آزمایش 5

نام و نام خانوادگی: هلیا سادات هاشمی پور، روژینا کاشفی

شماره دانشجویی:9831106، 9831118

نام استاد:استاد خجسته دانا

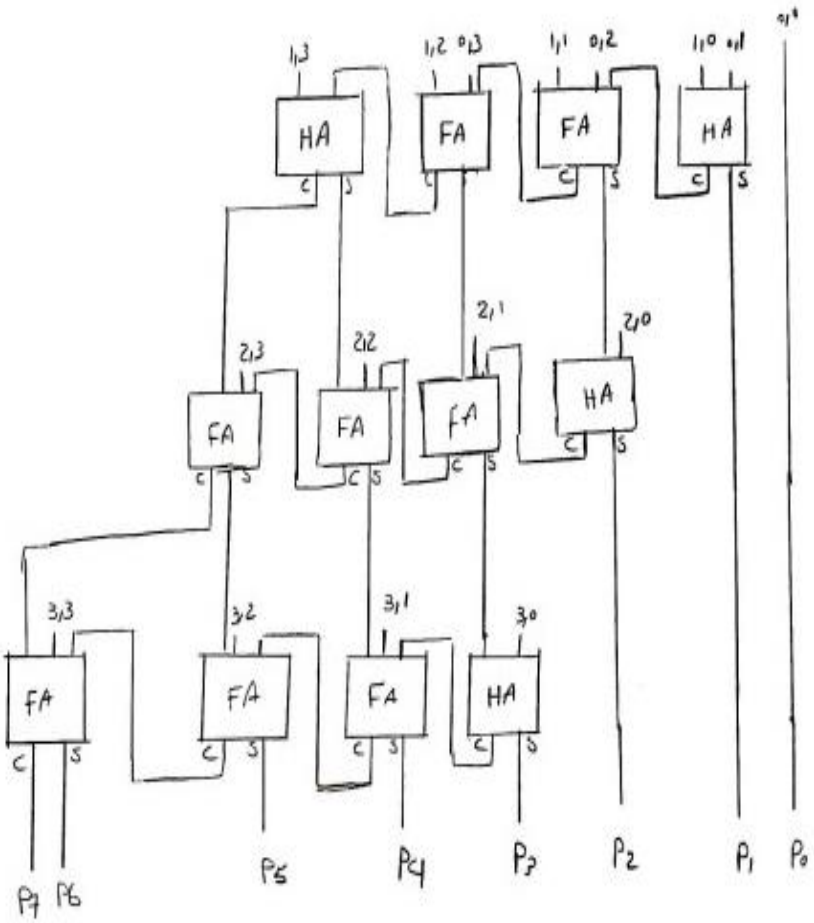
تاریخ:8 اردیبهشت 1400

هدف آزمایش: پیاده سازی انواع ضرب کننده های 4 بیتی

- سه ضرب کننده ی اول(الف)ضرب کننده معمولی و ب) ضرب کننده آرایه ای و ج)ضرب کننده carry save adder)در سیستم بی علامت هستند و ضرب بوث در سیستم علامتدار (مکمل دو)است.

الف) ضرب کننده معمولی(Basic multiplier)

