**数字逻辑与部件设计实验报告**

姓名：

学号：

专业：计算机科学与技术

指导老师：唐志强

日期：2016年12月12日

**目录**

实验一 译码器和编码器 3

实验二 七段显示译码器的设计 5

实验三 加法器及快速进位电路的设计 7

实验四 算术逻辑单元的设计 10

实验五 触发器和寄存器 13

实验六 计数器的设计 16

实验七 有限状态机 18

**实验一 译码器和编码器**

1. 实验目的

1.用实验实现《数字逻辑与部件设计》课程中的相关设计，加深对实验的理解并增强实验能力。

2.熟悉和掌握Vivado软件中创建工程、模拟电路、综合电路和生成bit文件等功能。

3.初步掌握用Verilog语言进行数字电路设计的技能。

4.认识和了解译码器和编码器在计算机系统中的功能。

5.设计具有特定功能的译码器和编码器。

1. 实验内容

1.用与非门设计一个3-8译码器74LS138，除了上述真值表中的输入输出信号外，增加三个控制信号G1、G2A、G2B，当G1为高电平，G2A和G2B同时为低电平时，实现如表2-1的正常译码，否则Y0---Y7都为高电平。

3-8译码器有3个输入S0、S1、S2,8个输出Y0、Y1、Y2、Y3、Y4、Y5、

Y6、Y7。3-8译码器的真值表如2-1所示。

表2-1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输入 | | | 输出 | | | | | | | |
| S2 | S1 | S0 | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

考虑到有G1、G2A、G2B这三个信号，当（G1&~G2A&~G2B）为真时，Y0---Y7

的值由S0、S1、S2决定，否则Y0---Y7的值都为高电平。当Y0---Y7的值由S0、S1、S2决定时，每个输出都只在一种情况下为0，其它情况下都为1，所以令每个输出等于它对应情况下三个输入布尔值乘积的反，例如Y0的值为~(~S2&~S1&~S0)，那么当S0、S1、S2都为0时，Y0也为0，而在其它情况下，Y0为1，其它的输出也是类似的，此时各输出表达式如下：

Y[0] = S[0]|S[1]|S[2]

Y[1] = ~S[0]|S[1]|S[2]

Y[2] = S[0]|~S[1]|S[2]

Y[3] = ~S[0]|~S[1]|S[2]

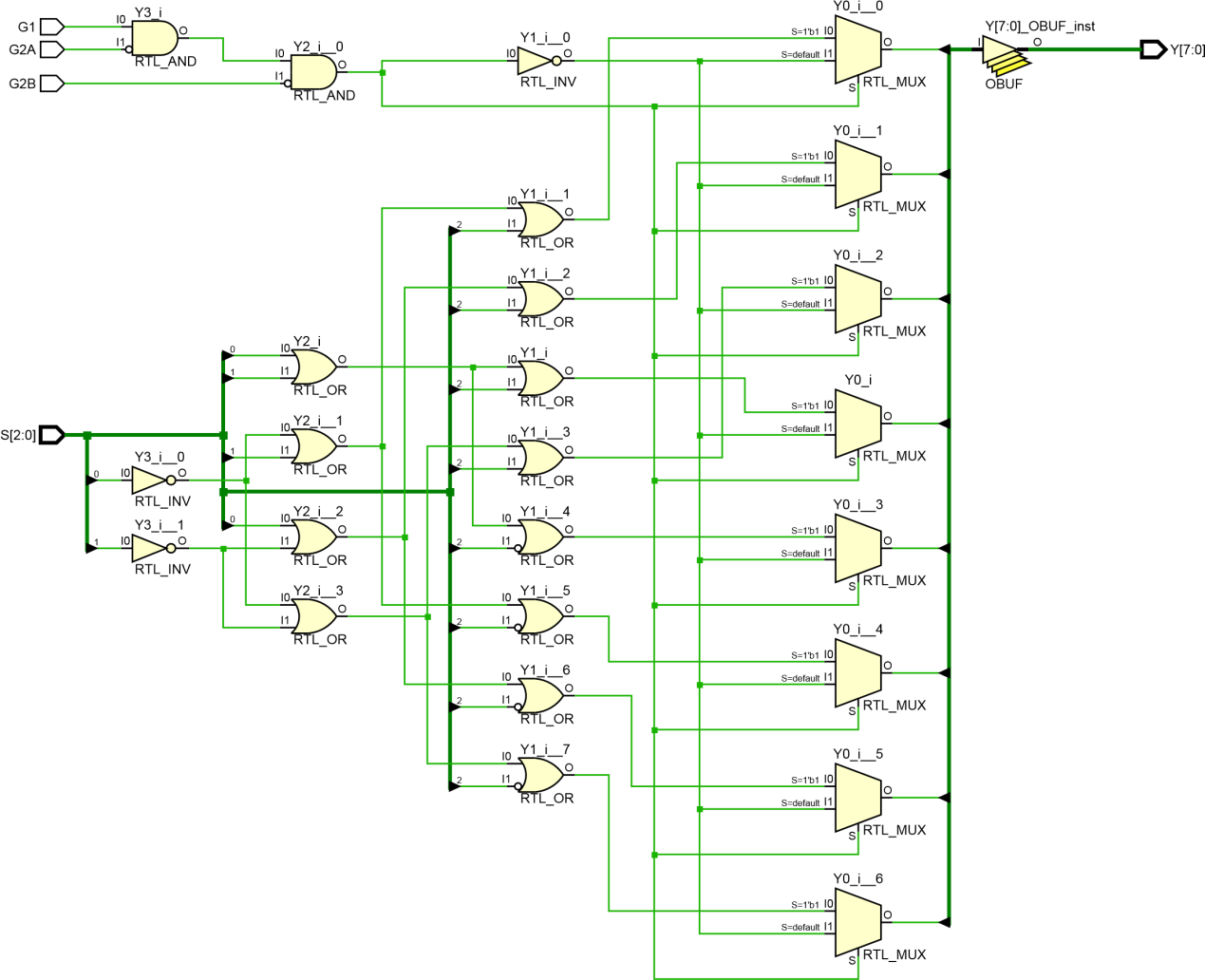
Y[4] = S[0]|S[1]|~S[2]

Y[5] = ~S[0]|S[1]|~S[2]

Y[6] = S[0]|~S[1]|~S[2]

Y[7] = ~S[0]|~S[1]|~S[2]

