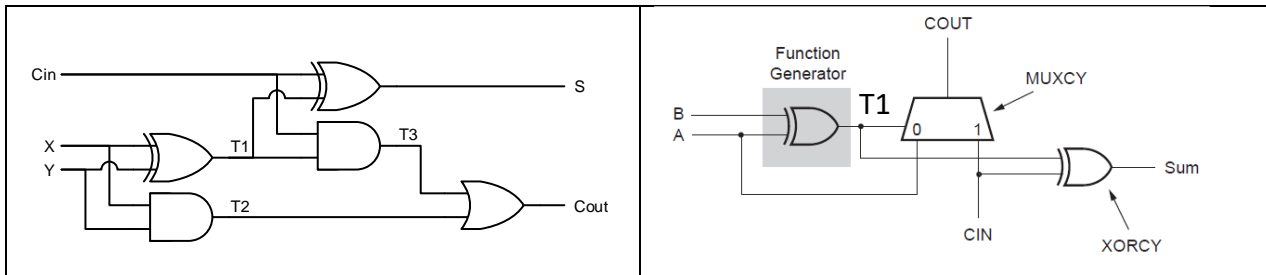


Exercices #6 – Conception et implémentation de fonctions arithmétiques sur FPGA

0601

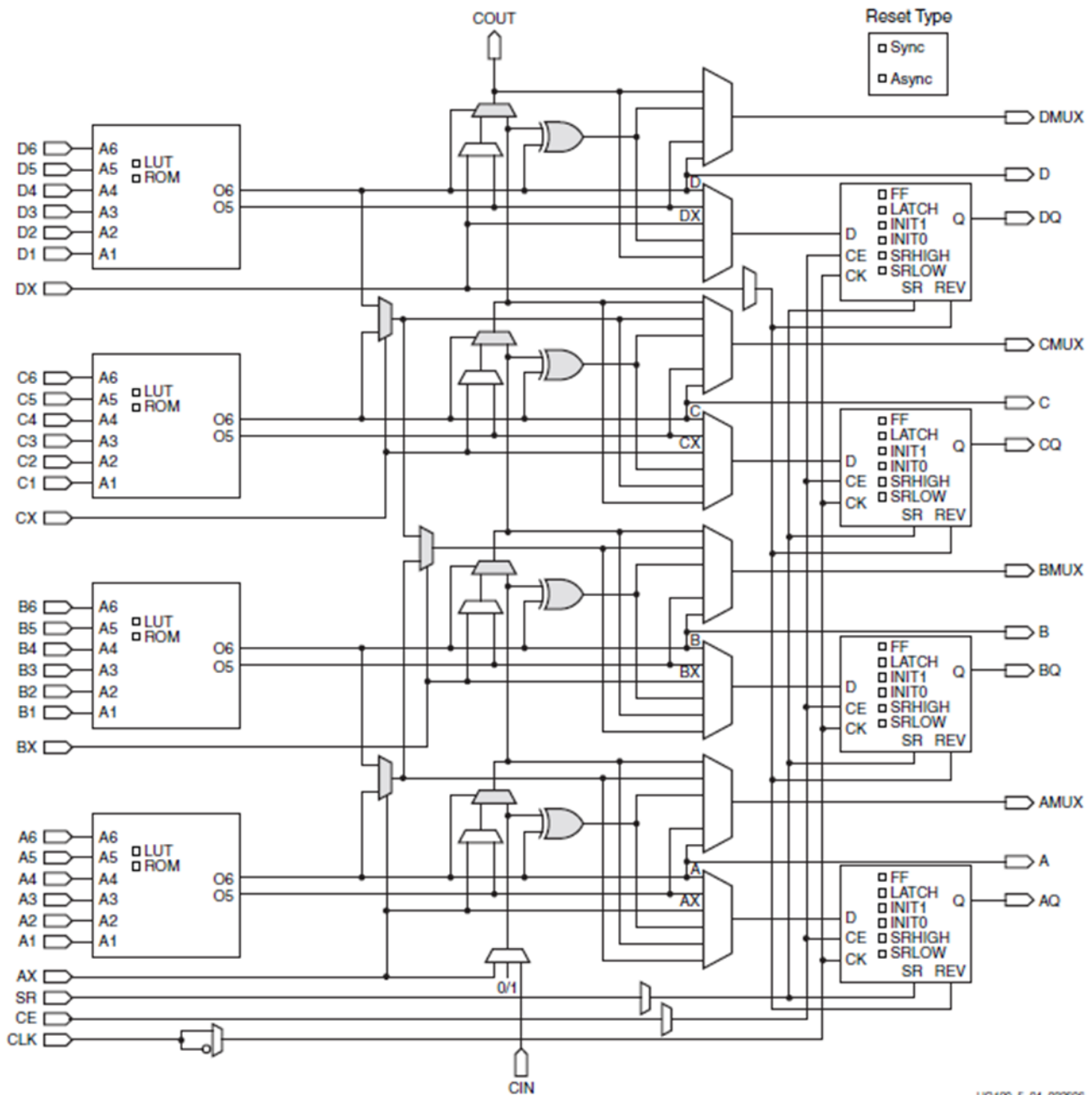
Pour l'additionneur à trois bits, démontrer que le circuit original avec des portes OUX et ET a la même fonction que celui avec un multiplexeur utilisé pour implémenter l'addition sur une tranche de type L. Effectivement, démontrez que $Cout = T1 \text{ Cin} + T1' Y = T1 \text{ Cin} + T1' X = T1 \text{ Cin} + XY$.

