3. INTEGRATION V.L.S.I.

3.1. Processeur d'estimation de mouvement.

Toutes les solutions de types microprogrammées ont été écarté, au vu de l'estimation de la charge de calcul à mettre en oeuvre. Le choix s'est donc porté sur un circuit spécifique de la fonction à réaliser, autorisant une architecture aussi proche que possible du haut degré de parallèlisme permis avec ce type d'algorithme. Cette partie est consacrée à l'étude d'un tel circuit.

Tout algorithme de block-matching nécessite le calcul de la distorsion entre deux blocs de pixels; cette fonction est réalisée par un opérateur de matching, que nous nommons aussi bloc de manhattan (BM), de par le fait qu'il réalise le calcul d'une distance d'ordre 1 (du point de vue de la luminance). Cet opérateur est réalisé en logique TSPC (True Single Phase Clock).

Par ailleurs l'algorithme d'estimation de mouvement étant caractérisé par deux étapes successives, l'architecture choisie sera séparée en deux parties. 9 blocs de Manhattan sont associés aux 9 points de matching de la première étape de l'algorithme. La seconde partie est constituée de seulement 8 opérateurs. Ces deux ensembles fonctionnent en pipeline, figure 3.1 .

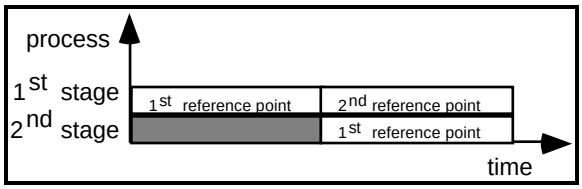


Figure 3.1 .

Lorsque le calcul des distorsions est terminé à la première étape, le meilleur matching est choisi à l'aide d'un détecteur de distorsion minimum. Ce résultat est d'une part communiqué à l'extérieur comme étant la première partie du vecteur de mouvement, mais il est aussi utilisé dans le calcul de la seconde étape.

L'un des problèmes majeur dans l'intégration de ce type d'algorithme reste le nombre considérable de pixels à traiter. L'accés aux mémoires externes étant plus lent que le potentiel de calcul à l'intérieur du circuit. En considérant un temps d'accés pour des RAM statiques de 15 à 20 ns, et un temps de cycle pour un additionneur pipeline de l'ordre de 5 ns nous voyons que les deux entités ont un rapport de vitesse de l'ordre de 4. Il est donc nécessaire de réduire au maximum le nombre d'accés à la mémoire externe, pour cela nous avons choisi :

- De parallèliser l'acquisition des pixels. En tenant compte du fait que les dimensions des fenêtres de matching et de recherche sont paires; une adresse dans la mémoire externe adresse en fait 4 pixels.

- D'éviter les acquisitions multiples. En effet la majorité des pixels utilisés au premier étage sont potentiellement utilisables au second, en fonction du résultat de la première estimation. Il est donc préférable de réaliser une fois l'acquisition de tous ces pixels, plutôt que de réaliser deux acquisitions successives, associées à chacun des deux étages, figure 3.2 .

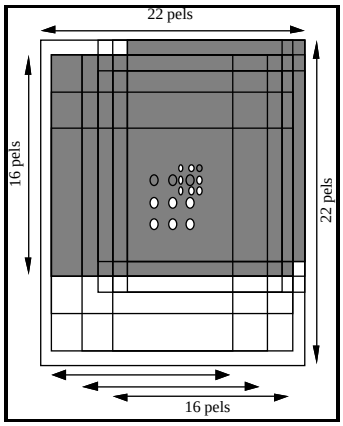


Figure 3.2.

Dans le même ordre d'idées, la fenêtre de référence de 256 pixels n'est acquise qu'une seule fois, mais utilisée successivement pour les étage 1 et 2 (ce qui nécessite tout de même une zone de stockage de 512 pixels de référence).

Dans le but de ne pas accroître considérablement le nombre de plots et donc la surface du circuit, nous avons choisi de n'utiliser qu'un seul bus de 32 bits.

Le problème est donc maintenant de choisir le mode de stockage des pixels sur le circuit. Deux types de stockage pouvaient être envisagés.

- Un stockage avec accés aléatoire (RAM).

- Un stockage avec accés séquentiel (SHIFTER).

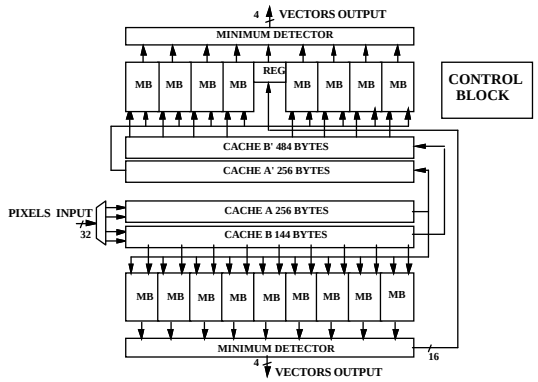
Le premier type nécessite un contrôle plus complexe, par la nécessaire gestion des adresses. De plus l'architecture parallèle adoptée imposerait d'avoir, soit une mémoire multiport, ce qui est à écarter à cause du degré de parallèlisme et de la vitesse intrinséque à chaque opérateur de matching; soit d'éclater la mémoire cache en plusieurs bancs, chacun étant affecté à un opérateur de matching. Cette dernière configuration aurait pour conséquence de supprimer la redondance de pixels inhérente à l'algorithme et par conséquent à augmenter la taille des RAM. Le contrôle nécessaire pour réaliser cette opération d'éclatement introduirait un retard important pour le démarrage du calcul.

