

ECOLE NATIONALE POLYTECHNIQUE



DEPARTEMENT D'ELECTRONIQUE
LABORATOIRE SIGNAL ET COMMUNICATIONS

THESE

POUR OBTENIR LE GRADE DE

DOCTEUR EN ELECTRONIQUE

Présentée par

Mountassar MAAMOUN

Ingénieur – Magister USDB

THEME

**NOUVELLES ARCHITECTURES D'INTERFAÇAGE
GRAPHIQUE DANS LES SYSTEMES A
MICROPROCESSEUR**

Devant le jury composé de :

M. Mohamed TADJINE	Professeur à l'ENP	Président
M. Daoud BERKANI	Professeur à l'ENP	Rapporteur
M. Abdelhamid MERAGHNI	Maître de Conférences à l'ENS-Kouba	Examineur
M. Abderrezak GUESSOUM	Professeur à l'Université de Blida	Examineur
M. Hassen SALHI	Professeur à l'Université de Blida	Examineur
M ^{me} . Latifa HAMAMI	Maître de Conférences à l'ENP	Examinatrice

DEDICACE

À ma femme Ghania

ملخص

الأطروحة تمثل إسهاما في تقنيات الإظهار الرقمية المتخصصة و العاملة مع الأجهزة المعتمدة على "الميكروبروسيسور" و الحاسوب. خلال العمل تم التركيز، من جهة على تطوير تقنيات "الميكروبروسيسور"، و من جهة أخرى تم التركيز آليات المخاطبة الإلكترونية بين أنظمة الإظهار و على تصميم هندسة الكترونية جديدة لإعادة تشكيل الصور الرقمية. العمل المنجز يمثل حلا لتصميم أنظمة إظهار عاملة مع أحدث تقنيات المعالجة الآنية للصور، و يمكن بلورتها بالمنطق الرقمي المبرمج. جزء من عملنا المنجز تم الإشارة إليه في براءة اختراع سجلت الأمريكية في 20 سبتمبر 2005 و مستعملة من طرف شركة Sony® بالولايات المتحدة. الكلمات المفتاحية: تقنيات الإظهار الرقمية، تقنيات المخاطبة الإلكترونية، المعالجة الآنية للصور، المنطق الرقمي المبرمج

SUMMARY

This thesis stands as a contribution to the graphic interfacing in microprocessor-based systems using new interfacing techniques and an advanced video generating architecture. This architecture is a solution for the real-time image processing systems. In the interfacing part, we have developed the Extended Physical Addressing, the Fast Physical Addressing, the Accelerated Physical Addressing and the Advanced Physical Addressing. The video generating architecture investigates our simultaneous video memory read-write system. This latter is ensured by a split of the hardware video memory in separate capacities. The proposed systems are implemented with the programmable logic. A part of our work has been referred to in « DIGITAL IMAGE TRANSFER CONTROLLER » patent, used by Sony® and recorded in United States Patent on September 20, 2005.

Keywords: Graphic interfacing, microprocessor-based system, real-time image processing, software/hardware system.

RESUME

Cette thèse présente une contribution à l'interfaçage graphique dans les systèmes à microprocesseur. Nous nous sommes posé comme objectif de concevoir de nouveaux systèmes d'adressage physique et de développer de nouveaux principes de reproduction vidéo. Dans le contexte de l'interfaçage nous avons développé l'Adressage Physique Etendu, l'Adressage Physique Accéléré, l'Adressage Physique Rapide et l'Adressage Physique Avancé. Pour le principe de la reproduction d'images, nous avons élaboré une nouvelle architecture de reconstruction d'images basée sur la décomposition physique de la mémoire vidéo pour assurer un accès simultané de lecture/écriture. Les systèmes proposés sont réalisables avec la logique programmable et constituent une solution pour les systèmes d'affichages avec un traitement d'images en temps réel. Une partie des travaux réalisés est référenciée dans le brevet « DIGITAL IMAGE TRANSFER CONTROLLER ». Le brevet est exploité par Sony® et enregistré dans U.S Patent le 20 septembre 2005.

Mots clés : Interfaçage graphique, systèmes à microprocesseur, traitement d'images en temps réel, système logiciel/matériel.

Table des matières

Table des matières	1
Remerciements	4
Liste des abréviations	5
Table des figures	6
Introduction	8
Chapitre 1	
Interfaces d'entrées-sorties et bus d'extension	11
1. Introduction	12
2. Caractéristiques d'un bus	13
2.1. Identité du bus	14
2.2. Usage du bus	14
2.3. Paramètres physiques et électriques	14
2.4. Caractéristiques fonctionnelles	15
2.5. Caractéristiques dynamiques	15
3. Modèles de Bus	16
3.1. Bus de composant	16
3.2. Bus de processeur	16
3.3. Bus de mémoire	17
3.4. Bus d'une interface	17
4. Interfaces d'extension	18
4.1. Bus AT ou ISA	19
4.1.1. Utilisation du Bus ISA	20
4.1.2. Signification des principaux signaux du bus ISA	21
4.2. Bus PCI	22
4.2.1. Description des signaux	24
4.2.2. Transfert de données	24
4.3. Bus PCMCIA	26
4.4. Bus graphique AGP	27
4.4.1. Architecture	27
4.4.2. Transfert de données	29
4.5. Bus PCI Express	30

5. Ports et Connecteurs PC	32
5.1. Bus SCSI	32
5.2. IDE ou ATA	32
5.3. Serial ATA	34
5.4. Port série RS-232	34
5.5. Port parallèle	35
5.6. Port USB	36
5.7. Port FireWire (IEEE 1394)	38
6. Conclusion	39
Chapitre 2	
Nouvelles architectures logicielle/matérielle d'interfaçage	40
1. Introduction	41
2. Adressage Physique Etendu	42
2.1. Principe de l'Adressage Physique Etendu.....	42
2.2. Adressage Physique Etendu d'ordre supérieur.....	45
2.2.1. Système d'ordre deux	45
2.2.2. Système d'ordre n	49
2.3. Compatibilité de l'Adressage Physique Etendu	51
3. Adressage Physique Accéléré	52
3.1. Principe de l'Adressage Physique Accéléré	52
3.2. Compatibilité de l'Adressage Physique Accéléré	55
4. Adressage Physique Rapide	56
4.1. Principe de l'Adressage Physique Rapide	56
4.2. Compatibilité de l'Adressage Physique Rapide	60
5. Adressage Physique Avancé	61
6. Conclusion	63
Chapitre 3	
Logique programmable et implémentation de l'interfaçage	64
1. Introduction	65
2. Processeur	65
3. Technologies de programmation	67
3.2. Fusible	67
3.3. Antifusible	67
3.4. EPROM	68
3.5. EEPROM/Flash	69
3.6. SRAM	70
4. Circuits logiques programmables	71
4.1. SPLD	71
4.2. CPLD	74
4.3. FPGA	77
5. Implémentation de l'Adressage Physique Etendu	84
5.1. Implémentation schématique de l'Adressage Physique Etendu	84
5.2. Implémentation VHDL de l'Adressage Physique Etendu	84

6. Implémentation de l'Adressage Physique Rapide	87
6.1. Implémentation schématique de l'Adressage Physique Rapide.....	87
6.2. Implémentation VHDL de l'Adressage Physique Rapide	89
7. Conclusion	93
Chapitre 4	
Nouvelles architectures de génération d'images	94
1. Introduction	95
2. Construction d'image dans les cartes vidéo	96
3. Video RAM et Window RAM	97
4. Génération d'Images avec l'Adressage Physique Etendu	98
4.1. Unité de Stockage	99
4.2. Unité de Lecture	101
4.3. Emission d'image	102
4.4. Traitement Numérique de base réalisé	103
4.4.1. Requantification	103
4.4.2. Changement de la définition horizontale	106
4.4.3. Addition des impulsions de synchronisation	107
5. Génération d'Images avec Lecture/Ecriture Simultanée	108
5.2. Construction d'Image avec Accès Simultané à Deux Niveaux	108
5.2.1. Partie Matérielle	108
5.2.2. Partie Logicielle	109
5.3. Construction d'Image avec Accès Simultané à Plusieurs Niveaux	110
5.3.1. Partie Matérielle	110
5.3.2. Partie Logicielle	111
6. Stockage et Conversion	112
6.1. Conversion numérique analogique	112
6.2. Stockage et mémoire vidéo	114
7. Conclusion	117
Conclusion	118
Bibliographie	120
Annexes	125

Remerciements

Je tiens à adresser mes sincères remerciements à Monsieur Mohamed TADJINE, Professeur à Ecole Nationale Polytechnique de l'honneur qu'il m'a fait en acceptant de présider ce jury.

Je tiens également à exprimer ma profonde gratitude et ma reconnaissance envers Monsieur Daoud BERKANI, Professeur à l'ENP, directeur de cette thèse. Ses idées, ses conseils et ses critiques m'ont été d'une aide précieuse pour mener ce travail à bien. Au-delà de l'aspect scientifique de nos discussions, j'ai été particulièrement sensible à ses qualités humaines et à l'excellent climat relationnel qu'il a su établir entre nous.

J'adresse à Madame Latifa HAMAMI, Maître de Conférences à l'ENP, à Messieurs Abdelhamid MERAGHNI, Maître de Conférences et directeur de l'Ecole Normale Supérieure de Kouba, Abderrezak GUESSOUM, Professeur à l'Université de Blida et Hassen SALHI, Maître de Conférences à l'Université de Blida, mes plus vifs remerciements pour l'intérêt qu'ils ont manifesté pour mon travail et pour avoir accepté la charge d'examineurs.

Je remercie vivement Monsieur Larbi TALBI, Professeur à l'Université du Québec en Outaouais de m'avoir accueilli dans son laboratoire au département d'informatique et d'ingénierie de l'UQO.

Je suis très reconnaissant à Monsieur Abdelhalime BENBELKACEM pour son aide précieuse, pour sa disponibilité et pour ses conseils.

Mes remerciements vont également à tous les collègues du Laboratoire LSIC de l'ENS de Kouba. Je tiens à remercier tout particulièrement le directeur du laboratoire, Monsieur Djamel KENDIL, Maître de conférences à l'ENS de Kouba, pour les moyens qu'il a mis à ma disposition.

Mes sincères remerciements vont aussi aux personnes qui m'ont aidé en contribuant, de près ou de loin, à l'aboutissement de ce travail. Qu'ils trouvent dans ce mémoire une trace de ma reconnaissance. Je cite en particulier mes amis doctorants et jeunes docteurs des laboratoires LSC (ENP), LATSI (USDB) et LSIC (ENS).

Ces dernières lignes sont pour mes parents, ma famille ainsi que mes amis. Je tiens ici à leur exprimer toute ma reconnaissance pour tout le soutien et tous les encouragements qu'ils ont su me donner tout au long de ce travail et jusqu'au dernier instant. Que ceux que j'oublie ici veuillent bien me pardonner.

Liste des abréviations

AGP	Accelerated Graphics Port
ASIC	Application Specific Integrated Circuit
BIOS	Basic Input Output System
CLB	Configurable Logic Block
CPLD	Complex Programmable Logic Device
DAC	Digital to Analog Converter
DCM	Digital Clock Manager Blocks
DMA	Direct Memory Access
DSP	Digital Signal Processor
EPROM	Erasable Programmable Read Only Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
EIA	Electronic Industries Association
FPGA	Field Programmable Gate Array
FSB	Front Side Bus
ICR	In Circuit Reconfigurability
IDE	Integrated Disk Electronics
ISA	Industry Standard Architecture
LUT	Look-up table
MPEG	Moving Picture Experts Group
MIPS	Millions d'Instructions Par Seconde
MUX	Multiplexeurs
PAL	Programmable Array Logic
PCI	Peripheral Component Interconnect
PCMCIA	Personal Computer Memory Card Interface Association
PLA	Programmable Logic Array
PLD	Programmable Logic Device
PROM	Programmable Read Only Memory
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer, microprocesseur à jeu d'instructions réduit)
ROM	Read Only Memory
SCSI	Small Computer System Interface
SPLD	Simple Programmable Logic Device
SRAM	Static Random Access Memory
USB	Universal Serial Bus
VESA	Video Electronics Standards Association
VGA	Video Graphics Array
VHDL	VHSIC (Very High Speed Integrated Circuit) Hardware Description Language
VRAM	Video Random Access Memory
WRAM	Window Random Access Memory

Table des figures

Fig. 1.1. Structure générale d'un ordinateur.....	12
Fig. 1.2. Le chronogramme d'accès aux données.....	16
Fig. 1.3. Architecture de l'interface.	18
Fig. 1.4. Bus local et bus d'extension type ISA.....	20
Fig. 1.5. Principe d'utilisation du bus ISA.....	21
Fig. 1.6. Liste des signaux du bus PCI suivant la norme.....	23
Fig. 1.7. Schéma bloc d'une architecture PCI.....	23
Fig. 1.8. Opération de base en écriture.....	25
Fig. 1.9. Opération de base en lecture	25
Fig. 1.10. Opération d'accès exclusif sur PCI.....	26
Fig. 1.11. Different Corelogic Architectures.....	28
Fig. 1.12. Schéma bloc avec chipset 82845G de Intel®.....	28
Fig. 1.13a. Concept de base du transfert A.G.P 1x.	29
Fig. 1.13b. Concept de base du transfert A.G.P 2x.	29
Fig. 1.14. Interconnexion des périphériques PCI Express.....	31
Fig. 1.15. Schéma bloc avec chipset 82915P de Intel®.....	31
Fig. 1.16. L'interface parallèle	36
Fig. 2.1. Principe de l'interfaçage des périphériques.....	41
Fig. 2.2. Synoptique de l'Adressage Physique Etendu du premier ordre.....	42
Fig. 2.3. Organigramme de la procédure logicielle de l'Adressage Physique Etendu du premier ordre	44
Fig. 2.4. Le chronogramme des transactions de l'Adressage Physique Etendu du premier ordre.....	45
Fig. 2.5. Synoptique de l' Adressage Physique Etendu du deuxième ordre.....	46
Fig. 2.6. Organigramme de la procédure logicielle de l'Adressage Physique Etendu d'ordre deux.....	47
Fig. 2.7. Le chronogramme des transactions de l'Adressage Physique Etendu du deuxième ordre.....	48
Fig. 2.8. Synoptique de l' Adressage Physique Etendu d' Ordre n	50
Fig. 2.9. Compatibilité de l'Adressage Physique Etendu.....	51
Fig. 2.10. Synoptique de l'Adressage Physique Accéléré.....	52
Fig. 2.11. Le chronogramme des transactions de l'Adressage Physique Accéléré.....	54
Fig. 2.12. Compatibilité avec l'Adressage Physique Etendu.....	55
Fig. 2.13. Compatibilité avec les systèmes parallèles.	56
Fig. 2.14. Synoptique de l'Adressage Physique Rapide du premier ordre.....	57
Fig. 2.15. Les phases de la génération d'adresses.....	58
Fig. 2.16. Le chronogramme des transactions de l'Adressage Physique Rapide.....	59
Fig. 2.17. Compatibilité avec l'Adressage Physique Etendu.....	60
Fig. 2.18. Compatibilité avec la communication parallèle.....	61
Fig. 2.19. Synoptique de l'Adressage Physique Avancé.....	62
Fig. 3.1. Architecture de base d'un système à microprocesseur.....	66
Fig. 3.2. Un circuit contenant 4 fusibles non programmés, puis le circuit résultant d'une programmation.	67
Fig. 3.3. Un circuit contenant 4 antifusibles non programmés, puis le circuit résultant d'une programmation..	67
Fig. 3.4. Architecture d'une ROM.....	68
Fig. 3.5. Un transistor CMOS standard et un transistor EPROM.....	69
Fig. 3.6. Synoptique d'une EEPROM "Xicor" série de 128Kb.	69
Fig. 3.7. (a) Une cellule SRAM à quatre transistors. (b) Une cellule SRAM à six transistors.....	70
Fig. 3.8. (a) Synoptique d'une SRAM typique. (b) Brochage d'une SRAM typique.	70
Fig. 3.9. Une classification des circuits logiques programmables.....	71

Fig. 3.10. L'architecture d'un SPLD.....	72
Fig. 3.11. L'architecture fonctionnelle d'une PROM.....	72
Fig. 3.12. Architectures d'un PLA et d'un PAL.....	73
Fig. 3.13. L'architecture d'un CPLD.....	74
Fig. 3.14. Architecture du XC9500.....	75
Fig. 3.15. Bloc fonctionnel du XC9500.....	76
Fig. 3.16. Connexion du bloc fonctionnel du XC9500.....	77
Fig. 3.17. L'architecture générale du FPGA.....	78
Fig. 3.18. Le CLB d'un XC2000.....	78
Fig. 3.19. Le schéma d'interconnexions d'un XC2000.....	79
Fig. 3.20. Les connexions d'un bloc d'un XC2000.....	79
Fig. 3.21. L'unité fonctionnelle d'un XC6200.....	80
Fig. 3.22. La cellule de base du XC6200.....	81
Fig. 3.23. L'architecture générale de la série Spartan-3E.....	82
Fig. 3.24. L'architecture générale de la série Virtex-II.....	83
Fig. 3.25. L'architecture du "Virtex-5 SXT DSP48E Slice".....	83
Fig. 3.26. Schéma de l'Adressage Physique Etendu avec ECS.....	84
Fig. 3.27. "VHDL instantiation template" de l'Adressage Physique Etendu.....	85
Fig. 3.28. La description VHDL de l'entité et de l'architecture, du décodeur d'adresses dec301_2_3.....	85
Fig. 3.29. Les signaux "Test Bench Waveform" pour l'Adressage Physique Etendu.....	86
Fig. 3.30. Les phases de la conversion données-adresses.....	86
Fig. 3.31. Le résultat de la simulation de l'Adressage Physique Etendu.....	87
Fig. 3.32. Schéma de l'Adressage Physique Rapide avec ECS.....	88
Fig. 3.33. Schéma du Générateur d'Adresses proposé.....	88
Fig. 3.34. "VHDL instantiation template" de l'Adressage Physique Rapide.....	89
Fig. 3.35. La description VHDL de l'entité et de l'architecture du décodeur d'adresses.....	89
Fig. 3.36. Entité du Générateur d'Adresses.....	90
Fig. 2.37. Les signaux "Test Bench Waveform" d'activations de la DMA.....	90
Fig. 2.38. Les phases de la génération d'adresses.....	91
Fig. 2.39. Le résultat de la simulation du Générateur d'Adresses.....	91
Fig. 4.1. Synoptique d'une carte vidéo.....	95
Fig. 4.2. Synchro verticale.....	96
Fig. 4.3. Synchro horizontale.....	96
Fig. 4.4. Schéma bloc de KM4216C256.....	97
Fig. 4.5. Schéma bloc de KM4232W259A.....	98
Fig. 4.6. Synoptique de la production d'images avec l'Adressage Physique Etendu.....	99
Fig. 4.7. Unité de stockage.....	100
Fig. 4.8. Structure de stockage du fichier image.....	100
Fig. 4.9. Synoptique du circuit de lecture.....	101
Fig. 4.10. Organigramme de l'émission des fichiers image.....	102
Fig. 4.11. Organigramme du traitement pour un affichage entrelacé.....	105
Fig. 4.12. Organigramme du traitement pour un affichage progressif.....	106
Fig. 4.13. Principe du changement de la définition horizontale.....	106
Fig. 4.14. Schéma bloc du Système avec Accès Simultané à Deux Niveaux.....	108
Fig. 4.15. Organigramme principal de la partie logicielle.....	109
Fig. 4.16. Schéma bloc du Système avec Accès Simultané à N Niveau.....	110
Fig. 4.17. Variation de W_{FN} avec le nombre des RAM.....	111
Fig. 4.18. Organigramme principal de la partie logicielle du système à plusieurs niveaux.....	112
Fig. 4.19. Schéma synoptique du TDA8702.....	113
Fig. 4.20. Schéma synoptique du μ PD431000.....	114
Fig. 4.21. Cycle de lecture.....	115
Fig. 4.22. Cycle d'écriture.....	115
Fig. 4.23. Schéma synoptique du IS61C1024.....	116
Fig. 4.24. Cycle de lecture.....	116
Fig. 4.25. Cycle d'écriture.....	116

Introduction

Les concepteurs d'architectures de traitement du signal et d'image sont souvent forcés à choisir entre flexibilité et performance à cause des solutions limitées disponibles des circuits de traitement et des systèmes d'interfaçages. Pour le traitement, les microprocesseurs et les processeurs DSP sont peu coûteux, flexibles et peuvent être utilisés dans de nombreuses applications. Cependant, ces processeurs ne peuvent donner qu'une puissance de calcul limitée en temps réel à cause des exécutions séquentielles. Les concepteurs qui désirent de meilleurs résultats doivent alors utiliser plusieurs processeurs en parallèle. D'autre part, les processeurs DSP dédiés et les ASICs (Application Specific Integrated Circuits) offrent de meilleures performances, mais ne sont pas flexibles. Peu de processeurs dédiés existent sur le marché, et les ASICs demandent un temps énorme pour le développement, le test et la production.

Les applications professionnelles d'imagerie demandent un traitement en temps réel ce qui réclame une performance de plusieurs milliers de MIPS (Millions d'Instructions Par Seconde). Les fonctions de traitement numérique du signal sont de plus en plus utilisées dans le traitement des signaux RF pour des applications telles que l'analyseur de spectre RF, radar... Ces applications nécessitent des solutions de traitement numérique de signal puissantes et flexibles. Les circuits de la logique programmable offrent performance et flexibilité requises dans de telles applications, parce que les performances de ces circuits peuvent dépasser celles des processeurs DSP et ASICs.

Les solutions classiques de traitement numérique du signal sont les suivantes :

- Microprocesseurs : Ils sont toujours utilisés dans des applications moins exigeantes en puissance de calcul.

Processeurs DSP : Le processeur DSP a été développé pour répondre à la demande réclamée dans de le calcul des algorithmes complexes. Un DSP a généralement une architecture de type RISC (Reduced Instruction Set Computer, microprocesseur à jeu d'instructions réduit), et fournit une puissance de calcul de 3 à 50 MIPS. Ces processeurs peuvent réaliser en un cycle une opération qui requiert plusieurs cycles dans un microprocesseur.

- Les ASICs : L'ASIC est un circuit intégré spécifique à une application telle que le décodage MPEG. Cependant les coûts et temps de développement de ces circuits les dédient aux grandes séries.

Aujourd'hui, la situation des systèmes d'interfaçages a beaucoup évolué depuis l'apparition du bus ISA [1,2] d'IBM. Les fréquences de fonctionnement des processeurs et des mémoires ont considérablement augmenté. Certes, la bande passante du bus PCI-X a permis d'augmenter la fréquence du bus PCI [3,4] en passant de 33 à 66, 100 ou encore à 133 MHz. Cependant, cette augmentation de la bande passante ne résout pas les problèmes des temps d'attente et s'accompagne d'une réduction importante de la distance sur laquelle les données peuvent être acheminées. Le nombre de connecteurs que les transmetteurs de bus peuvent gérer est aussi réduit.

Les applications telles que l'acquisition de données, la génération de formes d'ondes, le "streaming" (diffusion en mode continu) audio et vidéo nécessitent une bande passante garantie et des temps d'attente déterministes. Les spécifications du PCI dans sa version originale ne répondent pas à ces besoins car ces applications n'étaient pas déterminantes au moment du développement du bus. Les applications de transfert de données en temps réel qui se développent aujourd'hui en vidéo, gourmandes en bande passante et exigeant un temps d'attente critique, nécessitent également la prise en charge de ce type de transfert.

Par ailleurs l'AGP a remplacé le PCI pour accroître les performances du circuit vidéo, tout en évitant de saturer la bande passante que se partagent les autres composants connectés sur le bus du PC. Le transfert est ainsi passé de 132 Mo/s (32 bits/33 MHz) avec le PCI à des débits théoriques successifs de 266 Mo/s, 533 Mo/s, 1,04 Go/s et 2,13 Go/s avec respectivement l'AGP 1X, 2X, 4X, 8X [5,6].

En pratique, le débit réel est considérablement inférieur à sa valeur théorique. En effet, le gain théorique entre l'AGP 8X et l'AGP 4X est de 100%. Cependant, le gain pratique est inférieur à 5%. Cette situation impose une nouvelle réflexion architecturale pour l'interfaçage graphique temps réel et qui prendrait en considération l'augmentation de la bande passante pratique avec la réduction des temps d'attente.

Notre travail s'inscrit dans ce contexte de convergence et vise essentiellement deux objectifs : premièrement, la conception et la réalisation de nouvelles architectures flexibles d'interfaçages, basées sur les systèmes mixtes logiciels/matériels, pour l'échange de données entre les systèmes à microprocesseur et des périphériques externes. Le deuxième objectif est l'élaboration de nouvelles techniques de production d'images utilisant un parallélisme matériel et garantissant un débit pratique de transfert élevé. Ces nouvelles techniques visent, essentiellement, l'élargissement de la période d'accès réservée à l'enregistrement sur la mémoire vidéo. Afin d'assurer des transactions en temps réel, l'architecture de reconstruction d'images que nous proposons utilise un accès simultané de lecture/écriture basé sur la décomposition physique de la mémoire vidéo.

Le traitement d'images en temps réel exige l'emploi de circuits rapides capables de manipuler les grandes quantités d'informations générées par la source vidéo. Les circuits programmables fournissent une puissance de calcul importante en maintenant la complète flexibilité soft d'un processeur DSP. La possibilité de reconfiguration sur la carte (In Circuit Reconfigurability) permet une optimisation et une adaptation spécifique à chaque architecture.

L'implantation des systèmes dans un composant programmable offre les avantages suivants :

- Parallélisme : Implanter des fonctions au niveau matériel permet un parallélisme au niveau calcul, ce qui conduit à des performances supérieures à celles des microprocesseurs et des processeurs DSP.
- In Circuit Reconfigurability (ICR) : Ceci permet de charger de nouvelles versions du système dans le circuit (mises à jour matérielles).

Les travaux que nous avons réalisés dans le cadre de l'interfaçage, nous ont permis de développer quatre nouvelles techniques. La première version nommée "**Adressage Physique Etendu**" constitue une solution pour l'élargissement de la capacité d'adressage matériel des systèmes à microprocesseur. L'**Adressage Physique Accéléré**, publiés en 2002 dans les actes du " IEEE Canadian Conference on Electrical and Computer Engineering", a été développé dans une deuxième phase pour apporter une amélioration à la fréquence maximale de travail de la première variante. Les deux dernières versions, à savoir, l'**Adressage Physique Rapide** et l'**Adressage Physique Avancé**, publiés respectivement en 2003 et 2004 dans les actes du "The IEEE International Workshop on System-on-Chip for Real-Time Applications", réalisent des tâches plus complexes.

Sur le plan de la conception des systèmes de production d'images, nous avons élaboré de nouvelles architectures de reconstruction vidéo en temps réel. Ces dernières se basent sur un parallélisme matériel associé à nos techniques d'interfaçage. Une partie des travaux d'interfaçage graphique, réalisés dans le cadre de cette thèse, est référencée dans le brevet "DIGITAL IMAGE TRANSFER CONTROLLER" [7], exploité par Sony® et enregistré dans "United States Patent" en septembre 2005.

Cette thèse est répartie en quatre chapitres :

Le premier chapitre est consacré à la présentation de l'état de l'art des interfaces d'entrées-sorties et le fonctionnement général des bus actuels.

Le deuxième chapitre expose nos quatre nouvelles techniques d'interfaçages dans les systèmes à microprocesseur : l'Adressage Physique Etendu, l'Adressage Physique Accéléré, l'Adressage Physique Rapide et l'Adressage Physique Avancé.

Dans le troisième chapitre nous exposons une revue des principaux types de circuits intégrés programmables et reprogrammables, allant de la simple mémoire programmable au circuit FPGA (Field Programmable Gate Array). Ensuite nous présentons l'implémentation et le test de nos techniques d'interfaçage sur des cibles de la logique programmable en utilisant des solutions schématiques et VHDL.

Le quatrième chapitre présente les nouvelles techniques de production d'image basées sur les nouveaux systèmes d'interfaçages ainsi que les techniques d'affichage admettant un accès simultané de lecture/écriture sur la RAM vidéo.

La thèse se termine par une conclusion générale où nous commentons le travail effectué et nous proposons des améliorations et perspectives.

