

## CHAPITRE 5

### COMPARAISON AVEC DES *CORES* COMMERCIAUX

#### 5.1 Introduction

L'évolution rapide de la microélectronique met à la disposition des concepteurs de FPGA comme d'ASIC des puces intégrant de plus en plus de portes, permettant d'intégrer des systèmes de plus en plus complexes sur une même puce. Ainsi, en gardant un processus de conception identique à ce qui se fait à l'heure actuelle, il faudrait augmenter la taille des équipes de développement et ceci malgré l'évolution des outils et des méthodes. Toutefois, il est démontré qu'au-delà d'une dizaine d'ingénieurs travaillant sur une même puce, l'efficacité n'évolue plus de manière proportionnelle avec l'augmentati [MCours.com](http://MCours.com) le montre la figure 44, il apparaît une faille de productivité [24]. Sans évolution de la méthode de conception, l'exploitation des ressources disponibles sur une puce, en un temps de design raisonnable, devient impossible. Une méthodologie possible pour exploiter cette faille est le *design-reuse*, permettant de réutiliser dans un autre contexte des modules déjà conçus. Ceci explique la disponibilité de plusieurs noyaux (*core*) sur le marché. Ainsi, le but de ce chapitre, dans un premier temps, est de présenter les ressources utilisées et la fréquence d'opération du CIC polyphasé. Dans un deuxième temps, un aperçu sur les *cores* disponibles sur le marché sera présenté. Enfin, une comparaison des *cores* de CIC au niveau des ressources utilisées et de la fréquence d'opération sera faite.

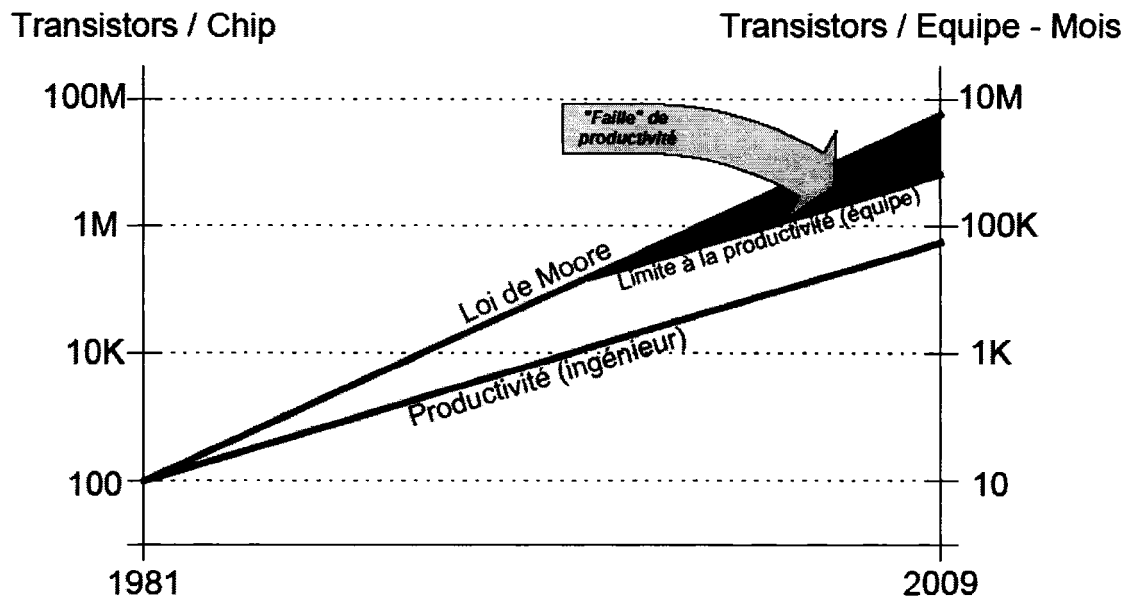


Figure 44 Faille de productivité [24]

## 5.2 Estimation des ressources requises par le CIC polyphasé

La plate-forme choisie pour tester l'implémentation est une puce programmable de la famille VirtexII Pro de Xilinx (Xilinx 2004). Ces FPGA possèdent des ressources logiques sous forme d'unités de base nommées *slices*. Chaque *slice* contient notamment deux générateurs de fonctions à quatre entrées, une chaîne de retenue, des multiplexeurs et deux registres. Chaque générateur de fonction peut être programmé soit en une table de conversion (LUT, Look Up Table) à quatre entrées, soit en 16 bits de mémoire ou soit comme un registre à décalage de 16 bits. De plus, le FPGA possède des ressources dédiées telles que les multiplicateurs ayant une largeur d'entrée de 18 bits (BMULT). Une bascule peut être activée environ au centre de la logique de multiplication lorsque l'option « pipeline » est sélectionnée. Cette option permet d'accélérer la fréquence d'horloge et introduit un cycle de latence. De la mémoire embarquée est aussi disponible sous forme de *Block RAM* (BRAM) de taille 18 kbits. Ces blocs sont configurables en

mode simple ou double ports et utilisent des mots d'entrée de 1, 2, 4, 18 ou 36 bits. Par exemple, on peut avoir une RAM 18x1024 ou 36x512.

Traditionnellement, les résultats en termes de ressources requises pour l'implémentation dans ces puces sont exposés par le nombre de *slices*, de blocs de mémoire RAM (BRAM) et de multiplicateurs dédiés (BMULT). Des processeurs PowerPC sont aussi présents dans les VirtexII Pro. Cependant, les blocs mémoire RAM, les multiplicateurs dédiés et les processeurs PowerPC ne sont pas utilisés par le CIC polyphasé.

Pour l'étape de la synthèse du code VHDL, le logiciel Synplify Pro 7.7 de Synplicity a été employé. Aucun paramètre dans l'interface usager de Synplify n'a été sélectionnée parmi les options disponibles : *pipelining*, *retiming*, *resource sharing*, *modular design* et *FSM compiler*. Ces optimisations n'occasionnent pas de modifications sur les résultats finaux, mais augmente le temps requis pour effectuer la synthèse.

Pour la plupart des scénarios de vérification, le FPGA choisi est le XC2VP20-7. Le suffixe -7 indique la performance sur le plan de la vitesse. En synthétisant avec un suffixe différent, on obtient une contrainte d'horloge plus restrictive. Lorsqu'on choisit un autre FPGA de la famille VirtexII Pro suffisamment grand pour contenir le design, on obtient les mêmes besoins en ressources. La seule contrainte imposée concerne la période de l'horloge qui est fixée à 5 ns (200Mhz) dans le cas d'une puce -7. Le rapport de synthèse nous donne les ressources sous forme du nombre de LUT, de registres, de BMULTS et de BRAM.