**0602**

Montrez, sur le diagramme suivant, comment implémenter un additionneur/soustracteur de 4 bits sur une tranche de type L du Virtex 5.

Les signaux d'entrée sont A(3:0), B(3:0) et Sub/Add'. Les signaux de sortie sont S(4:0).

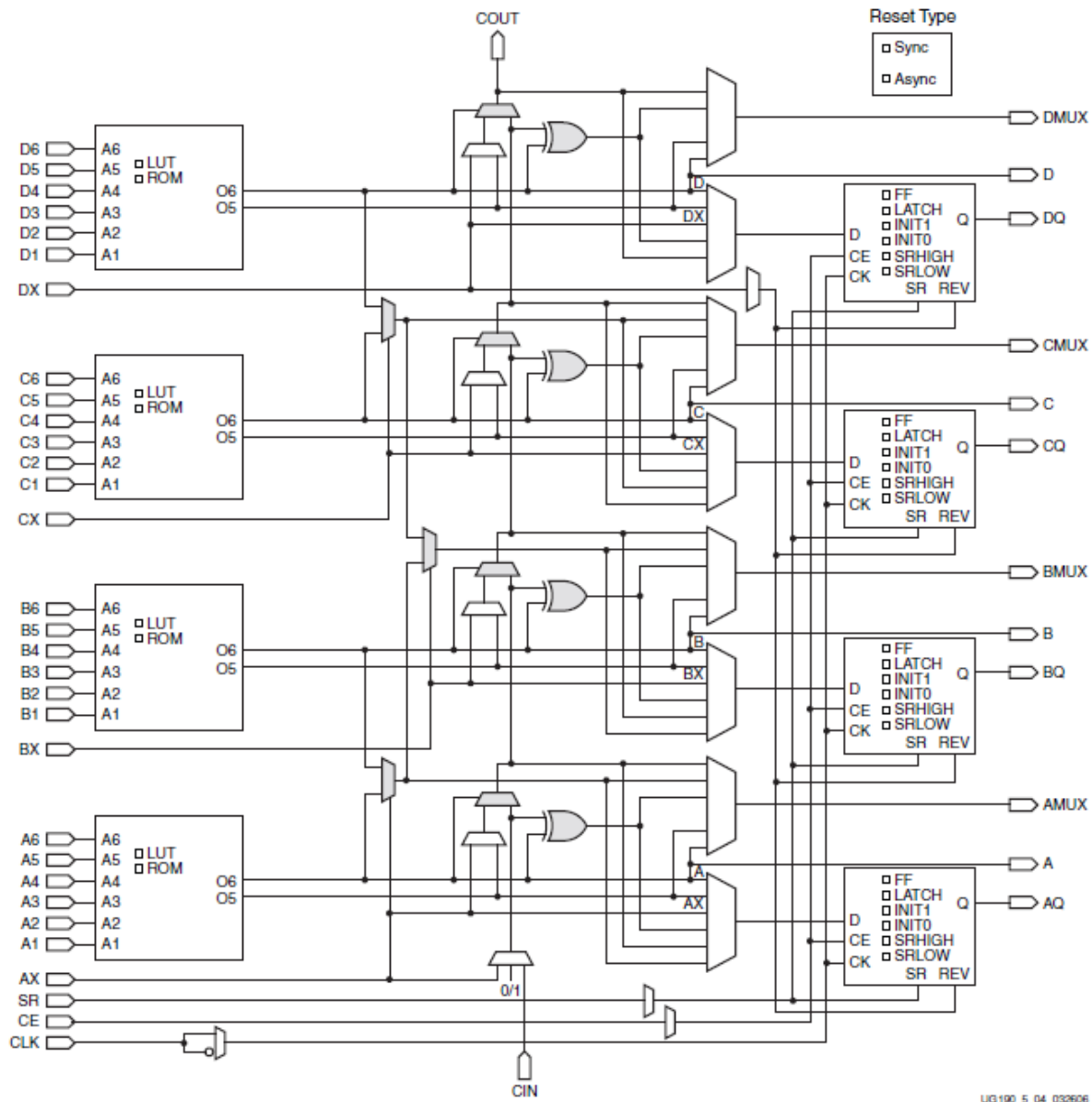
Indice : Révisez la diapositive #6 de la série 0601. Chacune des quatre tables de conversion de la tranche a deux sorties indépendantes, O6 et O5. Ces sorties peuvent correspondre à une fonction différente des entrées 1 à 5, ou bien la sortie O6 peut être une fonction des entrées 1 à 6.



