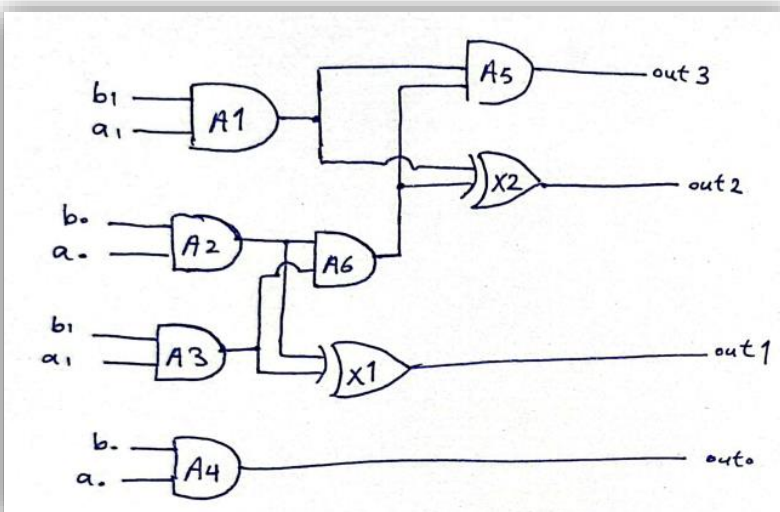


ایده پیاده سازی به این صورت بود که با استفاده از ضرب کننده 2×2 ضرب کننده های 4×4 , 8×8 , 16×16 بسازیم. ابتدا ماژول ضرب کننده ۲ بیتی را مطابق بلوک دیاگرام زیر پیاده سازی میکنیم:



همچنین میدانیم که میتوان یک عدد ۴ بیتی را به صورت زیر نوشت:

$$a_3 a_2 a_1 a_0 = a_3 a_2 \times 2^2 + a_1 a_0$$

← بقیه ۲ واحد سیفیت به چپ

در نتیجه یک ضرب 4×4 را میتوان به این صورت نوشت:

$$a_3 a_2 a_1 a_0 \times b_3 b_2 b_1 b_0 = (a_3 a_2 \times 2^2 + a_1 a_0) (b_3 b_2 \times 2^2 + b_1 b_0)$$

$$= a_3 a_2 \times b_3 b_2 \times 2^4 + a_3 a_2 \times b_1 b_0 \times 2^2 + b_3 b_2 \times a_1 a_0 \times 2^2 + a_1 a_0 \times b_1 b_0$$

← بقیه ۴ واحد سیفیت به چپ ← بقیه ۲ واحد سیفیت به چپ ← بقیه ۲ واحد سیفیت به چپ

پس الگوریتم به صورت زیر میشود:

$$\begin{bmatrix} a_3 & a_2 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & a_1 & a_0 \end{bmatrix}$$

ساخت ضرب کننده 4×4 از ضرب کننده 2×2

هر عدد 4 بیتی را می توان به این صورت نوشت:

۲ واحد سیگنیت به جبه

$$a_3 a_2 a_1 a_0 = a_3 a_2 \ll 2 + a_1 a_0$$

$$b_3 b_2 b_1 b_0 = b_3 b_2 \ll 2 + b_1 b_0$$

$$(a_3 a_2 \ll 2 + a_1 a_0) (b_3 b_2 \ll 2 + b_1 b_0) = (a_3 a_2 \times b_3 b_2) \ll 4 + (a_3 a_2 \times b_1 b_0) \ll 2 + (a_1 a_0 \times b_3 b_2) \ll 2 + (a_1 a_0 \times b_1 b_0)$$

ضرب ۸ در ۸:

$$\begin{aligned} & (a_7 a_6 a_5 a_4 \times 2^4 + a_3 a_2 a_1 a_0) (b_7 b_6 b_5 b_4 \times 2^4 + b_3 b_2 b_1 b_0) \\ &= (a_7 a_6 a_5 a_4 \times b_7 b_6 b_5 b_4) \times 2^8 + (a_7 a_6 a_5 a_4 \times b_3 b_2 b_1 b_0) \times 2^4 + (b_7 b_6 b_5 b_4 \times a_3 a_2 a_1 a_0) \times 2^4 + (a_3 a_2 a_1 a_0 \times b_3 b_2 b_1 b_0) \end{aligned}$$

۸ واحد سیگنیت به جبه
۴ واحد سیگنیت به جبه
۴ واحد سیگنیت به جبه

کد مربوط به این قسمت هم مشابه با قبلی نوشته میشود...

نتایج شبیه سازی:

```
C:\Users\Ofogh Rayaneh 325\Desktop\LUMOS-main>iverilog -o LUMOS.vvp LUMOS_Testbench.v
```

```
C:\Users\Ofogh Rayaneh 325\Desktop\LUMOS-main>vvp LUMOS.vvp
```

```
VCD info: dumpfile LUMOS.vcd opened for output.
```

```
Execution Finished.
```

```
LUMOS_Testbench.v:187: $finish called at 24256 (1ns)
```

