به نام خدا

تمرین اول

محمدمهدی آقاجانی

استاد : مرتضی صاحب الزمانی

تمارین اجباری

سوال ۵ :

$$c_1 = 100 * x$$
$$c_2 = 10^6 + x$$
$$if\ c_2 < c_1 \Rightarrow 10^6 + x < 100 * x$$
$$\Rightarrow 10^6 < 99x \Rightarrow x > 10101$$

در نتیجه برای ساخت ۱۰۱۰۱ دستگاه به بالا استفاده از متد ASIC به صرفته تر است.

سوال ۶ :

الف) برای جواب دادن به این سوال باید ابتدا حدسی درباره تعداد فروش داشته باشیم . با توجه به محاسبه بالا و فرض سوال فرض میکنیم میخواهیم ۱۰۰۰۰۰۰ نسخه به فروش برسانیم و همین تعداد تولید کرده ایم :

$$\text{هزینه هر نسخه} = \frac{10^6 + 10^6}{10^6} = 2 \text{ هزار تومان}$$

پس قیمت تمام شده هر محصول را ۲ هزار تومان در نظر میگیریم.(فرض میکنیم سودی بر روی هر دستگاه اضافه نمکنیم) :

$$\text{خسارت} = 2 * 10^6 - 1000 * 2 = 1998000 \text{ هزار تومان}$$

در نتیجه حدود یک میلیارد و نهصد و نود و هشت میلیون تومان ضرر میکنیم که بسیار ضرر عظیمی ست!

ب)

$$c = c_1 - c_2 =$$
$$100 * 10^6 - 2 * 10^6 = 99 * 10^6 \text{ هزار تومان}$$

این اختلاف برابر با ۹۹ میلیارد تومان است!

ج) ابتدا تعداد محدودی را با FPGA تولید میکنیم و اگر بازار رغبت نشان داد و میل به خرید بسیار زیاد شد سپس به سمت ASIC میرویم.

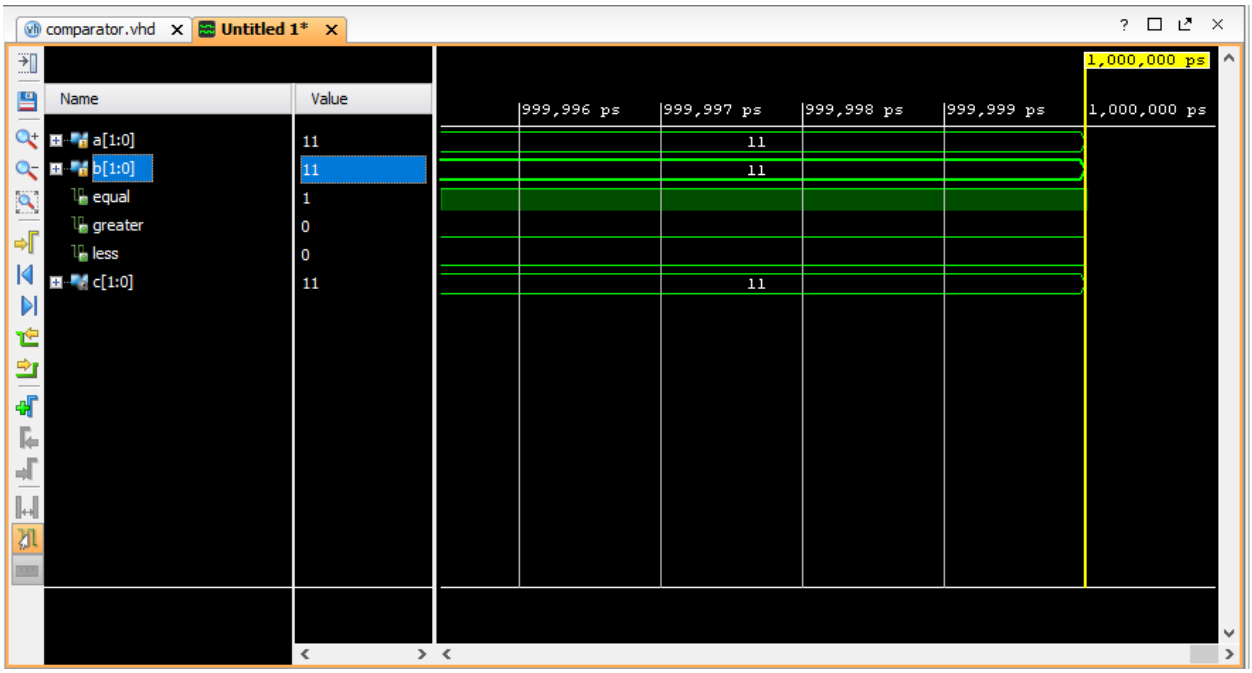
سوال ۷ :

