رابعاً- دارة جامع/طارح:

كما الحال مع دارة الجامع، نعلم أنه يمكن رسم وتصميم دارة الطارح عبر كتابة جدول الحقيقة لها واستنتاج معادلاتها، كما نعلم أيضاً أنه يمكن تعديل دارة الجامع لتصبح دارة طارح باعتبار أن عملية الطرح هي عملية جمع مع المتمم الثاني. نستنتج أنه يمكن انجاز دارة واحدة تقوم إما بعملية الجمع أو الطرح، وتدعى عندئذٍ بدارة جامع/طارح، وسنتعرف على تصميمها في هذه التجربة.

لحساب المتمم الثاني لعدد ما: نقوم بقلب خانات العدد لنحصل على المتمم الأول له، ومن ثم نضيف إليه القيمة "1".

بفرض رغبتنا إجراء عملية الطرح التالية:

$$A-B$$

A: المطروح منه.

B: المطروح.

تؤول عملية الطرح السابقة إلى عملية جمع مع المتمم الثاني كالتالي:

$$A-B = A + \overline{B} + 1$$

يمكن تحقيق دارة الجامع/الطارح بأكثر من طريقة منها المبينة بالشكل (1).

يقوم المدخل Mode بتحديد عمل الدارة وفق الآتي:

- عندما Mode = 0، تكون دارة جمع بأربع بتات.

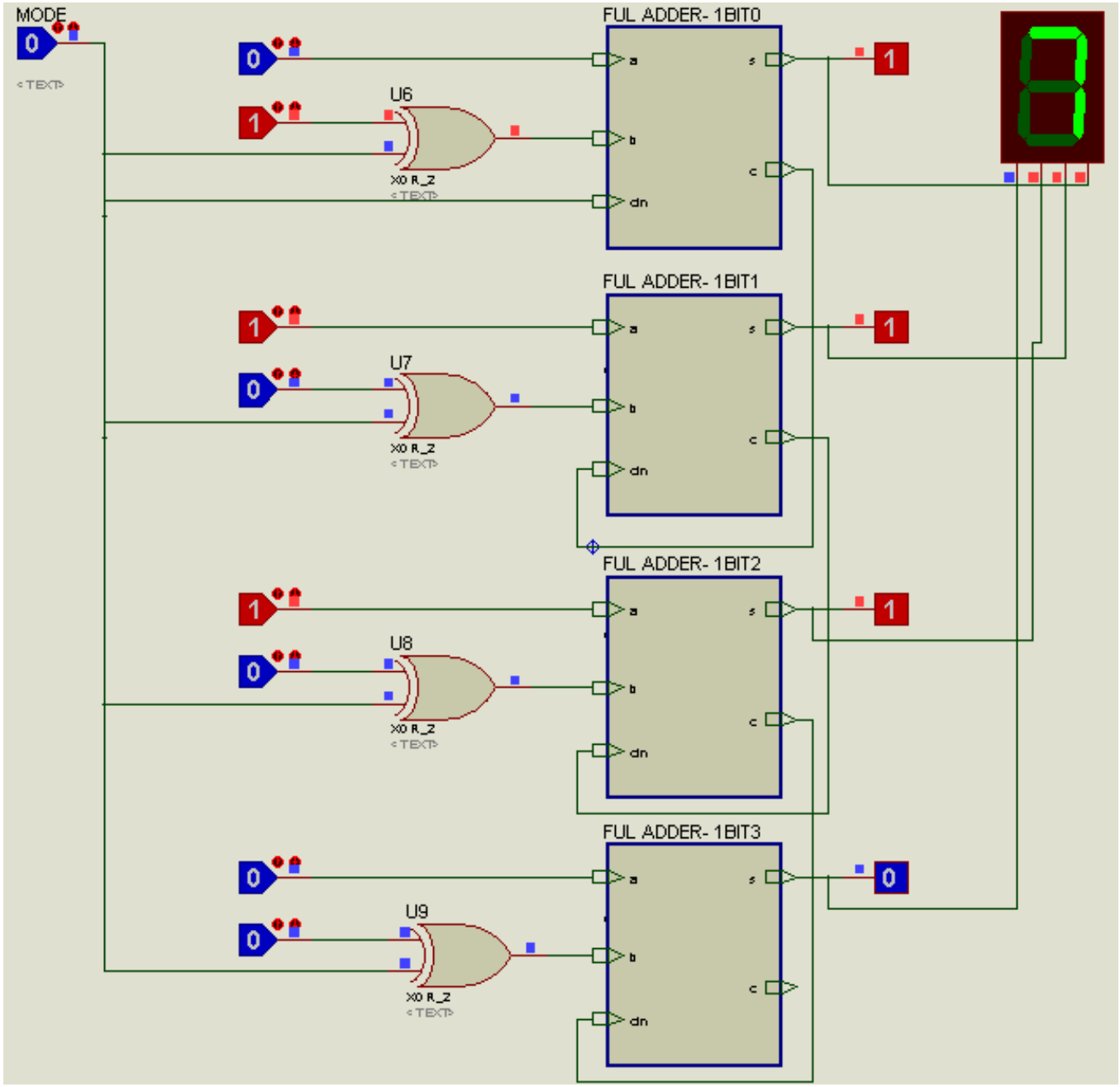
نلاحظ من الشكل (1) أن: $A = (0110)_2 = (6)_{10}$ و $B = (0001)_2 = (1)_{10}$.

