

(1) جدول کارنو عبارت منطقی داده شده بصورت روبرو است

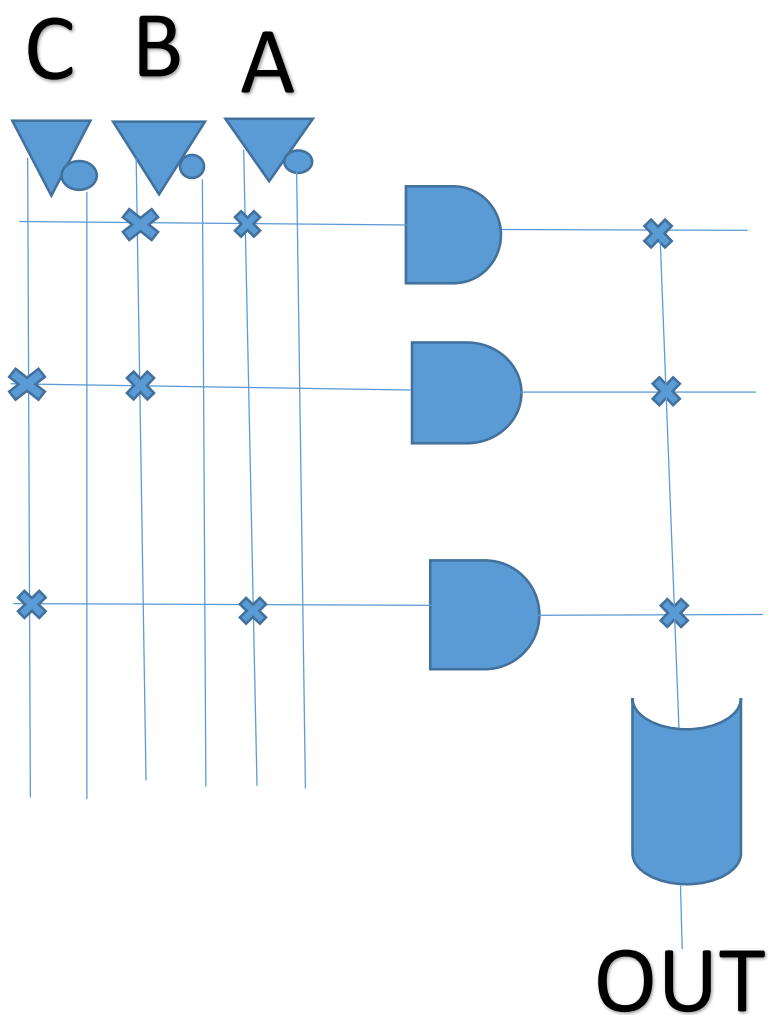
0	0	1	0
0	1	1	1

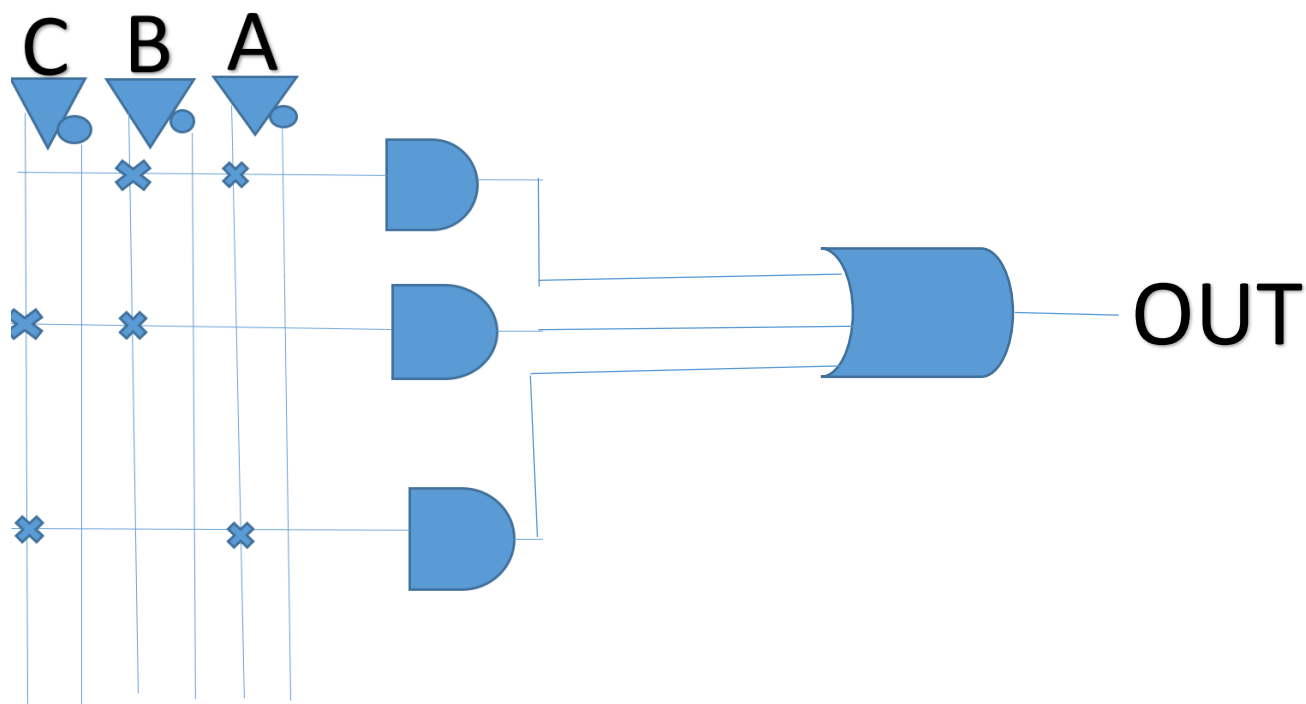
$$\text{Out} = AB + BC + AC$$

با توجه به تعریف PLA میدانیم این نوع معماری دارای AND , OR های قابل برنامه ریزی است بنابراین تمامی اتصالات را خودمان برنامه ریزی میکنیم

اما در حالت PAL گیت های AND قابل برنامه ریزی اند اما گیت های OR بصورت ثابت به یک سری گیت های AND خاصی متصل اند بنابراین اتصال گیت های OR فاقد برنامه ریزی میباشد

معماری PLA





در واقع تفاوت این دو معماری در تعداد خروجی های بیشتر از یک واضح تر میشود زیرا میبینیم که هر خروجی بصورت ثابت به گیت های AND مشخصی متصل است یا بصورت متغیر و قابل برنامه ریزی به گیت های AND متفاوتی متصل میشود

(2)

$$A_{higher} = A0.\overline{B0} + A0.A1.\overline{B1} + A1\overline{B0}.\overline{B1}$$

$$A_{lower} = \overline{A0}.B0 + B0.B1.\overline{A1} + B1.\overline{A0}.\overline{A1}$$

$$A_{same} = A0.A1.B0.B1 + \overline{A0}.\overline{A1}.\overline{B0}.\overline{B1} + A1.B1.\overline{A0}.\overline{B0} + A0.B0.\overline{A1}.\overline{B1}$$

جدول صحت معادلات بالا بصورت زیر است

A0	A1	B0	B1	Ahigher	Alower	Asame
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
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	0	1