Finalement, les étapes de placement et de routage ont été effectuées avec l'outil de Xilinx ISE v7.1i avec les options suivantes sélectionnées :

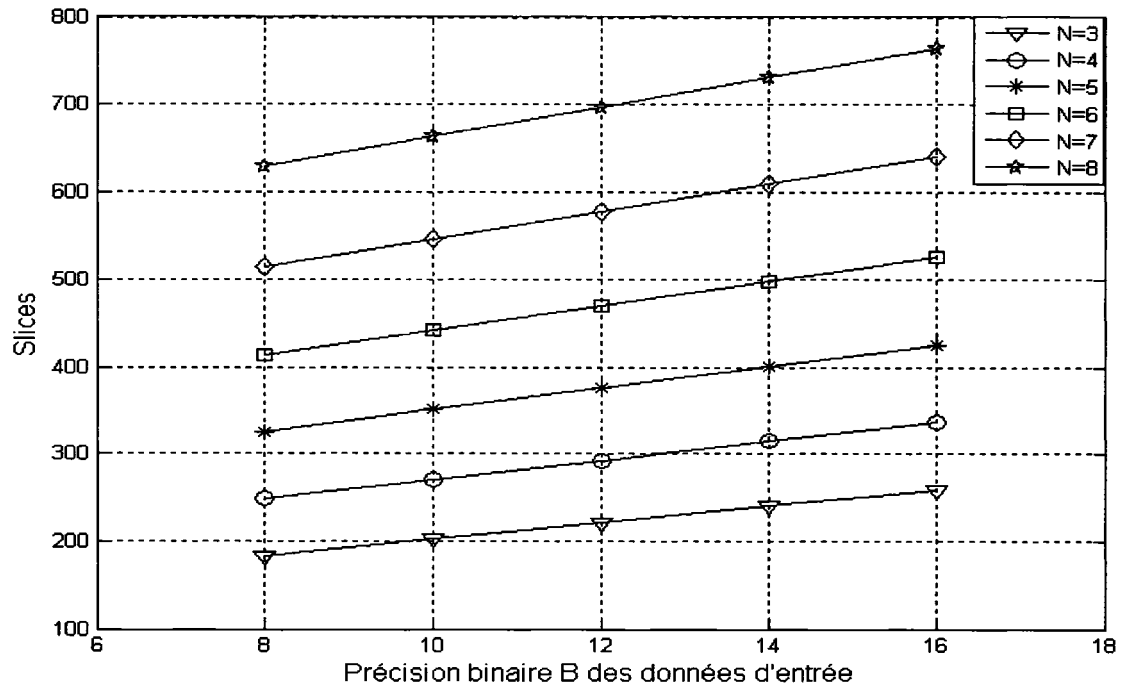
- *optimization Strategy : speed*
- *Perform Timing-Driven Packing : yes*
- *Place & Route Effort Level : high*

Tableau X présente les ressources, en terme de *slices*, requises par le *core* en fonction de l'ordre  $N$  du filtre et de la largeur binaire  $B$  des données d'entrée. Ce tableau est généré pour un filtre CIC polyphasé avec un facteur de décimation  $R = 32$ . La représentation graphique de ce tableau est présentée à la figure 45.

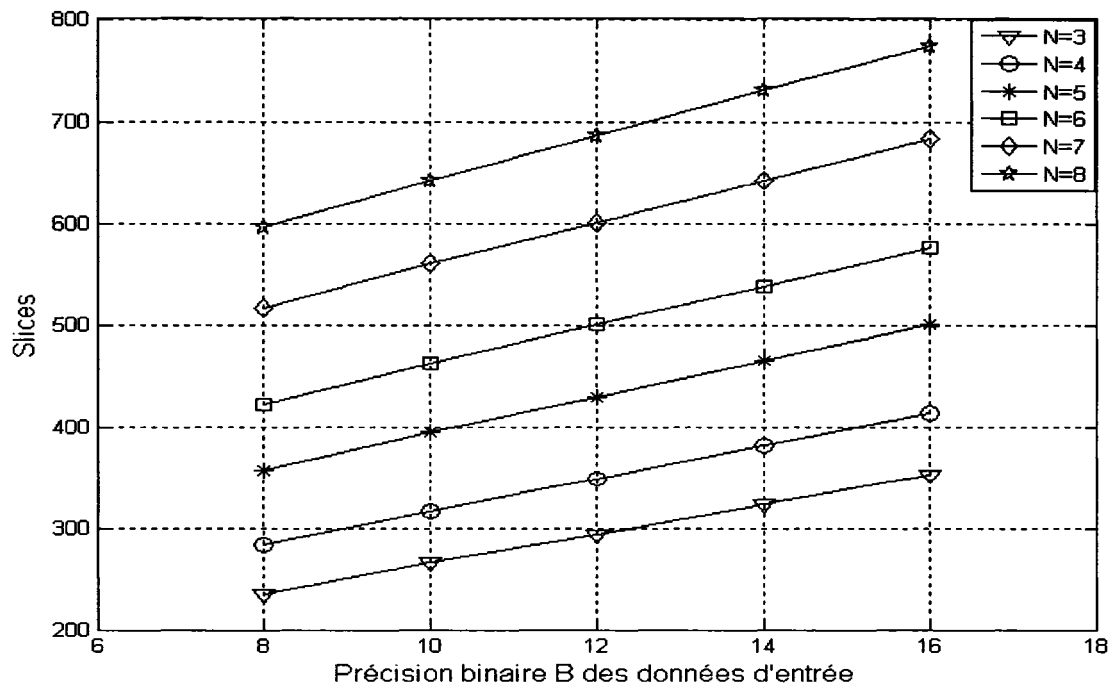
Tableau X

Nombre de *slices* du *core* en fonction de l'ordre  $N$  du filtre et de la largeur binaire  $B$  des données d'entrée;  $R = 32$ .

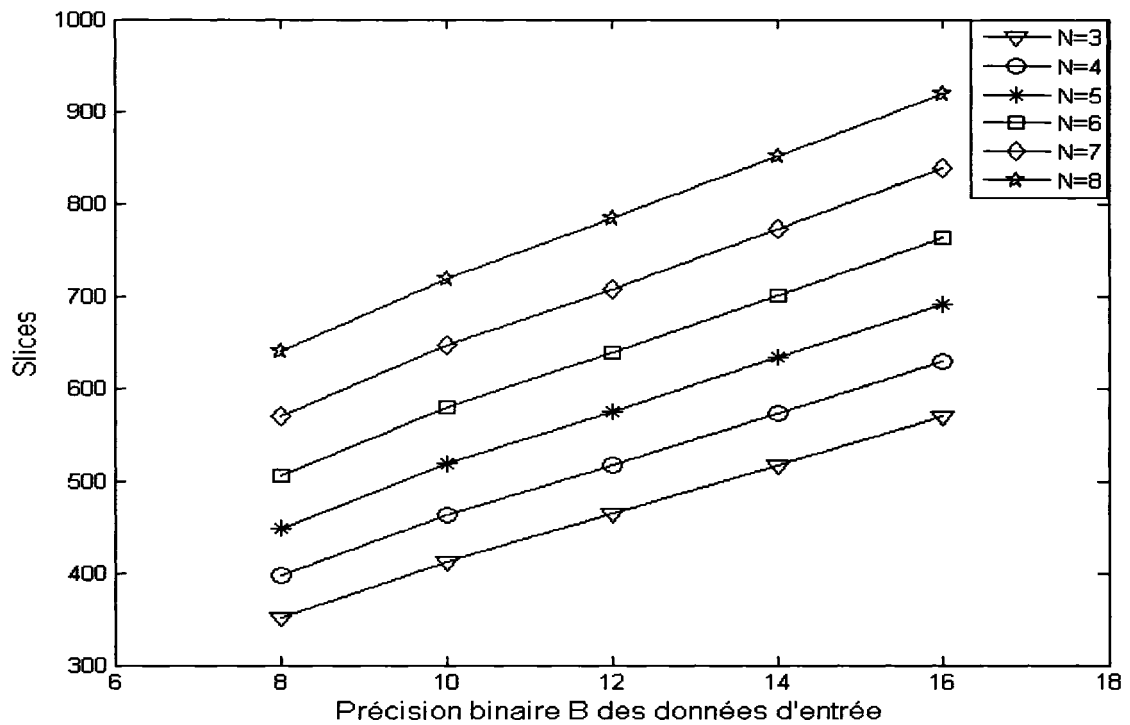
N	Nombre de <i>slices</i>					
		B = 8	B = 10	B = 12	B = 14	B = 16
3	CIC_2-phase	184	203	222	241	260
	CIC_4-phase	235	266	294	323	353
	CIC_8-phase	351	413	465	518	570
4	CIC_2-phase	249	271	293	315	337
	CIC_4-phase	283	317	349	382	414
	CIC_8-phase	397	464	518	573	629
5	CIC_2-phase	326	351	376	401	426
	CIC_4-phase	358	395	430	465	500
	CIC_8-phase	448	519	575	634	692
6	CIC_2-phase	415	443	471	499	527
	CIC_4-phase	422	462	500	538	576
	CIC_8-phase	507	580	640	701	764
7	CIC_2-phase	516	547	578	609	640
	CIC_4-phase	517	560	601	642	683
	CIC_8-phase	571	647	709	774	839
8	CIC_2-phase	629	663	697	731	765
	CIC_4-phase	596	642	686	730	774
	CIC_8-phase	641	720	785	853	920