Chapitre 1

Interfaces d'entrées-sorties et bus d'extension

1. Introduction

Un système à microprocesseur, ou un ordinateur, est composé principalement de trois composants : le microprocesseur (CPU), la mémoire et les interfaces d'entrées-sorties, encore appelées simplement entrées-sorties ou IO (In/Out) [8]. Le microprocesseur et les mémoires forment le cœur du système de traitement et de transformation des données. Les interfaces relient le microprocesseur au monde extérieur des périphériques (écran, clavier, souris, imprimante, machines dans les ateliers de production, robots, etc.). Les périphériques sont soit des parties indispensables au fonctionnement de l'ordinateur (écran, clavier...), soit des parties destinés à élargir les possibilités de l'ordinateur vers d'autres applications (imprimante, scanner, système d'acquisition de données ...). La figure 1.1 illustre la structure générale d'un ordinateur avec ses interfaces [9].

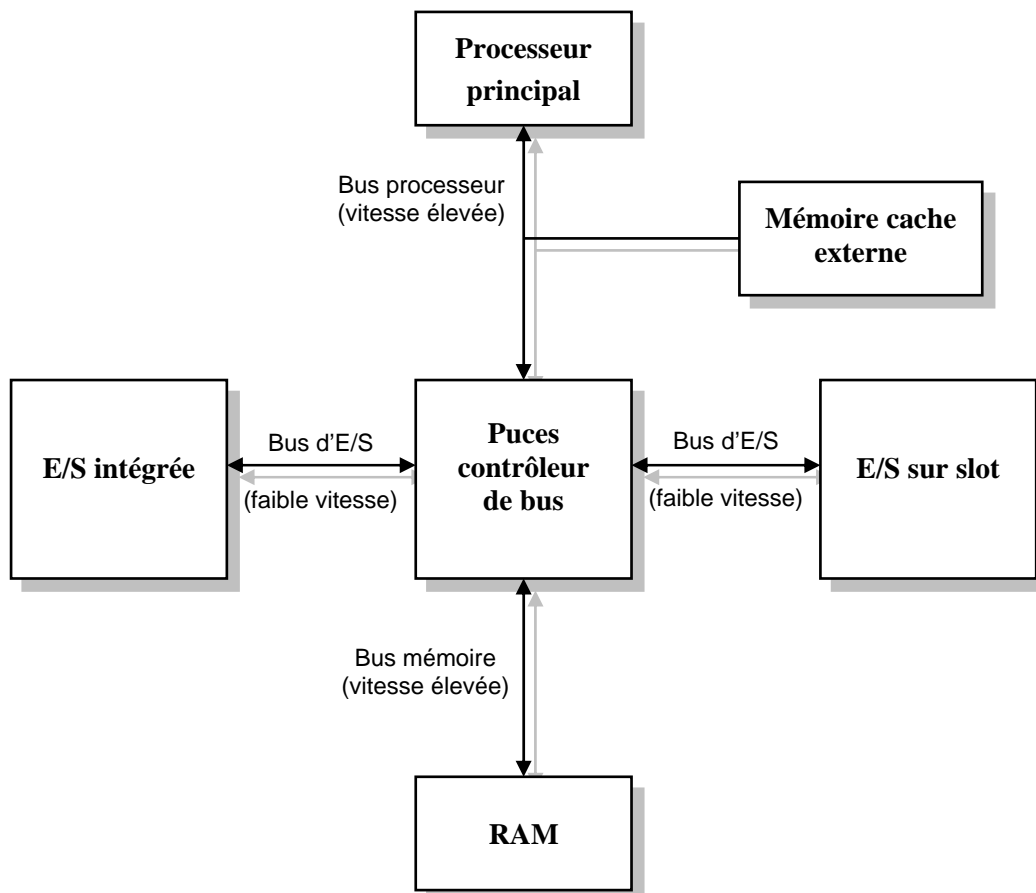


Fig. 1.1. Structure générale d'un ordinateur

Dans cette structure, le rôle des interfaces est triple. Les interfaces assurent l'Adaptation des signaux entre le microprocesseur et les périphériques. Elles déchargent le microprocesseur des tâches non adaptées à sa nature comme la communication avec les périphériques de faible vitesse. Pour accroître les performances du système, elles assurent les communications périphériques-périphériques et les communications interfaces-périphériques pendant l'exécution des tâches internes, comme le dialogue entre microprocesseur et la mémoire centrale.

Les transactions dans un ordinateur sont réparties principalement en quatre catégories :

- Transactions Microprocesseur - Mémoire
- Transactions Microprocesseur - Registres d'interface
- Transactions Registres d'interface - Mémoire
- Transactions Interfaces - Périphériques

Pour assembler les différents composants d'un ordinateur, on a besoin de faisceaux de lignes électriques appelés "bus". Un bus assure les liaisons ou les connexions physiques entre les composants d'un ordinateur. La description des caractéristiques des bus et des protocoles de transfert permet de connaître le fonctionnement général des interfaces et des bus qui lui sont associés.

Une interface possède généralement deux bus, un qui le connecte au microprocesseur et un autre qui le connecte à son périphérique. Généralement, du côté microprocesseur, le mode de connexion est le même pour toutes les interfaces ; il s'agit des signaux de types TTL (0, 5V) et de types LVTTTL (0, 3,3V) travaillant à la cadence déterminée par le microprocesseur. Du côté des périphériques, les signaux de l'interface sont adaptés à la nature du périphérique que l'interface contrôle.

2. Caractéristiques d'un bus

Le cœur de tout système à microprocesseur est représenté par les bus qui le composent. Un bus n'est rien d'autre qu'un chemin commun emprunté par les données qui circulent au sein d'un ordinateur. Ce chemin est l'élément qui permet à l'ordinateur d'établir des communications entre ses différents composants. Physiquement, un bus apparaît sous plusieurs formes :

- Un connecteur sur une carte mère disposé à recevoir des cartes électroniques (slots d'extension comme ISA, PCI, PCI Express, AGP ...)
- Un câble électrique. Il s'agit d'un ensemble de lignes électriques, d'une certaine longueur qui varie entre un mètre et plusieurs dizaines de mètre, regroupées en une nappe de fils plats ou dans un câble multifilaire, montées aux deux bouts par des connecteurs pour faciliter le montage et le démontage (câble USB, ligne série RS-232, ligne parallèle, connexions de réseau locaux,...).
- Un ensemble de connexions électriques d'un circuit, identiquement à un bus d'un processeur, bus d'une mémoire, bus d'une interface Le constructeur du circuit est responsable des spécifications complètes du circuit et de son bus.

Les ordinateurs ou les systèmes à microprocesseurs reposent sur un système de bus organisés hiérarchiquement. La plupart des ordinateurs modernes possèdent au moins trois bus différents. Ils sont organisés hiérarchiquement, car chaque bus est relié à un bus plus rapide situé au-dessus de lui. Chaque composant de l'ordinateur est connecté à l'un de ces bus. Pour définir un bus, il faut fournir les caractéristiques suivantes :

- L'identité du bus (appellation habituelle). Exemple : AGP

- L'usage du bus.
- Les paramètres physiques et électriques.
- Les caractéristiques fonctionnelles
- Les caractéristiques dynamiques.

2.1. Identité du bus

Un bus possède un ou plusieurs noms dont les provenances sont très différentes:

- Nom technique venant de l'organisme qui le standardise. Exemple : FireWire ; **IEEE 1394...**
- Nom commun fondé souvent sur la structure physique ou électrique de l'interface. Exemples : parallèle, série...
- Nom portant l'empreinte de la compagnie qui le commercialise. Exemple : ST-506 dans lequel "ST" vient de Seagate...
- Nom se référant aux objets sur lesquels le bus se connecte. Exemples : bus de processeur, bus de mémoire
- Nom synthétique quelconque. Exemple : PCI : Peripheral Component Interconnect.

2.2. Usage du bus

Tout bus est conçu pour un usage spécifique. Dans ce sens, on peut distinguer quelques sous-familles (certains bus se retrouvent dans plusieurs sous-familles en raison de l'évolution historique) :

- Bus d'usage général sous forme de câble qui se trouve à l'extérieur de l'ordinateur. Il s'agit souvent d'un câble servant à relier plusieurs objets. Nous citons le bus parallèle qui connecte un ordinateur à une imprimante, le bus Laplink qui connecte deux ordinateurs ensemble et bus PS2 qui connecte un ordinateur à une souris.
- Bus interne à l'ordinateur conçu pour un périphérique désigné. Dans cette catégorie, on trouve par exemple, l'ancien IDE qui ne peut supporter que deux disques durs, le EIDE (Extended IDE) qui peut prendre en charge quatre disques durs, des CD-ROM et d'autres unités de stockage de masse.
- Bus de composant. Dans cette catégorie, on trouve les bus des microprocesseurs et des mémoires.
- Bus adapté à un type de transfert. Pour le transfert graphique, nous citons le bus AGP (Accelerated Graphics Port). Ce bus est remplacé par le PCI Express.

2.3. Paramètres physiques et électriques

Une spécification précise des paramètres physiques et électriques facilite la connexion entre les divers modules. Ces paramètres sont :

- Les caractéristiques physiques (nombre de fils, type de conducteurs, type d'isolants, connecteurs, longueur ...).

- Les caractéristiques électriques (tensions, courants, fréquence maximale qui est l'image du taux de transfert ...).
- Les types de signaux. Les signaux sont majoritairement de type binaire. Certains bus comportent des signaux analogiques ou des signaux modulés (signaux d'interfaçage avec les unités de disques ou avec les modems).
- L'arrangement des signaux (multiplexage des adresses et des données, multiplexage des lignes de contrôle, activation des lignes à l'état bas ou à l'état haut). Un bus est multiplexé lorsque les données, de nature différente, sont acheminées par les mêmes fils. Le multiplexage temporel diminue le nombre de connexions physiques, mais réduit également la bande passante du bus (2 fois ou plus) pour une même qualité de bus.

2.4. Caractéristiques fonctionnelles

Les caractéristiques fonctionnelles décrivent les transactions sur le bus, les possibilités de transfert ainsi que les paramètres qui s'y rattachent. Nous citons:

- Composition du bus. Un bus peut être décomposé en un certain nombre de bus plus élémentaires (bus d'adresses, bus de données, bus de contrôle, bus utilitaire, bus d'interruption, bus d'arbitrage pour l'acquisition du bus par un maître ...).
- Bus mono maître ou multi maître. Un bus mono maître gère uniquement le dialogue d'un maître (CPU, DMA ...) avec un ensemble d'esclaves (mémoire, interfaces ...). Dans les bus multi maîtres, plusieurs maîtres peuvent soumettre leurs requêtes pour effectuer un transfert. Un certain nombre de lignes de contrôle sont destinées à gérer le partage du bus. Le bus ISA est un bus mono maître. Le bus PCI est multi maîtres.
- Nombre maximum de maîtres et d'esclaves. La capacité d'accueil d'un bus a des limites, les spécifications du bus précisent le nombre maximum de cartes qui peuvent être connectées ensemble. Par exemple, un bus IDE ne peut connecter que deux disques durs à l'interface IDE.
- Largeur du bus ou largeur des données. Les bus sont souvent catégorisés comme 8, 16, 32, 64 bits ... Cette information vient de la largeur des données. Les bus modernes sont extensibles. Sous la même désignation, un bus peut être de 8, 16 ou 32 bits, dépendant de l'appareil.
- Bande passante. La bande passante d'un bus est liée à la fréquence maximale du bus. Cette mesure est exprimée souvent en bits par seconde (bps ou ses multiples Mbps, Gbps) ou en octets par seconde et ses multiples (Ko/s, Mo/s ou Go/s).

2.5. Caractéristiques dynamiques

Pour décrire les caractéristiques dynamiques d'un bus, on établit tous les scénarios qui peuvent se produire. Les scénarios sont des descriptions des séquences de fonctionnement dans l'accomplissement d'une fonction spécifique du bus.

Les chronogrammes sont employés pour décrire les signaux électriques et les protocoles de transfert. Un protocole est une description dynamique complète d'un type de transfert. L'ensemble des spécifications dynamiques d'un bus peut atteindre facilement quelques centaines de pages de description.

Le chronogramme de la figure 1.2 représente le mécanisme d'accès aux données d'une mémoire. Toutes les adresses sont regroupées dans la première ligne et toutes les données dans la seconde. Les zones non hachurées sont des zones valides dans lesquelles le CPU peut lire les informations de façon sécuritaire. Les zones hachurées sont des périodes transitoires où les données ne sont pas encore stabilisées. Les adresses et les données véhiculent des informations dans les deux états, bas et haut. La flèche montre le temps d'accès à partir des adresses dans le cas d'une mémoire, elle représente l'intervalle qui sépare deux événements : la commutation des adresses et l'instant où les données placées par la mémoire sur le bus deviennent stables et sécuritaires pour une lecture. Le chronogramme constitue une spécification dynamique d'un bus.

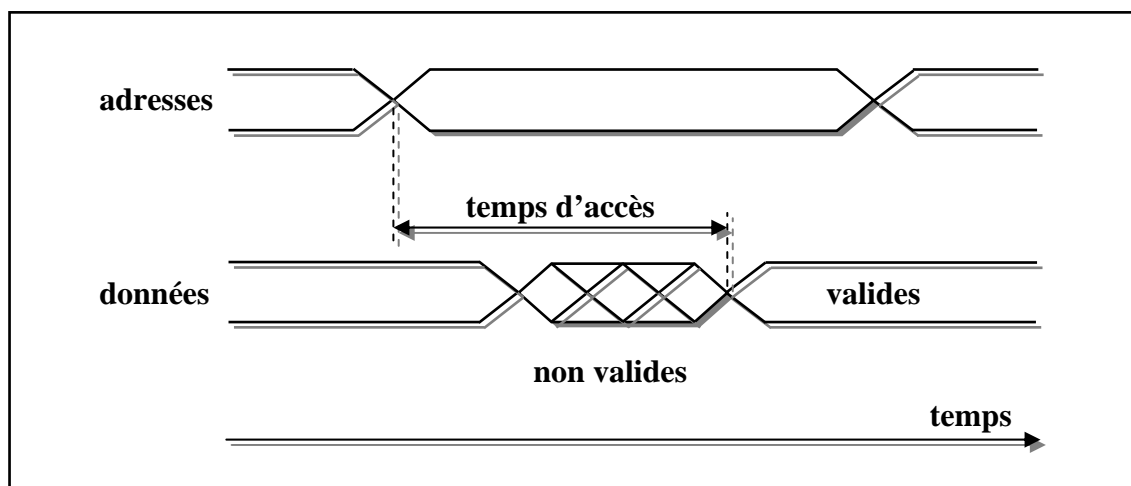


Fig. 1.2. Le chronogramme d'accès aux données

3. Modèles de Bus

3.1. Bus de composant

Dans cette catégorie, le bus apparaît sous forme d'un ensemble de lignes électriques liées à la nature du composant ou du module électronique. La description est "ouverte" et ne tient pas compte des organes connectés au composant.

Les bus internes aux chips ne nécessitent aucune standardisation, seul le fabricant du chip est responsable de sa définition (exemple : le bus interne entre le Pentium III et sa cache L1 n'étant pas externe, Intel est le seul intervenant dans sa définition). Il en est de même pour le bus entre le CPU et ses coprocesseurs intégrés. On voit de temps à autre quelques indications sur la taille des bus internes, leur rapidité, leurs performances dans un but plutôt publicitaire.

3.2. Bus de processeur

Ce bus est le plus rapide de l'ordinateur. Il est utilisé essentiellement par le processeur pour faire circuler des informations à destination ou en provenance de la mémoire cache, de la mémoire vive et du circuit North Bridge. Les processeurs possèdent quatre bus élémentaires, bus d'adresses, bus de données, bus de contrôle et bus utilitaires (alimentation électrique, économie d'énergie). Ces bus élémentaires sont fusionnés pour constituer un bus d'ensemble appelé "bus du processeur". Comme tous les processeurs sont différents, ce bus est spécifique ou propre à chaque processeur. On donne à ce bus le nom du processeur lui-même, on parle alors de bus du Pentium 4, bus du 68030, bus du 486, bus du 6809, etc.

Le bus de processeur fonctionne à la même fréquence d'horloge que le processeur en externe. Cela peut être trompeur pour l'utilisateur, car actuellement, la plupart des processeurs fonctionnent à une fréquence plus élevée en interne qu'en externe. Pour déterminer le taux de transfert du bus de processeur, il faut multiplier la quantité de données pouvant être transférées simultanément (64 bits pour un Pentium) par la fréquence d'horloge du bus (identique à la fréquence du processeur avant multiplication). Si un système est équipé d'un processeur de 3,2 GHz avec un bus de 200 MHz et s'il est capable de transférer un bit de données par cycle sur chaque ligne de données, le taux de transfert instantané maximal est de 1,6 Go/s. Pour obtenir ce chiffre, il suffit d'utiliser la formule suivante :

$$\begin{aligned} 200 \text{ MHz} \times 64 \text{ bits} &= 12,8 \text{ Gbps} \\ 12,8 \text{ Gbps} \div 8 &= 1.6 \text{ Go/s} \end{aligned}$$

Ce taux de transfert, souvent appelé "largeur de bande du bus", représente un maximum. Il faut donc s'attendre à un débit moyen plus faible.

3.3. Bus de mémoire

Le bus de mémoire assure les transferts de données entre le processeur et la mémoire principale. Les mémoires disposent également de trois bus : bus d'adresses, bus de données, bus de contrôle. Les lignes de contrôle sont très simples dans le cas des mémoires

1. CS : chip select ou CAS (Column Address Strobe) et RAS (Row Address Strobe)
2. R/W (Read/Write) ou W (Write) pour une lecture écriture
3. OE (Output Enable) pour une autorisation de sortie des données

Lorsque les mémoires sont regroupées en module, on considère le module comme un ensemble et on caractérise le bus du module. Avec l'arrivée des SDRAM, les modules de RAM dynamique possèdent plus de lignes de contrôle que les mémoires classiques SRAM, ou DRAM élémentaires.

3.4. Bus d'une interface

Les entrées-sorties, IO (In/Out) ou interfaces sont des chips qui s'intercalent entre deux systèmes de bus. D'un côté, l'interface doit se connecter au CPU ; de l'autre côté, elle doit avoir toute la logique nécessaire pour gérer les protocoles de communication pour lesquels elle a été conçue. Ci-dessous est la description des bus d'une interface générique.

Le schéma de la figure 1.3 montre l'interface avec ses deux bus. D'un côté, elle possède la logique nécessaire pour dialoguer avec le CPU ; de l'autre, un bus spécialisé est conçu selon l'usage de l'interface. Du côté de la logique d'interfaçage avec le processeur, chaque interface est munie d'une ligne de sélection CS et d'une ligne R/W pour mettre l'interface en mode lecture ou en mode écriture. Les n lignes "Register Selects" RS_0 à RS_{n-1} permettent de sélectionner un registre parmi les 2^n registres internes de l'interface. La ligne d'interruption IRQ est une ligne matérielle qui permet à l'interface d'interrompre le processeur pour demander un traitement particulier.

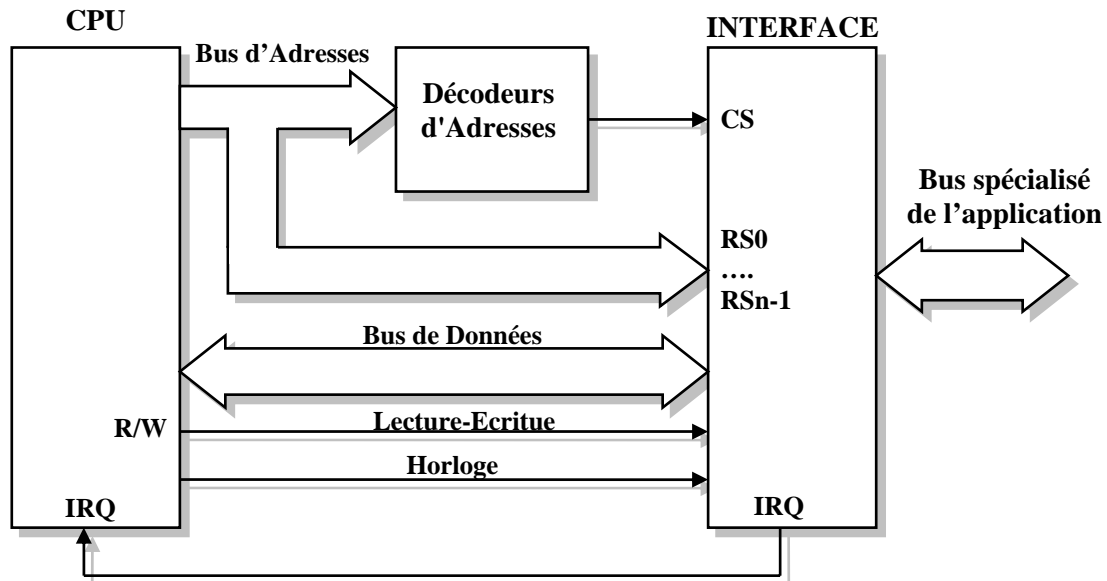


Fig. 1.3. Architecture de l'interface

4. Interfaces d'extension

Depuis le premier PC, de nombreux bus d'extension ont vu le jour, et ce, pour une raison très simple : pour améliorer les performances d'un système donné, il faut accélérer le débit de ses entrées-sorties [10]. Le besoin croissant de performances toujours accrues provient principalement :

- des processeurs plus rapides;
- des exigences aiguës des logiciels modernes;
- des nouveaux besoins dans le domaine du multimédia.

Chacun de ces points requiert un débit d'entrées-sorties maximal. Aussi surprenant que cela puisse paraître, la plupart des PC vendus aujourd'hui présentent toujours la même architecture de bus de base que l'AT sorti en 1984. Cependant, la plupart de ces ordinateurs possèdent désormais un second bus local d'entrées-sorties plus rapide, de type PCI ou PCI Express, qui assure des performances bien supérieures aux cartes d'extension qui le requièrent.

Si les nouvelles structures des bus d'extensions ont été si longues à implémenter, c'est essentiellement pour des raisons de compatibilité. L'un des facteurs du succès du PC est sa standardisation. Celle-ci a donné naissance à des milliers de cartes d'extension produites par des sociétés indépendantes et fondées sur les premières spécifications des bus PC. Si un nouveau système de bus à performances accrues voit le jour, il devra être compatible avec les anciens systèmes pour éviter que les anciennes cartes d'extension deviennent obsolètes.

Les différents bus d'extension peuvent être identifiés par leur architecture. En voici les principaux types:

- ISA;
- PCI;
- PCMCIA ;
- AGP ;
- PCI Express.

Ces bus se différencient essentiellement par la quantité de données qu'ils sont capables de transférer simultanément et par la vitesse à laquelle ils le font. Chaque architecture est implémentée par un chipset connecté au bus de processeur. Ce chipset contrôle en général aussi le bus de mémoire. Les parties suivantes décrivent les différents types de bus pour PC.

4.1. Bus AT ou ISA

Le bus ISA est le bus 8 bits qui équipait le premier PC d'IBM lancé en 1981 (le bus PC-XT 4.77 Mhz). Ce bus a ensuite été étendu à 16 bits sur l'ordinateur AT lancé en 1984 (figure 1.4). Comme le terme "AT" est une marque de commerce de IBM, ce bus est connu plus tard sous le nom de ISA (Industry Standard Architecture) [1,2].

Il existe deux versions de bus ISA, qui se distinguent par le nombre de bits de données qu'elles peuvent transférer simultanément. La version la plus ancienne est de type 8 bits; la plus récente fonctionne en 16 bits. La première était cadencée à 4,77 MHz sur les PC et XT. La version 16 bits en usage sur les AT était cadencée à 6, puis à 8 MHz. Par la suite, les grands fabricants de matériel informatique ont choisi une vitesse standard maximale de 8,33 MHz pour les versions 8 et 16 bits du bus ISA, afin d'en assurer la compatibilité descendante.

Certains systèmes peuvent accélérer la vitesse d'exécution du bus ISA (10 ou 12 Mhz pour certains fabricants), mais il ne faut pas oublier que certaines cartes d'extension risquent de ne pas fonctionner correctement à une vitesse accrue. Les transferts de données effectués par l'intermédiaire d'un bus ISA nécessitent de deux à huit cycles. Pour ce faire, le taux de transfert théorique maximal du bus ISA est d'environ 8 Mo/s, comme le démontre la formule mathématique suivante:

$$\begin{aligned}8 \text{ MHz} \times 16 \text{ bits} &= 128 \text{ Mbps} \\128 \text{ Mbps} \div 2 \text{ cycles} &= 64 \text{ Mbps} \\64 \text{ Mbps} \div 8 &= 8 \text{ Mo/s}\end{aligned}$$

Le bus 8 bits offre une largeur de bande deux fois moins importante (limitée à 4 Mo/s).

Cependant, ces chiffres ne sont que des valeurs maximales théoriques. En raison des protocoles de bus d'extension, la quantité d'informations transférées à un instant T est inférieure d'au moins la moitié. Même ainsi, à 8 Mo/s, le bus ISA reste plus rapide que beaucoup de périphériques qui y sont connectés.

Malgré la simplicité de conception de ce bus, ce n'est qu'en 1987 qu'IBM a publié des spécifications exhaustives sur les caractéristiques de ses lignes de données et d'adresses. Pour mettre au point les premiers ordinateurs compatibles, les fabricants ont dû effectuer leurs propres recherches, dont les résultats n'étaient pas toujours optimaux pour l'utilisateur. Cependant, ce problème a été résolu avec la consécration des ordinateurs individuels compatibles PC, considérés comme un standard dans le monde de l'informatique. Les fabricants ont alors disposé de plus de temps et de moyens pour élaborer des cartes d'extension fonctionnant correctement avec le bus ISA.

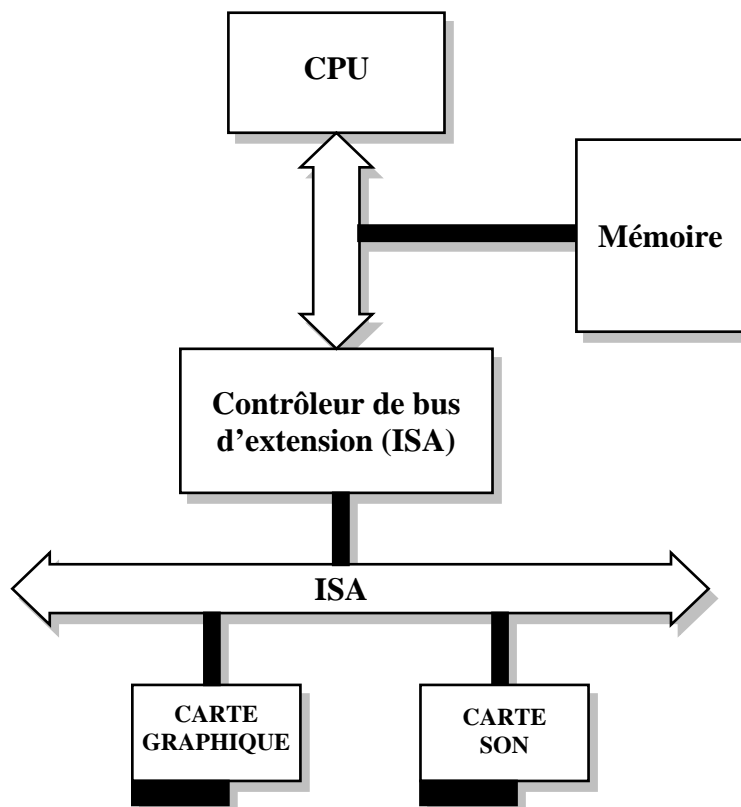


Fig. 1.4. Bus local et bus d'extension type ISA

En 1997, les cartes mères contenaient encore quelques slots ISA, à côté des fentes PCI, pour autoriser l'usage d'un immense stock de cartes d'interface du passé. En 2000, plusieurs cartes mères ont déjà abandonné complètement ce bus, d'autres gardaient encore 1 seule fente, exceptionnellement 2. En 2002, le slot du bus ISA disparaît de presque toutes les cartes mères. Le standard ISA existe toujours dans les circuits South Bridge pour assurer le fonctionnement des entrées-sorties classiques.

4.1.1. Utilisation du Bus ISA.

Le principe de l'utilisation du bus ISA pour l'interfaçage est illustré sur la figure 1.5. Les adresses des ports des circuits auxiliaires et de certaines interfaces ont été fixées à l'origine par IBM et font donc partie des paramètres standard de la microinformatique.

