**BINARY ADDERS**

Half Adder

The digital circuit used to sum two bits is called half adder.

1

1



+\_\_\_\_

0



|  |  |  |  |
| --- | --- | --- | --- |
| x | y | sum | carry |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |



s = x’y + xy’ = x ⊕ y c = xy

XOR

Icon

Description automatically generatedIcon

Description automatically generated

THIS IS A HALF ADDER CIRCUIT

Full Adder

A combination circuit that adds 3 bits (2 significant bits and a previous carry)

+1 c0 (carry bit), let’s name it z



|  |  |  |
| --- | --- | --- |
| x | 1 | 1 |
| y | 1 | 1 |

+\_\_\_\_\_\_\_\_\_\_\_\_

0

You should also consider carry bit which is coming from right.

Full adder is expected to sum significant bits and carry bit coming from previous summation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | z | s | c |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 |  |  | 1 |  |
| 1 |  | 1 | 1 | 1 |

carry bit:

xz

xy

yz

c (output for carry) = xz + xy + yz

A picture containing icon

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| yz  x | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 |  | 1 |  | 1 |
| 1 | 1 |  | 1 |  |

sum bit:

There is no option we can bring them together

s = x’y’z + x’yz’ + xy’z’ + xyz

Diagram

Description automatically generated with medium confidence

Can we do simplification with boolean algebra?

s = x’y’z + x’yz’ + xy’z’ + xyz = z(x’y’ + xy) + z’(x’y + xy’)

🡪 (x’y + xy’)’ = (x’y)’(xy’)’ = (x+y’)(x’+y) = xx’ + xy + x’y’ + yy’ = xy + x’y’



🡪 Let’s say (x’y + xy’) = k = x ⊕ y:

s = zk’ + z’k = z ⊕ k = z ⊕ x ⊕ y

A picture containing text

Description automatically generatedA picture containing linedrawing

Description automatically generated

THIS IS A FULL ADDER CIRCUIT

Let’s say:

A = 1 0 1 1 -----> a3 a2 a1 a0

B = 0 1 1 1 -----> b3 b2 b1 b0



c0 s3 s2 s1 s0

Yellow 🡪 Half adder

Green 🡪 Full adder

Diagram, engineering drawing

Description automatically generated



**4 bit adder**

If we convert half adder to full adder, that makes things easy for us. For example:

A = a7 a6 a5 a4 a3 a2 a1 a0

B = b7 b6 b5 b4 b3 b2 b1 b0

A picture containing diagram

Description automatically generatedDiagram

Description automatically generated

You can concatenate 2 8-bit-adders and create an 16-bit-adder etc.

Substraction

Usually we use 2nd complement for substraction.

15 - 6 = 15 + (-6) = 9

How to express -6 as 2nd complement?

1st complement: substract it from 9 (biggest digit)

* 9 - 6 = 3

2nd complement: add 1 to the 1st complement

* 3 + 1 = 4

----> 15 - 6 = 15 + (-6) = 15 + 4 = 19

4 (negative element’s 2nd complement) is 1-digit number, but 19 is 2-digit number.

So we have an overflow. If we have an overflow, value is positive, and ignore the overflow.

15 - 33 = 15 + (-33)

1st complement:

* 9 - 3 = 6 , 9 - 3 = 6 --> 66 ( do it for every digit )

2nd complement:

* 66 + 1 = 67 (67 means -33)

15 + 67 = 82

We don’t have an overflow (67 and 82 are both 2-digit number), number (82) is negative.

82’yi gerisi geri çevirmemiz lazım, çünkü negatifliği temsil ediyor, bu yüzden 82’nin 2nd complementini alıyoruz.

1st complement:

* 9 - 8 = 1, 9 - 2 = 7 --> 17

2nd complement:

* 17 + 1 = 18

Output is going to be “-18”.

33 - 15 = 33 + (-15)

1st complement: 84

2nd complement: 85

33 + 85 = 118

There is an overflow, number is positive.

Ignore overflow, so result is 18.

**Binary**

Biggest digit in binary is 1, so we are going to subtract from 1.

0110

1st complement:

* 1 - 0 = 1 , 1 - 1 = 0 , 1 - 1 = 0 , 1 - 0 = 1 ----> 1001

2nd complement:

* 1001 + 1 = 1010

3 - 1 = 011 - 001

011 + (- 001)

1st complement : 110

2nd complement : 111

011 + 111 = 1010

There is an overflow, so number is positive. Our result is 010.

4 - 5 = 100 - 101 = 100 + (-101)



We can use an extra number to show digits of the values. 0 for positive, 1 for negative.

0100 + 1…

1st complement : 010

2nd complement : 011

0100 + 1011 🡪 1011 doesn’t mean -3 just bc 1 is negative 011 is 3 so it is -3 (don’t think as this)  
 If you want to learn this number, you have to take 2nd complement of it.

= 1111

There is no overflow, number is negative.

1st complement : 0000

2nd complement: 0001

Output is -0001 = -1 = 1111

A picture containing text, weathervane

Description automatically generated

Means A, B, and S has 4 cables.

SUBSTRACTION

Diagram

Description automatically generated

We take 2s complement of B (B’ + 1) and add with A.