在Vivado中生成的RTL结构图如下：



1. 实验结论

在译码器中，三个输入变量表示0-7这8个数字，译码器将输入中的8个数字分别对应到8个输出中，当输入为某个数字时，其对应的输出为1，其余输出为0。

**实验二** **七段译码显示器的设计**

1. 实验目的

1.用实验实现《数字逻辑与部件设计》课程中的相关设计，加深对知识的理解并增强实验能力。

2.熟悉和掌握Vivado软件中创建工程、模拟电路、综合电路和生成bit文

件等功能。

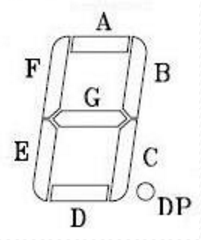
3.初步掌握用Verilog语言进行数字电路设计的技能。

4.完成七段译码显示器的设计，使其能显示数字0-9。

1. 实验内容
2. 设计一个BCD-7段译码显示器。

七段数码管的7个发光二极管分别由a-g的7个输入控制，如图3-1：

图3-1



译码器的输入为BCD码，输出为7段控制信号，真值表见表3-2所示：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输入 | | | | 输出 | | | | | | |
| D | C | B | A | a | b | c | d | E | f | g |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | X | X | X | X | X | X | X |
| 1 | 0 | 1 | 1 | X | X | X | X | X | X | X |
| 1 | 1 | 0 | 0 | X | X | X | X | X | X | X |
| 1 | 1 | 0 | 1 | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 0 | X | X | X | X | X | X | X |
| 1 | 1 | 1 | 1 | X | X | X | X | X | X | X |

设计显示器时，只有4个输入，产生7个输出，控制7个发光二极管的开关，一个需要注意的地方是位和段为0才有效，比如a为0时，其对应的发光二极管才会亮，需要让某个显示器显示时，也需要使对应的An为0，化简后的各输出表达式如下：

a =((~D)&(~C)&(~B)&(A))|((~D)&(C)&(~B)&(~A))

b =((~D)&(C)&(~B)&(A))|((~D)&(C)&(B)&(~A))

c =((~D)&(~C)&(B)&(~A))

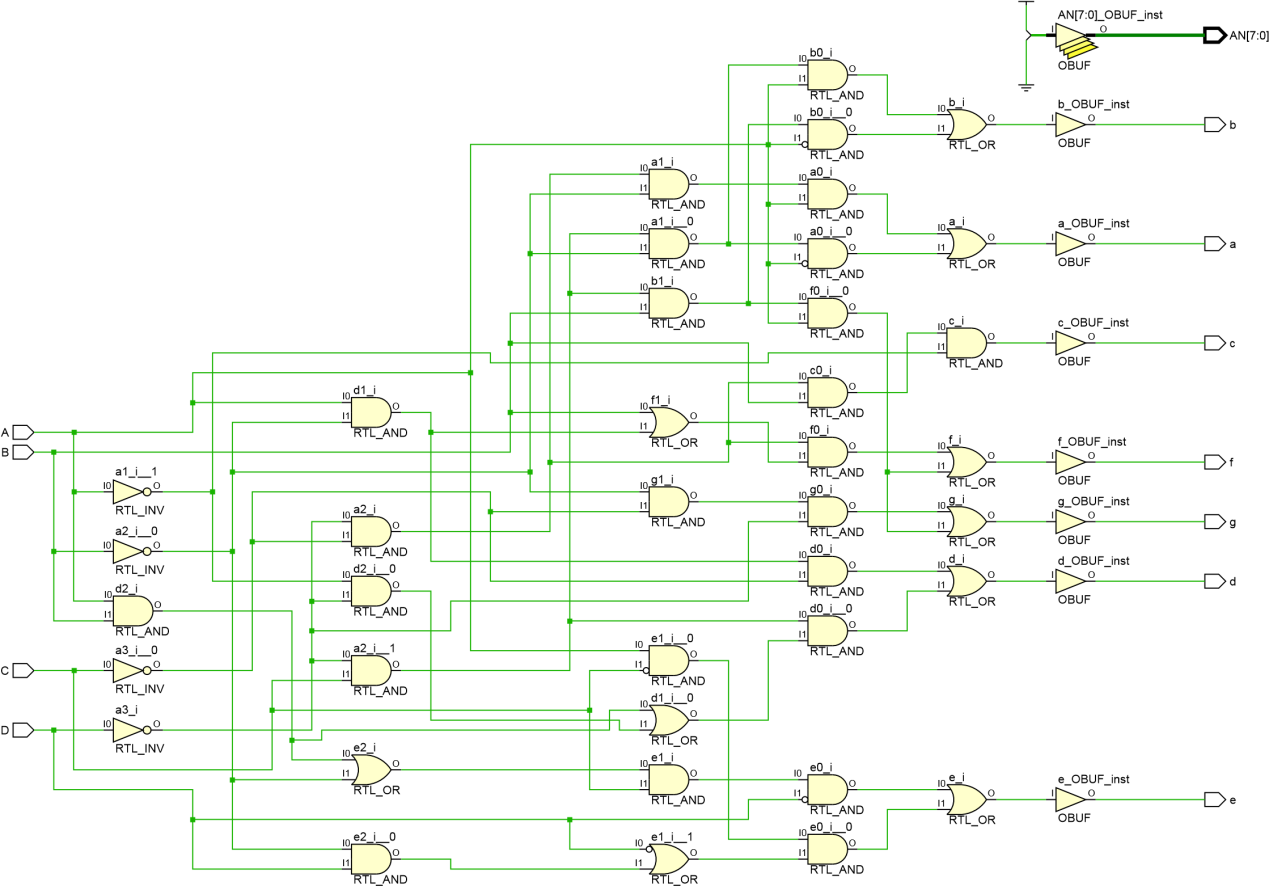
d =(A&(~B)&(~C))|(C&(~D)&((A&B)|((~A)&(~D))))

e =(((A&B)|(~B))&C&(~D))|((A&~C)&(~D|(~B&D)))

f =((~C&~D)&(B|~B&A))|((~D)&(C)&(B)&(A))

g =(~B&~C&~D)|((~D)&(C)&(B)&(A))

在Vivado中生成的RTL结构图如下：



1. 实验结论

BCD-7段译码显示器的4个输入为BCD码，可输入0-9这10个数字，每个数字在显示器中有其对应的若干个发光二极管，当输入为某个数字时，其对应的发光二极管发光，从而显示出这个数字的数码排列。