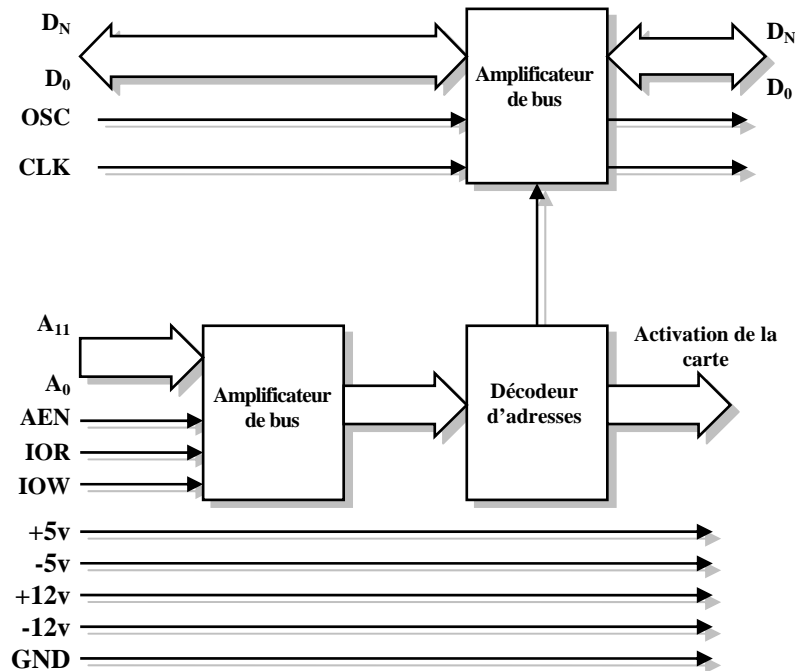


Fig. 1.5. Principe d'utilisation du bus ISA

Les adresses des cartes de prototype sont comprises entre 300 et 31F (adresses exprimées en hexadécimal). Par exemple si on veut adresser la carte par l'adresse 304, il faut concevoir la logique combinatoire de telle manière à avoir un « 1 » si l'adresse 304 (Hexa) est disponible au bus d'adresse.

Exp: pour avoir «a» au bus de données on peut utiliser l'instruction {outport(304, a)} en langage C.

4.1.2. Signification des principaux signaux du bus ISA.

OSC : Fréquence de l'oscillateur (*oscillator*). Ces broches permettent de mesurer la fréquence du quartz.

CLK : Horloge système (*clock*). Signal utilisé pour générer un signal de rafraîchissement de la RAM.

AEN : Contrôle le bus d'adresses (*address enable*). Cette ligne permet à la carte de continuer à exécuter des accès directs à la mémoire même si elle venait à être séparée du bus. Lorsque ce signal est égal à 1, le DMA a le contrôle sur les lignes de données, d'adresses et de contrôle.

D₀ à D_N : Lignes de données

A₀ à A_M : Bit 0 au bit M du bus d'adresses

IOW : Carte : écriture de données (*input/output data write*). Ce signal avertit la carte qu'elle doit enregistrer des données, c'est-à-dire extraire le contenu des bus de données. Le signal est produit par le processeur ou par l'unité DMA.

IOR : Carte : lecture de données (*input/output data read*). Ce signal avertit la carte qu'elle doit extraire des données de la mémoire et les transmettre sur le bus de données. Le signal est produit par le processeur ou par l'unité DMA.

4.2. Bus PCI

Au début de 1992, Intel a été à l'origine de la création d'un nouveau groupe de standards rassemblant des grands acteurs du monde de l'informatique. Ce groupe avait le même objectif vis-à-vis des bus de PC que celui de la VESA précédemment: pallier les faiblesses des bus ISA et EISA. Le bus PCI (Peripheral Component Interconnect) reçoit une standardisation et un groupe spécial d'intérêt PCI SIG (Special Interest Group) veille à son développement.

Adopté en 1994, avec l'introduction du 486, ce bus supplante le bus ISA et la vitesse du bus atteignait 133 MHz en 1998 (révision PCI-X endossée par Compaq, Hewlett-Packard et IBM). Existait initialement sous forme 32 bits à 33 MHz, la vitesse du PCI évolue rapidement vers le mode 64 bits (une rallonge supplémentaire au connecteur initial était alors nécessaire) à 66 Mhz puis 133 MHz avec la version PCI-X (voisinage de 1Gbits/s de taux de transfert ou "transfert rate"). Les BIOS assez performants sont capable de régler la vitesse du PCI pour adapter ce bus à la vitesse des interfaces connectées avec le bus. Le PCI est capable d'opérer avec une tension de 5 volts ou 3.3 volts [3,4].

Le PCI est un bus qui adopte les fonctionnalités du bus ISA et du bus industriel VESA (VL-Bus: VESA Local Bus. VESA signifie Video Electronics Standards Association). PCI est un bus multiprocesseur qui supporte l'arbitrage de bus et plusieurs maîtres. PCI utilise un "pont" pour se connecter sur le FSB (Front Side Bus). Il supporte l'accès DMA et 4 lignes d'interruption. Le PCI bridge régularise la vitesse du PCI quelle que soit la vitesse du CPU. Le PCI est un bus multiplexé (multiplexage des adresses et des données), c'est-à-dire chaque pin électrique assure une fonction différente à divers moments. Ce multiplexage permet à ce bus d'avoir un facteur de forme physique assez réduit. Malgré le fait qu'il soit multiplexé, sa rapidité vient du fait qu'il est capable de transférer en mode synchrone en plus du mode asynchrone, qu'il est capable d'opérer des transferts en mode rafale (burst mode).

Les codes des fabricants pour la reconnaissance sur le bus PCI sont gérés par PCI SIG. PCI n'est pas un bus uniquement pour la plateforme PC. Il se retrouve aussi dans SUN, Apple. D'après la spécification du bus, un minimum de 49 signaux est nécessaire pour un module capable d'assurer des fonctions de maître, tandis que les autres signaux normalisés peuvent n'être que des options. La figure 1.6 montre la liste des signaux du bus PCI suivant la norme et les interconnexions entre le bus et un module.

De plus, la spécification du bus autorise l'interconnexion, sur le bus PCI, jusqu'à quatre cartes entre elles. Egalement, la norme prévoit l'emploi de passerelles (bridges) d'un bus PCI vers un autre bus PCI, ou bien un bus d'un autre type.

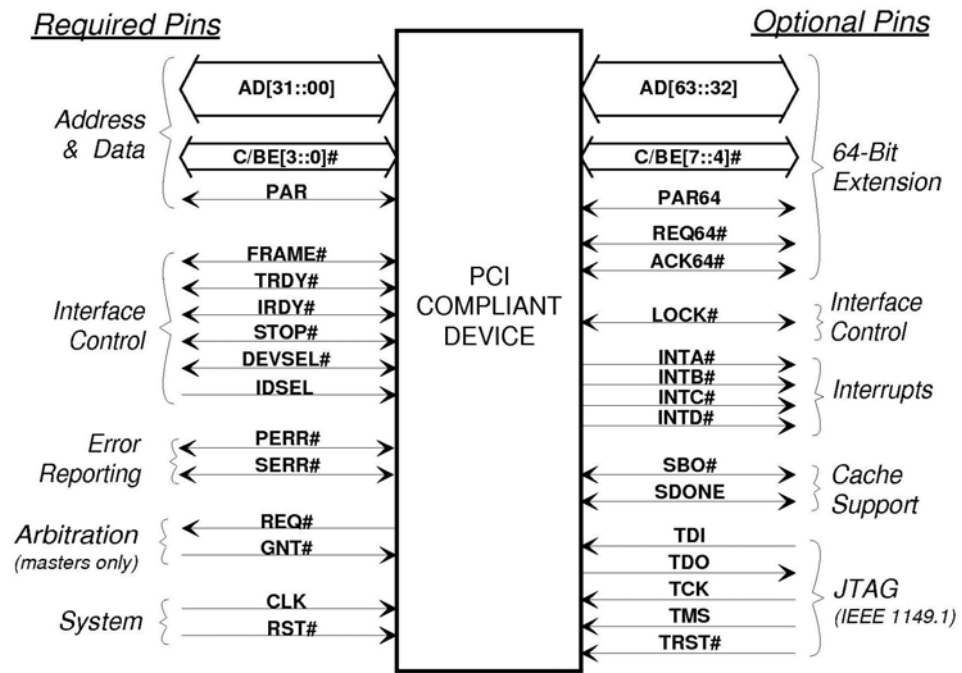


Fig. 1.6. Liste des signaux du bus PCI suivant la norme

Dans le cas de l'utilisation de bridges PCI/PCI, jusqu'à 256 passerelles sont possibles. Enfin, le bus comprend obligatoirement une unité de gestion d'accès au bus, tandis qu'il peut intégrer dans ce même arbitre un gestionnaire d'interruptions. La spécification 2.1 suggère une architecture classique d'utilisation du bus PCI. La figure 1.7 présente le schéma bloc d'une architecture PCI.

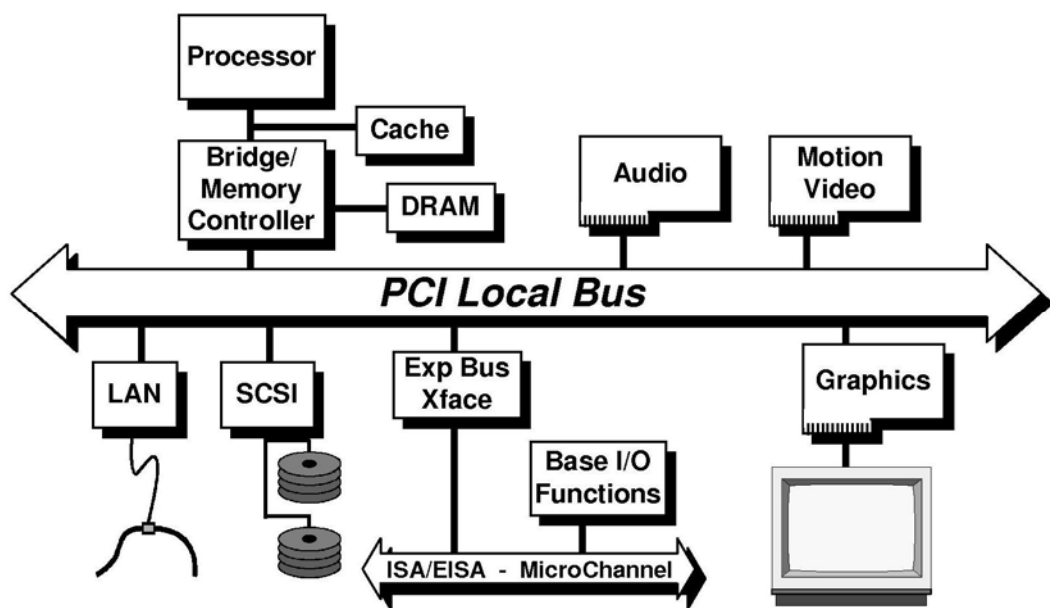


Fig. 1.7. Schéma bloc d'une architecture PCI

4.2.1. Description des signaux

Signaux système :

CLK : est l'horloge du bus PCI (généralement à 33 et 66 MHz).

RST# : est le reset utilisé pour l'initialisation de tous les composants du bus.

Adresse et données :

AD[31:0] : Bus d'adresses et de données multiplexé, noté AD[31..00], qui permet donc d'exploiter 32 bits en adressage, comme en donnée. On obtient donc un espace adressable de 4 Go, sachant que la norme réclame un décodage complet sur l'ensemble des bits nécessaires par les esclaves. Ce bus prend la fonction de porteur d'adresses dans la phase dite d'adressage sur le bus, tandis qu'à l'issue de cette phase, il devient bus de données.

C/BE[3:0]# : Bus de commandes et de validation d'octets, noté C/BE[3..0], qui permet, durant la phase d'adressage, de transmettre la commande de bus qui va être réalisée, tandis qu'à cette issue, il devient bus de validation des octets circulant en groupes de 32 bits sur le bus de données.

Signaux de contrôle d'interface:

Bus de contrôle qui regroupe les signaux gérés par le maître :

FRAME : Un signal fourni par l'initiateur d'une transaction (le maître) pour indiquer le début d'une transaction ainsi que sa durée.

IRDY : Le signal indiquant que le maître est prêt à passer à la phase de donnée suivante (état du maître de l'échange).

En réponse l'esclave fournit :

TRDY : Indique l'état de l'esclave dans échange et que la cible est prête pour la phase suivante.

DEVSEL : Indique au maître que l'esclave s'étant reconnu après l'adressage et qu'il répond à l'accès réalise sur le bus.

STOP : Demande d'arrêt de l'échange.

4.2.2. Transfert de données

Le principe de transfert de données sur le bus PCI exploite les échanges par paquets (burst), qui mettent en jeu la quasi-totalité des signaux de la configuration minimum.

Pour exécuter un transfert sur le bus PCI, le maître concerné doit d'abord s'assurer de la disponibilité du bus pour son propre usage. Cette phase étant réalisée avec succès, le maître engage l'accès bus en positionnant l'adresse et la commande, en même temps que le FRAME. Après un front d'horloge, le maître positionne alors l'IRDY, qui signale qu'il est paré par le transfert, mais surtout termine la phase d'adressage. Les esclaves potentiellement concernés sont alors amenés à répondre au maître consécutivement aux décodage d'adresses qu'ils ont chacun réalisé. Le signal DEVSEL positionné indique alors qu'un esclave s'est reconnu pour l'échange, et le positionnement de TRDY indique qu'il est prêt pour le transfert. Le maître positionne alors les signaux de validation des octets C/BE[3..0], et dans le cas de l'écriture, émet la donnée. L'esclave positionne, ou lit, la donnée et informe qu'il est paré pour la suite en conservant TRDY. Le maître termine enfin l'échange, ou l'esclave

réclame la fin du transfert par STOP. La figure 1.8 illustre l'opération de base en écriture fournie par la spécification. La figure 1.9 illustre l'opération de base en lecture fournie par la spécification.

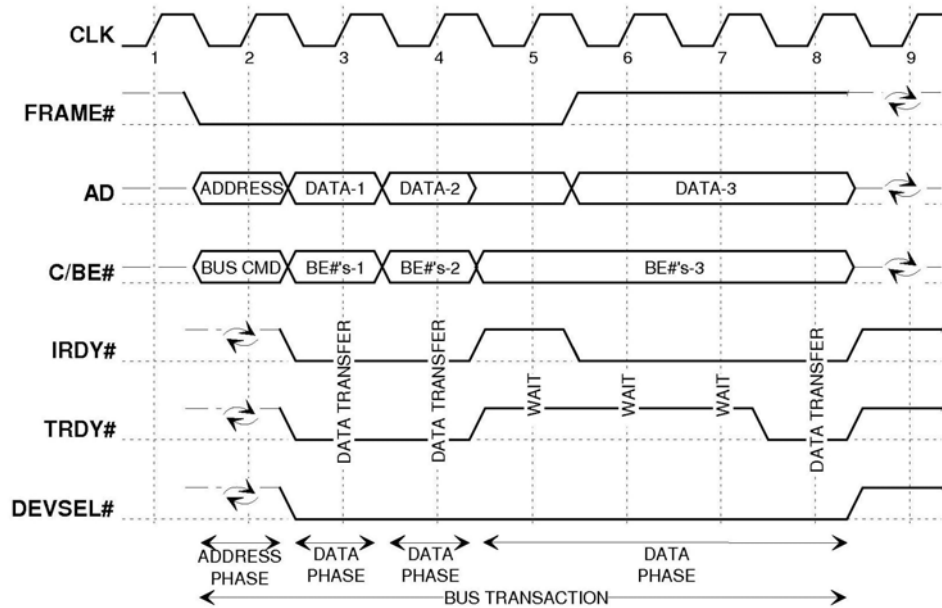


Fig. 1.8. Opération de base en écriture

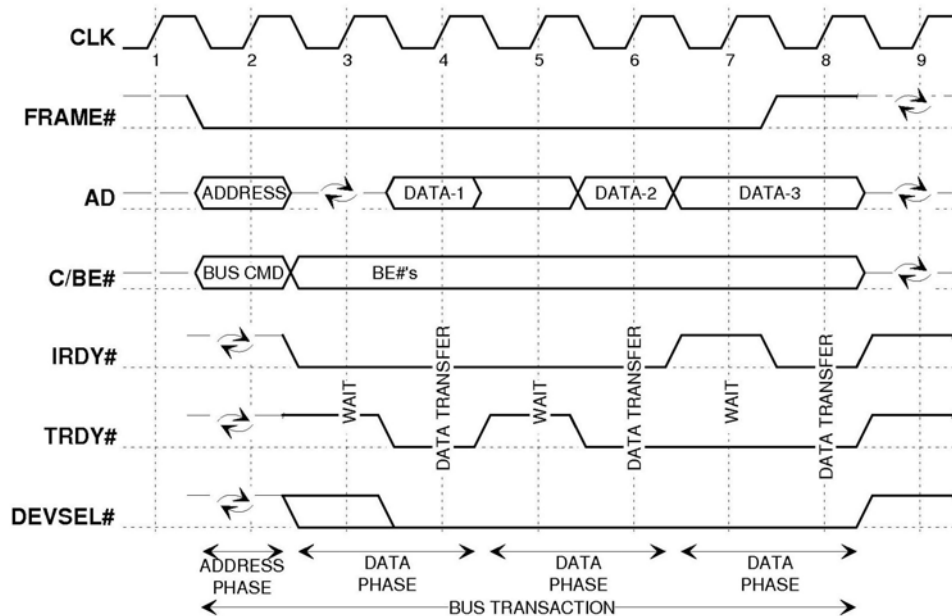


Fig. 1.9. Opération de base en lecture

Le bus PCI spécifie de nombreuses façons de mettre fin aux transferts de données. Ainsi, la transaction peut naturellement se terminer à l'initiative du maître, mais aussi bien suivant celle de l'esclave, avec ou sans l'échange de données.

Une autre fonctionnalité du bus PCI est de permettre des opérations différées. Ainsi, une transaction peut être initialisée par un maître, ce qui, du point de vue esclave, engage un traitement local. Seulement, l'esclave mémorise la commande reçue et demande la déconnexion. Il lui appartient ensuite de solder la tâche qui lui a été assignée. Enfin, le maître vient recontacter l'esclave et reçoit alors l'information d'acquittement de la tâche.

L'avantage de cette fonctionnalité est de permettre la libération du bus pour en permettre l'utilisation par d'autres maîtres. Ces transactions différées sont de cinq types :

- PWR est une écriture en mémoire postée. Une transaction qui se termine sur le bus d'origine avant d'aboutir sur le bus de destination.
- DRR est une requête de lecture différée. Une transaction qui doit se terminer sur le bus de destination avant d'être reçue sur le bus d'origine.
- DWR est une requête d'écriture différée. Une transaction qui doit se terminer sur le bus de destination avant d'être reçue sur le bus d'origine.
- DRC est un acquittement de lecture différée. Une transaction qui s'est terminée sur le bus de destination et qui remonte maintenant vers le bus d'origine.
- DWC est un acquittement d'écriture différée. Une transaction qui s'est terminée sur le bus de destination et dont l'acquittement remonte maintenant vers le bus d'origine.

Par ailleurs, le bus PCI autorise la réservation de ressource par l'intermédiaire d'un signal LOCK. Ainsi, un maître peut se réserver l'exploitation d'un esclave pour se garantir les résultats d'une tâche, vis à vis de l'intervention d'un autre maître. Après avoir réussi sa réservation, le maître est alors libre d'accéder à sa ressource et de prolonger, ou non, son exclusivité. La figure 1.10 présente l'opération d'accès exclusif sur PCI, illustre un maître ayant réalisé une réservation et accédant à sa ressource. En fin de cycle, il lui appartient de libérer, ou non, cet état.

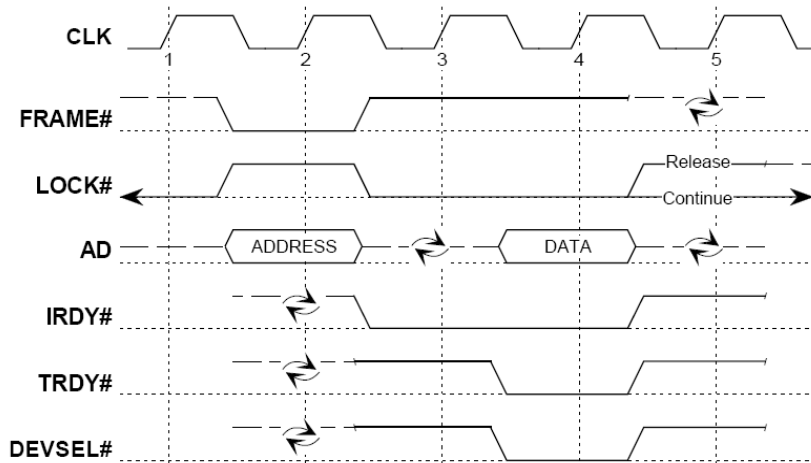


Fig. 1.10. Opération d'accès exclusif sur PCI

4.3. Bus PCMCIA

PCMCIA (Personal Computer Memory Card Interface Association) est aussi connu sous le nom de "PC card bus". Les versions sont nombreuses : version 1.0 en juin 1990 ; version 2.0 en 1991, version 2.1 en 1993 ; Cardbus en février 1995. Il s'agit d'un bus d'extension pour les

ordinateurs portables (extension de mémoire, modem, fax, disque dur, mémoire à bulles ...), les "notebooks", les équipements d'ordinateurs dans les voitures, les plates-formes mobiles.

Il comporte des fentes de taille triple des "cartes de crédit" 5.4 x 8.6 cm (2.126 " x 3.37"). Ce bus de 68 pins fonctionne aussi bien en mode 8 bits que 16 bits. Il supporte un domaine d'adressage de 64 Mo. Limité à 16 bits avec des vitesses d'horloge faibles, le débit maximum est estimé à 20Mo/s.

La version "Cardbus", nouvelle classe pour les "PC cards" et extension de PCMCIA à 32 bits, autorise des transferts jusqu'à 33 Mhz (132 Mo/s) et conduit ce bus vers les applications réseau à 100 Mbps, la vidéo en temps réel, le multimédia.

"PC Card" existe sous trois épaisseurs :

1. carte type I : 3.3 mm (typiquement pour les extensions mémoire)
2. carte type II : 5mm (typiquement pour les IO par exemple modem)
3. carte type III : 10.5 mm (typiquement pour les disques durs).

La version "carte type IV" de 18 mm d'épaisseur destinée aux disques durs à très grande capacité n'a pas encore été ratifiée.

Ces cartes sont majoritairement "Plug and Play", donc ne nécessitent pas la fermeture de l'ordinateur portable pour les installer. Vitesse plus élevée, facteur de forme plus petite, facilité d'usage et insertion "à chaud" (hot swap) ... sont parmi les critères retenus pour cette version. ExpressCard est conçu à la fois pour les PC de bureau et les portables. Il a la particularité de pouvoir jouer le rôle d'interface avec les liaisons USB 2.0 et PCI-Express. Cette souplesse augmente le nombre de périphériques compatibles. Deux formats sont prévus: l'ExpressCard/34 à très basse consommation et l'ExpressCard/54 que l'on retrouvera sur le PC de bureau et qui sera compatible avec les cartes mémoire actuelles.

4.4. Bus graphique AGP

Bien que ce type va probablement disparaître, il équipe néanmoins un grand nombre de systèmes actuels et il serait tout de même intéressant de le connaître et de pouvoir le comparer avec son remplaçant qui est le PCI-Express. Le bus AGP (Accelerated Graphics Port) annoncé en 1997 par Intel, a une vitesse d'horloge minimale de 66 Mhz, tolère des taux de transfert atteignant 266 Mo/s (32 bits). Même si un bus d'extension (comme le PCI-32 bits) peut atteindre 132 Mo/s, la conception d'un bus uniquement dédié à l'affichage graphique permet d'éviter le partage de la bande passante avec d'autres périphériques [5,6].

4.4.1. Architecture

Étant un bus dédié, d'autres améliorations à l'architecture deviennent possibles. Le bus AGP est ainsi capable d'aller chercher directement les données en mémoire vive, d'opérer des changements en mémoire (Direct Memory Execution) et de concevoir de nouvelles fonctionnalités. Comme exemple de la flexibilité de l'architecture AGP nous exposons deux architectures dans la figure 1.11. L'approche intégrée typique, des ordinateurs personnels (PC), est présentée dans la partie gauche de la figure. L'approche serveur multiprocesseur symétrique est présentée dans la partie droite de la figure.

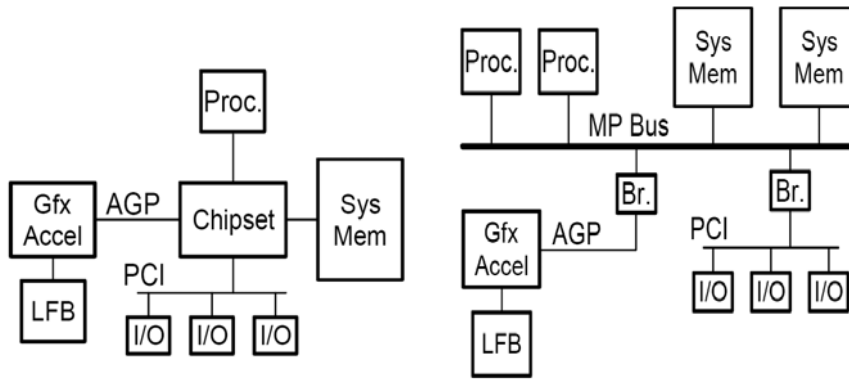


Fig. 1.11. Different Corelogic Architectures

Le "Side Band Signaling" est une bande latérale supplémentaire de 8 bits permettant à la carte AGP d'envoyer des requêtes successives à la carte CPU sans avoir à attendre l'achèvement des transferts sur le bus principal. Le "Graphic Address Remapping Table" permet de réaffecter dynamiquement les adresses de la mémoire vidéo et de la mémoire vive réservée à la carte graphique rendant le fonctionnement plus souple et évitant des transferts de relocalisation entre la mémoire centrale et la mémoire vidéo. La figure 1.12 montre un schéma bloc d'une architecture avec chipset 82845G de Intel® [11].

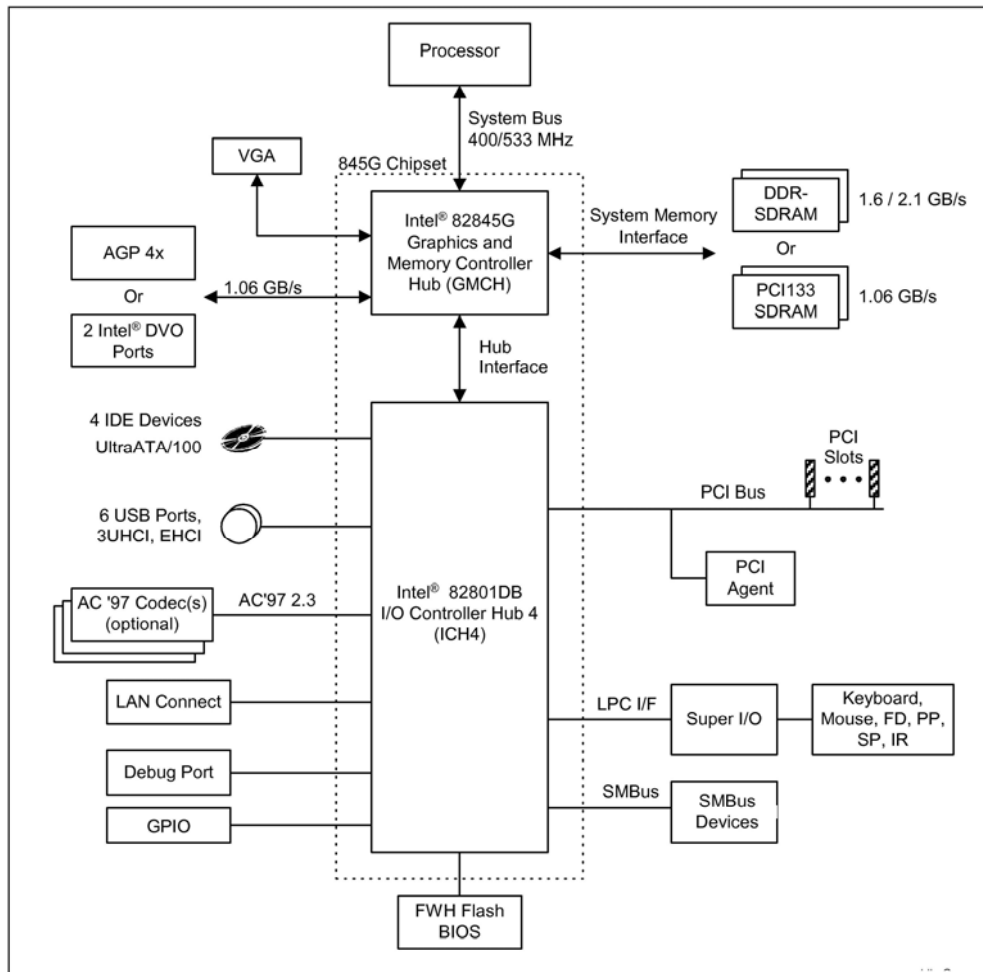


Fig. 1.12. Schéma bloc avec chipset 82845G de Intel®

4.4.2. Transfert de données

L'AGP 1X, offre un débit théorique de 266 Mo/s sur 32 bits (vitesse de 66 Mhz). La tension des signaux prend 3,3 V ou 1,5 V

Dans le mode AGP 2X, les données sont transférées sur le front montant, mais également sur le front descendant de l'horloge, multipliant ainsi le débit théoriquement par deux. L'architecture AGP tolère cependant une seule carte d'affichage ; les autres cartes d'affichage en mode "multi écrans" doivent passer normalement par le bus PCI. Les figures 1.13a et 1.13b montrent le concept de base du transfert A.G.P 1x et 2x. L'AGP 2X fonctionne à 66Mhz et offre une bande passante théorique de 528Mo/sec vers la mémoire centrale pour le transfert des images et des textures. La bande passante de la mémoire sur la carte graphique fonctionne, pour les cartes graphiques performantes, à des vitesses théoriques avoisinant 2Go/s.

