**信息科学与工程学院**

**2019－2020学年第二学期**

实 验 报 告

课程名称： 电子设计自动化

实验名称： 实验三 乘法器和除法器设计

专 业 班 级 通信工程 二班

学 生 学 号 201800121050

学 生 姓 名 孟麟芝

实 验 时 间 2020年4月

实验报告

【实验目的】

1.掌握使用VHDL语言进行硬件设计的基本方法

2.掌握基本的Quartus Ⅱ使用方法

3.学会testbench的编写，掌握波形仿真的基本方法

【实验要求】

1.移位相加型8位硬件乘法器设计。

2.高速硬件除法器设计。

3.串行静态显示控制电路设计。

【实验具体内容】

### 【第一个实验】

1.实验原理

该电路模拟人手算乘法时的算法而设计，根据电路图，详细操作步骤如下：

（1）一个时钟沿到来且选通端为0时，八位右移寄存器将输入的A0右移一位，QB输出了A0目前的最低位传给选通与门。

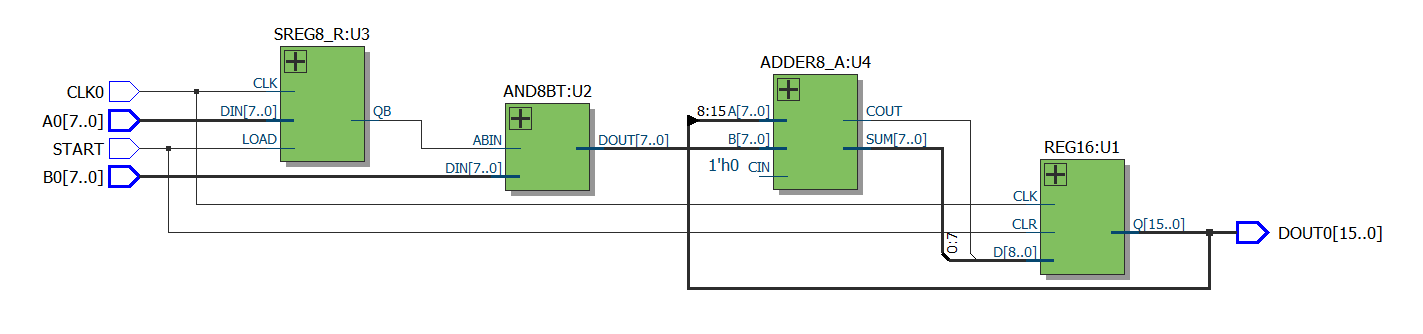
（2）若该位为0，则选通与门输出DOUT为“00000000”，若该位为1，则DOUT输出为B0

（3）十六位锁存器的低8位右移一位

（4）十六位锁存器高8位和选通与门输出的结果通过八位全加器相加

（5）十六位锁存器从D口接受全加器的结果，存到高9位

（6）重复操作上述步骤，第八个时钟沿到来时完成乘法



2.设计模块代码（Design Block）

（1）16位锁存器

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY REG16 IS

PORT ( CLK,CLR : IN STD\_LOGIC;

D : IN STD\_LOGIC\_VECTOR(8 DOWNTO 0);

Q : OUT STD\_LOGIC\_VECTOR(15 DOWNTO 0):="0000000000000000"

);

END REG16;

ARCHITECTURE behav OF REG16 IS

SIGNAL R16S :STD\_LOGIC\_VECTOR(15 DOWNTO 0);

BEGIN

PROCESS(CLK,CLR)

BEGIN

IF CLR = '1' THEN R16S <= (OTHERS =>'0');

ELSIF CLK'EVENT AND CLK = '1' THEN

R16S(6 DOWNTO 0) <= R16S(7 DOWNTO 1);

R16S(15 DOWNTO 7) <= D;

END IF;

END PROCESS;

Q <= R16S;

END behav;

（2）8位右移寄存器

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY SREG8\_R IS

PORT(CLK : IN STD\_LOGIC;

LOAD: IN STD\_LOGIC;

DIN : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

QB : OUT STD\_LOGIC

);

END SREG8\_R;

ARCHITECTURE BHV OF SREG8\_R IS

SIGNAL REG8: STD\_LOGIC\_VECTOR(7 DOWNTO 0);

BEGIN

PROCESS(CLK,LOAD)

BEGIN

IF (LOAD = '1') THEN

REG8 <= DIN;

ELSE

IF(CLK'EVENT AND CLK ='1') THEN

REG8(6 DOWNTO 0) <= REG8(7 DOWNTO 1);

END IF;

END IF;

END PROCESS;

QB <= REG8(0);

END BHV;

（3）选通与门

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY AND8BT IS

PORT ( ABIN : IN STD\_LOGIC;

DIN : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

DOUT : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0));

END AND8BT;

ARCHITECTURE BHV OF AND8BT IS

BEGIN

PROCESS(ABIN,DIN)

BEGIN

FOR I IN 0 TO 7 LOOP

DOUT(I) <= DIN(I) AND ABIN;