مال فرض کنید اعداد چهار بیتی داده شده به صورت $a_0a_1a_2a_3, b_0b_1b_2b_3$ هستند در ابتدا دو بیت پرارزش را مقایسه میکنیم

مالتی که a بزرگتر است : یا دو بیت بالا a بزرگتر است یا دو بیت بالا مساوی اند و دو بیت پایین a بزرگتر است (and) فروجی

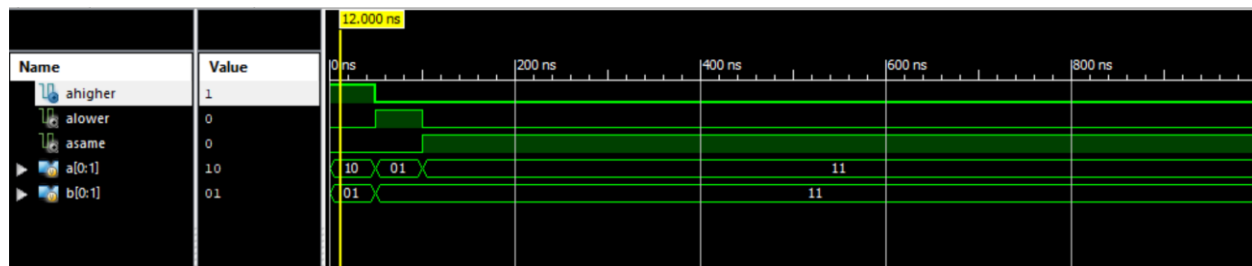
higher مقایسه کننده 4 بیتی فعال میشود بنابراین این دو حالت باید or شوند

حالتی که a کوچکتر است : یا دو بیت بالا a کوچکتر است یا دو بیت بالا مساوی اند و دو بیت پایین a کوچکتر است (and) خروجی higher مقایسه کننده 4 بیتی فعال میشود بنابراین این دو حالت باید or شوند

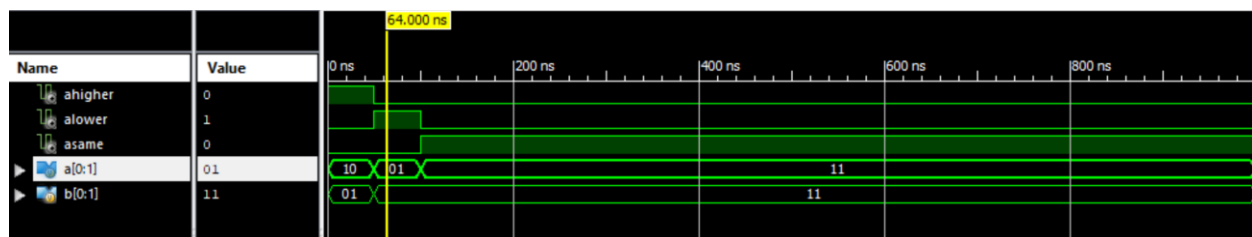
حالتی که دو عدد مساوی اند : باید دو بیت بالا مساوی باشد و دو بیت پایین هم مساوی باشد (and)

