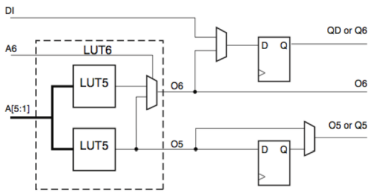
1 Mémoires

- RAM
- ROM
- CAM
- FIFO

Amélioration dans le Spartan-6 : LUT4 → LUT6 pour de meilleures performances.
Les BRAM sont disposées à côté des CLB pour améliorer la vitesse.

- SliceX (50 %) : LUT6, 8 flip-flop
- SliceL (25 %) : + MUX, Carry
- SliceM (25 %) : + distributed RAM, shift register



1.1 Types de mémoires

- On-chip distribuée, utilisation des LUT des CLB (SliceM uniquement).
  - "Granulaire"
  - Peu d'impact sur le routage
  - Petites FIFO / machines d'état
  - **Rapide**
  - Écriture synchrone, lecture asynchrone (synchrone avec utilisation des flip-flops)
  - Initialisation à la configuration
  - LUT5 = 32 bits. LUT6 = 64 bits.
  - Possibilité de combiner en 256x1, 32x8, etc...
- On-chip block RAM (BRAM)
  - Buffer de moyenne taille
  - **Rapide**
- Externe
  - Memory Controller Block (MCB)
  - Flexible
  - Très grande capacité

Toutes les solutions mémoire doivent être rapides et flexibles. Un bitstream doit pouvoir être chargé dans une RAM ou ROM.

Example avec un slice contenant 4 LUT6 :

Single Port	Dual Port	Simple Dual Port	Quad Port
32x2	32x2**D**	32x6**SDP**	32x2**Q**
32x4	32x4**D**	64x3**SDP**	64x1**Q**
32x6	64x1**D**		
32x8	64x2**D**		
64x1	128x1**D**		
64x2			
64x3			
64x4			
128x1			
128x2			
256x1			

1.1.1 Shift register

SRL16CE ou SRL32 (sur SliceM uniquement)

- Initialisation possible
- Chargement impossible
- Pas de reset
- Longueur variable (déterminée par l'adresse). Utile pour des buffers dynamiques.
- Max 128x1 dans une slice (32 par LUT).
- FIFO synchrone
- Content-addressable memory (CAM)
- Générateur de pattern
- Compensation de délai

1.1.2 BRAM

Possibilité de

- Diviser et adresser individuellement les BRAM.
- Grouper et adresser de manière unique des BRAM.
- Initialisation possible
- Reset possible
- Bits de parité (pour 8 bits de données)
- Placés à côté des multiplicateurs pour multiply-accumulate

  $$1 \times 18\text{K BRAM} \longleftrightarrow 2 \times 9\text{K BRAM}$$

- True dual port
  - Horloge différente pour chaque port
  - Lecture et écriture synchrone (en même temps)
- Simple dual port
  - Un port en lecture, un port en écriture
  - Lecture et écriture en même temps en 1 cycle
- Single port
  - Lecture OU écriture en 1 cycle

1.1.3 Types de mémoires externes

- DRAM
  - SDRAM
  - DDR / DDR3
  - FCRAM
  - RLDRAM
- SRAM
  - Sync SRAM
  - DDR SRAM
  - ZBT
  - QDR
- Flash
- EEPROM

1.2 Implémentation

RAM distribuée : Process synchrone, pas de reset asynchrone

Shift register : Process synchrone, pas de reset asynchrone, serial in et out.

