

Comprendre comment une nouvelle architecture FPGA peut améliorer les performances de débit

Par Mike Hutton, architecte, conception de circuits intégrés et ingénierie, Altera

Afin de suivre le rythme très soutenu de la demande pour des composants toujours plus performants, la conception de nouveaux circuits intégrés doit garder plusieurs longueurs d'avance pour avoir à disposition des produits répondant parfaitement aux exigences du marché. Bien que tous les secteurs de la conception des semi-conducteurs subissent ces fortes attentes, il n'y en a aucun où la pression soit aussi grande que dans le domaine du traitement à haute performance. En effet, il impose de fournir une bande passante réseau et des débits qui augmentent de façon exponentielle. Dans ce secteur, de nombreux systèmes utilisent des FPGA pour délivrer les performances attendues. Cependant, même avec des architectures avancées, les concepteurs ne peuvent trouver de solutions qu'en implémentant sur la puce des bus extrêmement larges, cette largeur atteignant souvent jusqu'à 2048 bits.

Les architectures de routage peuvent gagner en efficacité en utilisant des techniques de hiérarchisation et d'optimisation. Toutefois, en doublant le nombre de pistes, on accroît également la surface des puces ce qui augmente la dissipation d'énergie. La méthodologie traditionnelle basée sur la formation de paires registre-LUT (*Look-Up Table*), technique utilisée dans toutes les actuelles architectures FPGA, signifie qu'une partie de la logique est sacrifiée aux dépens de ces registres de pipeline. L'utilisation de pipelines dans les architectures classiques se traduit également par des pénalités en temps puisqu'un signal doit être acheminé dans et hors d'un bloc logique. Résultat : une efficacité décroissante de la technique de pipeline, en particulier lorsque les retards dus au routage dominent le retard total.

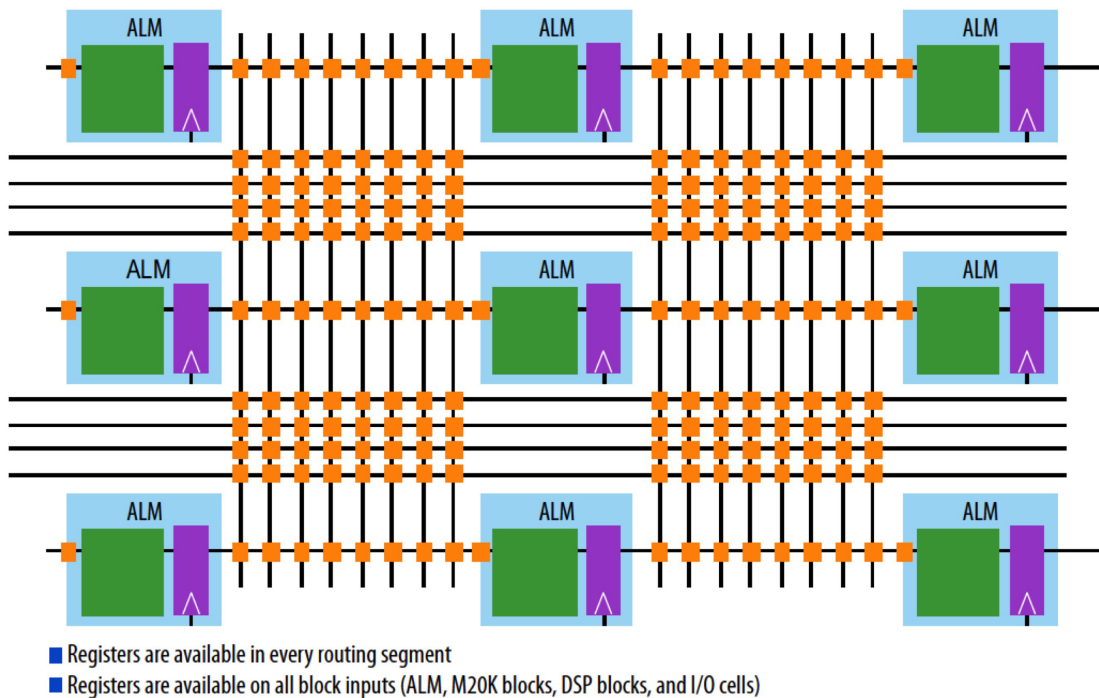
Un autre défi consiste à minimiser la variation et le décalage de l'horloge car ces caractéristiques deviennent de plus en plus critiques lorsque les vitesses augmentent. Pour accéder à des fréquences supérieures à 500 MHz, il faut carrément opter pour une nouvelle solution de synchronisation plutôt que d'ajouter des arbres d'horloge équilibrés.