شبیه سازی مقایسه کننده دو بیتی

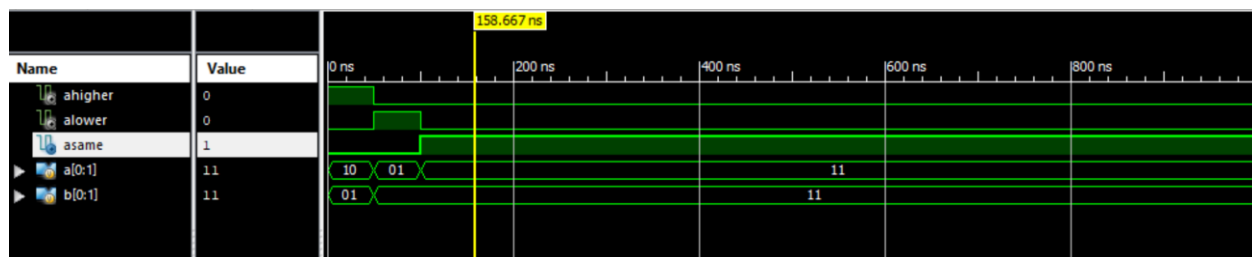
خروجی بزرگتر



خروجی کوچکتر



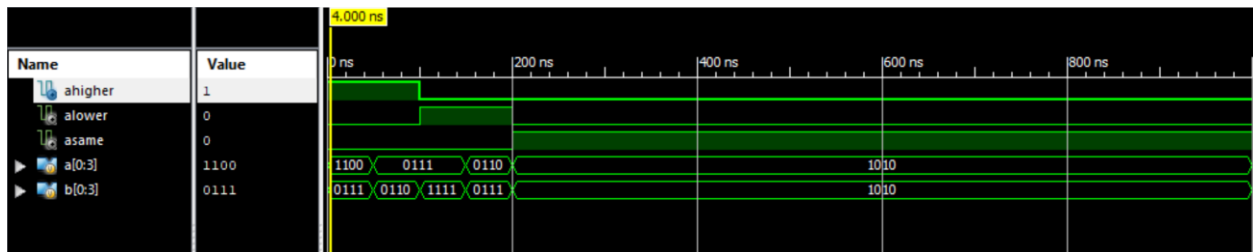
خروجی مساوی



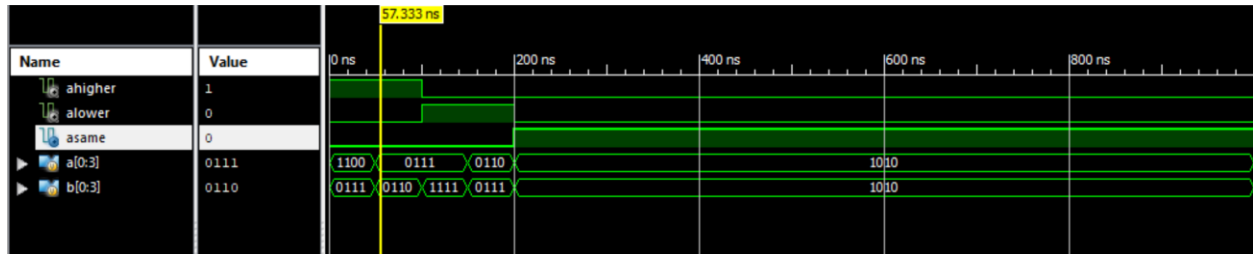
شبیه سازی مقایسه کننده چهار بیتی

خروجی بزرگتر

الف) در حالتی که دو بیت بالای a بزرگتر است

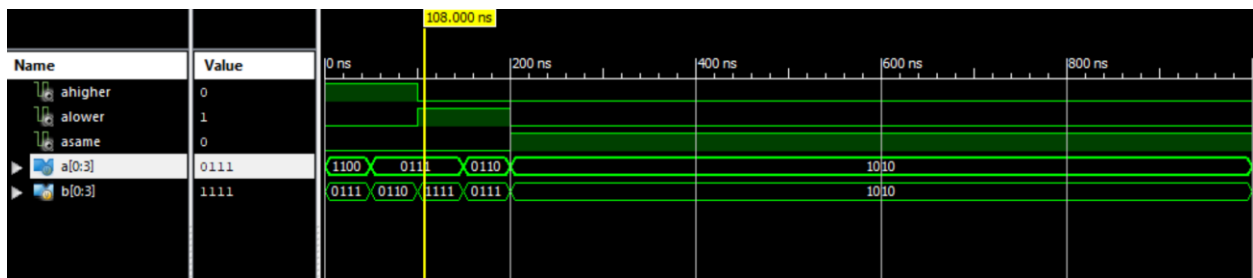


ب) در حالتی که دو بیت بالا مساوی اند و خروجی بسته به بیت های پایین است

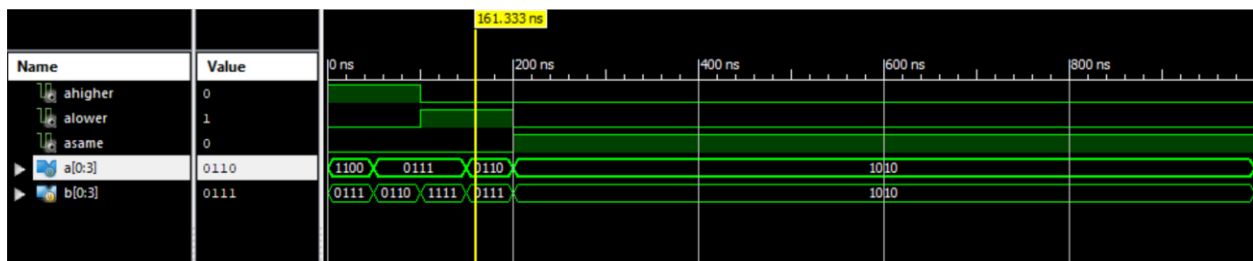


خروجی کوپکت

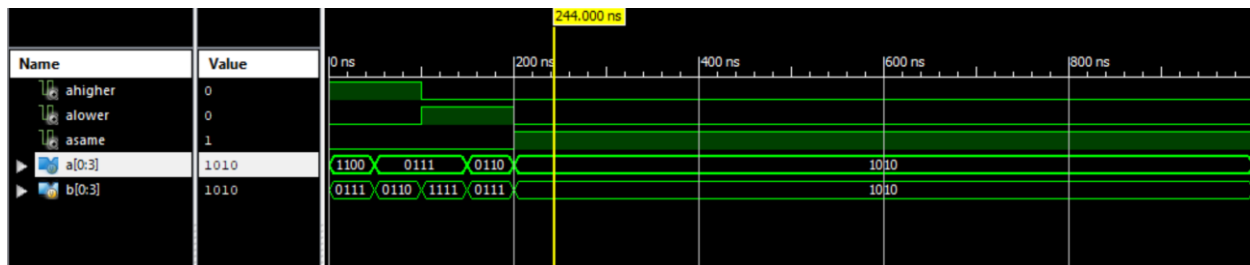
الف) در حالتی که دو بیت بالای a کوپکت است



ب) در حالتی که دو بیت بالا مساوی اند و خروجی بسته به بیت های پایین است



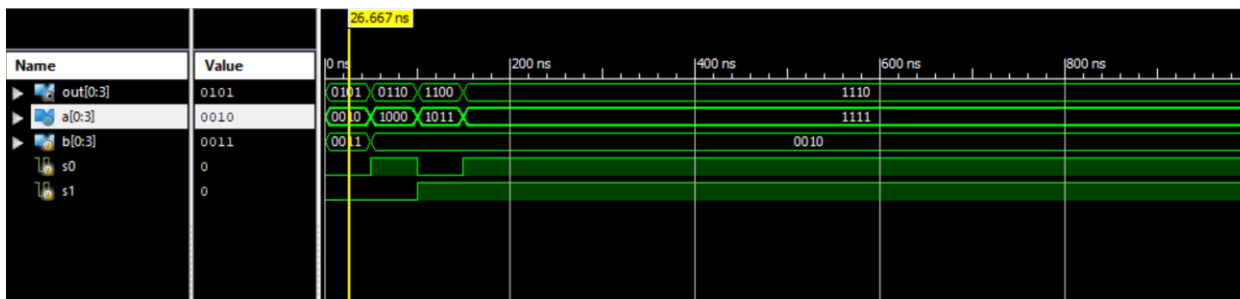
خروجی مساوی



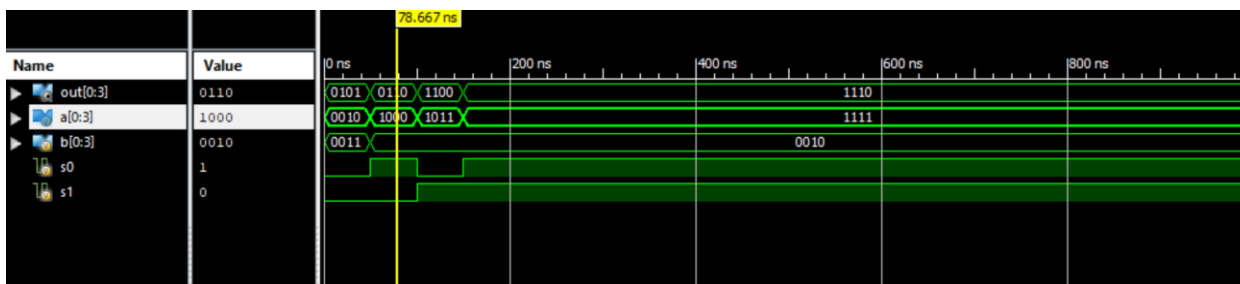
3) با توجه به جدول داده شده در سوال مشاهده میشود که تمامی فروجی ها در سمت راست فود A را دارند بنابراین کافی است ماکس مورد نظر یکی از حالات 1, -B, -B را تولید کند. از انجایی که اعداد کار شده در سوال چهار بیتی هستند باید چهار تا ماکس 4×1 داشته باشیم تا هر بار یکی از بیت های 1, -B, -B را تولید کند و در نهایت بوسیله یک جمع کننده 4 بیتی فروجی ماکس را با A جمع کند
 برای سافت -B که همان مکمل دو B است ابتدا انرا با 0 وارد گیت NOR میکنیم تا مکمل یک B سافت شود سپس بوسیله یک جمع کننده با یک جمع میکنیم تا مکمل دو B یعنی -B سافت شود

نتایج شبیه سازی

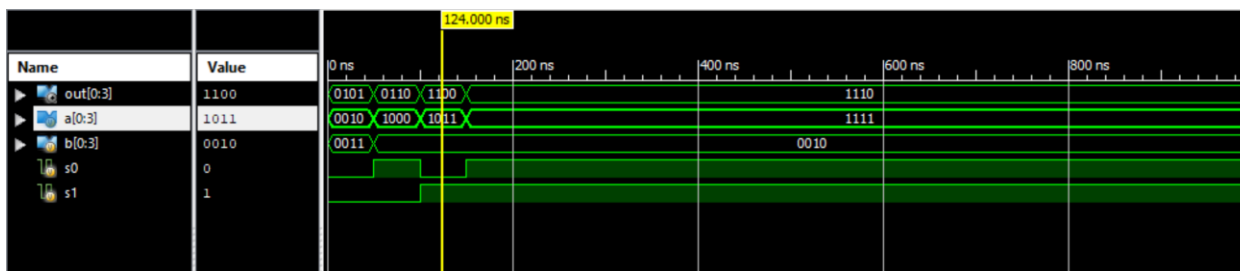
الف) $S0=0, S1=0$ باید $OUT=A+B$ باشد $A=2, B=3$ پس باید (0101) ببینیم