La seconde solution (stockage séquentiel) présentent beaucoup d'avantages. Elle supprime pratiquement tout contrôle, puisqu'une horloge suffit à permettre l'avancée des données à l'intérieur de la structure. L'accés des opérateurs de matching à cette

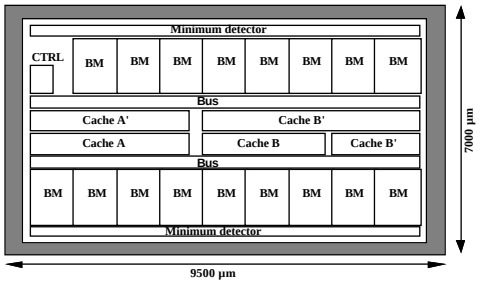
mémoire est réalisée de façon hardware par cablage de différents étages de la structure. Celle-ci permet donc aprés une phase de chargement initial (positionnement des premiers pixels de chaque fenêtre face à leur pointeur respectif) de commencer le calcul tout en continuant le chargement.

Cette structure exploite donc au maximum la redondance des pixels. Seul sa consommation peut poser un problème puisqu'à chaque nouvelle acquisition tous les pixels de la structure sont mis en mouvement, alors qu'en fait un trés petit nombre d'entre eux sont réellements utilisés par les opérateurs de matching à cet instant du traitement. Malgré cet inconvénient nous avons choisi cette solution.

Pour résumer donnons ci-dessous le schéma bloc, le floorplan du circuit, ainsi que les performances attendues.



(a)



(b)

Figure 3.3:

-- Débit d'entrée des pixels sur 32 bits 200Mbytes/s .

-- Débit de sortie des vecteurs 100Kbytes/s.

-- Surface 66mm^2 pour 140 000 transistors en technologie CMOS 1.2µm CMOS.

L'architecture de ce circuit est détaillée dans [DUC92].

(a) Synoptique du processeur élémentaire.

(b) Floorplan.

La complexité ainsi que la taille importante du circuit, nous ont conduit à développer dans un premier temps un circuit de test, dont la finalité fut la validation de différents blocs de base en termes de fonctionnalité, de vitesse ainsi que de consommation.

3.2. Premier circuit de test.

Ce circuit a été réalisé au cours de l'hiver 1992, en technologie AMS CMOS 1.2µm. Il comporte 64 plots pour une surface d'environ 10mm^2. Ce circuit nous a permis d'évaluer les performances de cette technologie, ainsi que de vérifier le fonctionnement de certains opérateurs de base, avant de se lancer dans une intégration plus importante.

Le circuit est constitué de deux blocs de Manhattan dont le fonctionnement est dissocié, l'un permet le test de la fonctionnalité générale, l'autre permet d'isoler chacun des blocs constitutifs pour en vérifier la fonctionnalité séparement. Un autre bloc réalise un registre à décalage sur 22 étages pour des mots de 16 bits; il est constitué par la cellule de base des structures de mémoires décrites dans le paragraphe 3.1 . Enfin le troisième bloc constitue le contrôle du circuit tel qu'il fut conçu initialement. Ce dernier n'a pas été testé pour des raisons d'erreurs de conception qui ont limité son fonctionnement d'une part, ainsi que pour son obsolescence au moment du test. La figure 3.4 représente le layout de ce circuit de test.

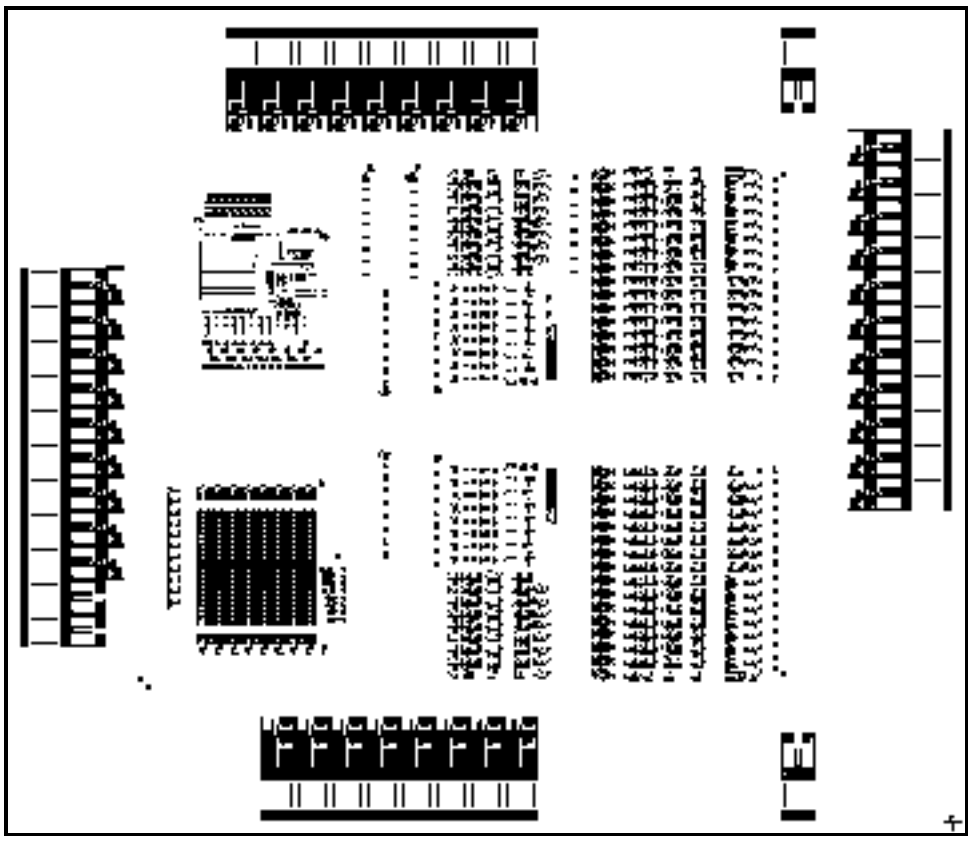


Figure 3.4: Layout du premier circuit de test.

