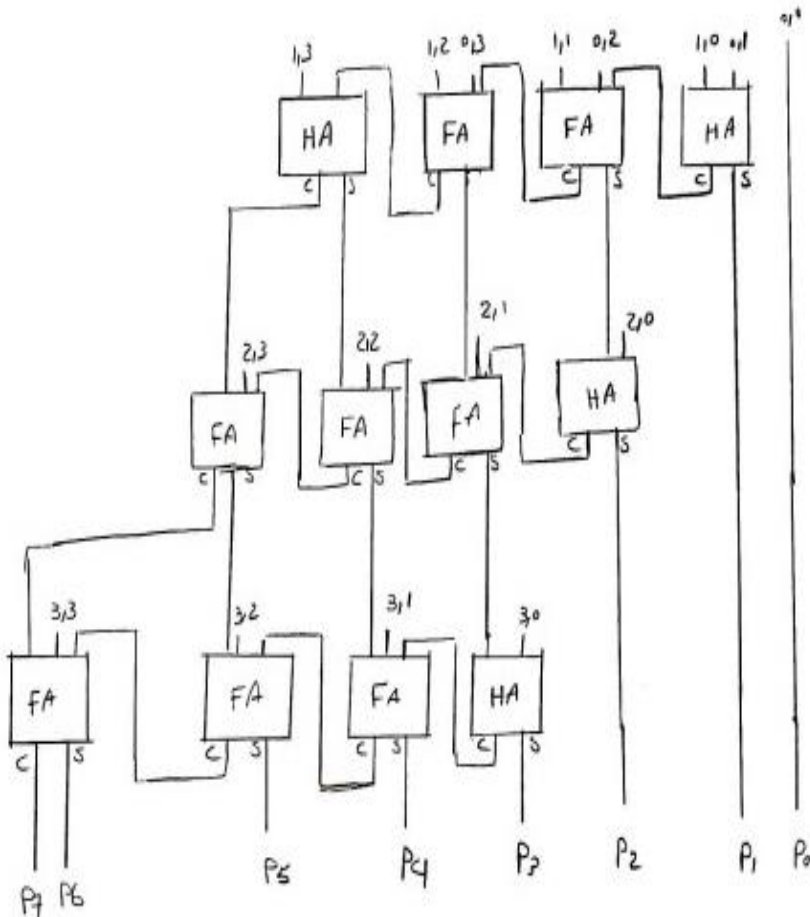


هدف آزمایش: پیاده سازی انواع ضرب کننده های 4 بیتی

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)