(a) CIC\_2-phase



(b) CIC\_4-phase



(c) CIC\_8-phase

Figure 45 Nombre de slices en fonction de l'ordre  $N$  du filtre et de la précision binaire  $B$  des données d'entrée

Le nombre de *slices* occupés, lorsque le placement est effectué, est plus élevé que la moitié du plus grand nombre entre les LUT et les registres obtenus à la synthèse. Un *slice* comporte deux LUT et deux registres, mais l'outil de placement ne prend pas toutes les ressources disponibles dans chaque *slice* parce qu'il a suffisamment d'espace pour s'étendre dans la puce. On peut donc affirmer que la quantité de *slices* présentée est le pire cas de ressources requises. En réalité, moins de *slices* seraient nécessaires dans le cas de l'utilisation du design dans un FPGA plus rempli.

Les tableaux XI, XII et XIII présentent les ressources, en terme de *slices*, requises par le *core* en fonction de l'ordre  $N$  du filtre et de la largeur binaire  $B$  des données d'entrée.

Ces tables sont générées pour un filtre CIC polyphasé avec un facteur de décimation  $R = 16$ ,  $R = 48$  et  $R = 4096$  respectivement.

Tableau XI

Nombre de *slices* du *core* en fonction de l'ordre  $N$  du filtre et de la largeur binaire  $B$  des données d'entrée;  $R = 16$ .

N	Nombre de slices					
		B = 8	B = 10	B = 12	B = 14	B = 16
3	CIC_2-phase	171	190	209	228	247
	CIC_4-phase	209	240	269	298	327
	CIC_8-phase	338	402	453	507	558
4	CIC_2-phase	217	239	261	285	307
	CIC_4-phase	252	287	318	350	382
	CIC_8-phase	365	430	485	541	594
5	CIC_2-phase	288	313	340	364	390
	CIC_4-phase	301	339	373	408	443
	CIC_8-phase	412	480	537	596	652
6	CIC_2-phase	350	378	406	434	463
	CIC_4-phase	357	397	434	472	510
	CIC_8-phase	444	512	573	639	694
7	CIC_2-phase	442	474	505	536	567
	CIC_4-phase	418	461	501	542	583
	CIC_8-phase	496	570	634	703	761
8	CIC_2-phase	519	553	587	621	655
	CIC_4-phase	485	531	574	618	662
	CIC_8-phase	529	606	672	740	812

Tableau XII

Nombre de *slices* du *core* en fonction de l'ordre  $N$  du filtre et de la largeur binaire  $B$  des données d'entrée;  $R = 48$ .

N	Nombre de slices					
		B = 8	B = 10	B = 12	B = 14	B = 16
3	CIC_2-phase	199	218	237	256	275
	CIC_4-phase	247	278	307	336	365
	CIC_8-phase	363	427	478	532	583
4	CIC_2-phase	283	305	327	349	371
	CIC_4-phase	315	349	381	413	445
	CIC_8-phase	428	494	549	604	661
5	CIC_2-phase	366	391	416	441	466
	CIC_4-phase	376	413	448	483	518
	CIC_8-phase	486	557	613	673	731
6	CIC_2-phase	461	489	517	545	573
	CIC_4-phase	465	505	543	581	619
	CIC_8-phase	551	625	685	746	809
7	CIC_2-phase	593	624	655	686	717
	CIC_4-phase	566	609	650	691	732
	CIC_8-phase	647	723	786	850	914
8	CIC_2-phase	715	749	783	817	851
	CIC_4-phase	679	725	769	813	857
	CIC_8-phase	726	805	869	940	1005



Tableau XIII

Nombre de *slices* du *core* en fonction de l'ordre  $N$  du filtre et de la largeur binaire  $B$  des données d'entrée;  $R = 4096$ .

N	Nombre de slices					
		B = 8	B = 10	B = 12	B = 14	B = 16
3	CIC_2-phase	333	352	371	390	409
	CIC_4-phase	371	402	431	460	489
	CIC_8-phase	500	563	615	668	721
4	CIC_2-phase	479	501	523	545	567
	CIC_4-phase	514	548	580	612	644
	CIC_8-phase	627	694	748	804	859
5	CIC_2-phase	674	699	724	749	774
	CIC_4-phase	687	724	759	794	829
	CIC_8-phase	797	867	925	983	1040
6	CIC_2-phase	883	911	939	967	995
	CIC_4-phase	890	930	968	1006	1044
	CIC_8-phase	975	1047	1108	1170	1231
7	CIC_2-phase	1147	1178	1209	1240	1271
	CIC_4-phase	1123	1166	1207	1248	1289
	CIC_8-phase	1202	1278	1341	1406	1470
8	CIC_2-phase	1419	1453	1487	1521	1555
	CIC_4-phase	1386	1432	1476	1520	1564
	CIC_8-phase	1428	1509	1576	1644	1711

Les tableaux ci-dessus montrent bien que le traitement parallèle qu'effectue le CIC polyphasé est très efficace du point de vue ressources matérielles. Pour un peu plus de *slices*, le nombre d'entrées parallèles peut être doublé ou quadruplé. Ainsi il est

intéressant de remarquer que le nombre de *slices* nécessaires évolue plus rapidement avec une augmentation de l'ordre du filtre ou du facteur de décimation qu'avec une augmentation de la largeur binaire des données d'entrée.