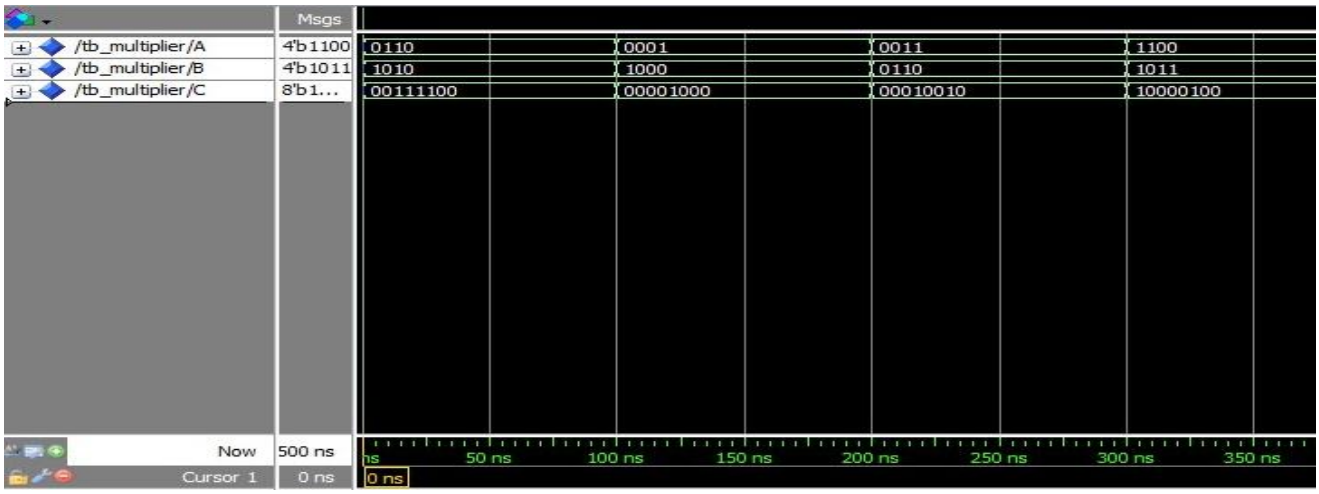
این ضرب کننده همانند ضرب اعداد مبنای 10می باشد.یعنی ارقام ضرب کننده را در ضرب شونده ضرب میکنیم و هر بار حاصل ضرب جزیی را شیفت به چپ داده و با حاصل قبلی جمع می کنیم.



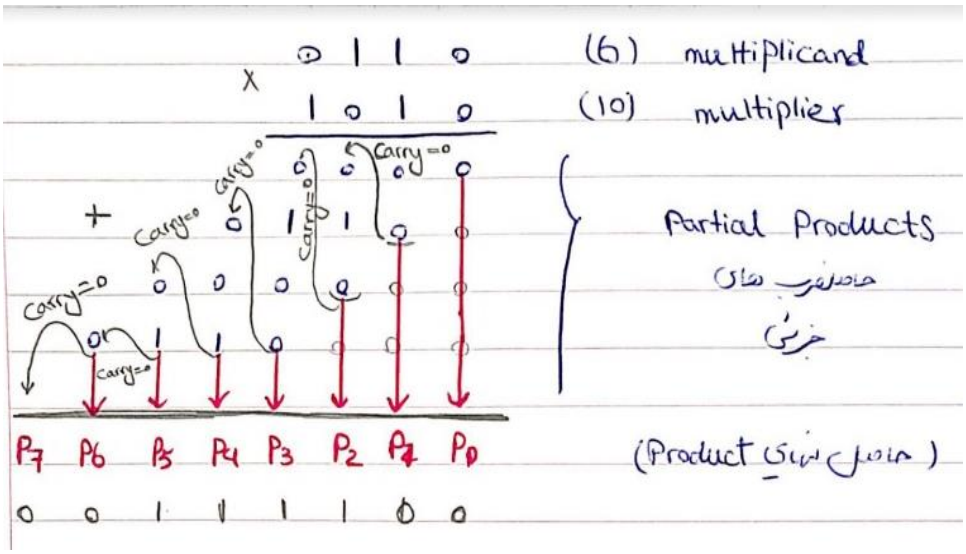
$$Cost = n^2 + (n-1)(5n-3) - 3 = 64\ gate$$

$$Delay = 14d$$

شکل موج آن به صورت زیر است:



برای 0110 و 1010 :

