



(12) FASCICULE DE BREVET

- (11) N° de publication : **MA 34265 B1** (51) Cl. internationale : **G06T 1/20; G06F 1/24**
- (43) Date de publication : **01.06.2013**

-
- (21) N° Dépôt : **34305**
- (22) Date de Dépôt : **26.10.2011**
- (71) Demandeur(s) : **MAScIR (Moroccan Foundation for Advanced Science, Innovation & Research), 303 Business Center Technopolis RABATSHORE 11000 RABAT-SALE RABAT-SALE (MA)**
- (72) Inventeur(s) : **Noredine KACHKACHI ; François BOURZEIX**
- (74) Mandataire : **MOHAMMED EL AMRANI**

-
- (54) Titre : **CONCEPTION D'UN MODULE VHDL POUR CONVERSION AUTOMATIQUE DE DEBIT D'IMAGES.**
- (57) Abrégé : LE CONVERTISSEUR AUTOMATIQUE DE DÉBIT D'IMAGES EST UN BLOC ÉLECTRONIQUE ÉCRIT EN LANGAGE DE PROGRAMMATION HARDWARE QUI EST LE VHDL. CE BLOC VHDL PERMET LA GESTION AUTOMATIQUE D'UN FLUX VIDÉO PROVENANT D'UNE SOURCE EXTERNE (EX CAMÉRA NUMÉRIQUE). LE RÔLE DU BLOC EST D'ADAPTER LE FLUX D'IMAGES ENTRANT (PROVENANT DE LA CAMÉRA) À LA BANDE PASSANTE DE L'INTERFACE QUI FAIT LE TRANSFERT DE LA VIDÉO VERS LA DESTINATION (EX INTERFACE USB). L'UTILISATEUR N'A PAS BESOIN DE PROGRAMMER LE RAPPORT ENTRE LE DÉBIT ENTRANT ET LE DÉBIT SORTANT, LA SYNCHRONISATION SE FAIT AUTOMATIQUÉMENT.

3. ABREGE TECHNIQUE

Le convertisseur automatique de débit d'images est un bloc électronique écrit en langage de programmation Hardware qui est le VHDL. Ce bloc VHDL permet la gestion automatique d'un flux vidéo provenant d'une source externe (ex Caméra numérique). Le rôle du bloc est d'adapter le flux d'images entrant (provenant de la caméra) à la bande passante de l'interface qui fait le transfert de la vidéo vers la destination (ex interface USB). L'utilisateur n'a pas besoin de programmer le rapport entre le débit entrant et le débit sortant, la synchronisation se fait automatiquement.

1. DESCRIPTION DE L'INVENTION

Le produit réalisé est un système hardware numérique appelé « Frame Rate Converter » dénommé ci-après par le sigle FRC, codé en VHDL et destiné à être synthétisé sur des FPGA ou des ASICS. C'est un module VHDL qui fait office d'adaptateur de qualité et de débit d'images entre une source (camera) et entre la destination appelée Host (Interface de transfert de données). Le FRC peut être utilisé pour transmettre tout type de données (Vidéo, audio, datas,...) entre une source et une destination non synchronisées qui disposent de domaine d'horloges et de bandes passante différentes. Le rôle du FRC est de synchroniser la réception des données par la destination par rapport leur émission par la source.

Dans le cas des applications vidéo, la source peut débiter les images dans un rythme donné avec une résolution donnée dans les standards de transfert parallèle ou série (MIPI). Le FRC se charge d'adapter ce débit d'images au rythme supporté par le contrôleur de transfert de donnée vers le Host.

1.1. DOMAINE DE LA TECHNIQUE

Le domaine de l'invention correspond à la conception micro-électronique et plus particulièrement au design numérique RTL Front End.

La réalisation rentre dans le domaine de la conception numérique VHDL pour FPGA.

Le métier consiste dans un certain nombre d'étapes de la conception depuis l'idée jusqu'à la livraison finale.

Les étapes majeures de cette réalisation sont :

- Spécification de la fonctionnalité et écriture du document d'architecture.
- Codage VHDL et implémentation de la spécification.
- Simulation du module réalisé en VHDL.
- Synthèse et mappage du module sur la cible, qui est une FPGA d'Altera.
- Intégration et validation et du module dans son système complet qui s'agit d'une carte caméra.

1.2. ETAT DE LA TECHNIQUE

Les “Frame Rate Converter” actuels utilisent des rapports de conversion programmable.

L'utilisateur doit configurer l'IP ou le système en lui rentrant les valeurs de la résolution, ainsi que le rapport entre le débit entrant et le débit sortant.