3.2.1. Tests et conclusions sur la conception des blocs de Manhattan.

Ces blocs,figure 3.5, sont réalisés en logique dynamique à une phase d'horloge, True Single Phase Clock (TSPC) [YUAN89] [KOWA92],il est donc impératif de maitriser les temps de commutation de l'horloge ainsi que la stabilisation des données avec précision.

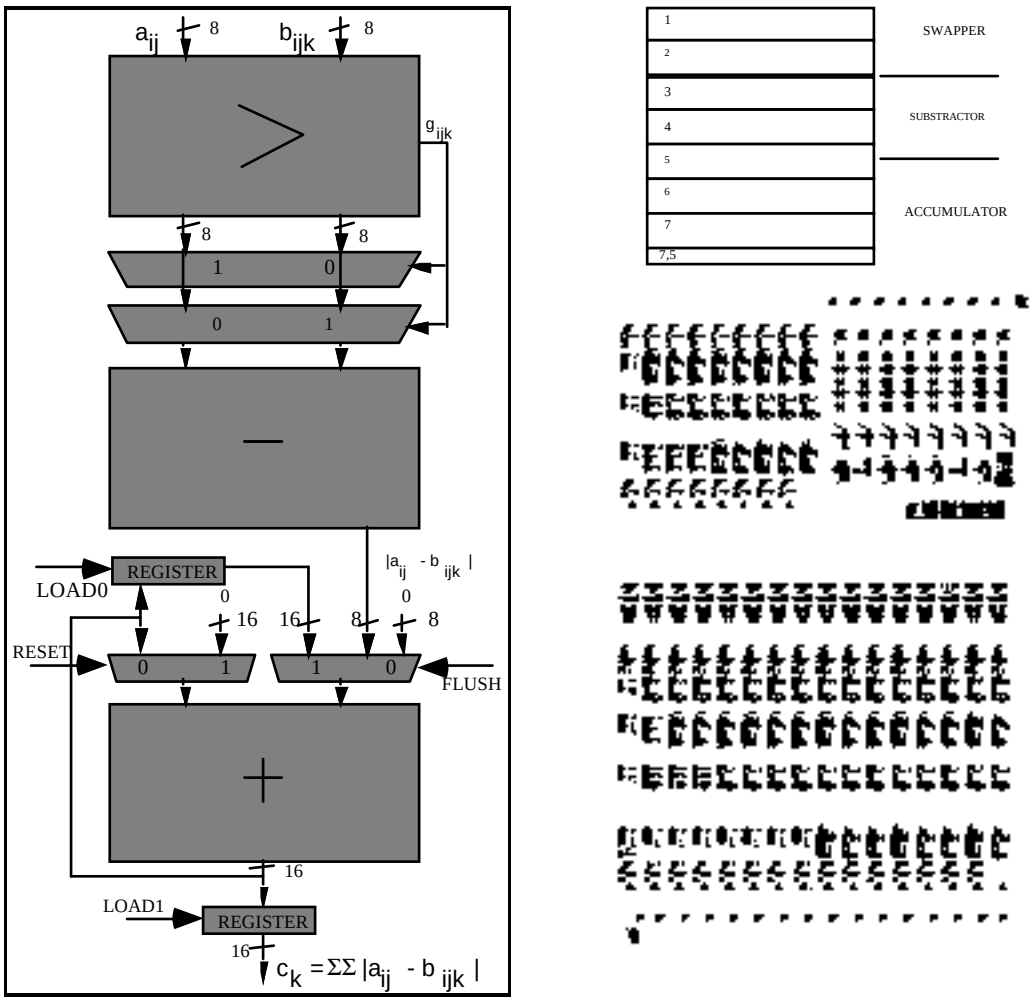


Figure 3.5.

A ce propos, les différents tests effectués ont montré que des temps de transition supérieurs à 5 ns entrainent un mauvais fonctionnement.

La méthodologie de test est la suivante. Le circuit ayant été concut de telle façon que les deux blocs de Manhattan possédent le même bus de sortie (limitation du nombre de plots), ces deux blocs devaient pouvoir être mis en haute impédance. Cette opération est réalisée à partir de portes de transmission minimales. Par ailleurs, l'importance de la charge sur chaque ligne du bus ne permet pas d'observer le fonctionnement des blocs de Manhattan en temps réels pour des vitesses de fonctionnement élevées. Nous avons donc isolé le bloc concerné de ce bus de façon à le faire fonctionner à pleine vitesse, à la fin du test le résultat accumulé est envoyé sur le bus de sortie.

Cette méthodologie nous a permis de mettre en évidence un fonctionnement correct jusqu'à 141MHz, pour une tension d'alimentation variant de 0.5V autour de 5V. Dans un second temps, nous avons voulu localiser l'élément le plus lent; pour cela la séquence de test a été modifié de façon à ne réaliser que des différences de pixels positives, c'est à dire à inhiber la fonction de permutation des opérandes (calcul de la valeur absolue). La vitesse de fonctionnement maximale est alors passée à 153MHz.