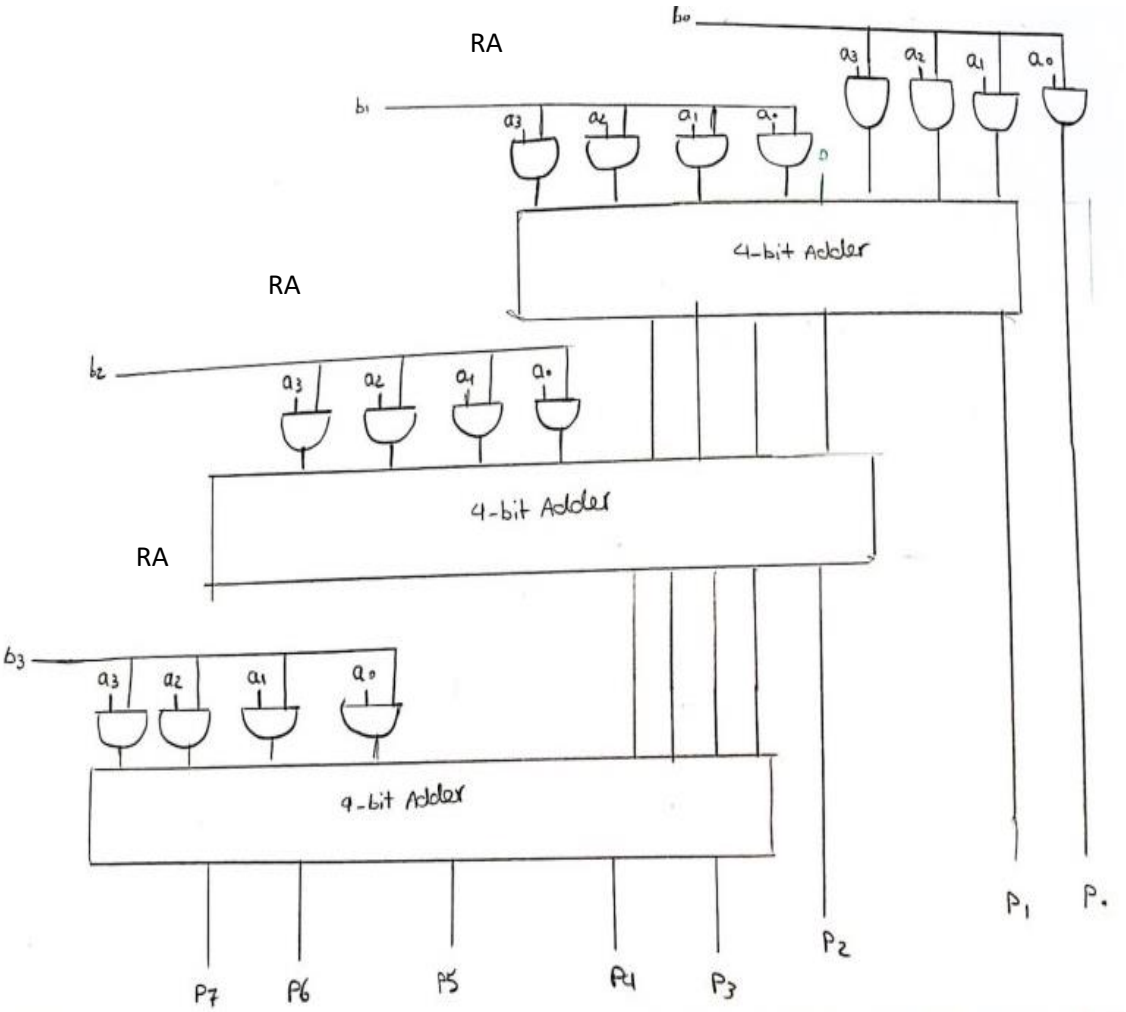


ب)ضرب کننده آرایه ای

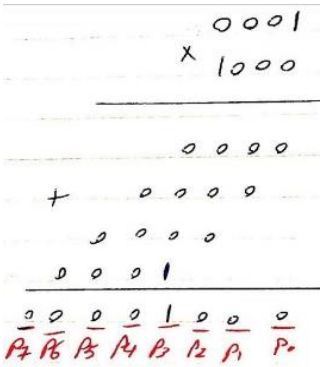
در این ضرب کننده از واحد های جمع کننده 4 *بیتی استفاده شده است.*

$$\boldsymbol{Cost = n^2 + (n-1)(5n) = 76gate}$$

$$\boldsymbol{Delay = 25d}$$



شکل موج آن به صورت زیر است.



