Unité: Systèmes logiques 2 (SysLog2)
Conception de systèmes numériques (CSN)

Arithmétique

Décomposition spatiale & temporelle

HAUTE ÉCOLE D'INGÉNIERIE ET DE GESTION DU CANTON DE VAUD

Etienne Messerli

Institut REDS, HEIG-VD

mars 2023

Contenu de la présentation

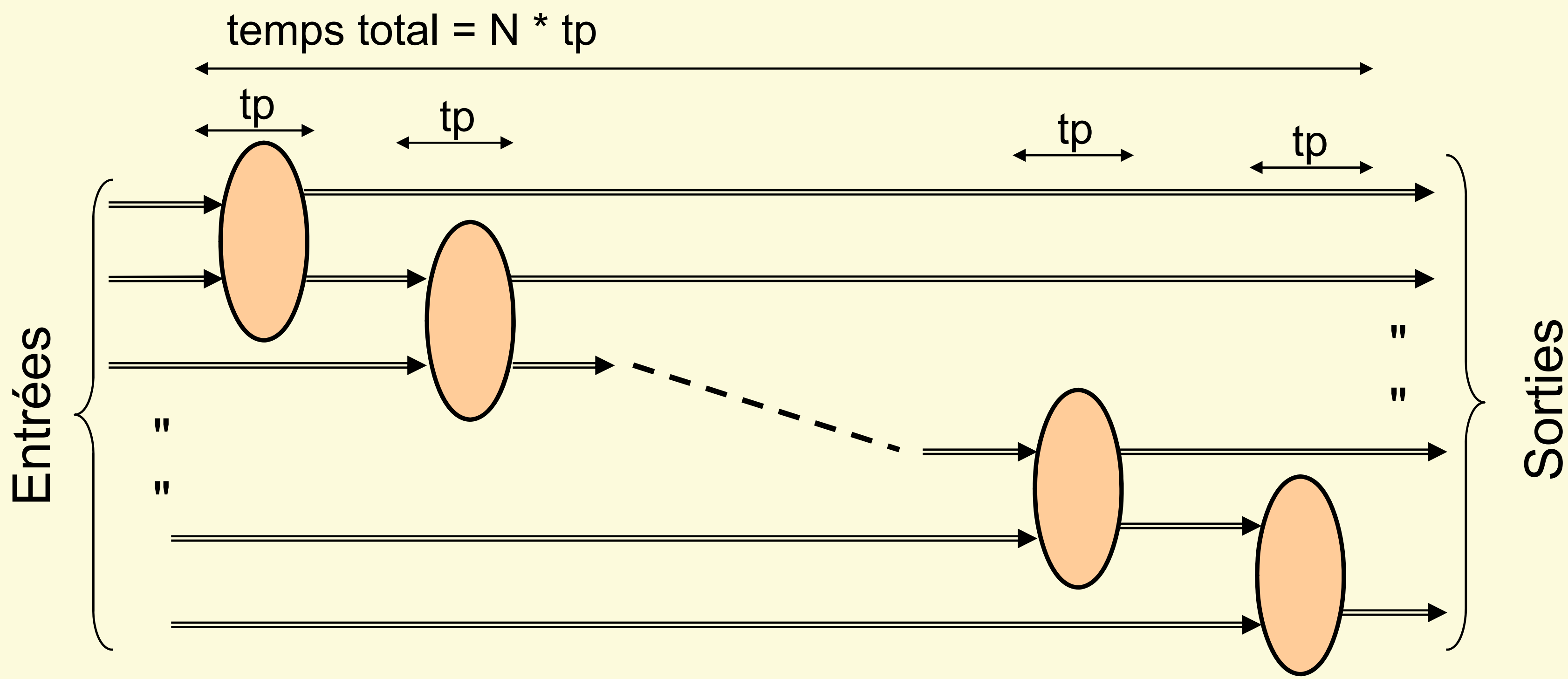
- Décomposition de circuits combinatoires
  - Décompositions spatiale et temporelle
- Addition:
  - Décomposition combinatoire & séquentielle
- Multiplication d'entiers binaires sans signe
  - Décomposition & Algorithmes
  - Réalisation combinatoire & séquentielles
- Multiplication d'entiers binaires signés C2
- Division d'entiers binaires sans signe
  - Décomposition & Algorithmes

Décomposition circuits combinatoires

- Réalisation de système de traitement de données comprend souvent des opérations arithmétiques, équations ou formules:
  - il s'agit souvent d'opérations combinatoires complexes
  - simplification de la conception via une décomposition
- Deux structures possible:
  - solution purement combinatoire : décomposition spatiale
  - solution séquentielle : décomposition temporelle

Décomposition spatiale …

- La complexité est décomposée en modules de base combinatoire qui sont ensuite interconnectés dans l'espace

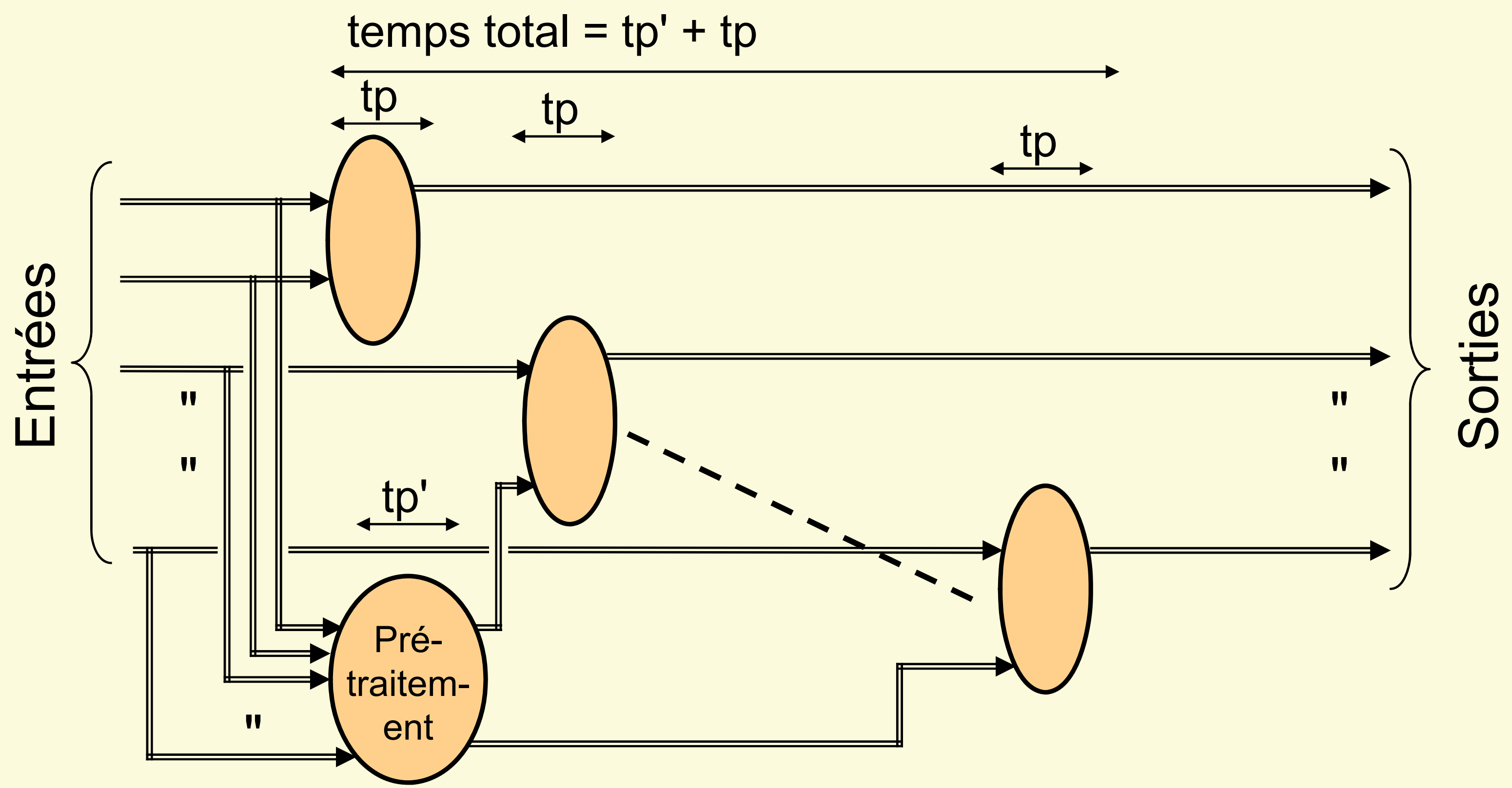


Décomposition spatiale : optimisation ...

- L'optimisation d'une décomposition spatiale nécessite de connaître les chemins critiques
  - soit ceux ayant le temps de propagation (tp) les plus importants
- Cas de l'addition:
  - Le chemin critique est le report
- L'optimisation est de trouver un ou plusieurs chemins parallèles qui permettent de couper ce chemin critique, pour diminuer le temps de calcul.
  - Nécessite du matériel supplémentaire.
  - Cas de l'addition: technique d'anticipation du report (carry look-ahead)

… décomposition spatiale : optimisation

- Un ou plusieurs modules en début de traitement permettent de supprimer le chainage des modules



Optimisation de l'addition

- Exemples d'optimisations:
  - Voir présentation Yann Thoma sur les additionneurs

Décomposition séquentielle

- Complexité est décomposée dans le temps, soit :
  - Utilisation d'un seul bloc qui est réutilisé plusieurs fois de façon séquentielle.
    - Celui-ci peut-être optimisé soit en quantité de logique ou en vitesse selon les besoins du projet.
  - Cela implique de gérer la connexion des valeurs d'entrées sur le bloc. Il faudra sélectionner les valeurs à traiter à un instant donné.
  - A chaque étape il sera nécessaire de mémoriser les résultats intermédiaires.