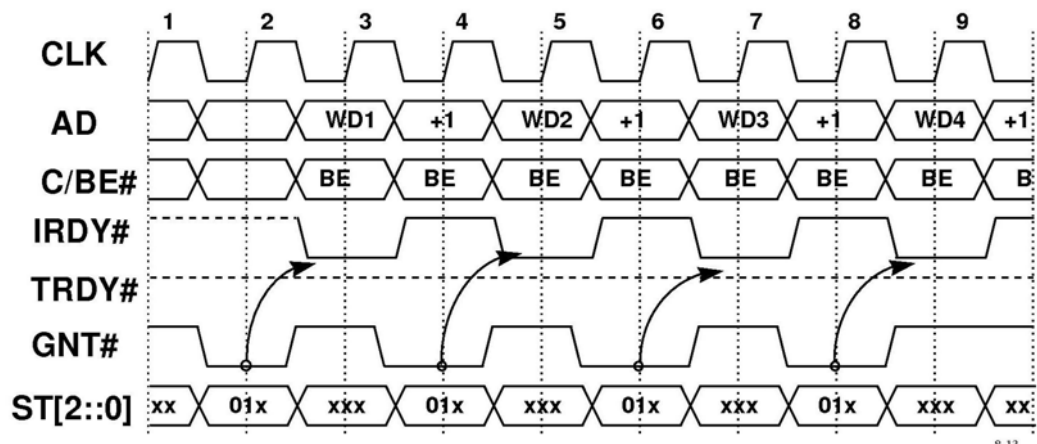


Fig. 1.13a. Concept de base du transfert A.G.P 1x.

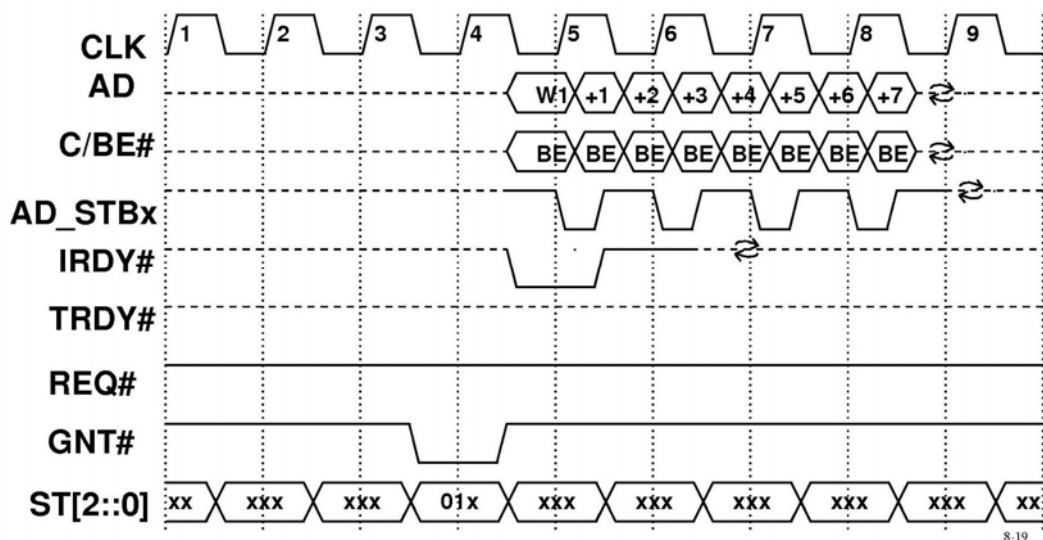


Fig. 1.13b. Concept de base du transfert A.G.P 2x.

L'**AGP 4X** naissant au milieu de 1998, prend son réel envol en 2000. Il transmet 4 données en un seul cycle d'horloge et offre une bande passante théorique plus intéressante environ 1.06 Go/s.

L'**AGP 8X** : Sorti réellement début 2003, l'**AGP 8X** reste en 32 bits mais à 266 Mhz. Certains chipsets supportent le 8X découlant de la version 3.0. NVidia, SiS et ATI sont les compagnies derrière cette nouvelle norme. La vitesse théorique de transfert théorique du bus AGP dépasse 2Go/s et la vitesse des échanges entre processeur graphique/mémoire locale est estimée à plus de 8 Mo/s. Seule réelle avancée, le transfert atteint en théorie jusqu'à 2,13 GB/s. En pratique, le gain de performance est inférieur à 5% par rapport à l'AGP 4X. La mémoire maximale sur les cartes graphiques reste aux alentours de 512 Mo.

4.5. Bus PCI Express

Le bus PCI Express (Peripheral Component Interconnect Express, noté 3GIO (Third Generation I/O), est un bus d'interconnexion permettant l'ajout de cartes d'extension dans l'ordinateur [12]. Mis au point en 2003 ce bus adopte la stratégie actuelle de "sérialisation" des données au lieu d'adopter le transfert parallèle, coûteux en termes de coût de connexion (beaucoup de fils, câbles encombrants).

Les connecteurs PCI Express avec leurs tailles variables sont incompatibles avec les anciens connecteurs PCI. Une des caractéristiques intéressantes du bus PCI Express est la possibilité de brancher ou débrancher des composants à chaud (hot plug) sans éteindre ou redémarrer l'ordinateur. Le connecteur PCI Express 1X possède 36 pins et est destiné à un usage d'entrées-sorties usuelles, le PCI Express 4X (64 pins) et 8X (94 pins) sont destinés aux serveurs et le PCI Express 16X à 164 pins est destiné à remplacer à courte échéance le bus AGP qui s'essouffle sur le plan de l'extension.

Le standard PCI Express prévoit supplanter la technologie PC Card des portables sous la forme de connecteurs « PCI Express Mini Card ». De plus, contrairement aux connecteurs PCI dont l'usage était limité à l'interconnexion des cartes à l'intérieur de l'ordinateur, le PCI Express permet de connecter des périphériques externes à l'aide de câbles. Il n'est pas en concurrence avec les ports USB ni FireWire bien que les périphériques peuvent adopter dès maintenant un choix ou un autre. Un bus PCI Express peut transférer des données dans les deux sens en même temps. L'infrastructure de l'architecture PCI Express diffère de beaucoup de celle du bus PCI. D'une part, le PCI Express utilise un schéma de transfert de données en série, qui exigent alors des fréquences d'horloge très élevées pour atteindre les mêmes performances qu'un bus parallèle. Plusieurs canaux (ou voies) PCI Express peuvent être ajoutés en jouant sur les versions 1X, 4X, 8X ou 16X.

Pour l'instant, le slot d'extension standard sur les futures cartes mères sera le PCI Express 1X (figure1.14). Dans ce cas, nous avons une voie PCI Express, soit une bande passante de 250 Mo/s, 500 Mo/s si l'on compte la bande passante dans les deux sens (on dit dans ce cas qu'il s'agit d'un mode de communication full-duplex dans lequel le transfert s'effectue en parallèle dans les deux sens simultanément). La vitesse juste pour une voie avoisine le PCI le plus performant. De plus, comme les canaux de communication sont plus ou moins réservés, chaque voie ayant l'intégralité des 250 Mo pour elle toute seule, on évite l'alternance et l'arbitrage coûteux à la fois en temps et en logique de gestion.

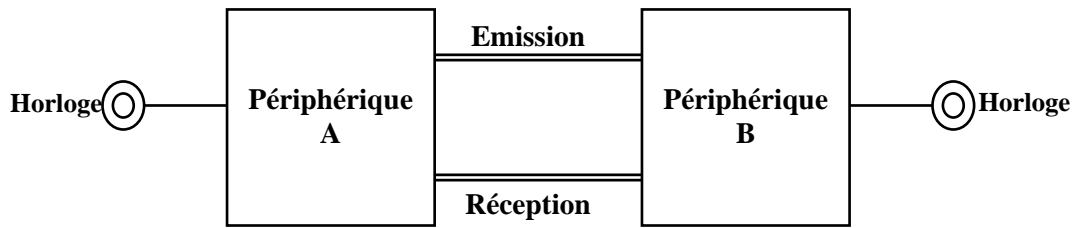


Fig. 1.14. Interconnexion des périphériques PCI Express

Les cartes graphiques utilisent le slot PCI Express 16X, ce qui équivaut à 16 voies, soit une bande passante maximale de 4 Go/s, ou 8 Go/s si l'on compte les 4 Go/s dans les deux sens (le full-duplex est un peu du marketing car les transactions sur les cartes graphiques ne sont pas symétriques). Même avec 4Go/s, on double déjà la vitesse en comparaison avec le AGP 8X avec son 2Go/s. La figure 1.15 montre un schéma bloc d'une architecture avec chipset 829155P de Intel® [13].

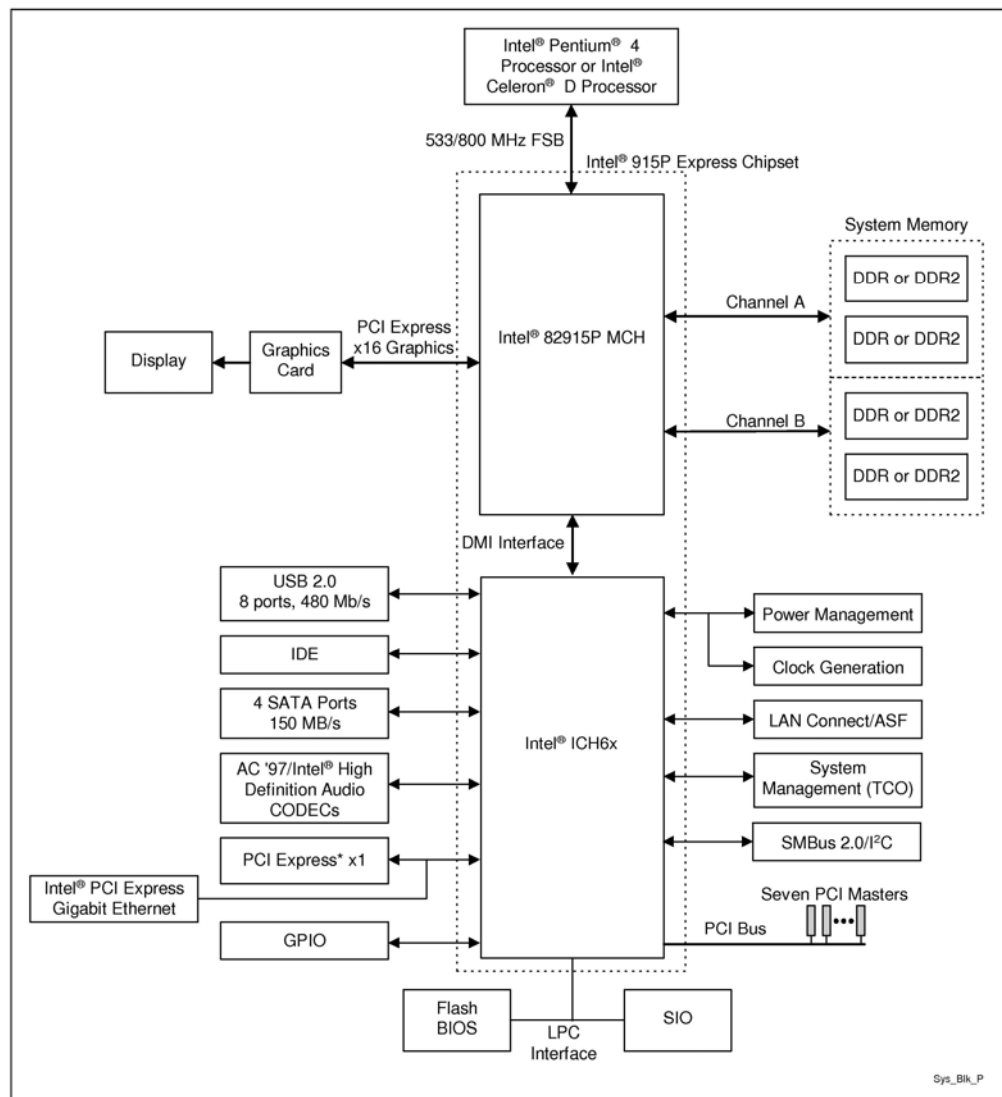


Fig. 1.15. Schéma bloc avec chipset 82915P de Intel®

5. Ports et Connecteurs PC.

5.1. L'interface SCSI

L'interface SCSI "Small Computer System Interface" est intéressant par son degré de connectivité et de la diversité des plates-formes qu'il supporte (il connecte les disques durs, les disques souples, les disques amovibles, les CD-ROM, tous les disques optiques, les rubans, les cassettes, les scanners, les imprimantes ...). Ce bus supporte les plates-formes PC, Mac, Sun, IBM, HP, DEC ...

Le standard SCSI définit les paramètres physiques et électriques. Le standard SCSI-1 original (ANSI X3.131-1986) a été adopté en 1986 ; le SCSI-2 a été adopté en janvier 1994. La plupart des fabricants indiquent que leurs cartes contrôleurs sont conformes tant au standard SCSI-1 qu'au standard SCSI-2 (ANSI X3.131-1994). A titre d'exemple, le standard SCSI-2 prévoit en option un mode synchronisé rapide qui permet de doubler le taux de transfert de données, le portant de 5 à 10 Mo/s. Ce mode de transfert rapide, appelé Fast SCSI, peut être combiné à une interface Wide SCSI 16 bits pour obtenir un taux de transfert atteignant 20 Mo/s.

Le tableau 1 présente le débit maximal des bus SCSI suivant leur vitesse et leur largeur, et il indique quel câble il faut utiliser suivant le taux de transfert concerné.

TABLEAU 1.1

Taux de transfert et largeur de données du standard SCSI

Termes définis par SCSI Trade Association	Vitesse en Mo/s	Largeur en bits	Longueur max. Câble (m)			Nombre max. de "devices"
			SE	DIF	LVD	
<i>SCSI-1</i>	5	8	6	25	12	8
<i>Fast SCSI</i>	10	8	3	25	12	8
<i>Fast Wide SCSI</i>	20	16	3	25	12	16
<i>Ultra SCSI</i>	20	8	1.5	25	12	8
<i>Ultra SCSI</i>	20	8	3	25	12	4
<i>Wide Ultra SCSI</i>	40	16		25	12	16
<i>Wide Ultra SCSI</i>	40	16	1.5			8
<i>Wide Ultra SCSI</i>	40	16	3			4
<i>Ultra2 SCSI</i>	40	8			12	8
<i>Wide Ultra2 SCSI</i>	80	16			12	16
<i>Ultra160 SCSI</i>	160					

SE : Single Ended. Câble de signaux à simple terminaison. DIF : Differential Drivers. Câble différentiel.
LVD: Low Voltage Differential Drivers.

5.2. IDE ou ATA

Malgré l'appellation officielle "ATA", ce standard est plus connue sous le terme commercial IDE (Integrated Disk Electronics) ou EIDE (Extended ou Enhanced IDE). IDE et Enhanced IDE (EIDE ou E-IDE) sont spécifiques au monde des PC.

IDE est encore appelé ATA (AT Attachment) référant à l'ancien PC-AT mis au monde par IBM, ATA-1 ou encore ATA IDE. Il s'agit d'un bus spécialisé de 40 pins destiné à connecter le CPU à deux disques durs.

En 1993, pour prolonger l'existence de cette interface peu coûteuse, l'industrie du disque inventa le EIDE (ATA-2) qui supporte 4 unités mélangées de disques durs et de CD-ROM sous forme de deux câbles séparés (pour la compatibilité) ; chaque câble supporte un "maître" et un "esclave". Ceci permet de soutenir des transferts plus rapides de type DMA et PIO (Processor Input/Output). Dans un transfert PIO, le CPU s'occupe du transfert entre l'interface et la mémoire. Sur un bus ATA-2, on peut attacher un CD-ROM ou un support de stockage de masse quelconque (cassette, dérouleur de bandes...) à travers un protocole appelé ATAPI (ATA Packet Interface).

Vers la fin de 1997 apparaît le Ultra DMA, ATA-4 ou Ultra ATA ou Fast ATA-2. Le nouveau bus Ultra DMA garde les mêmes paramètres physiques et électriques que ATA-2 mais porte la vitesse de transfert vers 33.3 Mo/s. Le miracle vient de trois améliorations : un meilleur arrangement des signaux, un pipelining des signaux et un transfert synchrone. Cette nouvelle version ATA-4 de l'interface offre en plus un contrôle CRC (Cyclic Redundancy Check) permettant d'assurer une meilleure intégrité des données transférées. La liste suivante résume les vitesses de cette famille de bus :

- PIO mode 0 (3.3 Mo/s. ATA-1)
- PIO mode 1 (5.2 Mo/s. ATA-1)
- PIO mode 2 (8.3 Mo/s. ATA-1)
- PIO mode 3 (11.1 Mo/s. ATA-2)
- PIO mode 4 (16.6 Mo/s. ATA-2)
- DMA mode 1 (13.3 Mo/s. ATA-2)
- DMA mode 2 (16.6 Mo/s. ATA-2)
- Ultra DMA (33.3 Mo/s. Ultra ATA, Fast ATA-2 ou ATA-4).

Les signaux du bus ATA-1 peuvent dériver à partir d'un bus d'extension comme ISA. ATA-2 et ATA-4 doivent être pris à partir du bus d'extension PCI à cause de leur bande passante plus élevée.

Cette famille a survécu et a su évoluer grâce à sa simplicité et à son côté "bon marché". En 1998, le bus ATA-4 est capable de livrer toute la performance des disques durs lorsqu'on n'a qu'un ou deux disques durs dans le système et que l'on ne demande pas à tous les périphériques de fonctionner ensemble. Dans le cas où l'interface d'un des deux éléments est supérieure à l'autre, l'élément le plus récent va automatiquement s'ajuster sur l'interface la moins puissante utilisée par l'autre élément afin d'offrir une compatibilité parfaite.

Vers l'année 1999, cette interface subit encore une mutation réellement importante qui le place cette fois-ci, en termes de vitesse de transfert, à la même classe que le bus SCSI. Les compagnies comme CMD ou Promise ont su fabriquer un autre ultra DMA qui est capable de transférer jusqu'à 66 MHz, 2 fois plus que la vitesse du ultra DMA de base qui est l'ultra DMA 33.

À court de qualificatif, les compagnies encodent directement la fréquence du bus dans l'appellation, ce qui donne ultra DMA 66, ultra DMA 100, ultra DMA 133. Avec le ultra DMA 133 par exemple, on est très loin du vieux ultra DMA 33. Cette classe de port IDE est destinée aux disques durs rapides avec des vitesses de rotation de 7500 tours/mn, 10000 tours/mn, voire plus.

Comme point de repère, un disque dur en 2003 avec une vitesse de rotation de 7200 tours/mn a environ une vitesse de transfert intrinsèque avoisinant 50 Mo/s. Comme les disques modernes ont des "caches" de taille importante (environ 8 Mo et plus), la mémoire cache absorbe localement le flot de données venant des pistes du disque puis effectue un transfert en mode rafale vers l'interface disque Ultra DMA situant du côté du processeur. La vitesse de transfert au niveau du bus dépasse largement 50 Mo/s, ce qui justifie la présence du DMA-133.

Avant de connecter un tel disque, il faut vérifier que le chipset de la carte mère ainsi que le système d'exploitation supportent un DMA-133 pour être à la hauteur des disques durs performants.

5.3. Serial ATA

La vitesse théorique du SATA est de 150 Mo/s. Des évolutions comme le Serial-ATA II (Serial-ATA 3 Gb/s) mènent la vitesse maximale théorique de 150 Mo/s à plus de 300 Mo/s soit grosso modo 3 Gb/s. Techniquement, les disques durs Serial-ATA 3 Gb/s sont annoncés comme compatibles avec les contrôleurs Serial-ATA. Le SATA reste une interface interne ce qui le différencie des interfaces comme le USB 2 ou le Firewire.

L'interface Serial ATA (SATA) n'a que 7 fils pour transporter les données, comparé aux 80 fils de l'Ultra DMA, ce qui explique les avantages de cette nouvelle interface (légèreté, encombrement, souplesse du fil de connexion). Les câbles peuvent avoir une longueur de 1 mètre, comparé au 0,46 mètre de l'Ultra DMA (certains câbles Ultra DMA trouvés dans le commerce mesurent 0.7 m à 0.8 m. Ils sont hors norme et ne fonctionneront pas avec les fréquences élevées).

Aujourd'hui, des chipsets comme NVIDIA nForce 4 Ultra, Intel i945G, Intel i945P et Intel i955X supportent le Serial-ATA 150 et le Serial-ATA 3Gb/s. En effet, les constructeurs de disques durs choisissent de désactiver le mode de transfert SATA 3Gb/s sur ses disques durs Serial-ATA II pour assurer la compatibilité. Pour profiter du SATA II, il faut activer cette option.

5.4. Port série RS-232

Cette interface est normalisée par EIA (Electronic Industries Association) et par CCITT (Comité Consultatif International Téléphonique et Télégraphique). C'est un standard réglementant au départ la communication asynchrone entre un DCE (Data Communications Equipment) du type modem et un DTE (Data Terminal Equipment) du type télétype ou ordinateur.

Par la suite, la simplicité de ce standard incite les constructeurs à utiliser un sous-ensemble de ce standard pour d'autres types de communication (entre une imprimante et un ordinateur,

un ordinateur et un clavier, un ordinateur et un crayon optique, un ordinateur et une souris, entre un ordinateur et un autre ordinateur ...).

Lorsque la communication s'établit entre deux ordinateurs il n'est plus possible de distinguer DCE et DTE. Le standard ne s'applique plus, et il faut suivre les recommandations du constructeur du matériel spécifique.

5.5. Port parallèle

Le bus parallèle a été utilisé initialement dans le monde des ordinateurs personnels pour connecter l'ordinateur à une imprimante. L'interface parallèle constitue, comme l'interface série, l'élément de base d'un ordinateur. À travers les diverses générations d'ordinateurs, ces deux interfaces ont défié l'usure du temps, malgré les diverses propositions alternatives comme USB (Universal Serial Bus). Elles sont solidement installées et permettent de connecter, sans ouvrir le "capot" de l'ordinateur, un grand nombre de périphériques à l'ordinateur ou deux ordinateurs ensemble.

À la différence des interfaces série, les interfaces parallèles effectuent le transfert des données sur un bus parallèle. Tous les bits sont donc disponibles simultanément, d'où une vitesse élevée de transfert variant entre 400 Kbps et une dizaine de mégabits par seconde.

L'usage de l'interface parallèle est très étendu. Moyennant des adaptations électroniques, les interfaces parallèles peuvent constituer des commandes de relais, des commandes de moteurs, des interrupteurs fonctionnant à très haute vitesse.

L'interface parallèle communique avec le CPU selon les mêmes principes que l'interface série. Du côté du bus spécifique, on retrouve les données sous forme de lignes parallèles qui sont munies d'amplificateurs de courant capable de débiter un courant important. Cette caractéristique très intéressante épargne souvent du temps au concepteur le soin de fabriquer lui-même ses "buffers" de ligne.

Sur la figure 1.16, on remarque que les bus spécifiques de l'interface parallèle disposent d'un très grand nombre de lignes parallèles (8, 16, 24, 32 ...). Ces lignes sont regroupées en "ports" A, B, C, D... pour faciliter la programmation. Ces lignes sont bidirectionnelles, ce qui veut dire que chaque ligne est capable d'agir à la fois comme entrée ou sortie par programmation (non en même temps). Pour chaque bit de données de l'interface parallèle, on a un bit programmable qui définit le sens de circulation des données. Il s'agit du bit de direction (Data Direction bit). L'interface possède des lignes supplémentaires gérant les protocoles de transfert, à la fois en transmission ou en réception sur ses ports.

La versatilité d'une interface parallèle est étonnante. Si on veut contrôler une interface parallèle autrement que par les lignes de dialogue qu'elle possède, il suffit de prendre n'importe quel port, puis d'implanter la logique de contrôle à votre guise. Vous pouvez, par exemple, prendre une interface parallèle et programmer tous les signaux de contrôle et de données de l'interface série RS-232C comme TD, RD, CD, RTS, CTS, DTR ... étudiée dans l'étude de la liaison série (bien sûr, il faut effectuer une adaptation électrique pour transformer les tensions TTL en tensions -24 volts, + 24 volts, mais la logique peut être facilement implantée).

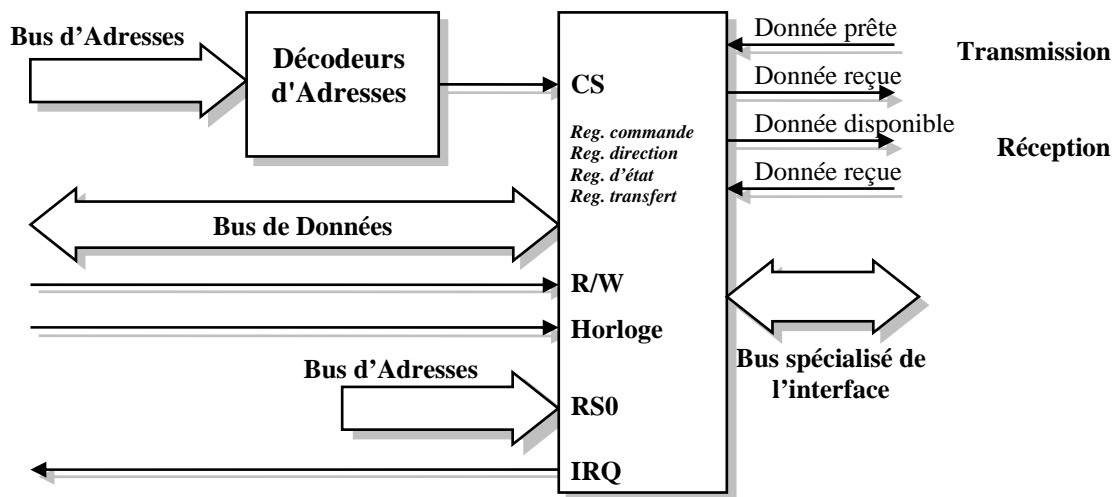


Fig. 1.16. L'interface parallèle

Registres généraux d'une interface parallèle

Les registres généraux d'une interface parallèle sont :

- **Registres de commande.** Ils permettent de configurer le mode de fonctionnement général de l'interface (type de poignée de main ; front montant ou front descendant des signaux qui détermine la disponibilité des données ; fonctionnement par interruption matérielle ou par inspection du registre d'état par le processeur...).
- **Registres de direction.** Ils déterminent par programmation le sens de chaque bit individuel sur les ports. Chaque bit de port peut agir comme entrée ou sortie. Il y a autant de bits de direction que de bits de port.
- **Registre d'état.** Il détermine la disponibilité des données à transmettre ou à recevoir, avertit le processeur des erreurs de communication ou des problèmes potentiels sur les lignes de contrôle de l'interface, donne les états internes de fonctionnement de l'interface.
- **Registres de transmission ou de réception des données.** Le processeur lit les données qu'il reçoit dans les registres de réception et dépose les données qu'il transmet dans les registres de transmission.

5.6. Port USB

USB (Universal Serial Bus), développé en 1995 par Intel, Compaq, Hewlett-Packard, Microsoft, NEC & Philips, veut être le port universel capable, comme son nom l'indique, de relier un ordinateur aux appareils les plus divers, et ceci sans ouvrir l'ordinateur: manette pour jeux vidéo, numériseur (scanner), haut-parleurs digitaux, appareil photo digital, imprimante, modem, fax, souris, connexion Ethernet, autre ordinateur, etc. Ce bus ou port USB peut remplacer les deux ports classiques, série et parallèle, pour la majorité des applications à basse et moyenne vitesse (le port USB dans sa version 1.1 est limité à 12 Mbps). On peut connecter jusqu'à 127 appareils sur le même port [14].