END LOOP;

END PROCESS;

END BHV;

（4）8位全加器

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY ADDER8\_A IS

PORT (A : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

B : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

CIN : IN STD\_LOGIC;

SUM : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0);

COUT : OUT STD\_LOGIC

);

END ENTITY ADDER8\_A;

ARCHITECTURE BHV OF ADDER8\_A IS

SIGNAL DATA :STD\_LOGIC\_VECTOR(8 DOWNTO 0);

BEGIN

DATA <= ('0'&A) + ('0'&B) + ("00000000"&CIN);

COUT <= DATA(8);

SUM <= DATA(7 DOWNTO 0);

END ARCHITECTURE BHV;

（5）总设计：移位相加型8位硬件乘法器

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY MULT8\_B IS

PORT(CLK0,START : IN STD\_LOGIC;

A0,B0 : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

DOUT0 : BUFFER STD\_LOGIC\_VECTOR(15 DOWNTO 0)

);

END MULT8\_B;

ARCHITECTURE BHV OF MULT8\_B IS

COMPONENT REG16 IS

PORT ( CLK,CLR : IN STD\_LOGIC;

D : IN STD\_LOGIC\_VECTOR(8 DOWNTO 0);

Q : OUT STD\_LOGIC\_VECTOR(15 DOWNTO 0)

);

END COMPONENT;

COMPONENT AND8BT IS

PORT ( ABIN : IN STD\_LOGIC;

DIN : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

DOUT : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0));

END COMPONENT;

COMPONENT SREG8\_R IS

PORT(CLK : IN STD\_LOGIC;

LOAD: IN STD\_LOGIC;

DIN : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

QB : OUT STD\_LOGIC

);

END COMPONENT;

COMPONENT ADDER8\_A IS

PORT (A : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

B : IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

CIN : IN STD\_LOGIC;

SUM : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0);

COUT : OUT STD\_LOGIC

);

END COMPONENT;

SIGNAL CONVEY1:STD\_LOGIC\_VECTOR(8 DOWNTO 0);

SIGNAL CONVEY2:STD\_LOGIC;

SIGNAL CONVEY3:STD\_LOGIC\_VECTOR(7 DOWNTO 0);

SIGNAL CONVEY4:STD\_LOGIC:='0';

BEGIN

U1:REG16 PORT MAP(CLK=>CLK0,CLR=>START,D=>CONVEY1,Q=>DOUT0);

U2:AND8BT PORT MAP(ABIN=>CONVEY2,DIN=>B0,DOUT=>CONVEY3);

U3:SREG8\_R PORT MAP(CLK=>CLK0,LOAD=>START,DIN=>A0,QB=>CONVEY2);

U4:ADDER8\_A PORT MAP(B=>CONVEY3,A=>DOUT0(15 DOWNTO 8),CIN=>CONVEY4,SUM=>CONVEY1(7 DOWNTO 0),COUT=>CONVEY1(8));

END BHV;

3.激励模块代码（Test Bench）

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY MULT8\_BTB IS

END MULT8\_BTB;

ARCHITECTURE ONE OF MULT8\_BTB IS

COMPONENT MULT8\_B

PORT(CLK0,START:IN STD\_LOGIC;

A0,B0:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

DOUT0:BUFFER STD\_LOGIC\_VECTOR(15 DOWNTO 0));

END COMPONENT;

SIGNAL CLK01:STD\_LOGIC:='0';

SIGNAL START1:STD\_LOGIC:='1';

SIGNAL A1:STD\_LOGIC\_VECTOR(7 DOWNTO 0);

SIGNAL B1:STD\_LOGIC\_VECTOR(7 DOWNTO 0);

SIGNAL DOUT1:STD\_LOGIC\_VECTOR(15 DOWNTO 0);

CONSTANT CLK\_P: TIME:=500 ns;

BEGIN

I1:MULT8\_B PORT MAP(CLK0=>CLK01,START=>START1,A0=>A1,B0=>B1,DOUT0=>DOUT1);

PROCESS BEGIN

CLK01<='0'; WAIT FOR CLK\_P;

CLK01<='1'; WAIT FOR CLK\_P;

END PROCESS;

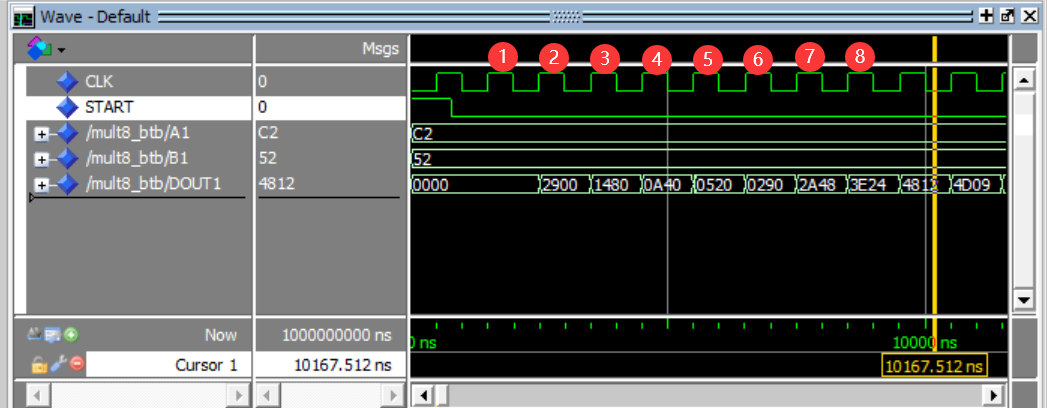
START1<='1', '0' AFTER 800 ns;