**实验三 加法器及快速进位电路的设计**

1. 实验目的

1.用实验实现《数字逻辑与部件设计》课程中的相关设计，加深对知识的理解并增强实验能力。

2.熟悉和掌握Vivado软件中创建工程、模拟电路、综合电路和生成bit文件等功能。

3.初步掌握用Verilog语言进行数字电路设计的技能。

4.了解一位全加器的功能和作用。

5.学会在Verilog语言中设计不同的模块并进行拼装，体会模块化和分而治之的工程思想。

6.以一位全加器为模块设计串行4位全加器和包含快速进位电路的4位全加器。

1. 实验内容

1.用与非门和反相器设计一个一位全加器

一个一位全加器电路的输入信号为：本位输入A、B，低位进位信号C；输出为本位输出S，进位输出C0。

一位全家器的真值表如表4-1所示。

表4-1

|  |  |
| --- | --- |
| A B C | S CO |
| 0 0 0 | 0 0 |
| 0 0 1 | 1 0 |
| 0 1 0 | 1 0 |
| 0 1 1 | 0 1 |
| 1 0 0 | 1 0 |
| 1 0 1 | 0 1 |
| 1 1 0 | 0 1 |
| 1 1 1 | 1 1 |

首先可得出S和C0关于A、B和C的表达式：

S=~A&~B&C+~A&B&~C+A&~B&~C+A&B&C

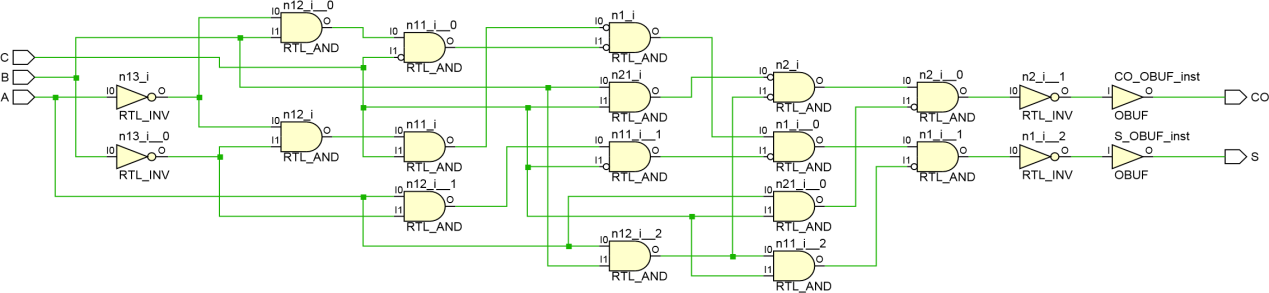
CO=A&B+B&C+A&C

由于需要用与非门和反相器来设计，所以将表达式转化为：

S=~(~(~A&~B&C)&~(~A&B&~C)&~(A&~B&~C)&~(A&B&C))

CO= ~(~(A&B)&~(B&C)&~(A&C))

在Vivado中生成的RTL结构图如下：



2.以上述一位全加器为模块，设计一个串行4位全加器

串行4位全加器需要4个一位的全加器，输入为A[3：0],B[3：0],C,输出为进位CO和本位输出S[3：0],其中每个一位全加器使A和B的对应位相加，最低位的一位全加器的进位信号为C，其余3个一位全加器的进位信号为低一位的一位全加器的进位输出，每个全加器将本位输出输出到S，最高位的一位全加器的进位输出为CO，其余进位输出输出到高一位的一位全加器的进位信号。

在Vivado中生成的RTL结构图如下：

lab3

lab3\_2\_1\_HD2

A

B

C

CO

S

lab1

lab3\_2\_1

A

B

C

CO

S

lab4

lab3\_2\_1\_HD3

A

B

C

CO

S

lab2

lab3\_2\_1\_HD1

A

B

C

CO

S

a[3:0]

b[3:0]

cc

s[4:0]

s[4:0]\_OBUF\_inst

OBUF

O

I

0

1

2

3

0

1

2

3

3.设计一个包含快速进位电路的4位全加器

一个包含快速进位电路的4位全加器的输入有a[3:0],b[3:0],进位信号cc,输出为进位信号s[4]和本位输出s[3:0]。

由于需要快速进位，首先确定4个一位全加器的进位，分别记为s[4],c[2],c[1]和c[0]，为方便计算，引入中间变量p[3:0]和g[3:0],其表达式计算如下：

p[0]=(a[0]&~b[0])|(~a[0]&b[0])

p[1]=(a[1]&~b[1])|(~a[1]&b[1])

p[2]=(a[2]&~b[2])|(~a[2]&b[2])

p[3]=(a[3]&~b[3])|(~a[3]&b[3])

g[0]=a[0]&b[0]

g[1]=a[1]&b[1]

g[2]=a[2]&b[2]

g[3]=a[3]&b[3]

从而得出s[4],c[2],c[1]和c[0]的表达式：

c[0]=g[0]|(p[0]&cc)

c[1]=g[1]|(p[1]&g[0])|(p[1]&p[0]&cc)

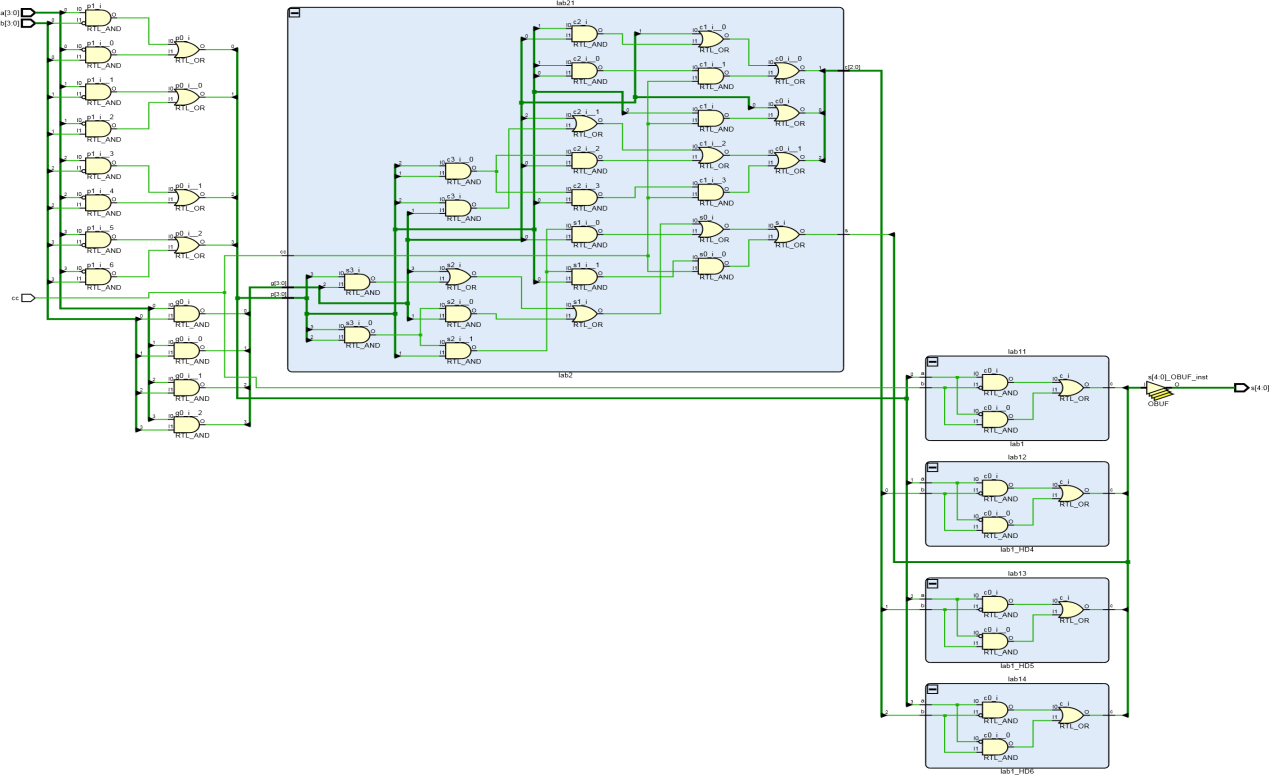
c[2]=g[2]|(p[2]&g[1])|(p[2]&p[1]&g[0])|(p[2]&p[1]&p[0]&cc)