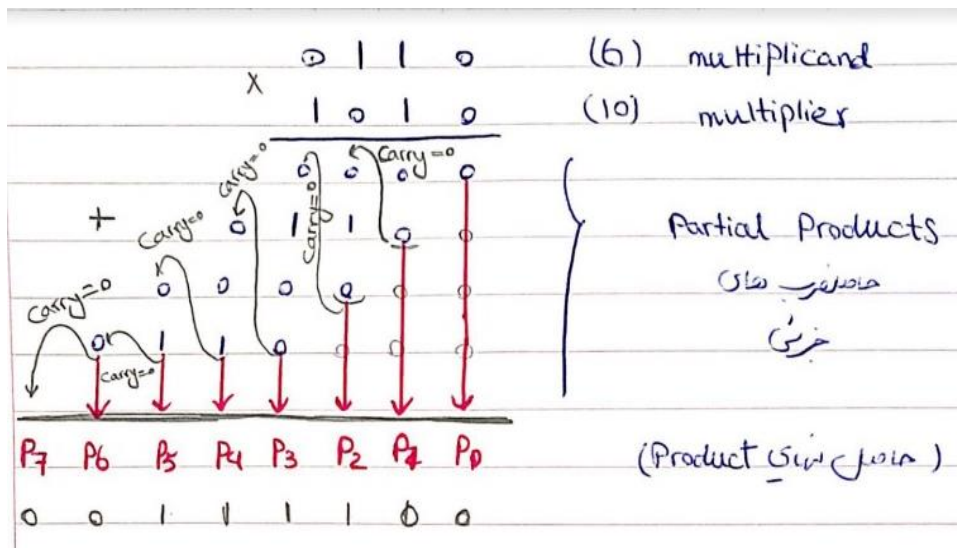
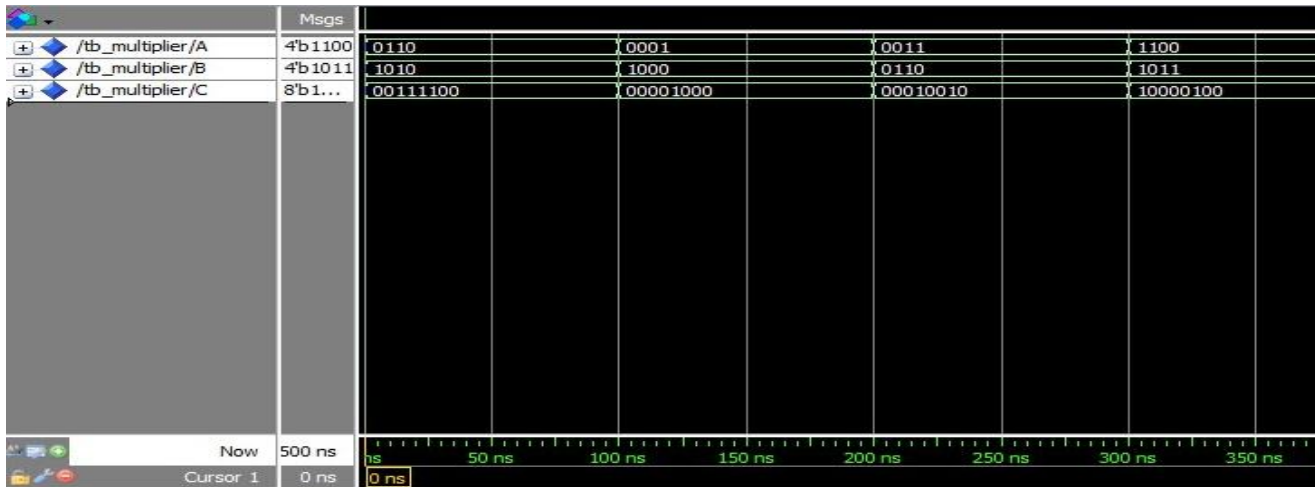
این ضرب کننده همانند ضرب اعداد مبنای 10 می باشد. یعنی ارقام ضرب کننده را در ضرب شونده ضرب میکنیم و هر بار حاصل ضرب چیزی را شیفت به چپ داده و با حاصل قبلی جمع می کنیم.



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

$$\text{Delay} = 14d$$

شکل موج آن به صورت زیر است:



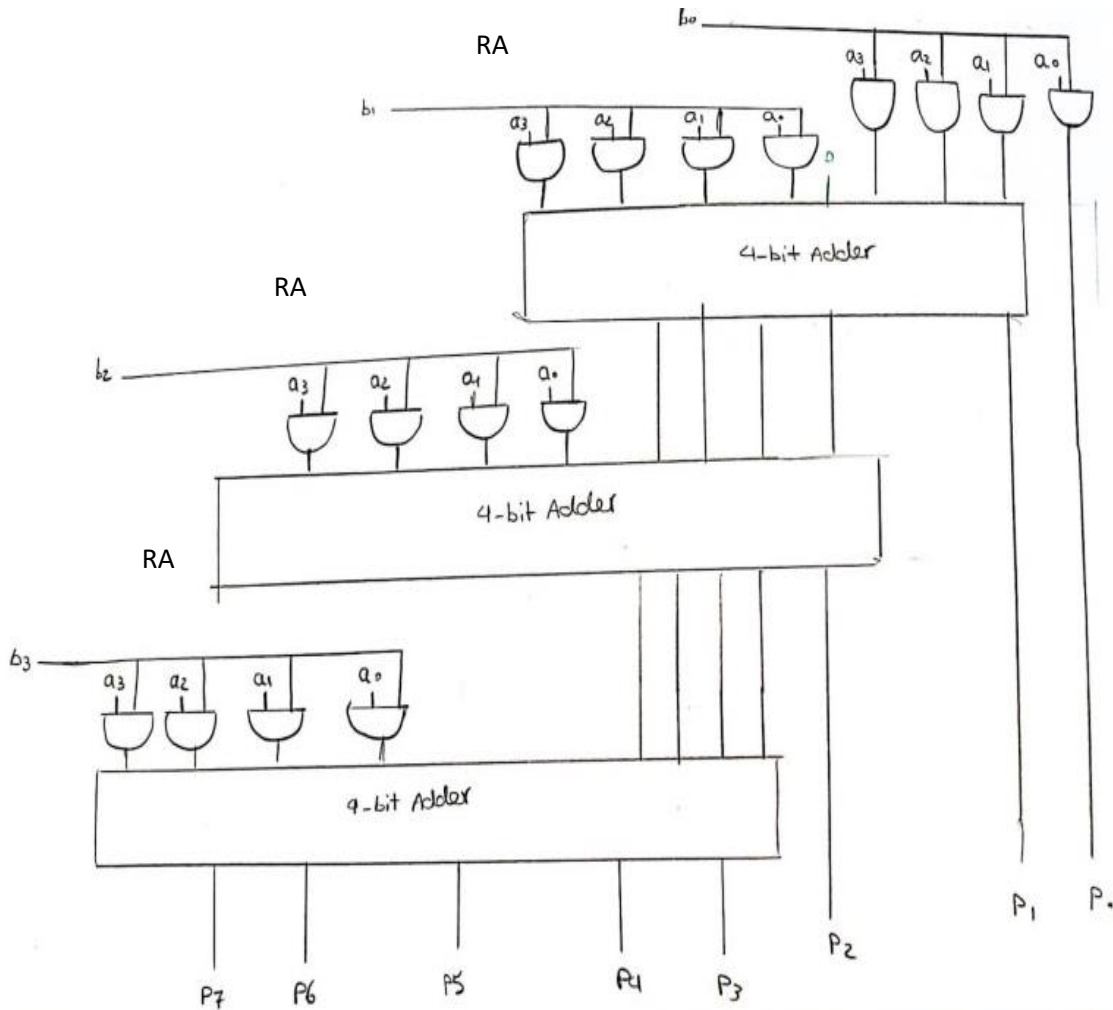
برای 0110 و 1010 :