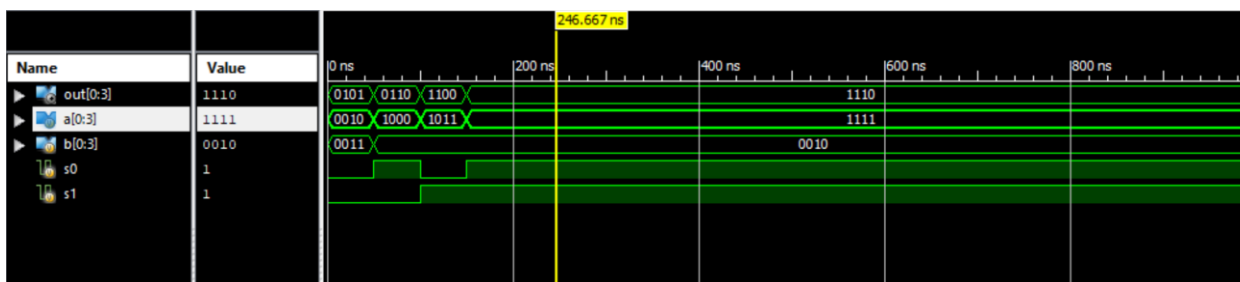
ب) $S1=0, S0=1$ باید $OUT=A-B$ باشد $A=8, B=2$ پس باید (0110) ببینیم



ج) $S1=1, S0=0$ باید $OUT=A+1$ و $A=11$ پس باید (1100) ببینیم

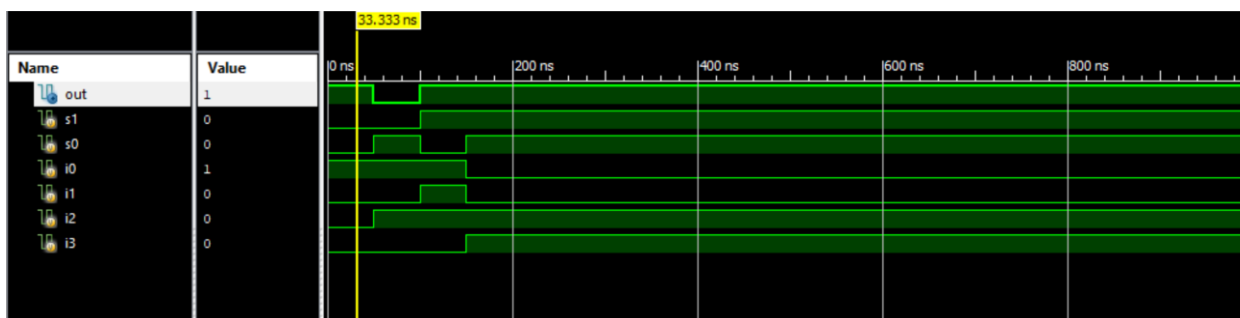


د) $S0=1$, $S1=1$ باید $OUT = A-1$ و $A=15$ پس باید 14(1110) ببینیم

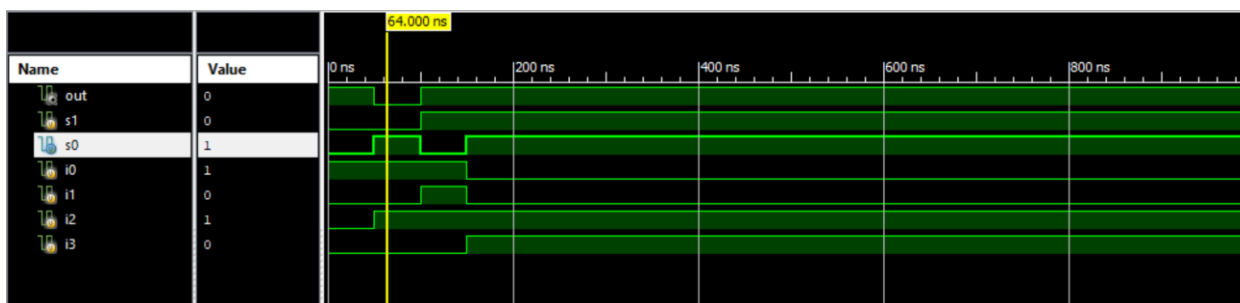


نتایج شبیه سازی ماکس 4*1

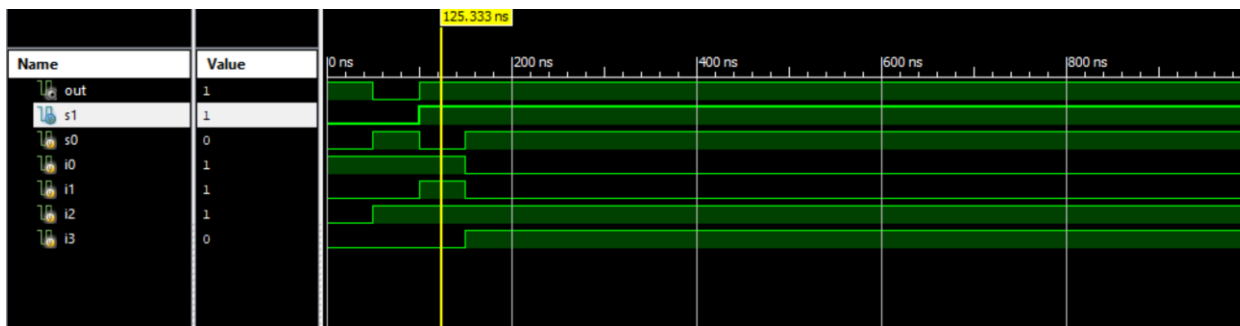
الف) $S0=0$, $S1=0$ و $I0=1$, $I1=I2=I3=0$ در این حالت باید $I0$ روی خروجی قرار گیرد



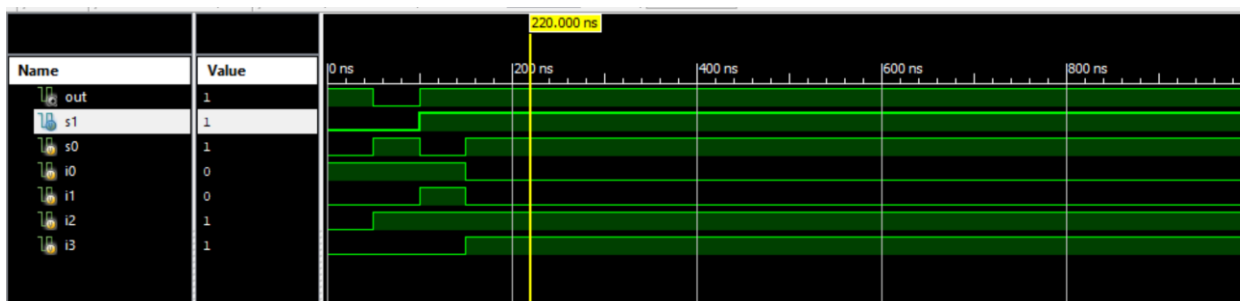
ب) $S0=1$, $S1=0$ و $I0=1$, $I1=0$, $I2=1$, $I3=0$ در این حالت باید $I1$ روی خروجی باشد