s=g[3]|(p[3]&g[2])|(p[3]&p[2]&g[1])|

(p[3]&p[2]&p[1]&g[0])|(p[3]&p[2]&p[1]&p[0]&cc)

得到4位进位后，只需将每个一位全加器对应的a,b和进位c相加，即可得到本位输出s。

在Vivado中生成的RTL结构图如下：



三、实验结论

1.一位全加器可将2个输出和1个低位进位输入相加，得到本位输出S和进位输出C0。

2.在Verilog语言中可以组建多个模块，然后在顶层模块中对底层模块进行拼装。

3.将4个一位全加器进行串行地接口和拼装，可得到一个串行4位全加器。

4.将串行4位全加器中每个一位全加器的进位预先计算好之后，可直接得到4个进位和4个本位输出，从而得到能够快速进位的4位全加器。

**实验四 算术逻辑单元的设计**

1. 实验目的

1.用实验实现《数字逻辑与部件设计》课程中的相关设计，加深对知识的理解并增强实验能力。

2.熟悉和掌握Vivado软件中创建工程、模拟电路、综合电路和生成bit文件等功能。

3.初步掌握用Verilog语言进行数字电路设计的技能。

4.了解算术逻辑单元在CPU中的作用并进行设计。

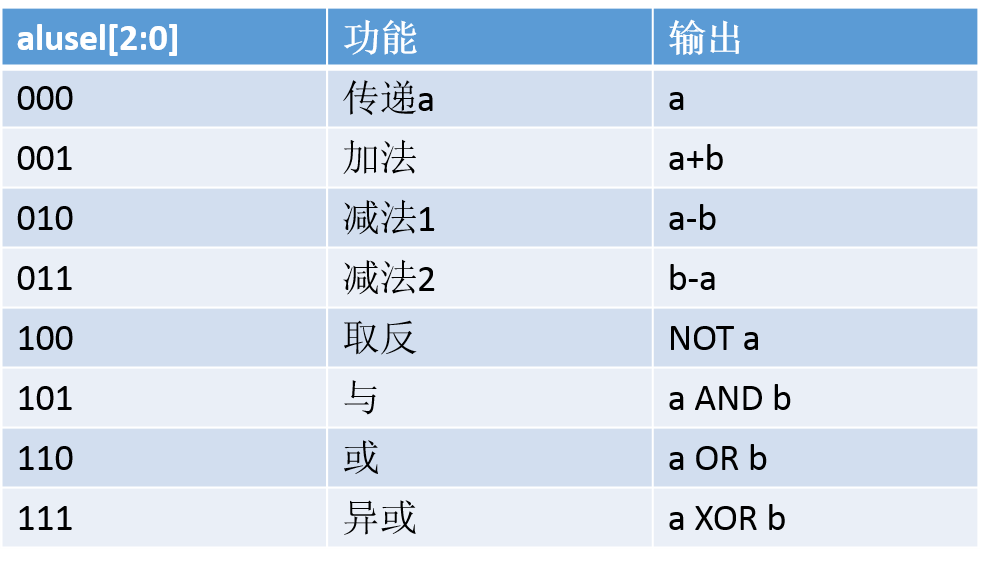
5.了解寄存器传输的原理并进行设计寄存器传输的实验。

1. 实验内容

1.设计一个4位算术逻辑单元，输入信号为：两组4位数据信号（a3-

a0，b3-b0）；数据输出信号为：4位数据信号（y0-y3）,负标志nf（最高位为1）,零标志zf,溢出标志ovf,进位标志cf。3位功能控制信号f[2:0]。