1.3 Memory Controller Block

Sur le Spartan-6

- 4 contrôleurs
- Mémoires 4, 8 ou 16 bits.
- Support pour DDR1/2/3, LP DDR

2 PLL

CMT : Clock Management Tile

PLL : Phase Lock Loop

DCM : Digital Clock Manager

GCLK : Global Clock

VCO : Voltage Controlled Oscillator

DLL : Delay-locked loop

DFS : Digital Frequency Synthesis

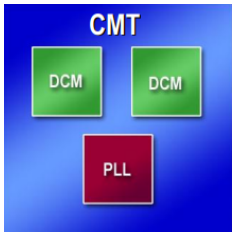
CDC : Clock domain crossing

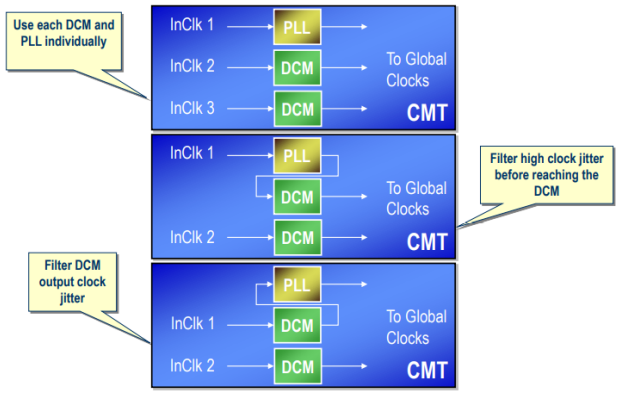
2.1 Horloges

2 domaines d'horloge dans le spartan-6

- 2 horloges I/O jusqu'à 1 GHz, gérées par BUFPLL, 1 pour chaque côté du FPGA
- 4 horloges I/O jusqu'à 1 GHz, gérées par BUFIO2, 1 pour chaque demi-côté du FPGA
- max 16 horloges internes jusqu'à 400 MHz

1 à 6 CMT (Au maximum 12 DCM et 6 PLL).





2.2 Exemple de génération d'horloge

- Utilisation d'une pin GCLK reliée à une horloge précise à 50 MHz (single-ended clock input sur horloge 0 par exemple)
- Utilisation d'une PLL pour convertir l'horloge 0 en 500 MHz sur l'horloge 1

Aussi possible d'utiliser les pins GCLK comme I/O standards si elles ne sont pas utilisées pour les horloges.

2.3 Réseau d'horloge

Distribuée en arbre pour avoir un temps de propagation égal à tous les noeuds. Un réseau d'horloge peut aussi être utilisé pour la distribution d'un signal global (reset)

2.4 Multiplicateur d'horloge

Multiplexage de deux horloges pour en créer une nouvelle.

2.5 Gated clock

Activation et/ou désactivation d'une horloge sans risque de glitchs (activation de l'horloge hors des temps prévus).

2.6 PLL

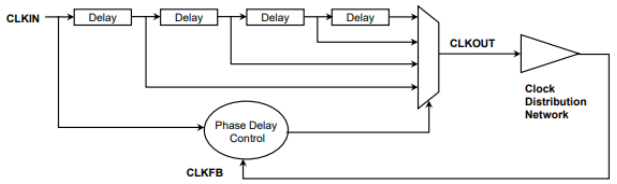
Possibilité d'utiliser une PLL pour

- Diminuer la fréquence
- Augmenter la fréquence
- Modifier la phase (par exemple compenser
- Diminution du jitter interne (ou encore plus sur des entrées externes)
- Remplace les PLL discrètes et les VCO

La PLL accepte une plus grande plage de fréquence, rapports cycliques et jitter que le DCM.

2.7 DCM

- Entièrement numérique
- Fonctionnalités variées



2.8 Routage

BUFIO2 permet de router les signaux d'horloge d'entrée sur des lignes dédiées pour atteindre des horloges I/O, des PLL/DCM ou des buffers BUFG.
Les CMT sont situés dans la colonne centrale du FPGA

2.9 Mémoires

- SDR : Lecture/écriture uniquement sur le flanc montant/descendant
- DDR : Lecture/écriture sur les flancs montants et descendants

2.10 DLL

- 5 MHz à 250 MHz
- De-skew
- Correction du rapport cyclique
- Décalage de phase

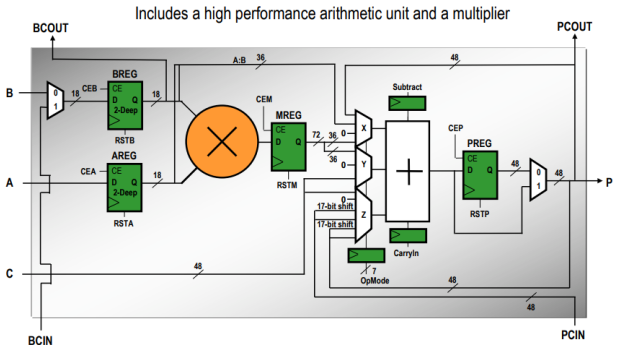
2.11 Synchronisation

Utilisation d'un 2-FF (dual flip-flip synchroniser)