### 5.3 Fréquence d'horloge

Les outils utilisés pour l'estimation des ressources requises permettent d'estimer la fréquence maximale de fonctionnement du *core*. Elle correspond à la fréquence de l'horloge unique qui dépend du taux d'entrées des échantillons dans le module. La contrainte imposée sur la période de l'horloge est de 5 ns (200 Mhz) dans le cas d'une puce -7. Aucun placement manuel n'a été effectué puisque le *core* doit être utilisable sans manipulation particulière par l'utilisateur. Par contre, de meilleures performances de fréquence d'horloge ou de ressources pourraient être obtenues en contraignant le design dans des régions. Le tableau XIII présente les performances du *core*. La fréquence de fonctionnement du *core* est  $f_{in}$ . Étant donné que ses entrées sont parallèles, alors il peut recevoir des données échantillonnées à la fréquence  $f_s$ . La fréquence  $f_s$  est obtenue en multipliant la fréquence de fonctionnement du *core*  $f_{in}$  par le nombre d'entrées parallèles, c'est-à-dire le nombre de phases.

Tableau XIV

Performance du CIC polyphasé dans un Virtex2P (-07 speed grade); N = 5, R = 48.

Performance $f_{in} / f_s$ (MHZ)					
	B = 8	B = 10	B = 12	B = 14	B = 16
CIC_2-phase	300 / <b>600</b>	289 / <b>578</b>	285 / <b>570</b>	275 / <b>550</b>	266 / <b>532</b>
CIC_4-phase	208 / <b>832</b>	208 / <b>832</b>	206 / <b>824</b>	203 / <b>812</b>	202 / <b>808</b>
CIC_8-phase	180 / <b>1440</b>	173 / <b>1384</b>	173 / <b>1384</b>	165 / <b>1320</b>	165 / <b>1320</b>

## 5.4 Aperçu des *cores* disponibles sur le marché

Il existe plusieurs compagnies sur le marché qui fournissent des *cores*. Le fabricant de FPGA Xilinx offre tout genre de *core* dans sa librairie *Logicore* accessible à l'aide de l'outil CORE Generator. Ce sont aussi ces modules qui sont instanciés dans le FPGA lorsque le *toolbox* de Simulink, Xilinx System Generator, est utilisé pour la conception. Parmi les *Logicores*, on retrouve un module de CIC reconfigurable. Cette section porte sur la performance et les ressources matérielles requises par ce *core* optimisé pour les Virtex et Virtex-II.

Les compagnies Lattice Semiconductor Corporation et Intersil proposent aussi, chacune, un *core* de CIC qui possède plusieurs paramètres configurables. En revanche, dans les sections suivantes, nous limiterons notre analyse au produit de Xilinx, puisque nous avons des données quantitatives permettant de le comparer au *core* CIC polyphasé.

### 5.4.1 Présentation des *cores* de CIC

La compagnie Intersil a présenté en juillet 2004 un *core* de filtre à décimation appelé HSP43220 [25]. Le HSP43220 est un filtre à décimation optimisé pour le filtrage des signaux à bande étroite. Il est implémenté en deux étages de décimation. La figure 46 montre le diagramme bloc du HSP43220.

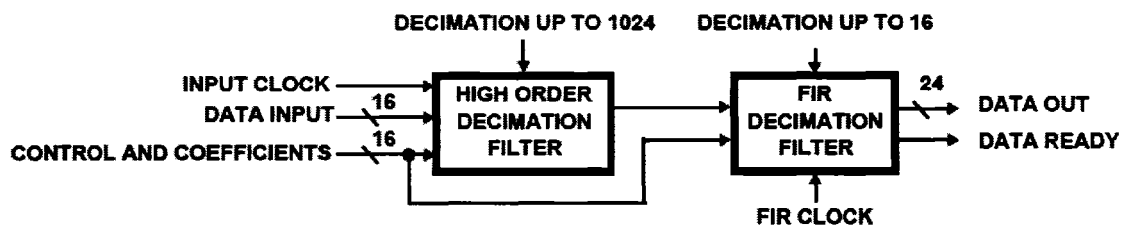
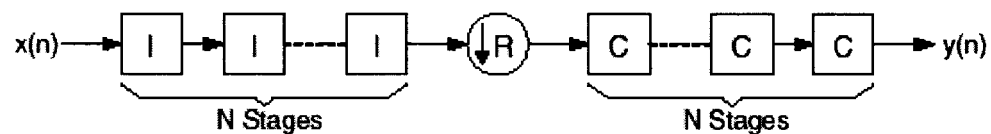


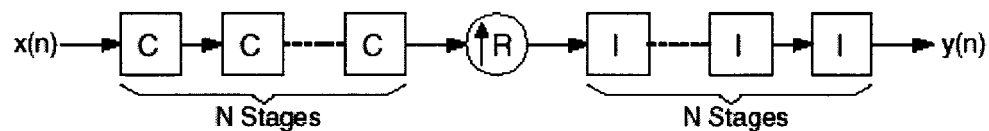
Figure 46 Diagramme bloc du HSP43220

Le premier étage de décimation est constitué d'un filtre à décimation de grand ordre (CIC d'ordre 5) qui utilise une technique de décimation efficace permettant d'atteindre un facteur de décimation maximale de 1024 et une atténuation maximale de 96 dB. Le deuxième étage de décimation est constitué d'un filtre FIR à décimation avec un nombre de coefficients symétriques maximal de 512. Le facteur de décimation maximale du filtre FIR à décimation est égal à 16. La combinaison des deux facteurs de décimation donne le facteur de décimation total du filtre qui est égal à 16384. Le HSP43220 offre la possibilité de contourner le filtre à décimation de grand ordre ou le filtre FIR à décimation pour de la flexibilité additionnelle. Cependant, le HSP43220 est très limité du point de vue de la programmabilité. La largeur binaire des données d'entrée et de sortie est fixée, respectivement, à 16 et 24 bits en complément à deux. La largeur binaire des coefficients du filtre FIR à décimation est fixée à 20 bits. Sa fréquence maximale de fonctionnement est de 33 MHz.

La Compagnie Lattice Semiconductor Corporation a présentée, en octobre 2005, un *core* de CIC [26]. Ce noyau peut être configuré pour être un CIC à décimation ou à interpolation. La figure 47 montre le CIC décimateur et interpolateur.



(a) CIC décimateur



(b) CIC interpolateur

Figure 47 CIC décimateur et interpolateur

Ce *core* est complètement paramétrable et offre les possibilités suivantes :

- Largeur binaire des données d'entrée : entre 1 et 32 bits
- Choix du type de filtre : décimation ou interpolation
- Facteur de décimation ou d'interpolation : fixe ou programmable
- Nombre d'étages (ordre du filtre) du CIC : entre 1 et 8
- Délai différentiel : entre 1 et 4
- Facteur de décimation : entre 2 et 16384
- Jusqu'à 4 canaux d'entrée pour la décimation ou l'interpolation

La largeur binaire des données de sortie de ce *core* n'est cependant pas programmable. Elle dépend de la largeur maximale de bus qui permet aux débordements à l'intérieur du circuit de ne pas affecter les résultats à la sortie du filtre. La fréquence maximale de fonctionnement de ce *core* est de l'ordre de 200 MHz.

#### **5.4.2 Présentation du Logicore de CIC et des ressources utilisées**

La compagnie Xilinx a présenté, en mars 2002, un *core* de CIC complètement paramétrable et qui offre les possibilités suivantes :

- Largeur binaire des données d'entrée : entre 1 et 32 bits
- Choix du type de filtre : décimation ou interpolation
- Facteur de décimation ou d'interpolation : fixe ou programmable
- Nombre d'étages (ordre du filtre) du CIC : entre 1 et 8
- Délai différentiel : entre 1 et 2
- Facteur de décimation varie entre 8 et 16383
- Jusqu'à 2 canaux d'entrée pour la décimation seulement

Pour assurer une grande fréquence de fonctionnement, le CIC décimateur est implémenté en utilisant une architecture pipeline présentée à la figure 48 [27]. Les registres de pipeline P0, P1, P2 et P3 permettent de couper la longue chaîne d'additionneurs causée par les différentiateurs en cascade. Ceci permet d'avoir un seul additionneur entre deux registres. Donc le chemin critique de cette architecture est constitué d'un additionneur. Cette technique permet d'augmenter la fréquence de fonctionnement du filtre.