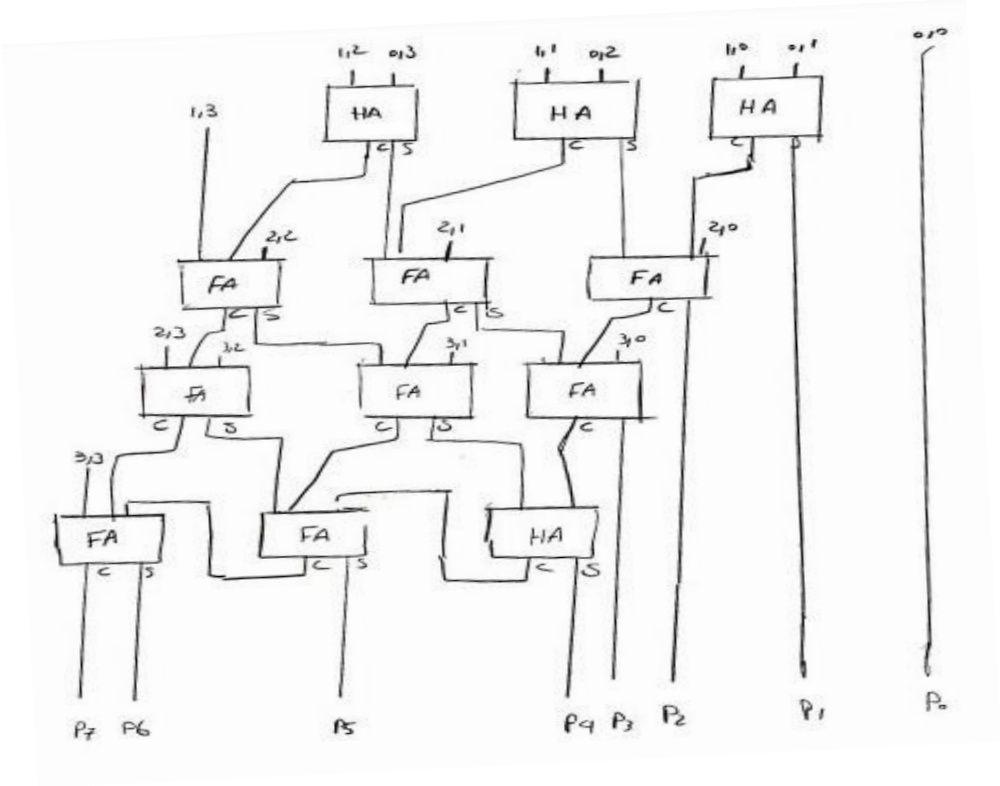
خروجی برای 0001 و 1000:

اگر بخواهیم طبق کارکرد شکل این قسمت توضیح بدهیم و ابتدا a0 با b0 and (0)می شود و p0 را میدهد.سپس b0 با a1 (0)سپس با a2 (0)و سپس با a3 (0) and می شود و به 4-bit adder اول داده می شود. b1 با a0 (0)سپس با a1 (0)سپس با a2 (0) و سپس با a3 (0) and می شود و به 4-bit adder اول داده می شود. p1 را به خروجی می دهد(0).حاصل جمع آن ها (0000)به 4-bit adder دوم داده می شود و b2با a0 (0)سپس با a1 (0)سپس با a2 (0)و سپس با a3 (0) and می شود و به 4-bit adder دوم داده می شود. p2را خروجی می دهد(0).

حاصل جمع آن ها (0000)به 4-bit adder سوم داده می شود و b3با a0 (1)سپس با a1 (0)سپس با a2 (0)و سپس با a3 (0) and می شود و به 4-bit adder سوم داده می شود.سپس حاصل جمع آن ها به ترتیب p3(1) و p4(0)و p5(0) و p6(0) و p7(0) را خروجی می دهد.

ج)ضرب کننده carry_save_adder

در این ضرب کننده بیت تقلی به مرحله ی بعد منتقل می شود.

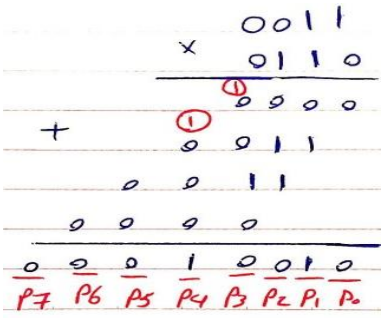


$Cost = 64gate$

$Delay = 11d$

شکل موج آن به صورت زیر است.