ج) $S0=0$, $S1=1$ و $I0=I1=I2=1$, $I3=0$ که باید $I2$ روی خروجی باشد

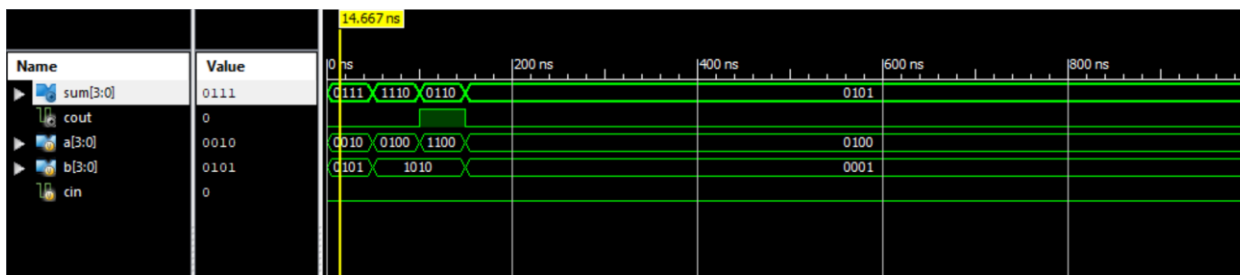


د) $S0=S1=1$ و $I2=I3=1$, $I0=I1=0$ که باید $I3$ روی خروجی باشد

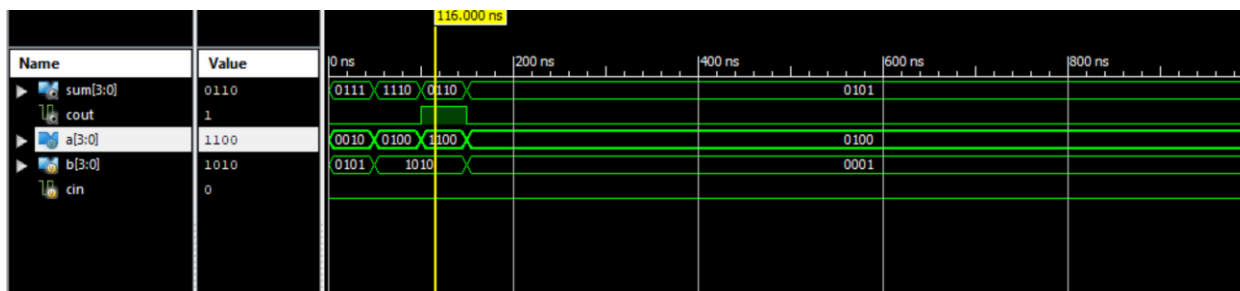


نتایج شبیه سازی جمع کننده چهاربیتی

جمع دو عدد $A=2$, $B=5$ که خروجی باید $7(0111)$ باشد



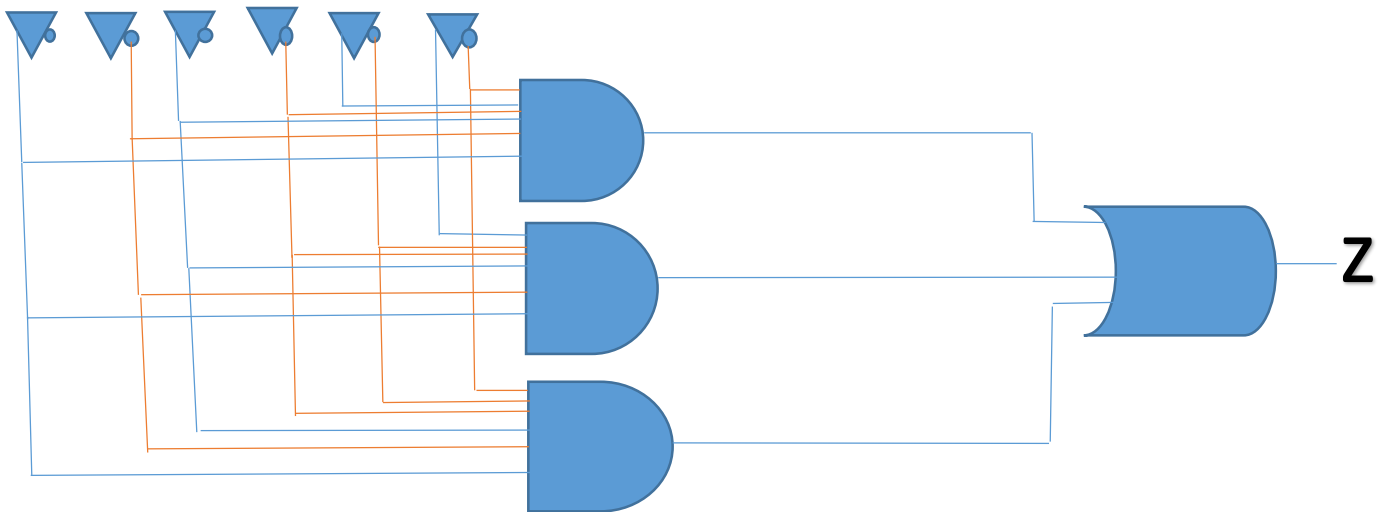
تفاضل $A=4$, $B=10$ که خروجی باید $6(0110)$ باشد



4) با توجه به دیاگرام داده شده می‌توانیم z را بصورت sum of product داده شده است بنویسیم توجه شود که

OFF بودن هر کلید را معادل با صفر بودن آن متغیر و On بودن را معادل با یک بودن آن کلید و متغیر در نظر داریم

$$z = \bar{a}.b.\bar{c}.d.\bar{e}.f + a.\bar{b}.\bar{c}.d.\bar{e}.f + \bar{a}.\bar{b}.\bar{c}.d.\bar{e}.f$$