Les nouveaux appareils conçus pour ce port (USB compliant) doivent respecter les principes de base suivants: plus de microcontacts (dip-switchs), plus d'interruption IRQ à régler, plus d'adresses de registres de base à préciser (en réalité, tous les USB partagent ensemble une interruption et un ensemble de registres configurés au préalable). Tous les réglages doivent être automatisés pour faciliter la vie de l'utilisateur peu averti.

Un autre intérêt réside dans sa capacité à accepter des connexions "à chaud" (hot swapping ou hot plugging), c'est-à-dire qu'on peut brancher ou débrancher sans éteindre l'ordinateur et, surtout, sans rebooter ce dernier.

Dans la mesure du possible, les alimentations électriques de la majorité des appareils qui sont connectés sur ce port sont fournies par le port lui-même (touts du moins pour les consommations inférieures à 2.5 watts). Les valeurs des tensions sont obtenues par une procédure standard de détection automatique (sensing). On s'affranchit ainsi de petits blocs d'alimentation "pendants" de 6V, 9V, 12V ... accompagnant souvent un répondeur téléphonique, un mini lecteur de CD, etc. Réellement, on désire instituer une nouvelle connexion, aussi facile à utiliser qu'un jack téléphonique dans un mur.

Dans le monde PC, ce port est supporté seulement par Windows 98 ; Windows 95 et WinNT l'ignorent, Windows 2000 et XP le supporte en standard. L'arborescence des connexions est assurée par des noeuds (hubs) qui agissent un peu comme les cordons d'extension pour les alimentations électriques dans les foyers.

La version 2 porte la vitesse à 480 Mbps ou 60Mo/s. USB 2.0 est donc 40 fois plus rapide que USB 1.1) [15]. Les premiers appareils apparaissaient sur le marché au 2e semestre de l'an 2000.

Bien que USB soit un port plus lent que SCSI dans sa version originale (la version 2 permet même avec la connexion avec des disques durs), il serait intéressant d'établir une comparaison puisque ces deux ports prétendent connecter à peu près les mêmes périphériques :

1. L'USB peut accepter 127 périphériques (adresse codée sur 7 bits), alors que le bus SCSI ordinaire est limité à 7
2. Dans les deux cas, les périphériques sont montés en feston sur le bus. Avec l'USB, il faut segmenter le bus avec des hubs quand le nombre de périphériques augmente. La structure de la connexion est arborescente avec l'USB alors qu'elle est linéique avec le SCSI
3. Avec USB, l'adressage est dynamique. C'est le contrôleur du bus qui se charge de l'opération d'affectation de l'adresse. L'utilisateur est obligé d'attribuer un numéro à chaque périphérique SCSI
4. Pour brancher un périphérique sur un bus SCSI, il faut arrêter tout le système. L'USB, par contre, est "hot plug" : on peut brancher ou débrancher un périphérique alors que l'ordinateur et les autres périphériques sont en fonctionnement. Le contrôleur détecte automatiquement que le périphérique vient d'être mis en service ou qu'il vient d'être retiré. Dès qu'il est détecté, le périphérique est automatiquement configuré

5. Sur un bus SCSI, deux périphériques peuvent échanger de l'information entre eux, sans que l'unité centrale soit impliquée. Sur un bus USB, ceci n'est pas possible. Le contrôleur, qui se trouve à la tête de l'arbre, fonctionne toujours en maître, et les périphériques en esclaves
6. L'USB est très économique alors que le SCSI est très onéreux. La plupart des ordinateurs arrivaient avec deux USB avant 2001 (6 à 8 actuellement). Quand il faut connecter plusieurs périphériques, il faut acquérir un hub.

Au début de son apparition, la pénétration de USB dans le marché était très lente. La raison réside dans le fait que l'USB n'apporte aucun plus à l'utilisateur lorsqu'il s'agit de connecter des périphériques courants à l'ordinateur. Toutes les machines possédaient déjà leur connecteur "derrière" l'ordinateur. Le clavier et la souris avaient déjà leur propre entrée, l'imprimante son port parallèle, l'Internet sa propre carte réseau.

À partir de 2000, son taux de croissance s'avère bien meilleur avec la baisse des prix des numériseurs et des appareils photos et des caméras numériques. La version 2.0 de USB porte ce bus presque parfait à la vitesse de 480 Mbps. Ce bus est réellement un bus d'avenir par sa facilité d'emploi, ses performances étonnantes.

De nos jours, avec l'arrivée des mémoires Flash performantes, ce bus remplace presque le marché des disques amovibles.

5.7. Port FireWire (IEEE 1394)

Développé par Apple, le bus FireWire a été adopté comme norme internationale sous l'appellation IEEE 1394 pour la vidéo numérique à la fin de l'année 1995. Le port FireWire ou IEEE-1394 est un bus série à très haut débit. Ses caractéristiques lui permettent d'envisager même la connexion avec les disques durs de moyenne vitesse. Le taux de transfert peut atteindre 400 Mbits/s (50 Mo/s).

Le bus FireWire peut gérer plusieurs flux à des vitesses différentes. L'avantage sur l'USB est la possibilité de "réservation" pour garantir un débit constant (optimisation de la bande passante). Il est possible de chaîner jusqu'à 63 périphériques (caméscopes, DVD...) rapides. Le bus FireWire peut, éventuellement, fournir de l'énergie aux appareils connectés (24 volt - 15 Watt max).

Les périphériques sont reliés entre eux par un câble souple (3 paires torsadées) d'une longueur maximale de 4,50 mètres. FireWire supporte également le hot plug, le plug and play et ne requiert aucun adressage. FireWire n'exige pas de maître et permet aux périphériques de dialoguer entre eux. Les caméras Sony par exemple peuvent transférer des données entre elles sans nécessiter un ordinateur. Sony commercialise le FireWire sous le nom de iLink.

Adoptés en 2001 avec une vitesse de 400 Mbits/s, FireWire 2 (IEEE 1394b) atteint 800 Mbits/s (100 Mo/s) puis 1,6 Gbits/s (200 Mo/s) et 3,2 Gbits/s (400 Mo/s) sur une connexion en fibre optique plastique supportant une distance de 100 mètres.

6. Conclusion

Les fréquences de fonctionnement des processeurs et des mémoires ont considérablement augmenté. Certes, la bande passante du bus PCI-X a permis d'augmenter la fréquence du bus PCI en passant de 33 à 66, 100 ou encore 133 MHz. Malheureusement, cette augmentation de la bande passante est théorique et s'accompagne d'une réduction importante de la distance sur laquelle les données peuvent être acheminées et du nombre de connecteurs que les transmetteurs de bus peuvent gérer.

Les applications comme l'acquisition de données, la génération de formes d'ondes, le *streaming* audio et vidéo nécessitent une bande passante garantie et des temps d'attente déterministes. Les spécifications du PCI ne répondent pas à ces besoins car ces applications n'étaient pas déterminantes au moment du développement du bus. Les applications de transfert de données en temps réel, gourmandes en bande passante et avec un temps d'attente critique, qui se développent aujourd'hui en vidéo, exigent de supporter aussi ce type de transfert.

L'AGP l'a d'ailleurs déjà remplacé pour accroître les performances du circuit vidéo, tout en évitant de saturer la bande passante que se partagent les autres composants connectés sur le bus du PC. On est ainsi passé de 132 Mo/s (32 bits/33 MHz) avec le PCI à des débits théoriques successifs de 266 Mo/s, 533 Mo/s, 1,04 Go/s et 2,13 Go/s avec respectivement l'AGP 1X, 2X, 4X, 8X.

En pratique, le gain de performance de débit est considérablement inférieur à la valeur théorique. Par exemple : le gain théorique, entre l'AGP 8X et l'AGP 4X, est de 100% et le gain pratique est inférieur à 5%. Cette situation impose une nouvelle réflexion architecturale pour l'interfaçage graphique temps réel et qui prend en considération l'augmentation de la bande passante pratique avec la réduction des temps d'attentes. Les chapitres qui suivent présentent des nouvelles solutions adaptées pour l'interfaçage graphique dans les systèmes à microprocesseurs.

Chapitre 2

**Nouvelles architectures
logicielle/matérielle
d'interfaçage**

1. Introduction

Dans un système à microprocesseur ou dans un ordinateur, tous les périphériques et les circuits auxiliaires sont reliés au bus système d'une manière ou d'une autre [16,17], et chaque périphérique occupe une adresse physique ou une plage d'adresses. Le bus système est composé d'un bus de données, d'un bus d'adresses et d'un bus de contrôle. Pour que le processeur effectue une communication avec une carte, il faut qu'il mette sur le bus d'adresses l'adresse physique de cette carte. Chaque carte est généralement équipée d'un décodeur d'adresse et d'un amplificateur de bus qui se validera dès que l'adresse de cette carte sera disponible sur le bus système Fig. 2.1.

Pour les cartes qui occupent une plage d'adresses, le décodeur de ces cartes doit valider l'amplificateur de bus pour toute la plage réservée à ce périphérique. Ce mécanisme explique pourquoi l'installation de cartes d'extensions provoque parfois des problèmes de conflit: il arrive que deux cartes revendiquent le même domaine d'adressage ou des domaines qui se chevauchent [18,19].

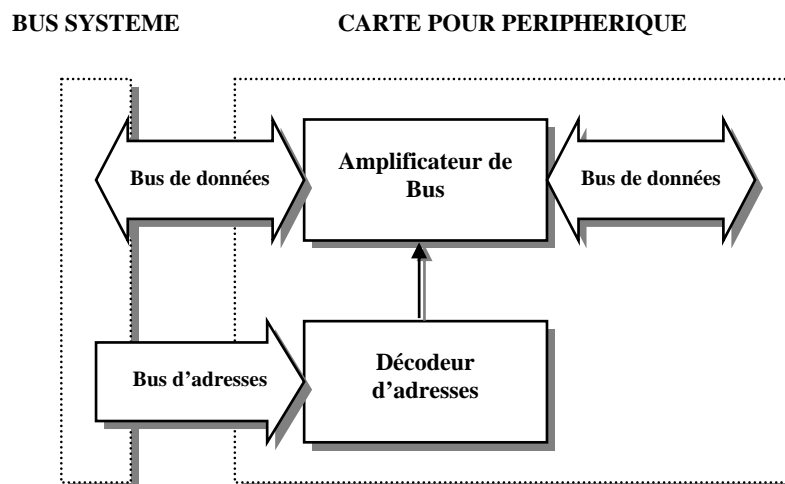


Fig. 2.1. Principe de l'interfaçage des périphériques.

Dans cette partie nous proposons des nouvelles techniques d'adressage matériel pour l'échange de données entre les systèmes à microprocesseur et des périphériques externes. Les nouvelles techniques présentent des solutions logicielles/matérielles pour augmenter la capacité d'adressage physique des ordinateurs et des systèmes à microprocesseurs. Ces solutions consistent à créer un nouveau bus, composé d'un bus de données, d'un bus d'adresses et d'un bus de contrôle. L'architecture proposée est constituée d'une partie matérielle et d'une partie logicielle ; la première est composée d'un nouveau bus et d'une interface entre le bus système et le nouveau bus. La partie logicielle assure la communication entre l'ordinateur ou le système à microprocesseur et la partie matérielle.

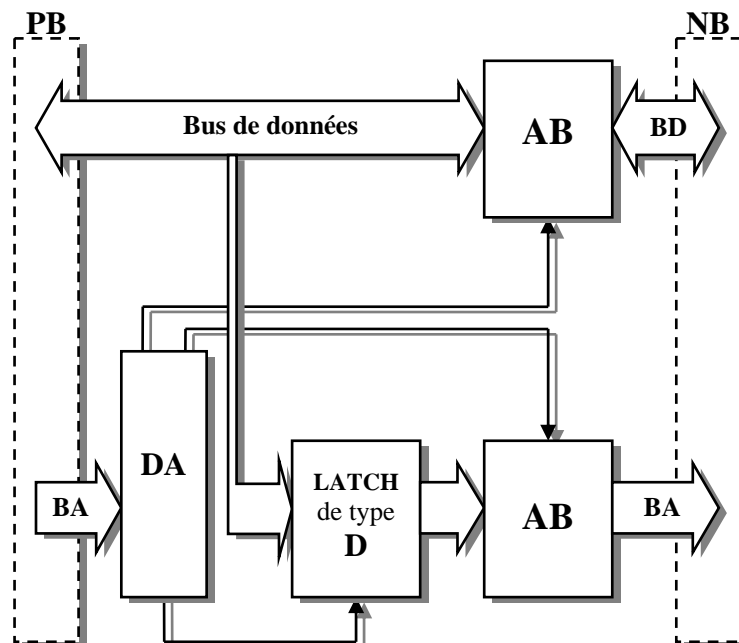
Nous exposons quatre nouvelles techniques d'interfaçage dans les systèmes à microprocesseur : l'Adressage Physique Etendu, l'Adressage Physique Accéléré, l'Adressage Physique Rapide et l'Adressage Physique Avancé.

2. Adressage Physique Etendu

2.1. Principe de l'Adressage Physique Etendu

Le principe de l'Adressage Physique Etendu [20] est basé sur une architecture mixte [21] logicielle/matérielle qui vise l'élargissement de la capacité d'adressage matériel des ordinateurs et des systèmes à microprocesseur. Nous avons utilisé un nombre limité d'adresses de la zone d'adressage du système pour adresser un espace mémoire externe plus important. La partie matérielle de notre système est composée d'un nouveau bus et d'une interface entre le bus système ou le bus d'extension et les périphériques qui vont être adressés par cette technique.

L'entrée de notre interface sera connectée au bus du système à microprocesseur. La sortie, qui est un nouveau bus, sera connectée à un périphérique externe. Le synoptique de base de la partie matérielle est illustré par la Fig. 2.2. Le nouveau bus contient un bus de données et un bus d'adresses. Dans le type d'adressage que nous proposons, les valeurs des adresses disponibles sur le nouveau bus dépendent des données envoyées sur le bus de données du bus système. Le bus de données assure la canalisation des données proprement dites, et la canalisation des données destinées à être converties en des adresses pour le nouveau bus. La partie logicielle de notre système assure l'émission des deux types de données en deux étapes et sur des adresses distinctes. Le décodeur d'adresses de la partie matérielle de l'Adressage Physique Etendu assure la séparation entre les deux types de données.



AB : Amplificateur de Bus, BA : Bus d'Adresses, BD : Bus de Données, DA : Décodeur d'Adresses.
NB : Nouveau Bus, PB : Premier Bus (Bus Système ou Bus d'Extension).

Fig. 2.2. Synoptique de l'Adressage Physique Etendu du premier ordre

Le mécanisme de cette technique passe par deux phases. Dans la première phase, le logiciel assure la présence des données destinées à être disponibles sur le nouveau bus comme étant des adresses. Le décodeur assure l'activation du circuit «LATCH» de type D pour enregistrer ces données jusqu'à la deuxième phase et la désactivation des deux

amplificateurs de bus. Les lignes d'adresses et de données du nouveau bus seront à l'état haute impédance. Dans la deuxième phase, le logiciel assure la présence des données proprement dites, et le décodeur d'adresses active les deux amplificateurs de bus. Les valeurs des données sur le nouveau bus sont identiques aux données du bus système de la deuxième étape et les valeurs des adresses sont identiques aux données de la première étape du bus système.

La figure 2.3 représente l'organigramme de base de la procédure logicielle. Le décodeur d'adresses est conçu pour réaliser deux tâches, la première consiste à actionner le circuit registre afin de mémoriser le contenu du bus de données et de le faire véhiculer à l'entrée de l'amplificateur de bus jusqu'à la prochaine validation de **adr1**. La deuxième phase débute par la présence des données destinées à être disponibles sur le nouveau bus, assurées par le logiciel. Ces données sont validées sur le bus système à l'adresse **adr2** (l'adresse doit être différente **adr1 # adr2**). La deuxième tâche réalisée par le décodeur consiste à valider les deux amplificateurs de bus si l'adresse **adr2** est validée sur le bus d'extension. Dans cette deuxième phase les valeurs des deux bus de données et d'adresses sont valides sur le nouveau bus.

Pour l'interfaçage avec l'Adressage Physique Etendu du premier ordre nous utilisons deux adresses physiques à savoir, une adresse pour la tâche de conversion donnée/adresse et une autre pour la canalisation des données.

L'utilisation de deux adresses sur un bus système qui possède un bus de données de N bits et une fréquence de travail F , conduit à un nouveau bus qui fonctionne à une fréquence de $F/2$ et qui possède un bus de données de N bits et une capacité d'adressage physique de 2^N . Si le système du premier ordre est appliqué sur un bus de 33 MHz et de 16 bits de données, le nouveau bus fonctionnera avec une fréquence maximale de 16,5 MHz et sur un bus de données de 16 bits et avec une capacité d'adressage de 65536. Le tableau 2.1 représente les caractéristiques de base de l'Adressage Physique Etendu du premier ordre.

TABLEAU 2.1

Caractéristiques de l' Adressage Physique Etendu du premier ordre

<i>FI</i>	<i>NDS</i>	<i>NAS</i>	<i>NDN</i>	<i>NAN</i>	<i>FO</i>
<i>F</i>	<i>8 bits</i>	<i>02</i>	<i>8 bits</i>	<i>256</i>	<i>F/2</i>
<i>F</i>	<i>16 bits</i>	<i>02</i>	<i>16 bits</i>	<i>64 K</i>	<i>F/2</i>
<i>F</i>	<i>32 bits</i>	<i>02</i>	<i>32 bits</i>	<i>4 G</i>	<i>F/2</i>

FI : Fréquence de travail du bus système, NAN : Nombre d'adresses physiques disponibles sur le nouveau bus.
 NAS : Nombre d'adresses physiques occupées sur le bus système, NDN : Taille des données sur le nouveau bus.
 FO : Fréquence de travail du nouveau bus, NDS : Taille des données sur le bus système.

Le chronogramme de la procédure logicielle de l'Adressage Physique Etendu du premier ordre (figure 2.4) montre que la capacité d'adressage du système proposé est liée directement à la taille du bus de données, du bus système ou du bus d'extension utilisé.

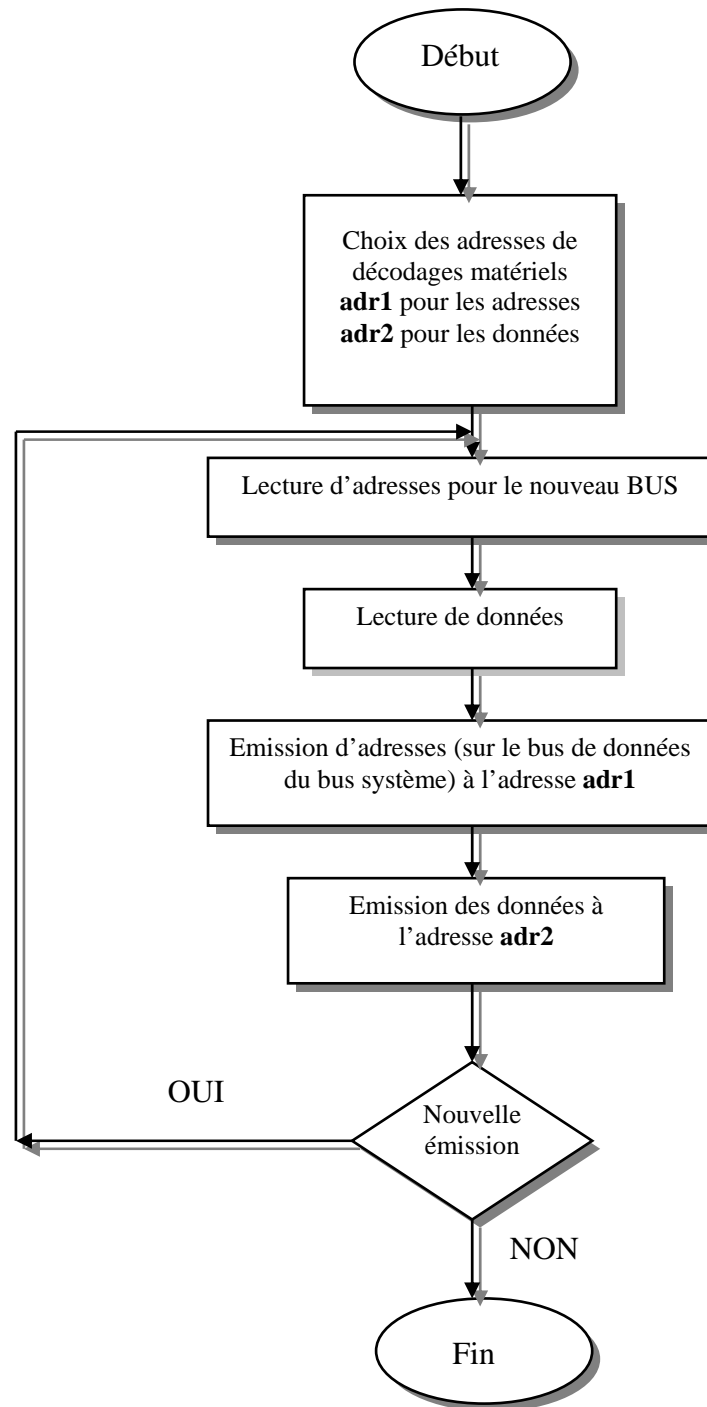


Fig. 2.3. Organigramme de la procédure logicielle de l'Adressage Physique Etendu du premier ordre.

Il est important de dire que les lignes de contrôle du premier bus sont utilisées avec les lignes d'adresses pour le décodage d'adresses dans la partie matérielle de notre système. Les lignes de contrôle du nouveau bus seront conçues suivant deux méthodes. Premièrement, les lignes de contrôle du bus système seront adaptées pour produire les nouvelles lignes de contrôle. Deuxièmement, les nouvelles lignes de contrôle seront conçues avec une logique combinatoire et suivant le type d'application de notre système.

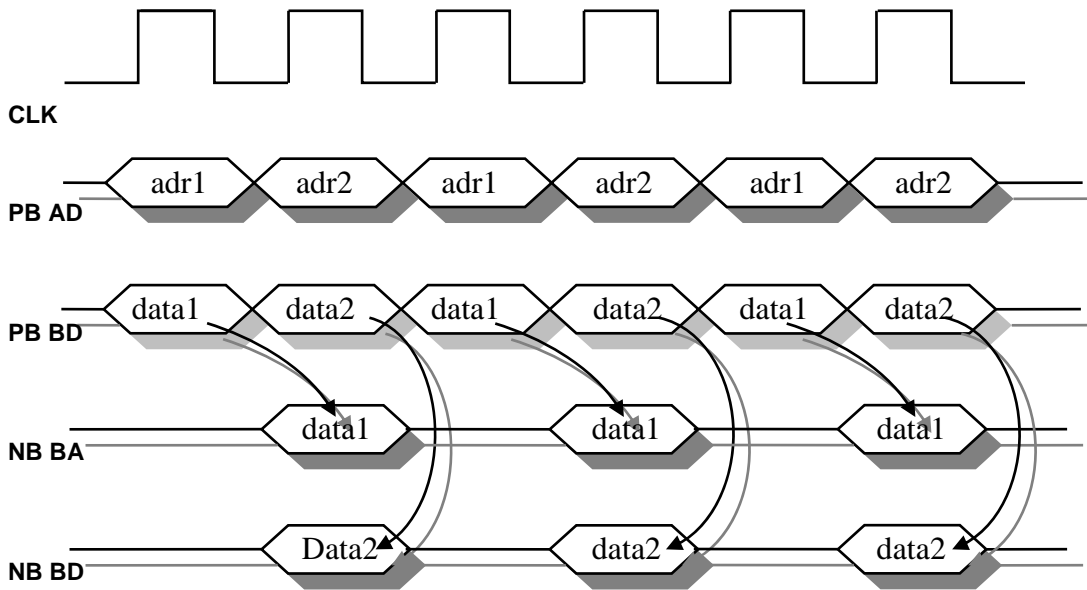


Fig. 2.4. Le chronogramme des transactions de l'Adressage Physique Etendu du premier ordre

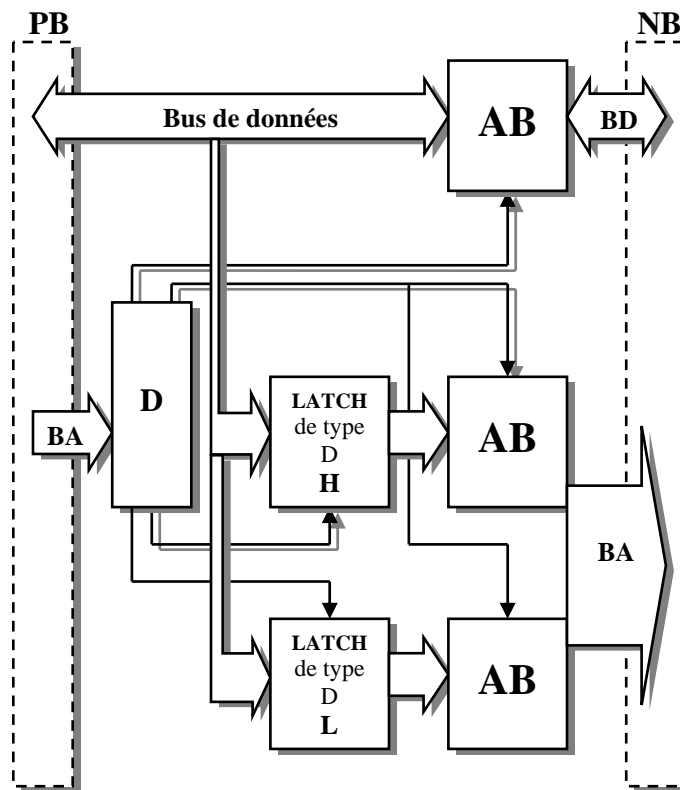
2.2. Adressage Physique Etendu d'ordre supérieur

L'Adressage Physique Etendu d'ordre supérieur propose une version plus complexe que celle du premier ordre. Nous avons vu que le principe du système proposé repose sur l'utilisation du facteur temps et sur l'utilisation d'une adresse physique supplémentaire pour avoir un nouveau bus. Ce dernier travaille à moitié fréquence du premier bus et avec une capacité d'adressage physique égale à 2^N où N représente la taille du bus de données du bus système et du nouveau bus. La version d'ordre supérieur de notre type d'adressage utilise un nombre d'adresses physiques du bus système supérieur à un, pour concevoir un nouveau bus qui travaille à des fréquences plus basses mais avec des capacités d'adressage plus grandes. Dans cette section nous exposons le système d'ordre deux. Ensuite, nous exposons le système d'ordre supérieur en général.

2.2.1. Système d'ordre deux

Le système d'ordre deux repose sur le même principe que celui du système du premier ordre pour réaliser la conversion donnée/adresse. Le fonctionnement du deuxième ordre passe par trois étapes. Les deux premières étapes sont similaires à la première étape du système du premier ordre et la troisième est la même que la deuxième du premier. La figure 2.5 représente le synoptique de la partie matérielle de l'Adressage Physique Etendu du deuxième ordre. L'entrée du système est connectée au bus système ou au bus d'extension et la sortie est connectée au périphérique externe. La capacité d'adressage du nouveau bus dans ce cas est largement plus grande que celle du premier ordre.

Il est à noter que l'augmentation de l'ordre de notre type d'adressage nécessite ; premièrement une utilisation supplémentaire des circuits "LATCH" de types D et des amplificateurs de bus associés. Deuxièmement une utilisation d'un décodeur d'adresses avec des sorties supplémentaires, ce qui implique l'addition d'autres fonctions logiques à ce décodeur d'adresses pour accomplir les nouvelles tâches.



AB : Amplificateur de Bus, BA : Bus d'Adresses, BD : Bus de Données, DA : Décodeur d'Adresses.
 H : Pour l'enregistrement du poids fort, L : Pour l'enregistrement du poids faible.
 NB : Nouveau Bus, PB : Premier Bus (Bus Système ou Bus d'Extension).

Fig. 2.5. Synoptique de l' Adressage Physique Etendu du deuxième ordre.

Le nombre d'adresses physiques occupées par le système d'ordre deux est égal à trois, une adresse pour la canalisation des données et deux adresses pour la conversion données/adresses. Dans la première et la deuxième étape, le logiciel assure la présence des données qui représentent le poids fort et le poids faible du bus d'adresses du nouveau bus. Cependant, le décodeur d'adresses active les deux circuits "LATCH" successivement pour enregistrer ces données jusqu' à la troisième étape et désactive les trois amplificateurs de bus. De la même manière que dans le premier ordre, l'activation des amplificateurs des bus se fait dans la dernière étape.