UG190_5_04_032606

0603

Montrez comment implémenter l'opération $P = A \times 3$, où A est un nombre de 3 bits, sur une tranche L du Virtex 5 telle que donnée ici. Les signaux d'entrée sont $A(2:0)$. Les signaux de sortie sont $P(4:0)$.



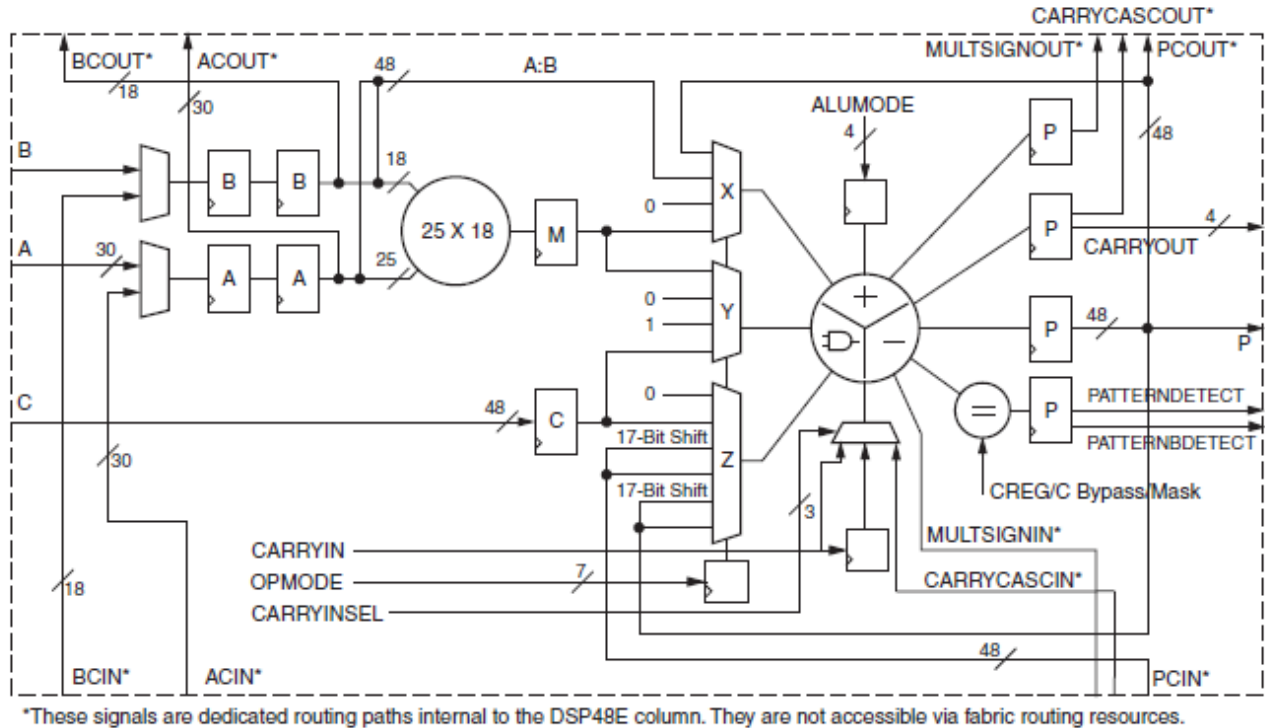
0604

Considérez le problème d'implémenter la multiplication par une constante. Montrez comment décomposer les facteurs suivants pour utiliser le moins d'additions possible pour effectuer une multiplication par ces facteurs.

