

بسم الله الرحمن الرحيم

# گزارش تحقیف اول *FPGA*

مهیار عنصری ۹۶۳۲۰۹۳

# سوال اول:

-۱

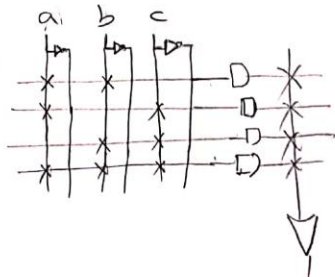
در معماری PAL ورودی AND قابل برنامه ریزی و ورودی OR ثابت است ولی در معماری PLA هر دو ورودی AND و OR قابل برنامه ریزی هستند.

ویژگی	PLA	PAL
کاربرد زیادی دارد	کمتر مورد استفاده قرار می گیرد	مصرف و کاربرد
کمتر انعطاف پذیر است اما پیچیدگی کمتری دارد	انعطاف پذیری زیادی دارد	انعطاف پذیری
هزینه متوسط	گران قیمت	هزینه
تعداد محدودی از توابع پیاده سازی می شود	تعداد زیادی از توابع را پیاده سازی می کند	تعداد توابع قابل پیاده سازی
زیاد	کم	سرعت

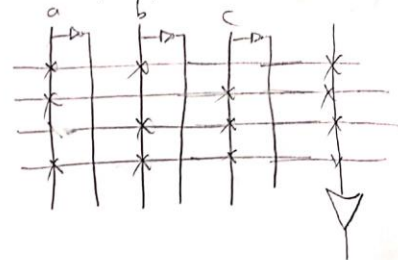
تابع صورت سوال:  $ab + ac + bc + abc$

جملات تکراری قابل استفاده در PAL نیستند یعنی مثلاً عبارت  $ab + ab + abc$  را با AND کنیم.

PLA



PAL



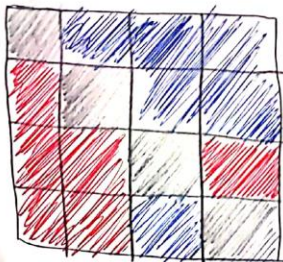
مهمترین تفاوت این دو معماری در این است که جملات تکراری

در معماری PAL دوباره قابل استفاده نیستند و باید ساخته شوند.

## سوال دوم:

۲- ابتدا جدول صحت مقایسه کرد دو بیتی را رسم می کنیم. فرض می شود:  $A_1 = ab$ ,  $A_2 = cd$

a	b	c	d	$A_1 > A_2$	$A_1 = A_2$	$A_1 < A_2$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0



$$A_1 = A_2 : (a \odot c) \cdot (b \odot d)$$

$$A_1 > A_2 : a\bar{c} + b\bar{c}d + abd$$

$$A_1 < A_2 : \bar{a}c + \bar{b}cd + \bar{a}bd$$

برای سافت مقایسه کرد ۴ بیتی ، می توان خروجی مقایسه کرد دو بیتی برابر از سر نو را (خروجی مساوی) به عنوان بیت فعال سار مقایسه کرد دو بیتی ست راست استفاده شود.

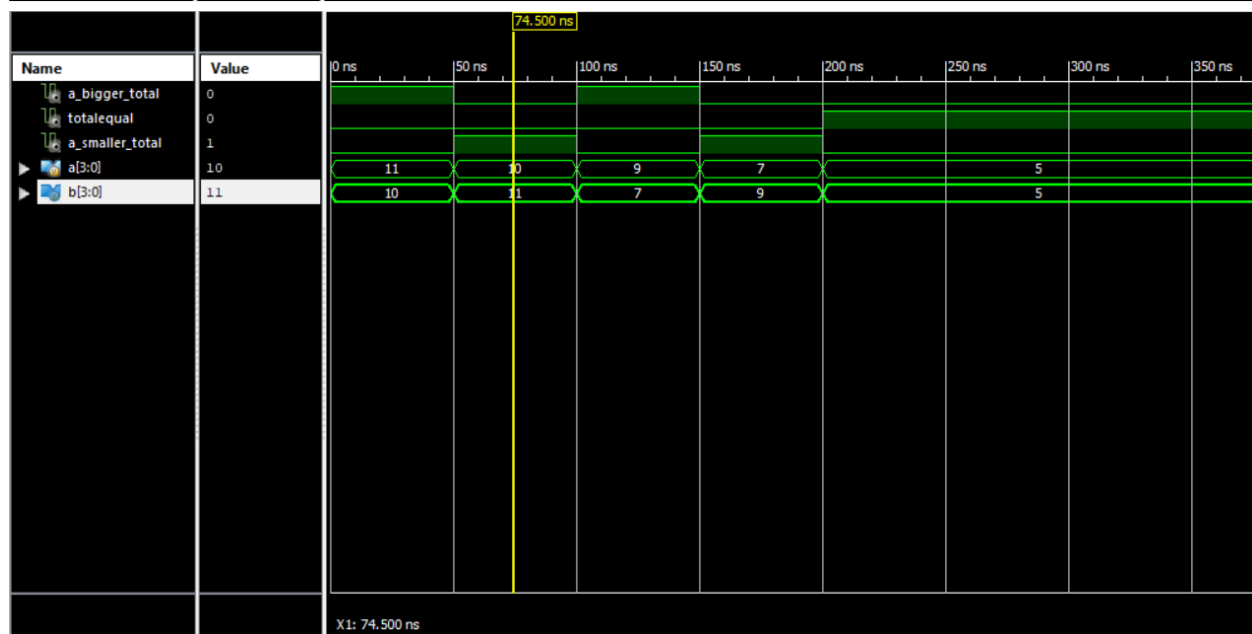
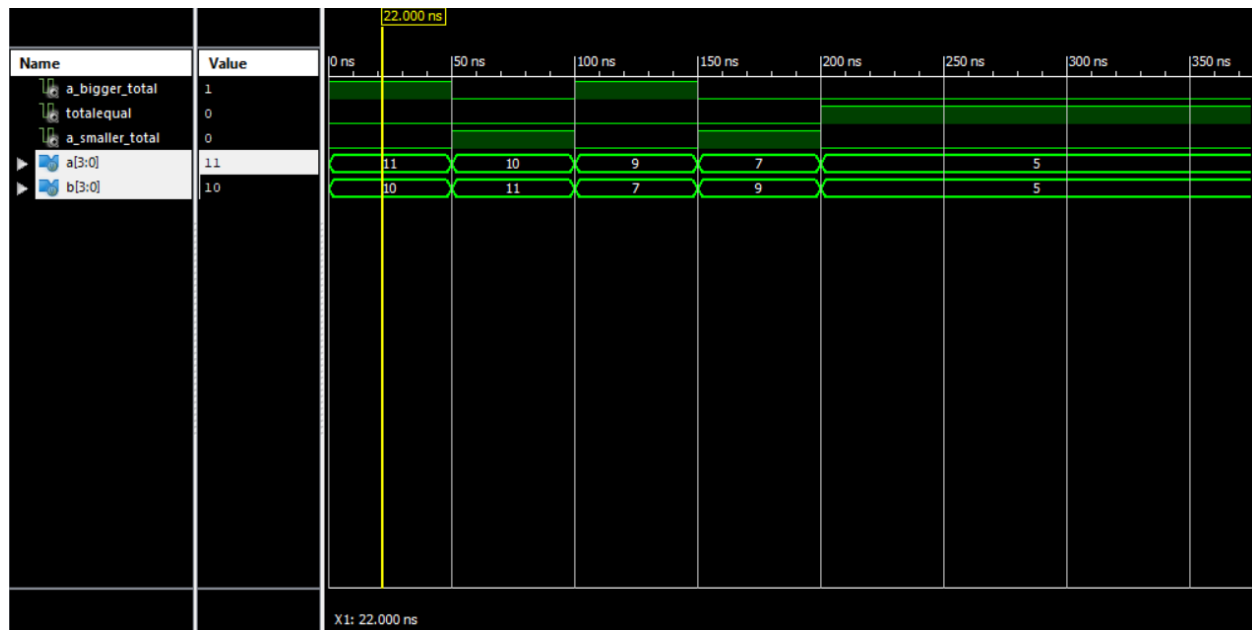
## توضیحات سوال دوم:

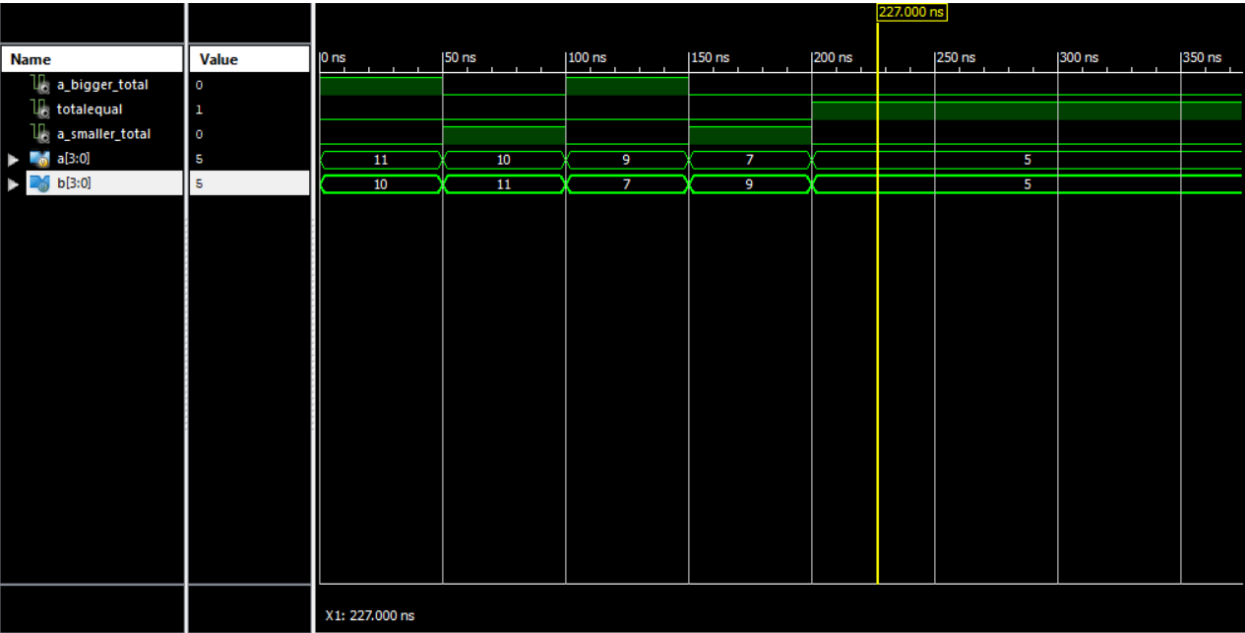
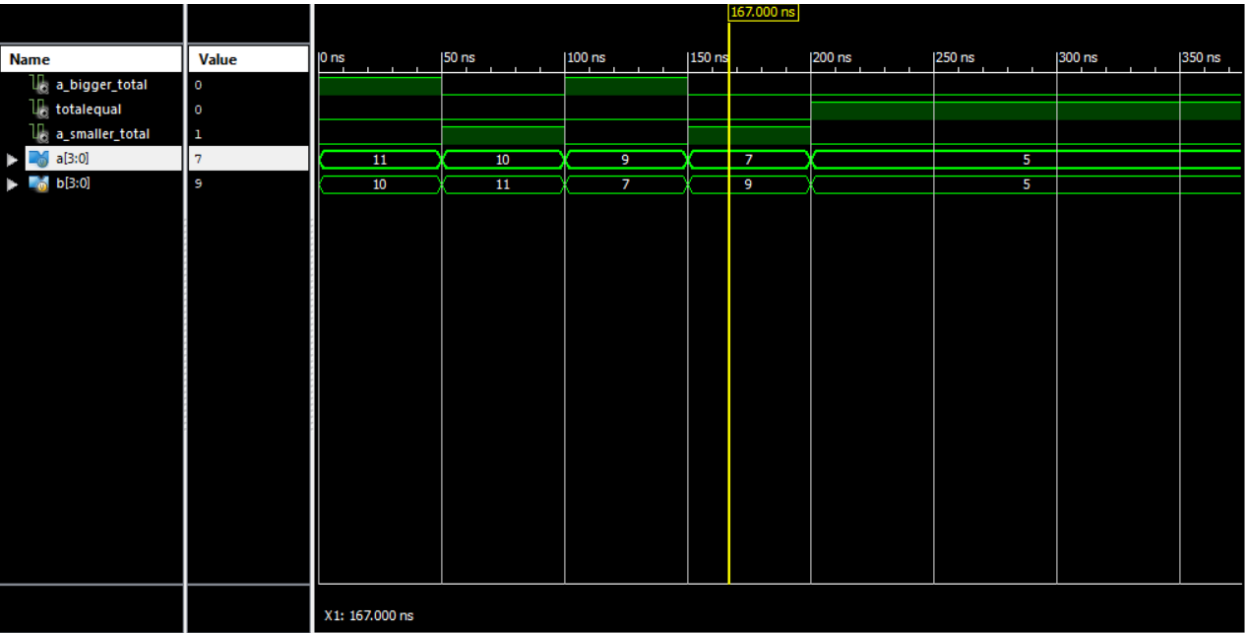
ابتدا جدول صحت مقایسه گتر دو بیت را تشکیل می دهیم و جملات مربوط به آن را بر حسب ورودی ها به دست می آوریم. سپس با ترکیب دو مقایسه گتر دو بیت یک مقایسه گتر چهار بیت می سازیم. ساخت مقایسه گتر ۴ بیت اینگونه است که ابتدا دو بیت سمت چپ با هم مقایسه می شوند اگر یکی بزرگتر از دیگری بود دیگر نیازی به مقایسه دو بیت سمت راست نیست و اگر دو بیت سمت چپ مساوی بودند کار به دو بیت سمت راست می کشد.