表5-2 算术逻辑单元的功能



有功能控制信号f，首先判别f的值，然后再进行与f的值对应的逻辑运算

和算术运算。其中输出y在逻辑运算和算术运算中的表达式如下：

输出：

a ⬄ y=a

a+b ⬄ {cf,y}=a+b

a-b ⬄ {cf,y}=a-b

b-a ⬄ {cf,y}=b-a

NOT a ⬄ y=~a

a AND b ⬄ y=a&b

a OR b ⬄ y=a|b

a XOR b ⬄ y=a^b

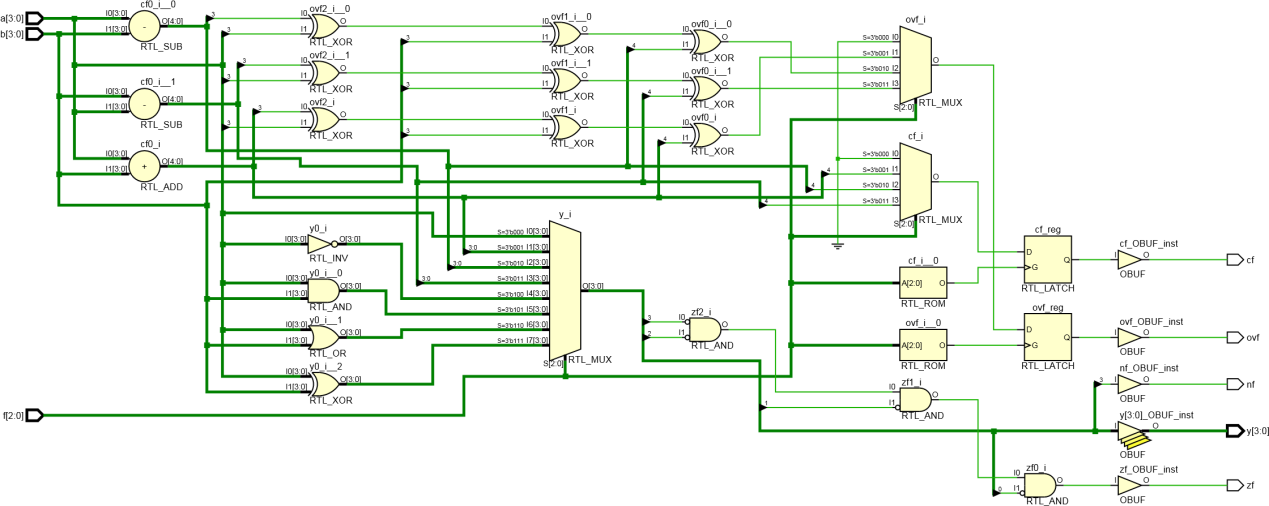
另外，各标志的表达式如下：

nf=y[3]

zf=~y[3]&~y[2]&~y[1]&~y[0]

ovf=(((y[3]^a[3])^b[3])^cf)

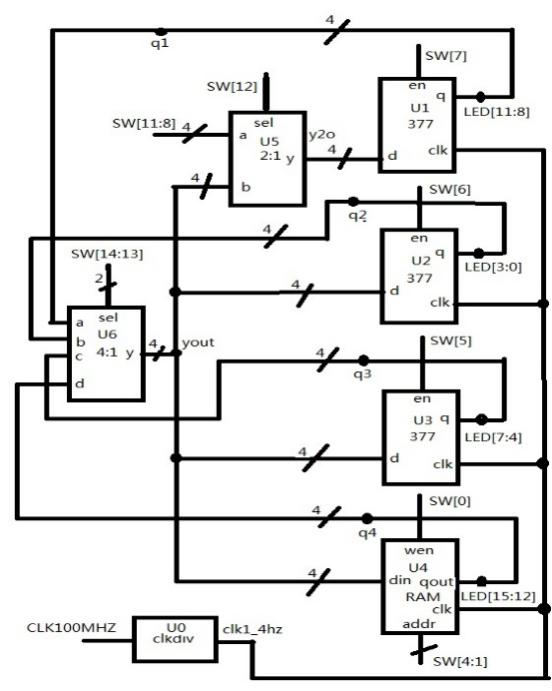
在Vivado中生成的RTL结构图如下：



1. 寄存器交换实验

系统由三个74377寄存器（U1,U2,U3），一片RAM（U4）一个2:1多路器(U5),一个4：1多路器(U6)及时钟分频电路（U0）组成，按附图要求，用Verilog结构模型描述上述系统。

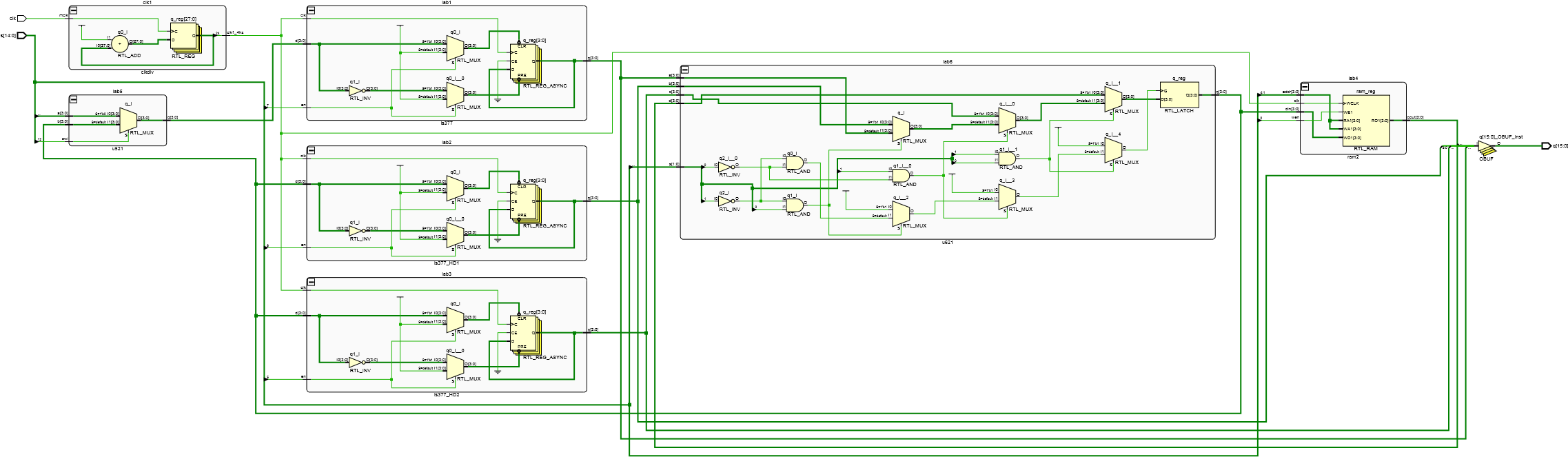
附图



首先确定这个系统的各模块U0-U6，其中4位寄存器U1,U2,U3在使能为1

时，在每个时钟上升沿将输入传送到输出，否则输出不变；U4为RAM；4位2:1多路器U5根据一位控制信号在2个4位输入中选择一个进行输出；4位4：1多路器U6根据2位控制信号在4个4位输入中选择一个进行输出；U0为时钟分频电路。然后在顶层模块中将U0-U6进行组装即可达到寄存器传输的功能。

在Vivado中生成的RTL结构图如下：



1. 实验结论
2. 算术逻辑单元是CPU的核心部件，实现算术及逻辑运算。
3. 用三个74377寄存器（U1,U2,U3），一片RAM（U4）一个2:1多路器(U5),一个4：1多路器(U6)及时钟分频电路（U0）可将数据传送到寄存器中并能将两个寄存器中的数据进行交换。

**实验五 触发器和寄存器**

1. 实验目的

1.用实验实现《数字逻辑与部件设计》课程中的相关设计，加深对知识的理解并增强实验能力。

2.熟悉和掌握Vivado软件中创建工程、模拟电路、综合电路和生成bit文件等功能。

3.初步掌握用Verilog语言进行数字电路设计的技能。

4.了解触发器和寄存器的作用和功能。

5.编译并实现具有特定功能的触发器和寄存器。

1. 实验内容
2. 用Verilog语言描述带异步清零功能的D触发器、JK触发器和T触发

器，编译并实现其逻辑功能。

D触发器有输入端D，考虑到需要异步清零，对上升沿触发的D触发

器，若清零信号CLR在低水平有效时，输出Q=0，否则在时钟CLK的上升沿输出Q=D，其余时刻Q不变。

在Vivado中生成的RTL结构图如下：

clk1

clkdiv

clka

mclk

unq\_i

RTL\_INV

I0

O

q\_reg[28:0]

