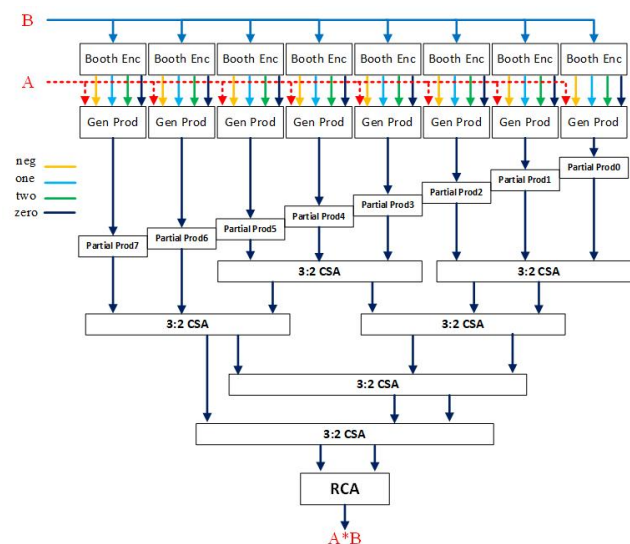


1、学习基于 Booth 编码器的乘法器，并画出模块图。

基于 Booth 编码器的乘法器通过减少部分积数目和加法次数实现了较快的计算速度。假设有一个 8 位乘数，它的二进制值为 01111110，它的 6 个 1 将产生 6 个非零部分积。我们在原值末尾加辅助位 0，变为 011111100，然后利用低位减去高位将这个二进制值改为 100000-10，从而我们只需要计算两个部分积的和。这就是 Booth 编码。



另外，亦可利用 Booth 算法以移位的方式计算乘法：以 4 位二进制数为例， $n=4$ 。设两乘数分别为 a 和 b ，则任取其中一乘数 a 使 $A=a$ ， $B=b0000$ 。 $q=0$ 。重复 n 次以下操作：若 A_q 的末位为 00 或 01，向右移位；若 A_q 的末位数为 10， $A=A-B$ （加上 $-B$ 的补码）后 A 向右移位；若 A_q 的末位数为 01， $A=A+B$ 后移位。如此即得到 A 与 B 的积。

2、课后习题：4.4、4.17

4.4 分析同步电路。写出激励方程、激励/转换表及状态/输出表。设对应 $Q_2Q_1Q_0 = 000 \sim 111$ 的状态名分别为 A~H。

答：

激励函数：

$$\begin{aligned} Q_2^{n+1} &= D_0 = Q_1 \\ Q_1^{n+1} &= D_1 = Q_2 \\ Q_0^{n+1} &= D_2 = \overline{Q_2 + Q_1} \oplus (Q_1 \oplus Q_0) \end{aligned}$$

激励转换表：

| $Q_2 Q_1 Q_0$ | $D_2 D_1 D_0$ | $Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$ |
|---------------|---------------|---------------------------------|
| 000 | 100 | 100 |
| 001 | 000 | 000 |
| 010 | 101 | 101 |
| 011 | 001 | 001 |
| 100 | 010 | 010 |
| 101 | 110 | 110 |
| 110 | 111 | 111 |
| 111 | 011 | 011 |

状态/输出表：

| $Q_2 Q_1 Q_0$ | $Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$ |
|---------------|---------------------------------|
| A | E |
| B | A |
| C | F |
| D | B |
| E | C |
| F | G |
| G | H |
| H | D |

4. 17 分析电路，分别写出激励函数、激励转换表和状态表，说明此电路的逻辑功能。

答：

激励函数：

$$\begin{aligned} D_1 &= \overline{Q_3 \oplus Q_4 + Q_2} \\ D_2 &= Q_1 \overline{Q_3 + Q_4} + \overline{Q_1} Q_2 \\ D_3 &= (Q_1 + \overline{Q_2}) Q_3 + \overline{Q_1} Q_2 \overline{Q_4} \\ D_4 &= \overline{Q_1} \overline{Q_2} Q_3 + (Q_1 + Q_2) Q_4 \end{aligned}$$

次态方程：

$$\begin{aligned} Q_1^{n+1} &= D_1 = \overline{Q_3 \oplus Q_4 + Q_2} = Q_4 Q_3 \overline{Q_2} + \overline{Q_4} \overline{Q_3} \overline{Q_2} \\ Q_2^{n+1} &= D_2 = Q_1 \overline{Q_3 + Q_4} + \overline{Q_1} Q_2 = \overline{Q_4} \overline{Q_3} Q_1 + Q_2 \overline{Q_1} \\ Q_3^{n+1} &= D_3 = (Q_1 + \overline{Q_2}) Q_3 + \overline{Q_1} Q_2 \overline{Q_4} = \overline{Q_4} Q_2 \overline{Q_1} + Q_3 \overline{Q_2} + Q_3 Q_1 \\ Q_4^{n+1} &= D_4 = \overline{Q_1} \overline{Q_2} Q_3 + (Q_1 + Q_2) Q_4 = Q_4 Q_2 + Q_4 Q_1 + Q_3 \overline{Q_2} \overline{Q_1} \end{aligned}$$

激励转换表：

| $Q_4Q_3Q_2Q_1$ | $D_4D_3D_2D_1$ | $Q_4^{n+1}Q_3^{n+1}Q_2^{n+1}Q_1^{n+1}$ |
|----------------|----------------|--|
| 0000 | 0001 | 0001 |
| 0001 | 0011 | 0011 |
| 0010 | 0110 | 0110 |
| 0011 | 0010 | 0010 |
| 0100 | 1100 | 1100 |
| 0101 | 0100 | 0100 |
| 0110 | 0110 | 0110 |
| 0111 | 0100 | 0100 |
| 1000 | 0000 | 0000 |
| 1001 | 1000 | 1000 |
| 1010 | 1010 | 1010 |
| 1011 | 1000 | 1000 |
| 1100 | 1101 | 1101 |
| 1101 | 1101 | 1101 |
| 1110 | 1010 | 1010 |
| 1111 | 1100 | 1100 |

状态输出表：

| $Q_4Q_3Q_2Q_1$ | $Q_4^{n+1}Q_3^{n+1}Q_2^{n+1}Q_1^{n+1}$ |
|----------------|--|
| 0 | 1 |
| 1 | 3 |
| 2 | 6 |
| 3 | 2 |
| 4 | 12 |
| 5 | 4 |
| 6 | 6 |
| 7 | 4 |
| 8 | 0 |
| 9 | 8 |
| 10 | 10 |
| 11 | 8 |
| 12 | 13 |
| 13 | 13 |
| 14 | 10 |
| 15 | 12 |

逻辑功能：将输入的任意四位二进制数 $Q_4Q_3Q_2Q_1$ 转化为 0110、1010、1101 中的一种，具体转化如上述状态输出表所示。