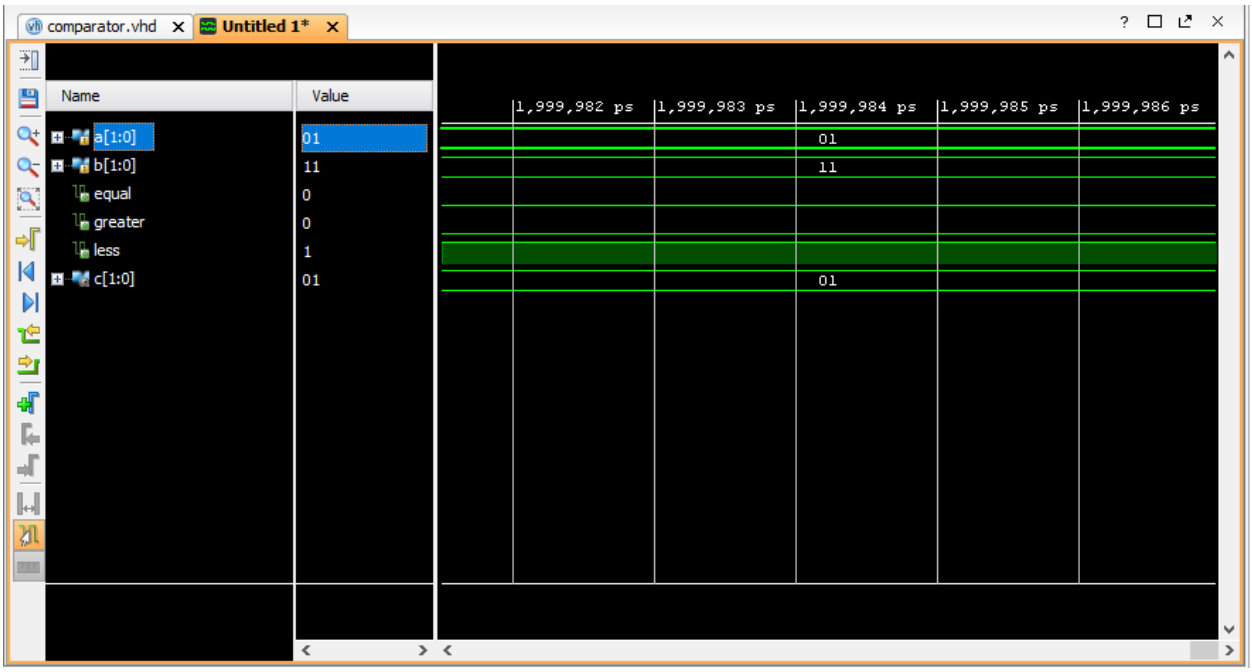
در ابتدا در سطح گیت پیاده سازی میکنیم :

