程窗口，将编译生成的配置文件 ALU.sof 下载 FPGA 芯片中。

6. 功能验证

将实验系统的电路模式设置为 0，根据功能表，通过按键 1 和 2 输入数据 A[7..0] 和 B[7..0]，并设置 S[3..0]、M 和 CN 的值，验证 ALU 运算器的算术运算和逻辑运算功能。记录实验数据以供分析。

四、 实验步骤及结果

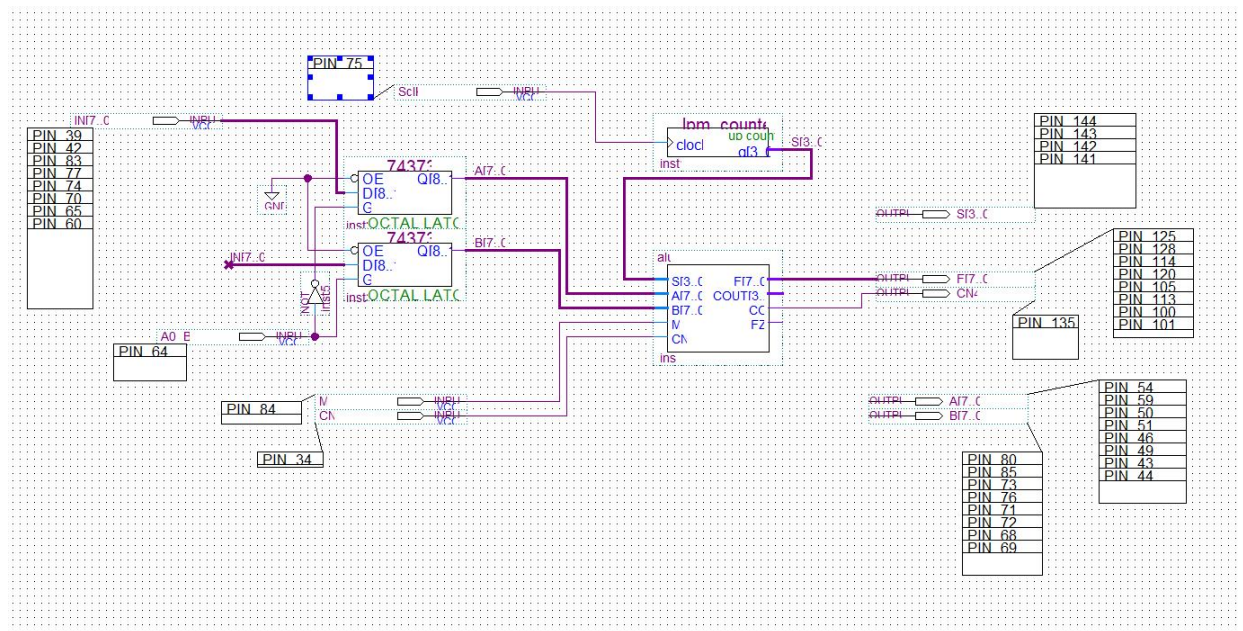
设计 ALU 元件

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY alu IS
PORT ( S: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
      A,B : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
      F: OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
      COUT : OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
      M,CN : IN STD_LOGIC;
      CO,FZ : OUT STD_LOGIC );
END alu;
ARCHITECTURE behav OF alu IS
SIGNAL A9,B9,F9 : STD_LOGIC_VECTOR(8 DOWNTO 0);
BEGIN
A9 <= '0' & A; B9 <= '0' & B;
PROCESS(M,CN,A9,B9)
BEGIN
CASE S IS
WHEN "0000"=>IF M='0' THEN F9<=A9 + CN; ELSE F9<=NOT A9; END IF;
WHEN "0001"=>IF M='0' THEN F9<=(A9 OR B9)+CN; ELSE F9<=NOT(A9 OR B9); END IF;
WHEN "0010"=>IF M='0' THEN F9<=(A9 OR (NOT B9))+CN; ELSE F9<=(NOT A9) AND B9; END IF;
WHEN "0011"=>IF M='0' THEN F9<= "0000000000"-CN; ELSE F9<="0000000000"; END IF;
WHEN "0100"=>IF M='0' THEN F9<=A9+(A9 AND NOT B9)+CN; ELSE F9<=NOT (A9 AND B9); END IF;
WHEN "0101"=>IF M='0' THEN F9<=(A9 OR B9)+(A9 AND NOT B9)+CN; ELSE F9<=NOT B9; END IF;
WHEN "0110"=>IF M='0' THEN F9<=A9 - B9 - CN; ELSE F9<=A9 XOR B9; END IF;
WHEN "0111"=>IF M='0' THEN F9<=(A9 OR (NOT B9))-CN; ELSE F9<=A9 AND (NOT B9); END IF;
WHEN "1000" =>IF M='0' THEN F9<=A9 + (A9 AND B9)+CN; ELSE F9<=(NOT A9) OR B9; END IF;
WHEN "1001" =>IF M='0' THEN F9<=A9 + B9 + CN; ELSE F9<=NOT(A9 XOR B9); END IF;
WHEN "1010" =>IF M='0' THEN F9<=(A9 OR (NOT B9))+(A9 AND B9)+CN; ELSE F9<=B9; END IF;