3 DSP

MAC : Multiply-accumulate
Utilisation typique pour les filtres FIR de taille N. Avec une seule "machine" : un résultat tous les N coups d'horloges. Avec N machines : un résultat tous les coups d'horloge.
Multiplicateur 18×18 dans les Virtex II, placés entre les BRAM et les CLB pour des MAC performants.

3.1 DSP48



- Plus de 40 opération possibles (7 bits OpMode)
- Possibilité de changer l'opération à chaque coup d'horloge

3.1.1 DSP48E

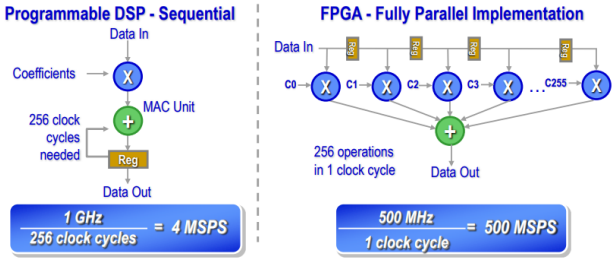
Ajout de :

- ALU haute performance.
- comparateur de pattern

3.2 Utilisations

- Charge de calcul importante
- Calculs FIR rapides (parallélisation)
- Calculs multi-canaux en parallèle
- Architecture dédiée pour l'application (coût vs vitesse)
- Intégration toute la logique dans une FPGA (réduction des coûts).

3.3 Exemple de parcellisation : filtre FIR



4 Liaisons série

4.1 Types de liens

- Bus parallèle, un byte par coup d'horloge
  - Fréquence limitée
  - Nombre de liens (encombrement)
- Série asynchrone (UART, CAN)
  - Trame
  - Débits limités
  - Signal de start
  - Le récepteur utilise sa propre horloge
- Série synchrone (HDMI, PCIe, USB, Ethernet)
  - Débits élevés
  - Signal d'horloge sur une ligne séparée ou sur la même ligne (préambule)
  - Le récepteur se synchronise avec l'horloge de l'émetteur

4.2 Encodage de l'horloge

4.2.1 8b10b ou 64b66b

- But de moyenner le nombre de 0 et de 1 sur la ligne (moyenne nulle)
- Pas de mots "0000000000" et "1111111111"
- Récupération du signal d'horloge avec les transitions supplémentaires.
- 12 caractères "K" qui permettent de contrôler le flux.

4.2.2 Vitesse de transmission / réception

Émetteur plus lent que le récepteur : le récepteur doit attendre.
Émetteur plus rapide que le récepteur : le récepteur rempli son buffer

Solution : Utiliser un "elastic buffer" et mettre en place une méthode de communication entre les deux parties pour éviter de perdre des données.

Deuxième solution : CC (Clock Compensation). Lorsque le récepteur tombe sur un CC, il attend quelques cycles d'horloge (lorsque l'émetteur est plus lent que le récepteur). Si l'émetteur est plus rapide que le récepteur, le récepteur va recevoir plusieurs CC et va sauter du nombre de CC introduits dans le flux. Ceci au pointeur de lecture de se retrouver en face du pointeur d'écriture.

4.3 GTP

- Encodage 8b10b
- 10 bits à 125 MHz
- 1 bit à 2.5 GHz (DDR)
- Buffer élastique sur le récepteur
- FIFO de compensation sur l'émetteur (au cas où les deux domaines d'horloge ne sont pas parfaitement synchronisés)

Un décalage volontaire permet de garantir que les buffer ne débordent jamais

- Le domaine d'horloge (DH) utilisateur de l'émetteur est légèrement plus faible que le DH de transmission
- le DH de l'émetteur est le même que le DH du récepteur (partie transmission)
- le DH de transmission du récepteur est légèrement plus lent que le DH utilisation du récepteur

3 configurations :

- Configuration commune
- Configuration faible latence
- Configuration Aurora