برای اینکه دو بیت سمت راست را وابسته به مساوی بودن دو بیت سمت چپ کنیم دو بیت سمت راست دو عدد (مجموعاً ۴ بیت) را از یک بافر کنترلی عبور می دهیم و پایه کنترلی بافر را خروجی مساوی مقایسه گتر سمت چپ قرار می دهیم. به این ترتیب وقتی دو بیت سمت چپ مساوی می شوند پایه مساوی یک شده و بافرهای کنترلی مقایسه گتر سمت راست فعال می شود و عمل مقایسه انجام می شود.

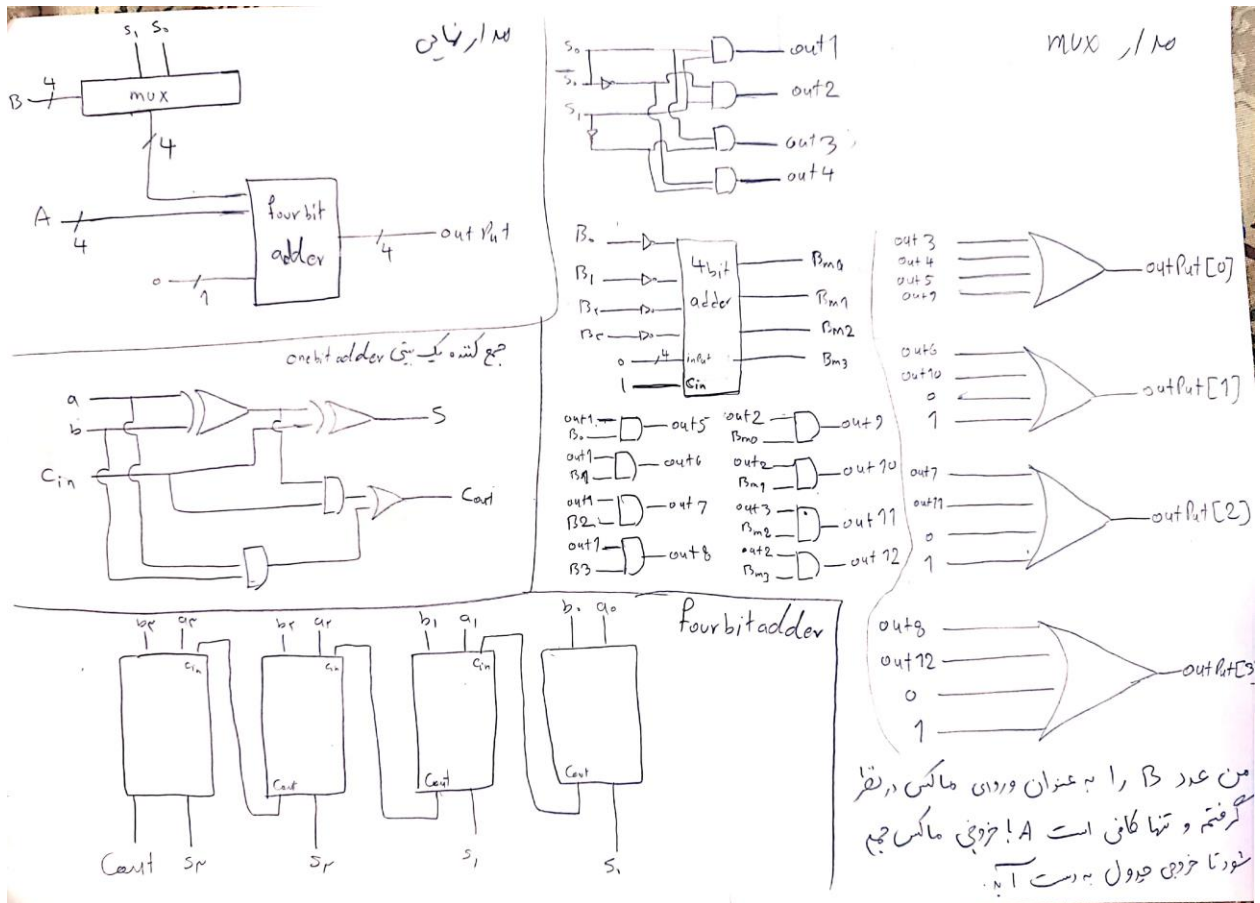
برای صحت اطمینان همه خروجی های سمت راست را با خروجی مساوی سمت چپ AND می کنیم و در نهایت حاصل AND دو خروجی غیر مساوی سمت راست را با خروجی های غیر مساوی سمت چپ OR می کنیم.

# تست سوال دوم:





سوال سوم:



در اینجا من عدد B را داخل ماکس در نظر گرفته‌م و خروجی ماکس عددی باشد که با A جمع می‌شود.

برای ساحت جمع‌کننده چهاربیتی ابتدا جمع‌کننده یکبیتی را مطابق شکل می‌سازیم و بعد چهارمقایسه‌گر یکبیتی را ترکیب می‌کنیم و مقایسه‌گر ۴بیتی می‌سازیم.