1.3. IDENTIFICATION

1.3.1. DEFINITION DE L'INVENTION

L'invention consiste dans un système d'acquisition hardware numérique FRC (1) conçu pour recevoir un flux de données numériques (Vidéo, Audio, ou autres,...) provenant d'une source donnée (Caméra, Capteur, émetteur ...) qui dispose d'un domaine d'horloge et d'une bande passante donnée, et adapter la qualité et le débit de ces données reçues avant de les retransmettre vers une destination asynchrone par rapport à la source. La destination peut posséder un domaine d'horloge et un débit différent de celui de la source. Le système inventé permet d'assurer la synchronisation automatique et transparente d'une manière dynamique entre le flux de données entrantes et le flux de données sortantes.

Le système comprend :

- Une interface d'entrée (2) qui fait l'acquisition et le traitement numériques des données numériques avec une fréquence d'horloge d'entrée variable.
- Une interface de sortie (8) numérique qui fait l'adaptation du format des données et l'envoi vers la destination avec une fréquence d'horloge de sortie variable et différente de la fréquence d'entrée.
- Une FIFO d'entrée (51) pour faire le passage du domaine d'horloge de l'interface d'entrée vers le domaine d'horloge de la SDRAM. L'horloge d'entrée de cette FIFO est l'horloge de l'interface d'entrée. L'horloge de sortie de cette FIFO est l'horloge de l'SDRAM. Elle sert pour stocker momentanément une partie de la donnée d'entrée. La communication entre la FIFO d'entrée et la SDRAM se fait à l'aide de rafales (Burst) de longueur variables génériques. Les données lues de cette FIFO sont envoyées vers la SDRAM.
- Une FIFO de sortie (52) pour faire le passage du domaine d'horloge de la SDRAM vers le domaine d'horloge de l'interface de sortie. L'horloge d'entrée de cette FIFO est l'horloge SDRAM. L'horloge de sortie est l'horloge de l'interface de sortie. Elle sert pour stocker momentanément une partie de la donnée de sortie. La communication entre la FIFO de sortie et la SDRAM se fait à travers des rafales (Bursts) de longueur variable générique. Les données

lues de la SDRAM sont provisoirement stockées dans cette FIFO avant d'être envoyées vers l'interface de sortie.

- Une FSM (4) (Machine d'états à 19 états) qui détermine si l'opération à effectuer dans la SDRAM est une rafale d'écriture ou bien une rafale de lecture. Cette FSM organise le passage entre les différentes étapes et notamment les conditions de passage d'une opération d'écriture de Burst vers une opération de lecture de Burst et vice-versa. L'opération d'écriture d'un Burst dans la SDRAM est prioritaire sur l'opération de lecture d'un Burst depuis la SDRAM.
- Un système d'arbitrage constitué de deux processus combinatoires (3) un processus pour l'écriture et un processus pour la lecture. Le processus d'écriture calcul le pointeur d'écriture vers l'une des trois zones (Page1 ou page2 ou page3) mémoires choisies pour l'écriture. La prochaine valeur calculée du pointeur d'écriture dépend de la valeur courante du pointeur d'écriture, de la valeur courante du pointeur de lecture et de l'avancement du pointeur d'écriture dans une page en cours d'écriture, qui peut correspondre à la fin de la page, à l'intérieur de la page ou bien au cas où aucune page n'a été écrite encore.

Le processus de lecture calcul le pointeur de lecture vers l'une des trois zones (Page1 ou page2 ou page3) mémoires choisies pour la lecture. La prochaine valeur calculée du pointeur de lecture dépend de la valeur courante du pointeur de lecture, de la valeur courante du pointeur d'écriture et de l'avancement du pointeur de lecture dans une page en cours de lecture, qui peut correspondre à la fin de la page, à l'intérieur de la page ou bien au cas où aucune page n'a été écrite lue.

- Un contrôleur de mémoire standard (6) qui permet de générer les différents signaux relatifs à une interface SDRAM standard.
- Une interface de sortie (8) numérique avec une fréquence de sortie différente de la fréquence d'entrée.

L'interface d'entrée (2) supporte des données d'entrées qui peuvent être des images vidéo, de différents formats, à résolution variables allant au delà du domaine de la haute définition et à débit variable allant au delà de 100 images par seconde.

Les données d'entrées peuvent être traitées en vue d'amélioration, d'adaptation et de correction.

Les données d'entrées peuvent être lues à partir de la SDRAM selon un débit (images par seconde) sans avoir besoin de programmer le rapport entre le débit de sortie et le débit d'entrée.

