جامعة حلب في المناطق المحررة
كلية الهندسة المعلوماتية
السنة الرابعة

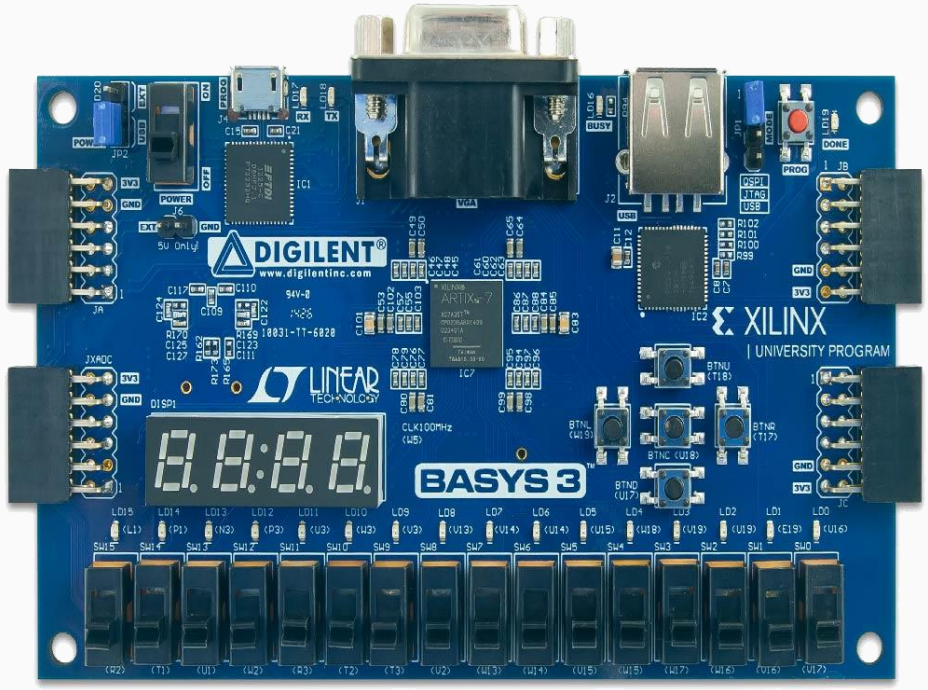
مقرر عملي

بنية وتنظيم الحواسيب2

تصميم معالج 4bit

د.م. عبد القادر غزال
م. محمد نور بدوي

العام الدراسي: 2023 - 2024



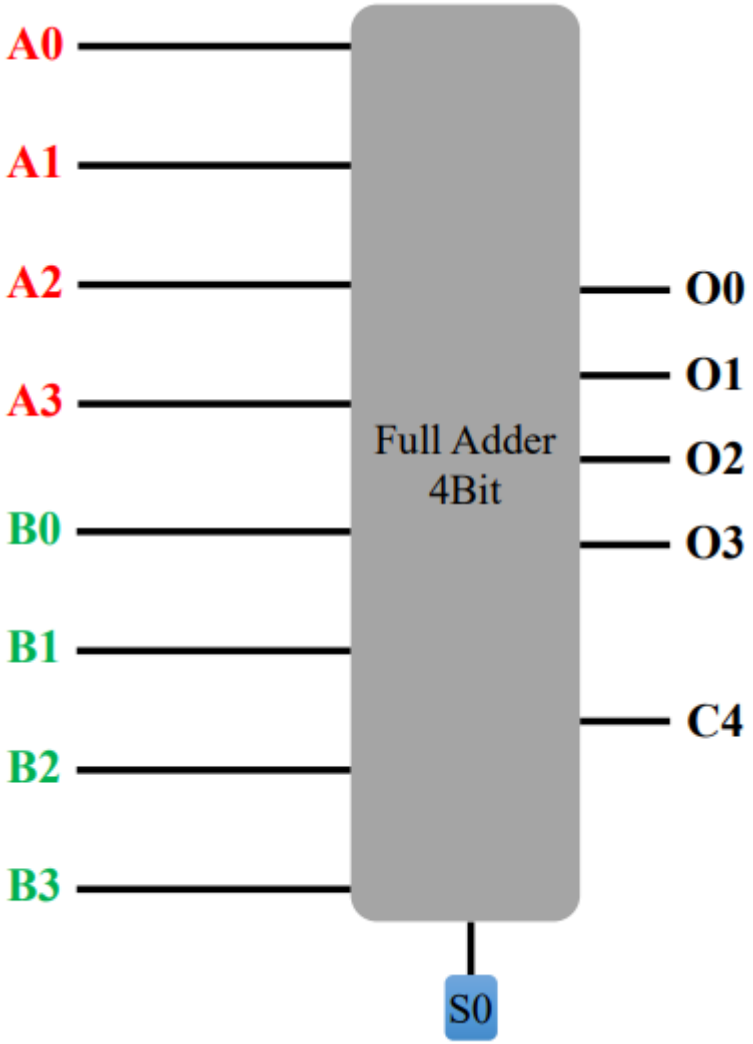
المحاضرة العملية الخامسة

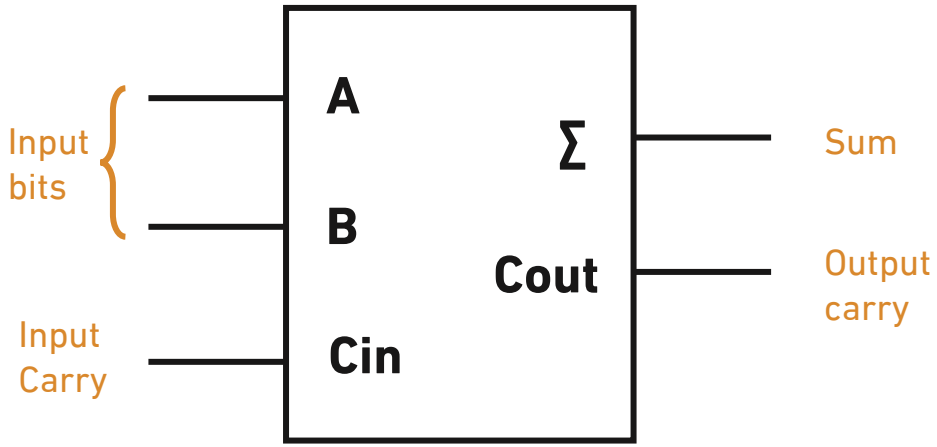
بناء معالج بسيط

