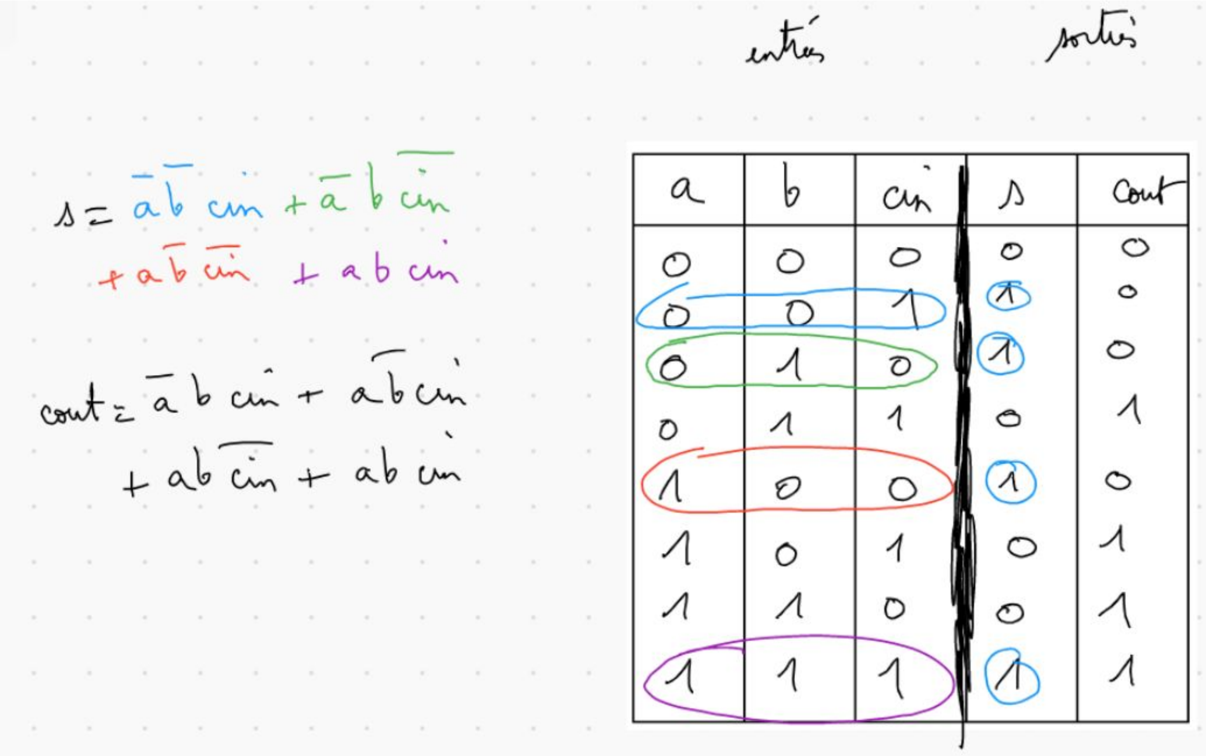
TD Architecture des ordinateurs

Groupe AB

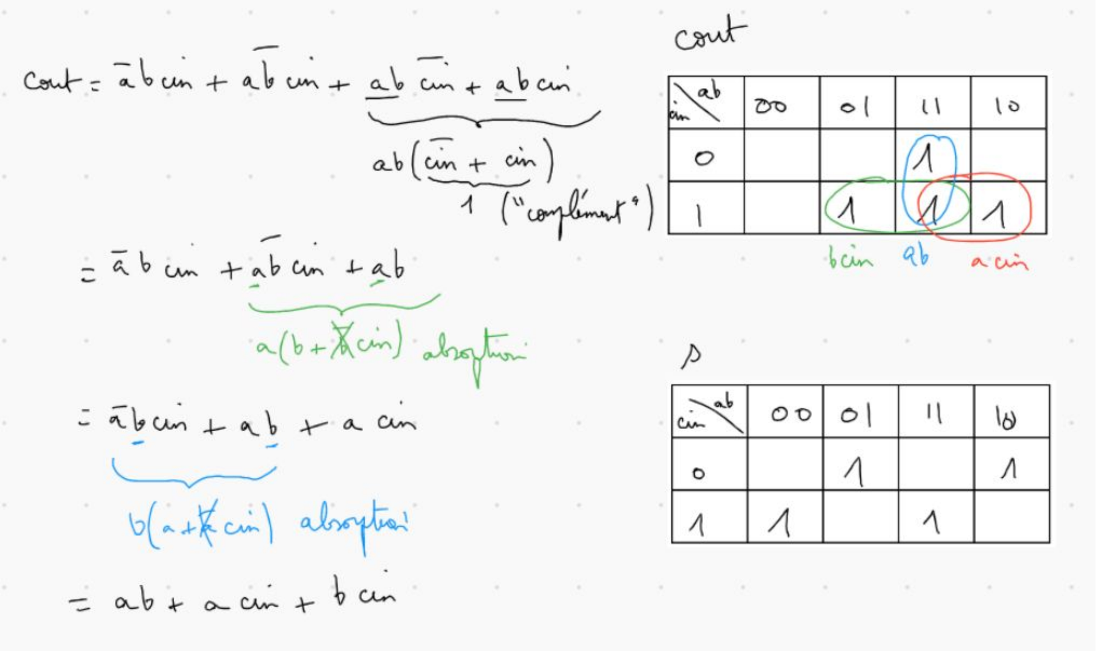
Exercice 1 :