Notre conception non suffisament orientée vers le test ne nous a pas permis de mettre en évidence d'autres limitations. Toutefois la corrélation que nous avons pu observer entre nos résultats de simulation et les résultats de test, nous a permis de dégager un autre facteur limitatifs de la vitesse, ce dont nous allons discuter.

La limitation à 153MHz est imposée par la sortie du bloc d'accumulation avec en particulier la contre réaction de la sortie sur l'entrée. Dans la version testée le dernier étage de cette fonction est bufferisé de façon à assurer un temps de transition optimal sur le bus de contre réaction. En contre partie l'étage de bufferisation introduit un nouveau délai. Pour un fonctionnement correct, il faut que le temps de stabilisation de la donnée à l'entrée de l'accumulateur soit inférieur à la demi période de l'horloge, il convient donc de faire un compromis entre le nombre d'étages à traverser et la rapidité de la transition.

Dans cette optique nous proposons une optimisation de l'accumulateur qui consiste à supprimer tous buffers de sortie et à optimiser le dernier étage. Les simulations ainsi réalisées ont permis d'accroître la vitesse jusqu'à 175MHz. Ce qui compte tenu des rapports observés entre le test et la simulation permet d'espérer, et cela sera vérifié dans la suite de ce rapport, un fonctionnement à prés de 200MHz sous cette même technologie.

Nos tests sur cette version des blocs de Manhattan nous conduisent à proposer une architecture optimisée qui sera décrite par la suite.

3.2.2. Test de l'unité de stockage.

Ce bloc est un registre à décalage de 22 étages travaillant sur des mots de 16 bits. La cellule de base est donnée à la figure 3.6 .

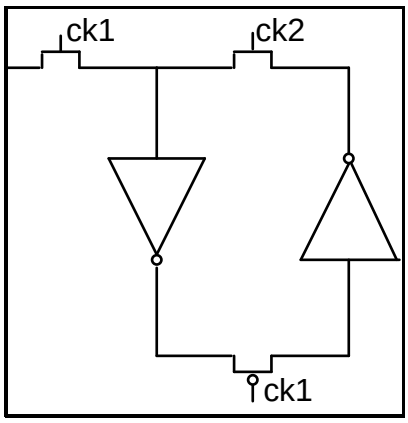


Figure 3.6 .

La structure de cette cellule élémentaire impose d'avoir des transitions d'horloge suffisamment rapides pour que les transistors N et P commandés par ck1 ne conduisent que trés peu de temps ensembles. Le test a mis en évidence la nécessité d'avoir des transitions d'horloges ck1 inférieurs à 5 ns. Nous avons testé le fonctionnement du

registre en mode dynamique (ck2=0), ce qui est le mode de fonctionnement principal dans ce type d'algorithme, la vitesse de fonctionnement maximale est alors de 70MHz avec un rapport cyclique de 50%. En mode statique l'horloge ck2 pourra atteindre la vitesse maximale de 70MHz avec un rapport cyclique de 21% pour assurer le non recouvrement des horloges. En tout état de cause il est inutile de travailler en statique à une telle vitesse.

3.2.3. Conclusions.

Malgré certaines lacunes, limitant la mise en oeuvre du test, ce circuit nous a permis d'évaluer les performances de notre opérateur de matching. Il en résulte que celui-ci est perfectible à deux niveaux. Tout d'abord la détermination de la valeur absolue doit être revue. Ce bloc là, étant difficilement optimisable, par ailleurs les étages de sorties des soustracteur et additionneur devront être optimisés de façon à supprimer les étages de bufferisation.

L'unité de mémorisation a donné entière satisfaction.

Nous avons pu noter un écart d'environ 15 à 20% entre les simulations, réalisées avec Lsim (GDT) en mode adept, et nos mesures. Le simulateur étant plus pessimiste.

La consommation mesurée à 5V pour une fréquence d'environ 150MHz est d'environ 350mwatts pour un opérateur de matching. De cette mesure nous pouvons extrapoler la consommation des 17 opérateurs de matching à prés de 6watts. En tenant compte de tous les autres dispositifs du processeur d'estimation de mouvement, en particulier des unités de stockage de pixels la consommation risque d'être prohibitive.

3.3. Second circuit de test.

Ce second circuit avait pour but la caractérisation de nos nouveaux opérateurs de matching. Il devait aussi permettre de valider, la synchronisation des pixels avec les signaux de contrôle et d'horloge. A cet effet toute la fonctionnalité pour calculer deux distorsions en parallèle a été mise en oeuvre. Le bloc permettant de déterminer le meilleur matching parmi les deux distorsions a lui aussi été implanté. Le floorplan ainsi que le layout du circuit sont donnés figures 3.7 et 3.8 .