- يتكون النظام الحاسوبي بشكل عام من المعالج والذواكر وتجهيزات متممة تشمل اللوحة الأم وتجهيزات الدخل والخرج بالإضافة إلى مصادر التغذية.
- لفهم آلية عمل المعالج سنقوم ببناء معالج بسيط يعالج بعض العمليات الحسابية.
- عملية بناء المعالج ستكون على شكل خطوات تطوير متتالية من خلال إضافة عمليات حسابية جديدة في كل مرحلة.

بناء معالج بسيط

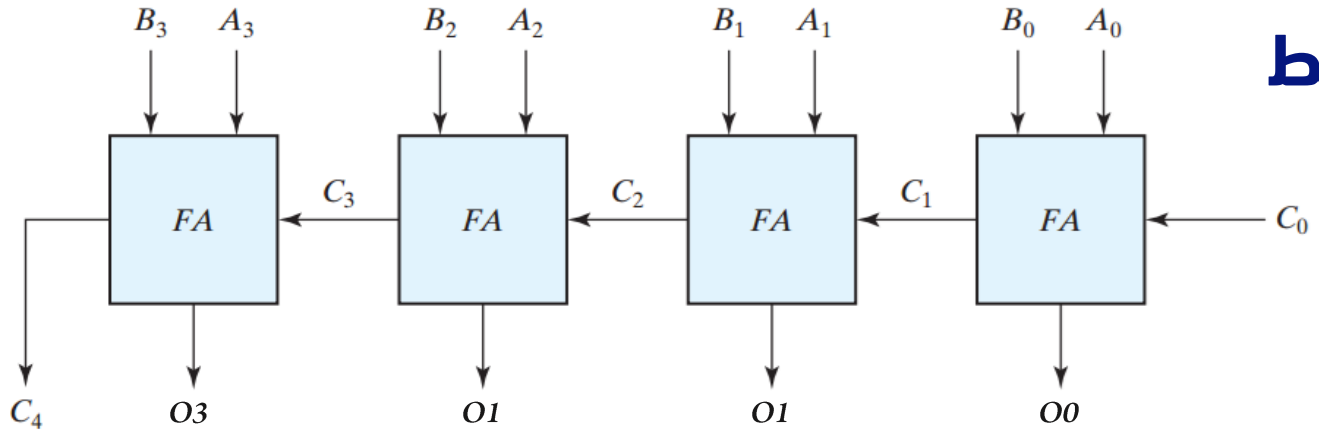
- يوضح الشكل الجانبي بنية معالج يقوم فقط بعملية جمع متحولين بطول 4bit