RTL\_REG

C

D

Q

+

q0\_i

RTL\_ADD

I1

I0[28:0]

O[28:0]

clr

mclk

Q

q\_OBUF\_inst

OBUF

O

I

q\_reg

RTL\_REG\_ASYNC

C

CLR

D

Q

~Q

unq\_OBUF\_inst

OBUF

O

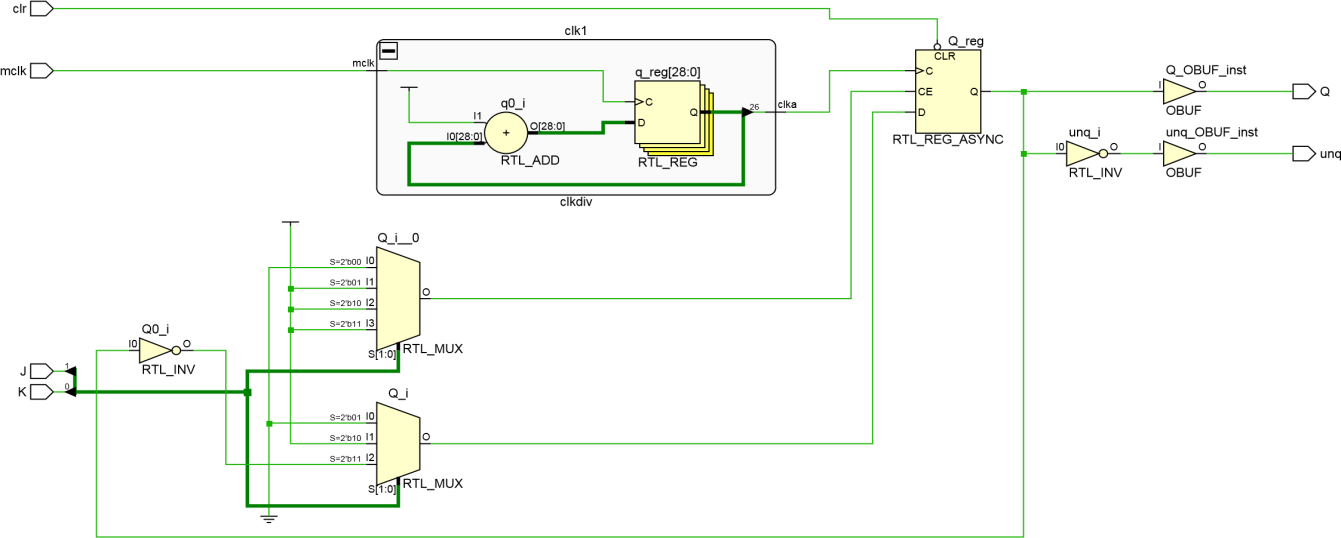
I

D

26

而JK触发器有两个输入端J和K，J相当于置0端，K相当于置1端，考虑到需要异步清零，对上升沿触发的JK触发器，若清零信号CLR在低水平有效时，输出Q=0，否则在时钟CLK的上升沿检测J和K的值：若J=0，K=0，Q保持不变；若J=0，K=1，Q变为0；若J=1，K=0，Q变为1；若J=1，K=1，Q的值翻转。

在Vivado中生成的RTL结构图如下：



T触发器有输入端T，考虑到需要异步清零，对上升沿触发的T触发器，若清零信号CLR在低水平有效时，输出Q=0，否则在时钟CLK的上升沿检测T的值：若T=1，Q的值翻转；若T=0，Q的值不变。

在Vivado中生成的RTL结构图如下：

clk1

clkdiv

clka

mclk

Q\_OBUF\_inst

OBUF

O

I

Q\_reg

RTL\_REG\_ASYNC

C

CE

CLR

D

Q

Q0\_i

RTL\_INV

I0

O

T

unq\_i

RTL\_INV

I0

O

q\_reg[28:0]

RTL\_REG

C

D

Q

+

q0\_i

RTL\_ADD

I1

I0[28:0]

O[28:0]