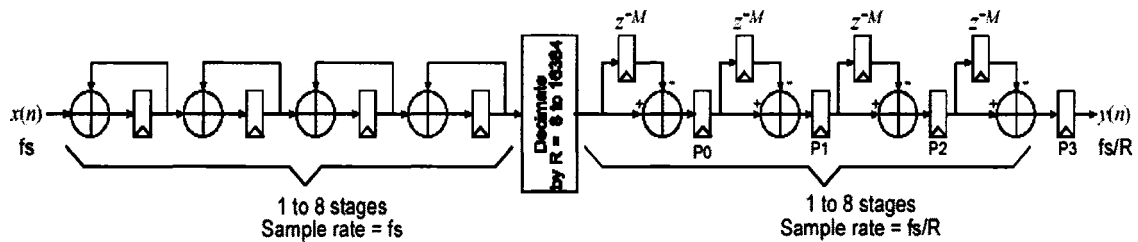


Figure 48 Architecture pipeline du CIC décimateur de Xilinx [27]

En échangeant le bloc d'intégrateurs avec celui de différentiateurs, comme présenté à la figure 49, le *core* devient un CIC interpolateur.

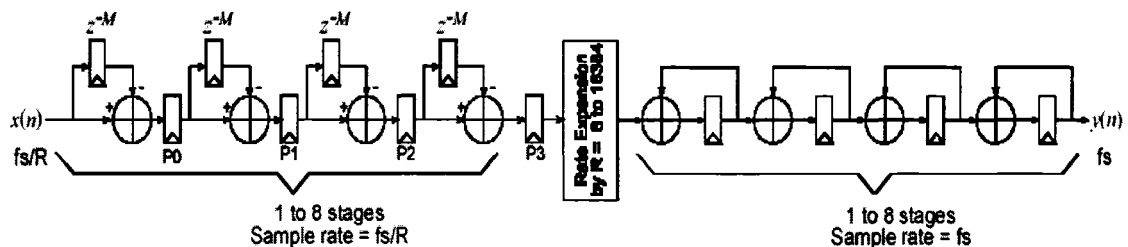


Figure 49 Architecture pipeline du CIC interpolateur de Xilinx [27]

La largeur binaire des données de sortie de ce *core* n'est pas programmable non plus. Elle dépend de la largeur maximale de bus qui permet aux débordements à l'intérieur du

circuit de ne pas affecter les résultats à la sortie du filtre. Dans le cas d'une configuration à deux canaux, c'est-à-dire deux entrées différentes, la plupart des ressources matérielles utilisées est multiplexée dans le temps. À cause de ce multiplexage temporel, la fréquence de chaque donnée d'entrée est réduite d'un facteur de 2 pour le cas de deux canaux. Ceci veut dire que si la fréquence maximale du CIC décimateur est de 100 MHz, une décimation avec un seul canal peut supporter des données qui arrivent à 100 MHz. Dans le cas d'une décimation avec deux canaux, le filtre supporte deux données qui arrivent chacune à 50 MHz.

Les tableaux XV, XVI, XVII et XVIII présentent, pour les mêmes configurations, les ressources utilisées par le *Logicore* de CIC et celles utilisées par le CIC polyphasé.

Tableau XV

Comparaison du *Logicore* de CIC versus le CIC polyphasé pour  $R = 16$

N	Nombre de slices					
		B = 8		B = 12		B = 16
3	Xilinx_CIC	100		118		136
	CIC_2-phase	171		209		247
	CIC_4-phase	209		269		327
	CIC_8-phase	338		453		558
4	Xilinx_CIC	156		180		204
	CIC_2-phase	217		261		307
	CIC_4-phase	252		318		382
	CIC_8-phase	365		485		594
5	Xilinx_CIC	224		254		284
	CIC_2-phase	288		340		390
	CIC_4-phase	301		373		443
	CIC_8-phase	412		537		652

Tableau XV (suite)

N	Nombre de slices					
		B = 8		B = 12		B = 16
6	Xilinx_CIC	304		340		376
	CIC_2-phase	350		406		463
	CIC_4-phase	357		434		510
	CIC_8-phase	444		573		694
7	Xilinx_CIC	396		438		480
	CIC_2-phase	442		505		567
	CIC_4-phase	418		501		583
	CIC_8-phase	496		634		761
8	Xilinx_CIC	500		548		596
	CIC_2-phase	519		587		655
	CIC_4-phase	485		574		662
	CIC_8-phase	529		672		812

Tableau XVI

Comparaison du *Logicore* de CIC versus le CIC polyphasé pour R = 32

N	Nombre de slices					
		B = 8		B = 12		B = 16
3	Xilinx_CIC	111		129		147
	CIC_2-phase	184		222		260
	CIC_4-phase	235		294		353
	CIC_8-phase	351		465		570



Tableau XVI (suite)

N	Nombre de slices					
		B = 8		B = 12		B = 16
4	Xilinx_CIC	182		206		230
	CIC_2-phase	249		293		337
	CIC_4-phase	283		349		414
	CIC_8-phase	397		518		629
5	Xilinx_CIC	256		286		316
	CIC_2-phase	326		376		426
	CIC_4-phase	358		430		500
	CIC_8-phase	448		575		692
6	Xilinx_CIC	360		396		432
	CIC_2-phase	415		471		527
	CIC_4-phase	422		500		576
	CIC_8-phase	507		640		764
7	Xilinx_CIC	461		503		545
	CIC_2-phase	516		578		640
	CIC_4-phase	517		601		683
	CIC_8-phase	571		709		839
8	Xilinx_CIC	598		646		694
	CIC_2-phase	629		697		765
	CIC_4-phase	596		686		774
	CIC_8-phase	641		785		920