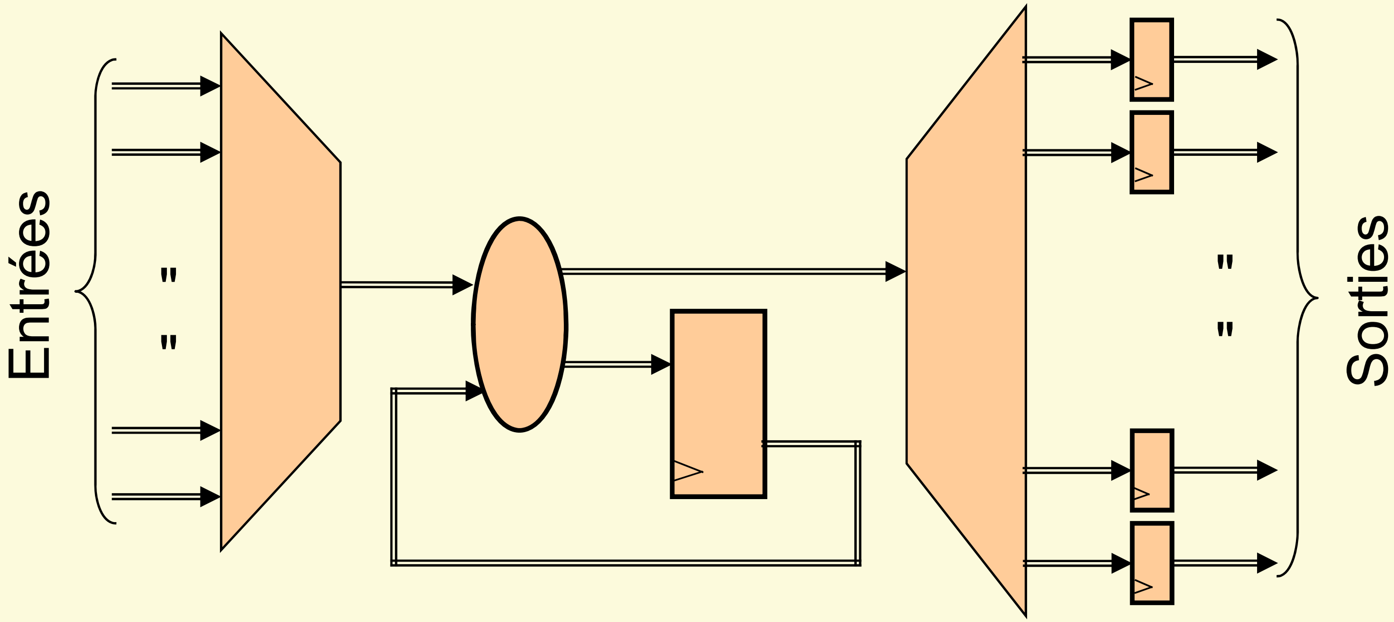
Addition: exemple de décomposition séquentielle

- Il est possible de décomposé une addition, opération combinatoire, de façon séquentielle. Soit :
  - Un processeur dispose d'ALU à taille fixe.
  - Etendre l'addition nécessite de faire des additions successives.
  - Indispensable de mémoriser et de chaîner le carry entre chaque étape.
  - L'overflow n'est utile que lors de la dernière addition, si les nombres sont signés, il n'est pas nécessaire de le chaîner.

Décomposition temporelle d'un système combinatoire

- La décomposition utilise un seul bloc combinatoire de base qui est utilisé séquentiellement pour réaliser le calcul par étapes successives.

temps total = N * Tclk



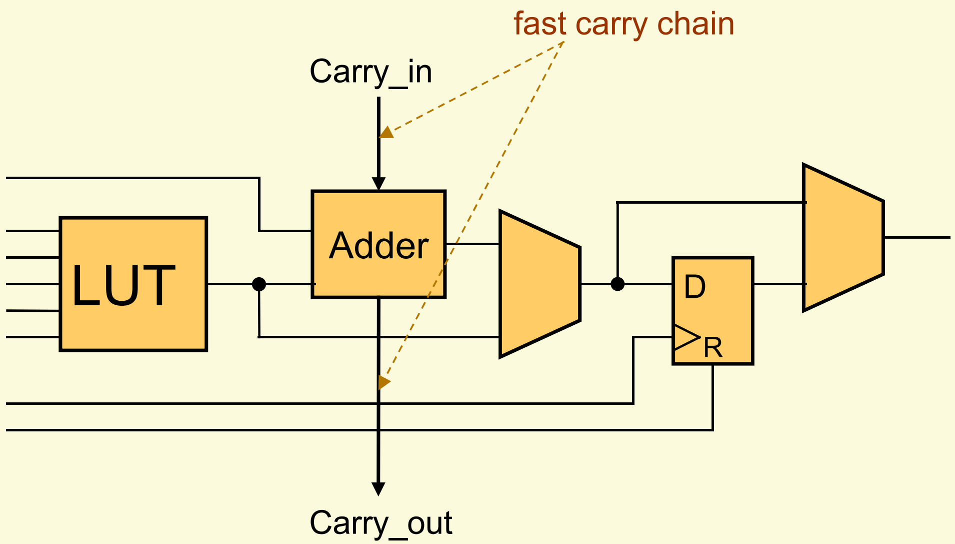
Décomposition temporelle : optimisation

- La partie principale à optimiser est le bloc combinatoire implémenté une seule fois:
  - Paralléliser le traitement même si cela augmente la quantité de matérielle.
  - Utiliser un bloc RAM du PLD, solution parfois plus rapide que des blocs logiques
- Concernant les registres :
  - seul un changement de technologie permet d'améliorer les performances
- Remarque pour les PLDs récents:
  - tp connexion > tp logique (LEs)
  - tp total dépendant fortement du routage !

Décomposition temporelle : optimisation

- Utilisation du registre modifie peu le temps de propagation dans le "Logic element" (LE)

Bloc logique de base d'une FPGA :
LE = Logic Element



LUT: Look-Up Table
de 4 à 6 entrées

Comparaison spatiale vs temporelle

Comparaison des performances d'une décomposition spatiale vs temporelle

Hypothèse: bloc de base avec une taille identique

temps total décomposition spatiale = N * tp_comb

temps total décomposition temporelle = N * tclk

- Quelle est l'écart entre tp_comb & tclk ?
- Dans quelle décomposition le tp de connexion peut être élevé ?
- Quelle décomposition est la plus performante ?

Exemple de décomposition

- Exemple de décomposition spatiale et séquentielle avec la comparaison de 2 nombres.

voir présentation séparée

Multiplication

- Réalisation d'une multiplication en binaire ?
- 1ère solution:
  - Utiliser une ROM qui contient toutes les combinaison de la multiplication (TDV)
  - Rapide
  - Couteuse en matériel (taille de la mémoire)
- 2ème solution:
  - Décomposition de la multiplication en opérations simples

Multiplication à la main : exemple

```
       1 2 5
    x  1 3 9
    --------
     1 1 2 5 ← 9x5 + 9x2x10  + 9x1x100
     3 7 5 0 ← 10x (3x5 +3x2x10 + 3x1x100)
   1 2 5 0 0 ← 100x (1x5  +1x2x10 + 1x1x100)
   ---------
   1 7 3 7 5
```

La multiplication de 2 nombres est réalisée à l'aide d'une suite d'opérations plus simples

Multiplication en décimal : décomposition

- En décimal, la multiplication peut être décomposée en
  - des multiplications Mcande x chiffre du Mcateur
    - des multiplications chiffre x chiffre (génèrent un report !)
    - des décalages (multiplications par la base, tiennent compte du poids des chiffres du Mcande)
    - des additions (avec reports)
  - des décalages (multiplications par la base, tiennent compte du poids du chiffre du Mcateur)
  - des additions (avec reports)

Multiplication en binaire…

- En binaire, la multiplication d'un chiffre par un chiffre se résume au livret suivant

 0 x 0 = 0

 0 x 1 = 0

 1 x 0 = 0

 1 x 1 = 1 il n’y a pas de report !

- Ce livret correspond à la TDV de la fonction logique ET !

… multiplication en binaire…

- En binaire, la multiplication est **plus simple**. Elle peut être décomposée en
  - des multiplications Mcande x chiffre du Mcateur
    - des multiplications chiffre x chiffre (pas de report)
    - des décalages (multiplications par la base, tiennent compte du poids des chiffres du Mcande)
  - des décalages (multiplications par la base, tiennent compte du poids du chiffre du Mcateur)
  - des additions (avec reports)

…multiplication en binaire…

```
        1 0 1 0
      x 1 0 0 1
      ---------
        1 0 1 0 ←           1010  x  1
      0 0 0 0 0 ← 10 x      1010  x  0
    0 0 0 0 0 0 ← 100 x     1010  x  0
  1 0 1 0 0 0 0 ← 1000 x    1010  x  1
  -------------
  1 0 1 1 0 1 0
```

La multiplication est réalisée à l'aide d'une suite d'opérations simples.