نموذج مبسط لمتحول واحد 1bit

بناء معالج بسيط



الكود البرمجي

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity processor is
    Port ( a : in STD_LOGIC_VECTOR (3 downto 0);
           b : in STD_LOGIC_VECTOR (3 downto 0);
           o : out STD_LOGIC_VECTOR (3 downto 0);
           c0 : in STD_LOGIC;
           c4 : out STD_LOGIC);
end processor;

architecture Behavioral of processor is

        signal c3, c2, c1: std_logic;
    begin
```

بناء معالج بسيط

```
        -- The first full adder
        o(0) <= a(0) xor b(0) xor c0;
        c1 <= ( (a(0) or b(0)) and c0 ) or (a(0) and b(0));

        -- The second full adder
        o(1) <= a(1) xor b(1) xor c1;
        c2 <= ( (a(1) or b(1)) and c1 ) or (a(1) and b(1));

        -- The third full adder
        o(2) <= a(2) xor b(2) xor c2;
        c3 <= ( (a(2) or b(2)) and c2 ) or (a(2) and b(2));

        -- The fourth full adder
        o(3) <= a(3) xor b(3) xor c3;
        c4 <= ( (a(3) or b(3)) and c3 ) or (a(3) and b(3));
end Behavioral;
```

$$S = A \oplus B \oplus Cin$$