Dans le cas de la vidéo, la résolution de l'image peut rester inchangée. Le système inventé permet de modifier d'une manière automatique et transparente le débit du flux sortant par rapport au débit du flux entrant afin de l'adapter à la bande passante de l'interface de destination qui assure l'acheminement et la transmission de la donnée acquise. Le bloc réalisé donne la possibilité, de faire la conversion de débit d'images ou de données sans aucun paramétrage préalable. L'utilisateur n'a pas besoin de programmer la valeur du débit entrant, ni celle du débit sortant. Il n'a pas besoin non plus de donner la valeur de la résolution d'image d'entrée ou la valeur du rapport entre le débit entrant et le débit sortant. Le système fait la conversion automatiquement et d'une manière transparente pour l'utilisateur. L'adaptation du débit qui se base sur l'opération de « Dropping » d'images non lues, en cas de diminution de débit (Downsampling) ou sur l'opération du « Repeating » d'images non disponibles dans le cas d'augmentation de débit d'images (Upsampling), se fait d'une manière dynamique et transparente. Il est possible de changer le débit d'images d'entrée et le débit d'images de sortie en cours du vidéo streaming, le système s'adapte automatiquement et la conversion de débit d'image se fait sans aucune intervention de l'utilisateur. Ceci est réalisé à travers la vérification en permanence de l'état de remplissage ou de vacuité des FIFOS d'écriture et de lecture qui assurent la synchronisation de la destination par rapport à la source, et en utilisant le système d'arbitrage qui décide d'une manière automatique et transparente pour l'utilisateur de la zone mémoire à accéder en lecture ou en écriture, ainsi que des images à relire ou à écraser selon le flux permis.

La FSM contenue dans le système gère les deux opérations d'écriture et de lecture à la fois.

Le mécanisme d'arbitrage est composée de deux processus séparés un pour calculer la valeur du pointeur d'écriture vers la page à écrire, et un pointeur de lecture vers la page à lire.

Le mécanisme d'arbitrage calcule la valeur des pointeurs d'écriture et de lecture à chaque cycle d'horloge système.

1.4. DESCRIPTION DES DESSINS ET MODES DE REALISATION

1.4.1. DESCRIPTION DES DESSINS

DESCRIPTION DU DESSIN DU FRC (Fig 1)

Le dessin de la page 6 décrit la Frame Rate Converter.

Le module global FRC (1) contient les sous-modules suivants :

- **Interface d'entrée (2) :** Ce bloc fait la conversion de format RAW vers RGB, vers YUV ou inversement. Il assure aussi l'amélioration de la qualité de l'image d'entrée à travers des algorithmes dédiés au traitement d'image tels que le debblurring, détection de mouvement, élimination du background, estimation de la vitesse, amélioration de la sharpness, de la luminosité, du contraste,...
- **Arbitre (3) :** Ce bloc fait le choix de la zone mémoire qu'on va accéder, il décide aussi de l'opération à faire écriture ou lecture.
- **FSM (4) :** Ce bloc s'appelle un Machine d'Etats Finis (Finite State Machine). Le rôle de ce bloc est de générer les bonnes adresses mémoire à accéder en lecture ou en écriture, il fait aussi le cadencement des opérations effectuées par le FRC.
- **W_FIFO(51) :** C'est une FIFO standard qui sert pour emmagasiner temporairement les parties des images entrantes avant de les stocker dans la SDRAM. Elle permet aussi d'assurer la synchronisation entre le domaine d'horloge du senseur et le domaine d'horloge de la SDRAM.
- **R_FIFO(52) :** C'est une FIFO standard qui sert pour emmagasiner temporairement les parties des images sortantes qui sont lues de la SDRAM. Elle permet aussi d'assurer la synchronisation entre le domaine d'horloge de la destination (USB, PCIExpress,...) et le domaine d'horloge de la SDRAM.
- **Memory Controller(6) :** C'est une IP standard fournie par un tiers qui sert à générer les signaux standard de la SDRAM tels que CS, WE, ADDR, CAS, RAS,...
- **Interface de sortie (8) :** C'est l'étage de sortie du système qui sert à adapter le format des données au support de transmission.

DESCRIPTION DU DESSIN LA FSM (Fig 2)

Le dessin de la page 7 décrit la FSM (Machine d'Etats Finis). Cette FSM comporte les états suivants :

IDLE(1): Etat de repos

SEND_PCH2(2): Envoi de la première commande de precharge

WAIT_ACK_PCH2(3): Attente de l'acknowledge du precharge

READY_FOR_RW(4): Attente de la commande read ou write

SEND_WRITE_CMD(5): Envoi de la commande write
WAIT_ACK_WRITE(6): Attente de l'acknowledge write
WAIT_RCD_2_WRITE(7): attente de RCD -2 cycle après reception de l'acknowledge
WRITE_BURST(8): Début d'écriture du burst
WRITE_SEND_PCH1(9): Envoi de la deuxième commande de precharge pour arrêter le burst
WRITE_WAIT_ACK_PCH1(10): Attente de l'acknowledge du precharge
SEND_READ_CMD(11): Envoi de la commande read
WAIT_ACK_READ(12): Attente de l'acknowledge read
READ_WAIT_RCD_CL_2(3): Attente de RCD + CL + 2 cycles après réception de l'acknowledge
READ_BURST(14): Début de lecture de burst
READ_SEND_PCH1(15): Envoi de la commande de precharge pour terminer le burst
READ_WAIT_ACK_PCH1(16): Attente de l'acknowledge sur le precharge
READ_WAIT_CL_2(17): Comptage de CL - 1 + 2 cycles avant arrêt de burst