…multiplication en binaire

- La multiplication par un bit sont des portes ET
- La multiplication par le poids du bit concerné du multiplicateur sont des décalages à gauche
  - exemple :
    la multiplication du multiplicande par le bit de poids 2^3 du multiplicateur est un décalage à gauche de 3 positions (multiplication par 2^3)
- Finalement les résultats intermédiaires sont additionnés en tenant compte de leur décalage

1er algorithme de multiplication

- La multiplication à la main peut être traduite par l'algorithme suivant :

```
Resultat:= 0;
i:=0;
while i < Nb_Bits_Mcateur loop
  if Mcateur(i)=1 then --bit de rang i
    Resultat:= Resultat + Mcande;
  end if;
  Mcande:= Mcande*2;
  i:=i+1;
end loop;
```

Décomposition combinatoire & séquentielle

- Toute fonction combinatoire peut-être décomposée dans l'espace, on parle d'une
  décomposition combinatoire/ spatiale
  - dans le cas de la multiplication, il s'agit de multiplicateur 1 bit (porte ET) et d'additionneurs
- La décomposition peut aussi être dans le temps, on parle d'une
  décomposition séquentielle/ temporelle
  - Il s'agit dès lors de minimiser le matériel
  - Il faut mémoriser les résultats intermédiaires

Multiplication combinatoire (spatiale)

- L'algorithme ci-dessus peut être implémenté sous forme d'un circuit combinatoire
  - Chaque opération requiert un exemplaire de l'opérateur concerné (ET, addition)
  - Les décalages s'obtiennent par câblage
  - La succession des opérations est obtenue par le passage des données à travers les divers opérateurs, dans l'ordre où ils sont câblés
- La boucle de l'algorithme est déroulée dans l'espace
  - décomposition spatiale (combinatoire)

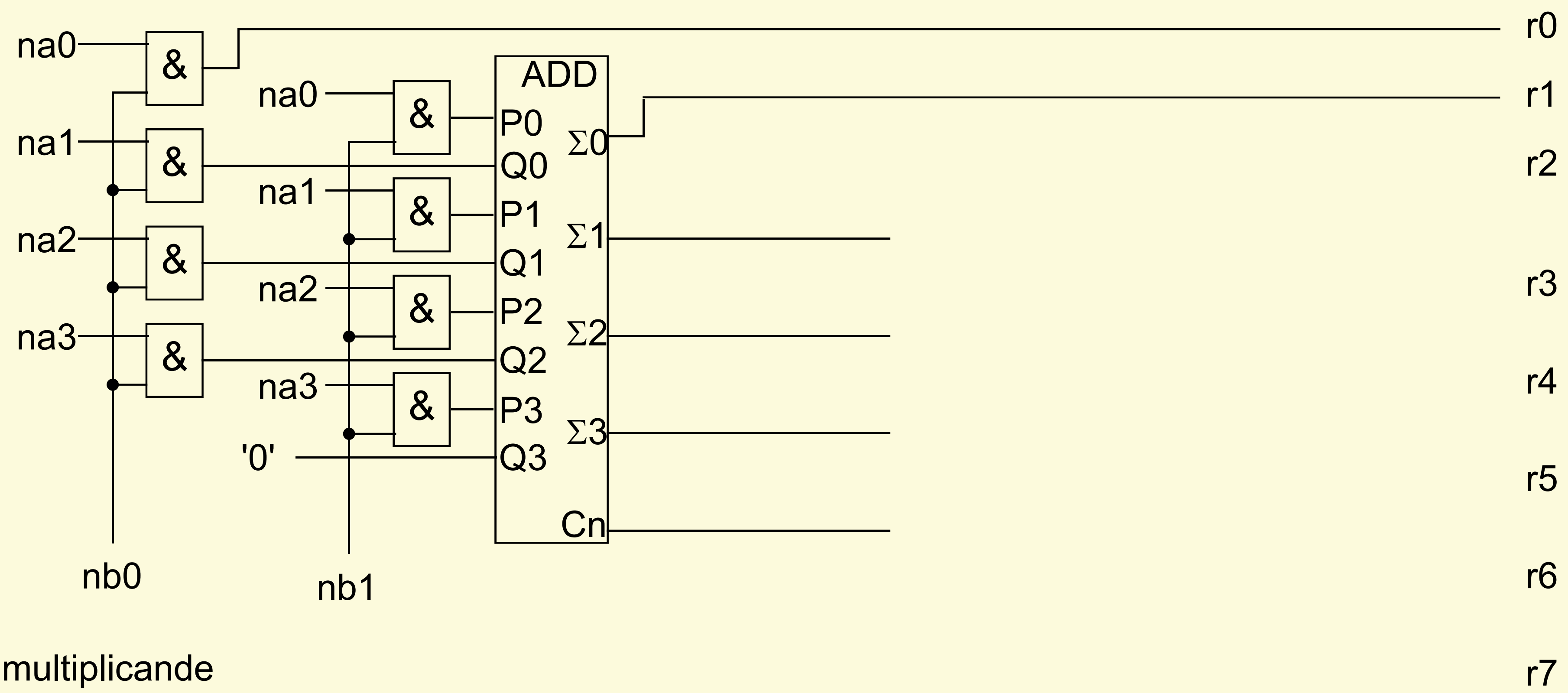
Multiplication combinatoire (spatiale)

Décomposition de la multiplication à l’aide d’une suite d'addition et de multiplication binaire :

1 0 1 0			
x 1 0 0 1			
1 0 1 0	←		1010 x 1
0 0 0 0 0	←	10 x	1010 x 0
0 0 0 0 0 0	←	100 x	1010 x 0
1 0 1 0 0 0 0	←	1000 x	1010 x 1
1 0 1 1 0 1 0			

Prévoir de gérer les dépassements entre chaque étage

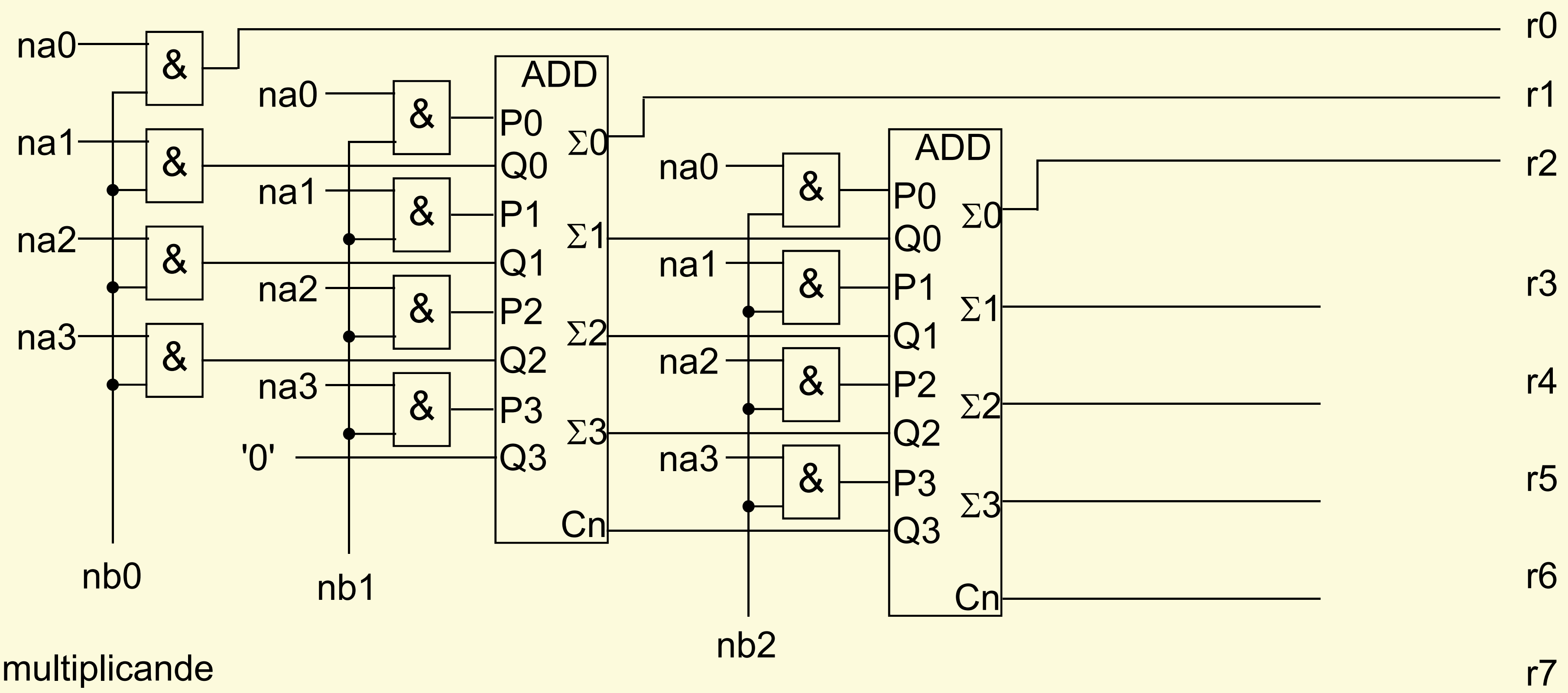
Exemple : multiplication 4 bits x 4 bits



