Informe previo Práctica-3

Apellidos y nombre: .... Grupo: .....

Apellidos y nombre: Juca Guirao, Ignasi .... Grupo: 33

(por orden alfabético)

$2^{16}-1$

Pregunta 1

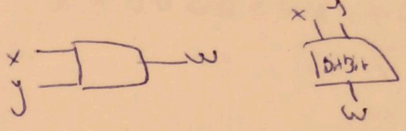
a)

X	0	1
0	0	0
1	0	1

b)

2 entrades
1 sortida

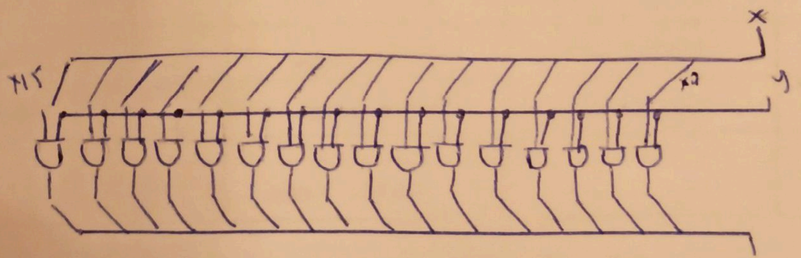
x y	w
0 0	0
0 1	0
1 0	0
1 1	1 → x·y



c) No existeix carry en la multiplicació en binari, ja que la operació es 1·1, la qual es la suma de l'1 una vegada que dona 1. A diferencia dels altres casos on la suma és 0 vegades 1 o 0 que dona 0. Al ser binari el resultat no pot ser en cap cas major que 1.

d) Son necesaris 16 bits com a molt ja que la sera multiplicació potser per 1 que es quedaria igual o per 0 que serien tots 0.

e)



f) T_p major entrada-sortida: x-y, tots els bits tarden el mateix, no hi ha cap carry al que s'ha d'esperar. El t_p serà el del bloc BitBit, la porta AND.

T_p(BitBit).

Pregunta 2

a) $x = 1101$ $x_u = 13$

$y = 1011$ $y_u = 11$

$w = 10001111$ $w_u = 143$

$13 \cdot 11 = 143$ correcte.

b)

$x_u = 23$ $x = 10111$

$y_u = 17$ $y = 10001$

$W_u = 391$ $w = 110000111$

```
          10111
        x 10001
        -------
          10111
         00000
        00000
       00000
      10111
  -------------
      110000111   = 391 → correcte
```

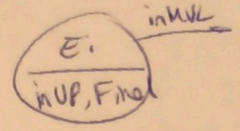
Pregunta 3 22 = 0001 0110 77 = 0100 1101

Estado inicial		W(0) =	D(0) =	B(0) =
Iteración / ciclo j	**M = MULBit (D(j), B(j)<0>)**	**W(j+1) = ADD(W(j), M)**	**D(j+1) = SL-1(D(j))**	**B(j+1) = SRL-1(B(j))**
0	0001 0110	0001 0110	0010 1100	0010 0110
1	0000 0000	0001 0110	0101 1000	0001 0011
2	0101 1000	0110 1110	1011 0000	0000 1001
3	1011 0000	~~0~~ 0001 1110	0110 0000	0000 0100
4	0000 0000	0001 1110	1100 0000	0000 0010
5	0000 0000	0001 1110	1000 0000	0000 0001
6	1000 0000	1001 1110	0000 0000	0000 0000
7	0000 0000	1001 1110	0000 0000	0000 0000
Resul. Final W		1001 1110		