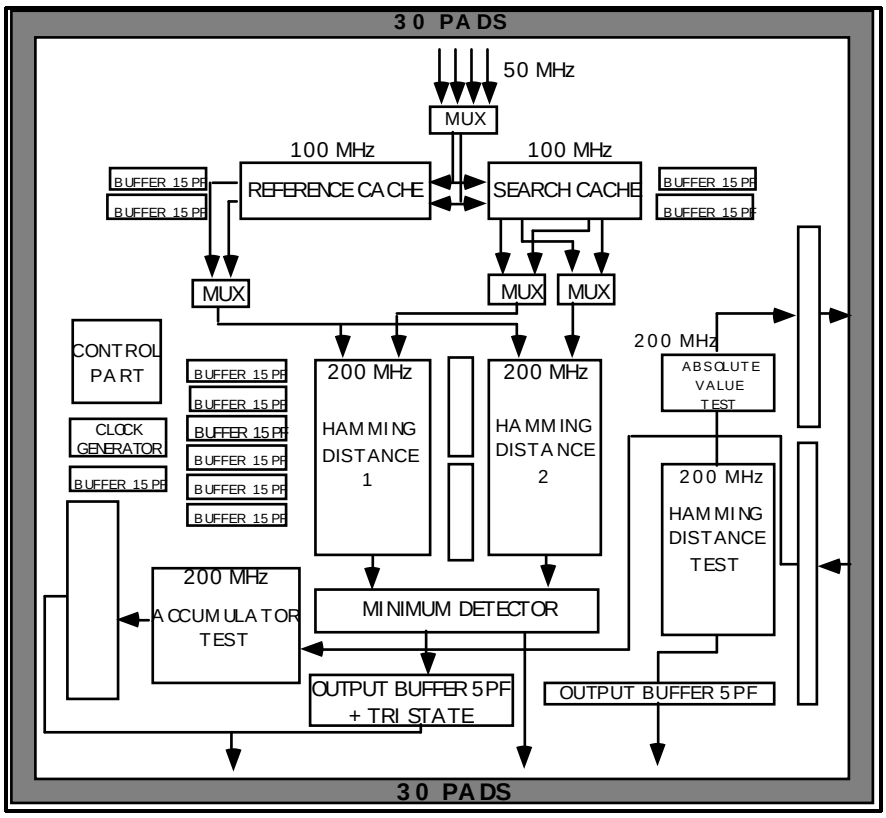


Figure 3.7 .

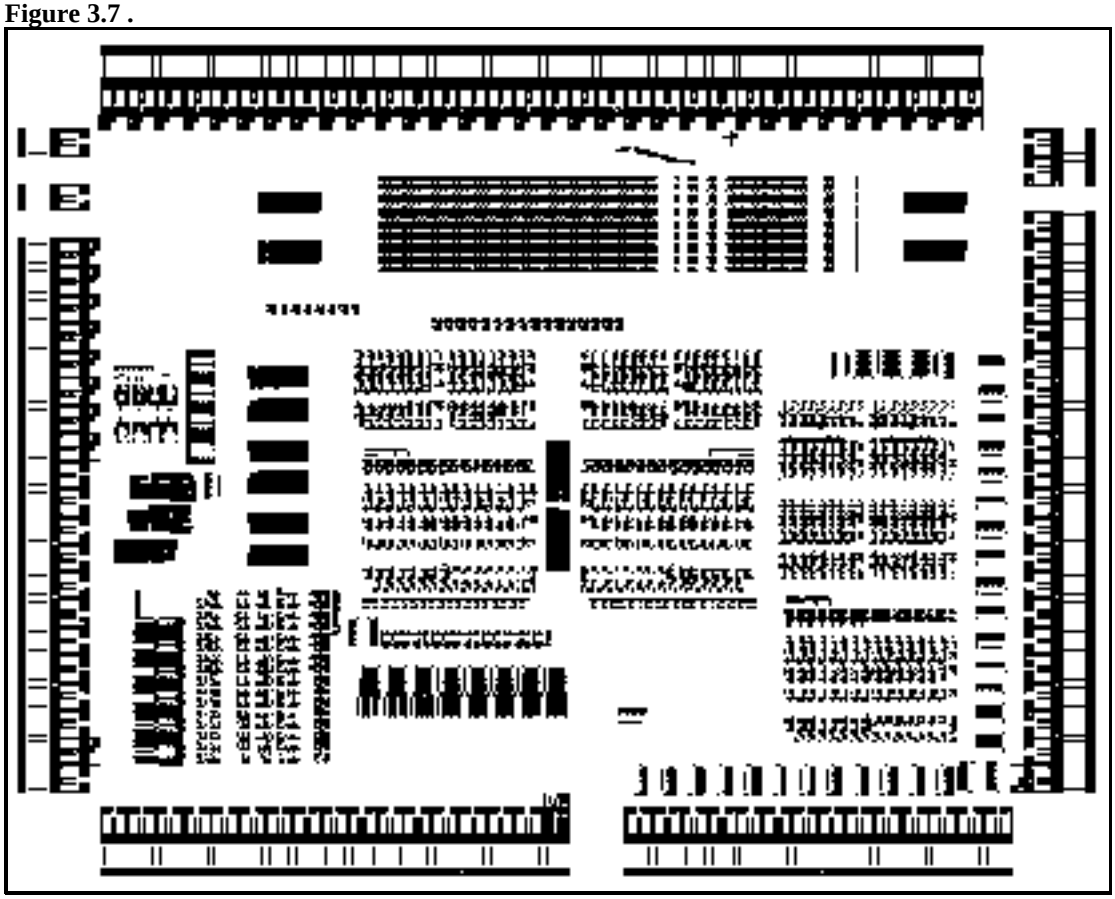


Figure 3.8 .

Une erreur de layout au niveau des drivers d'horloge (buffer 15picoFarad) n'a pas permis le test de cette dernière fonctionnalité. Nous allons tout de même décrire les

points critiques qui ont fait l'objet de simulations approfondies. Par la suite, nous présenterons les modifications faites au niveau de l'opérateur de matching ainsi que les résultats de tests obtenus.