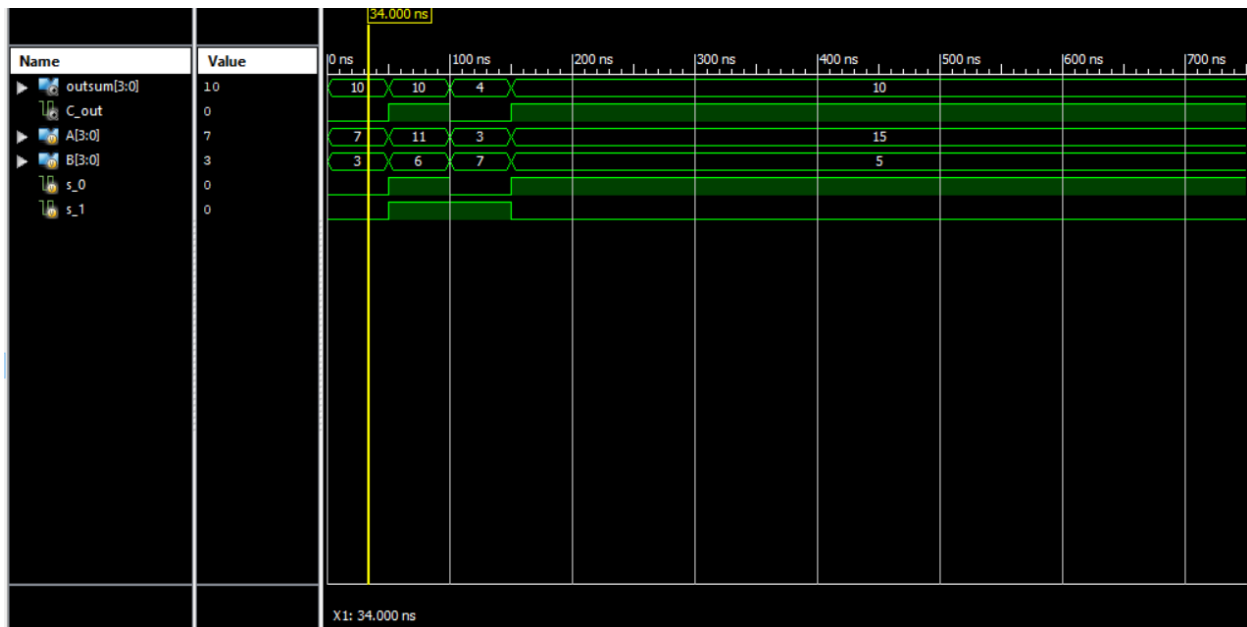
برای ساخت ماکس هم ابتدا من دویپیه کستری و نات های آن را با هم از AND عبور دارم (مجموعاً ۴ حالت) و سپس هر حالت را با خروجی مطلوب آن

AND کردن. یعنی مثلا AND حاصل شده از NOT کردن  $S_0$  و  $S_1$  که خروجی آن باید B باشد را با  $\oplus$  بیت B از  $\oplus$  گیت AND عبور داریم. برای  $B - 1$  و  $1 +$  و  $-1$  هم همین کار را انجام داریم یعنی حاصل AND مربوطه با  $\oplus$  بیت خروجی ما که یکی از  $\oplus$  حالت  $B +$  و  $B - 1$  و  $1 +$  و  $-1$  می باشد را دوباره AND کردن.

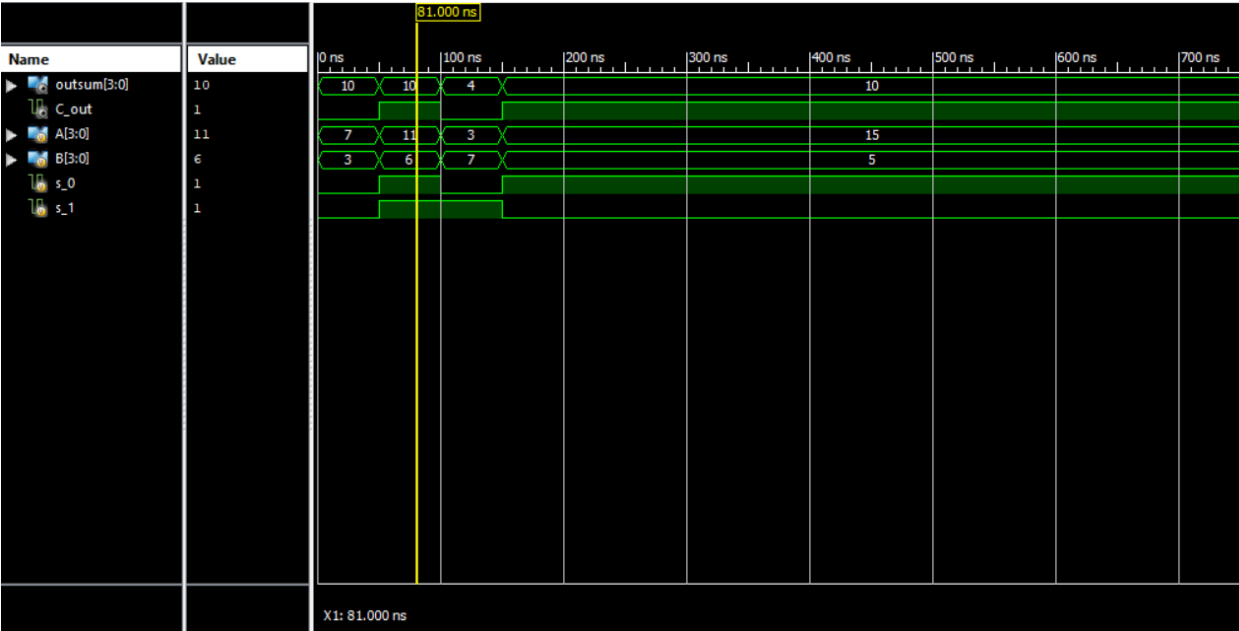
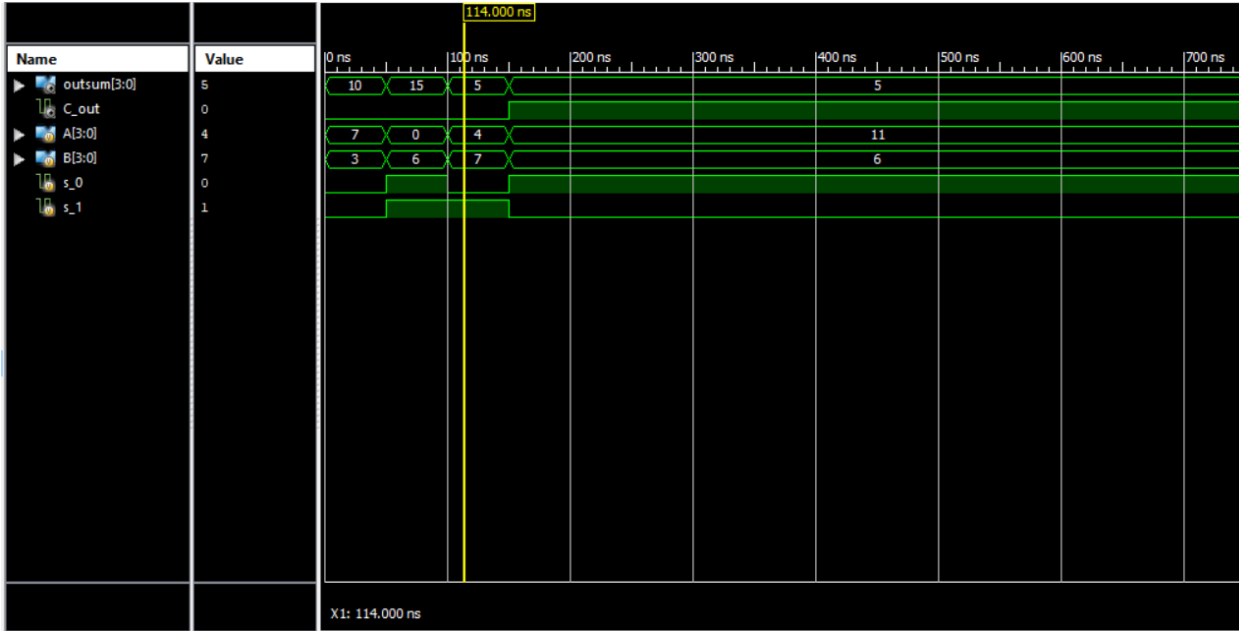
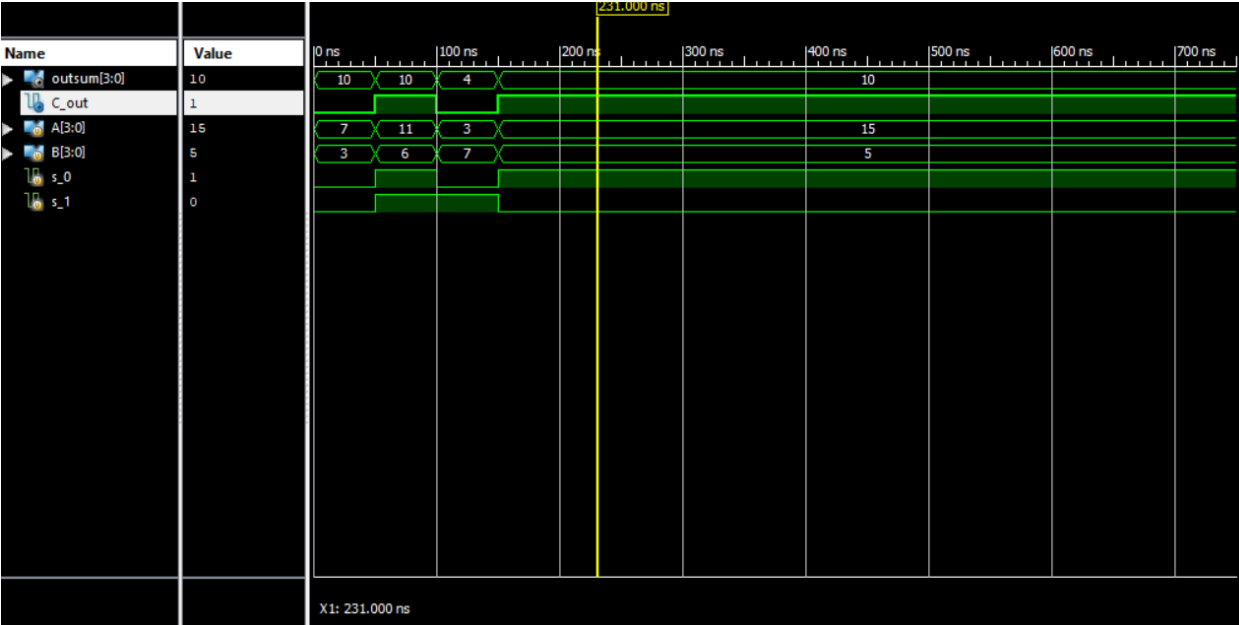
پس در مرحله آخر خروجی ما کس که بسته به پایه های کنترلی آکئون یکی از چهار حالت فوق ( $B +$  و  $B - 1$  و  $1 +$  و  $-1$ ) است را با A جمع کردن.