na : multiplicande

nb : multiplicateur

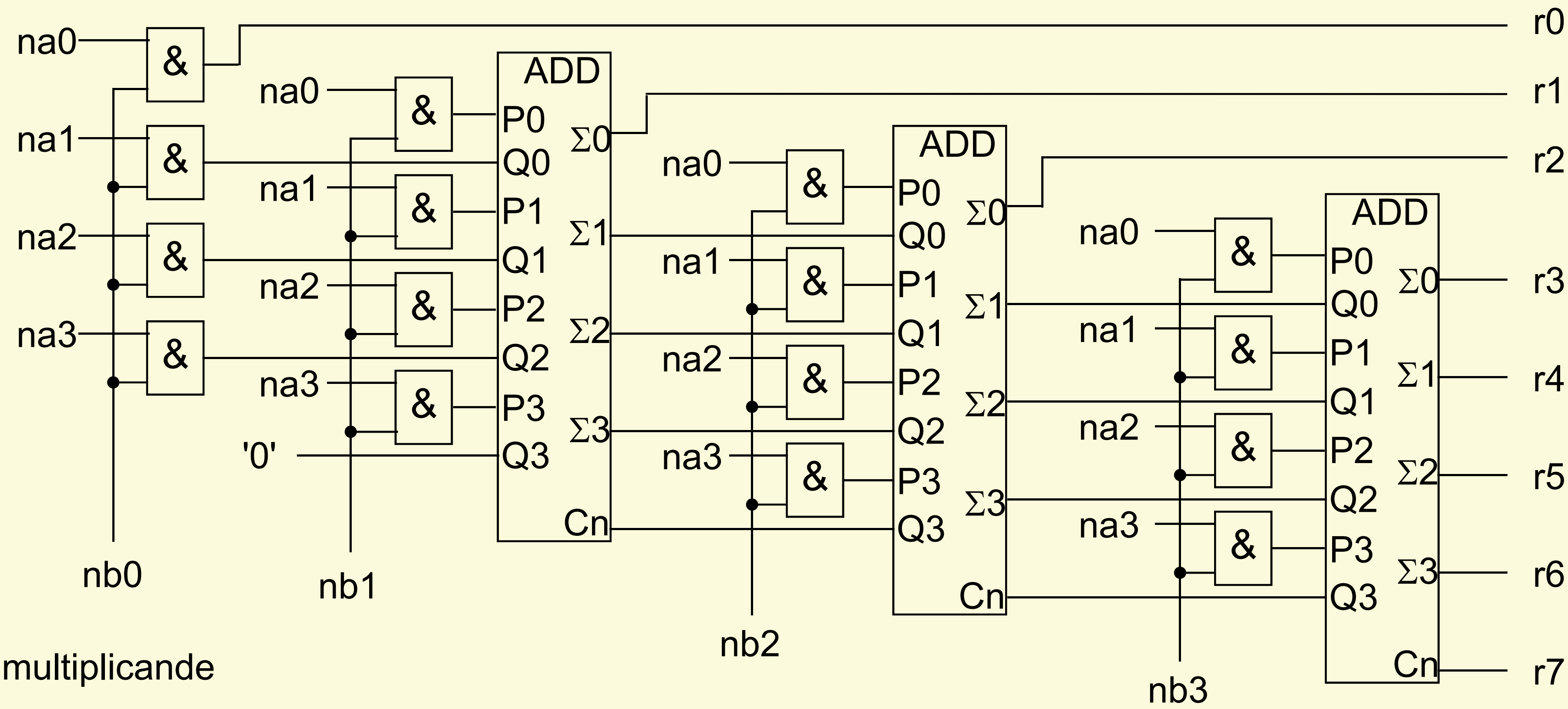
Exemple : multiplication 4 bits x 4 bits



na : multiplicande

nb : multiplicateur

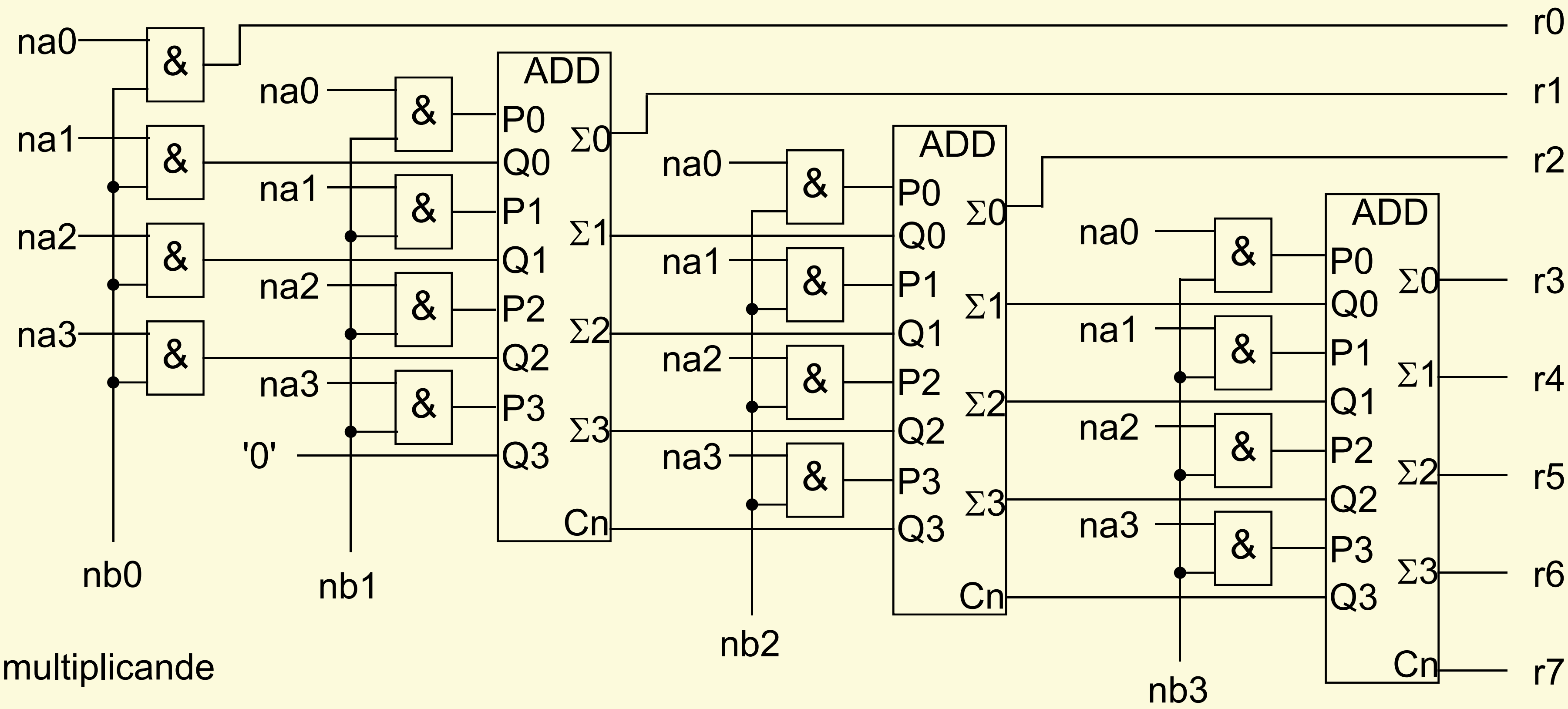
Exemple : multiplication 4 bits x 4 bits



na : multiplicande

nb : multiplicateur

Exemple : multiplication 4 bits x 4 bits



na : multiplicande

nb : multiplicateur

Multiplication séquentielle…

- Il est également possible de n'utiliser
  - qu'une seule rangée de fonctions ET pour calculer le produit du Mcande par un bit du Mcateur
  - qu'un seul additionneur pour additionner les résultats partiels successifs
- mais il faut alors répéter 4 fois la séquence
  - multiplier Mcande par 1 bit Mcateur et décaler
  - additionner au résultat partiel précédent
  - mémoriser le nouveau résultat partiel
  - passer au bit suivant du Mcateur

…multiplication séquentielle…

- Dans le cas général de la multiplication de deux nombre de n bits donné par l'algorithme de la page 20, nous avons besoin de :
  - ET 2n bits x 1 bit
  - Additionneur 2n bits (résultat sur 2x n bits)
  - Registre parallèle/série 2n bits pour le Mcande
  - Registre parallèle/parallèle 2n bits pour le résultat
  - Registre parallèle/série n bits pour le Mcateur (ou mux pour sélectionner les bits un à un)

…multiplication séquentielle…

- Optimisation :
  - Lors de la décomposition spatiale de la page 23, nous pouvons constater que :
    - Additionneur sur n bits
    - Un bit du résultat à chaque pas
    - Report de l'additionneur utilisé pour le prochain pas
  - Nous pouvons aussi constater que :
    - A chaque étape du calcul séquentiel, il y a un bit de plus pour le résultat et un de moins pour le multiplicateur
    - Disposer d'un registre parallèle/série de 2n bits pour le résultat et le multiplicateur

…multiplication séquentielle…

- Finalement la multiplication de deux nombres de n bits nécessite :
  - ET n bits x 1 bit
  - Additionneur n bits
  - Une bascule pour mémoriser le report
  - Registre parallèle/série 2n bits pour le résultat/Mcateur (ou deux registres n bits liées pour le décalage)
  - Registre parallèle/parallèle n bits pour le Mcande

…multiplication séquentielle…

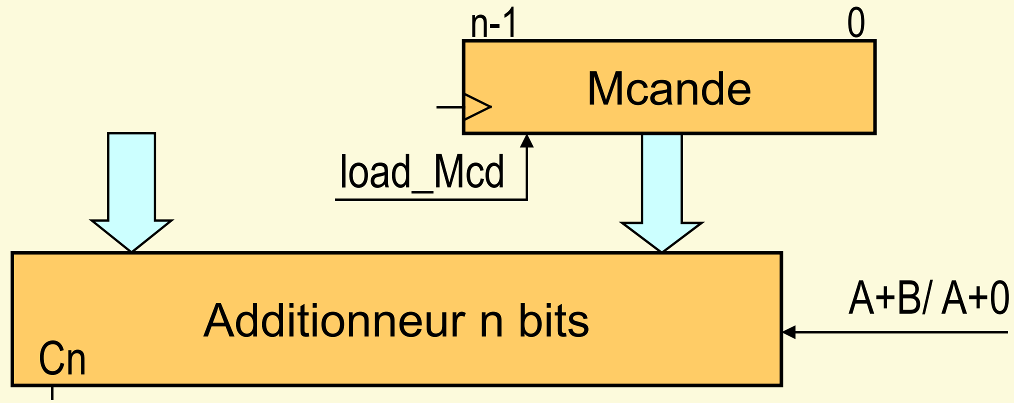
- 2ème algorithme pour la multiplication :

```
Initialiser: Mcateur et Mcande

Result_Haut:= 0;
for i in 1 to n loop
  if Bit_Poids_Faible(Mcateur) = 1 then
    Result_Haut:= Result_Haut + Mcande;
    --report est mémorisé dans un flip-flop
  else
    Result_Haut:= Result_Haut + 0;
  end if;
  Decale_a_Droite(Report, Result_Haut, Mcateur);
  --report precedent va dans MSB de Result_Haut
end loop;
```