3.3.1. Problème de la synchronisation des pixels avec les signaux de contrôle.

Nous avons vu précédemment que l'accés aux mémoires externes est réalisé à uen fréquence bien moindre que la fréquence de travail des opérateurs de matching. Dans notre cas, il existe un rapport 4 entre la fréquence d'acquisition des pixels et la fréquence de calcul, il faut donc réaliser un multiplexage de 32 bits vers 4 fois 8 bits. Cette opération est réalisée en deux temps, à savoir à l'entrée du circuit un premier multiplexage permet de doubler la vitesse, les données sont ensuite décalées à 100MHz à l'intérieur des registres à décalage (cache). La seconde partie du multiplexage est faite par des blocs interfaces au niveau de chacune des sorties des registres à décalages. L'opération de multiplexage est faite à 100MHz, puis le pixel sélectionné est synchronisé (à la fréquence de 200MHz), puis bufferisé progressivement à l'intérieur de deux latches TSPC. Un dernier buffer CMOS assure un temps de transition inférieur à la demi période de l'horloge de synchronisation (2.5 ns) sur une charge de 1pF dans le cas critique, figures 3.9 et 3.10 .

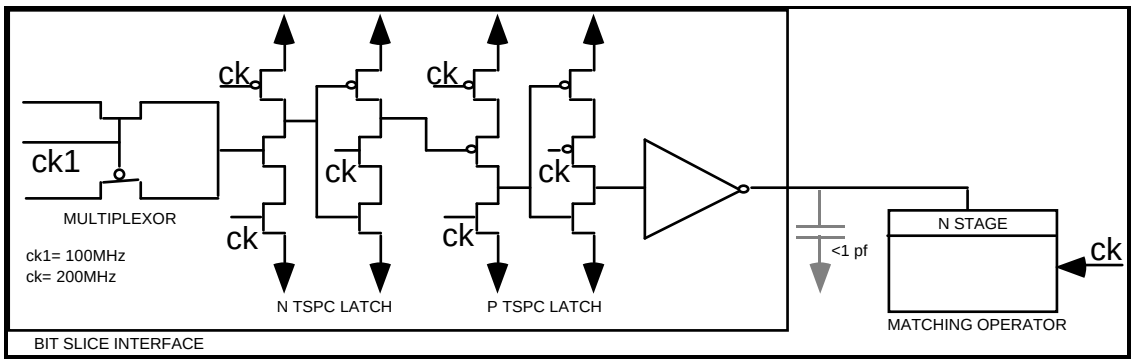


Figure 3.9 .

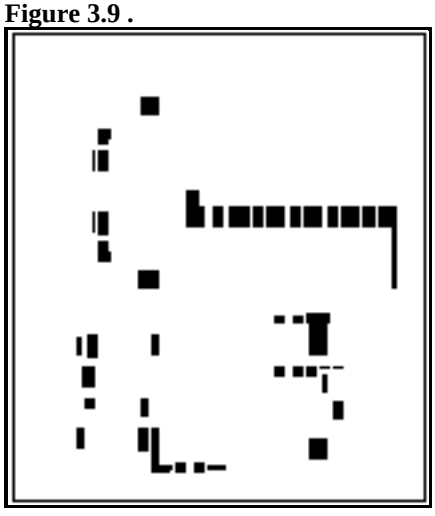


Figure 3.10 .

La synchronisation des signaux de contrôle à la fréquence de 200MHz peut être abordée selon deux stratégies.

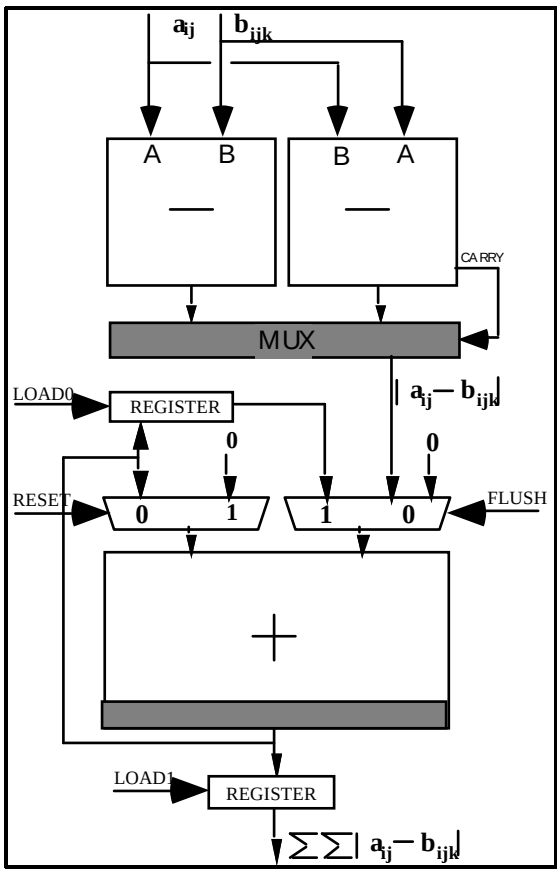
La première, celle que nous avons choisie dans ce second circuit, repose sur la centralisation du contrôle en un point, pour ensuite distribuer les différents signaux bufferisés dans tout le circuit et cela de façon indépendante de la fréquence de fonctionnement. Dans cette hypothèse, il est nécessaire de connaître le nombre de couches logiques (buffer) associées à chacun des signaux. Une synchronisation sûre impose d'avoir des retards identiques au retard critique, imposé par le signal le plus fortement chargé, pour tous les signaux agissant sur un même bloc fonctionnel. Concrétement dans notre cas l'horloge pilotant les opérateurs de matching voit une charge de 15pF, le délai introduit par cet étage de bufferisation est de l'ordre de 2 ns; ce qui veut dire que tous les signaux synchrones de cette horloge doivent subir le même délai, sous peine de risquer de décaler tout le fonctionnement d'une période d'horloge (200MHz). Donc des signaux tel que *flush*, *load* ou *reset* qui pourrait supporter des bufferisation plus modeste devront passer à travers ces buffers de 15pF.