L'application de l' Adressage Physique Etendu d'ordre deux sur un bus système qui travaille avec un bus de données de N bits et une fréquence maximale F , donne un nouveau bus avec une fréquence égale à $F/3$, un bus de données de N bits et une capacité d'adressage physique de $2^{2.N}$. L'organigramme de la procédure logicielle de l' Adressage Physique Etendu d'ordre deux est représenté sur la figure 2.6.

A titre d'exemple, si le bus système fonctionne à une fréquence de 66 MHz et un bus de données de 16 bits, le nouveau bus fonctionnera avec une fréquence maximale de 22 MHz et avec un bus de données de 16 bits et une capacité d'adressage de 4 G.

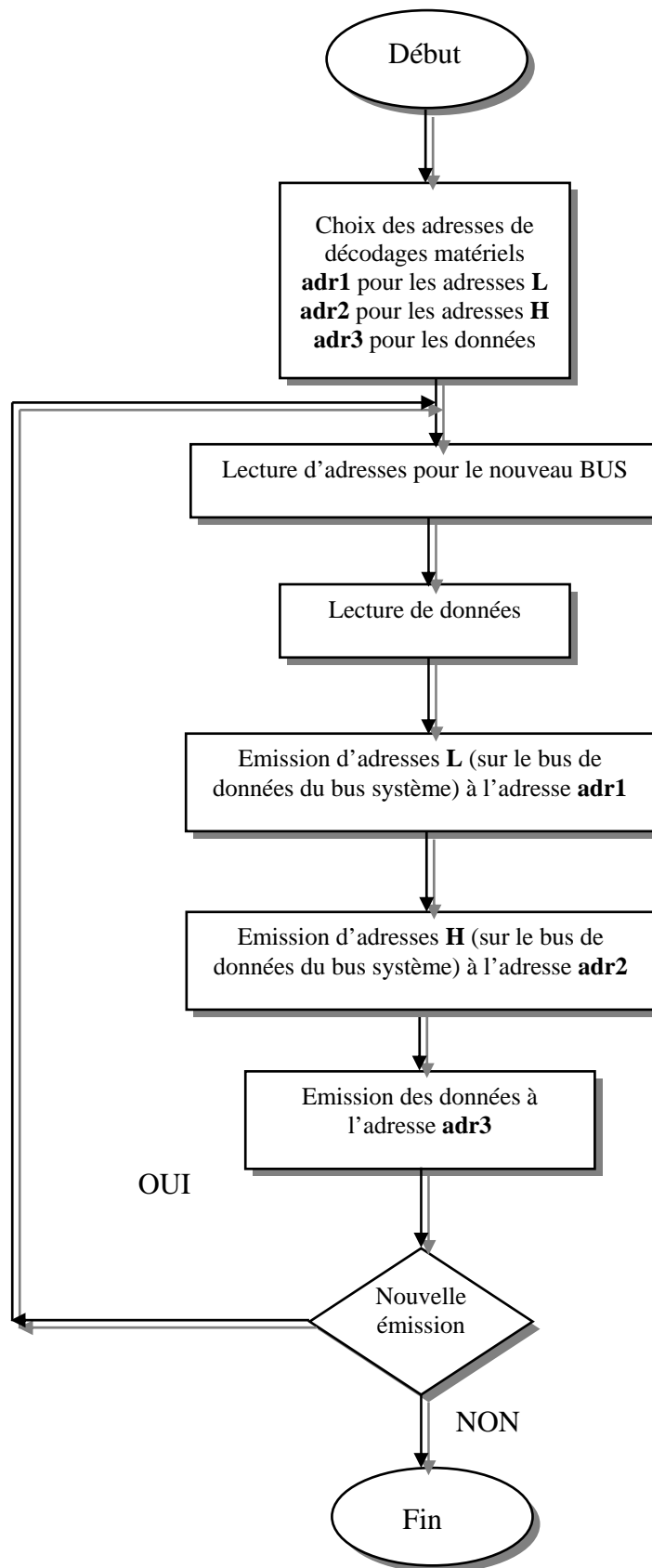


Fig. 2.6. Organigramme de la procédure logicielle de l'Adressage Physique Etendu d'ordre deux.

Le tableau 2.2 et le chronogramme de la figure 2.7 représentent les principales caractéristiques de l'Adressage Physique Etendu du deuxième ordre.

TABLEAU 2.2

Caractéristiques de l' Adressage Physique Etendu du deuxième ordre

<i>FI</i>	<i>NDS</i>	<i>NAS</i>	<i>NDN</i>	<i>NAN</i>	<i>FO</i>
<i>F</i>	8 bits	03	8 bits	64 K	<i>F</i> /3
<i>F</i>	16 bits	03	16 bits	4 G	<i>F</i> /3
<i>F</i>	32 bits	03	32 bits	2^{64}	<i>F</i> /3

FI : Fréquence de travail du bus système, *FO* : Fréquence de travail du nouveau bus.

NAN : Nombre d'adresses physiques disponibles sur le nouveau bus.

NAS : Nombre d'adresses physiques occupées sur le bus système.

NDS : Taille des données sur le bus système.

NDN : Taille des données sur le nouveau bus.

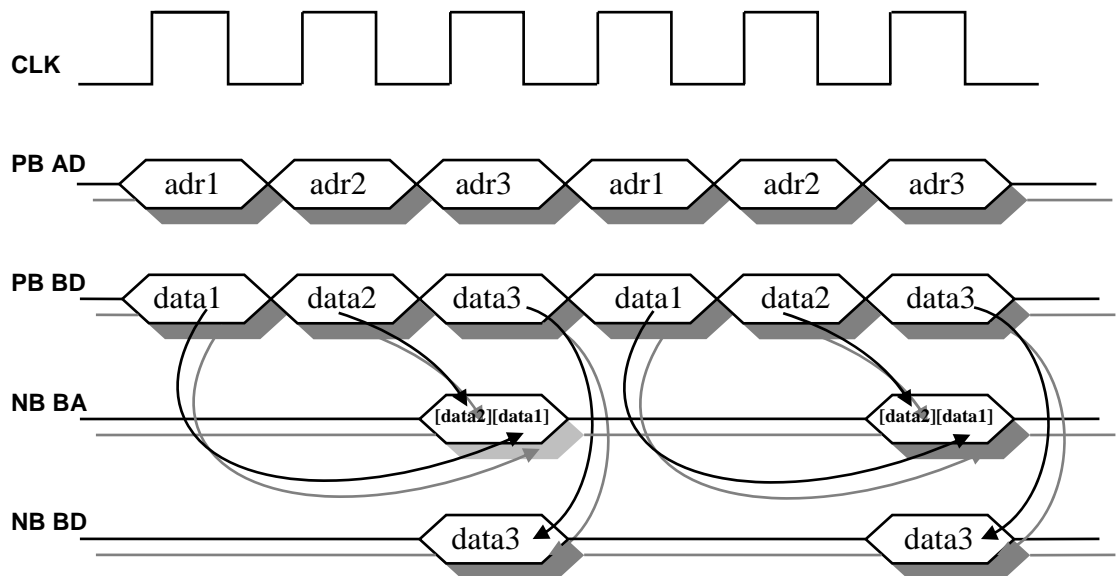


Fig. 2.7. Le chronogramme des transactions de l'Adressage Physique Etendu du deuxième ordre

Pour une première comparaison entre l'Adressage Physique Etendu du premier ordre et celui d'ordre deux, on constate que dans les deux cas, la capacité d'adressage du nouveau bus est liée à la taille du bus de données du bus système et aux nombres d'adresses utilisées pour la conversion donnée/adresse.

La fréquence du nouveau bus est liée seulement au nombre d'adresses physiques utilisées sur le bus système. La fréquence du nouveau bus est obtenue par la formule suivante :

$$FO = \frac{FI}{NAS} \quad (2.1)$$

Le nombre d'adresses physiques disponibles sur le nouveau bus dans le système d'ordre deux peut être obtenu comme suit :

$$NAN = 2^{2.NDS} \quad (2.2)$$

La solution proposée montre clairement que les nouveaux bus offrent un adressage physique plus souple, car la capacité adressage de ces bus est reliée à la largeur du bus de données du premier bus et non pas à la largeur du bus d'adresses. Le cas le plus général de notre solution est exposé dans la section suivante.

2.2.2. Système d'ordre n

A l'instar des deux types d'adressage d'ordre un et deux, le système d'ordre n suit la même logique de la conversion donnée/adresse, mais avec une structure plus complexe représentant le cas le plus général de notre technique. Nous appellerons ce cas; Adressage Physique Etendu d'ordre n , où les systèmes du premier ordre et du deuxième ordre représentent des cas particuliers du système d'ordre n . Le système d'ordre n utilise $(n+1)$ adresses physiques sur le bus système; une adresse pour la canalisation des données proprement dites et n adresses pour la conversion donnée/adresse. Il est à noter que l'augmentation de l'ordre du système, donne un nouveau bus avec une capacité d'adressage physique plus grande mais une fréquence de travail plus faible.

L'interface sera composée avec les mêmes types de circuits que les systèmes d'ordre inférieur ; un amplificateur de bus bidirectionnel pour le bus de données, n amplificateurs unidirectionnels pour le bus d'adresses et n circuits «LATCH » de type D pour l'enregistrement et la conversion des données. Le circuit décodeur possède une sortie pour la sélection de l'amplificateur de données, une sortie pour la sélection des amplificateurs d'adresses et n sorties pour l'activation des circuits «LATCH ».

Le fonctionnement dans ce cas général passe par $(n+1)$ étapes. De la première étape jusqu' à l' étape n , le logiciel assure la présence des données qui vont être converties en des adresses avec le même principe de conversion données/adresses exposé dans les sections de l'adressage du premier et du deuxième ordre. La dernière étape est consacrée à la canalisation des données proprement dites.

L'architecture du système d'ordre n donne un nouveau bus composé d'un bus de données, d'un bus d'adresses avec une fréquence FO donné par la formule suivante.

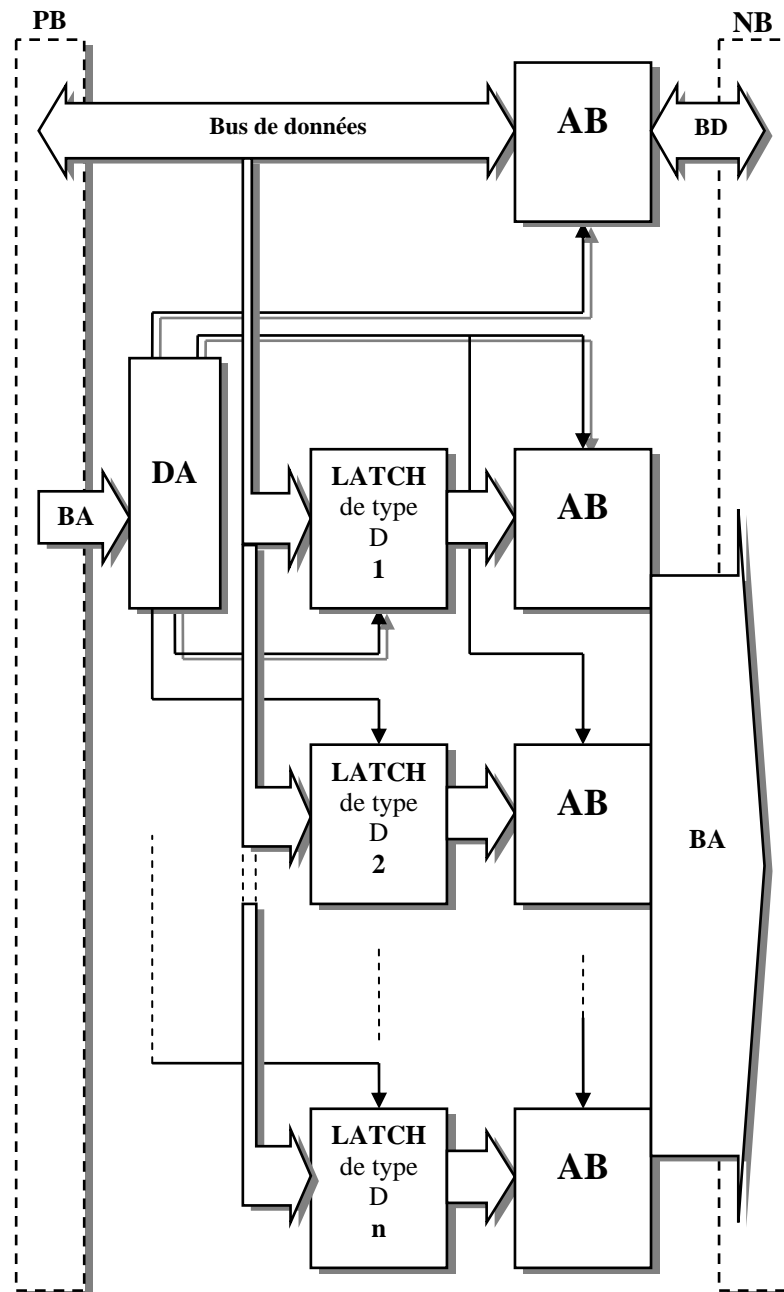
$$FO = \frac{FI}{(n+1)} \quad (2.3)$$

La capacité d'adressage NAN du nouveau bus du système d'ordre n , est donnée par la relation suivante :

$$NAN = 2^{n.NDS} \quad (2.4)$$

où n est l'ordre de l'Adressage Physique Etendu et NDS est la taille des données sur le bus système. La taille du bus de données du nouveau bus sera identique à la taille du bus de données du bus système ou du bus d'extension utilisé.

Le schéma synoptique de l'interface du système d'ordre n (figure 2.8) montre que la capacité d'adressage sur le nouveau bus est indépendante de la capacité d'adressage ou de la taille du bus d'adresses du premier bus. Le tableau 2.3 montre quelques caractéristiques de l'Adressage Physique Etendu d'ordre n pour des bus systèmes ou des bus d'extensions avec des bus de données de 8 bits, 16 bits et 32 bits.



AB : Amplificateur de Bus, BA : Bus d'Adresses, BD : Bus de Données, DA : Décodeur d'Adresses.
 1 : Pour l'enregistrement du poids le plus fort, n : Pour l'enregistrement du poids le plus faible.
 NB : Nouveau Bus, PB : Premier Bus (Bus Système ou Bus d'Extension).

Fig. 2.8. Synoptique de l' Adressage Physique Etendu d' Ordre n .

TABLEAU 2.3

Caractéristiques de l' Adressage Physique Etendu d'ordre n

<i>FI</i>	<i>NDS</i>	<i>NAS</i>	<i>NDN</i>	<i>NAN</i>	<i>FO</i>
<i>F</i>	8 bits	$n+1$	8 bits	2^{8n}	$F/n+1$
<i>F</i>	16 bits	$n+1$	16 bits	2^{16n}	$F/n+1$
<i>F</i>	32 bits	$n+1$	32 bits	2^{32n}	$F/n+1$

NDN : Taille des données sur le nouveau bus. NDS : Taille des données sur le bus système.
 FI : Fréquence de travail du bus système, FO : Fréquence de travail du nouveau bus.
 NAN : Nombre d'adresses physiques disponibles sur le nouveau bus.
 NAS : Nombre d'adresses physiques occupées sur le bus système.
 n : Ordre de l' Adressage Physique Etendu.

2.3. Compatibilité de l'Adressage Physique Etendu

Le système d'Adressage Physique Etendu est conçu pour être compatible avec des périphériques travaillant avec un système de communication parallèle fonctionnant avec un bus de données et un bus de contrôle. Nous avons vu que la canalisation des données proprement dites dans notre système provoque l'activation des amplificateurs unidirectionnels d'adresses et de l'amplificateur bidirectionnel des données. Ceci signifie que la canalisation des données proprement dites est possible sans l'émission des données destinées à être converties en des adresses.

On constate que la mise des lignes d'adresses du nouveau bus à l'état haute impédance n'influe pas sur la canalisation des données sur le nouveau bus. Les lignes de contrôle du nouveau bus peuvent être conçues avec plusieurs méthodes, suivant le type d'application et le type de périphérique en question (figure 2.9).

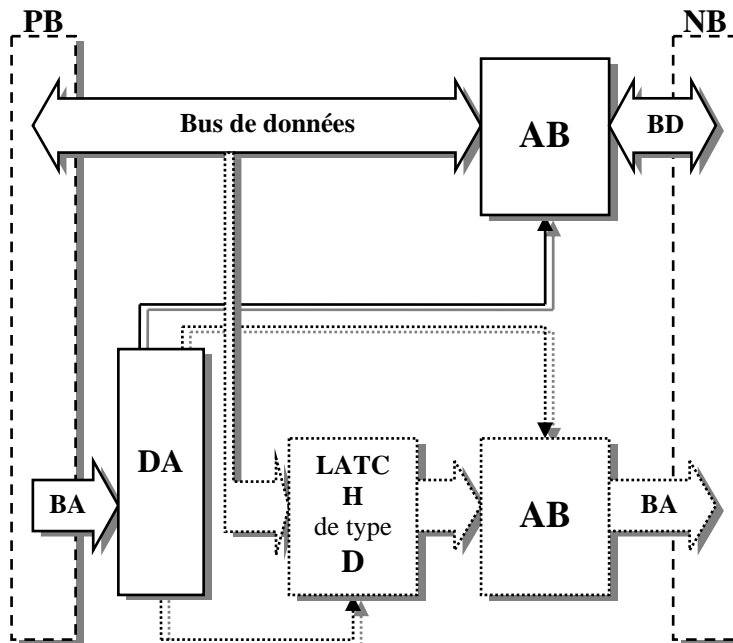


Fig. 2.9. Compatibilité de l'Adressage Physique Etendu

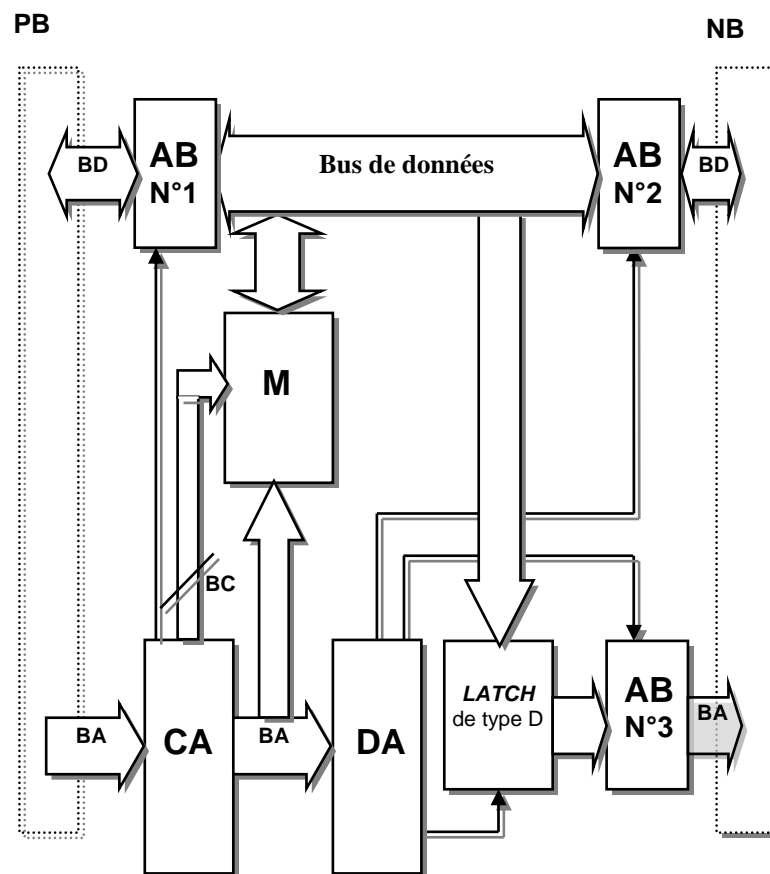
3. Adressage Physique Accéléré

3.1. Principe de l'Adressage Physique Accéléré

Avec le même esprit de la variante précédente et avec une structure modifiée, l'Adressage Physique Accéléré [22] est basé sur une architecture mixte logicielle/matérielle et vise deux objectifs : l'élargissement de la capacité d'adressage matériel des ordinateurs et des systèmes à microprocesseur, et l'augmentation de la fréquence maximale de travail du nouveau bus qui représente l'amélioration apportée par cette variante.

La figure 2.10 représente le synoptique de la partie matérielle de la variante de l'Adressage Physique Accéléré. L'Adressage Physique Accéléré utilise la même technique, de conversion données/adresses, développée dans la section précédente, pour la production des adresses du nouveau bus et un processus plus complexe pour le stockage des deux types de données dans la partie matérielle de ce système.

Le processus de stockage est utilisé afin d'atteindre des fréquences élevées durant l'échange de données avec les périphériques externes. L'entrée de l'interface sera connectée au bus du système à microprocesseur et la sortie sera connectée à un périphérique externe.



AB : Amplificateur de Bus, BA : Bus d'Adresses, BC: Bus de Contrôle, BD : Bus de Données.
 CA : Contrôleur d'adresses, DA : Décodeur d'Adresses, M : Mémoire, NB : Nouveau Bus.
 PB : Premier Bus (Bus Système ou Bus D'Extension).

Fig. 2.10. Synoptique de l'Adressage Physique Accéléré.

Afin d'assurer l'accélération du transfert de données, cette version d'interfaçage de notre système réalise un empilement de données dans sa mémoire [23,24]. L'empilement est activé dans les trois cas suivants : premièrement, le transfert de données du périphérique externe vers le système à microprocesseur est assuré sans l'utilisation des interruptions matérielles ; deuxièmement, le transfert de données du système à microprocesseur vers un périphérique externe s'effectue sans l'interruption de ce dernier ; troisièmement, les périphériques externes reliés à notre nouveau bus, réalisent un échange de données sans le passage par le système à microprocesseur. De même que dans le mécanisme de l'Adressage Physique Étendu, le bus de données du bus système assure la canalisation des données proprement dites, et la canalisation des données destinées à être converties en des adresses.

La partie logicielle de l'Adressage Physique Accéléré assure deux tâches supplémentaires : la commande d'activation du processus d'enregistrement sur la mémoire de notre système ; et la commande d'activation du processus de la lecture de la mémoire. La validation des deux commandes sera assurée par le contrôleur d'adresses **CA**, qui génère les commandes et les adresses de notre mémoire. L'enregistrement dans la mémoire de notre système est commandé par le contrôleur d'adresses qui pilote le bus d'adresses et de contrôle de notre mémoire. Le contrôleur d'adresses active et désactive les amplificateurs de bus selon le cas d'échange de données. Dans le cas de transfert du système à microprocesseur vers notre système, le contrôleur d'adresses active l'amplificateur n°1 et désactive l'amplificateur n°2. Dans le cas contraire l'amplificateur n°2 est activé, et la distinction entre les deux types de données est assurée par le décodeur d'adresses **DA**. Le processus de la canalisation des données proprement dites et le processus de la conversion données/adresses sont similaires à ceux de l'Adressage Physique Étendu. L'intégration d'une mémoire dans le système réclame au minimum une utilisation de deux adresses supplémentaires pour les deux commandes d'empilement et de dépilement des données.

La figure 2.11 représente le chronogramme des transactions de l'Adressage Physique Accéléré. CA_CLK est le signal d'horloge du contrôleur d'adresses. Le bus d'adresses M_adr de notre mémoire est piloté par le contrôleur d'adresses CA afin de transférer les données du bus mémoire M_data vers les deux nouveaux bus d'adresses et de données (NB_AD et NB_BD).

L'application de cette version sur un bus système qui travaille avec un bus de données de N bits et une fréquence maximale F , donne un nouveau bus de N bits de données, d'une capacité d'adressage physique de 2^N et qui travaille à une fréquence maximale égale à Fn . Le tableau 2.4 représente les principales caractéristiques de l'Adressage Physique Accéléré.

La fréquence maximale du nouveau bus est obtenue par la formule suivante :

$$Fn = \frac{1}{2 \times Ta} \quad (2.5)$$

Ta : Temps d'accès de la mémoire utilisée par notre système.

Il est clair que la fréquence maximale du nouveau bus est indépendante de la fréquence de travail du bus système, mais elle dépend du temps d'accès de la mémoire utilisée.

Pour un cas numérique, si l'Adressage Physique est appliqué sur un bus système qui fonctionne à une fréquence de 66 MHz, un bus de données de 32 bits et un temps d'accès de

la mémoire utilisée égale à 6ns, le nouveau bus fonctionnera avec une capacité d'adressage physique de 4 G et une fréquence de travail maximale de 83 MHz. La variante accélérée montre clairement que le nouveau bus résultant offre un adressage physique plus souple et une fréquence maximale qui peuvent dépasser la fréquence du bus système, si le temps d'accès de la mémoire utilisée dans le processus d'accélération le permet.

TABLEAU 2.4

Caractéristiques de l'Adressage Physique Accéléré

<i>FI</i>	<i>NDS</i>	<i>NAS</i>	<i>NDN</i>	<i>NAN</i>	<i>FO</i>
<i>F</i>	8 bits	02	8 bits	256	<i>F_n</i>
<i>F</i>	16 bits	02	16 bits	64 K	<i>F_n</i>
<i>F</i>	32 bits	02	32 bits	4 G	<i>F_n</i>

FI : Fréquence de travail du bus système, FO : Fréquence de travail du nouveau bus
 NAN : Nombre d'adresses physiques disponibles sur le nouveau bus.
 NAS : Nombre d'adresses physiques occupées sur le bus système.
 NDN : Taille des données sur le nouveau bus
 NDS : Taille des données sur le bus système.

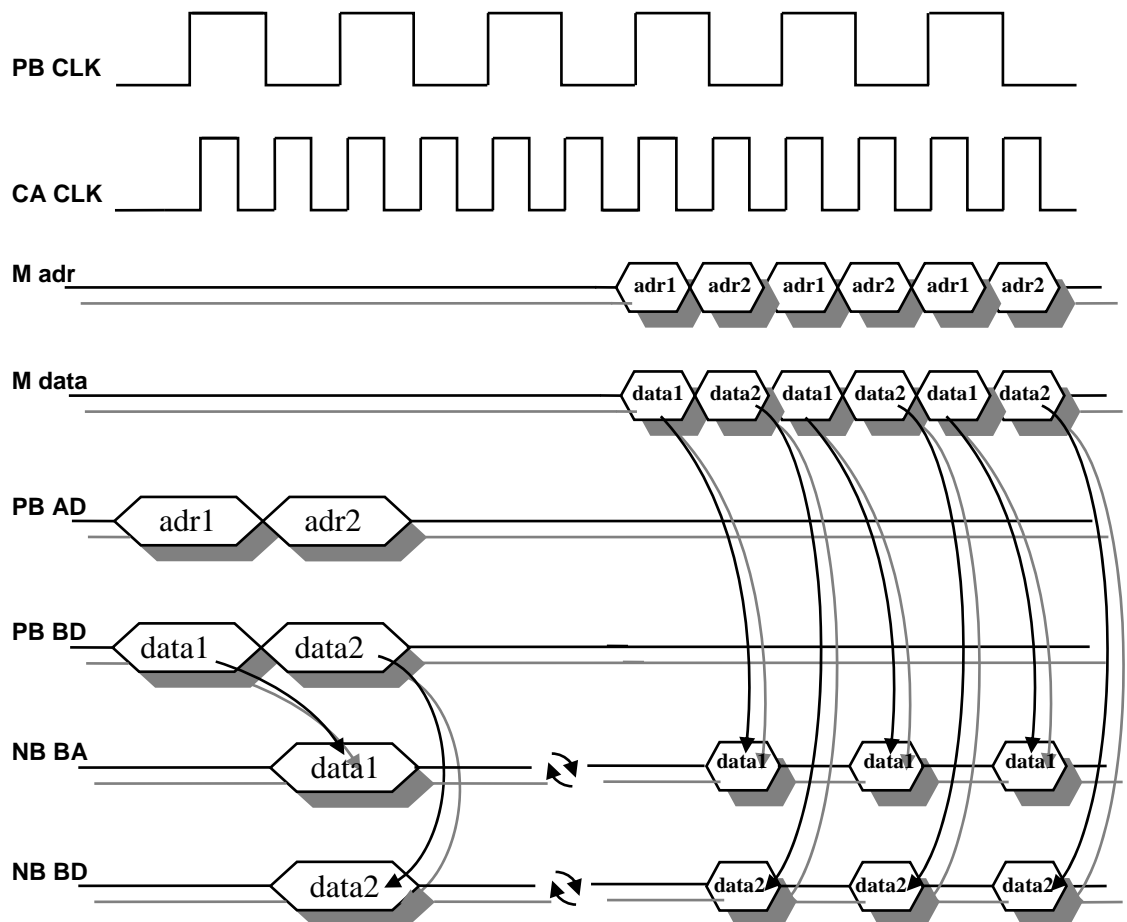
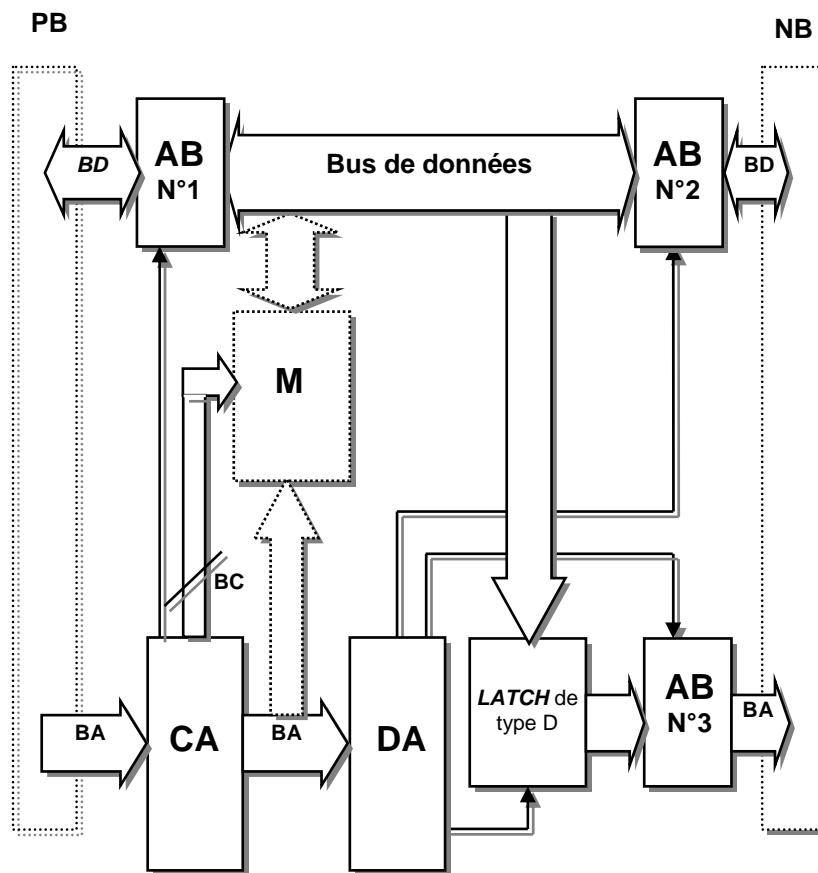


Fig. 2.11. Le chronogramme des transactions de l'Adressage Physique Accéléré

3.2. Compatibilité de l'Adressage Physique Accéléré

L'architecture proposée présente, une compatibilité avec l'Adressage Physique Etendu et en conséquence, une compatibilité avec les périphériques travaillant avec un système de communication parallèle qui fonctionne avec un bus de données et un bus de contrôle [25].