Tableau XVII

Comparaison du *Logicore* de CIC versus le CIC polyphasé pour R = 48

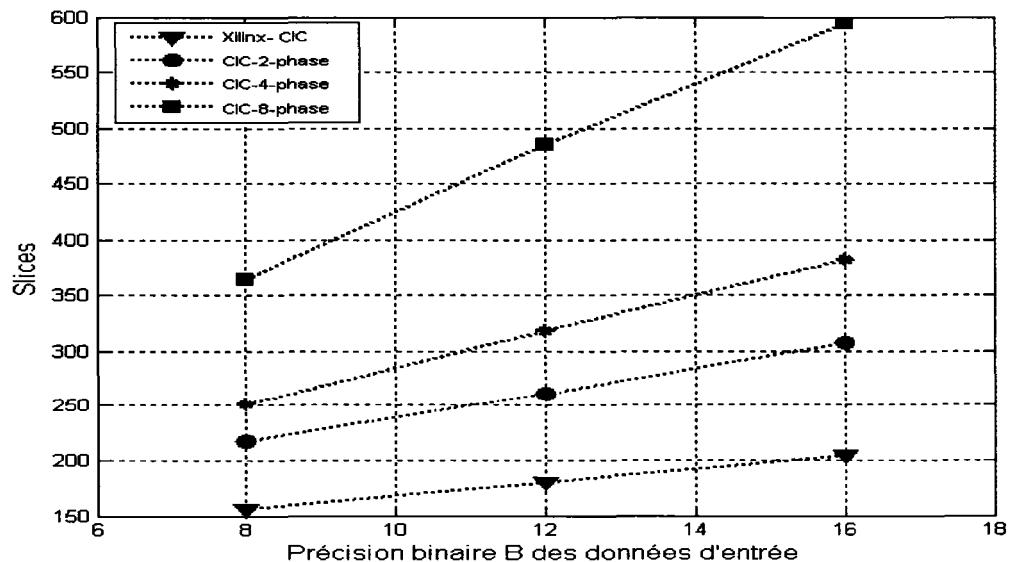
N	Nombre de slices					
		B = 8		B = 12		B = 16
3	Xilinx_CIC	120		138		156
	CIC_2-phase	199		237		275
	CIC_4-phase	247		307		365
	CIC_8-phase	363		478		583
4	Xilinx_CIC	194		218		242
	CIC_2-phase	283		327		371
	CIC_4-phase	315		381		445
	CIC_8-phase	428		549		661
5	Xilinx_CIC	286		316		346
	CIC_2-phase	366		416		466
	CIC_4-phase	376		448		518
	CIC_8-phase	486		613		731
6	Xilinx_CIC	396		432		468
	CIC_2-phase	461		517		573
	CIC_4-phase	465		543		619
	CIC_8-phase	551		685		809
7	Xilinx_CIC	524		566		608
	CIC_2-phase	593		655		717
	CIC_4-phase	566		650		732
	CIC_8-phase	647		786		914
8	Xilinx_CIC	646		694		742
	CIC_2-phase	715		783		851
	CIC_4-phase	679		769		857
	CIC_8-phase	726		869		1005

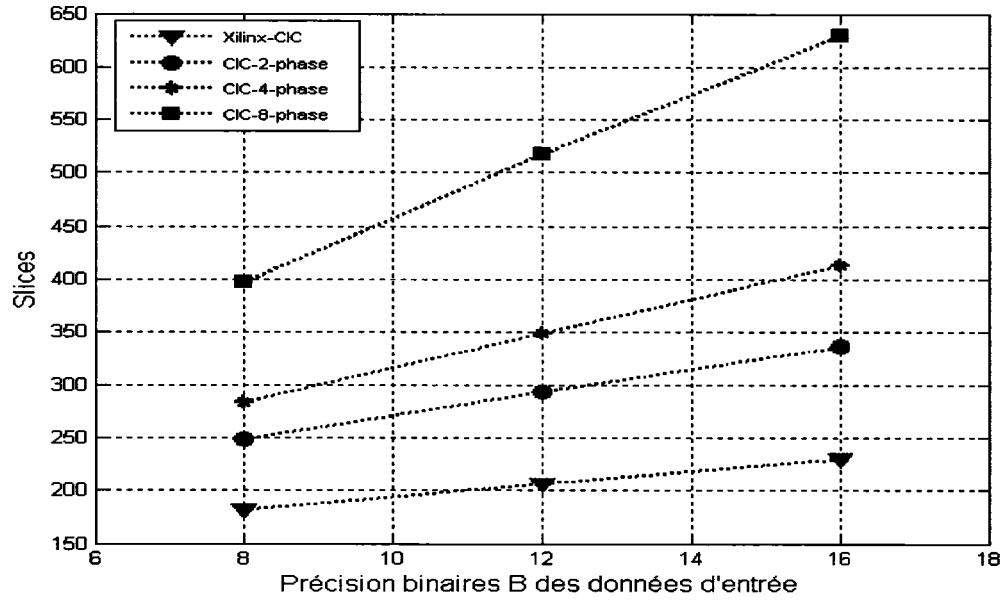
Tableau XVIII

Comparaison du *Logicore* de CIC versus le CIC polyphasé pour R = 4096

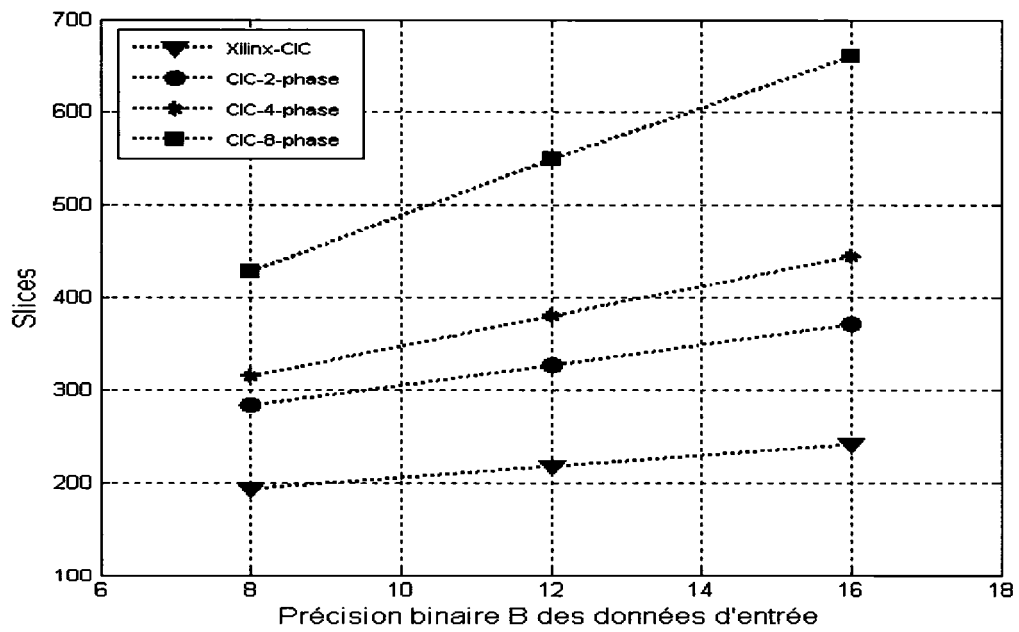
N	Nombre de slices					
		B = 8		B = 12		B = 16
3	Xilinx_CIC	214		232		250
	CIC_2-phase	333		371		409
	CIC_4-phase	371		431		489
	CIC_8-phase	500		615		721
4	Xilinx_CIC	354		378		402
	CIC_2-phase	479		523		567
	CIC_4-phase	514		580		644
	CIC_8-phase	627		748		859
5	Xilinx_CIC	890		940		990
	CIC_2-phase	674		724		774
	CIC_4-phase	687		759		829
	CIC_8-phase	797		925		1040
6	Xilinx_CIC	1246		1306		1366
	CIC_2-phase	883		939		995
	CIC_4-phase	890		968		1044
	CIC_8-phase	975		1108		1231
7	Xilinx_CIC	1662		1732		1802
	CIC_2-phase	1147		1209		1271
	CIC_4-phase	1123		1207		1289
	CIC_8-phase	1202		1341		1470
8	Xilinx_CIC	2138		2218		2298
	CIC_2-phase	1419		1487		1555
	CIC_4-phase	1386		1476		1564
	CIC_8-phase	1428		1576		1711