A1<="11000010";

B1<="01010010";

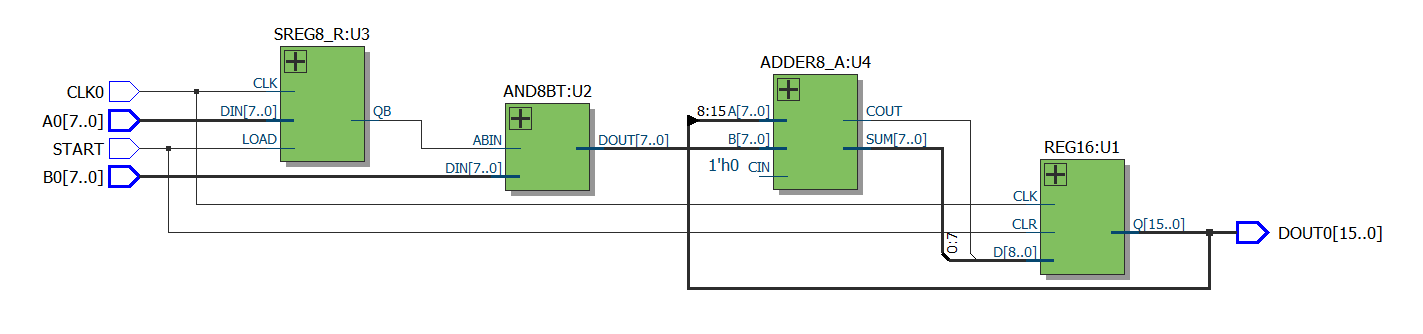
END ONE;