Récemment, Altera a lancé une nouvelle architecture appelée HyperFlex pour leurs FPGA qui utilise plusieurs nouvelles fonctionnalités permettant ensemble de doubler la performance du cœur logique. Pour profiter de cette amélioration des performances, les développeurs ont besoin d'utiliser un certain nombre de techniques qui doivent déjà leur être familières, comme la réévaluation de timing (*register retiming*), la mise en pipeline grâce aux registres (*register pipelining*), ainsi que d'autres techniques d'optimisation. En effet, c'est la combinaison de ces techniques et des améliorations architecturales qui compte pour accroître les performances. Le changement essentiel apporté à l'architecture est celui consistant à incorporer des registres (*Hyper-Registers*) à tous les segments de routage d'interconnexions ainsi qu'aux entrées de chaque bloc fonctionnel. L'utilisation de ces registres permet aux développeurs de réévaluer le timing des chemins critiques et d'utiliser des registres en pipeline pour éliminer les retards liés au routage. Cette approche, avec ces registres de « contournement », sépare complètement les registres fonctionnels de la structure logique ALM (*Adaptive Logic Module*) des nouveaux registres de type Hyper-Registers ajoutés.

Tous les segments de routage disposent d'un Hyper-Registre optionnel intégré dans le multiplexeur programmable de routage, ce qui permet au segment de routage d'être de type « registrable » ou « logique combinatoire ». Ces Hyper-Registers sont disponibles partout au cœur de la matrice comme le montre la figure 1. Les Hyper-Registers sont représentés par les petits carrés à l'intersection de chaque segment de routage horizontal ou vertical.

Figure 1 : L'approche «des registres partout» de l'architecture *HyperFlex*

Grâce à cette architecture, il n'y a pas besoin d'utiliser un ALM pour trouver un registre de pipeline. Chaque ligne d'interconnexion horizontale et verticale dans le composant contient un Hyper-Registre qui peut être activé ou désactivé en configurant le FPGA. Les Hyper-Registers sont des simples registres de passage, à une seule entrée et une seule sortie, sans multiplexeur de routage à l'entrée. Vous contrôlez ces registres via les bits de configuration. Ils sont peu coûteux et n'occupent qu'une minuscule surface de silicium sur le composant. Comme ces Hyper-Registers sont omniprésents au sein de la matrice, les concepteurs ne sont pas limités par le nombre de registres dans leur conception. Ils peuvent réévaluer le timing des chemins critiques et, si besoin, utiliser des registres en pipeline sans consommer des ressources LAB (*Logic Array Block*) supplémentaires. Dans de nombreux cas, la conception utilise moins de ressources LAB parce les registres sont implémentés en utilisant des Hyper-Registers pour le routage au lieu de mobiliser partiellement un ALM dans le simple but d'utiliser son registre.



Dans les architectures classiques, le logiciel effectue le recalage du timing en trouvant à proximité un registre ALM inutilisé et en l'incluant dans le circuit. Cette méthode de « retiming » est limitée par la granularité du placement du registre d'ALM et l'ALM inutilisé peut ne pas être idéalement situé, provoquant un retard supplémentaire simplement pour l'inclure dans le design. De plus, il y a un délai intrinsèque dû à l'acheminement au travers de l'ALM jusqu'au registre.