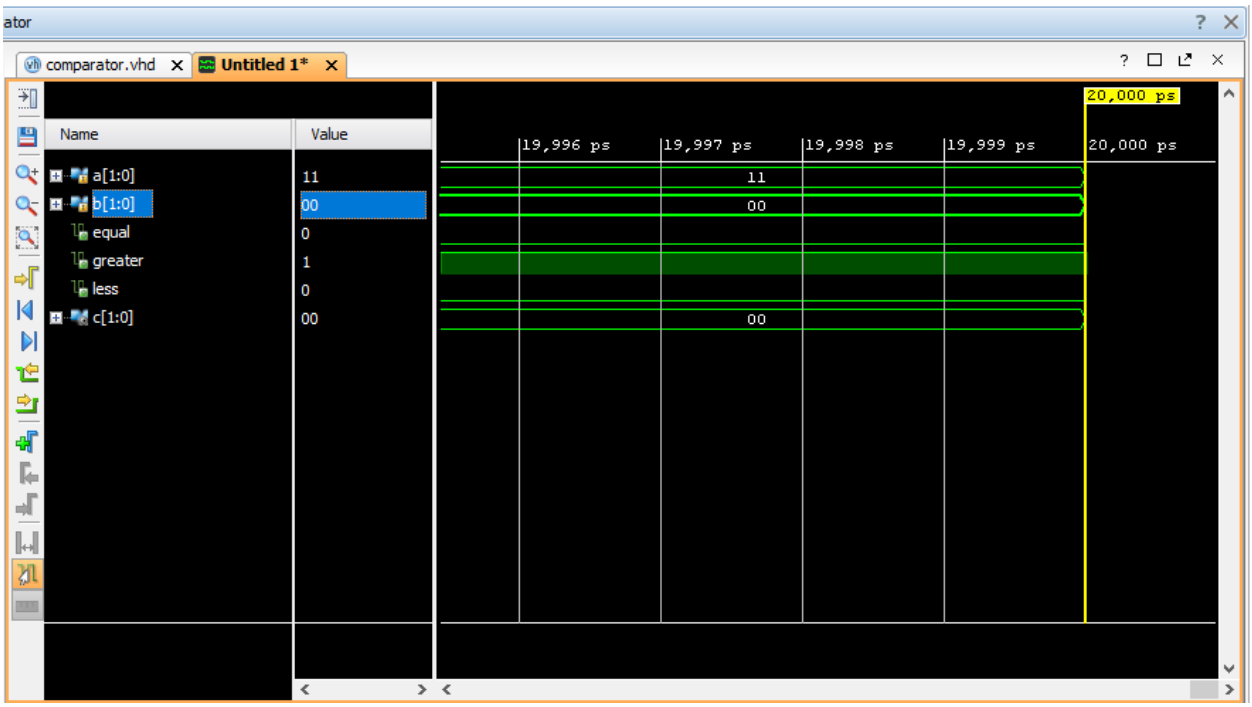
D:/Projects/HDL/xilinx/project_1/project_1.srcs/sources_1/new/comparator.vhd

```
32  --use UNISIM.VComponents.all;
33
34  entity comparator is port (
35    a : in std_logic_vector( 1 downto 0 ) ;
36    b : in std_logic_vector( 1 downto 0 ) ;
37    equal : out std_logic ;
38    greater : out std_logic ;
39    less : out std_logic
40    );
41  end entity ;
42
43  architecture Gate of comparator is
44  signal c : std_logic_vector( 1 downto 0 ) ;
45  begin
46    c <= a xnor b ;
47
48    less <= ( ( not a(1) and b(1) )
49          or ( c(1) and not a(0) and b(0) ) ) ;
50
51    greater <= ( ( not b(1) and a(1) )
52          or ( c(1) and not b(0) and a(0) ) );
53
54    equal <= c(0) and c(1) ;
55  end Gate;
56
```