4.仿真波形图

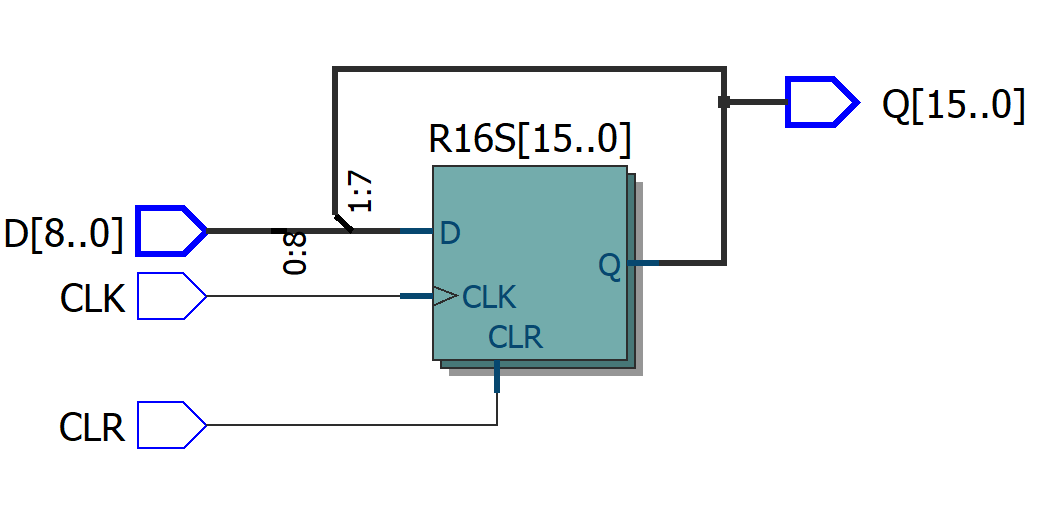


5.门级电路图

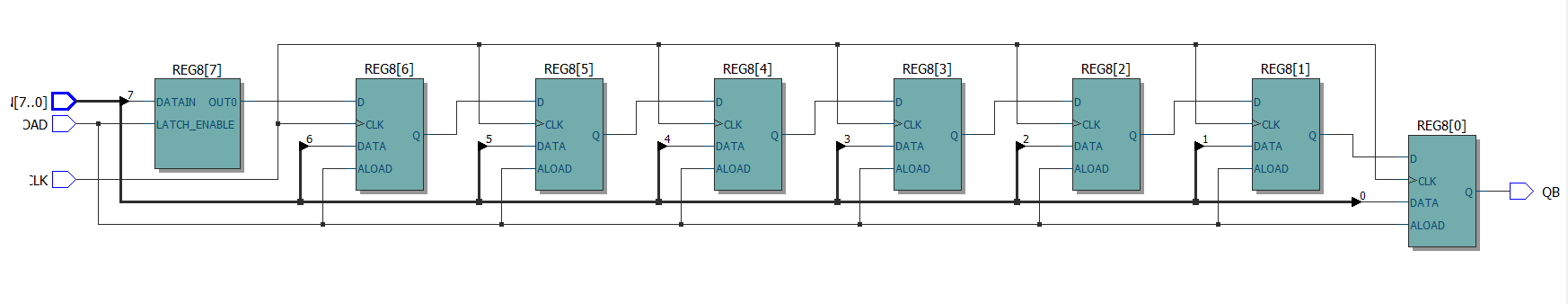
总设计：移位相加型8位硬件乘法器



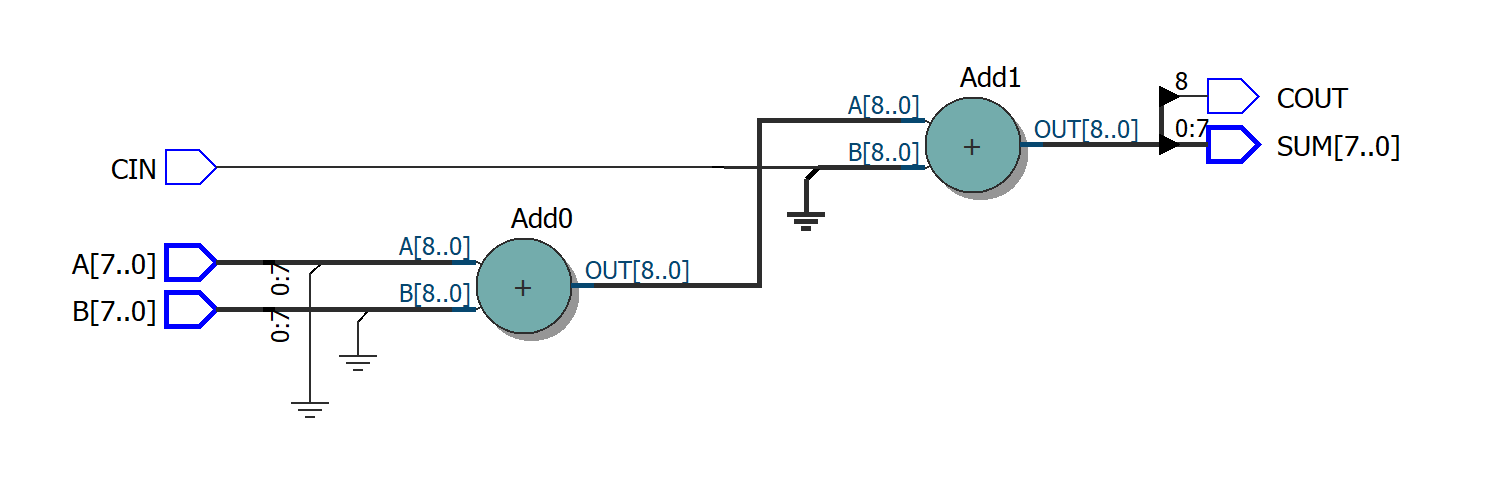
十六位锁存器



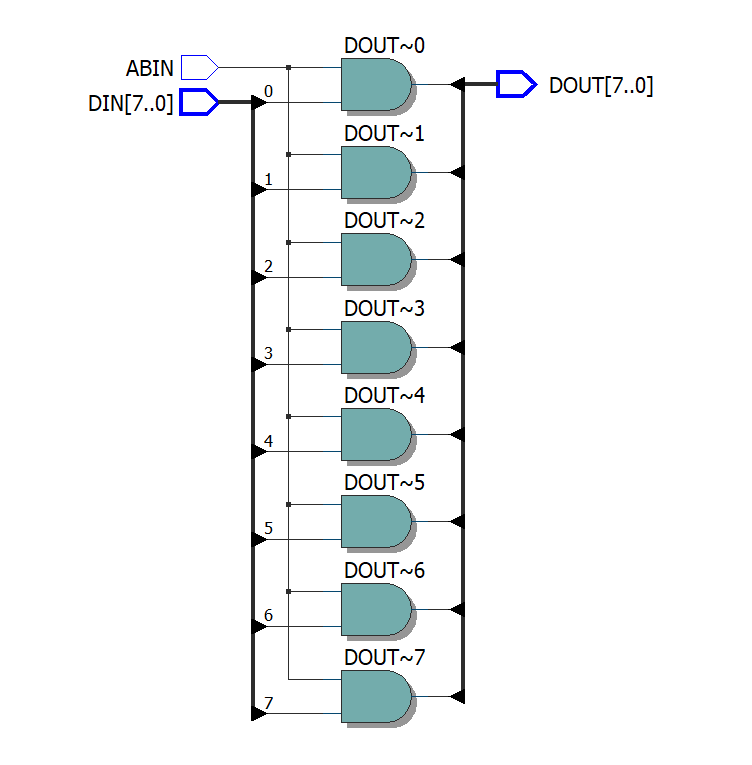
8位右移寄存器



8位全加器



选通与门



6.结果分析与思考

（1）一定要对十六位锁存器的输出设一个初值，不然选通端为0后第一个时钟沿到来时，会传给八位全加器的A口一个不定值，导致输出不定

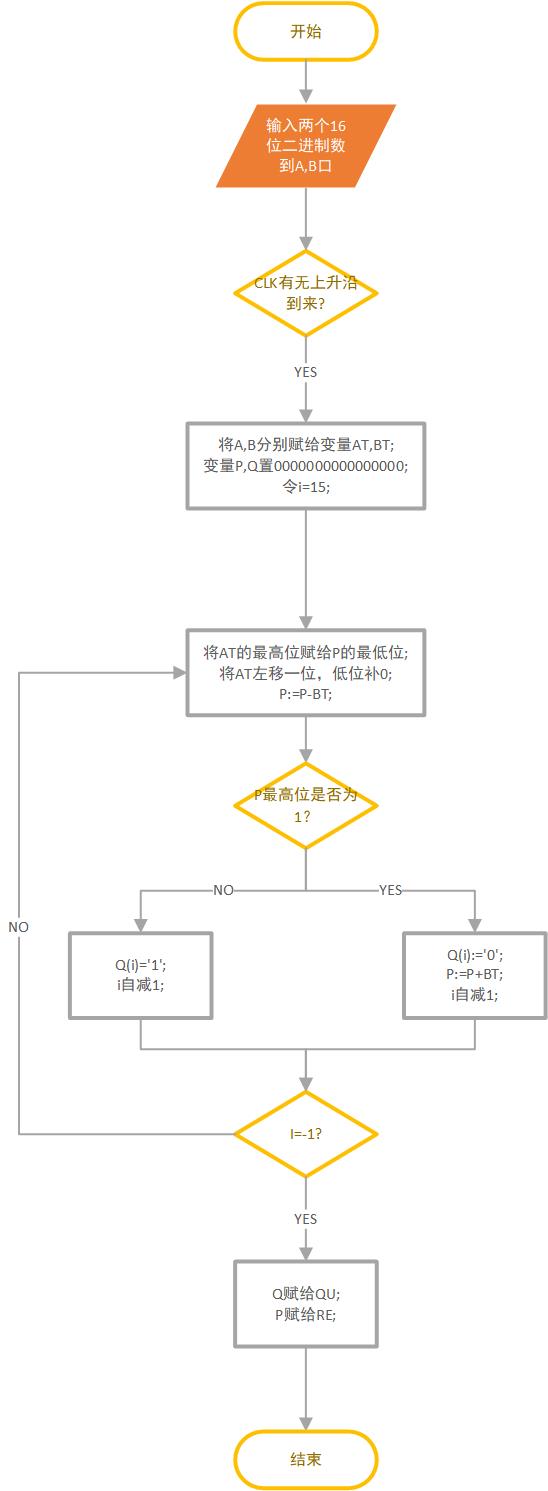
（2）正确结果在选通端为0后的第八个时钟上升沿到来时出现，如波形仿真图所示，后续的结果并无意义

（3）该器件的缺点在于无法达到结果时自动停止，如果增加一个计数保持功能的子模块，则可能可以实现

（4）要注意该设计出于设计方便性考虑，一开始必须让LD为1一段时间才能使数据进入运算过程，

### 【第二个实验】

1.实验原理



该程序模拟人手算乘法的过程，从被除数的高一位除起，若高一位即比除数大，那么商中的最高位就是1，若比除数小，则商中最高位为0，再从高两位除，与高一位的操作相似，依此类推得到商中的各位。

余数的得到方法是，若高x位时除被除数商0，则将高x位原封不动的保存在P中，若高x位时商1了，则将高x位与除数的差保留在P中，下一步将P中数据左移一位，最低位补成高第x+1位，对新的P进行相同的处理，依此类推，最后留在P中的就是余数

2.设计模块代码（Design Block）

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY DIV16\_A IS

PORT(CLK : IN STD\_LOGIC;

A,B : IN STD\_LOGIC\_VECTOR(15 DOWNTO 0);

QU,RE: OUT STD\_LOGIC\_VECTOR(15 DOWNTO 0)

);

END DIV16\_A;

ARCHITECTURE BHV OF DIV16\_A IS

BEGIN

PROCESS(CLK)

VARIABLE AT,BT,P,Q:STD\_LOGIC\_VECTOR(15 DOWNTO 0);

BEGIN

IF RISING\_EDGE(CLK) THEN

AT:=A; BT:=B;

P:="0000000000000000";

Q:="0000000000000000";

FOR I IN QU'RANGE LOOP

P := P(14 DOWNTO 0)&AT(15);

AT:= AT(14 DOWNTO 0)&'0';