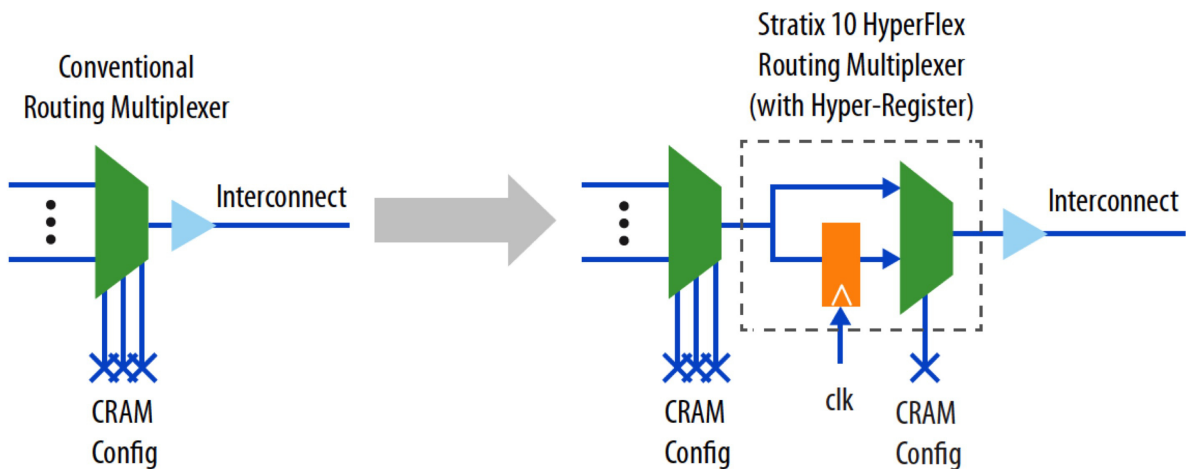


Figure 2 : Exemple de routage avant et après « retiming » avec les architectures classiques.

Avec HyperFlex et la réévaluation de timing (*register retiming*) par les registres, appelé *Hyper-Retiming*, la granularité est extrêmement fine; il n'y a plus à prendre en compte que le retard d'une piste individuelle de routage, qui n'est que de quelques dizaines de picosecondes. Les compromis faits en essayant de localiser les registres de « retiming » dans les architectures classiques sont évités et les chemins correspondant à des retards de quelques nanosecondes sont parfaitement réalisables durant l'*Hyper-Retiming* comme le montre la figure 3.

L'*Hyper-Retiming* n'a pas d'incidence sur les LAB et les ALM existants, ce qui signifie qu'aucun placement/routage incrémental n'est requis et qu'il n'y a aucun impact significatif sur le temps de compilation. Pour faire appel à un registre de « retiming », l'emplacement du registre est tout simplement inséré dans le routage à son emplacement final naturellement équilibré (voir la figure 3) après le placement/routage. Cette fonctionnalité est un immense avantage pour les conceptions impliquant de larges bus de données. En effet, dans les architectures classiques, des centaines ou des milliers d'ALM supplémentaires sont forcément mobilisées pour réaliser le « retiming » et implique généralement un nouveau routage complet.