ب) ضرب کننده آرایه ای

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

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

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



شکل موج آن به صورت زیر است.



فروپیی برای 0001 و 1000:

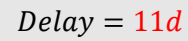
$$\begin{array}{r}
 0001 \\
 \times 1000 \\
 \hline
 0000 \\
 + 0000 \\
 0000 \\
 0000 \\
 \hline
 00001000
 \end{array}$$

$P_7 \ P_6 \ P_5 \ P_4 \ P_3 \ P_2 \ P_1 \ P_0$

اگر بفوایم طبق کارکرد شکل این قسمت توضیح بدهیم و ابتدا a_0 با b_0 and b_0 (0) می شود و p_0 را می دهیم. سپس a_1 با b_0 (0) سپس با a_2 (0) و سپس با a_3 (0) and می شود و به 4-bit adder اول داده می شود. b_1 با a_0 (0) سپس با a_1 (0) سپس با a_2 (0) و سپس با a_3 (0) and می شود و به 4-bit adder اول داده می شود. p_1 را به فروپیی می دهیم (0). حاصل جمع آن ها (0000) به 4-bit adder دوم داده می شود و b_2 با a_0 (0) سپس با a_1 (0) سپس با a_2 (0) و سپس با a_3 (0) and می شود و به 4-bit adder دوم داده می شود. p_2 را فروپیی می دهیم (0).

حاصل جمع آن ها (0000) به 4-bit adder سوم داده می شود و b_3 با a_0 (1) سپس با a_1 (0) سپس با a_2 (0) و سپس با a_3 (0) and می شود و به 4-bit adder سوم داده می شود. سپس حاصل جمع آن ها به ترتیب p_3 (1) و p_4 (0) و p_5 (0) و p_6 (0) و p_7 (0) را فروپیی می دهیم.

در این ضرب‌کننده بیت ثقلی به مرحله ی بعد منتقل می شود.



+	/tb_carry_save_adder/A	-No Data-	0110	0001	0011	1100
+	/tb_carry_save_adder/B	-No Data-	1010	1000	0110	1011
+	/tb_carry_save_adder/C	-No Data-	00111100	00001000	00010010	10000100
Now			400 ns			
Cursor 1			408 ns			

[illegible]