P := P-BT;

IF P(15)='1' THEN

Q(I):='0';

p := P+BT;

ELSE Q(I):='1';

END IF;

END LOOP;

END IF;

QU<=Q;

RE<=P;

END PROCESS;

END BHV;

3.激励模块代码（Test Bench）

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

ENTITY DIV16\_A\_vhd\_tst IS

END DIV16\_A\_vhd\_tst;

ARCHITECTURE DIV16\_A\_arch OF DIV16\_A\_vhd\_tst IS

SIGNAL A1 : STD\_LOGIC\_VECTOR(15 DOWNTO 0);

SIGNAL B1 : STD\_LOGIC\_VECTOR(15 DOWNTO 0);

SIGNAL CLK1 : STD\_LOGIC;

SIGNAL QU1 : STD\_LOGIC\_VECTOR(15 DOWNTO 0);

SIGNAL RE1 : STD\_LOGIC\_VECTOR(15 DOWNTO 0);

COMPONENT DIV16\_A

PORT (

A : IN STD\_LOGIC\_VECTOR(15 DOWNTO 0);

B : IN STD\_LOGIC\_VECTOR(15 DOWNTO 0);

CLK : IN STD\_LOGIC;

QU : OUT STD\_LOGIC\_VECTOR(15 DOWNTO 0);

RE : OUT STD\_LOGIC\_VECTOR(15 DOWNTO 0)

);

END COMPONENT;

BEGIN

i1 : DIV16\_A

PORT MAP (

A => A1,

B => B1,

CLK => CLK1,

QU => QU1,

RE => RE1

);

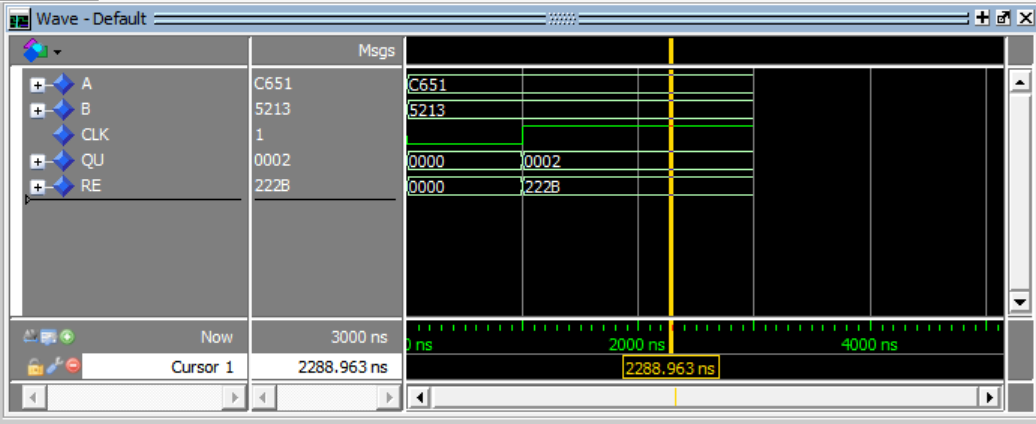
CLK1 <= '0', '1' AFTER 1000 NS;

A1 <= "1100011001010001";

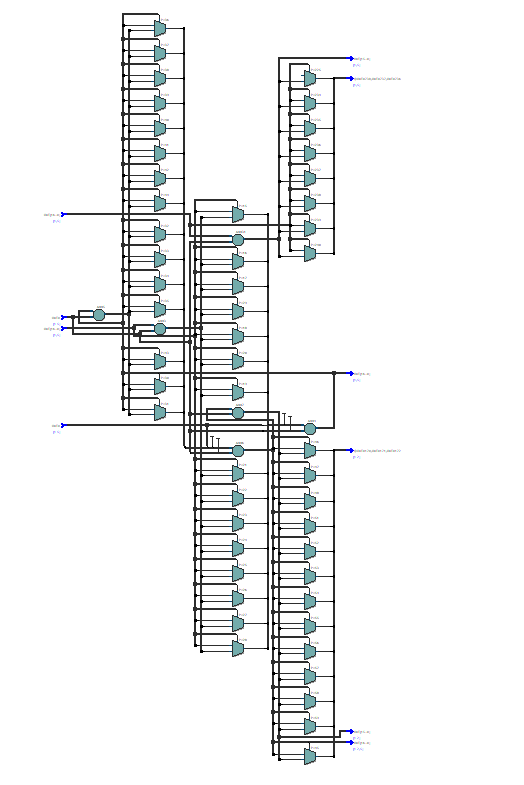
B1 <= "0101001000010011";

END DIV16\_A\_arch;