1.4.2. MODES DE REALISATION

Le FRC se compose d'une partie de contrôle combinatoire qui génère les signaux et flags (fin de frame, fin de comptage de latences, calcul d'adresses,...), d'un ensemble de processus synchrones pour calculer les valeurs de compteurs des mots d'une rafale appelée Burst, de frames, de pixels, de latences,... de deux processus combinatoires pour faire l'arbitrage et le choix de page à écrire et de page à lire, et de deux machine d'états de Moore. La première machine d'état sert faire l'initialisation de la SDRAM. La deuxième machine d'état sert à générer les commandes lecture ou écriture ou precharge ainsi que les requêtes vers les FIFOs d'écriture ou de lecture.

La machine d'état qui gère l'accès d'écriture ou de lecture à la SDRAM est une FSM de type Moore. Elle détermine l'action à effectuer. Par exemple écrire une rafale (Burst) dans la SDRAM ou de lire une rafale (Burst) à partir de la SDRAM ou attendre ou de faire une précharge. L'état suivant est choisi en fonction de l'état actuel et des entrées qui décrivent le statu des FIFOs, comme (l'état de plénitude de la FIFO d'écriture et l'état de vacuité de la FIFO lecture. ...)

Elle se compose de 17 états qui sont: IDLE, SEND_PCH2, WAIT_ACK_PCH2, READY_FOR_RW, SEND_WRITE_CMD, WAIT_ACK_WRITE, WAIT_RCD_2_WRITE, WRITE_BURST, WRITE_SEND_PCH1, WRITE_WAIT_ACK_PCH1,

SEND_READ_CMD, WAIT_ACK_READ, READ_WAIT_RCD_CL_2, READ_BURST, READ_SEND_PCH1, READ_WAIT_ACK_PCH1, et READ_WAIT_CL_2.

Le rôle des différents états est le suivant :

IDLE: Attente de démarrage

SEND_PCH2: Envoi de la première commande de precharge

WAIT_ACK_PCH2: Attente de l'acknowledge du precharge

READY_FOR_RW: Attente de la commande read ou write

SEND_WRITE_CMD: Envoi de la commande write

WAIT_ACK_WRITE: Attente de l'acknowledge write

WAIT_RCD_2_WRITE: attente de RCD -2 cycle après reception de l'acknowledge

WRITE_BURST: Début d'écriture du burst

WRITE_SEND_PCH1: Envoi de la deuxième commande de precharge pour arrêter le burst

WRITE_WAIT_ACK_PCH1: Attente de l'acknowledge du precharge

SEND_READ_CMD: Envoi de la commande read

WAIT_ACK_READ: Attente de l'acknowledge read

READ_WAIT_RCD_CL_2: Attente de $RCD + CL + 2$ cycles après réception de l'acknowledge

READ_BURST: Début de lecture de burst

READ_SEND_PCH1: Envoi de la commande de precharge pour terminer le burst

READ_WAIT_ACK_PCH1: Attente de l'acknowledge sur le precharge

READ_WAIT_CL_2: Comptage de $CL - 1 + 2$ cycles avant arrêt de burst

La machine d'état d'initialisation de la SDRAM sert à paramétrer les valeurs de la mémoire telles que la latence, le mode d'accès, la longueur du burst...

Elle contient les états suivants : ST_PCH, ST_LOAD_MODE, ST_LOAD_REG2, ST_LOAD_REG1 et ST_READY.

Le rôle des différents états est :

ST_PCH: effectue une precharge

ST_LOAD_MODE: Sert à programmer le registre LMR pour paramétrer la latence pour accéder à une colonne (CAS latency), le type de burst,...

ST_LOAD_REG1: Sert à programmer le registre REG1 pour paramétrer la longueur du burst, le type d'accès, le temps de la latence entre l'ouverture d'une page et l'accès d'une colonne (RAS to CAS Latency),...