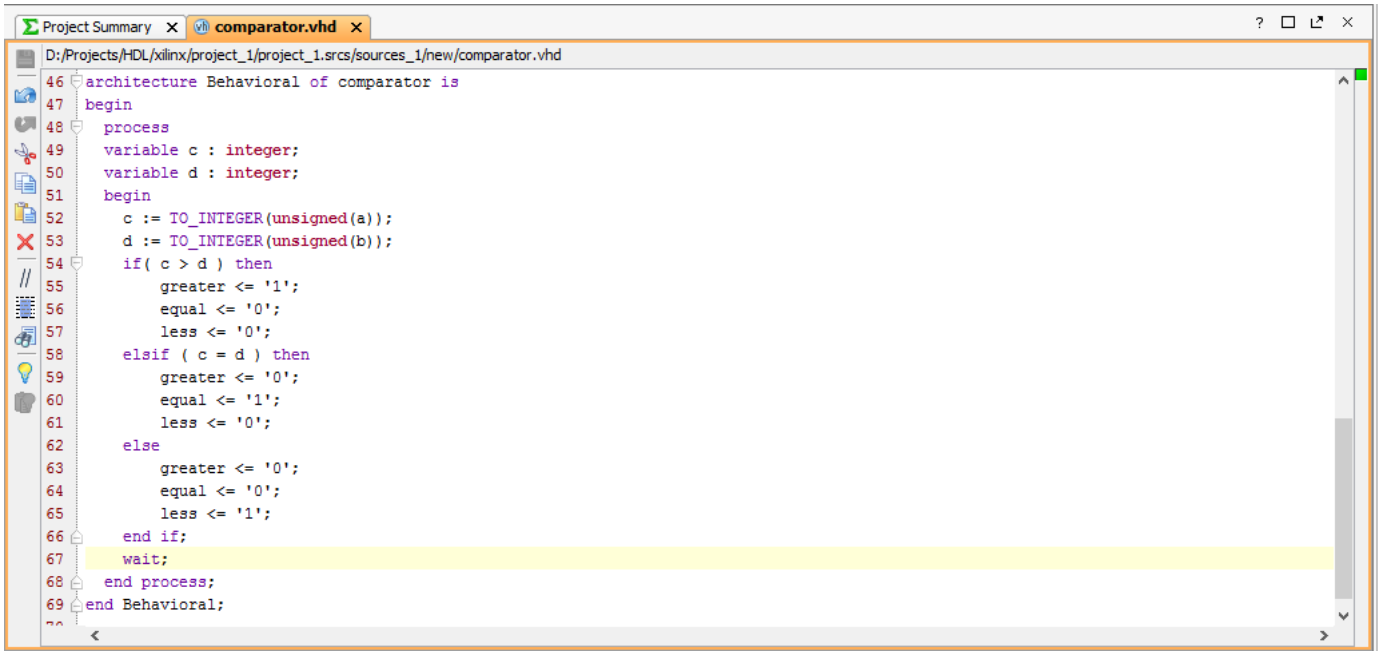
شکل موج ها به صورت زیر در می آید :


comparator.vhd
Untitled 1*
Name
Value
a[1:0]
b[1:0]
equal
greater
less
c[1:0]
11
11
1
0
0
11
999,996 ps
999,997 ps
999,998 ps
999,999 ps
1,000,000 ps
1,000,000 ps


comparator.vhd
Untitled 1*
Name
Value
a[1:0]
b[1:0]
equal
greater
less
c[1:0]
01
11
0
0
1
01
1,999,982 ps
1,999,983 ps
1,999,984 ps
1,999,985 ps
1,999,986 ps



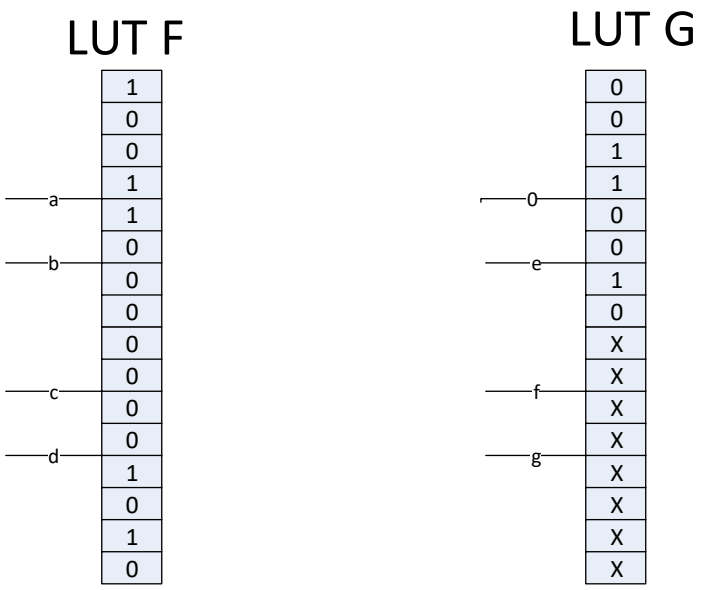
سپس به صورت رفتاری پیاده سازی مینماییم :