4.仿真波形图



5.门级电路图

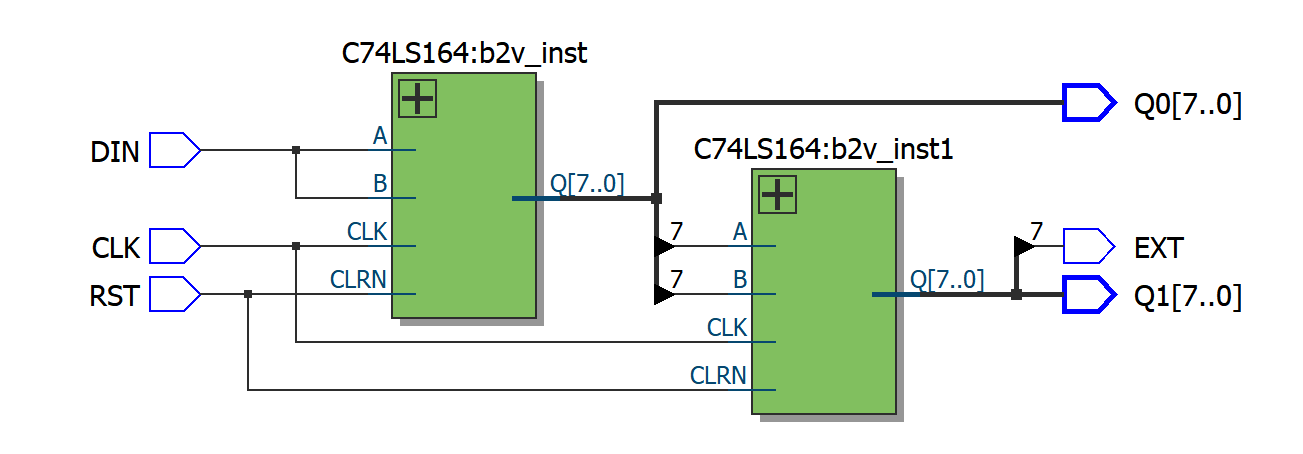


6.结果分析与思考

1. 从图中可见，到来一个时钟沿时除法器开始运算，得出的结果是正确的，5213\*2+222B正好等于C651
2. 该除法器高速的原理在于模拟了人手算除法的过程，有多少位就需要循环多少次
3. 虽然实现了高速除法，但该除法器对硬件资源占用较大

### 【第三个实验】

1.实验原理



如图所示为一个两数码管显示控制电路，RST为高电平清零端，CLK为时钟接口在DIN口串行输入一个八位数据以后，经过八个时钟信号被锁存于Q0口，再经过8个时钟信号即移入Q1中，以此类推，在Q0，Q1口连接译码器以及数码管即可完成数字显示，EXT口为扩展口，用于连接下一个芯片

2.设计模块代码（Design Block）

74LS164设计：

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

LIBRARY work;

ENTITY C74LS164 IS

PORT

(

CLK : IN STD\_LOGIC;

CLRN : IN STD\_LOGIC;

B : IN STD\_LOGIC;

A : IN STD\_LOGIC;

Q : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END C74LS164;

ARCHITECTURE bdf\_type OF C74LS164 IS

SIGNAL DFF\_9 : STD\_LOGIC;

SIGNAL SYNTHESIZED\_WIRE\_0 : STD\_LOGIC;

SIGNAL DFF\_3 : STD\_LOGIC;

SIGNAL DFF\_4 : STD\_LOGIC;

SIGNAL DFF\_5 : STD\_LOGIC;

SIGNAL DFF\_6 : STD\_LOGIC;

SIGNAL DFF\_7 : STD\_LOGIC;

SIGNAL DFF\_8 : STD\_LOGIC;

BEGIN

Q(6) <= DFF\_9;

Q(5) <= DFF\_8;

Q(4) <= DFF\_7;

Q(3) <= DFF\_6;

Q(2) <= DFF\_5;

Q(1) <= DFF\_4;

Q(0) <= DFF\_3;

PROCESS(CLK,CLRN)

BEGIN

IF (CLRN = '1') THEN

Q(7) <= '0';

ELSIF (RISING\_EDGE(CLK)) THEN

Q(7) <= DFF\_9;

END IF;

END PROCESS;

SYNTHESIZED\_WIRE\_0 <= A AND B;

PROCESS(CLK,CLRN)

BEGIN

IF (CLRN = '1') THEN

DFF\_3 <= '0';

ELSIF (RISING\_EDGE(CLK)) THEN

DFF\_3 <= SYNTHESIZED\_WIRE\_0;

END IF;

END PROCESS;

PROCESS(CLK,CLRN)

BEGIN

IF (CLRN = '1') THEN

DFF\_4 <= '0';

ELSIF (RISING\_EDGE(CLK)) THEN

DFF\_4 <= DFF\_3;

END IF;

END PROCESS;

PROCESS(CLK,CLRN)

BEGIN

IF (CLRN = '1') THEN

DFF\_5 <= '0';

ELSIF (RISING\_EDGE(CLK)) THEN

DFF\_5 <= DFF\_4;

END IF;

END PROCESS;

PROCESS(CLK,CLRN)

BEGIN

IF (CLRN = '1') THEN

DFF\_6 <= '0';