WHEN "1011" =>IF M='0' THEN F9<=(A9 AND B9) - CN; ELSE F9<=A9 AND B9; END IF;
WHEN "1100" =>IF M='0' THEN F9<=A9 + A9 + CN; ELSE F9<= "0000000001"; END IF;
WHEN "1101" =>IF M='0' THEN F9<=(A9 OR B9)+A9 + CN; ELSE F9<=A9 OR (NOT B9); END IF;
WHEN "1110" =>IF M='0' THEN F9<=(A9 OR (NOT B9))+A9+CN; ELSE F9<=A9 OR B9; END IF;
WHEN "1111" =>IF M='0' THEN F9<=A9-CN; ELSE F9<=A9; END IF;
WHEN OTHERS =>F9<= "0000000000" ;
END CASE;
IF (A9= B9) THEN FZ <= '0';END IF;
END PROCESS;
F<= F9(7 DOWNTO 0); CO <= F9(8);
COUT <= "0000" WHEN F9(8) = '0' ELSE "0001" ;
END behav;

```

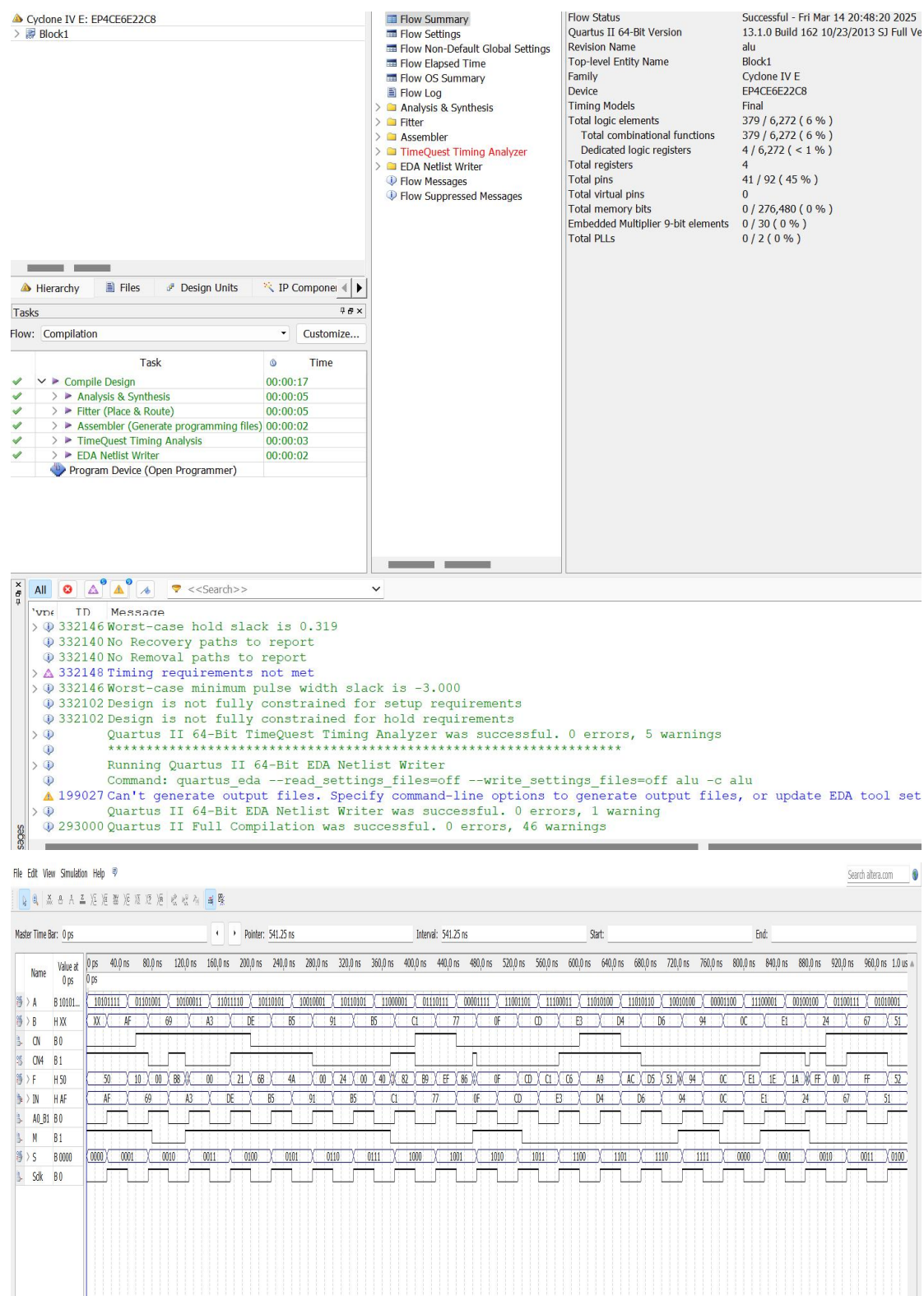
电路图绘制



绑定引脚

alu							
Home							
Block1.bdf*							
<<new>> Filter on node names: *							
	status:	From	To	Assignment Name	Value	Enabled	Entity
1	✓ ...		out A[7]	Location	PIN_54	Yes	
2	✓ ...		out A[6]	Location	PIN_59	Yes	
3	✓ ...		out A[5]	Location	PIN_50	Yes	
4	✓ ...		out A[4]	Location	PIN_51	Yes	
5	✓ ...		out A[3]	Location	PIN_46	Yes	
6	✓ ...		out A[2]	Location	PIN_49	Yes	
7	✓ ...		out A[1]	Location	PIN_43	Yes	
8	✓ ...		out A[0]	Location	PIN_44	Yes	
9	✓ ...		in A0_B1	Location	PIN_64	Yes	
10	✓ ...		out B[7]	Location	PIN_80	Yes	
11	✓ ...		out B[6]	Location	PIN_85	Yes	
12	✓ ...		out B[5]	Location	PIN_73	Yes	
13	✓ ...		out B[4]	Location	PIN_76	Yes	
14	✓ ...		out B[3]	Location	PIN_71	Yes	
15	✓ ...		out B[2]	Location	PIN_72	Yes	
16	✓ ...		out B[1]	Location	PIN_68	Yes	
17	✓ ...		out B[0]	Location	PIN_69	Yes	
18	✓ ...		in CN	Location	PIN_34	Yes	
19	✓ ...		out CN4	Location	PIN_135	Yes	
20	✓ ...		out F[7]	Location	PIN_125	Yes	
21	✓ ...		out F[6]	Location	PIN_128	Yes	
22	✓ ...		out F[5]	Location	PIN_114	Yes	