$s = \bar{a}\bar{b}\,cin + \bar{a}b\,\overline{cin} + a\bar{b}\,\overline{cin} + ab\,cin$

$cout = \bar{a}b\,cin + a\bar{b}\,cin + ab\,\overline{cin} + ab\,cin$

entrées			sorties	
a	b	cin	s	Cout
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



$$Cout = \bar{a}b\,cin + a\bar{b}\,cin + \underbrace{\underline{ab}\,\overline{cin} + \underline{ab}\,cin}_{ab(\underbrace{\overline{cin} + cin}_{1\ (\text{"complément"})})}$$

$$= \bar{a}b\,cin + \underbrace{a\bar{b}\,cin + ab}_{a(b + \cancel{\bar{b}}\,cin)\ \text{absorption}}$$

$$= \underbrace{\bar{a}b\,cin + ab}_{b(a + \cancel{\bar{a}}\,cin)\ \text{absorption}} + a\,cin$$

$$= ab + a\,cin + b\,cin$$

cout

cin \ ab	00	01	11	10
0			1	
1		1	1	1

b cin ab a cin

s

cin \ ab	00	01	11	10
0		1		1
1	1		1	

A	B	Cin		Cout	S
0	0	0		0	0
0	0	1		0	1
0	1	0		0	1
0	1	1		1	0
1	0	0		0	1
1	0	1		1	0
1	1	0		1	0
1	1	1		1	1

Pour Cout

Cin\AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Pour S

Cin\ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

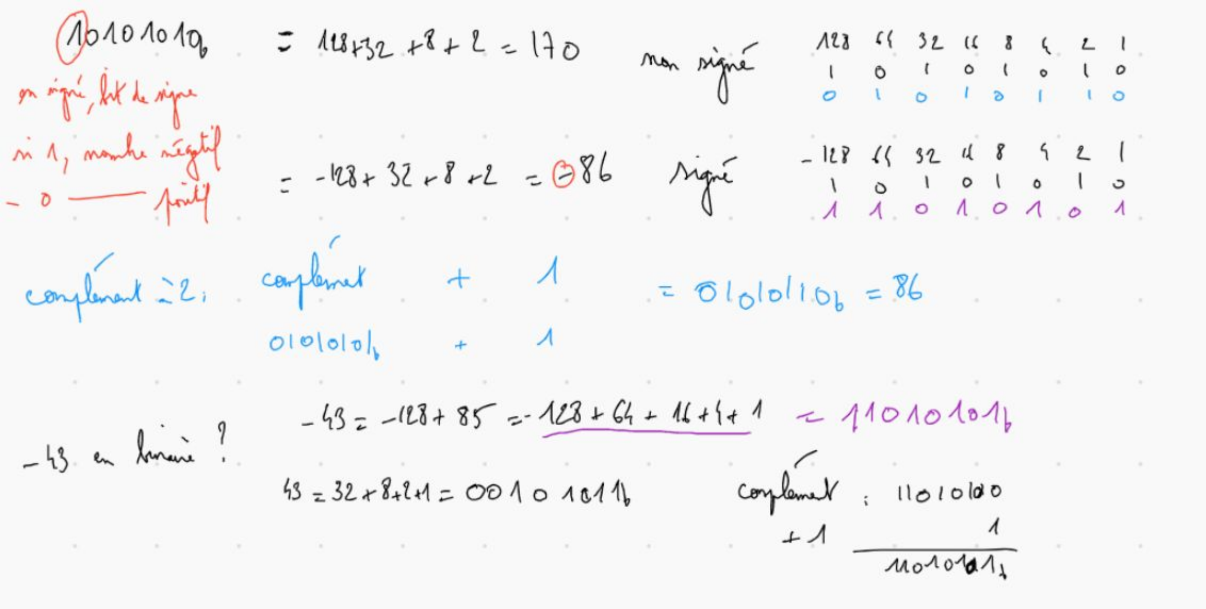
```
module addbin(a, b, cin : s, cout)
        s = /a*/b*cin + /a*b*/cin + a*/b*/cin + a*b*cin
        cout = a*(b+cin) + b*cin
end module
```