ELSIF (RISING\_EDGE(CLK)) THEN

DFF\_6 <= DFF\_5;

END IF;

END PROCESS;

PROCESS(CLK,CLRN)

BEGIN

IF (CLRN = '1') THEN

DFF\_7 <= '0';

ELSIF (RISING\_EDGE(CLK)) THEN

DFF\_7 <= DFF\_6;

END IF;

END PROCESS;

PROCESS(CLK,CLRN)

BEGIN

IF (CLRN = '1') THEN

DFF\_8 <= '0';

ELSIF (RISING\_EDGE(CLK)) THEN

DFF\_8 <= DFF\_7;

END IF;

END PROCESS;

PROCESS(CLK,CLRN)

BEGIN

IF (CLRN = '1') THEN

DFF\_9 <= '0';

ELSIF (RISING\_EDGE(CLK)) THEN

DFF\_9 <= DFF\_8;

END IF;

END PROCESS;

END bdf\_type;

总设计：

LIBRARY ieee;

USE ieee.std\_logic\_1164.all;

ENTITY DISCTR\_B IS

PORT

(

DIN : IN STD\_LOGIC;

CLK : IN STD\_LOGIC;

RST : IN STD\_LOGIC;

EXT : OUT STD\_LOGIC;

Q0 : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0);

Q1 : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END DISCTR\_B;

ARCHITECTURE bdf\_type OF DISCTR\_B IS

COMPONENT C74LS164

PORT(A : IN STD\_LOGIC;

B : IN STD\_LOGIC;

CLRN: IN STD\_LOGIC;

CLK : IN STD\_LOGIC;

Q : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0));

END COMPONENT;

SIGNAL Q\_ALTERA\_SYNTHESIZED0 : STD\_LOGIC\_VECTOR(7 DOWNTO 0);

SIGNAL Q\_ALTERA\_SYNTHESIZED1 : STD\_LOGIC\_VECTOR(7 DOWNTO 0);

BEGIN

b2v\_inst : C74LS164

PORT MAP(A => DIN,

B => DIN,

CLRN => RST,

CLK => CLK,

Q => Q\_ALTERA\_SYNTHESIZED0);

b2v\_inst1 : C74LS164

PORT MAP(A => Q\_ALTERA\_SYNTHESIZED0(7),

B => Q\_ALTERA\_SYNTHESIZED0(7),

CLRN => RST,

CLK => CLK,

Q => Q\_ALTERA\_SYNTHESIZED1);

EXT <= Q\_ALTERA\_SYNTHESIZED1(7);

Q0 <= Q\_ALTERA\_SYNTHESIZED0;

Q1 <= Q\_ALTERA\_SYNTHESIZED1;

END bdf\_type;

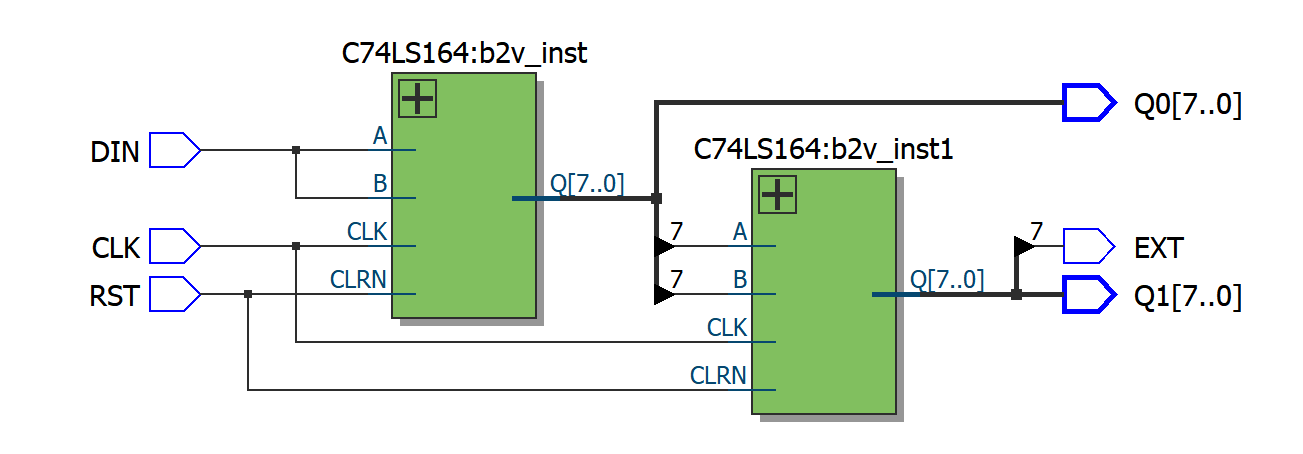
3.激励模块代码（Test Bench）

由于缺少硬件数码管，无法进行仿真测试

4.仿真波形图

由于缺少硬件数码管，无法进行仿真测试

5.门级电路图



6.结果分析与思考

若加以合适的外部电路，如单片机，译码器，数码管等则可以实现串行显示