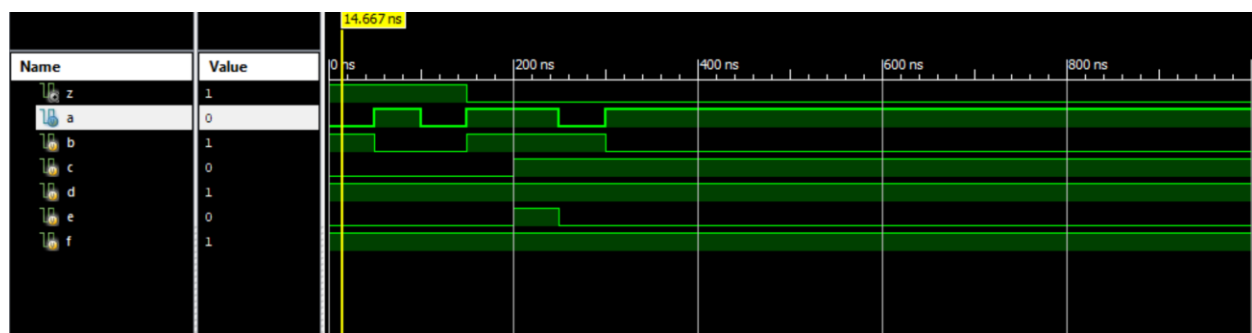


اما معادله ساده تر شده بصورت زیر است

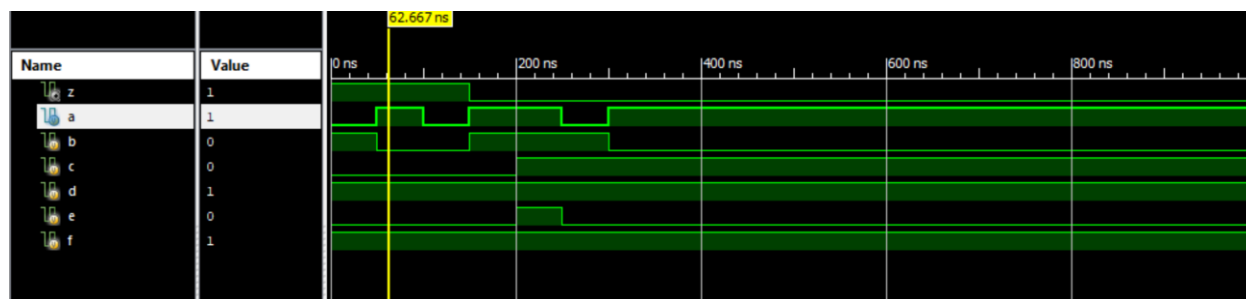
$$z = d.\bar{e}.f.\bar{b}.\bar{c} + d.\bar{e}.f.b.\bar{a}.\bar{c}$$