Exercice 2

```
module adder8(a[7..0], b[7..0], cin : s[7..0], cout)
        addbin(a[0], b[0],  cin : s[0], aux1)
        addbin(a[1], b[1], aux1 : s[1], aux2)
        addbin(a[2], b[2], aux2 : s[2], aux3)
        addbin(a[3], b[3], aux3 : s[3], aux4)
        addbin(a[4], b[4], aux4 : s[4], aux5)
        addbin(a[5], b[5], aux5 : s[5], aux6)
        addbin(a[6], b[6], aux6 : s[6], aux7)
        addbin(a[7], b[7], aux7 : s[7], cout)
end module
```

Rappel nombres binaires

10101010$_b$ **= 170** en codage non signé   (128 + 32 + 8 + 2)
10101010$_b$ **= -86** en codage signé (-128 + 32 +8 +2)
-43 = **11010101$_b$** (-128 + 64 + 16 + 4 + 1)



10101010$_b$ = 128+32 +8 + 2 = 170 non signé

128	64	32	16	8	4	2	1
1	0	1	0	1	0	1	0
0	1	0	1	0	1	1	0

en signé, bit de signe
si 1, nombre négatif
− 0 —— positif

= -128+ 32 +8 +2 = −86 signé

-128	64	32	16	8	4	2	1
1	0	1	0	1	0	1	0
1	1	0	1	0	1	0	1

complément à 2 : complément + 1 = 01010110$_b$ = 86
01010101$_b$ + 1

−43 en binaire ?

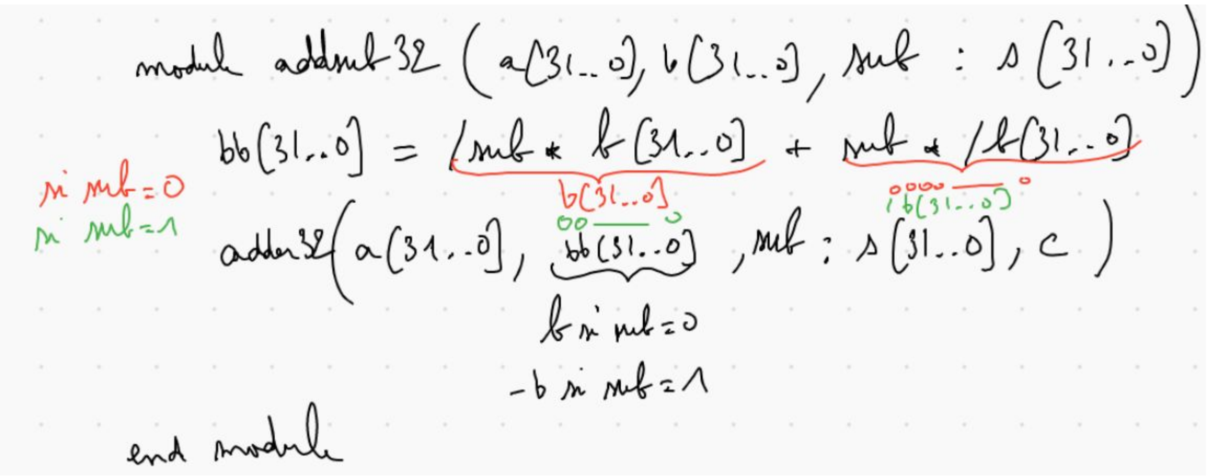
−43 = −128 + 85 = −128 + 64 + 16 + 4 + 1 = 11010101$_b$

43 = 32+8+2+1 = 00101011$_b$

complément : 11010100
+1 1
11010101$_b$

Exercice 3

```
module addsub32(a[31..0], b[31..0], sub : s[31..0])
        bb[31..0] = /sub*b[31..0] + sub*/b[31..0]
        adder32(a[31..0], bb[31..0],sub : s[31..0],c)
end module
```



module addsub32 (a[31..0], b[31..0], sub : s[31..0])

si sub=0

si sub=1

bb[31..0] = $\underbrace{\text{/sub * b[31..0]}}_{\text{b[31..0]}}$ + $\underbrace{\text{sub * /b[31..0]}}_{\text{/b[31..0]}}$

adder32(a[31..0], $\underbrace{\text{bb[31..0]}}$, sub : s[31..0], c)

b si sub=0

-b si sub=1

end module

Exercice 4

```
module addsub32(a[31..0], b[31..0], addsub : s[31..0], V, C)

        bb[31..0] = /addsub*b[31..0] + addsub*/b[31..0]
        adder32(a[31..0], bb[31..0],addsub : s[31..0],co)
        V = /addsub*a[31]*b[31]*/s[31] + /addsub*/a[31]*/b[31]*s[31]
           + addsub*a[31]*/b[31]*/s[31] + addsub*/a[31]*b[31]*s[31]
        C = /addsub*co+addsub*/co

end module
```