وبالتالي يكون خرج الدارة هو: $(0111)_2 = (7)_{10}$.

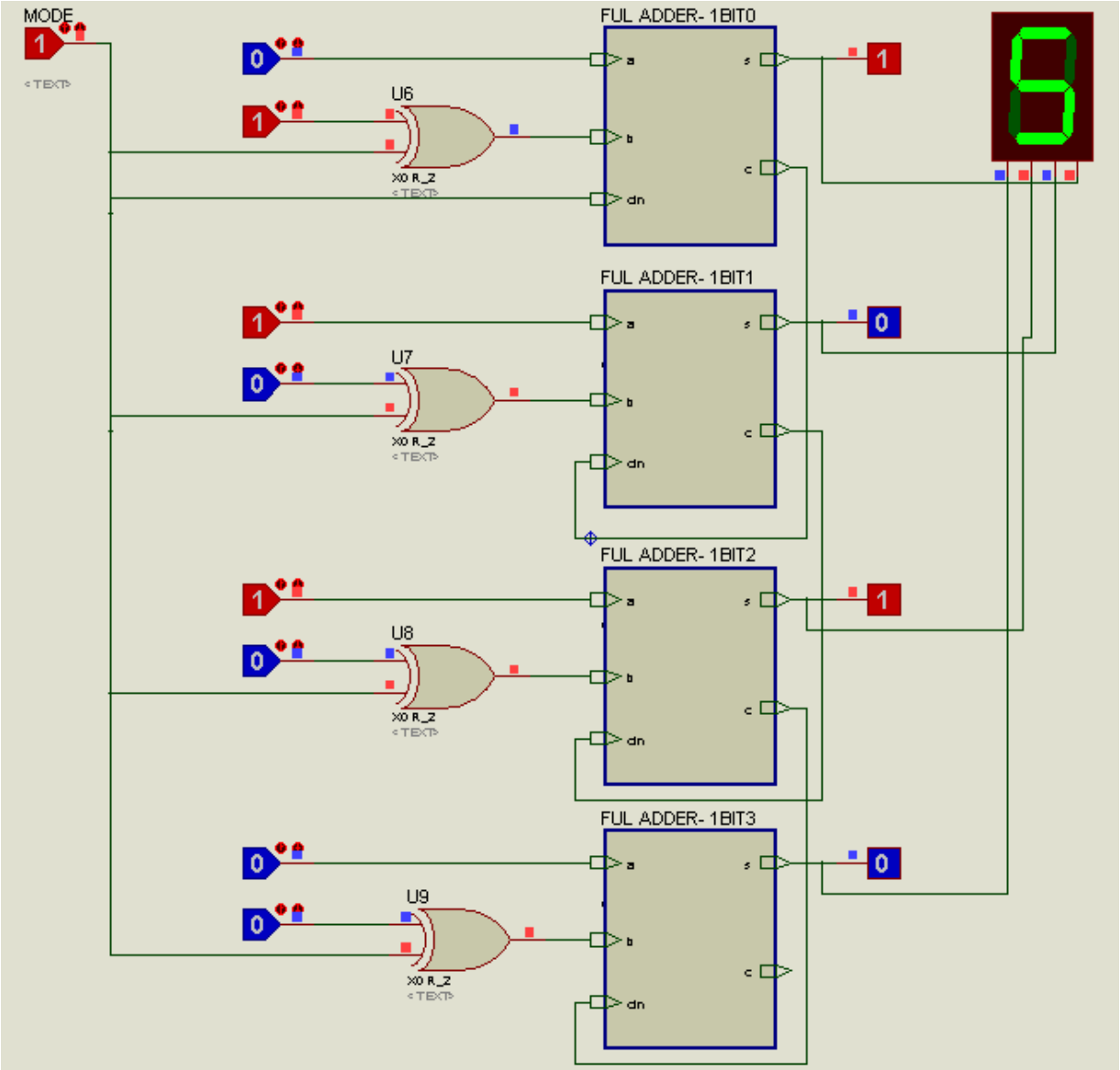
- عندما Mode=1، تكون دارة طرح بأربعة بتات.

نلاحظ من الشكل (2) أن: $A = (0110)_2 = (6)_{10}$، و $B = (0001)_2 = (1)_{10}$.

وبالتالي يكون خرج الدارة هو: $(0101)_2=(5)_{10}$.