ورودی های ماثول نهایی A و B و  $S_0$  و  $S_1$  می باشد و یک خروجی برای مجموع و یک خروجی هم برای CARRY احتمالی در نظر گرفته.

کس های تست در زیر آورده شده است ضمت های ماکس و جمع کننده نیز در فایل زیپ شده موجود می باشد.







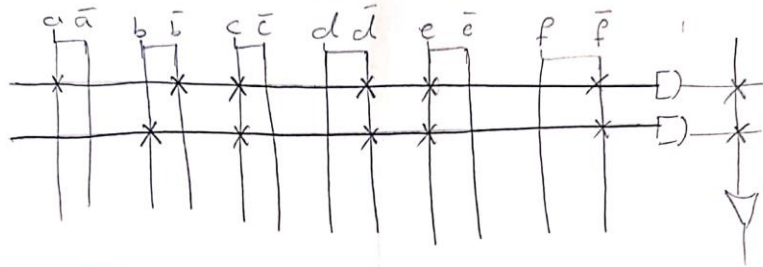
# سوالات چهارم و اختیاری:

ف - معادله بولی قفل را بنویسیم:

$$Z=1: a\bar{b}c\bar{d}e\bar{f} + \bar{a}bcd\bar{e}f + abcd\bar{e}f$$

$$= a\bar{b}c\bar{d}e\bar{f} + bcd\bar{e}f(a+\bar{a}) = a\bar{b}c\bar{d}e\bar{f} + bcd\bar{e}f$$

در این حالات  $Z=1$  و در بقیه 6 حالت ممکن  $Z=0$  است.



سوال اختیاری: حروف بزرگ در جدول اسکی (ASCII) از عدد 65 شروع می شوند.

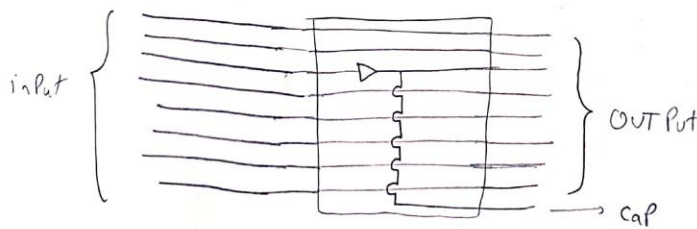
معادل باینری این عدد برابر 01000001 است، آخرین حرف هم Z است که که اسکی 90 معادل 01011010

دارد. می بینیم دیتا 8 بیتی این حرف صفر بدون سر بیت است. از طرفی اگر این بیت یک شود

حرف بزرگ به حرف کوچک تبدیل می شود و با همزن شدن حرف کوچک به بزرگ تبدیل می شود.