- $\times 14$
- $\times 49$
- $\times 125$
- $\times 250$

0605

Considérez l'implémentation de la multiplication accumulation, $S = S + A \times B$. Montrez, sur un diagramme de la tranche DSP48E, comment l'utiliser pour implémenter cette opération.



0606

Considérez le code VHDL suivant. Pour chacun des signaux en sortie, estimez le nombre de ressources nécessaires en termes de tables de conversion (LUT), bascules (FF) et tranches DSP48 pour l'implémenter sur un FPGA de la famille Virtex 5 de Xilinx. Justifiez complètement votre réponse.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity ual_ressources is
  generic (
    W : positive := 8
  );
  port (
    clk : in std_logic;
    A, B : in signed(W - 1 downto 0);
    F, H, J : out signed(W - 1 downto 0);
    K : out signed(W + 2 downto 0);
    L : out signed(2 * W - 1 downto 0)
  );
end ual_ressources;
```

```
architecture arch of ual_ressources is
begin
  process(clk)
  begin
    if rising_edge(clk) then
      F <= A + B;
      H <= abs(A);
      J <= A / 4;
    end if;
  end process;

  K <= A * 5;
  L <= A * B;
end arch;
```

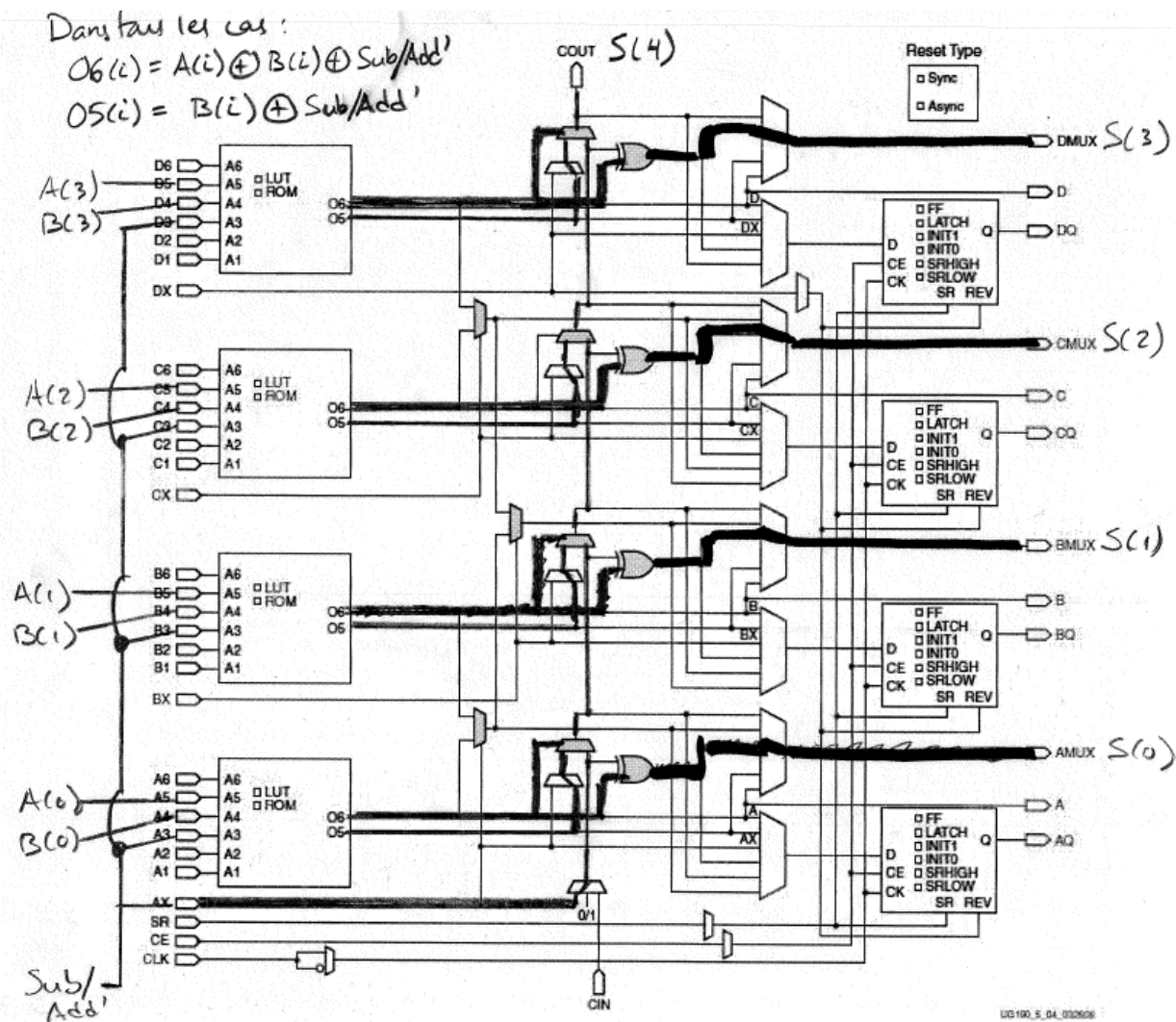

Solutions**0601**

Il suffit de démontrer que $T1'Y = T1'X = XY$.

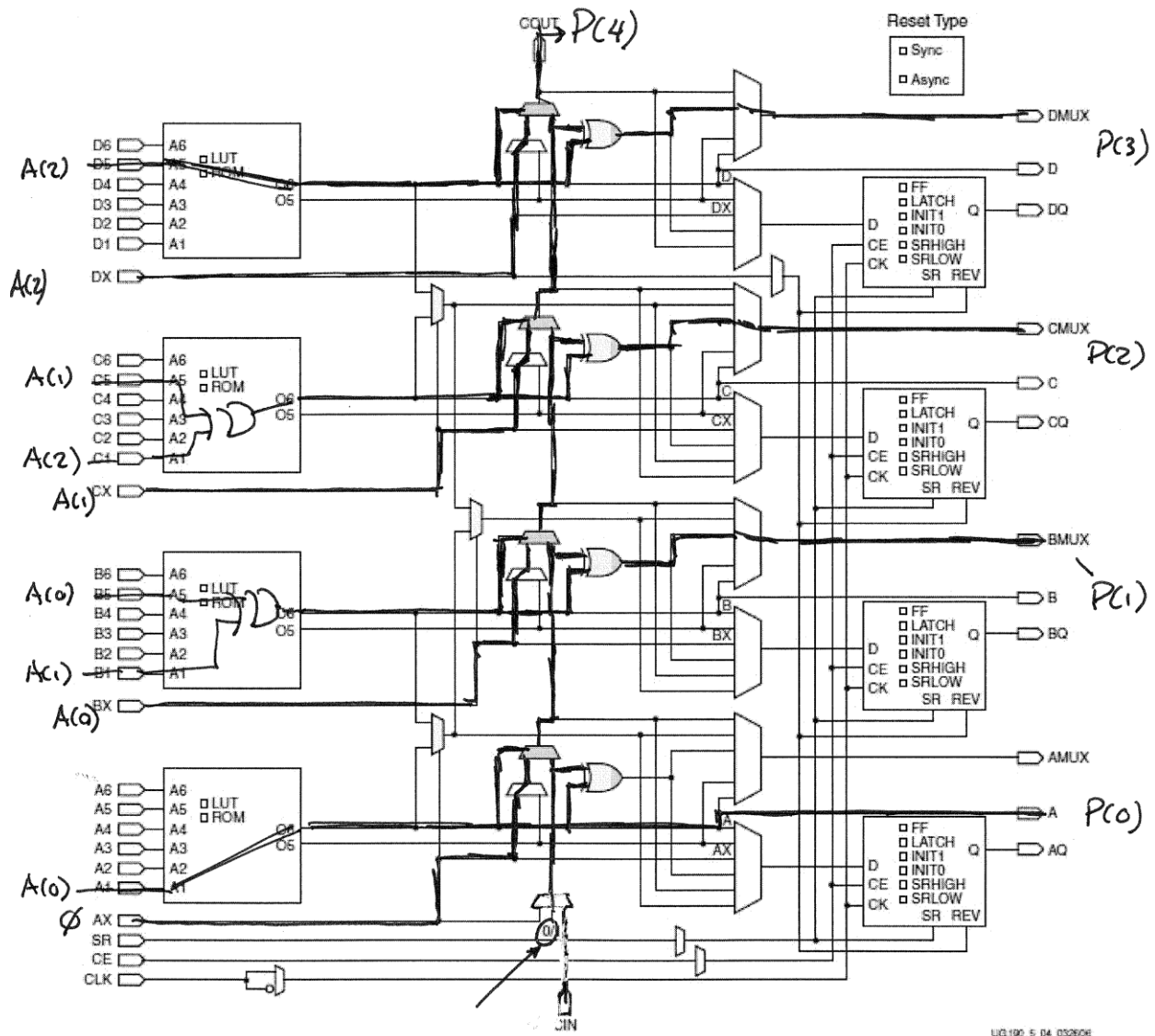
On a $T1 = X \text{ xor } Y$, donc $T1' = XY + X'Y'$

$T1'Y = XY + X'Y'Y = XY + 0 = XY$; $T1'X = XY + X'Y'X = XY + 0 = XY$. CQFD

On peut aussi faire la démonstration avec une table de vérité.

0602

0603