ST_LOAD_REG2: Sert à programmer le registre REG2 pour configurer le temps entre deux rafraichissement de la SDRAM

ST_READY: Mise en état de disponibilité

Ce bloc fonctionne avec quatre horloges différentes: l'horloge système, l'horloge d'entrée, l'horloge SDRAM et l'horloge de sortie. L'horloge d'entrée provient du flux vidéo en entrée (capteurs, caméras ...) l'horloge SDRAM est l'horloge utilisée par l'SDRAM externe. L'horloge de sortie est l'horloge utilisée par la destination pour lire les images stockées dans la mémoire SDRAM. L'architecture de ce bloc est basée sur deux FIFOs, une FIFO d'écriture et une FIFO de lecture. La FIFO d'écriture est utilisée comme une interface entre l'entrée (appareil photo, capteur, Modem, Tuner ...) et entre la SDRAM. Elle est utilisée pour recevoir des images vidéo d'entrée avec l'horloge d'entrée et de les envoyer sous forme de rafales à la SDRAM avec l'horloge SDRAM. La FIFO de lecture est utilisée comme une interface entre la SDRAM et la destination (USB, modem, sortie vidéo). Les images qui sont déjà stockées dans la mémoire SDRAM sont lues sous forme de trames à partir de la SDRAM avec l'horloge SDRAM, elles sont ensuite stockées temporairement dans la FIFO de lecture, et renvoyées vers la destination avec l'horloge de sortie ... Le bloc contient deux arbitres, un arbitre pour l'écriture et un arbitre pour la lecture, qui déterminent la zone mémoire concernée par l'opération. La SDRAM externe est vue par le bloc comme un ensemble de trois zones de mémoire séparées: la zone 1, zone 2 et zone 3. L'arbitre d'écriture détermine l'emplacement du pointeur d'écriture (zone 1 ou zone 2 ou 3), l'arbitre de lecture détermine l'emplacement du pointeur de lecture (zone 1 ou zone 2 ou 3). L'arbitre calcule la position du pointeur d'écriture et le pointeur de lecture chaque cycle d'horloge du système (l'horloge système est l'horloge interne utilisée dans tous les processus synchrones du bloc, elle est plus rapide à la plus rapide des trois horloges qui sont l'horloge d'entrée, l'horloge SDRAM et l'horloge de sortie) ... L'arbitre d'écriture calcule la position lors du prochain cycle du pointeur d'écriture en fonction de la position actuelle du pointeur de lecture (afin de ne pas effacer une image qui est en cours de lecture) et aussi en fonction de la position actuelle du pointeur d'écriture lui-même (en vue de continuer à écrire l'image courante dans la même zone mémoire ou pour passer à la zone mémoire appropriée afin d'écrire une nouvelle image). L'arbitre de lecture calcule la position lors du prochain cycle du pointeur de lecture en fonction de la position du pointeur d'écriture (afin d'éviter la lecture d'une zone de mémoire qui est en cours d'écriture) et en fonction de la position actuelle du pointeur de lecture afin de continuer la lecture de l'image courante ou pour passer à la zone mémoire appropriées afin de lire une nouvelle image. Lorsque le flux vidéo d'entrée est plus rapide que le flux vidéo de sortie, quelques images vidéo qui sont stockées dans SDRAM doivent être effacées, c'est à dire la zone mémoire dans laquelle

L'image à effacer est stockée doit être réécrite avec une nouvelle image, même si le contenu de cette zone mémoire n'a pas encore été lu. Les cas où une zone de mémoire est effacée (réécrite avant de lire son ancien contenu) sont les suivants: Premier cas: la zone 1 est en cours d'écriture et la zone 2 est en cours de lecture Dans ce cas, la zone 1 doit être réécrite. En effet, les seules zones d'écriture possible sont la zone 1 ou la zone 3. Si nous choisissons d'écrire la zone 3, le pointeur de lecture va lire, puis la zone 3 et après il va lire 1. Le problème qui se pose est que le contenu de la zone 3 sera plus récent que le contenu de la zone 1. Donc la réécriture doit être effectuée dans la zone 1 afin d'avoir une séquence chronologique correcte (zone 3: plus ancienne , puis zone 1: plus récente) (l'ordre de lecture est toujours la zone 1, zone 2, zone 3, zone 1, zone 2, la zone 3. Si un skip doit être effectué, l'ordre doit être respecté, par exemple la zone 1 zone 3 zone 1 zone 2 zone 1 zone 2 zone 3 zone 1). Le deuxième cas: La zone 2 est en cours d'écriture et de la zone 3 est en cours de lecture. Dans ce cas, la réécriture sera effectuée dans la zone 2. La raison est la même que dans le premier cas. Le troisième cas est: zone 3 est en cours d'écriture et de la zone 1 est en cours de lecture. Dans ce cas c'est la zone 3 qui sera réécrite.

1.5. APPLICATIONS INDUSTRIELLES POSSIBLES

Le bloc réalisé sert à implémenter sur FPGA un contrôleur ou un système d'acquisition et de traitement de flux vidéo. Parmi les applications possibles on trouve la gestion du flux vidéo provenant des caméras numériques et d'adapter ce flux au débit de l'interface de transfert. Le FRC permet d'afficher des vidéos HD et UHD à des débits d'images adaptables. Ceci est notamment utile pour les systèmes électroniques d'acquisition et de traitement vidéo telles que les caméscopes, les caméras numériques, les caméras industrielles, la vidéo surveillance...