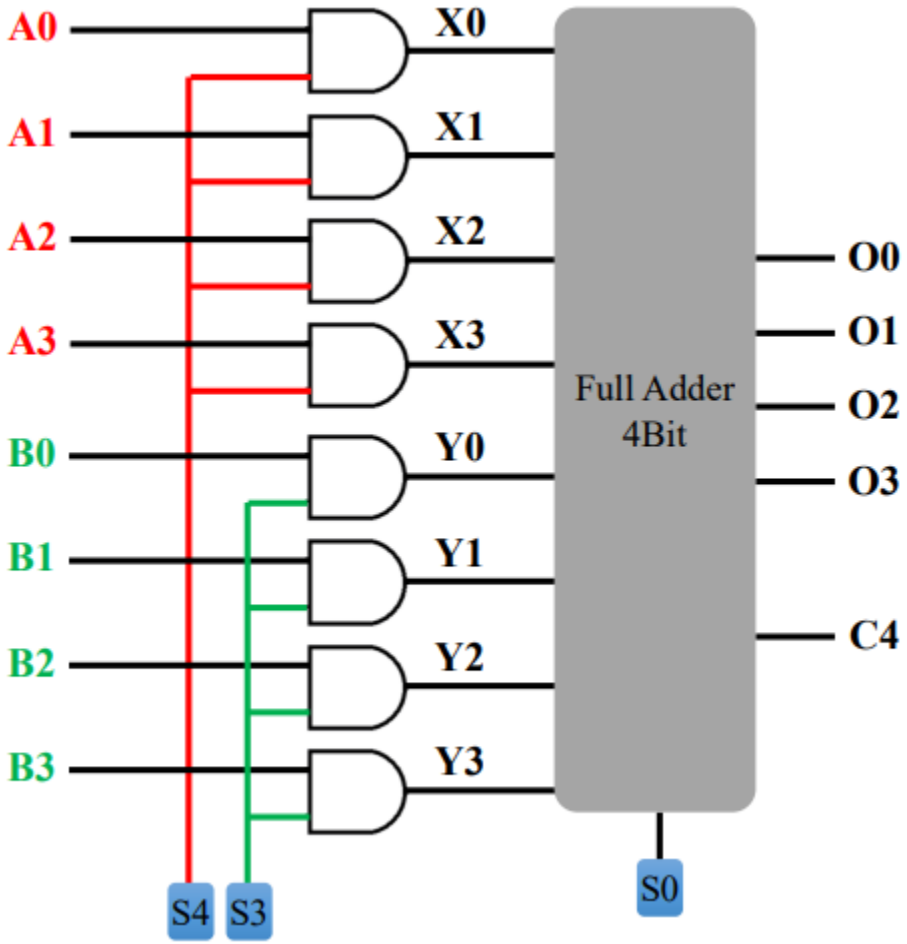
$$Cout = A.B + B.Cin + Cin.A$$

بناء معالج بسيط

ربط الأقطاب مع شريحة FPGA

Name	Direction	...	Package Pin	Fixed	Bank	I/O Std
All ports (14)						
a (4)	IN			☑	14	LVCMOS33*
a[3]	IN		W17	☑	14	LVCMOS33*
a[2]	IN		W16	☑	14	LVCMOS33*
a[1]	IN		V16	☑	14	LVCMOS33*
a[0]	IN		V17	☑	14	LVCMOS33*
b (4)	IN			☑	14	LVCMOS33*
b[3]	IN		W13	☑	14	LVCMOS33*
b[2]	IN		W14	☑	14	LVCMOS33*
b[1]	IN		V15	☑	14	LVCMOS33*
b[0]	IN		W15	☑	14	LVCMOS33*
s (4)	OUT			☑	14	LVCMOS33*
s[3]	OUT		V19	☑	14	LVCMOS33*
s[2]	OUT		U19	☑	14	LVCMOS33*
s[1]	OUT		E19	☑	14	LVCMOS33*
s[0]	OUT		U16	☑	14	LVCMOS33*
Scalar ports (2)						
c0	IN		R2	☑	34	LVCMOS33*
c4	OUT		L1	☑	35	LVCMOS33*

بناء معالج بسيط



- مـع إضـافة بوابـات And كمـا يوضـح الشـكل، أصـبح بالإمكان تمرير أو حجب قيمة متحولات الدخل A & B حسب قيمة قطبي التحكم S3 & S4

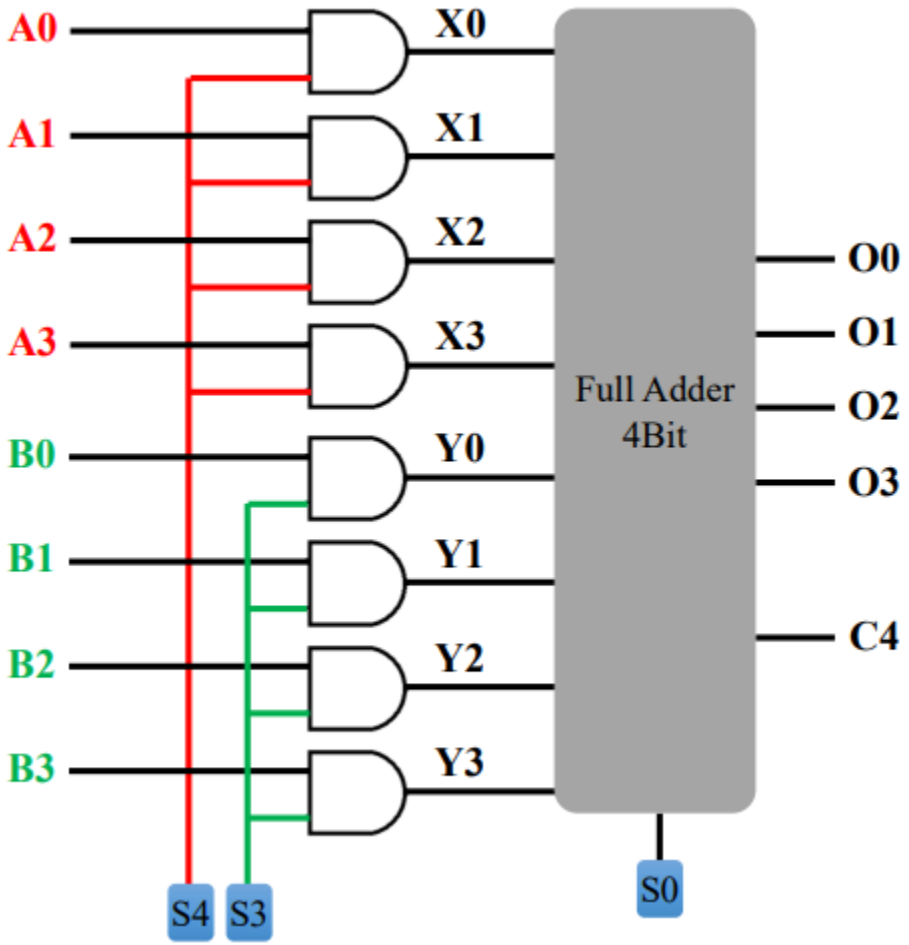
- نضيف على الكود البرمجي في قسم Entity المداخل **S3 & S4** كالتالي:

```
S3, s4 : in STD_LOGIC;
```

بناء معالج بسيط

- في قسم architecture نقوم بإنشاء متغيرات من

أجل أقطاب التحكم.