Actuellement, notre stratégie est plutôt axée sur la délocalisation du contrôle au plus prés de chaque bloc. Les blocs fonctionnant à des vitesses élevées possédent une unité de contrôle propre bâtie autour d'un compteur TSPC. Ce compteur étant lui-même commandé par des signaux de *start* et *stop* dont la fréquence est peu élevée. La communication des résultats est synchrone du contrôle de plus haut niveau.

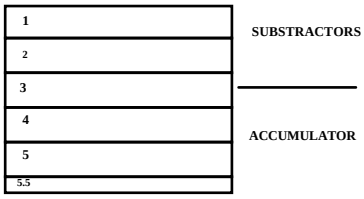
3.3.2. Tests effectués sur la seconde version de l'opérateur de matching.

Cette seconde version de l'opérateur de matching est différente au niveau de l'évaluation de la valeur absolue, qui comme nous l'avons vu produisait une limitation des performances. Dans ce cas deux différences (A-B et B-A) sont évaluées en parallèles, selon le résultat du bit de poids fort tel ou tel solution est choisie à l'aide d'un multiplexeur, comme l'indique la figure 3.11 .

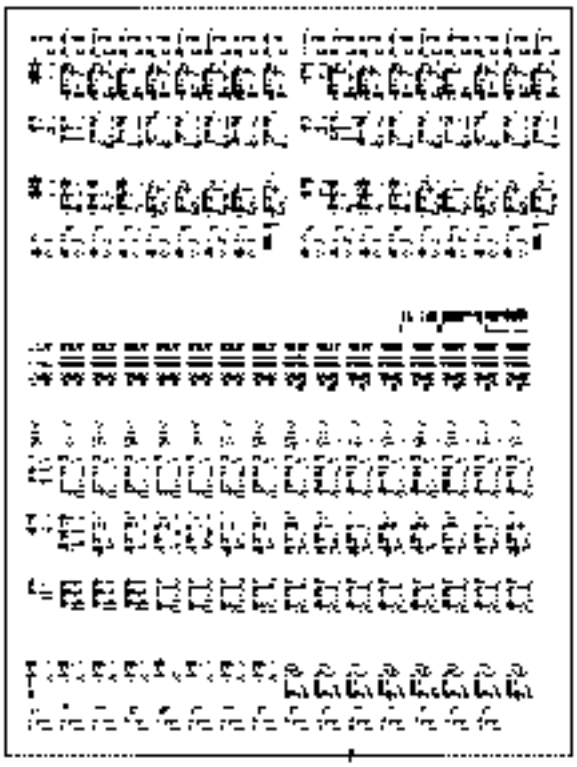
Les étages de sorties des soustracteurs ont été redimensionné de façon à permettre la stabilisation de la différence choisie en moins d'une demi période (2.5 ns) à l'entrée de l'accumulateur, sans utiliser d'étages de bufferisation.



(a)



(b)



(c)

Figure 3.11 : (a) Schéma bloc. (b) Etage de pipeline. (c) Layout.

4. CONCLUSIONS: ARCHITECTURES PROGRAMMABLES.

Les travaux décrits ci-dessus amènent plusieurs conclusions, qui constituent le point de départ d'une approche plus polyvalente des problèmes d'estimation de mouvement basés sur la technique du block-matching.

Le chapitre 2, a mis en évidence une des caractéristiques majeures de ce type d'algorithme à savoir comment gérer et répartir l'énorme quantité de pixels entre les différents processeurs. Deux solutions ont été envisagé.L'une propose une affectation fixe de chaque processeur à une partie de l'image pour un niveau de résolution donné. L'autre propose de faire un travailler un processeur du plus bas niveau de résolution jusqu'au plus haut. Cette dernière solution étant toutefois peu réaliste dans le cadre d'un algorithme hièrarchique travaillant sur plusieurs niveaux de résolution. Il ressort que l'approche multi-résolution est plus complexe à réaliser qu'une approche hièrarchique sur un seul niveau. A cause du partitionnement de la mémoire qu'il faut réaliser à chaque niveau si l'on veut utiliser une architecture verticale, et à cause de l'utilisation de deux circuits supplémentaires pour la génération des différents niveaux de résolution.

Le chapitre 3 propose une architecture de processeur élémentaire entiérement spécifique d'un algorithme dérivé de l'algorithme de Koga [KOGA81]. L'intégration de ce processeur a été entreprise, puis abandonnée pour les raisons suivantes:

- Le circuit trop spécifique, ne répond pas aux nécessaires évolutions de l'algorithmique. En particulier il ne permet pas de traiter les algorithmes hièrarchiques travaillant sur un seul niveau de résolution. Car dans ce cas il faut pouvoir varier les paramètres de matching tout au long de l'exécution de l'algorithme.

- La consommation du circuit estimé à environ 12 à 15Watts (200MHz) est trop élevée pour permettre une dissipation efficace d'une telle puissance.

- Un goulôt d'étranglement au niveau des possibilités d'acquisition de pixels dans cette approche ne permet pas d'utiliser au mieux la puissance de calcul de l'unité de matching. L'utilisation d'un seul port d'entrée ne permet pas de parallèliser l'acquisition des pixels de recherche et de référence, ce qui entraine une utilisation non optimale de la puissance de calcul.

En revanche nos deux circuits de test nous ont permis de valider le fonctionnement de différents opérateurs intervenant dans toute opération de block-matching. Ces opérateurs seront donc réutiliser par la suite.