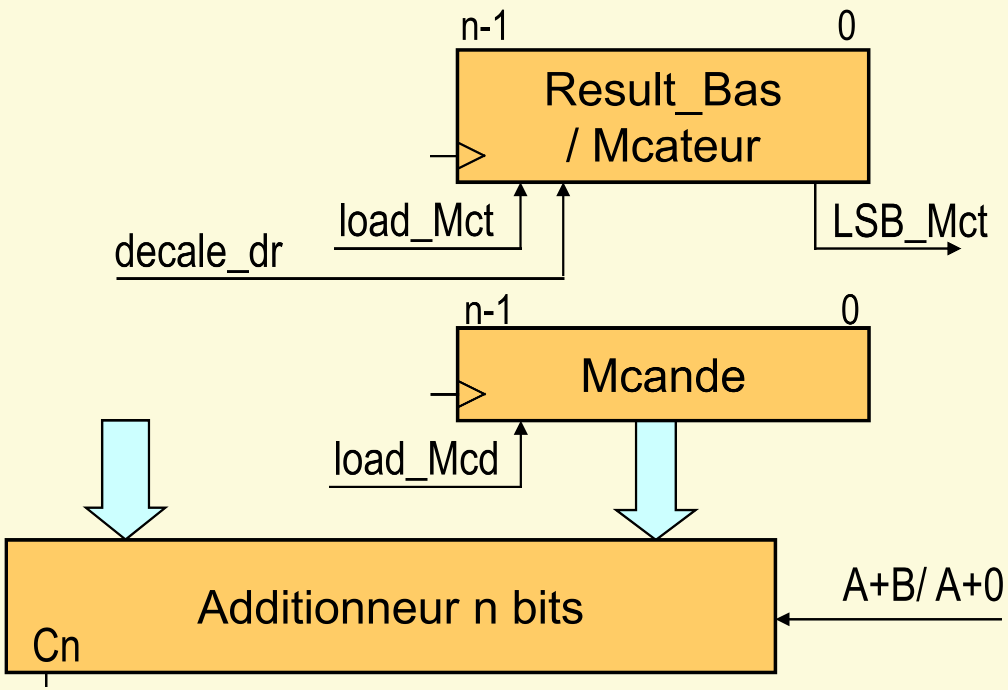
…multiplication séquentielle

- Schéma bloc de l'UT pour la multiplication séquentielle



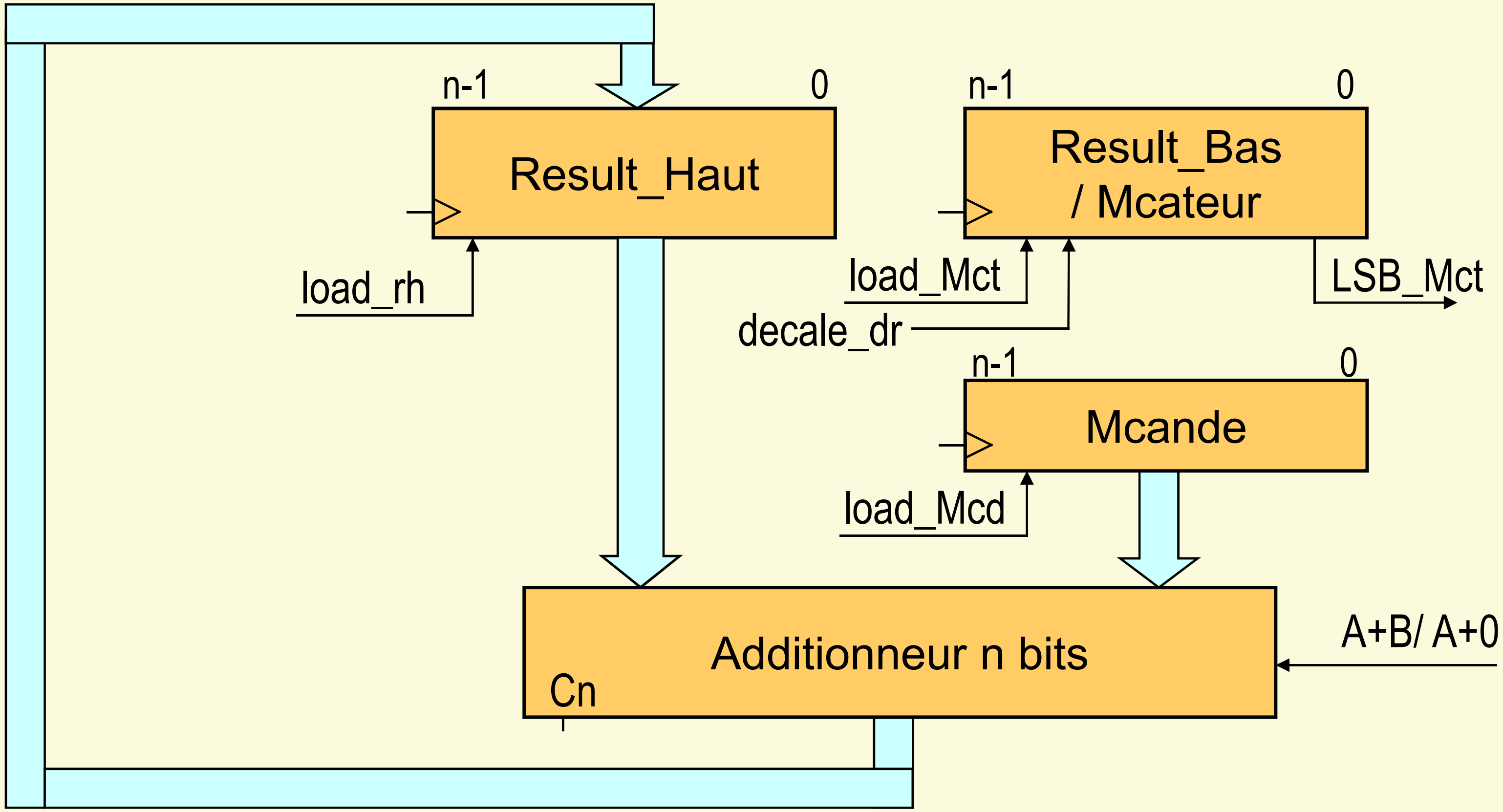
…multiplication séquentielle

- Schéma bloc de l'UT pour la multiplication séquentielle



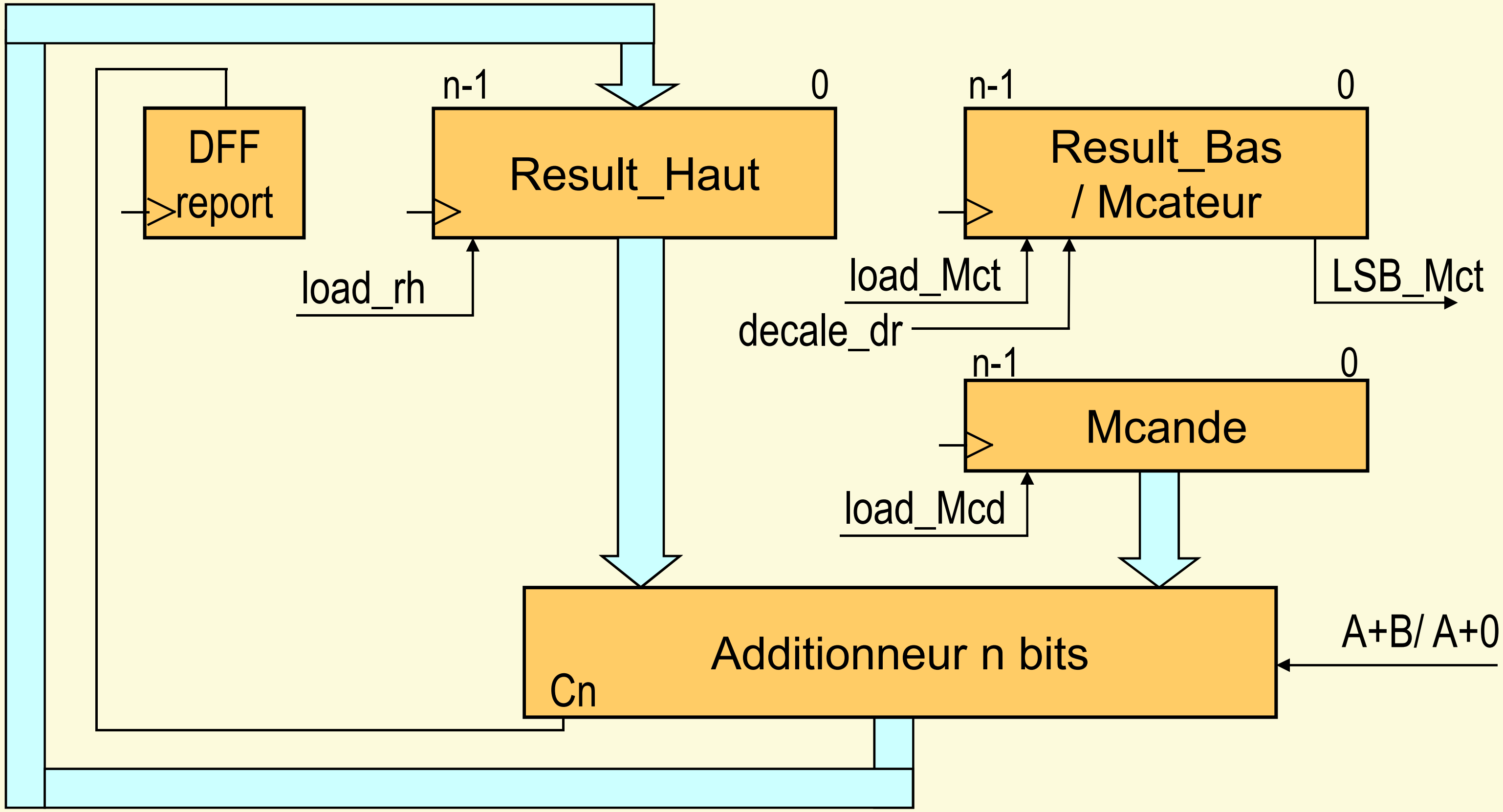
…multiplication séquentielle

- Schéma bloc de l'UT pour la multiplication séquentielle



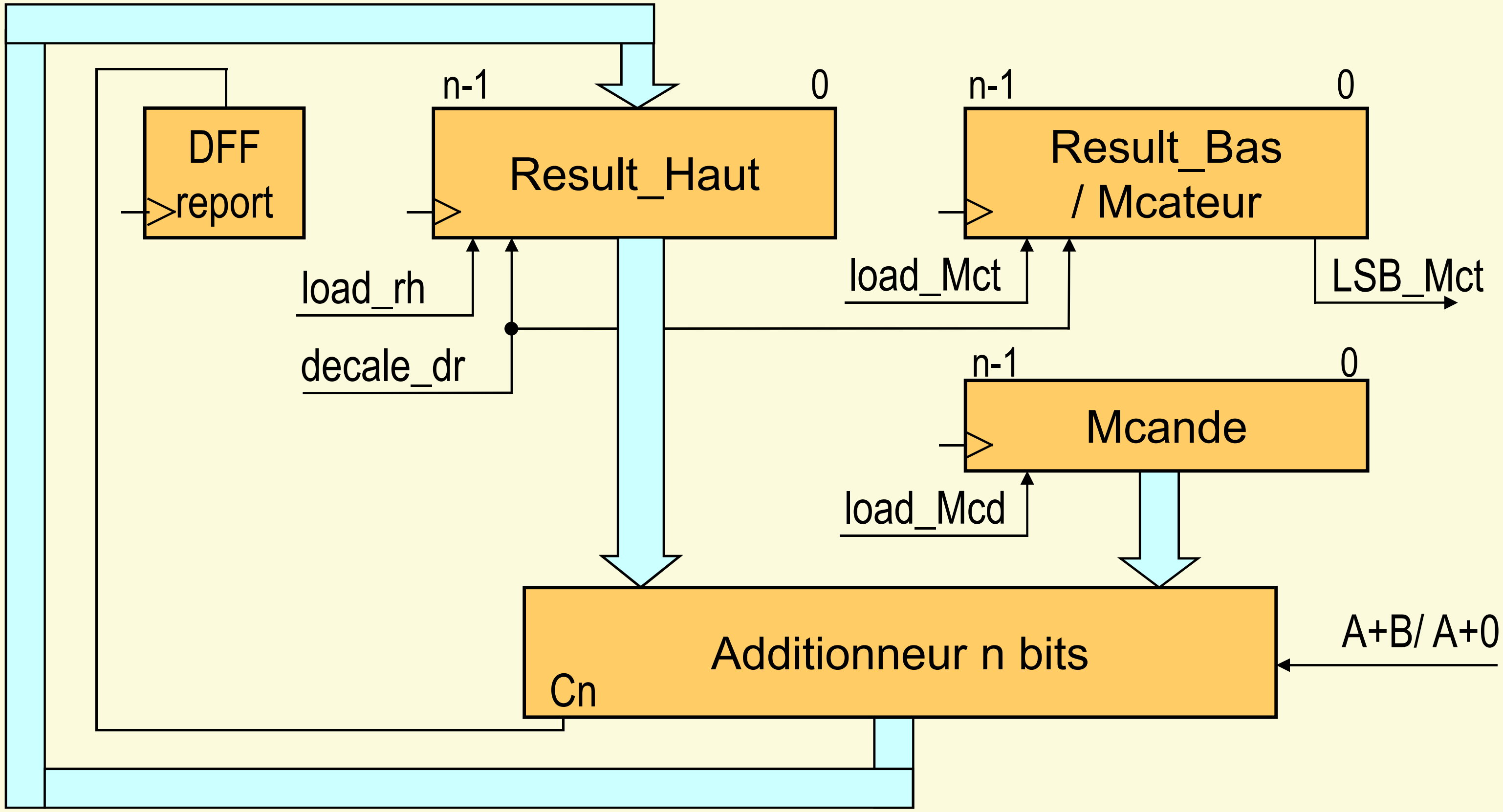
…multiplication séquentielle

- Schéma bloc de l'UT pour la multiplication séquentielle



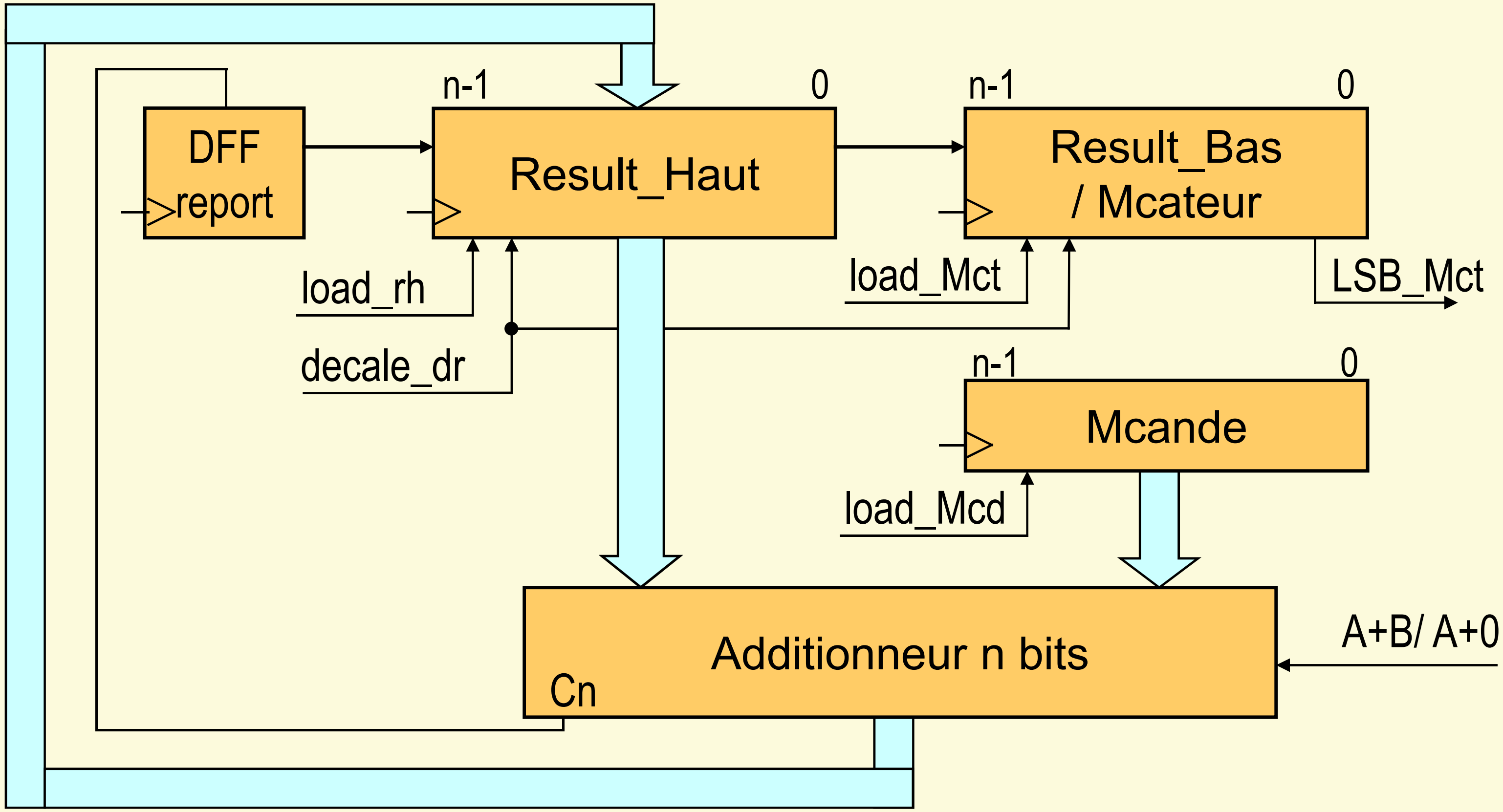
…multiplication séquentielle

- Schéma bloc de l'UT pour la multiplication séquentielle



…multiplication séquentielle

- Schéma bloc de l'UT pour la multiplication séquentielle



Exemple avec "Numlab"

- Etudier la multiplication spatiale (sans signe)
  - 1001 x 0101; 1100 x 0110; 1101 x 1011
- Etudier la multiplication séquentielle (sans signe)
  - 1001 x 0101; 1100 x 0110; 1101 x 1011

Exercices I

1. Nous souhaitons réaliser une système qui calcul le nombre total d'heures travaillées durant une semaine. Nous disposons du nombre d'heures travaillées durant le jour en cours (nbhr) et du nombre de jours travaillés depuis le début de la semaine (nbjr).
   - Vous devez concevoir et réaliser le système calculant le nombre total d'heures travaillées durant la semaine.