خروجی برای 0011 و 0110:



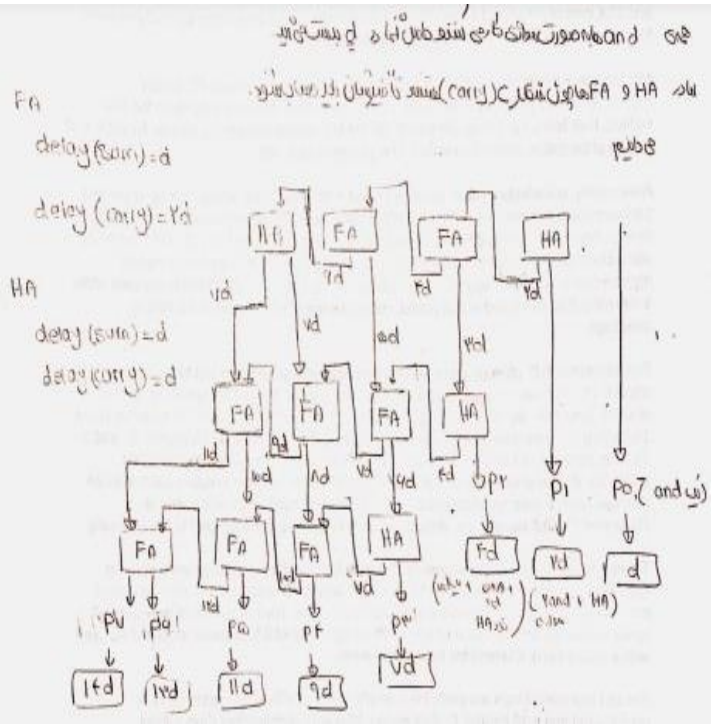
نتیجه

همانطور که می بینیم هزینه(cost)

carry_save_adder ضرب کننده = Basic Multiplier (ضرب کننده معمولی)< Array Multiplier(ضرب کننده آرایه ای)

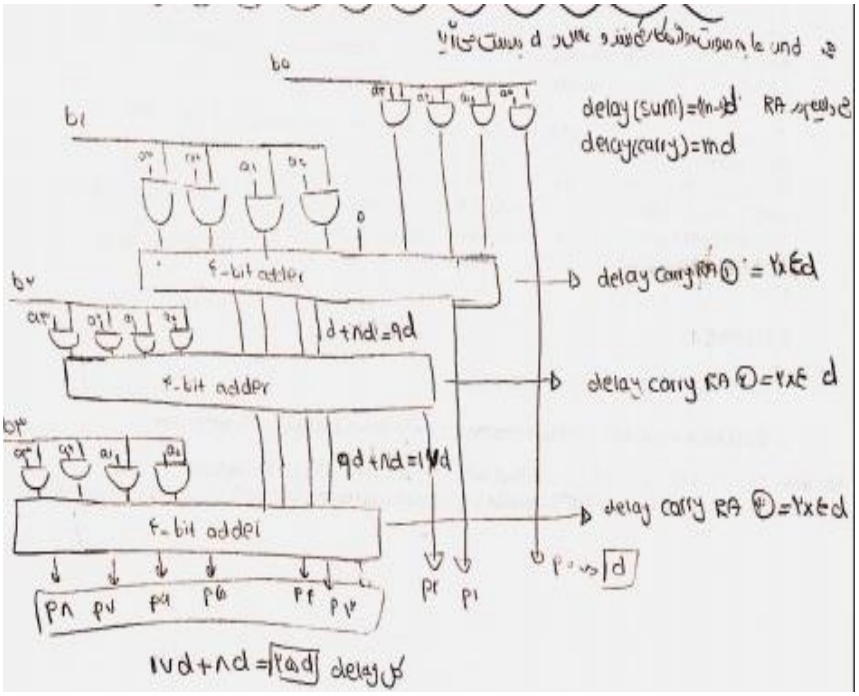
در واقع هزینه ی ضرب کننده ی معمولی((16 *تا گیت* and *و* 4 *تا* HA (*هرکدام* 2g) *و* 8 *تا* FA (*هرکدام* 5g)) *با ضرب کننده ی* carry save adder ((16 *تا گیت* and *و* 4 *تا* HA (*هرکدام* 2g) *و* 8 *تا* FA (*هرکدام* 5g)) 64g*است که با هم برابر می شود.*
ضرب کننده آرایه ای دارای سه RA *است که هر کدام* 5ng (*یعنی* 20g *برای هرکدام که در مجموع می شود* 60g) *هستند و دارای* 16 *تا گیت* and (16g) *است که در مجموع* 76g *می شود که از دو تا ضرب کننده ی قبلی هزینه ی بیش تری دارد.*