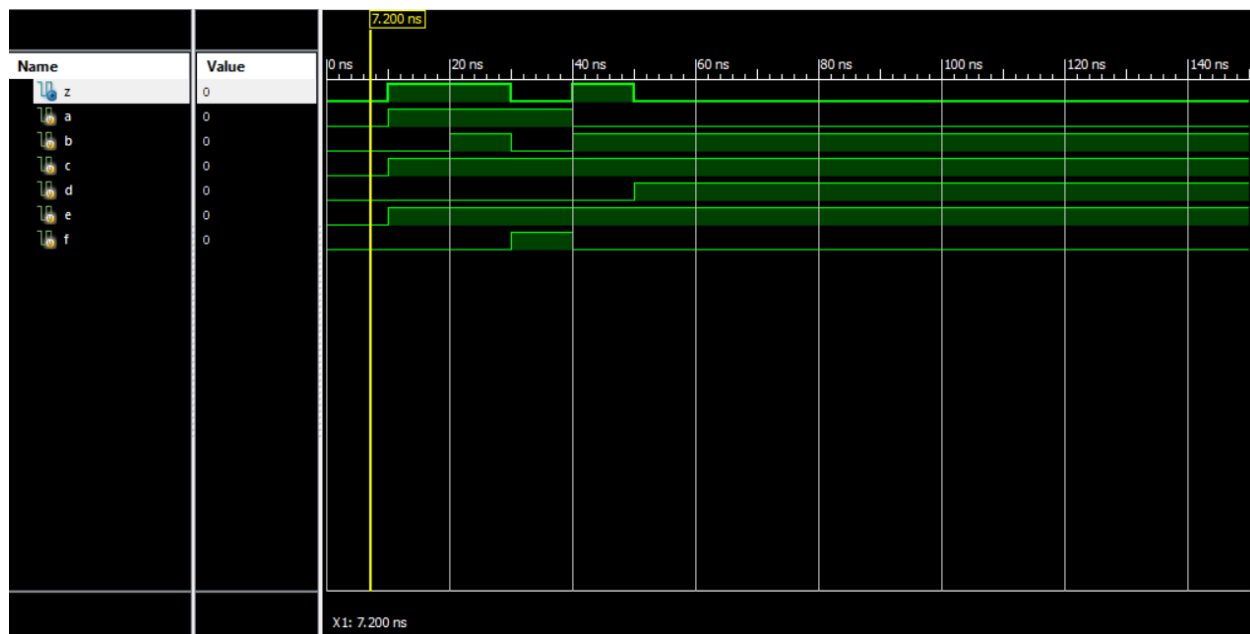
سه عدد 8 بیتی را که دریافت کردیم، بیت سوم بیت چپ را نام می کنیم و به عنوان CAP بیرون می دهیم

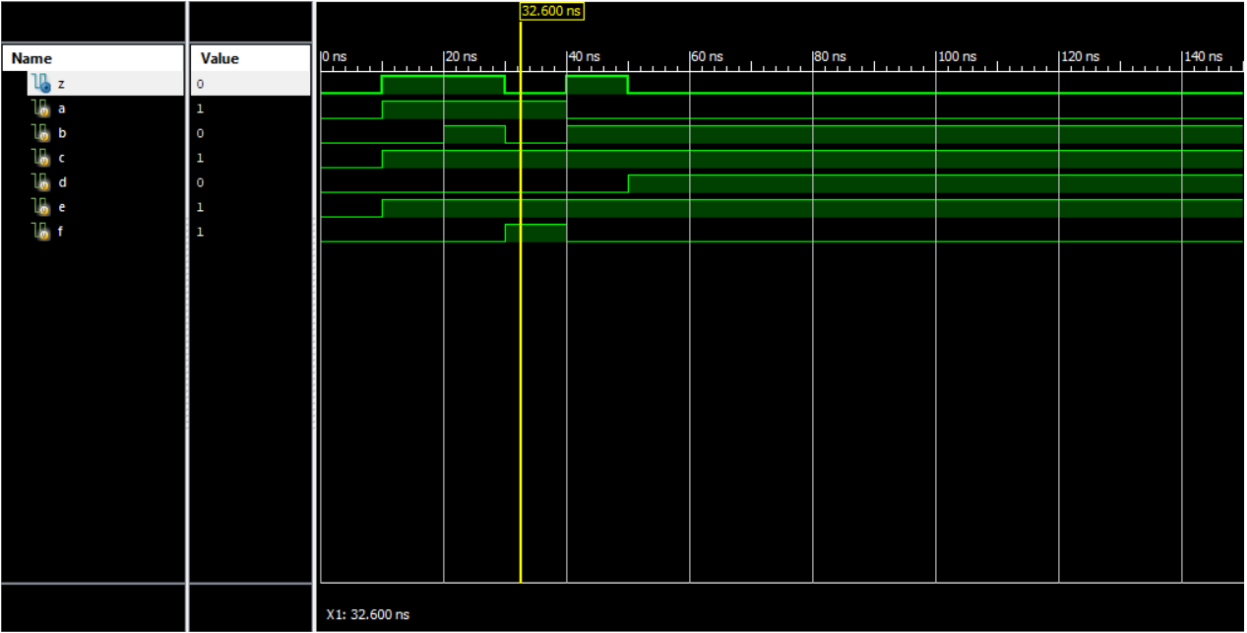
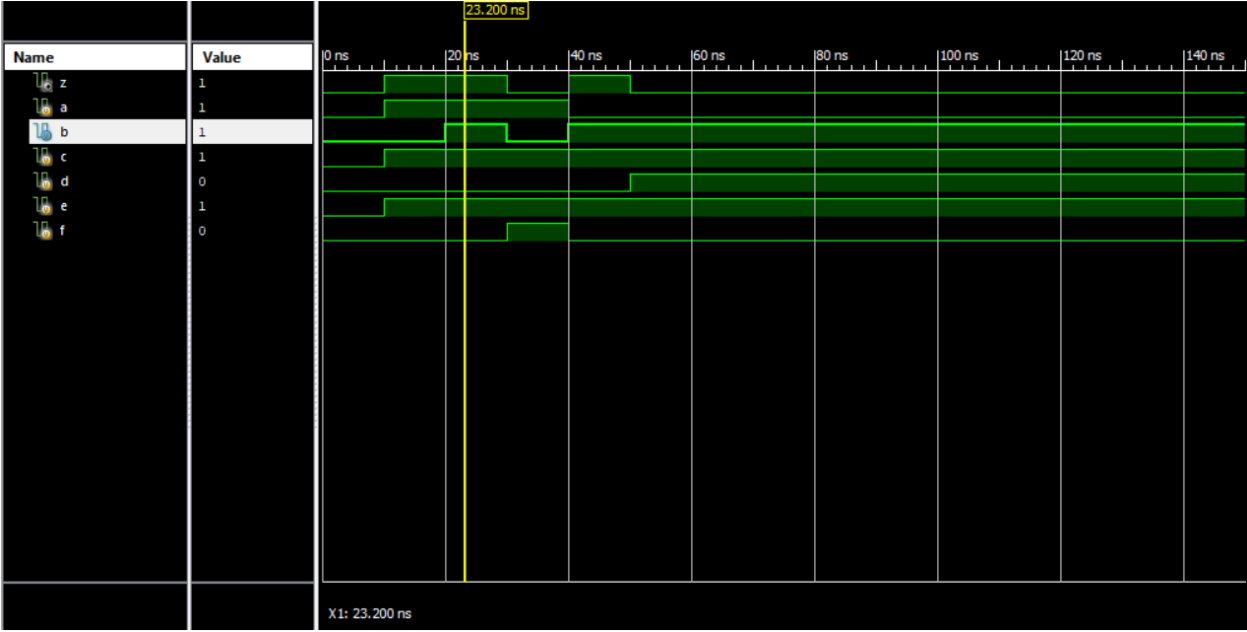
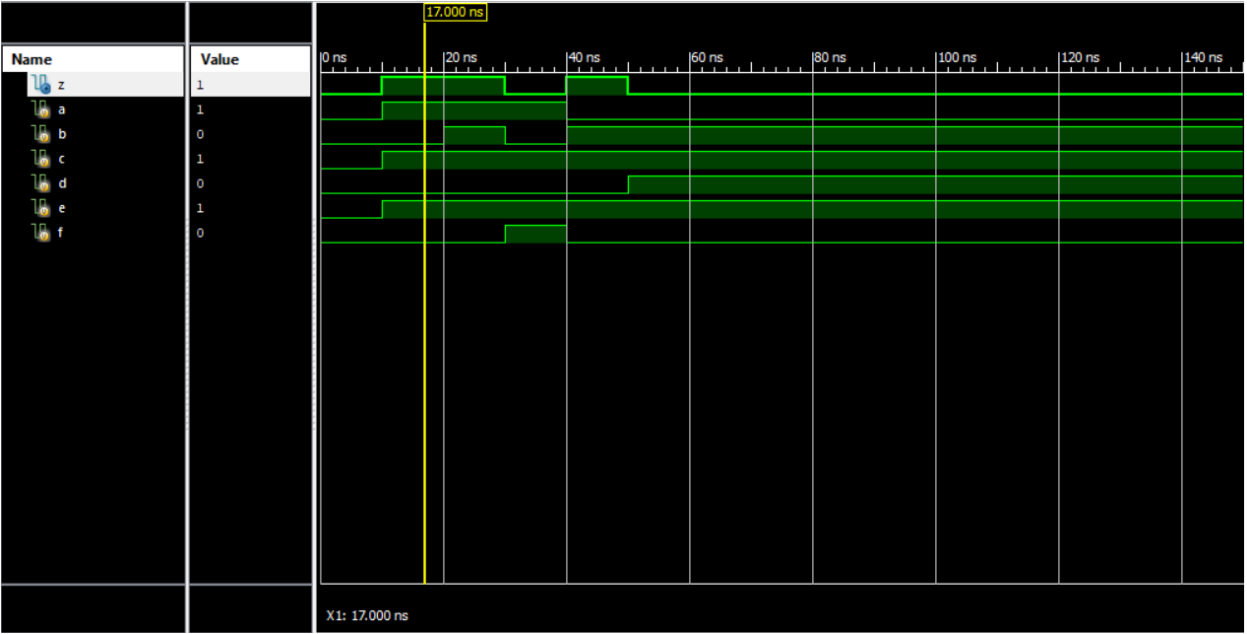
همچنین عدد 8 بیتی فردی هم از نات شدن بیت سوم بیت چپ بدست می آید.