On fait $P = A \times 3 = A \ll 1 + A$, donc :

	0	A(2)	A(1)	A(0)
+	A(2)	A(1)	A(0)	0
	P(4)	P(3)	P(2)	P(1)
				P(0)

Quelques notes :

À strictement parler, l'étage pour $A(0) + 0$ n'est pas nécessaire. On pourrait relier la sortie $P(0)$ directement à $A(0)$.

Les ports cin et cout ne sont pas directement accessibles. Ils sont reliés de tranches en tranche. Pour obtenir $P(4)$ il faudrait passer par une sortie de la tranche supérieure à celle montrée ici.

Pour une addition, l'entrée cin doit être fixée à 0, et cette option est possible.

0604

a. $A \times 14 = A \times (16 - 2) = A \times 16 - A \times 2 = A \ll 4 - A \ll 1$

- b. $A \times 49 = A \times (32 + 16 + 1) = A \times 32 + A \times 16 + A = A \ll 5 + A \ll 4 + A$
- c. $A \times 125 = A \times (128 - 2 - 1) = A \ll 7 - A \ll 1 - A$
- d. $A \times 250 = A \times (256 - 4 - 2) = A \ll 8 - A \ll 2 - A \ll 1$

0605

On place la somme S dans le registre de 48 bits relié à la sortie P. On place les signaux A et B aux ports A et B. On relie la somme S au port C de 48 bits. On programme l'UAL pour faire une addition entre les sorties des multiplexeurs X et Y.

0606

Port	#LUT	#FF	#DSP48E	Justification
F	8	8	0	<ul style="list-style-type: none"> 1 LUT par paire de bits additionnés = 8 LUT Registre pour F
H	8	8	0	<ul style="list-style-type: none"> Valeur absolue est comme l'addition : 1 LUT par paire de bits additionnés = 8 LUT Registre pour H