2. REVENDEICATIONS

Les revendications essentielles liées à ce brevet sont :

- 1) Système d'acquisition hardware numérique FRC (1) conçu pour recevoir un flux de données numériques (Vidéo, Audio, ou autres,...) provenant d'une source donnée (Caméra, Capteur, émetteur ...) qui dispose d'un domaine d'horloge et d'une bande passante donnée , et adapter la qualité et le débit de ces données reçues avant de les retransmettre vers une destination asynchrone par rapport à la source. La destination peut posséder un domaine d'horloge et un débit différent de celui de la source. Le système inventé permet d'assurer la synchronisation automatique et transparente d'une manière dynamique entre le flux de données entrantes et le flux de données sortantes.

Le système comprend :

- Une interface d'entrée (2) qui fait l'acquisition et le traitement numériques des données numériques avec une fréquence d'horloge d'entrée variable.
- Une interface de sortie (8) numérique qui fait l'adaptation du format des données et l'envoi vers la destination avec une fréquence d'horloge de sortie variable et différente de la fréquence d'entrée.
- Une FIFO d'entrée (51) pour faire le passage du domaine d'horloge de l'interface d'entrée vers le domaine d'horloge de la SDRAM. L'horloge d'entrée de cette FIFO est l'horloge de l'interface d'entrée. L'horloge de sortie de cette FIFO est l'horloge de l'SDRAM. Elle sert pour stocker momentanément une partie de la donnée d'entrée. La communication entre la FIFO d'entrée et la SDRAM se fait à l'aide de rafales (Burst) de longueur variables génériques. Les données lues de cette FIFO sont envoyées vers la SDRAM.
- Une FIFO de sortie (52) pour faire le passage du domaine d'horloge de la SDRAM vers le domaine d'horloge de l'interface de sortie. L'horloge d'entrée de cette FIFO est l'horloge SDRAM. L'horloge de sortie est l'horloge de l'interface de sortie. Elle sert pour stocker momentanément une partie de la donnée de sortie. La communication entre la FIFO de sortie et la SDRAM se fait à travers des rafales (Bursts) de longueur variable générique. Les données lues de la SDRAM sont provisoirement stockées dans cette FIFO avant d'être envoyées vers l'interface de sortie.
- Une FSM (4) (Machine d'états à 19 états) qui détermine si l'opération à effectuer dans la SDRAM est une rafale d'écriture ou bien une rafale de

lecture. Cette FSM organise le passage entre les différentes étapes et notamment les conditions de passage d'une opération d'écriture de Burst vers une opération de lecture de Burst et vice-versa. L'opération d'écriture d'un Burst dans la SDRAM est prioritaire sur l'opération de lecture d'un Burst depuis la SDRAM.

- Un système d'arbitrage constitué de deux processus combinatoires (3) un processus pour l'écriture et un processus pour la lecture. Le processus d'écriture calcul le pointeur d'écriture vers l'une des trois zones (Page1 ou page2 ou page3) mémoires choisies pour l'écriture. La prochaine valeur calculée du pointeur d'écriture dépend de la valeur courante du pointeur d'écriture, de la valeur courante du pointeur de lecture et de l'avancement du pointeur d'écriture dans une page en cours d'écriture, qui peut correspondre à la fin de la page, à l'intérieur de la page ou bien au cas où aucune page n'a été écrite encore.

Le processus de lecture calcul le pointeur de lecture vers l'une des trois zones (Page1 ou page2 ou page3) mémoires choisies pour la lecture. La prochaine valeur calculée du pointeur de lecture dépend de la valeur courante du pointeur de lecture, de la valeur courante du pointeur d'écriture et de l'avancement du pointeur de lecture dans une page en cours de lecture, qui peut correspondre à la fin de la page, à l'intérieur de la page ou bien au cas où aucune page n'a été écrite lue.

- Un contrôleur de mémoire standard (6) qui permet de générer les différents signaux relatifs à une interface SDRAM standard.
 - Une interface de sortie (8) numérique avec une fréquence de sortie différente de la fréquence d'entrée.
- 2) Système selon la revendication 1, caractérisé en ce que l'interface d'entrée (2) supporte des données d'entrées qui peuvent être des images vidéo, de différents formats, à résolution variables allant au delà du domaine de la haute définition et à débit variable allant au delà de 100 images par seconde.
 - 3) Système selon la revendication 1, caractérisé en ce que les données d'entrées peuvent être traitées en vue d'amélioration, d'adaptation et de correction.
 - 4) Système selon la revendication 1, caractérisé en ce que, dans le cas de la vidéo, la résolution de l'image peut rester inchangée.