23	✓ ...		out F[4]	Location	PIN_120	Yes	
24	✓ ...		out F[3]	Location	PIN_105	Yes	
25	✓ ...		out F[2]	Location	PIN_113	Yes	
26	✓ ...		out F[1]	Location	PIN_100	Yes	
27	✓ ...		out F[0]	Location	PIN_101	Yes	
28	✓ ...		in IN[7]	Location	PIN_39	Yes	
29	✓ ...		in IN[6]	Location	PIN_42	Yes	
30	✓ ...		in IN[5]	Location	PIN_83	Yes	
31	✓ ...		in IN[4]	Location	PIN_77	Yes	
32	✓ ...		in IN[3]	Location	PIN_74	Yes	
33	✓ ...		in IN[2]	Location	PIN_70	Yes	
34	✓ ...		in IN[1]	Location	PIN_65	Yes	
35	✓ ...		in IN[0]	Location	PIN_60	Yes	
36	✓ ...		in M	Location	PIN_84	Yes	
37	✓ ...		out S[3]	Location	PIN_144	Yes	
38	✓ ...		out S[2]	Location	PIN_143	Yes	
39	✓ ...		out S[1]	Location	PIN_142	Yes	
40	✓ ...		out S[0]	Location	PIN_141	Yes	
41	✓ ...		in Sclk	Location	PIN_75	Yes	
42		<<new>>	<<new>>	<<new>>			

编译并下载至 FPGA，仿真



五、 实验总结及问题分析

理论/实验值对照表如下：

S3 S2 S1 S0	A[7..0]	B[7..0]	M=0 CN=0 (无进位)	M=0 CN=1 (有进位)	逻辑运算 (M=1)
0000 实际	AA	55	AA	AB	55
0000 理论	AA	55	AA	AB	55
0001 实际	AA	55	FF	00	00
0001 理论	AA	55	FF	00	00
0010 实际	AA	55	AA	AB	55
0010 理论	AA	55	AA	AB	55
0011 实际	AA	55	00	FF	00
0011 理论	AA	55	00	FF	00
0100 实际	AA	55	AA	AB	FF
0100 理论	AA	55	AA	AB	FF
0101 实际	AA	55	AA	AB	AA
0101 理论	AA	55	AA	AB	AA
0110 实际	AA	55	55	54	FF
0110 理论	AA	55	55	54	FF
0111 实际	AA	55	AA	A9	AA
0111 理论	AA	55	AA	A9	AA
1000 实际	AA	55	54	55	AA
1000 理论	AA	55	54	55	AA
1001 实际	AA	55	FF	00	00
1001 理论	AA	55	FF	00	00

1010 实际	AA	55	54	55	55
1010 理论	AA	55	54	55	55
1011 实际	AA	55	00	FF	00
1011 理论	AA	55	00	FF	00
1100 实际	AA	55	54	53	01
1100 理论	AA	55	54	53	01
1101 实际	AA	55	A9	AA	AA
1101 理论	AA	55	A9	AA	AA
1110 实际	AA	55	54	55	FF
1110 理论	AA	55	54	55	FF
1111 实际	AA	55	AA	A9	AA
1111 理论	AA	55	AA	A9	AA

可见理论值与实验值相符。