clr

mclk

unq

unq\_OBUF\_inst

OBUF

O

I

Q

26

1. 用上述D触发器构成74LS175。

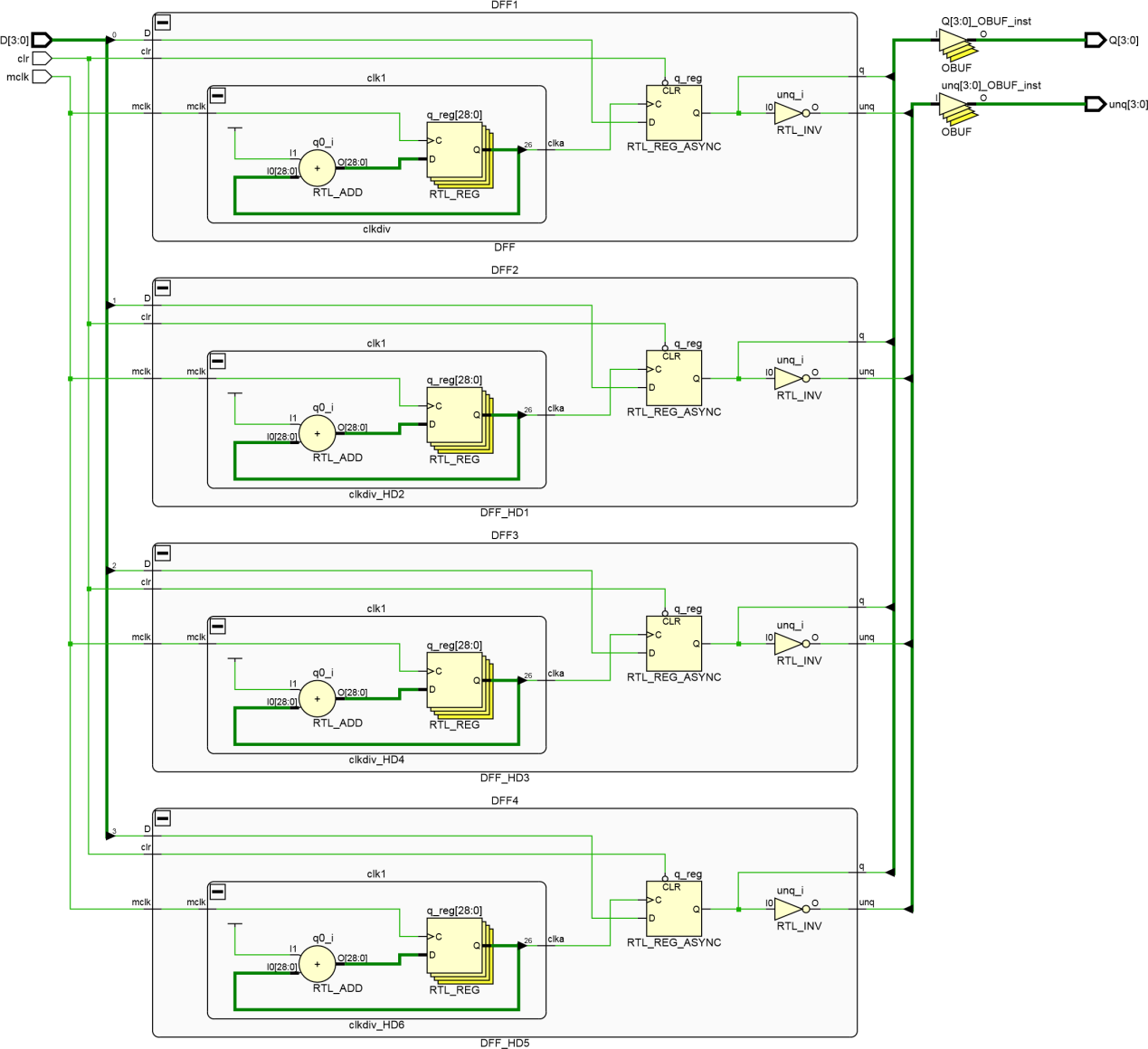
74LS175的功能表

|  |  |
| --- | --- |
| CLR CLK | 工作状态 |
| 0 X | 异步清零 |
| 1 上升沿 | 数据存放 |
| 1 0 | 数据保存 |

4位74LS175由4个D触发器组成，有4位输入D[3：0]，时钟信号

CLK,清零信号CLR，4位输出Q[3:0],分别将一位输入，时钟信号CLK,清零信号CLR，一位输出接到一个D触发器上，用4个D触发器可构成74LS175。

在Vivado中生成的RTL结构图如下：



1. 实验结论

1.带异步清零功能的D触发器在时钟CLK的上升沿输出Q=D，其余时刻Q不变。若清零信号CLR在低水平有效时，可异步清零。

2.带异步清零功能的JK触发器在时钟CLK的上升沿检测J和K的值：若J=0，K=0，Q保持不变；若J=0，K=1，Q变为0；若J=1，K=0，Q变为1；若J=1，K=1，Q的值翻转。若清零信号CLR在低水平有效时，可异步清零。