```
architecture Behavioral of processor is

    signal c3, c2, c1: std_logic;
    signal x: STD_LOGIC_VECTOR (3 downto 0);
    signal y: STD_LOGIC_VECTOR (3 downto 0);

 begin
```

x(0) <= a(0) and s4;	y(0) <= b(0) and s3;
x(1) <= a(1) and s4;	y(1) <= b(1) and s3;
x(2) <= a(2) and s4;	y(2) <= b(2) and s3;
x(3) <= a(3) and s4;	y(3) <= b(3) and s3;

بناء معالج بسيط

- يصبح لدينا الكود التالي:

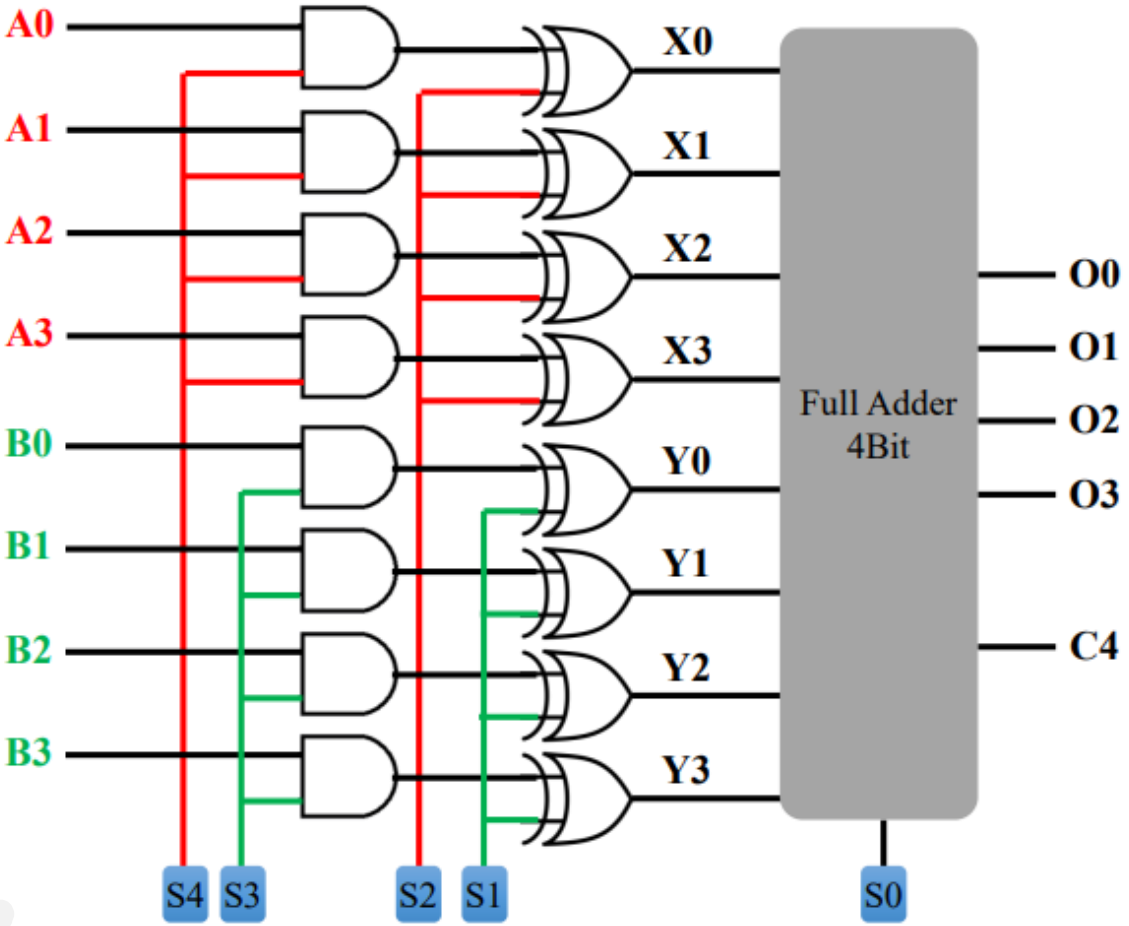
```
        -- The first full adder
        o(0) <= x(0) xor y(0) xor c0;
        c1 <= ( (x(0) or y(0)) and c0 ) or (x(0) and y(0));