Pour avoir un fonctionnement similaire à l'Adressage Physique Etendu, le contrôleur d'adresses réalise trois tâches. Premièrement, il met la mémoire de notre système à l'état haute impédance à l'aide de son bus de contrôle. Deuxièmement, il active l'amplificateur de bus n°1 d'une manière permanente. Troisièmement, il relie le bus d'adresses du bus système au décodeur d'adresses **DA**, qui sera activé comme un décodeur d'adresses du bus système (figure 2.12).



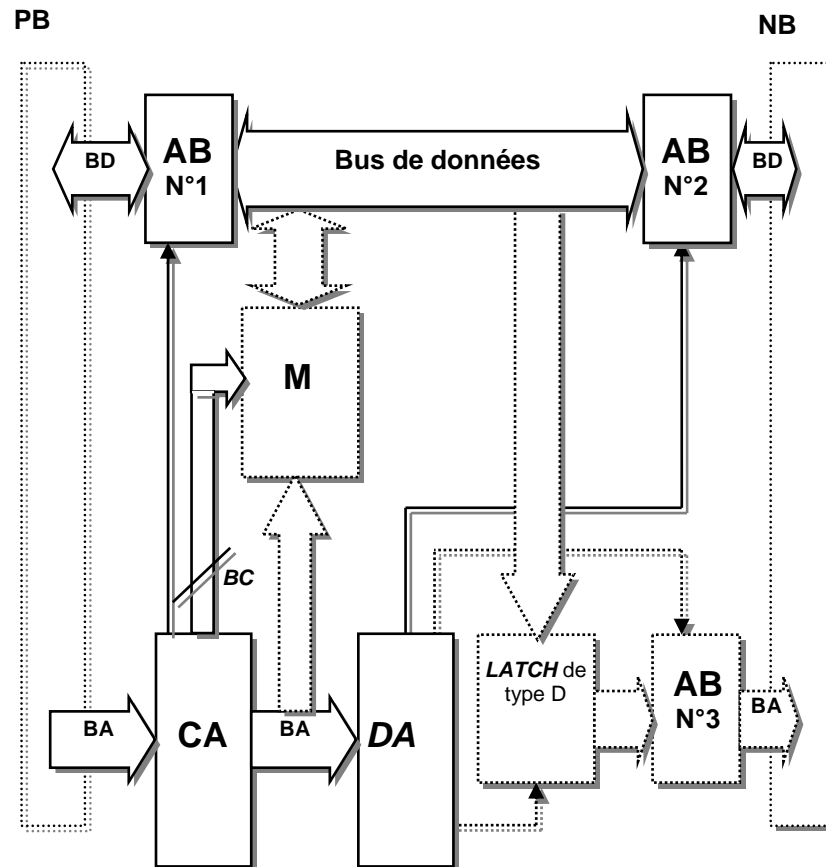
NB : Nouveau Bus, PB : Premier Bus (Bus Système ou Bus d'Extension).
 DA : Décodeur d'Adresses, CA : Contrôleur d'adresses
 AB : Amplificateur de Bus, BA : Bus d'Adresses.
 BC : Bus de Contrôle, BD : Bus de Données.

Fig. 2.12. Compatibilité avec l'Adressage Physique Etendu.

Nous avons vu dans le principe de l'Adressage Physique Etendu que la canalisation des données proprement dites dans notre système provoque l'activation simultanée des amplificateurs unidirectionnels d'adresses et de l'amplificateur bidirectionnel des données. Ce qui implique que la mise des lignes d'adresses du nouveau bus à l'état haute impédance n'influe pas sur la canalisation des données sur le nouveau bus. De ce fait, le deuxième type

de compatibilité est assuré par les trois conditions du premier type et par la mise des lignes d'adresses du nouveau bus à l'état haute impédance (figure 2.13).

Les lignes de contrôle du nouveau bus peuvent être conçues avec plusieurs méthodes, suivant le type d'application et le type de périphérique en question.



CA : Contrôleur d'adresses, DA : Décodeur d'Adresses. NB : Nouveau Bus. BD : Bus de Données.
 AB : Amplificateur de Bus. BA : Bus d'Adresses. BC: Bus de Contrôle.
 PB : Premier Bus (Bus Système ou Bus d'Extension).

Fig. 2.13. Compatibilité avec les systèmes parallèles.

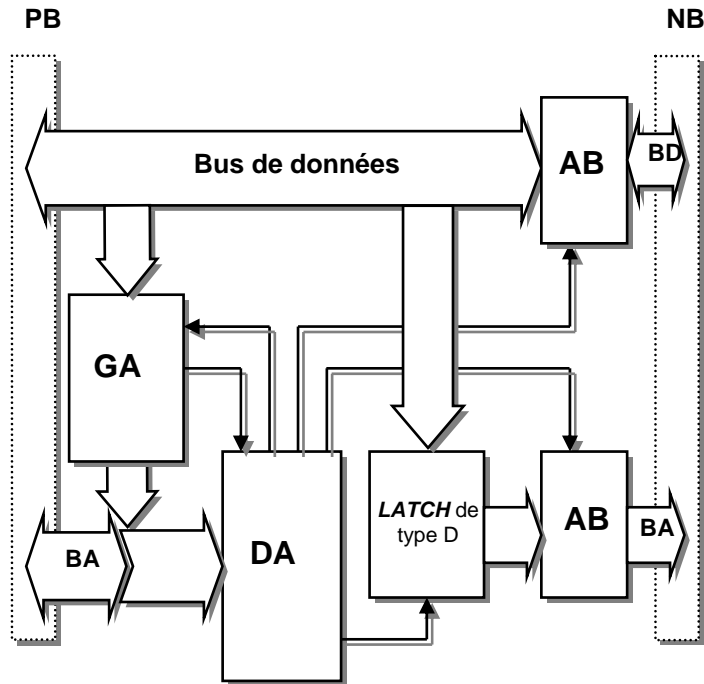
4. Adressage Physique Rapide

4.1. Principe de l'Adressage Physique Rapide

Avec la même logique des variantes précédentes et avec une structure modifiée, l'Adressage Physique Rapide [26] est basé sur une architecture mixte logicielle/matérielle et vise deux objectifs. Le premier objectif est l'élargissement de la capacité d'adressage matériel des ordinateurs et des systèmes à microprocesseur. Le deuxième objectif, qui représente l'amélioration apportée par cette variante, est l'augmentation de la fréquence maximale de travail du nouveau bus.

L'Adressage Physique Rapide utilise la même technique, de conversion données/adresses, illustrée dans la section précédente, pour la production des adresses du nouveau bus et un processus différent pour le transfert de données du système à microprocesseur vers l'entrée de la partie matérielle de notre système. L'entrée de l'interface sera connectée au bus du système à microprocesseur et la sortie sera connectée à un périphérique externe.

La figure 2.14 représente le synoptique de la partie matérielle de la variante de l'Adressage Physique Etendu avec accès direct à la mémoire.



AB : Amplificateur de Bus, BA : Bus d'Adresses, BD : Bus de Données, DA : Décodeur d'Adresses.
GA : Générateur d'Adresses, NB : Nouveau Bus, PB : Premier Bus (Bus Système ou Bus d'Extension).

Fig. 2.14. Synoptique de l'Adressage Physique Rapide

Dans cette version, la partie matérielle réalise des accès directs à la mémoire [27,28] pour l'échange de données, et par conséquent, les adresses disponibles sur le nouveau bus dépendent d'une partie des données échangées avec la mémoire du système à microprocesseur. De même que dans le mécanisme de l'Adressage Physique Etendu, le bus de données du bus système assure la canalisation des données proprement dites, et la canalisation des données destinées à être converties en des adresses.

La partie logicielle de l'Adressage Physique Rapide assure deux tâches supplémentaires. La première est l'envoi des deux positions, début et fin, de la zone mémoire autorisée à être accessible par notre système. La deuxième est l'arbitrage du début et de l'arrêt de l'accès direct à la mémoire. Les deux positions, limitant la zone mémoire adressée par notre système, seront envoyées vers le générateur d'adresses GA, sur le bus de données du bus système et sur des adresses distinctes.

Le processus DMA de notre système débute par l'envoi de ces deux données. Le décodeur d'adresses actionne l'enregistrement de ces deux données dans le circuit générateur d'adresses DMA. Une fois le processus d'accès direct à la mémoire est enclenché, le

générateur d'adresses pilote le bus d'adresses du bus système pour réaliser la lecture ou l'écriture sur la zone mémoire choisie et active le décodeur d'adresses de notre système pour réaliser le décodage relatif aux adresses générées. La distinction entre les deux types de données est assurée par le décodeur d'adresses. Le processus de la canalisation des données proprement dites et le processus de la conversion données/adresses sont similaires à ceux de l'Adressage Physique Etendu. L'utilisation de la DMA réclame au minimum une utilisation de deux adresses supplémentaires pour l'envoi du début et de la fin de la zone mémoire adressée.

Le chronogramme de la figure 2.15 présente les phases de la génération d'adresses. GA_CLK est le signal d'horloge du générateur d'adresses. Les signaux DB et DE activent l'enregistrement des données limitant la zone DMA. Le signal DMA amorce le processus d'accès direct à la mémoire.

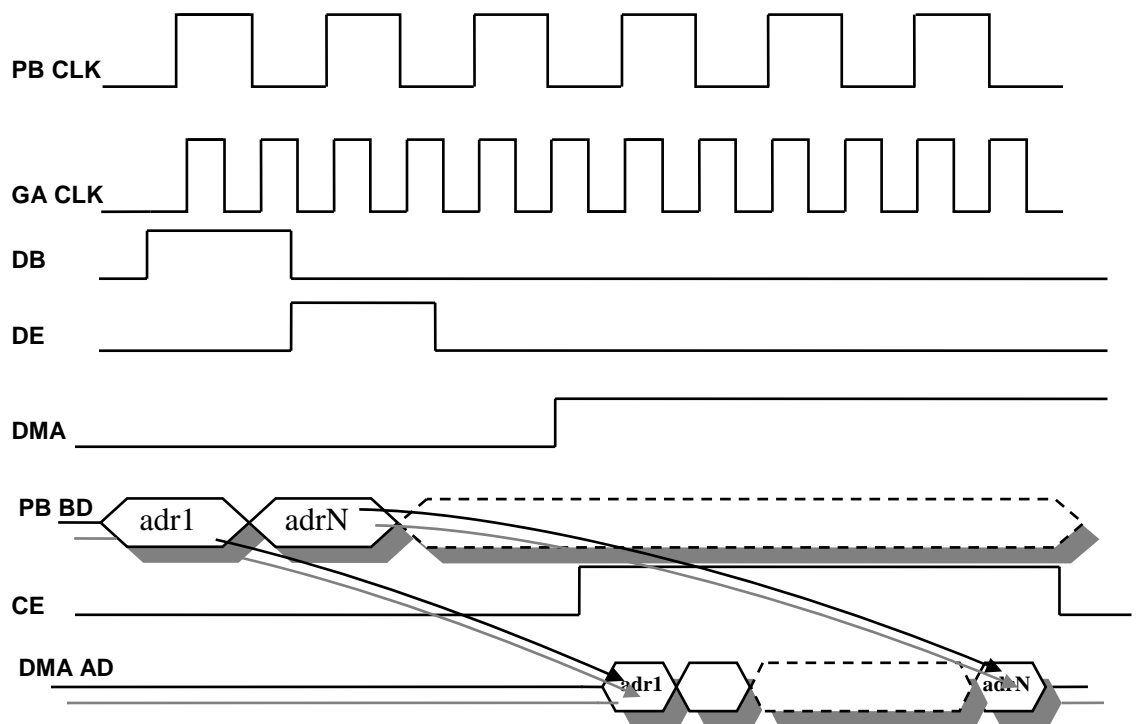


Fig. 2.15. Les phases de la génération d'adresses

L'application de cette version sur un bus système qui travaille avec un bus de données de N bits et une fréquence maximale F , donne un nouveau bus de N bits de données, d'une capacité d'adressage physique de 2^N et qui travaille à une fréquence maximale égale à F_n . Le tableau 2.5 représente les principales caractéristiques de l'Adressage Physique Rapide.

La fréquence maximale du nouveau bus s'obtient par la formule suivante.

$$F_n = \frac{1}{2 \times T_a} \quad (2.6)$$

T_a : Temps d'accès de la mémoire utilisée par le système à microprocesseur durant le processus de l'accès direct à la mémoire.

La fréquence maximale du nouveau bus est indépendante de la fréquence de travail du bus système, mais elle dépend du temps d'accès de la mémoire utilisée dans le processus DMA. La figure 2.16 représente le chronogramme des transactions de l'Adressage Physique Rapide. GA_CLK est le signal d'horloge du générateur d'adresses. Le bus d'adresses du système à microprocesseur est piloté par le générateur d'adresses avec GA_adr afin de récupérer et de véhiculer les données DMA_data vers les deux nouveaux bus d'adresses et de données (NB_AD et NB_BD).

TABLEAU 2.5

Caractéristiques de l'Adressage Physique Rapide

<i>FI</i>	<i>NDS</i>	<i>NAS</i>	<i>NDN</i>	<i>NAN</i>	<i>FO</i>
<i>F</i>	8 bits	04	8 bits	256	<i>Fn</i>
<i>F</i>	16 bits	04	16 bits	64 K	<i>Fn</i>
<i>F</i>	32 bits	04	32 bits	4 G	<i>Fn</i>

FI : Fréquence de travail du bus système, FO : Fréquence de travail du nouveau bus.
 NAN : Nombre d'adresses physiques disponibles sur le nouveau bus.
 NAS : Nombre d'adresses physiques occupées sur le bus système.
 NDN : Taille des données sur le nouveau bus.
 NDS : Taille des données sur le bus système.

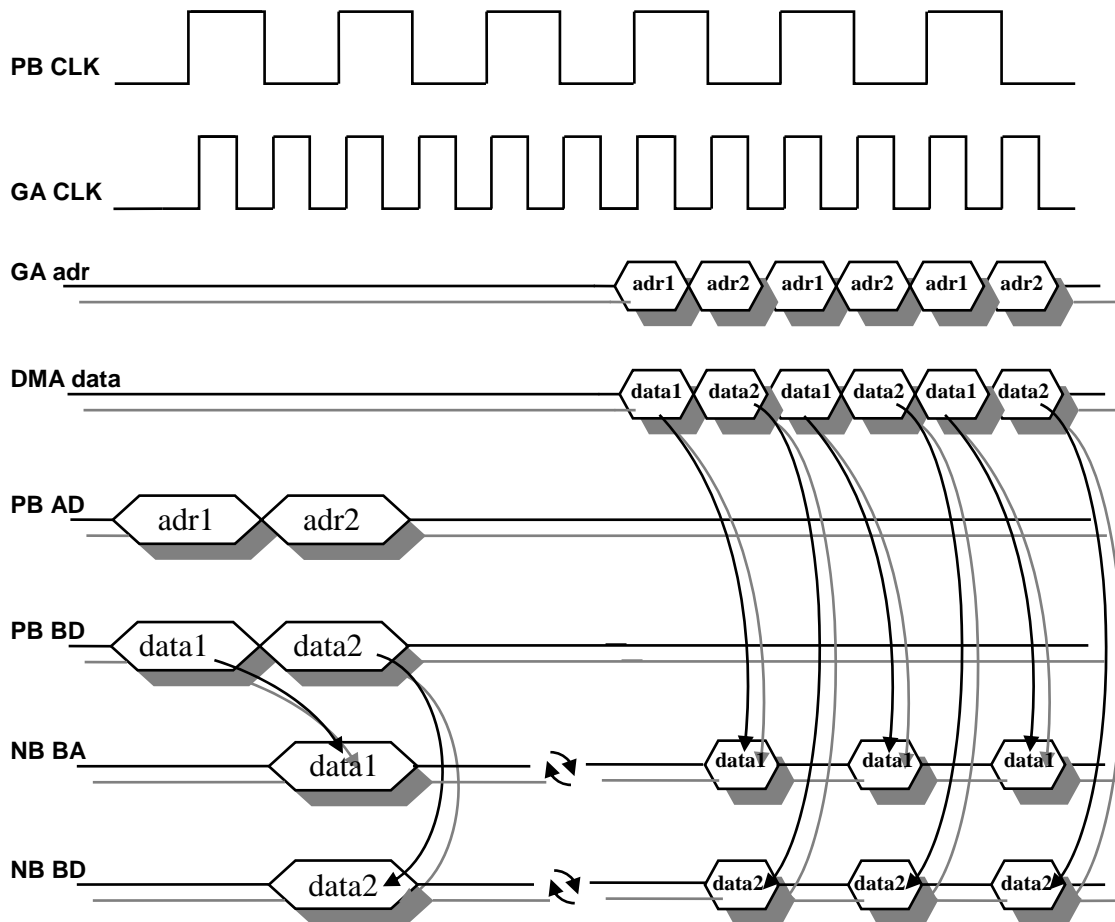


Fig. 2.16. Le chronogramme des transactions de l'Adressage Physique Rapide.

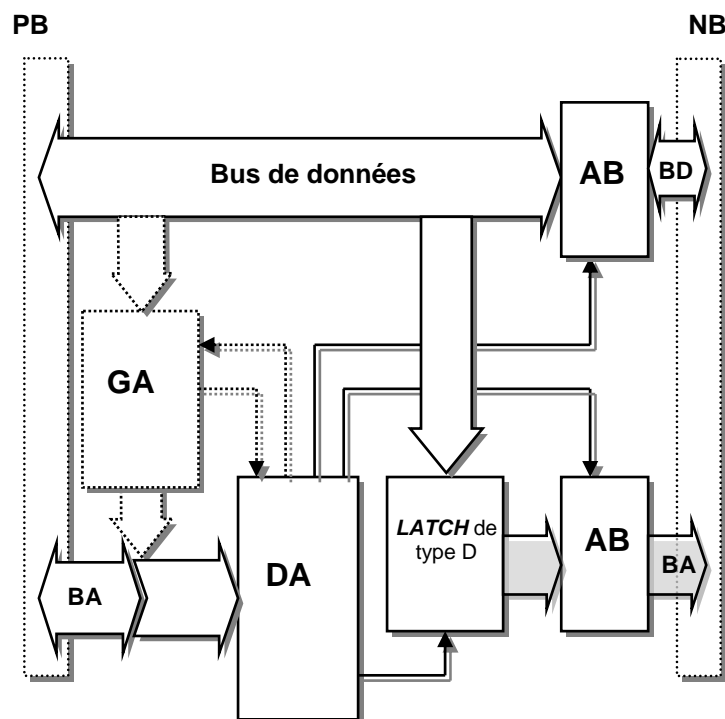
La variante avec DMA montre clairement que le nouveau bus résultant offre un adressage physique plus souple et une fréquence maximale qui peut dépasser la fréquence du bus système, si le temps d'accès de la mémoire utilisée dans le processus DMA le permet.

A titre d'exemple, si l'Adressage Physique Rapide est appliqué sur un bus système qui fonctionne à une fréquence de 66 MHz, un bus de données de 32 bits et un temps d'accès de la mémoire utilisée dans la DMA égale à 6ns, le nouveau bus fonctionnera avec une capacité d'adressage physique de 4 G et une fréquence maximale de 83 MHz.

4.2. Compatibilité de l'Adressage Physique Rapide

L'Adressage Physique Rapide offre une compatibilité avec l'Adressage Physique Etendu et par conséquent, une compatibilité avec les périphériques travaillant avec un système de communication parallèle qui fonctionne avec un bus de données et un bus de contrôle.

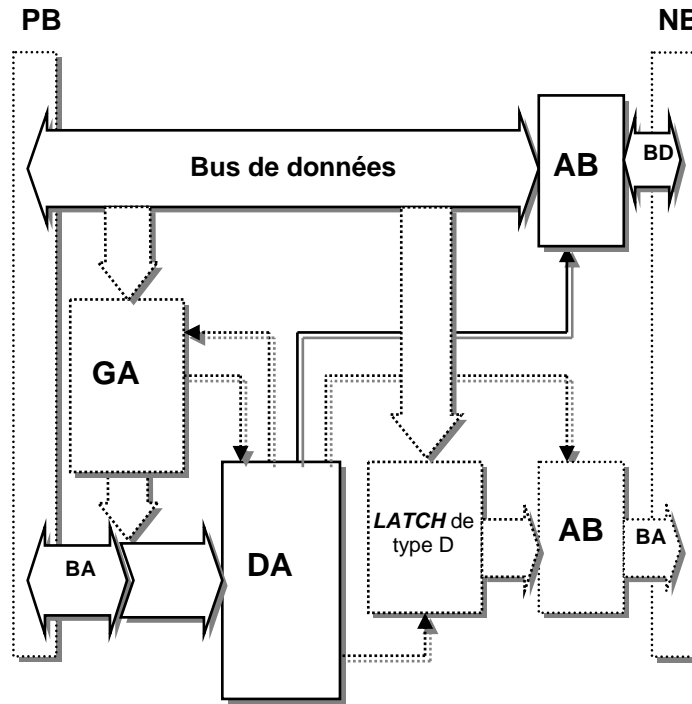
La compatibilité avec l'Adressage Physique Etendu est assurée par la non-sélection du générateur d'adresses et par l'élimination des deux premières tâches réalisées par la partie logicielle. Le processus de la canalisation des données proprement dites et le processus de la conversion données/adresses seront identiques à ceux de l'Adressage Physique Etendu (figure 2.17).



AB : Amplificateur de Bus, BA : Bus d'Adresses, BD : Bus de Données, DA : Décodeur d'Adresses.
GA : Générateur d'Adresses, NB : Nouveau Bus, PB : Premier Bus (Bus Système ou Bus d'Extension).

Fig. 2.17. Compatibilité avec l'Adressage Physique Etendu.

La deuxième compatibilité est assurée par la non-sélection du générateur d'adresses et par la mise des lignes d'adresses du nouveau bus à l'état haute impédance (figure 2.18). Les lignes de contrôle du nouveau bus peuvent être conçues avec plusieurs méthodes, suivant le type d'application et le type de périphérique en question.



AB : Amplificateur de Bus, BA : Bus d'Adresses, BD : Bus de Données.
 DA : Décodeur d'Adresses. GA : Générateur d'Adresses, NB : Nouveau Bus.
 PB : Premier Bus (Bus Système ou Bus d'Extension).

Fig. 2.18. Compatibilité avec la communication parallèle

5. Adressage Physique Avancé

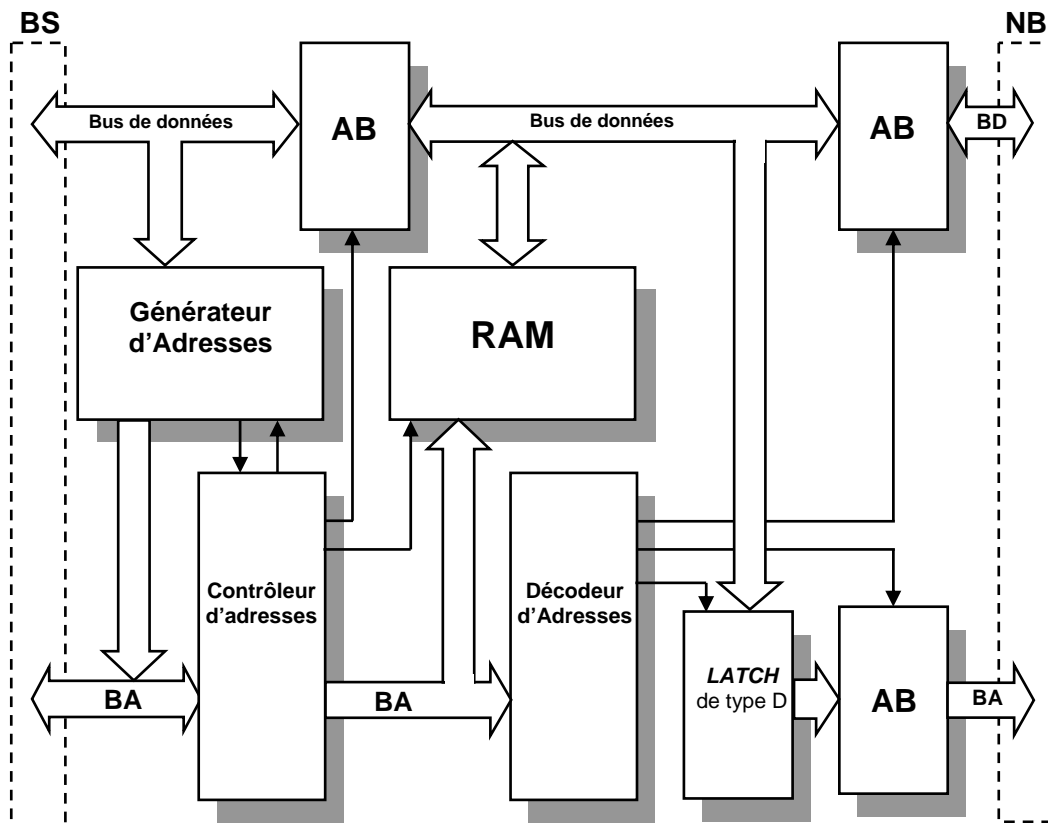
Cette variante est une combinaison de l'Adressage Physique Rapide et de l'Adressage Physique Accéléré [29]. La figure 2.19 représente le synoptique de la partie matérielle de la variante de l'Adressage Physique Avancé.

La partie logicielle de l'Adressage Physique Avancé assure quatre tâches supplémentaires par rapport à l'Adressage Physique Etendu. La première est l'envoi des deux positions, début et fin, de la zone mémoire autorisée à être accessible par notre système. La deuxième est l'arbitrage du début et de l'arrêt de l'accès direct à la mémoire. Les deux tâches fonctionnent avec la méthode de l'Adressage Physique Rapide mais le générateur d'adresses est contrôlé par le contrôleur d'adresses AC. La troisième tâche est l'activation du processus d'écriture dans la mémoire de la partie matérielle notre système [30,31]. La quatrième tâche est l'activation du processus de lecture. Le processus de la lecture et d'écriture est contrôlé par le contrôleur d'adresses AC avec la même méthode de l'Adressage Physique Accéléré.