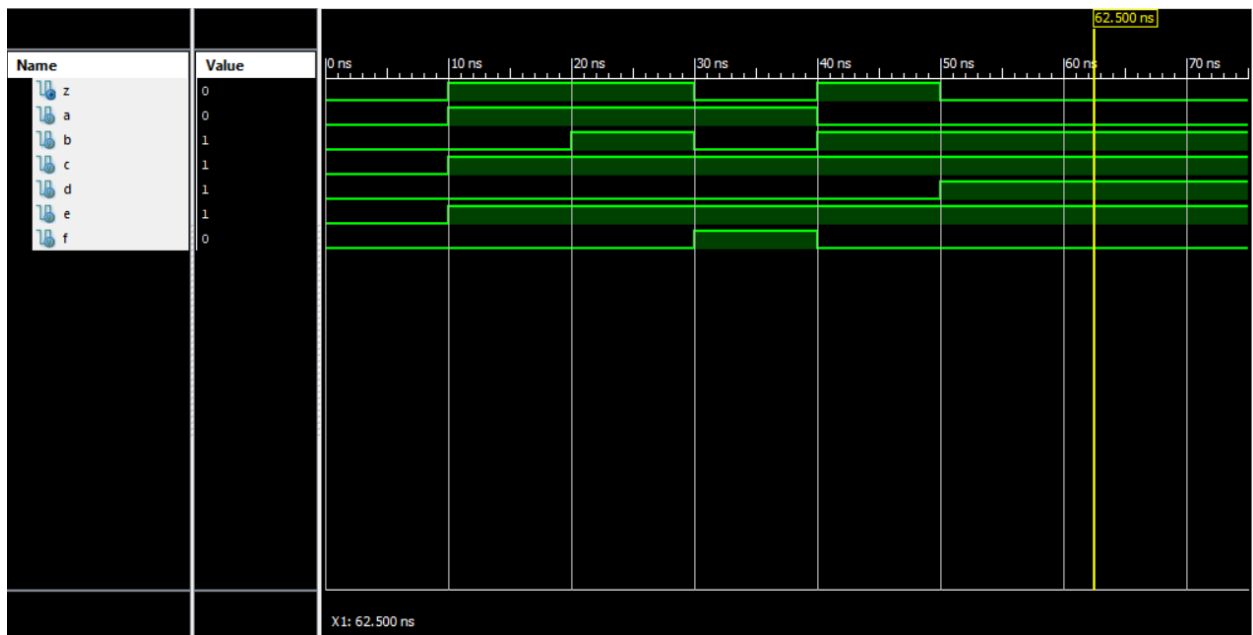
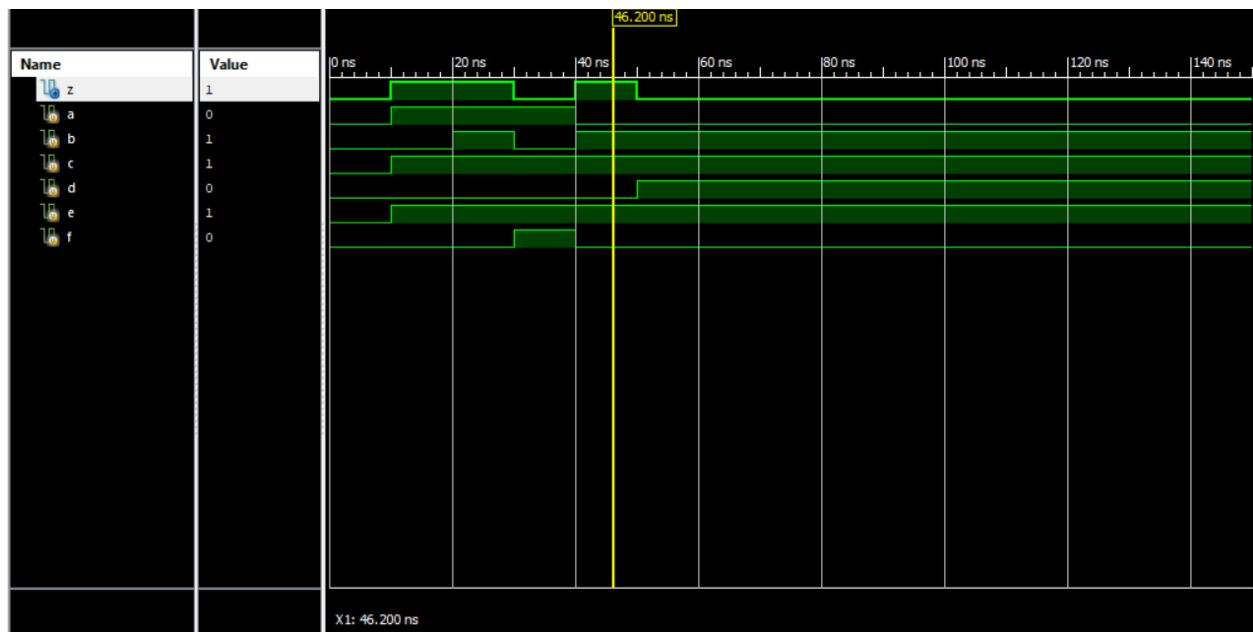