نتایج شبیه سازی

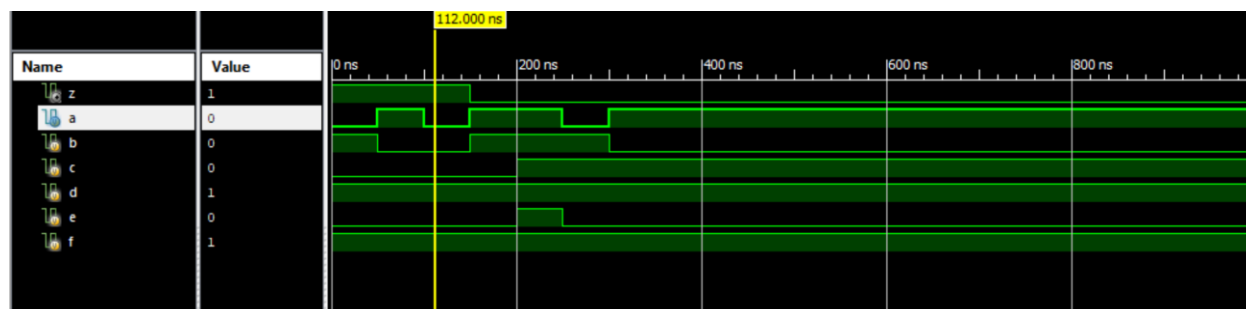
فروچی برای کد 010101



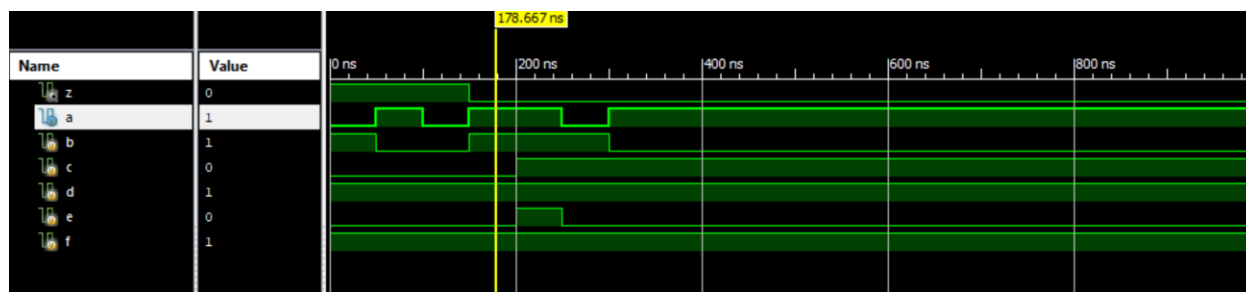
فروچی برای کد 100101



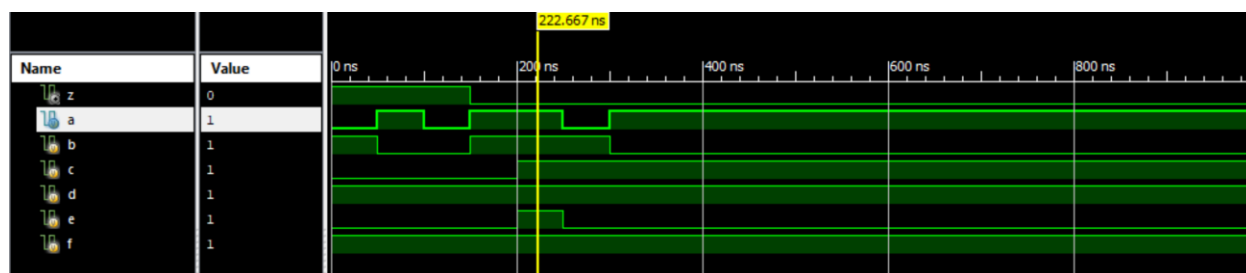
فروچی برای کد 000101



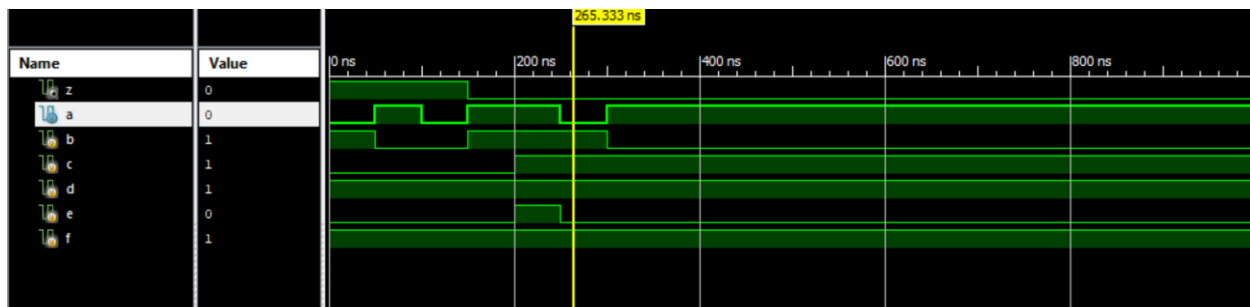
فروچی برای کد 110101



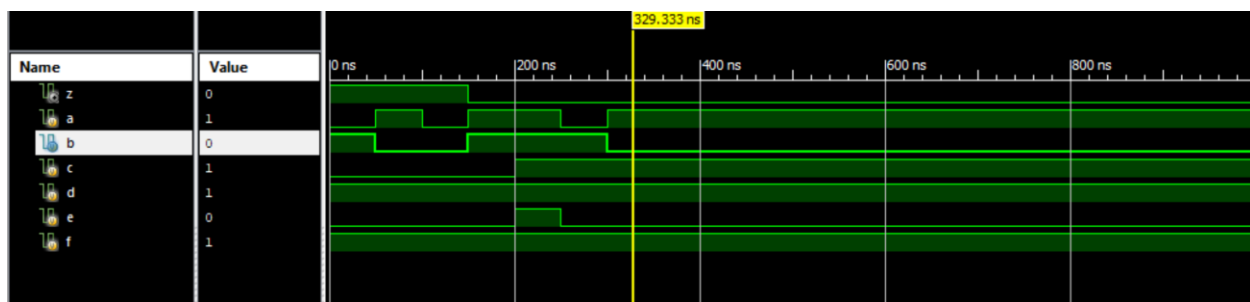
فروچی برای کد 111111



فروچی برای کد 011101



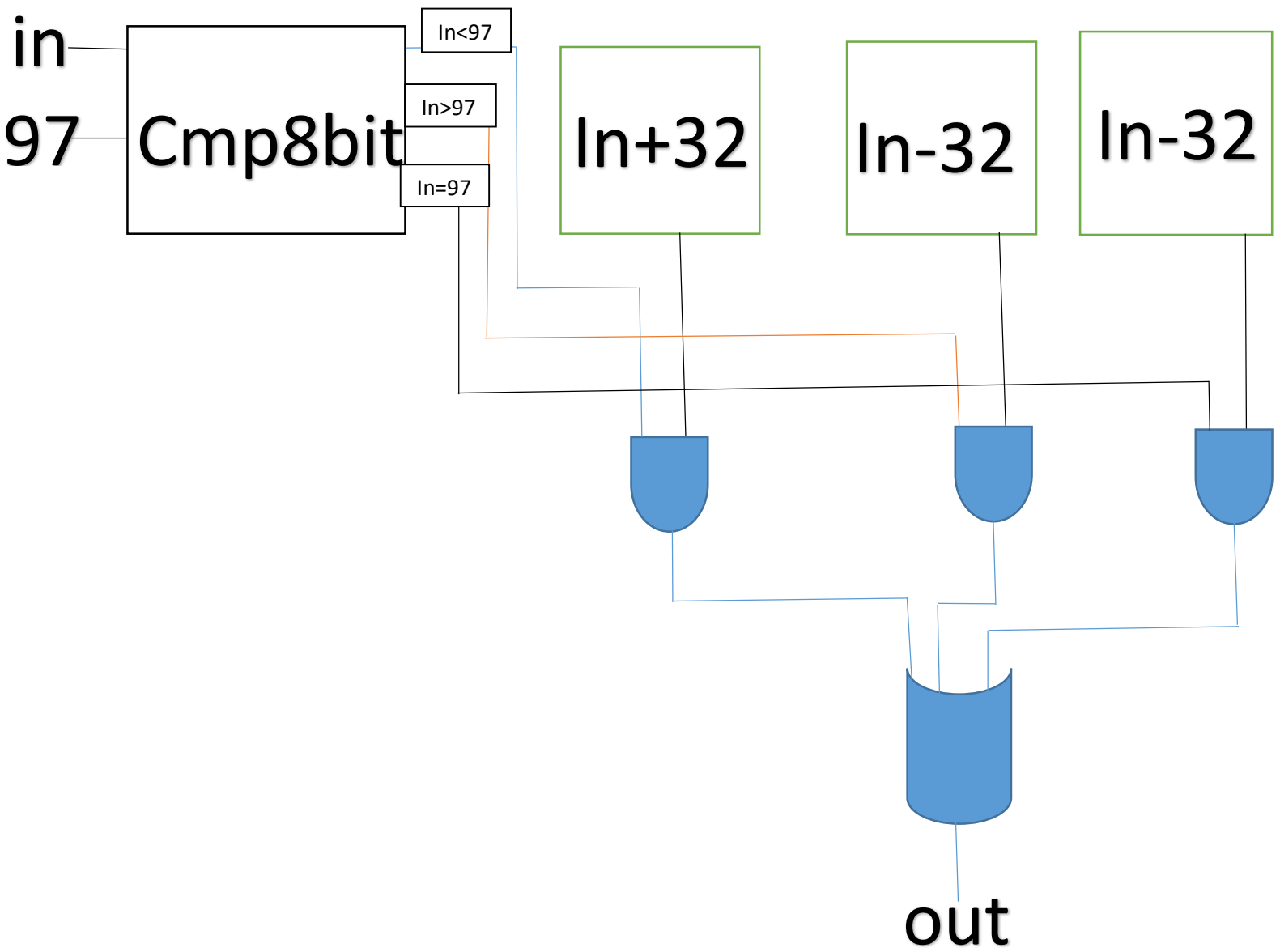
فرومی برای کد 101101



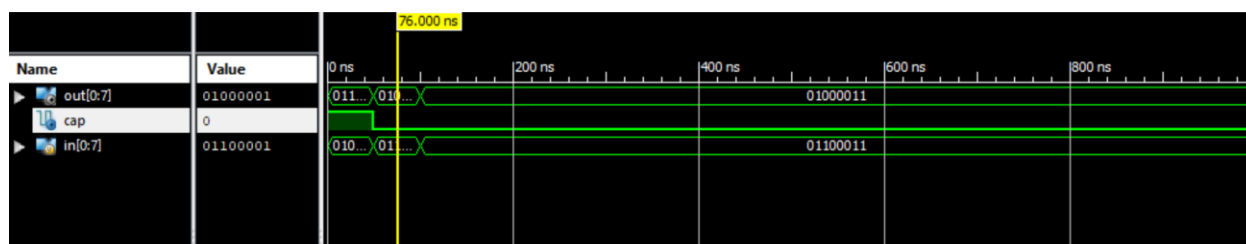
سوال (اختیاری) میدانیم کد اسکی های مروف کوچک و بزرگ 32 تا فاصله دارند بنابراین کد اسکی ورودی یا با این عدد جمع و یا 32 از آن کم میشود همچنین میدانیم مروف بزرگتر در ابتدا ینی کد اسکی کوچکتری دارند پس باید با 32 جمع شوند تا کد اسکی مرف کوچک را بدهند و مروف کوچک کد اسکی بزرگتر دارند پس باید از 32 کم شوند اگر فرض کنیم عدد وارد شده معتبر است باید با 97 که کد اسکی a است مقایسه شود تا مشخص شود اگر بزرگتر است مرف وارد شده مرف کوچک و اگر کوچکتر است مرف بزرگ است همچنین اگر خود 97 است باید A بدهد

برای فلگ cap در صورتی که فرومی کمتر مقایسه کننده فعال باشد یعنی مرف بزرگ باشد یک میشود و اگر فرومی بزرگتر مساوی مقایسه گر فعال باشد ینی مرف کوچک بوده است و فلگ صفر است