Aprés une étude approfondie des différents algorithmes de block-matching [HERV93]. Il apparait que la majorité de ceux-ci sont en fait constitués de sous blocs capables de traiter en parallèles trois distorsions. Le but est donc de constituer une cellule de base qui contiendrait trois opérateurs de matching tel qu'ils ont été caractérisé dans nos circuits de test. Le parallèlisme au niveau de l'acquisition des pixels est accru, de façon à égaler au mieux le flux d'acquisition avec la puissance de calcul de l'unité de matching. L'aspect évolutif du circuit est principalement réalisé grâce à la structure de mémorisation des pixels, qui permet de faire varier à la fois la taille de la fenêtre de matching mais aussi l'intervalle entre deux points de matching. Un degré de souplesse supplémentaire est adjoint par la possibilité de modifier les paramètres de l'algorithme sans interrompre de façon durable le travail du processeur. Ce mode de fonctionnement est réalisé à l'aide d'un port d'entrées-sorties série.

La modification en cours de traitement des paramétres de l'algorithme doit se faire avec un degré de fiabilité accru. Pour cela les paramétres de configuration peuvent être transmis avec leur signature.

Le processeur élémentaire est en permanence asservit à un contrôleur de plus haut niveau, mais dont la fréquence de fonctionnement est moins critique (environ 50MHz). A ce niveau, plusieurs options sont possibles. Soit la puissance de calcul offerte par le processeur élémentaire (environ 600Mpixels/s) est suffisante pour notre application. Soit l'algorithme ou la définition de l'image sont tels qu'il faut recourir au parallélisme. Parallélisme qui peut être réalisé de façon discréte, ou par la définition d'un processeur contenant trois de ces processeurs élémentaires. Ce dernier cas nous aménerait à définir un circuit d'environ 250 plots pour une surface de 150 à 200 mm^2, avec bien sur de gros problèmes de consommation (de l'ordre de 15Watts). L'approche Multi Chip Module pourrrait être alors une alternative intéressante pour un parallèlisme massif, avec une meilleure répartition de la puissance.

REFERENCES.

[DUC92] P. Duc, P. Favrat, D. Nicoulaz,"Estimateur de mouvement intégré",Rapport interne, Laboratoire d'Electronique Générale / Centre de Conception de Circuits Intégrés, Ecole Polytechnique Fédérale de Lausanne, Février 1992.

[BIER88] M.Bierling,"Displacement Estimation by Hierarchical Block Matching", Visual Communications and Image Processing, Volume 1001, pp 942-951, 1988.

[EBRA91] T.Ebrahimi, F.Dufaux, I.Moccagatta, P.Cicconi and M.Kunt, "EPFL proposal for MPEGII",Technical Report 40, ISO-IEC/JTC1/SC29/WG11, Kurihama , Japan, November 1991.

[HERV92] R.Hervigo, J.Kowalczuk, D.Mlynek, "A Multiprocessor Architecture for HDTV Motion Estimation system" published in IEEE Transactions on Consumer Electronics, August 1992 Volume38 Number3, and presented in june 1992 in Chicago to International Conference on Consumer Electronics (ICCE).

[HERV93] R.Hervigo, "Etat de l'art dans le domaine des algorithmes de Block-matching: Description d'un processeur élémentaire polyvalent", Rapport interne, Laboratoire d'Electronique Générale / Centre de Conception de Circuits Intégrés, Ecole Polytechnique Fédérale de Lausanne, Novembre 1993.

[KOGA81] T.Koga et al., "Motion Compensated Interframe Coding for Video Conferencing", in Proceedings Nat. Telecommunication conf., New Orleans, LA, November 1981, pp. G5.3.1-G5.3.5 .

[KOWA92a] J. Kowalczuk, R. Hervigo, T. Ebrahimi, M. Mattavelli, D. Mlynek, M.Kunt, "*Hardware evaluation of EPFL proposal for MPEGII*, Technical Report 40, ISO-IEC/JTC1/SC29/WG11, Kurihama , Japan, November 1991.

[KOWA92b] J. Kowalczuk and D. Mlynek, "Un Nouvel Algorithme De Generation D'Additionneurs Rapides Dédiés Au traitement d'images", Proceeding of the Industrial Automation Conference, p. 20.9-20.13, Montreal, Québec, Canada, June 1992.

[KOWA92c] J.Kowalczuk, R.Hervigo, V.Bonzom, Z.Triba and D.Mlynek" A 200MHz Macrocell Family for High Throughput Video Processing", presented in february 1993 in Paris to EUROASIC.

[KOWA93] J.Kowalczuk, "On the design and implementation of algorithms for multi-media systems", Thèse de doctorat soutenue 1er décembre 1993, Ecole Polytechnique Fédérale de Lausanne.

[LSIL91] LSILOGIC, Digital Signal Processing Databook, description of motion estimation processor L64720, page 51, September 1991.

[MUSM85] H.G.Musmann, P.Pirsch and H.J.Grallert,"Advances in Picture Coding", Proceedings of the IEEE, Vol.73(4), pp523-548, April 1985.

[THOM90] SGS-THOMSON Image Processing Databook, 1st edition, description of motion estimation processor STI3220, page 115, October 1990.

[TRIB93] Z.Triba,"Codage Arithmétique", Rapport interne, Laboratoire d'Electronique Générale / Centre de Conception de Circuits Intégrés, Ecole Polytechnique Fédérale de Lausanne, Septembre 1993.

[YUAN89] J. Yuan, I. Karlsson and C. Svensson, "A True Single Phase Clock Dynamic CMOS Circuit Technique", IEEE JSSC Vol 22, No 5 Oct. 1989.e