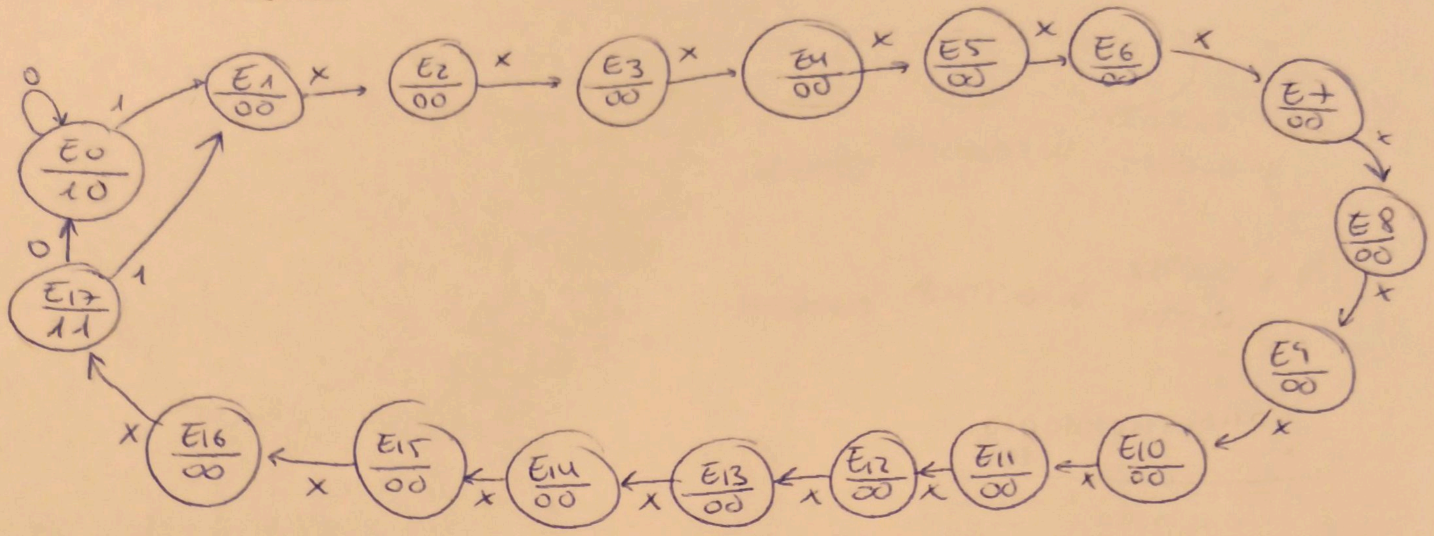
¿Cuál es el resultado correcto de la multiplicación, $W_u = X_u \times Y_u$? 1694

¿Los 8 bits que se obtienen como resultado del algoritmo anterior, representan el resultado correcto de la multiplicación?

¿Por qué? No, perque el cicle 3 no agafem el dígit de pes 8. Per tant el resultat que hauria de ser 1694, 0110 1001 1110 necessita 11 bits i la nostra sortida està limitada a 8 bits.



Pregunta 4



Pregunta 5

ROM_Q+MUL
0x00, 0x01, 0x02, 0x02, 0x03, 0x03, 0x04, 0x04, 0x05, 0x05, 0x06, 0x06, 0x07, 0x07, 0x08, 0x08, 0x09, 0x09, 0xA, 0xA, 0xB, 0xB, 0xC, 0xC, 0xD, 0xD, 0xE, 0xE, 0xF, 0xF, 0x10, 0x10, 0x11, 0x11, 0x00, 0x01

ROM_OutMUL
0x2, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x3

Pregunta 6

a) Camí critic REG → BLOC x·y → ADD → MUX → REG
(AND)

b) T_p = 780 ut REG = 100 + BLOC x·y que es una AND = 20, bloc ADD = 610,

MUX = 50 ut (NOT → AND → OR), i acaba en un REG.

Informe Final

Nom: Juet Guirao Ignasi Grup: 33

① a) $x = 0x0003$
$y = 0x0005$ $w = 0x000F$ correcto

b) $x = 0x6752$
$y = 0x0004$ $w = 0x9D48$ incorrecto. $w_{correcto} = 0x1\ 9D48$

```
   0110011101010010
 ×              100
 ------------------
   0000000000000000
  0000000000000000
 0110011101010010
 ------------------
 000110011101010010 00
```

$\underbrace{000110011101010010\ 00}_{18\,bits} = \boxed{0x19D48}$

② ccl.

③

a) C+17 C INIMUL=1 + ... C+m FINMUL=1 té 18 cicles per tant C+17.

$T_p = 780 \times 18 = 14040$ u.t.
cicles.

Cada 17 cicles pot iniciarse una nova multiplicació ja que en el cicle 17 pot entrar inMUL i anar al cicle 1 per començar nova multiplicació.

b) k=4 Perque ja has fet totes les sumes que havies de fer. El multiplicand es 5, 101, per tant el resultat ja serà el bò en el cicle 4 perque $2^2 = 4$.

3×5

```
   0000 0011
 × 0000 0101
 -----------
   0000 0011
 + 00000 000
  000000 11
 -----------
  000000 1111
```

$\underbrace{1111}_{F}$

No fa falta que faci totes les multiplicacions de 0s despres de l'últim 1 de més pes. Un cop multiplicat per l'últim 1 de més pes el resultat ja és el correcte però s'espera ha a donar el resultat fins el últim cicle perque va fent l'operació per 0, en aquest cas que tot son 0s no canvia el resultat. Si hagues un 1 de més pes si canviaria el resultat. obvi pk no seria 3×5 sino 3×13 per exemple.