در این حالت هم شکل موج همان مانند بالاست

سوال ۸ :

ابتدا محتوای LUT ها را مشخص میکنیم :



برای اینکه خروجی های مورد نظر را ببینیم باید مقادیر s ها به صورت زیر باشد :

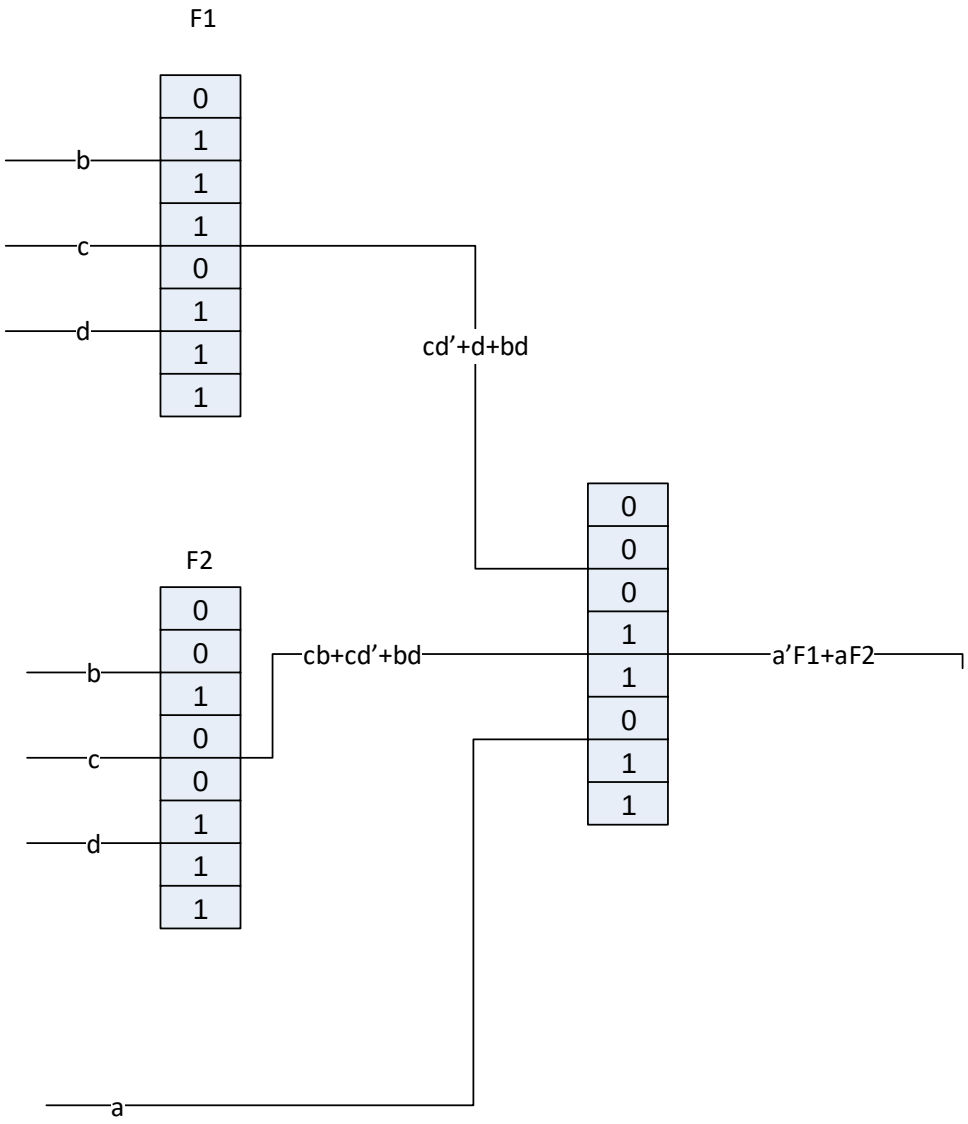
S12 = 1 , s6 = 01 , s11 = 1 , s3 = 10

همچنین برای اینکه فلیپ فلاپ ها کار کنند باید s10 , s9 نیز برابر با ۱ باشد.