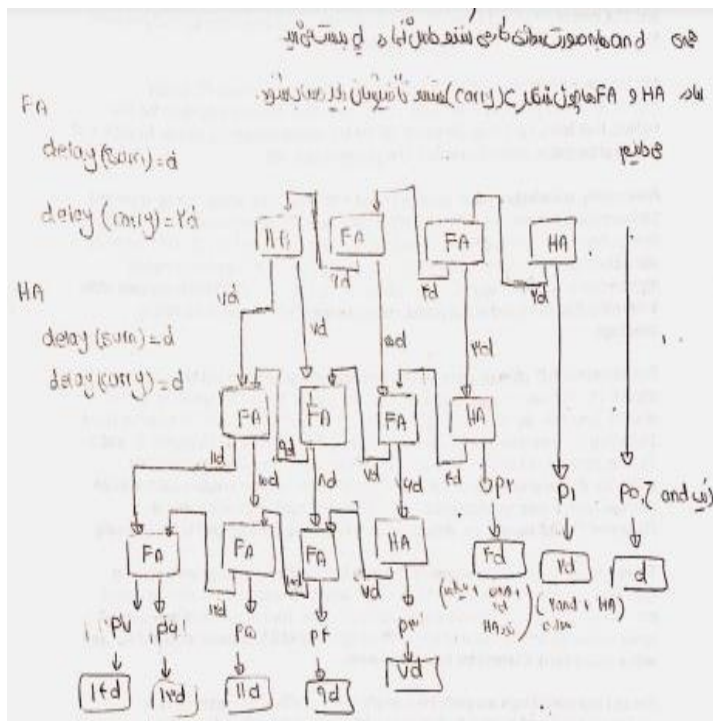
نتیجه

همانطور که می بینیم هزینه (cost)

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

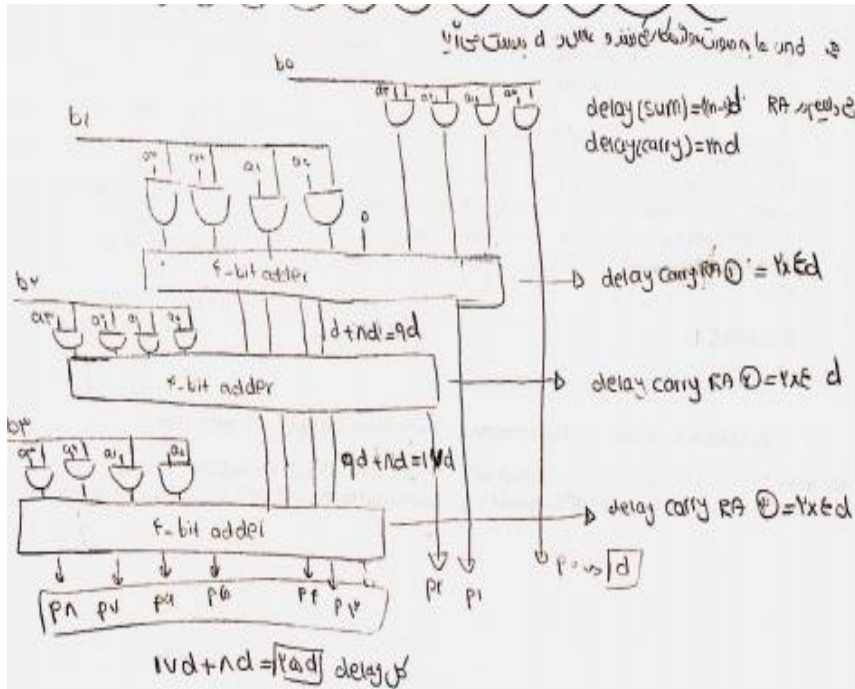
در واقع هزینه ی ضرب کننده ی معمولی ((16 تا کیت and و 4 تا HA (هرکدام 2g) و 8 تا FA (هرکدام 5g)) با ضرب کننده ی carry save adder ((16 تا کیت and و 4 تا HA (هرکدام 2g) و 8 تا FA (هرکدام 5g)) 64g است که با هم برابر می شود. ضرب کننده آرایه ای دارای سه RA است که هرکدام 5ng (یعنی 20g برای هرکدام که در مجموع می شود 60g) هستند و دارای 16 تا کیت 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.