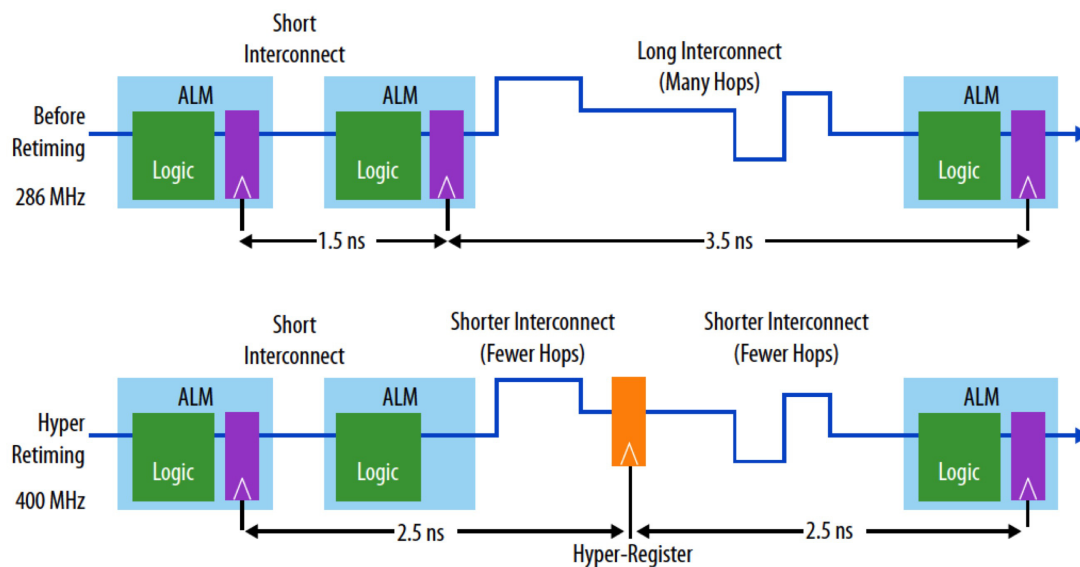


Figure 3 : *Hyper-Retiming* dans l'architecture *HyperFlex*

Le pipelining conventionnel souffre des mêmes inconvénients que le « retiming » conventionnel, et la faible granularité des registres réduit l'efficacité de cette optimisation. C'est aussi intrinsèquement un processus itératif parce que le nombre d'étages de pipeline requis, et leur localisation optimale, est inconnu au début du processus. Par conséquent, le placement et le routage doivent être réitérés plusieurs fois lors de la conception en essayant de converger vers une solution de pipeline qui réponde aux objectifs de performance. La figure 4 montre un exemple simple avant et après le pipelining classique.

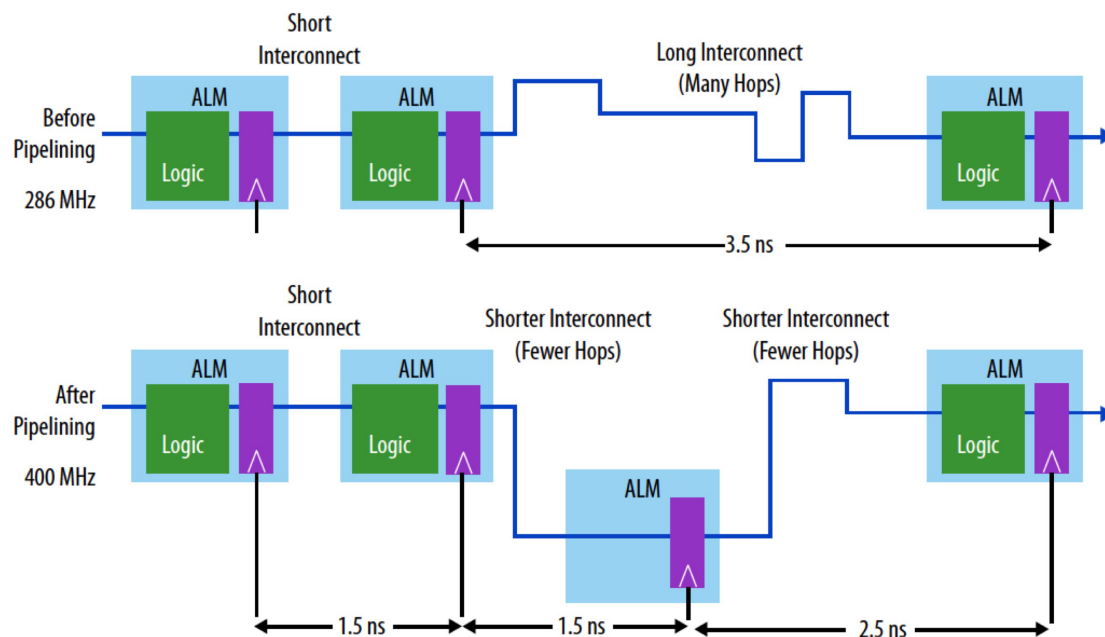


Figure 4 : Mise en pipeline dans une architecture de FPGA conventionnelle

Lorsque vous utilisez l'architecture HyperFlex, vous pouvez utiliser à volonté des pipelines grâce aux Hyper-Registres et ceci sans augmenter la taille du design. Ce processus est dénommé *Hyper-Pipelining*. Dans de nombreux cas, une conception faisant un usage intensif des registres voit diminuer le nombre d'ALM requis pour l'implémentation, car aucun registre « orphelin » n'est nécessaire. Avec ce qui équivaut à du pipelining gratuit, vous pouvez utiliser sans modération cette technique, particulièrement pour la logique des chemins de données et du « *feed-forward* ». La figure 5 montre un exemple de *Hyper-Pipelining*.