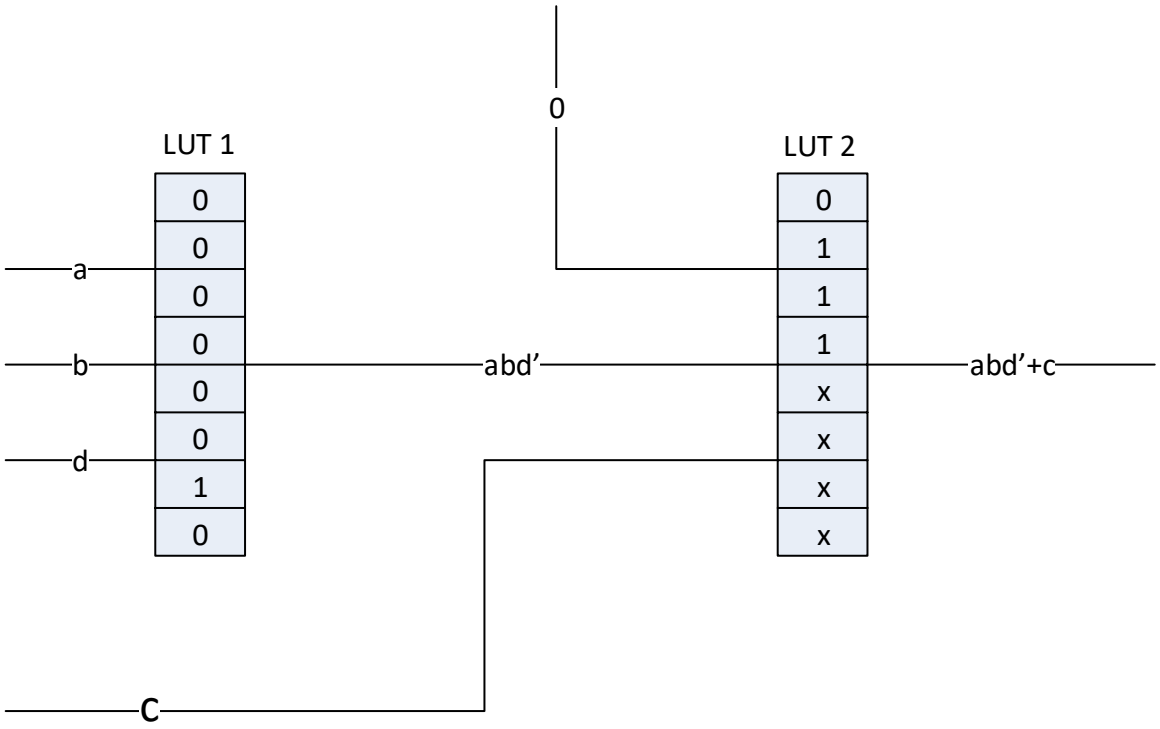
سوال ۹:

در نهایت می توان عبارت E’ را از عبارت نهایی فاکتور گرفت و عبارت باقی مانده یک تابع چهار ورودی ست که با یک LUT قابل پیاده سازی می باشد و بعد برای AND کردن از یک LUT استفاده میکنیم.در نهایت دو LUT استفاده میکنیم.

سوال ۱۰ : الف) حداقل به LUT ۴ نیاز داریم :