        -- The second full adder
        o(1) <= x(1) xor y(1) xor c1;
        c2 <= ( (x(1) or y(1)) and c1 ) or (x(1) and y(1));

        -- The third full adder
        o(2) <= x(2) xor y(2) xor c2;
        c3 <= ( (x(2) or y(2)) and c2 ) or (x(2) and y(2));

        -- The fourth full adder
        o(3) <= x(3) xor y(3) xor c3;
        c4 <= ( (x(3) or y(3)) and c3 ) or (x(3) and y(3));
end Behavioral;
```

بناء معالج بسيط



- لإضافة عملية طرح على هذا المعالج

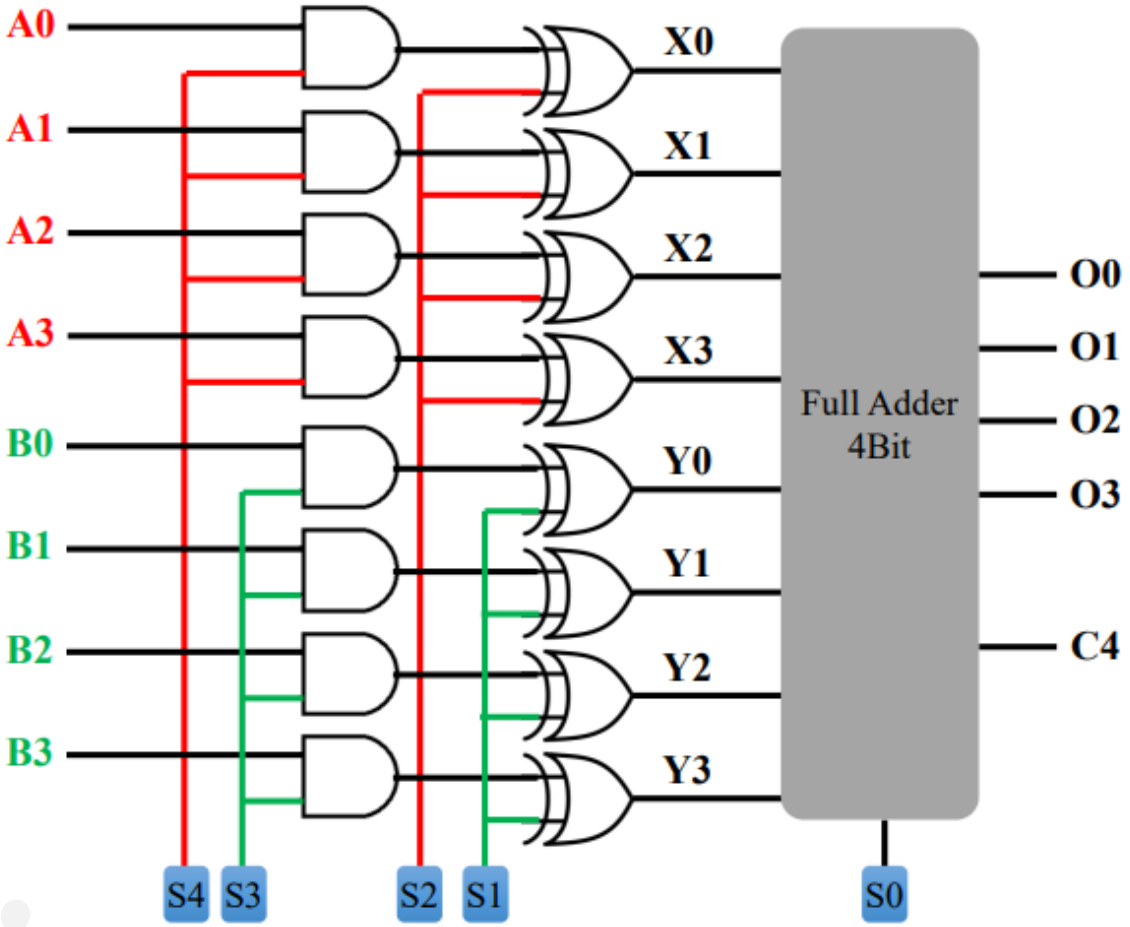
سنعتمد على الفكرة التالية:

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

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

- S1 = 1 يدخل المتحول B إلى الجامع معكوسًا
- S1 = 0 يدخل المتحول B دون تغيير

بناء معالج بسيط



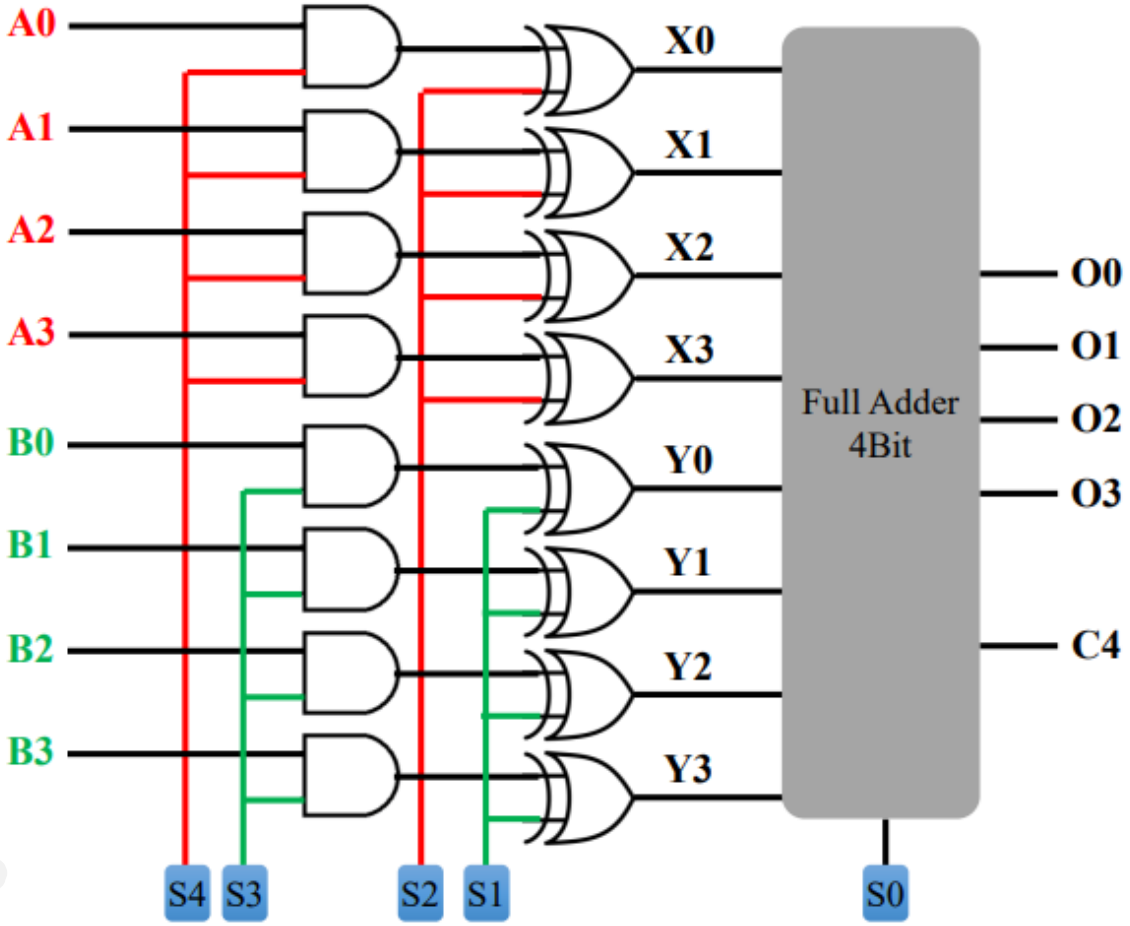
- مـع إضـافة بوابـات Xor كمـا يوضـح الشـكل، أصبح بالإمكان عكس متحولات الدخل A & B حسب قيمة قطبي التحكم S1 & S2

- نضيف على الكود البرمجي في قسم Entity المداخل **S1 & S2** كالتالي:

```
S1, s2 : in STD_LOGIC;
```

بناء معالج بسيط

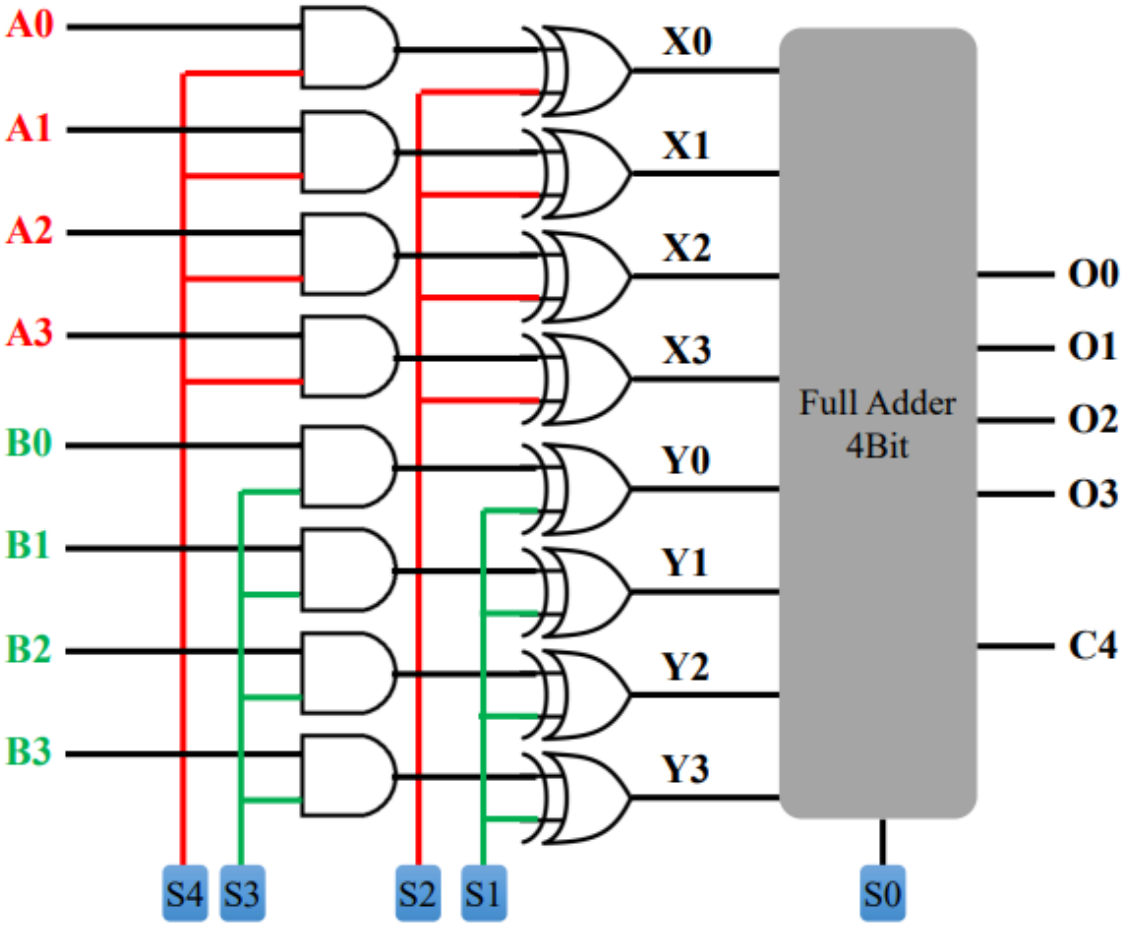
- في قسم architecture نقوم بالتعديلات اللازمة:




```
x(0) <= (a(0) and s4) xor s2;
x(1) <= (a(1) and s4) xor s2;
x(2) <= (a(2) and s4) xor s2;
x(3) <= (a(3) and s4) xor s2;

y(0) <= (b(0) and s3) xor s1;
y(1) <= (b(1) and s3) xor s1;
y(2) <= (b(2) and s3) xor s1;
y(3) <= (b(3) and s3) xor s1;
```

بناء معالج بسيط



A = 1101 & B= 0101

- ناقش على شريحة FPGA حالة:

s0=0; s1=0; s2=0; s3=1; s4=1

- ناقش على شريحة FPGA حالة:

s0=1; s1=1; s2=0; s3=1; s4=1

انتهت المحاضرة