   - Vous utiliserez le principe de la décomposition spatiale pour la multiplication
   - Le nombre d'heures travaillées par jour est de 8.

Exercices I

2. Nous souhaitons réaliser une système qui calcul le nombre total d'heures (tothr) travaillées durant le mois. Nous disposons du nombre de semaine travaillées (nbsem) et du nombre de jour de la semaine en cours (nbjr).
    - Vous devez concevoir et réaliser le système calculant le nombre total d'heures travaillées durant le mois.
    - Vous utiliserez le principe de la décomposition spatiale pour la multiplication
    - Le nombre d'heures travaillées par semaine est de 40, et le nombre d'heures travaillées par jour est de 8.

Exercices I

3. Modifier le schéma de la page 30 afin de pouvoir réaliser l'opération d'addition et de décalage en un seul coup d'horloge. Compléter le schéma avec le matériel nécessaire.

Multiplication d'entier signé …

- Application de l'algorithme utilisé pour les nombres non signés

Nombres non signés

```
  11
x 13
----
  33
 11
----
 143
```

OK
<=

```
     1011
   x 1101
   ------
     1011
   0 000
  10 11
 101 1
---------
1000 1111
```

Nombres signés

```
1011 = -5
1101 = -3
```

Faux
=> -113

```
  -5
x -3
----
  15
```

Impossible d'utiliser le **même circuit** pour la multiplications non signé & signé !