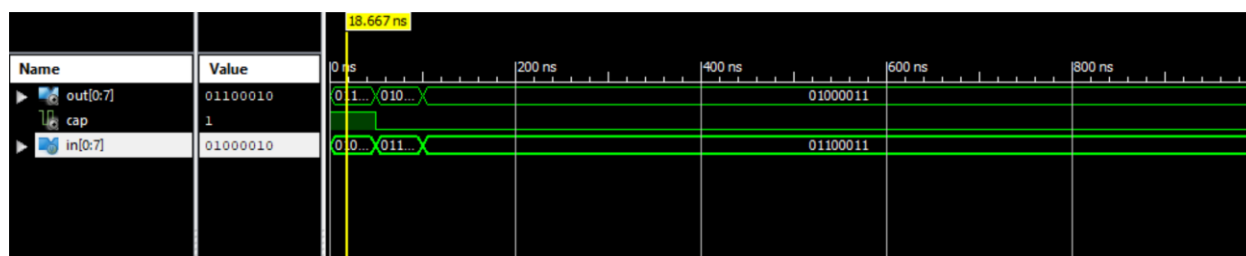
گیت های AND استفاده شده در فرومی های جمع کننده ها بعنوان پایه Enable عمل میکند تا اگر آن حالت با توجه به ورودی برقرار نیست در فرومی اثر نرارد و تنها یکی از فرومی های این سه جمع کننده به فرومی اصلی منتقل شوند



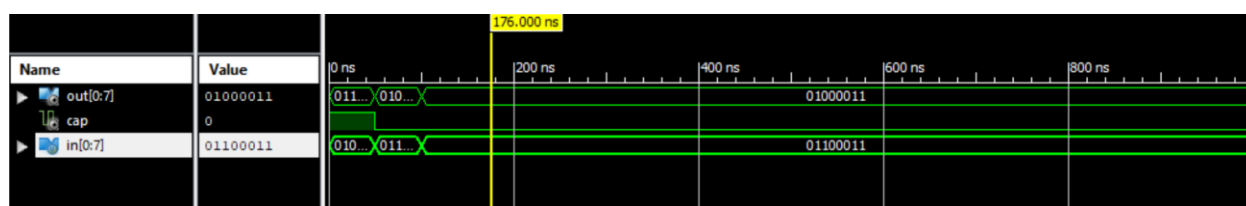
مرز مقایسه گر :فروچی برای مرف a با کد اسکی (01100001) 97 که انتظار داریم فلگ صفر شود و کد اسکی A که
 (01000001)65 است را در فروچی ببینیم



فعال شدن فروچی کمتر مقایسه گر :فروچی برای مرف B با کد اسکی (01000010) 66 که انتظار داریم فلگ یک شود و کد
 اسکی b که (01100010) 98 است را ببینیم

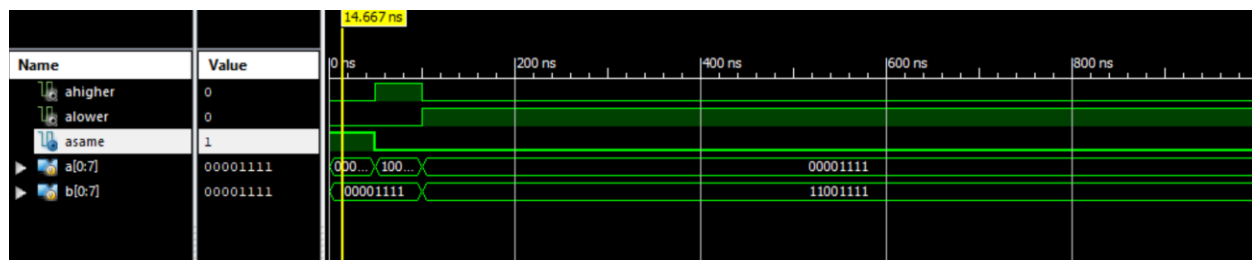


فعال شدن فروچی بیشتر مقایسه گر : فروچی برای c با کد اسکی (01100011) 99 که باید مرف بزرگتر یعنی C با کد اسکی
 (01000011)67 را برگرداند و فلگ صفر باشد

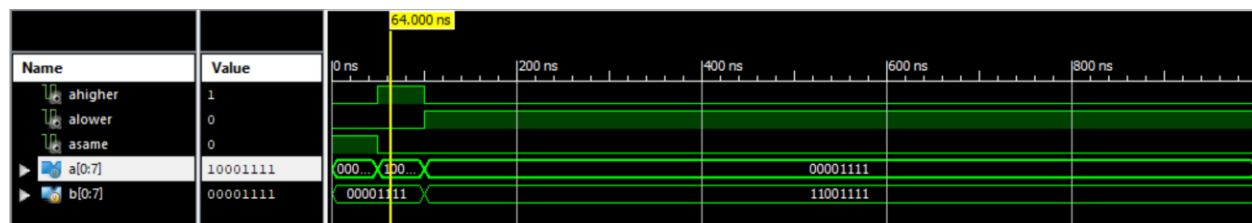


نتیجه شبیه سازی مقایسه گر هشت بیتی

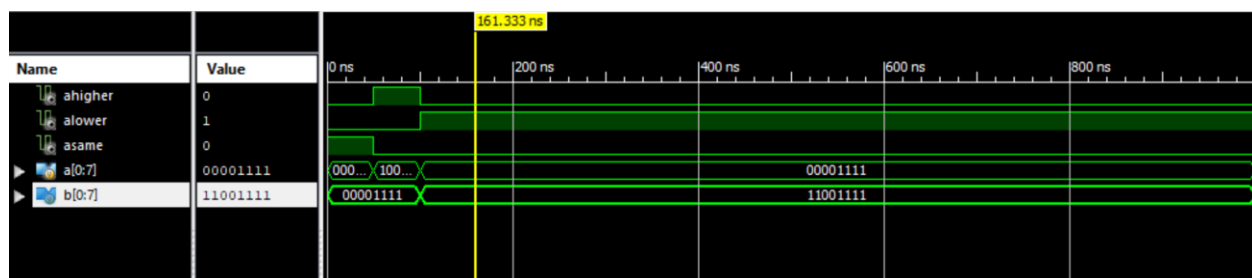
حالت تساوی برای 00001111



حالت a بزرگتر برای a= 10001111 و b=00001111

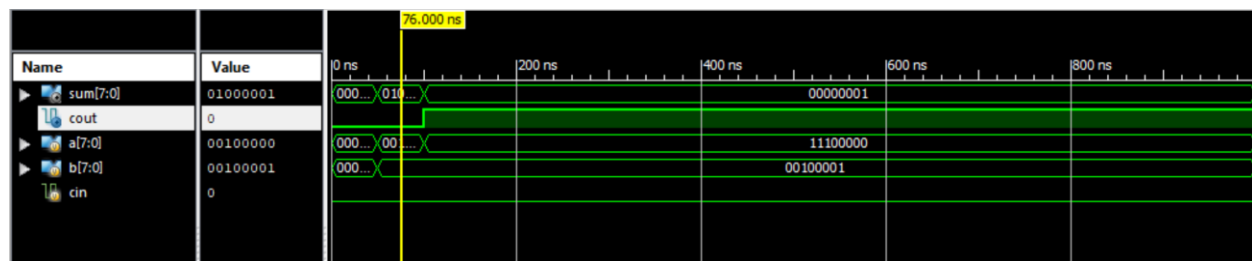


حالت a کوچکتر برای a=00001111 و b=11001111



نتیجه شبیه سازی جمع کننده هشت بیتی

جمع دو عدد a=32 و b=33 که مجموع آنها 65 است



تفریق دو عدد 33 و 32 که یک است