3.带异步清零功能的T触发器时钟CLK的上升沿检测T的值：若T=1，Q的值翻转；若T=0，Q的值不变。若清零信号CLR在低水平有效时，可异步清零。

4.可用4个1位的D触发器构成一个4位的74LS175。

**实验六 计数器的设计**

一、实验目的

1.用实验实现《数字逻辑与部件设计》课程中的相关设计，加深对知识的理解并增强实验能力。

2.熟悉和掌握Vivado软件中创建工程、模拟电路、综合电路和生成bit文件等功能。

3.初步掌握用Verilog语言进行数字电路设计的技能。

4.了解计数器的概念和作用。

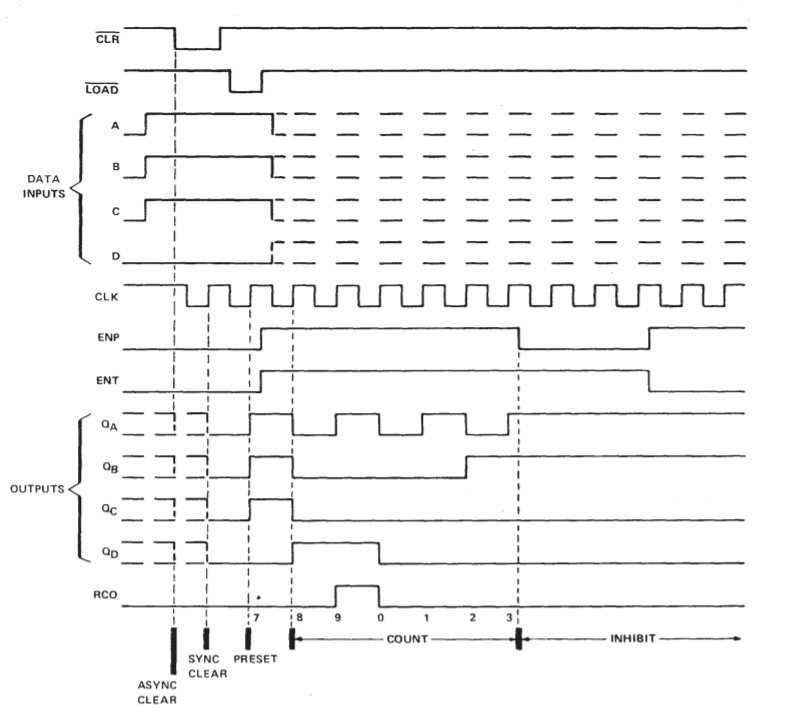
5.设计并实现同步二进制计数器。

6.利用计数器完成时钟的设计。

二、实验内容

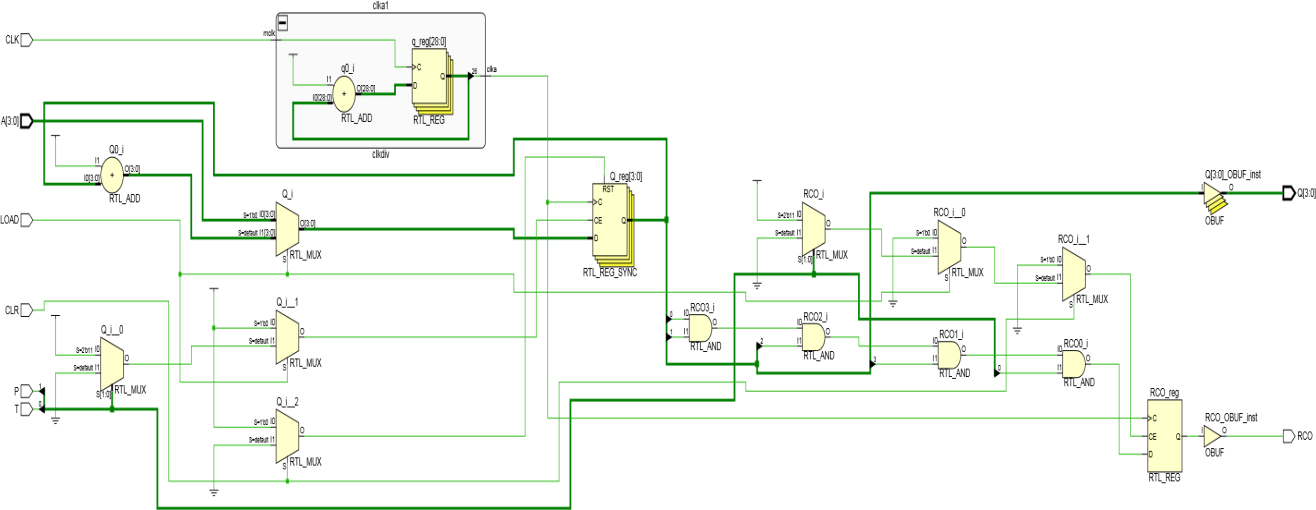
1.设计同步二进制计数器74LS163。

74LS163是一个同步二进制计数器，它有4位数据输入D、C、B、A，4位数据输出QD、QC、QB、QA；同步清零端CLR，同步置数端LOAD，计数允许端P、T，进位输出RCO，时钟信号CLK。74LS163的功能表见图6-1：



由于计数器是同步的，所以在每个时钟上升沿检测同步清零端CLR的值，若CLR=0，则Q=0，若CLR=1，再检测同步置数端LOAD的值，若LOAD=0，则Q=A，否则再检测计数允许端P、T的值，只有当P=1，T=1时才计数，否则Q保持不变。计数时，Q=Q+1，此时若Q[0]&Q[1]&Q[2]&Q[3]&T为1，则进位输出RCO=1，否则RCO=0。

在Vivado中生成的RTL结构图如下：

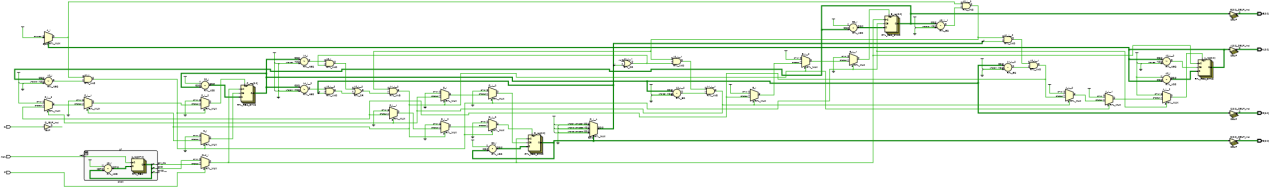


2. 时钟，显示时和分，分别用两个BCD码表示个位和十位，共16位二进制；显示时间范围00：00-23：59，输入时钟1HZ左右。

用A[3：0]表示分钟个位，B[3：0]表示分钟十位，C[3：0]表示时钟个位，D[3：0]表示时钟十位。

在每个时钟上升沿，首先判断A是否等于9，若A不等于9，则A=A+1，否则A=0，B进位。当B进位时，判断B是否等于5，若B不等于5，则B=B+1，否则B=0，C进位。当C进位时，判断C是否等于9，若C等于9，则C=0，D=D+1，否则判断C是否等于3且D等于2，若C等于3且D等于2，则C=0，D=0，否则C=C+1。

在Vivado中生成的RTL结构图如下：



三、实验结论

1.同步二进制计数器74LS163在每个时钟上升沿检测同步清零端CLR，可进行同步清零，同步置数端LOAD，可同步置数，计数允许端P、T，当P=1，T=1时可进行计数，若计数到15时，进位输出RCO=1。

2.可以用计数器的原理设计一个时钟，显示时间范围为00：00-23：59。

实验七 有限状态机

1. 实验目的

1.用实验实现《数字逻辑与部件设计》课程中的相关设计，加深对知识的理解并增强实验能力。

2.熟悉和掌握Vivado软件中创建工程、模拟电路、综合电路和生成bit文件等功能。

3.初步掌握用Verilog语言进行数字电路设计的技能。

4.了解状态机的概念。

5.设计并实现具有特定逻辑功能的状态机。

1. 实验内容

用Verilog语言描述交通控制器及15秒、45秒定时输入信号的逻辑功能，定时输入信号由1秒的信号经分频得到。

交通灯的功能如下：

交通灯位于四个方向（东、南、西、北）

南北和东西向在同一时间亮相同的灯

灯变化的次序为：绿-黄-红，循环

南北和东西向灯不同时为绿灯或黄灯

灯的持续时间：绿-45秒，黄-15秒，红-60秒

设置状态转变过程如下,N代表北，S代表南，E代表东，W代表西：

有控制信号15S，45S和清零信号CLR,若CLR低电平有效时，回到状态一，否则继续状态转变。

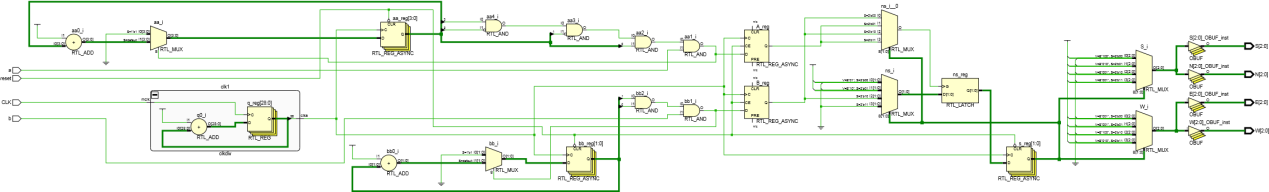
状态转变首先进入状态一：N和S为绿灯，E和W为红灯，若45S信号有效，则进入状态二，否则状态不变。

状态二：N和S为黄灯，E和W为红灯，若15S信号有效，则进入状态三，否则状态不变。

状态三：N和S为红灯，E和W为绿灯，若45S信号有效，则进入状态四，否则状态不变。

状态四：N和S为红灯，E和W为绿灯，若15S信号有效，则进入状态一，否则状态不变。

在Vivado中生成的RTL结构图如下：



1. 实验结论

可利用有限状态机的原理设计红绿灯，将东西南北的红绿灯分成4个状态，若CLR低电平有效时，回到状态一，否则进行状态转变，从而实现东西南北4个方向的红绿灯有序变换颜色，实现交通灯的功能。