- 5) Système selon la revendication 1, caractérisé en ce que, dans le cas de la vidéo, le système inventé permet de modifier d'une manière automatique et transparente le débit du flux sortant par rapport au débit du flux entrant afin de l'adapter à la bande passante de l'interface de destination qui assure l'acheminement et la transmission de la donnée acquise. Le bloc réalisé donne la possibilité, de faire la conversion de débit d'images ou de données sans aucun paramétrage préalable. L'utilisateur n'a pas besoin de programmer la valeur du débit entrant, ni celle du débit sortant. Il n'a pas besoin non plus de donner la valeur de la résolution d'image d'entrée ou la valeur du rapport entre le débit entrant et le débit sortant. Le système fait la conversion automatiquement et d'une manière transparente pour l'utilisateur. L'adaptation du débit qui se base sur l'opération de « Dropping » d'images non lues, en cas de diminution de débit (Downsampling) ou sur l'opération du « Repeating » d'images non disponibles dans le cas d'augmentation de débit d'images (Upsampling), se fait d'une manière dynamique et transparente. Il est possible de changer le débit d'images d'entrée et le débit d'images de sortie en cours du vidéo streaming, le système s'adapte automatiquement et la conversion de débit d'image se fait sans aucune intervention de l'utilisateur. Ceci est réalisé à travers la vérification en permanence de l'état de remplissage ou de vacuité des FIFOs d'écriture et de lecture qui assurent la synchronisation de la destination par rapport à la source, et en utilisant le système d'arbitrage qui décide d'une manière automatique et transparente pour l'utilisateur de la zone mémoire à accéder en lecture ou en écriture, ainsi que des images à relire ou à écraser selon le flux permis.
- 6) Système selon la revendication 1, caractérisé en ce qu'une seule FSM gère les deux opérations d'écriture et de lecture.
- 7) Système selon la revendication 1, caractérisé en ce que l'arbitrage est composée de deux processus séparés un pour calculer la valeur du pointeur d'écriture vers la page à écrire, et un pointeur de lecture vers la page à lire.
- 8) Système selon la revendication 1, caractérisé en ce que le mécanisme d'arbitrage calcule la valeur des pointeurs d'écriture et de lecture à chaque cycle d'horloge système.