J	0 ou 8	8	0	<ul style="list-style-type: none"> Division par 4 = décalage vers la droite de 2 positions, pas de calculs nécessaires Registre pour J
K	8 ou 11	0	0	<ul style="list-style-type: none"> $A \times 5 = A \times 4 + A \times 1$ Donc une addition est requise, à proprement parler sur 8 bits seulement; le décalage est gratuit
L	0	0	1	<ul style="list-style-type: none"> Multiplication générale, utilisation d'une tranche DSP48E

0607

Faites les opérations suivantes en BCD (en binaire) :

- e. $17 + 23$: s'effectue comme une addition décimale habituelle
- f. $45 - 46$:

-46 en complément à 10 est $953 + 1 = 954$

Donc l'opération à faire est $45 - 46 = 045 + 954 = 999$

$999 \geq 500$, donc représente un nombre négatif. On prend le complément à 10: $000 + 1 = 001$, qui est la bonne réponse (-1).

- g. $-150 - 22$:

-150 en complément à 10 est $849 + 1 = 850$

-22 en complément à 10 est $977 + 1 = 978$

Donc l'opération à faire est $850 + 978 = (1)828$

$828 \geq 500$, donc c'est un nombre négatif. On prend le complément à 10 : $171 + 1 = 172$, qui est la bonne réponse (-172)

0608

Selon le document '7 Series FPGAs Overview' de Xilinx.:

- Le FPGA XC7A35T contient 1800 Kb
- Pour conserver les réciproques, exprimées sur 16 bits, d'un diviseur qui peut prendre des valeurs dans l'intervalle $[0, 4095]$, il faudrait 4096 mots de 16 bits, soit 65536 bits.
- Le pourcentage de la mémoire bloc RAM totale utilisée sur le FPGA du labo serait $65536 / 1800000 \times 100$, environ 3.6 %.