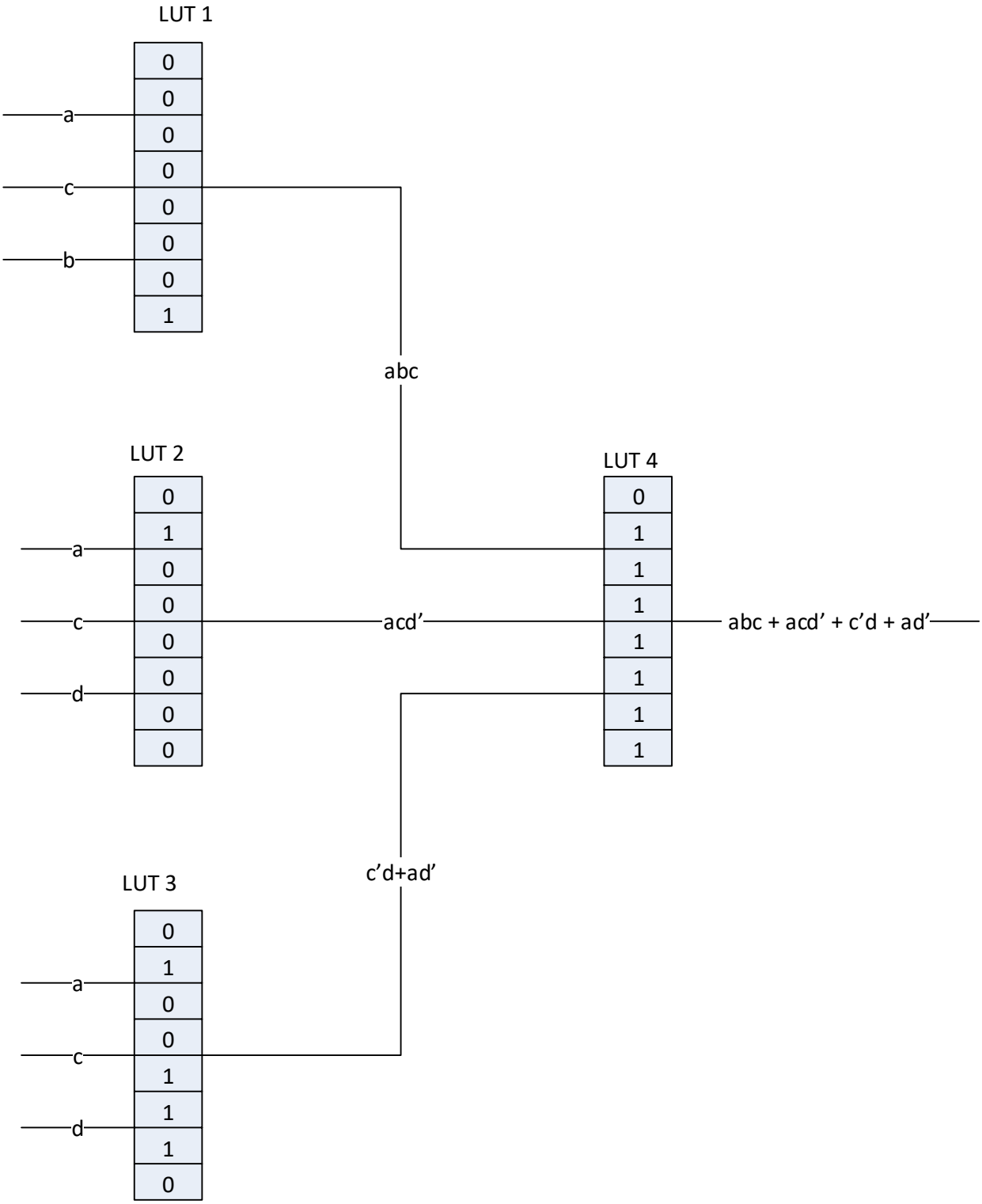


ب) به ۲ LUT نیاز داریم :



سوال ۱۱ :

در حالت اول با LUT انجام میدهیم : (در این حالت با توجه به اینکه حداقل تعداد LUT ها مطرح نیست لزوما تعداد LUT های پایین کمینه نیستند.این مدار را میتوان با ۳ LUT نیز پیاده سازی نمود)


LUT 1
0
0
0
0
0
0
0
1
a
c
b
abc
LUT 2
0
1
0
0
0
0
0
0
a
c
d
acd’
LUT 4
0
1
1
1
1
1
1
1
abc + acd’ + c’d + ad’
c’d+ad’
LUT 3
0
1
0
0
1
1
1
0
a
c
d

حال اگر با mux طراحی کنیم به صورت زیر میشود :