همانطور که می بینیم تاخیر(delay) ها به صورت زیر است.



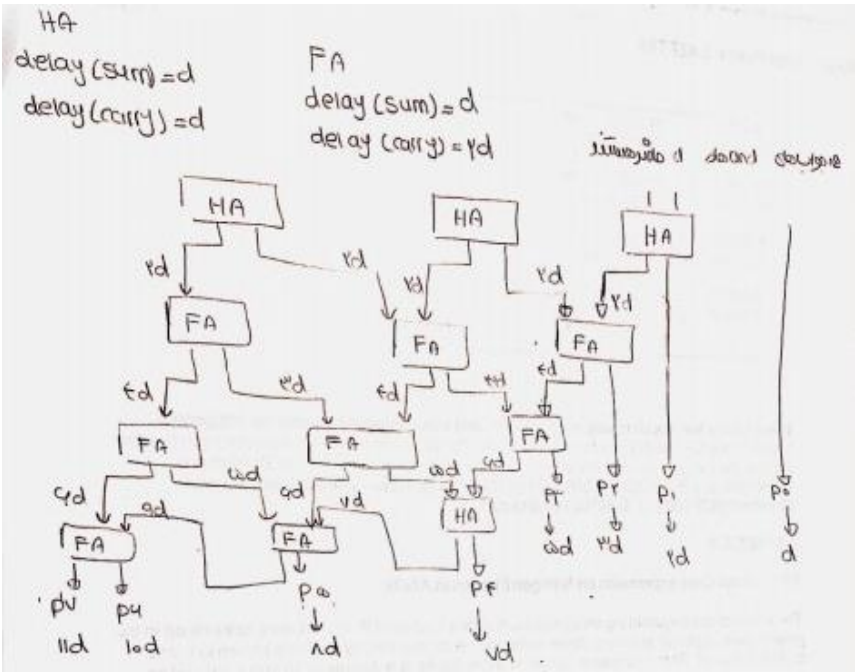
Basic multiplier:

Delay = 14d



Array multiplier:

Delay = 25d



Carry save adder multiplier:

Delay = 11d

همانطور که دیدیم تاخیر ضرب کننده ها به صورت زیر است:

carry_save_adder ضرب کننده **> Basic Multiplier** (ضرب کننده معمولی **)> Array Multiplier** (ضرب کننده آرایه ای)

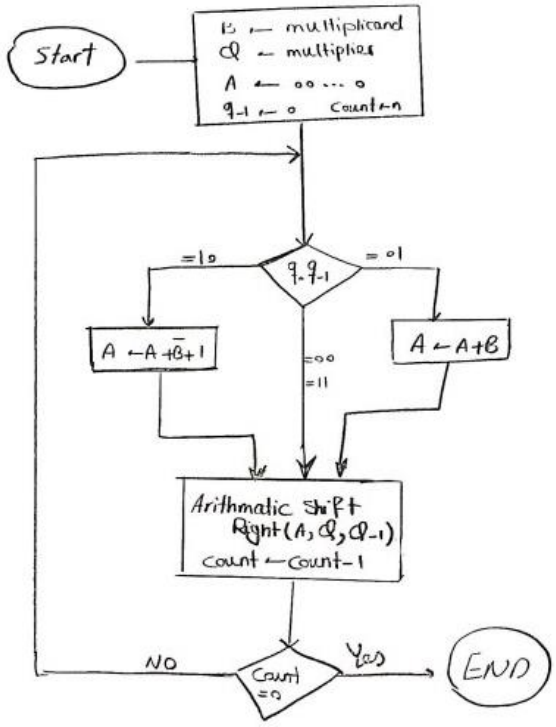
در واقع تاخیر ضرب کننده **carry select adder** از همه کم تر و تاخیر ضرب کننده آرایه ای از همه بیش تر است.بنابراین **carry save adder** بهتر است زیرا بیت نقلی به مرحله ی بعدی خود منتقل می شود.