… multiplication d'entier signé …

- Nombre signé : notation en complément à 2

  $- A = C_2(A) = 2^n - A$

- Multiplication avec nombre signé
  - utiliser l'algorithme pour nombre non signé et traiter le signe séparément
  - adapter l'algorithme pour pouvoir l'utiliser avec des nombres signés

… multiplication d'entier signé …

- Multiplication avec nombre signé :

  cas $B * A$ avec $A < 0$ d'où

  $$B * A = B * C_2(|A|) = B * (2^n - |A|) = \underbrace{B * 2^n}_{\text{terme en trop!}} + B * (-|A|)$$

  d'où :

  $B * (-|A|) = B * A - B * 2^n$ le second terme est la correction

- Calculs intermédiaires :
  - nombres signés => extension du signe !

… multiplication d'entier signé …

- Correction de l'algorithme

```
         1011 = -5
       x 1101 = -3
    1111 1011
  + 0000 000
  + 1110 11
  + 1101 1
    1011 1111
  - 1011 0000
    0000 1111 = +15
```

Résultats partiels signés : extension du signe

correction –(multiplicande * 2^n)

```
   -5
x  -3
  +15
```

< = >

OK

… multiplication d'entier signé …

- Optimisation de l'algorithme
  - Il y a une correction seulement si multiplicateur négatif, soit lorsque le MSB est à '1'

    dans ce cas il y a l'opération

    $+ B * 2^{n-1}$

    qui est suivie par la correction

    $- B * 2^{n}$
  - il est possible de simplifier ces deux opérations, soit :

    $+ B * 2^{n-1} - B * 2^{n} = B * 2^{n-1} – 2 * B * 2^{n-1} = - B * 2^{n-1}$

… multiplication d'entier signé …

- Algorithme de Robertson

```
Result_Haut:= 0;
for i in 1 to n-1 loop   -- n = nombre de bits
  if Bit_Poids_Faible(Mcateur) = 1 then
    Result_Haut:= Result_Haut + Mcande;
  else
    Result_Haut:= Result_Haut + 0;
  end if;
  Decale_a_Droite(Ovr xor MSBit, Result_Haut, Mcateur);
end loop;
if Bit_Poids_Faible(Mcateur)= 1 then
  Result_Haut:= Result_Haut - Mcande; --cor. anticipée
else
  Result_Haut:= Result_Haut - 0;
end if;
Decale_a_Droite(Ovr xor MSBit, Result_Haut, Mcateur);
```

… multiplication d'entier signé …

- Exemple avec algorithme de Robertson

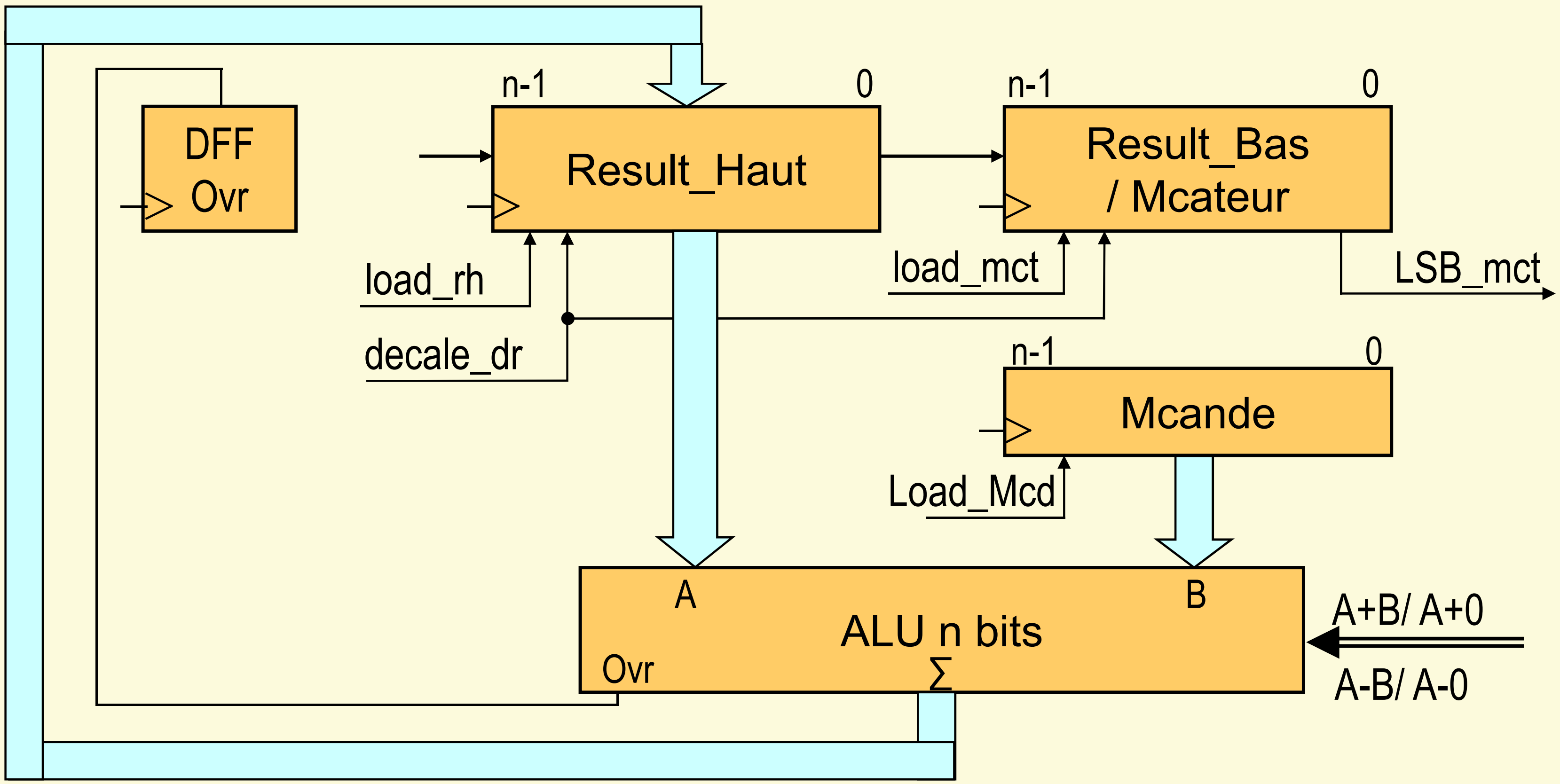
```
       1011  = -5
     x 1101  = -3
  ----------
  1111 1011
+ 0000 000
+ 1110 11
- 1101 1
  ----------
  0000 1111  = + 15
```