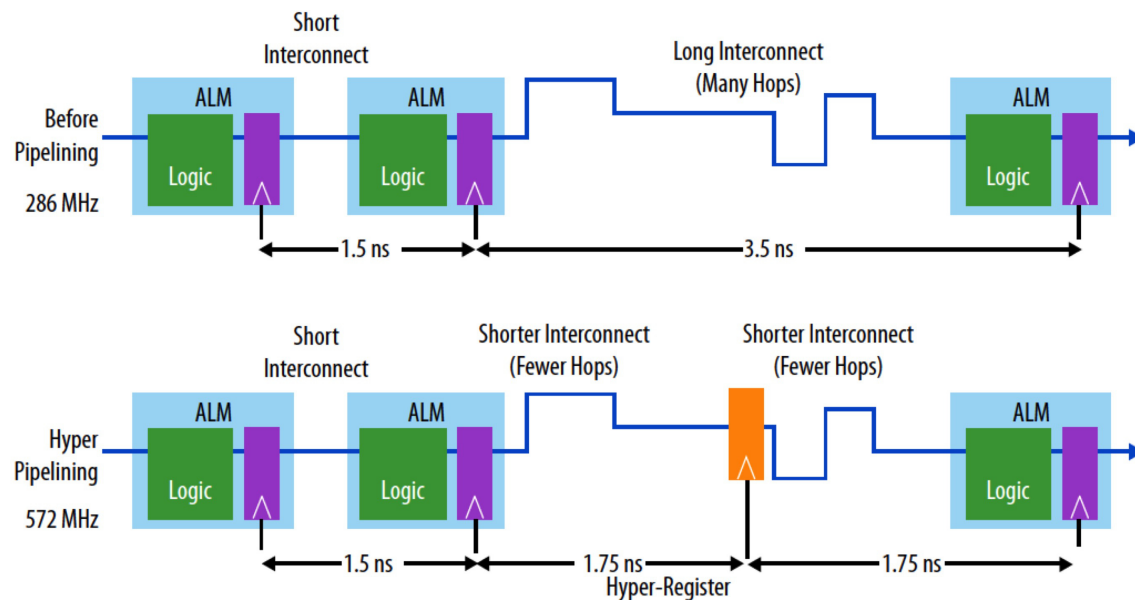


Figure 5 : *Hyper-Pipelining* dans l'architecture *HyperFlex*

Comme le logiciel de développement des FPGA, en l'occurrence le logiciel Quartus II dans le cas d'Altera, peut automatiquement réévaluer les chemins critiques de la logique en déplaçant les registres au sein de l'interconnexion, il vous suffit de spécifier le nombre de registres de pipeline requis à l'entrée du système d'horloge ou à la broche logique d'un sous-système. Ensuite le logiciel déplace les registres comme requis dans le routage, après placement/routage, résolvant ainsi les multiples problèmes d'itération qui existent avec le pipelining dans une architecture conventionnelle. Placer les registres ensemble dans le RTL (*Register Transfer Level*) facilite aussi le paramétrage de la logique lorsque les bibliothèques d'IP (*Intellectual Property*) ciblent plus d'une fréquence d'horloge (fMAX).

Les concepteurs qui favorisent les pipelines dans leurs réalisations apprécieront d'autant plus l'architecture HyperFlex. Par exemple, les types de contrôle de flux tolérant les latences, comme l'utilisation de signaux « données valides » au lieu des « enables » d'horloge à fortes impédances de sortie (*high-fanout clock enables*) permettent au logiciel de déplacer facilement les registres dans toute la matrice du FPGA.

Après avoir accompli les étapes d'*Hyper-Retiming* et d'*Hyper-Pipelining*, il est possible que les gains en performances dans certains domaines de la conception mettent en évidence d'autres goulets d'étranglement empêchant de progresser plus en performance. Un circuit comportant une longue boucle de rétroaction ou une complexe machine d'état fini pourraient être des causes possibles. Par conséquent, il serait prudent d'optimiser ces parties spécifiques de la conception. Typiquement, de telles « Hyper-Optimisations » ciblent les boucles de contrôle pour lesquelles le gain de performances l'emporte largement sur l'éventuelle logique supplémentaire requise.

Répondre aux exigences de la prochaine génération de conceptions à haute performance se révèle être un défi avec des architectures de FPGA conventionnelles. Grâce à l'approche « des registres partout », cette nouvelle architecture HyperFlex promet de satisfaire pleinement les niveaux de hautes performances exigés par les bandes passantes toujours plus élevées des applications.

+++FIN