نتیجه: از تاخیر و هزینه های به دست آمده می توان نتیجه گرفت که **carry save adder** از دو تای دیگر (از هر دو جهت) بهتر است.

د)ضرب کننده Booth(اختیاری)

الگوریتم ضرب بوث با استفاده از یک الگوریتم ساده دو عدد علامت دار را در یکدیگر ضرب می کند.الگوریتم ضرب بوث در واقع رویه ای است برای ضرب اعداد binary در سیستم مکمل دو است.در واقع در این نوع ضرب کننده که از روشهای ضرب سریع محسوب میشود ضربهای جزئی کمتری خواهیم داشت که میتوان این ضربهای جزئی را به روشهای مختلف با هم جمع کرد. این روش هر چند روند ضرب را سریعتر میکند ولی آن را پیچیده تر خواهد کرد. تفاوت اصلی این روش با روشهای قبل این است که این روش به صورت ترتیبی است .این ضرب کننده روش ضرب سرع است و سریع تر عمل می کند اما پیچیده تر است.

فلوچارت ای ضرب کنند به صورت زیر است:



Bضرب شونده، Q ضرب کننده و حاصل A:Q است . بیت سمت راست Q را q0 و یک بیت q_{-1} سمت راست Q با مقدار اولیه ی صفر تعریف می کنیم.الگوریتم به این صورت است که n بار (به تعداد بیت های Q)مقدار $q_0 q_{-1}$ را بررسی می کنیم.

1-اگر 10 باشد عمل تفریق $A \leftarrow A - B(A + \overline{B} + 1)$ انجام می شود.

2- اگر 01 باشد عمل جمع $A \leftarrow A + B$ انجام می شود.

3-اگر 00 یا 11 باشد هیچ عملی انجام نمی شود.

پس از انجام عملیات بالا A و Q و q_{-1} یک شیفت به سمت راست داده می شوند.

شکل موج خروجی:

توضیح درمورد خروجی:

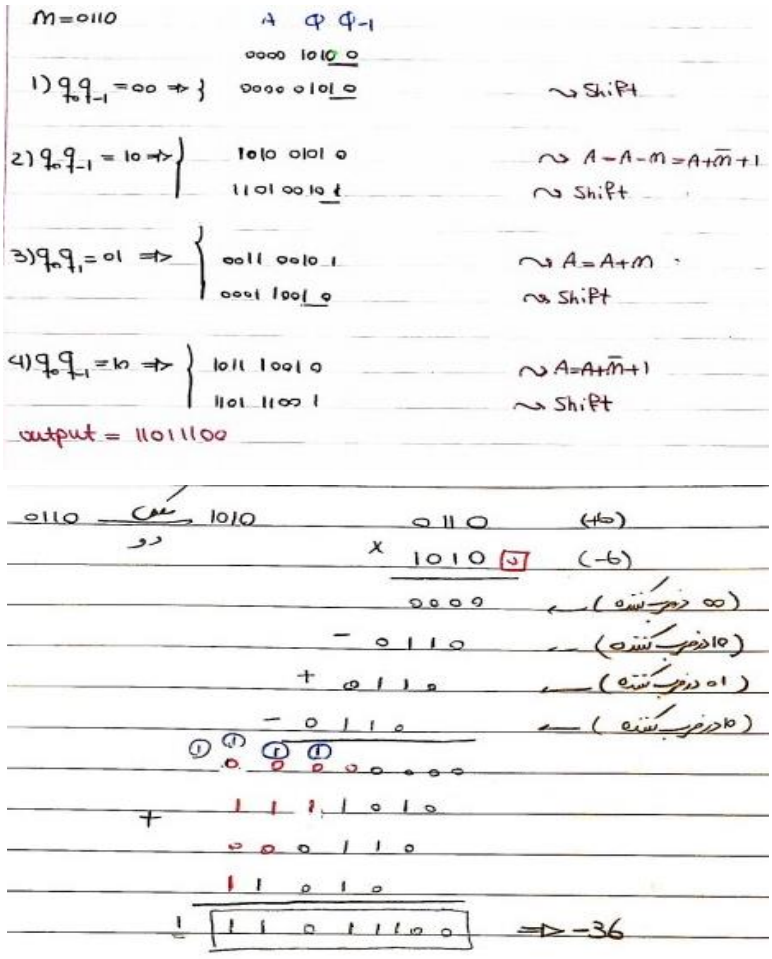
برای 6 و -6 (0110 و 1010)؛ (به روش ترتیبی با ثبات های **4** بیتی)