Les tableaux ci-dessus résument les différentes ressources utilisées par le *Logicore* et les comparent à des configurations semblables du CIC polyphasé. Les ressources présentées sont en fonction de l'ordre du filtre et de la précision binaire des données d'entrée. Il est important de remarquer que le CIC polyphasé nécessite toujours un peu plus de ressources que le CIC de Xilinx sauf dans le cas où l'ordre du filtre et le facteur de décimation sont grands. Ceci est bien illustré par le tableau 9 pour  $N$  égale à 6, 7 ou 8. Ces résultats sont expliqués par le gain en largeur binaire du CIC polyphasé par rapport au CIC de Xilinx. En fait, comme expliqué à la section 3.5.2, la largeur maximale des bus du CIC décimateur polyphasé est plus courte que celle des bus du CIC régulier de  $(\lceil N_2 \log_2 R_1 \rceil - \lfloor N_1 \log_2 R_1 \rfloor)$  bits. À titre d'exemple, si  $N_1 = 2$  et  $N_2 = 6$  alors la largeur binaire des bus du CIC\_2-phase, CIC\_4-phase et CIC\_8-phase est plus courte que celle des bus du CIC régulier de 4, 8 et 12 bits respectivement. Ceci est dû, principalement, à la décimation multi-étage. Cette réduction de la largeur des registres est très intéressante car diminue les ressources requises par le CIC polyphasé. Pour mieux voir les ressources utilisées par ces deux noyaux programmables, les figures 50 et 51 présentent, pour quelques paramètres sélectionnés les résultats sous forme graphique.