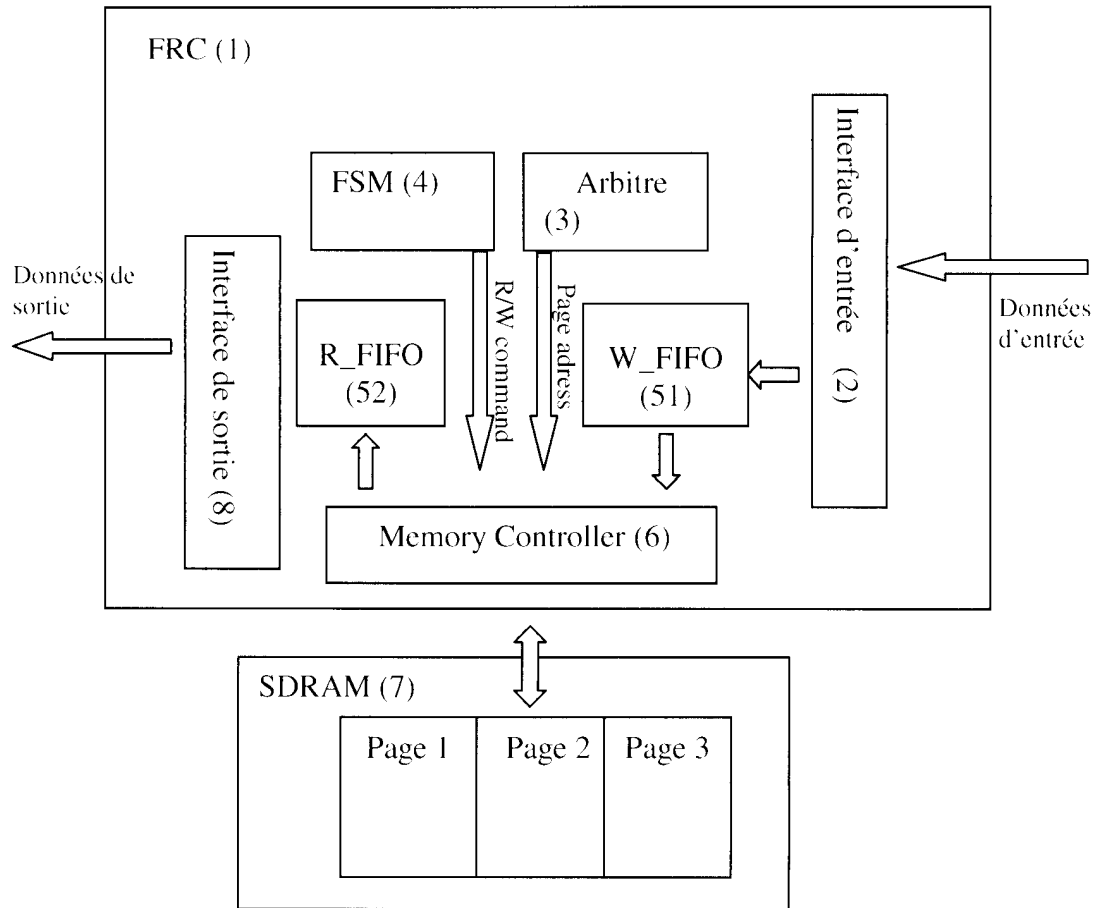
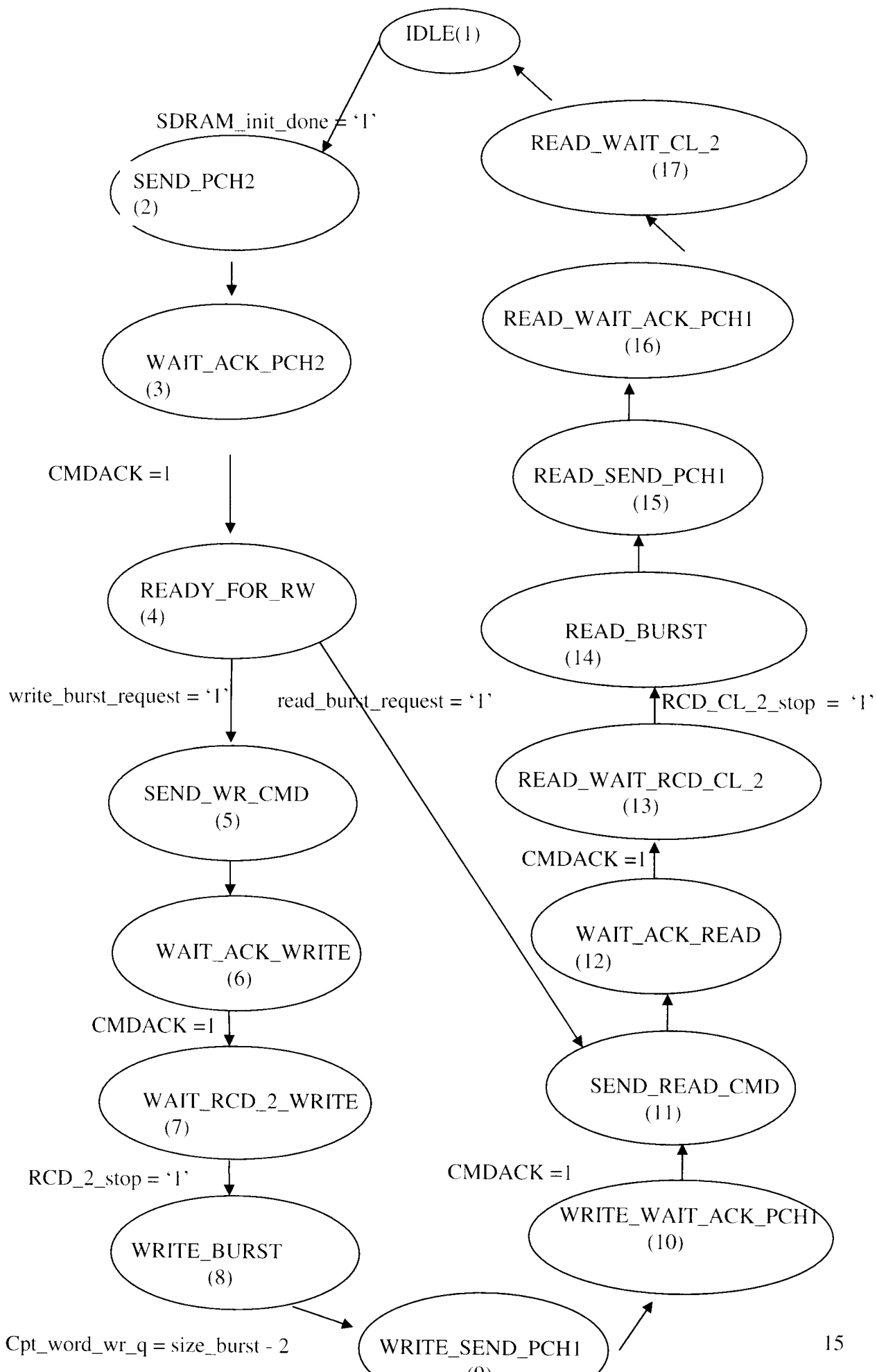


FIG 1

NK

FIG 2



NK