-1 می بینیم که $q_0q_{-1} = 00$ است بنابراین فقط شیفت به راست را می دهیم.

-2 بعد از مرحله ی 1 می بینیم که $q_0q_{-1} = 10$ است پس $A \leftarrow A - M$ را حساب کرده و سپس یک شیفت به سمت راست می دهیم.

-3 بعد از شیفت می بینیم که $q_0q_{-1} = 01$ است پس $A \leftarrow A + M$ را حساب کرده و یک شیفت به سمت راست می دهیم.

-4 بعد از مرحله ی 3 می بینیم که $q_0q_{-1} = 10$ است پس $A \leftarrow A - M$ را حساب کرده و سپس یک شیفت به سمت راست می دهیم.



به شکل دیگر:

سمت راست ضرب کننده یک صفر بی ارزش قرار می دهیم سپس از سمت راست ضرب کننده حرکت می کنیم اگر الگوی 00 یا 11 دیدیم صفر می نویسیم.هر جا الگوی 10 دیدیم ضرب شونده را با علامت منفی می نویسیم اگر الگوی 01 دیدیم ضرب شونده را با علامت مثبت می نویسیم.البته با هر حرکت به سمت چپ، شیفت نیز می دهیم سپس حاصل ضرب های جزیی که دارای علامت منفی هستند را مکمل دو کرده و همه ی حاصلضرب های جزیی را جمع می کنیم.

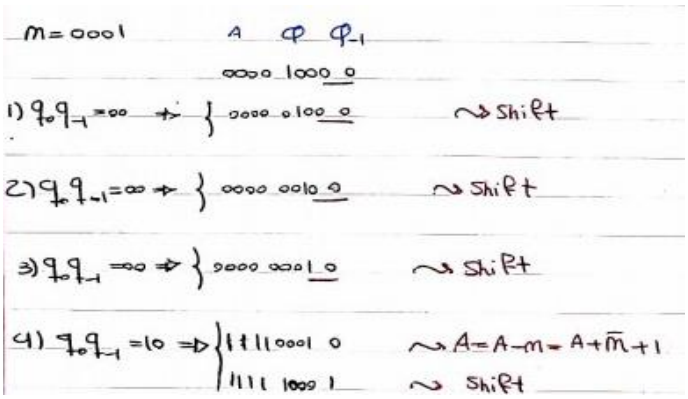
برای 1 و -8 (0001 و 1000) (به روش ترتیبی با ثبات های 4 بیتی)

1- می بینیم که $q_0q_{-1} = 00$ است بنابراین فقط شیفت به راست را می دهیم.

2-بعد از مرحله ی 1 می بینیم که $q_0q_{-1} = 00$ است بنابراین فقط شیفت به راست را می دهیم.

3-بعد از شیفت می بینیم که $q_0q_{-1} = 00$ است بنابراین فقط شیفت به راست را می دهیم.

4-بعد از مرحله ی 3 می بینیم که $q_0q_{-1} = 10$ است پس $A \leftarrow A - M$ را حساب کرده و سپس یک شیفت به سمت راست می دهیم.



Output=11111000