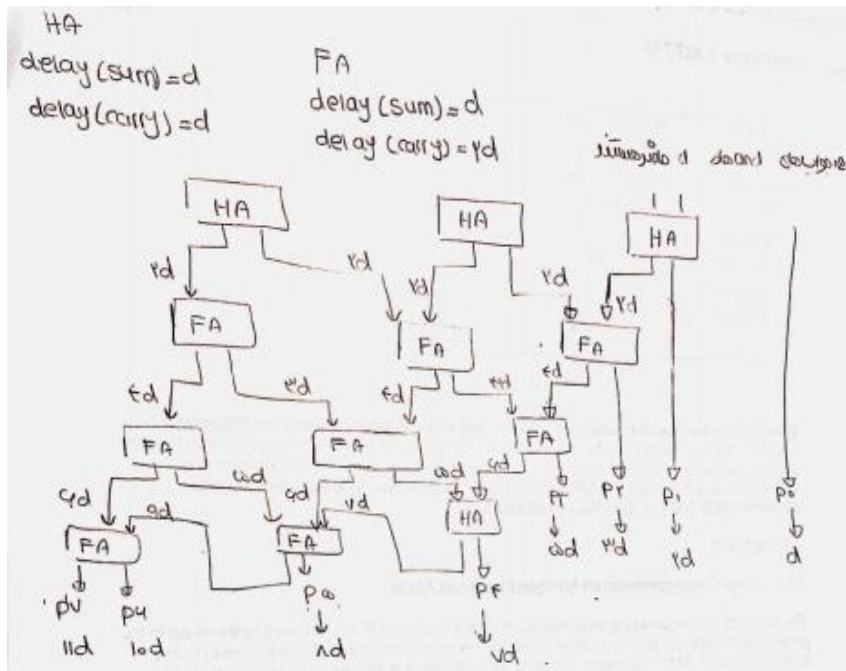
Basic multiplier:

Delay = 14d



Array multiplier:

Delay = 25d



Carry save adder multiplier:

Delay = 11d

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

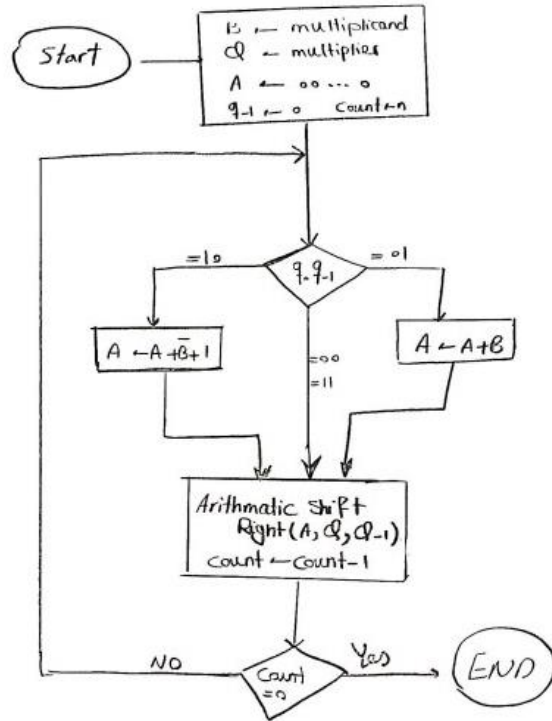
carry_save_adder ضرب کننده > **Basic Multiplier** (ضرب کننده معمولی) > **Array Multiplier** (ضرب کننده آرایه ای)

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

نتیجه: از تأخیر 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 مصوب میشود ضربهای جزئی کمتری خواهیم داشت که میتوان این ضربهای جزئی را به روشهای مقلف 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.



فلوپارت ای ضرب کنند به صورت زیر است:

B ضرب شونده، Q ضرب کننده و حاصل A:Q است. بیت سمت راست Q، q_0 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 $A \leftarrow A - B(A + \bar{B} + 1)$ 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 فروبی:



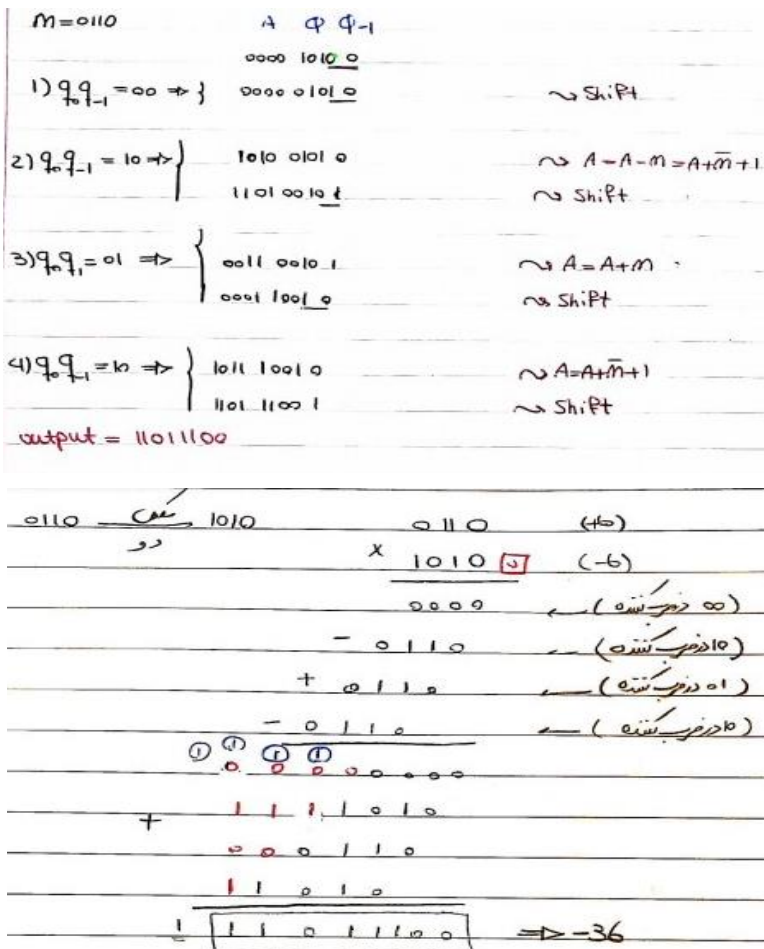
توضیح در مورد فروبی:

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

- 1- می بینیم که $q_0 q_{-1} = 00$ است بنابراین فقط شیفت به راست را می دهیم.
- 2- بعد از مرحله ی 1 می بینیم که $q_0 q_{-1} = 10$ است پس $A \leftarrow A - M$,
حساب کرده و سپس یک شیفت به سمت راست می دهیم.
- 3- بعد از شیفت می بینیم که $q_0 q_{-1} = 01$ است پس $A \leftarrow A + M$,
حساب کرده و یک شیفت به سمت راست می دهیم.
- 4- بعد از مرحله ی 3 می بینیم که $q_0 q_{-1} = 10$ 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 داریم صفر می نویسیم. هر با الگوی 10 داریم ضرب شونده را با علامت منفی می نویسیم اگر الگوی 01 داریم ضرب شونده را با علامت مثبت می نویسیم. البته با هر حرکت به سمت چپ، شیفت نیز می دهیم سپس حاصل ضرب های جزئی که دارای علامت منفی هستند را مکمل دو کرده و همه ی حاصلضرب های جزئی 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.

$$\begin{array}{l}
 m = 0001 \quad A \quad q_{-1} \\
 \quad \quad \quad 0000 \quad 1000 \quad 0 \\
 1) q_0q_{-1} = 00 \rightarrow \left\{ \begin{array}{l} 0000 \quad 0100 \quad 0 \\ \sim \text{Shift} \end{array} \right. \\
 2) q_0q_{-1} = 00 \rightarrow \left\{ \begin{array}{l} 0000 \quad 0010 \quad 0 \\ \sim \text{Shift} \end{array} \right. \\
 3) q_0q_{-1} = 00 \rightarrow \left\{ \begin{array}{l} 0000 \quad 0001 \quad 0 \\ \sim \text{Shift} \end{array} \right. \\
 4) q_0q_{-1} = 10 \Rightarrow \left\{ \begin{array}{l} 11110001 \quad 0 \quad \sim A = A - m = A + \bar{m} + 1 \\ 1111 \quad 1000 \quad 1 \quad \sim \text{Shift} \end{array} \right.
 \end{array}$$

Output=11111000

$$\begin{array}{r}
 0001 \xrightarrow{\text{کپی دو}} 1111 \quad 0001 \quad (1) \\
 \quad \quad \quad \times 1000 \quad (-8) \\
 \hline
 0000 \quad \text{(در ضرب شده)} \\
 0000 \quad \text{(در ضرب شده)} \\
 0000 \quad \text{(در ضرب شده)} \\
 -0001 \quad \text{(در ضرب شده)} \\
 \hline
 00000000 \\
 + 00000000 \\
 \hline
 00000000 \\
 \hline
 11111111 \\
 \hline
 11111000 \Rightarrow -8
 \end{array}$$