## سوال چهار:

با نگاهی در معادله بولی داده شده متوجه می شویم دو جمله دوم و سوم قابل ساده سازی هستند و این کار را انجام میدهیم تا معادله بولی ساده شده به دست بیاید. پس از ساده سازی به دو ترم نهایی که نوشته می رسیم و از مجموع ۶ حالت ممکن فقط همین سه حالت که به دو جمله ساده شده باعث باز شدن قفل یعنی  $Z=1$  می شود. با کمک این اطلاعات پیاده سازی را انجام می دهیم.





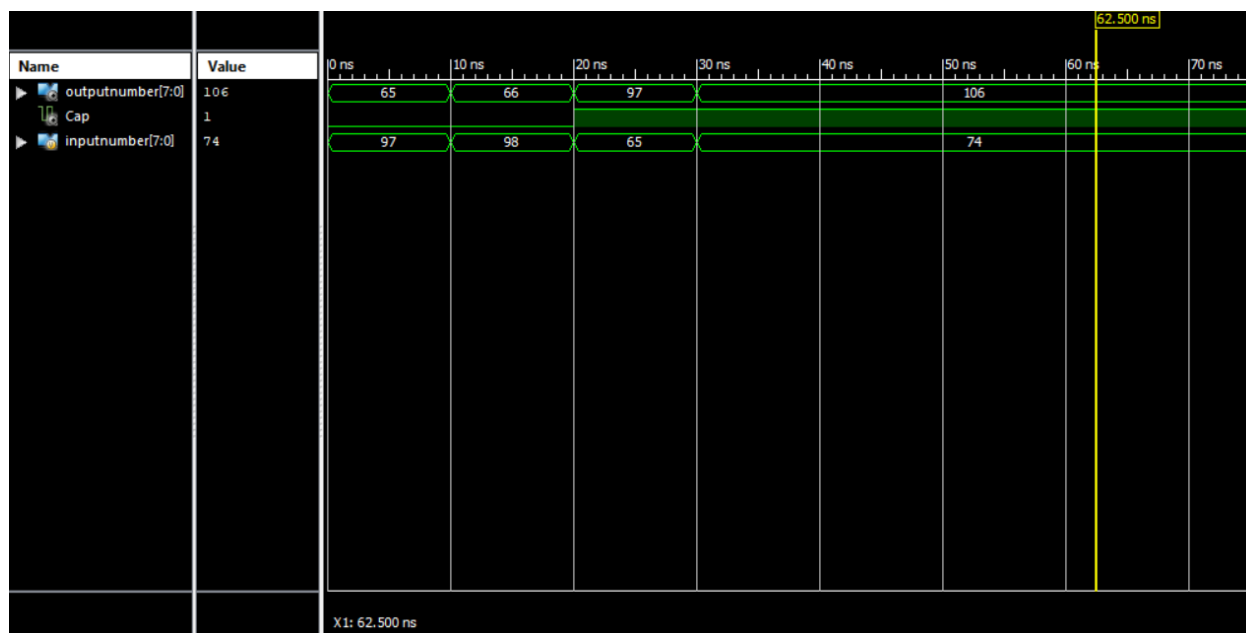


می بینیم که در ۳ حالت مطلوب سوال  $z=1$  شده و در دیگر حالات  $z=0$  است.

# سوال اختیاری:

کد اسکی حروف بزرگ از عدد ۶۵ که معادل باینری ۰۱۰۰۰۰۰۱ دارد شروع میشود و  
ت عدد ۹۰ ادامه دارد که معادل باینری آن ۰۱۰۱۱۰۱۰ است. متوجه می شویم در  
همه اعداد مربوط به حروف بزرگ بیت سوم از سمت چپ صفر است و می توان  
با دادن not این بیت به cap به یکی از خواسته های سوال رسید.

از طرفی فاصله بین کد اسکی حروف بزرگ و کوچک ۳۲ است یعنی ما اگر not  
بیت سوم از سمت چپ را انجام دهیم (که ۳۲ به عدد اضافه یا کم شون) حرف  
بزرگ به کوچک و کوچک به بزرگ تبدیل می شود.



دو ورودی اول a و b هستند و به درستی خروجی دریافت شده و cap هم صفر  
است و دو ورودی بعدی A و I هستند و باز هم خروجی درست داده شده و  
CAP هم برابریک است.