MODE
FUL ADDER- 1BIT0
FUL ADDER- 1BIT1
FUL ADDER- 1BIT2
FUL ADDER- 1BIT3
U6
U7
U8
U9
XOR_2
a
b
cin
s
c

الشكل (1)



الشكل (2)

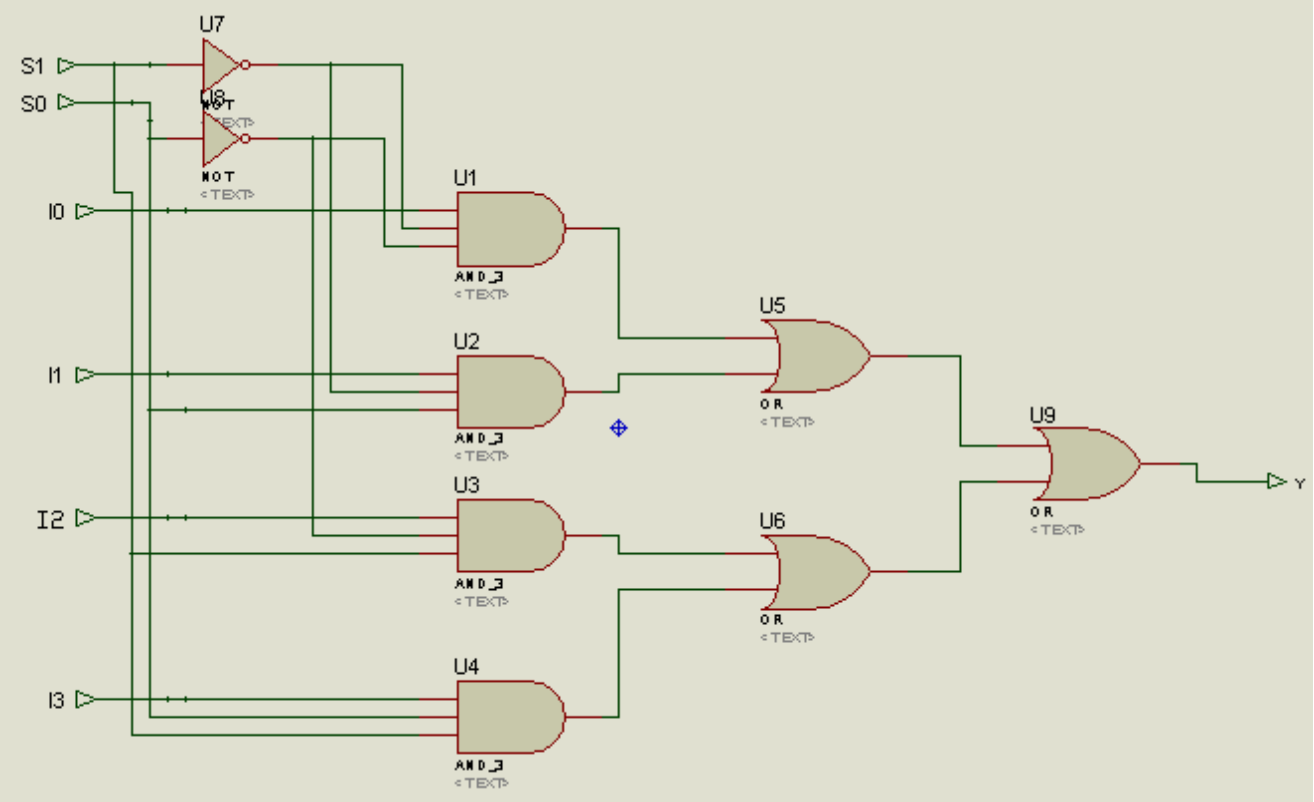
يمكن تعميم الطريقة السابقة المستخدمة في تصميم دارة جامع/طارح واستبدال بوابات XOR بنواخب[1] للتبديل بين عملية الجمع أو الطرح.

نستخدم ناخب (4×1) بجدول الحقيقة والمعادلات التالية:

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

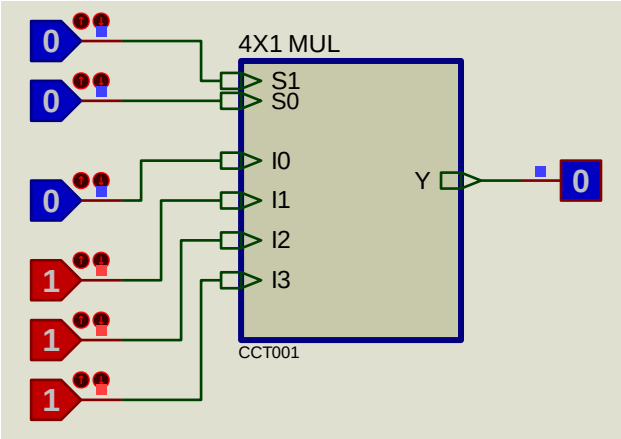
[1] يكفينا ناخب (2 ×1) في هذه المرحلة، إلا أننا سنصمم ناخب (4×1) لأننا سنستخدمه لاحقاً في التبديل بين العمليات التي ستنفذها وحدة الحساب والمنطق.

نقوم بتصميم الدارة في Proteus كما في الشكل (3).



الشكل (3)

ويكون لها المخطط الصندوقي كما في الشكل (4).



الشكل (4)