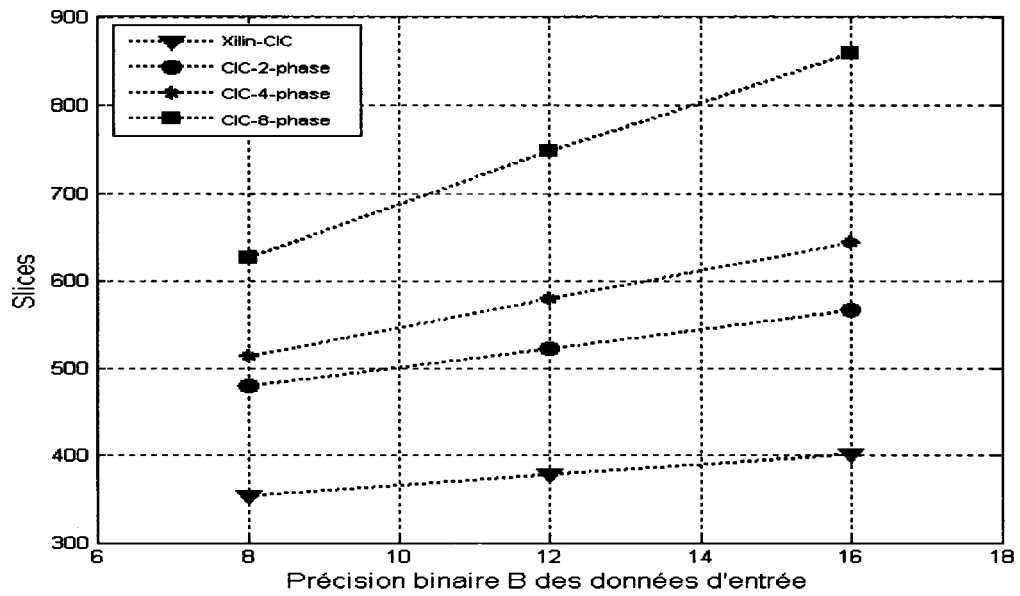
(a)  $R = 16$



(b) R = 32

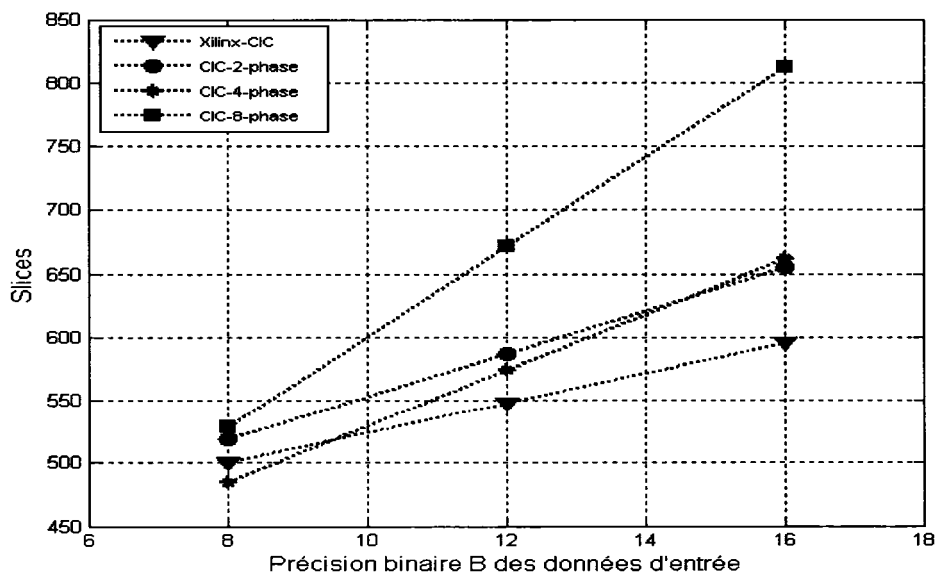


(c) R = 48

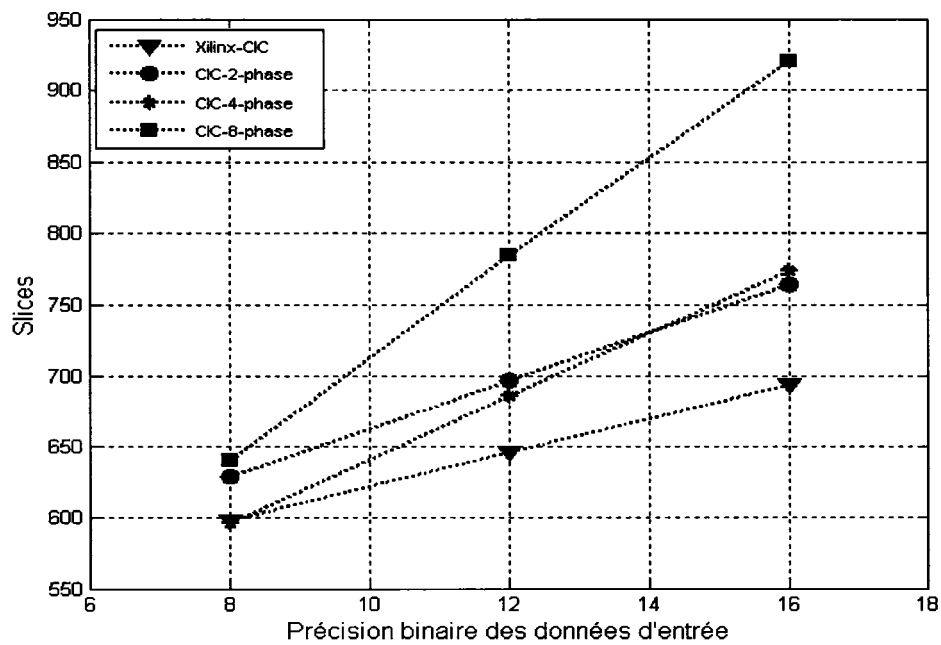


(d) R = 4096

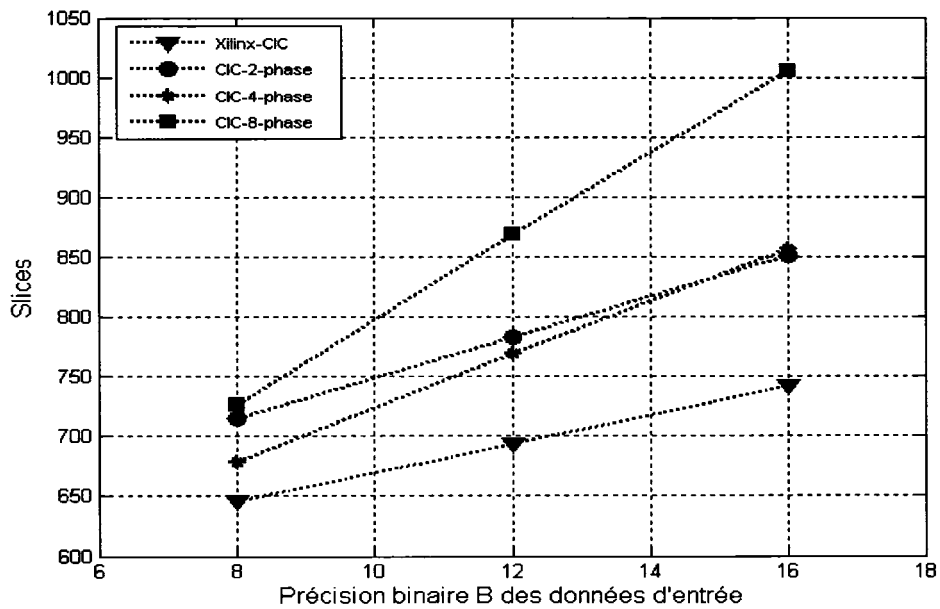
Figure 50 Comparaison des ressources FPGA requises par le CIC polyphasé et par le *Logicore* de CIC de Xilinx pour différentes précisions binaires des données d'entrée avec  $N = 4$



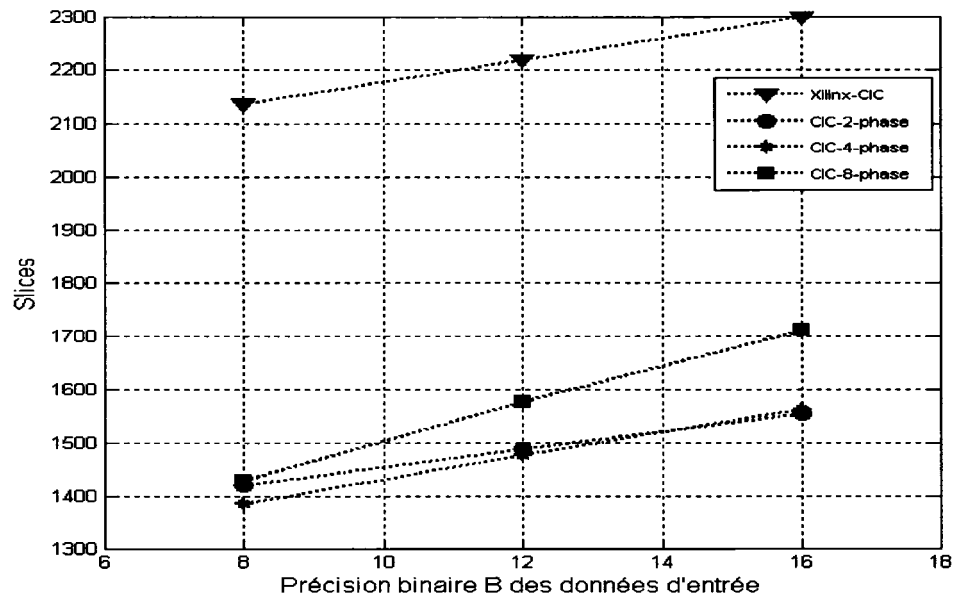
(a) R = 16



(b) R = 32



(c) R = 48



(d) R = 4096

Figure 51 Comparaison des ressources FPGA requises par le CIC polyphasé et par le *Logicore* de CIC de Xilinx pour différentes précisions binaires des données d'entrée avec N = 8

### 5.4.3 Comparaison au niveau de la fréquence de fonctionnement

La comparaison des deux noyaux, au niveau de la fréquence de fonctionnement, est présentée par le tableau XIX. Ainsi, nous pouvons remarquer que le CIC polyphasé peut recevoir des données échantillonnées à des fréquences 2, 4 ou 8 fois plus grandes que celles du CIC de Xilinx pour juste un peu plus de *slices*. Par exemple, pour N = 4, R = 48 et B = 16, le CIC de Xilinx peut recevoir des données échantillonnées à la fréquence  $f_s = 191$  MHz pour 242 *slices* alors que le CIC\_4-phase peut recevoir des données à la fréquence  $f_s = 936$  MHz pour 445 *slices*. Donc, le CIC\_4-phase peut recevoir des données échantillonnées à une fréquence 4 fois plus grande que celle du CIC de Xilinx pour moins que le double des ressources requises par ce dernier.



Tableau XIX

Comparaison entre le CIC de Xilinx et le CIC\_polyphasé au niveau de la fréquence de fonctionnement pour  $N = 4$  et  $R = 48$

Device	Performance $f_{in} / f_s$ (MHZ)			
		B = 8	B = 12	B = 16
Virtex-E -08 speed grade	Xilinx_CIC	162/162	162/162	162/162
	CIC_2-phase	166/332	165/330	155/310
	CIC_4-phase	174/696	166/664	161/644
	CIC_8-phase	165/1320	146/1168	145/1160
Virtex-II -06 speed grade	Xilinx_CIC	208/208	203/203	191/191
	CIC_2-phase	259/518	248/496	234/468
	CIC_4-phase	260/1040	248/992	234/936
	CIC_8-phase	208/1664	194/1552	188/1504

## 5.5 Conclusion

L'estimation des ressources, la fréquence de l'horloge et une comparaison du CIC polyphasé avec les noyaux commerciaux ont été l'objet de ce chapitre. Ainsi du point de vue ressources, le CIC polyphasé présente des résultats très intéressants. Par exemple, le CIC\_8-phase d'ordre  $N = 8$ ,  $R = 4096$  et  $B = 16$  bits occupe 4% des ressources d'un VirtexII Pro (xc2vp100). Du point de vue de la fréquence de fonctionnement, le CIC polyphasé est capable de supporter des données échantillonnées à des fréquences de l'ordre des GHz. Ceci veut dire que le CIC polyphasé est capable de filtrer des données numériser directement en IF ou RF. Enfin, la comparaison avec les noyaux commerciaux a montré que le CIC polyphasé tire bien son épingle du jeu par rapport au compromis fréquence de fonctionnement / ressources requises. Pour  $R = 48$ ,  $N = 5$  et  $B = 8$ , le CIC polyphasé peut recevoir des données échantillonnées à une fréquence 8 fois

plus grande que celle du CIC de Xilinx pour moins que le double des ressources de ce dernier.

[MCours.com](http://MCours.com)