correction anticipée: –(multiplicande * 2^{n-1})

```
  -5
x -3
----
 +15
```

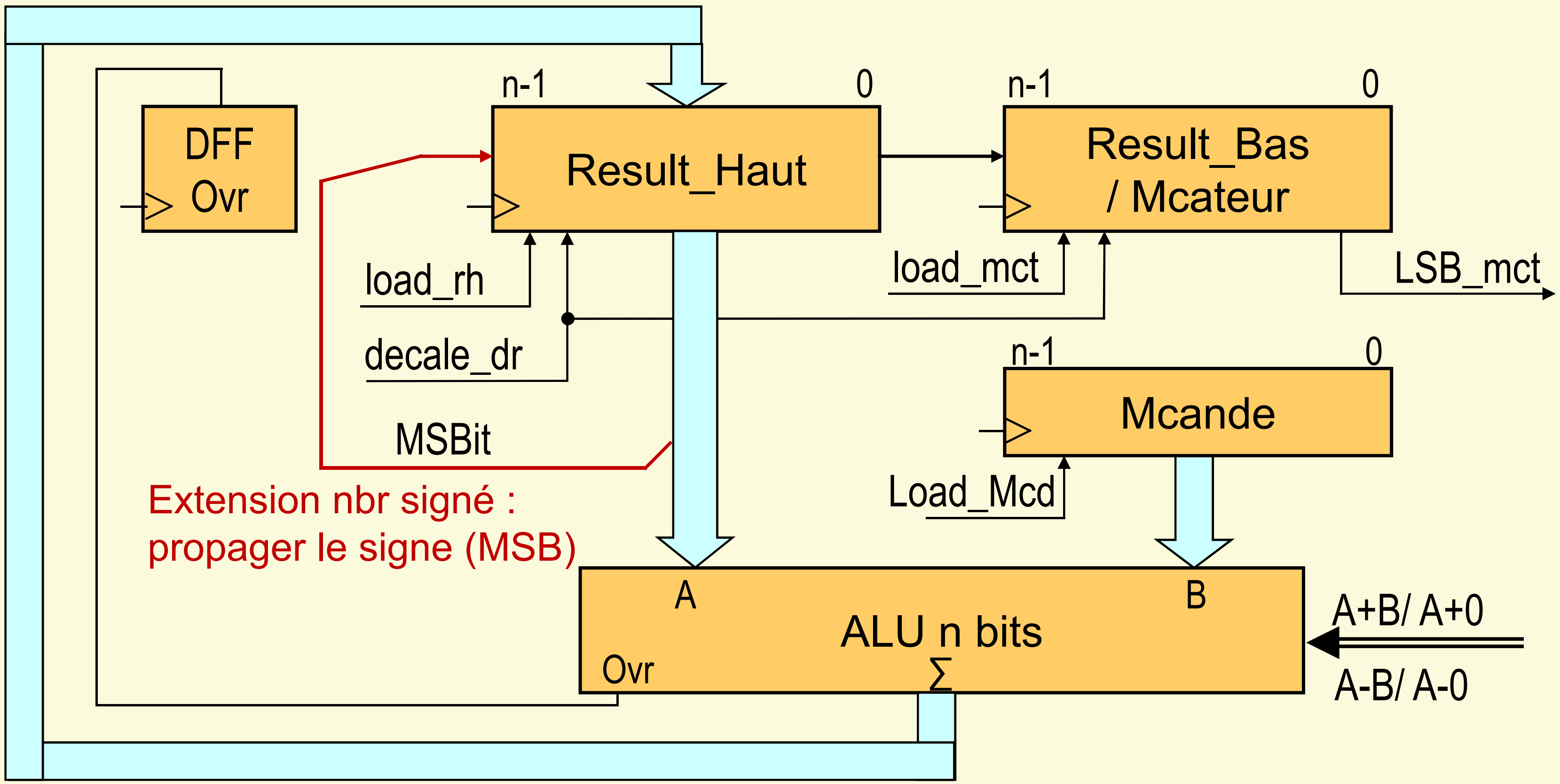
… multiplication d'entier signé

- Schéma bloc de l'UT pour la multiplication d'entier signé



… multiplication d'entier signé

- Schéma bloc de l'UT pour la multiplication d'entier signé

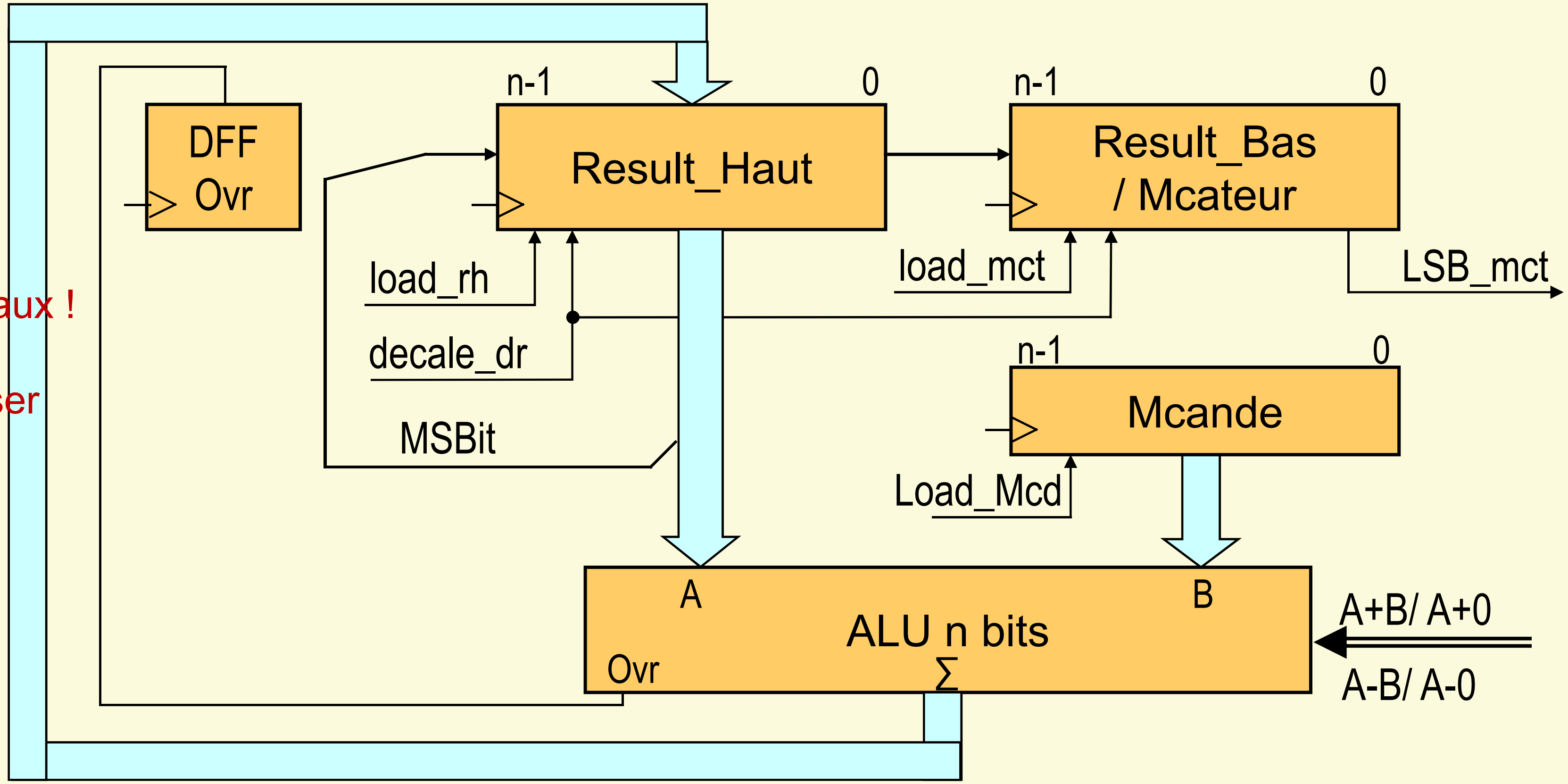


… multiplication d'entier signé

- Schéma bloc de l'UT pour la multiplication d'entier signé

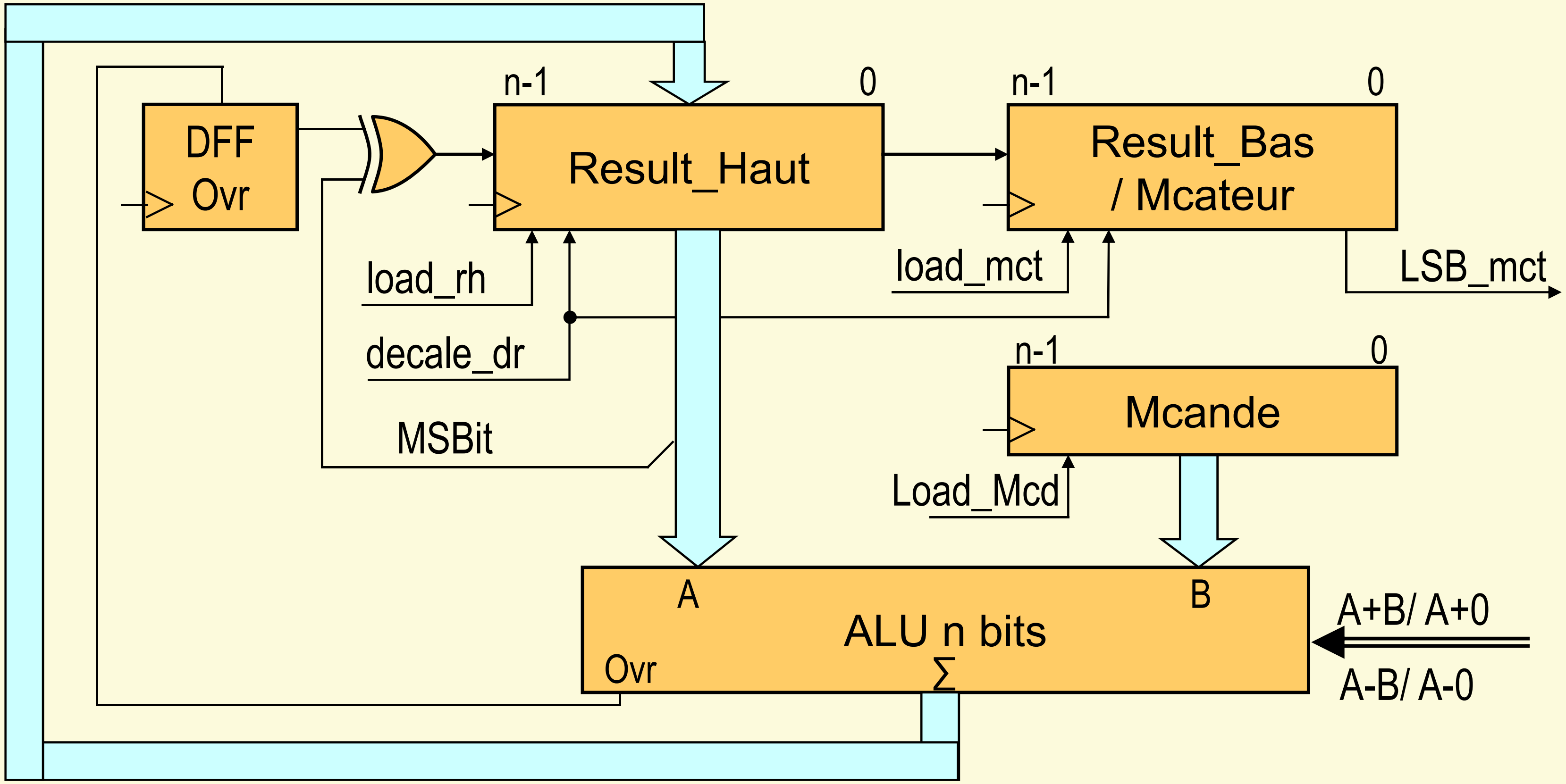
Ovr = '1' indique que le signe du résultat est faux !

Il faut l'inverser



… multiplication d'entier signé

- Schéma bloc de l'UT pour la multiplication d'entier signé



Exercices II

1. Etudier la multiplication séquentielle selon Robertson (nombres signés) avec "Numlab"
   - -5 x -3 (1011 x 1101); -5 x 3 (1011 x 0011); 5 x -3 (0101 x 1101)
   - -7 x -3 (1001 x 1101), -1 x -1 (1111 x 1111); 7 x 7 (0111 x 0111)
2. Calculer à la main en binaire les multiplications suivantes :
   a) 27 x 5 = 135
   b) 30 x 6 = 180
   c) 25 x -5 = -125
   d) -25 x -5 = 125
   - Vous utiliserez une représentation sur 6 bits avec signe en complément à 2, le résultat sera donné sur 12 bits (en C2).

Exercices II

3. Modifier le schéma de la multiplication spatiale de la page 24 pour réaliser une multiplication de nombres signés en C2 selon l'algorithme de Robertson.