La fréquence maximale du nouveau bus est obtenue par la formule suivante.

$$Fn = \frac{1}{2 \times Ta} \quad (2.7)$$

Ta : Temps d'accès de la mémoire utilisée par le mécanisme DMA, ou le temps d'accès de la mémoire utilisée par notre système.



BA : Bus d'Adresses, BD : Bus de Données, AB : Amplificateur de Bus. NB : Nouveau Bus. BS : Premier Bus (Bus Système ou Bus d'Extension).

Fig. 2.19. Synoptique de l'Adressage Physique Avancé

TABLEAU 2.6

Caractéristiques de l'Adressage Physique Avancé

<i>FI</i>	<i>NDS</i>	<i>NDN</i>	<i>NAN</i>	<i>FO</i>
<i>F</i>	8 bits	8 bits	256	<i>Fn</i>
<i>F</i>	16 bits	16 bits	64 K	<i>Fn</i>
<i>F</i>	32 bits	32 bits	4 G	<i>Fn</i>

NAN : Nombre d'adresses physiques disponibles sur le nouveau bus. NDN : Taille des données sur le nouveau bus. NDS : Taille des données sur le bus système, FI : Fréquence de travail du bus système. FO : Fréquence de travail du nouveau bus.

L'application de cette version sur un bus système qui travaille avec un bus de données de N bits et une fréquence maximale F , donne un nouveau bus de N bits de données, d'une capacité d'adressage physique de 2^N et qui travaille à une fréquence maximale égale à Fn . Le tableau 2.6 représente les principales caractéristiques de l'Adressage Physique Avancé.

Par exemple, si l'Adressage Physique Avancé est appliqué sur un bus système avec une fréquence de 66 MHz, un bus de données de 32 bits et un temps d'accès de la mémoire utilisée égale à 5ns, le nouveau bus fonctionnera avec une capacité d'adressage physique de 4 G et une fréquence de travail maximale de 100 MHz.

6. CONCLUSION

L'interfaçage et l'adressage physique des périphériques posent dans plusieurs cas des conflits d'adresses ou des limitations d'adressage physique. Dans ce travail, nous avons développé des solutions permettant de perfectionner l'adressage physique et qui consistent à élargir et accélérer la capacité d'adressage des systèmes à microprocesseur ou des ordinateurs. Ces solutions que nous appelons Adressage Physique Etendu, Accéléré, Rapide et Avancé sont basées sur l'utilisation des systèmes mixte logiciel/matériel qui font appel à des principes simples à réaliser.

Le système d'Adressage Physique Etendu d'ordre un et qui représente la base de nos systèmes est exposé en premier. Par la suite nous avons présenté les solutions des différentes variantes. Les nouveaux systèmes que nous exposons présentent des remèdes à l'inconvénient de la basse fréquence des débits pratiques de transferts liés à l'adressage physique classique, et présentent une large capacité d'adressage physique non liée à la taille du bus d'adresses du bus système.

Ces systèmes d'interfaçages ont été appliqué pour des implémentations sur **CPLD** [32] et **FPGA** de **Xilinx** [33,34] avec des cartes qui fonctionnent sur des ordinateurs **IBM**TM et compatibles. Le principe des implémentations est exposé dans le chapitre trois et les applications d'interfaçage graphique est exposé dans le chapitre quatre.

Chapitre 3

**Logique programmable et
implémentation de
l'interfaçage**

1. Introduction

La fabrication d'un circuit intégré spécifique à une application (ASIC) est une tâche complexe et coûteuse, en temps et en argent [35]. En effet, la réalisation d'un prototype nécessite la création d'un ou plusieurs masques, qui sont très coûteux s'ils ne sont destinés qu'à une petite quantité. De plus, une simple erreur dans la conception du système implique la création d'un ou plusieurs nouveaux masques. Et finalement, l'investissement en temps de développement est important, et n'est pas toujours tolérable si le produit doit très rapidement être mis sur le marché.

Parallèlement à la progression des ASICs, divers circuits programmables ont fait leurs apparitions, afin de réduire le temps et le coût de développement des circuits électroniques, tout en restant relativement compétitifs sur le plan de la rapidité d'exécution. Le problème est alors de pouvoir concevoir des systèmes contenant des circuits intégrés de la manière la moins coûteuse en termes de temps et d'argent. En fonction de différents paramètres tels que la complexité du système, le nombre de pièces à fabriquer ou le temps accordé, un type d'implémentation est choisi parmi la gamme des possibilités.

Dans ce chapitre, nous allons présenter les principaux types de circuits intégrés programmables et reprogrammables, allant de la simple mémoire programmable au circuit FPGA (Field Programmable Gate Array). Nous allons tout d'abord présenter brièvement le processeur puis les différents circuits logiques programmables. Ensuite nous présenterons l'implémentation et le test de notre technique d'interfaçage avec la logique programmable sur CPLD. L'implémentation est effectuée avec des solutions schématiques et VHDL.

2. Processeur

Un microprocesseur est un circuit intégré permettant d'implémenter n'importe quelle fonction en exécutant de manière séquentielle un code compilé. Le grand avantage du microprocesseur ou d'un processeur est sa généricité, puisque n'importe quel programme peut y être exécuté. De plus, il est très facile de le programmer, c'est-à-dire de transcrire dans le langage du microprocesseur un problème particulier.

L'architecture d'un système à microprocesseur est divisée en deux catégories, architecture matérielle et architecture logicielle. L'architecture matérielle de base est représentée sur la figure 3.1. Le microprocesseur est chargé de traiter toute information passant par l'ordinateur, il constitue l'unité centrale de traitement. Il gère les mémoires centrales, **RAM**, **ROM** et **EPROM**, les « entrées-sorties » ainsi que des circuits d'interface qui assurent la communication avec le monde extérieur. Il faut noter que le contenu des **EPROMs** appelé le **BIOS** est le premier lien entre les deux architectures matérielle et logicielle.

IL est clair que la communication des différentes parties du système avec le microprocesseur, qui est le coeur du système, passe à travers le bus système. Ce dernier est composé d'un bus de données, d'un bus d'adresses et d'un bus de contrôle.

Après le démarrage, l'ordinateur a directement accès au **BIOS**, un espace mémoire lui y est réservé. Le BIOS est logé sur une ou plusieurs puces (**ROM**, **EPROM** et **EEPROM**). A la mise sous tension, le PC commence une série de tests qui porte sur les composants

importants de l'ordinateur. Lorsque ces tests sont terminés, tous les composants matériels sont initialisés et le système d'exploitation est chargé à partir d'un support. Après l'initialisation, le contenu de l'enregistrement de démarrage est chargé en mémoire. Alors, nous entrons dans l'architecture logicielle.

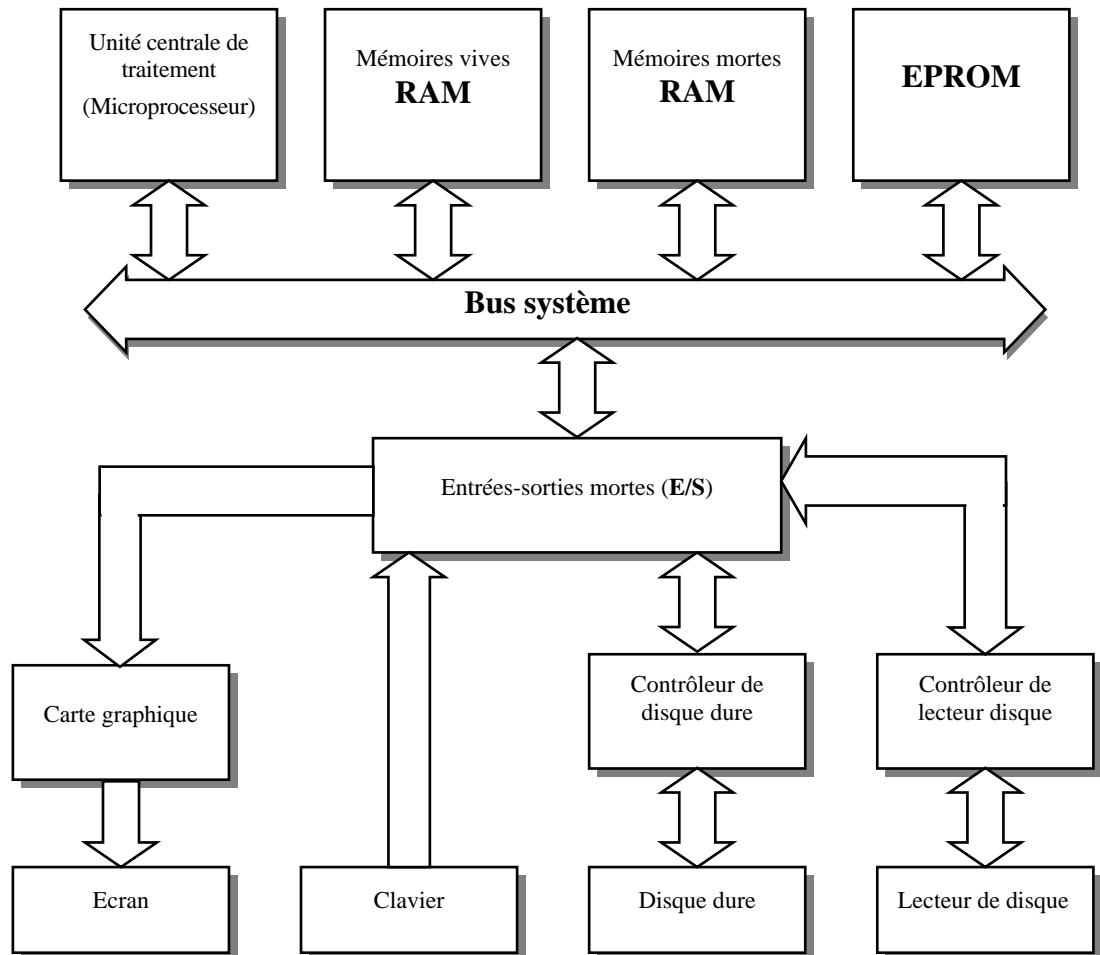


Fig. 3.1. Architecture de base d'un système à microprocesseur.

En contrepartie, la flexibilité d'un système à microprocesseur s'accompagne d'un coût, à savoir le temps d'exécution d'une tâche. En effet, les instructions sont exécutées une à une, bien que certains processeurs présentent un certain parallélisme. L'opération d'addition successive de cinq nombres nécessite au minimum quatre micro-instructions. Un système matériel peut, lui, effectuer cette opération en un seul pas. De plus, le processeur étant l'unité chargée du calcul, une seule erreur compromet l'ensemble de son exécution. Les systèmes parallèles laissent quant à eux la porte ouverte à l'autoréparation, une unité pouvant potentiellement prendre la place d'une autre se trouvant dans un état défectueux.

Les applications intrinsèquement parallèles telles que les réseaux de neurones ou les automates cellulaires sont donc exécutées de manière séquentielle par un processeur, alors qu'un système matériel a la possibilité d'exploiter tout le parallélisme possible. Les systèmes cellulaires composés de plusieurs cellules effectuant des tâches en parallèle sont donc bien plus efficacement réalisés en matériel qu'en logiciel.

3. Technologies de programmation

Notons premièrement qu'un circuit programmable doit pouvoir être programmé, de même qu'un circuit reprogrammable doit pouvoir être reprogrammé. Nous allons présenter dans ce qui suit les différentes technologies servant à la réalisation de circuits programmables ou reprogrammables. Nous commencerons par la technologie des fusibles et les antifusibles, non reprogrammables, puis nous enchaînerons sur les technologies des circuits reprogrammables, à savoir l'EPROM, l'EEPROM, la FLASH et la SRAM.

3.2 Fusible

Avec le même principe des fusibles en électricité, qui se détruisent par un trop fort courant, la technologie basée sur des fusibles fut une des premières technologies à être utilisée. Le principe est simplement d'avoir des fusibles sur certains fils, et d'en faire brûler certains en leur appliquant un courant trop important. Sur chacun de ces fils, une résistance en pull-up ou pull-down force la valeur à 1 ou à 0 si le fusible a été détruit, comme illustré à la figure 3.2.

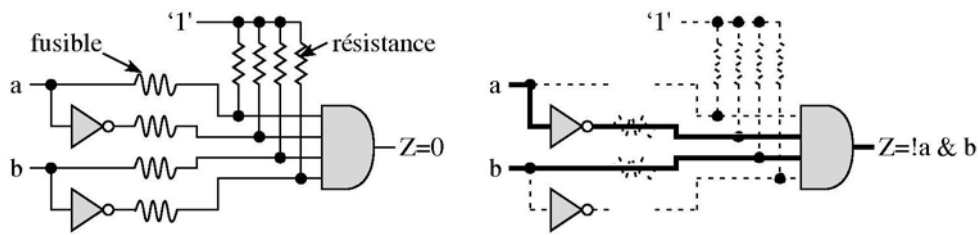


Fig. 3.2. Un circuit contenant 4 fusibles non programmés, puis le circuit résultant après avoir brûlé le premier et le quatrième fusible.

3.3 Antifusible

A ce que l'on peut considérer comme l'opposé des fusibles, se trouvent les antifusibles. De la même manière, sur certains fils du circuit se trouvent des antifusibles, mais contrairement aux fusibles, lorsqu'ils ne sont pas programmés, ils agissent comme une résistance infinie, comme si le fil était coupé. En appliquant un fort courant, l'antifusible est programmé, et laisse dès lors passer le courant. Une résistance est également placée après chaque antifusible, de manière à forcer la ligne à 1 ou 0 dans le cas où il n'est pas programmé (figure 3.3).

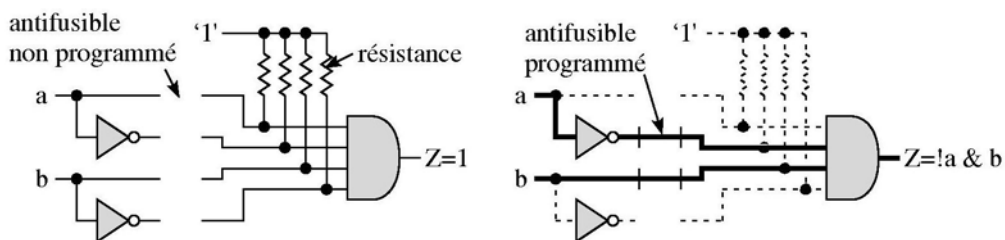


Fig. 3.3. Un circuit contenant 4 antifusibles non programmés, puis le circuit résultant d'une programmation.

Notons que les circuits à base de fusibles et d'antifusibles ne peuvent être programmés qu'une seule fois. Chaque nouvelle implémentation, en cas d'erreur de design par exemple, implique dès lors la programmation d'un nouveau composant et l'élimination de l'ancien. Cependant, cette programmation étant électrique, un simple appareil est nécessaire, rendant le processus nettement plus simple et rapide que le masquage.

3.4 EPROM

EPROM (UV Erasable Programmable Read Only Memory) est une mémoire PROM (Programmable ROM) spéciale, programmable électriquement et effaçable par la lumière UV. Le fonctionnement d'une mémoire PROM est identique que celui d'une ROM (figure 3.4) avec un mode programmation spécifique [36]. Au lieu de devoir physiquement créer des connexions grâce à une couche de métal, un fusible peut être brûlé ou non, ce qui est nettement plus rapide et moins coûteux. En effet, lors de la réalisation d'un système informatique, il n'est pas rare que des erreurs se glissent dans un design, et la destruction d'une PROM à chaque erreur n'est pas le remède.

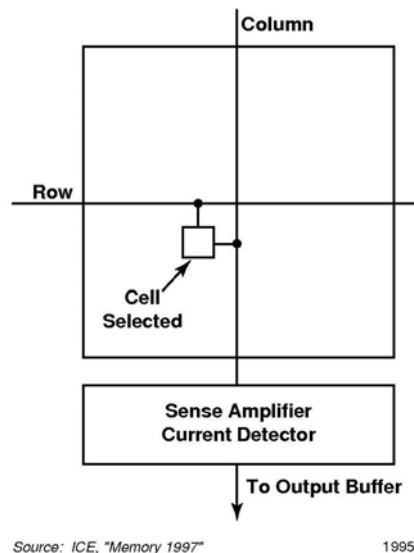


Fig. 3.4. Architecture d'une ROM.

La technologie de type Erasable Programmable Read-Only Memory (EPROM), introduite en 1971 par Intel est dès lors une nette amélioration de la PROM sur le plan de la flexibilité, puisqu'elle permet une reprogrammation du circuit. Il s'agit d'un nouveau type de transistor, basé sur un transistor MOS, auquel une couche de silicium polycristallin, appelé porte flottante, a été ajoutée, isolée par des couches d'oxyde (figure 3.5). Dans son état initial, le transistor agit normalement, tel un MOS standard. En appliquant un courant de haut voltage (typiquement 12V) entre la grille et le drain, un effet tunnel charge la porte flottante en électrons, ce qui a pour effet de bloquer le transistor en état ouvert. A la suite de la programmation, quelle que soit la tension appliquée au contrôle, aucun courant ne peut passer entre la source et le drain. La charge créée par la programmation demeure, même lorsque le circuit est hors tension, et ce n'est qu'avec une exposition du circuit à des rayons UV que la programmation est effacée. Lié à l'effacement, un des désavantages de l'EPROM est qu'une fenêtre en quartz doit être apposée sur le circuit, afin de pouvoir laisser passer les rayons UV, ce qui augmente grandement le prix du boîtier.

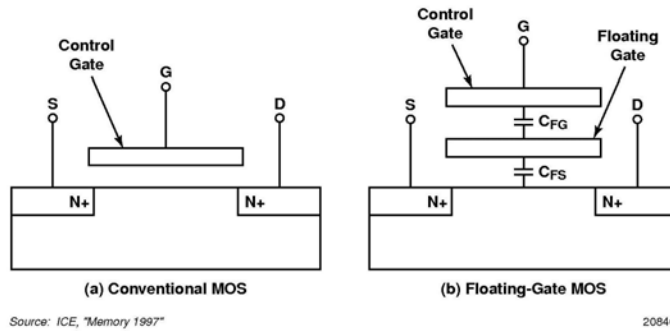


Fig. 3.5. Un transistor CMOS standard et un transistor EPROM.

3.5 EEPROM/Flash

Le principal inconvénient des EPROMs est la déprogrammation par rayons UV, ceci est oublié dans les EEPROMs, qui sont effaçables électriquement. Il n'est plus nécessaire d'intervenir physiquement (avec la lumière UV), un système électronique suffit pour reprogrammer l'EEPROM de manière autonome [37]. Sur le plan de l'espace pris sur le silicium, une cellule mémoire d'EEPROM occupe toutefois environ 2,5 fois plus de place qu'une cellule d'EPROM, car elle est composée de deux transistors, ainsi que de l'espace nécessaire entre les deux.

Les mémoires FLASH sont basées sur le même principe d'effacement électronique et de non-volatilité. De nombreuses expériences et implémentations en ont fait des composants très largement utilisés à l'heure actuelle. Vues de l'extérieur, les FLASH sont quasiment identiques aux EEPROMs, si ce n'est que l'effacement se fait par secteur, et non octet par octet. Elles sont toutefois souvent préférées, pour leur vitesse et leur taille, qui peut être plus imposante que celle des EEPROMs. La figure 3.6 présente le synoptique d'une EEPROM "Xicor".

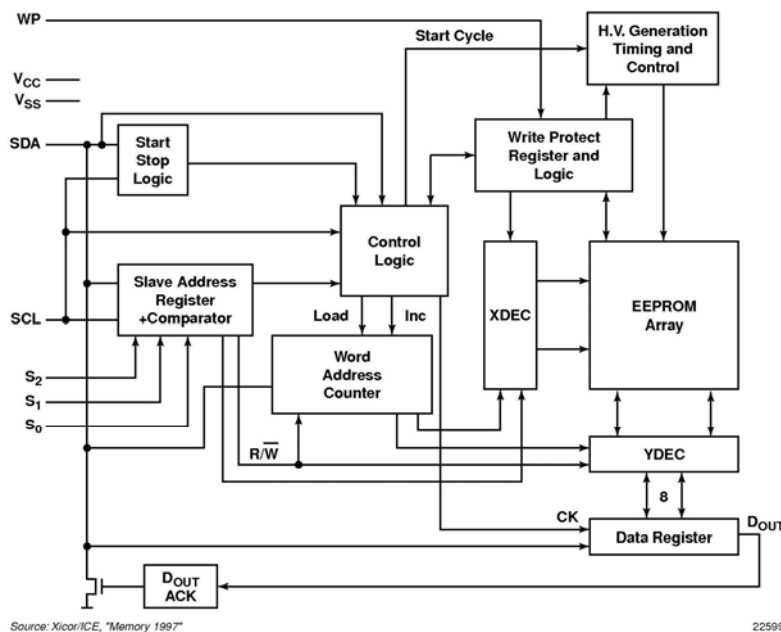


Fig. 3.6. Synoptique d'une EEPROM "Xicor" série de 128Kb.

3.6 SRAM

La technologie SRAM, pour Static Random Access Memory, est, contrairement aux technologies déjà présentées, volatile, et doit donc être reprogrammée à chaque remise en marche du système [38]. Leur second désavantage réside dans la place nécessaire à une cellule, qui est composée de quatre à six transistors formant un latch (figure 3.7).

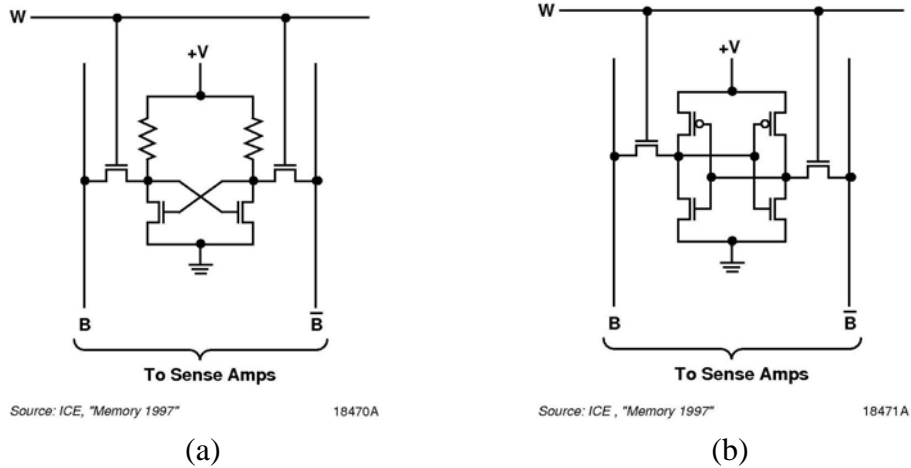


Fig. 3.7. (a) Une cellule SRAM à quatre transistors. (b) Une cellule SRAM à six transistors.

Toutefois, la grande facilité de programmation des circuits en technologie SRAM en a fait le choix privilégié des deux plus grands fabricants de FPGAs, à savoir Xilinx et Altera. Les figures 3.8a et 3.8b illustrent le synoptique et le brochage d'une mémoire SRAM typique.

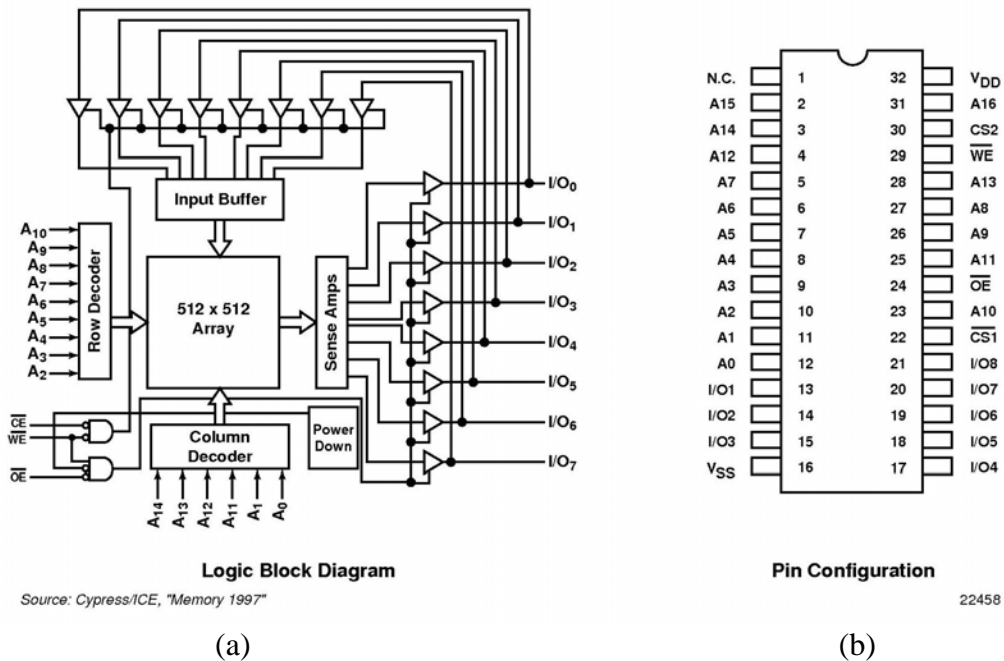


Fig. 3.8. (a) Synoptique d'une SRAM typique. (b) Brochage d'une SRAM typique.

Les FPGAs commerciaux basés sur cette technologie sont le plus souvent implémentés avec des pass-transistors ou des portes de transmission. Seul le fameux XC6200 est réalisé avec des multiplexeurs. Cette dernière technique offre l'avantage d'interdire tout court-circuit, contrairement aux deux autres. Toutefois, cet avantage se paie par une baisse de performances en terme de rapidité, un signal met plus de temps à passer un multiplexeur qu'un simple transistor.

4. Circuits logiques programmables

Les circuits programmables sont apparus en 1970, et depuis ils ont évolué d'une manière étonnante et plusieurs familles se sont distinguées. Il existe dans la littérature plusieurs manières de les classer. Nous choisirons l'approche illustrée à la figure 3.9 consistant à séparer les PLDs (Programmable Logic Device) en trois sous-classes, à savoir les SPLDs (Simple Programmable Logic Device), les CPLDs (Complex Programmable Logic Device) et les FPGAs (Field Programmable Gate Array).

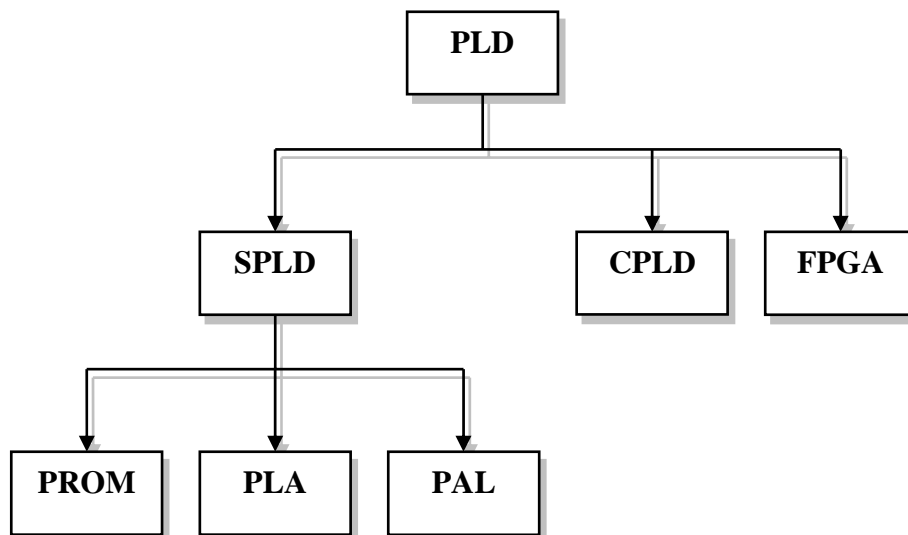


Fig. 3.9. Une classification des circuits logiques programmables.

Nous allons décrire les différents types de circuits programmables, dans l'ordre chronologique de leur apparition, correspondant également à leur complexité, en commençant par les SPLD, les CPLDs, puis les FPGAs.

4.1. SPLD

Les SPLDs (Simple Programmable Logic Device) [39], dans une description haut niveau, sont composés d'une grille de portes ET et d'une grille de portes OU, les deux étant reliées. Les entrées du système peuvent être connectées aux portes ET, et le résultat des portes OU correspond à la sortie du système (figure 3.10). Dans ces circuits, les connexions sont préexistantes, les différentes lignes étant reliées par des fusibles, des transistors EPROM, ou des transistors EEPROM. En brûlant certains de ces fusibles, ou en programmant les transistors, il est alors possible de réaliser différentes fonctions logiques.

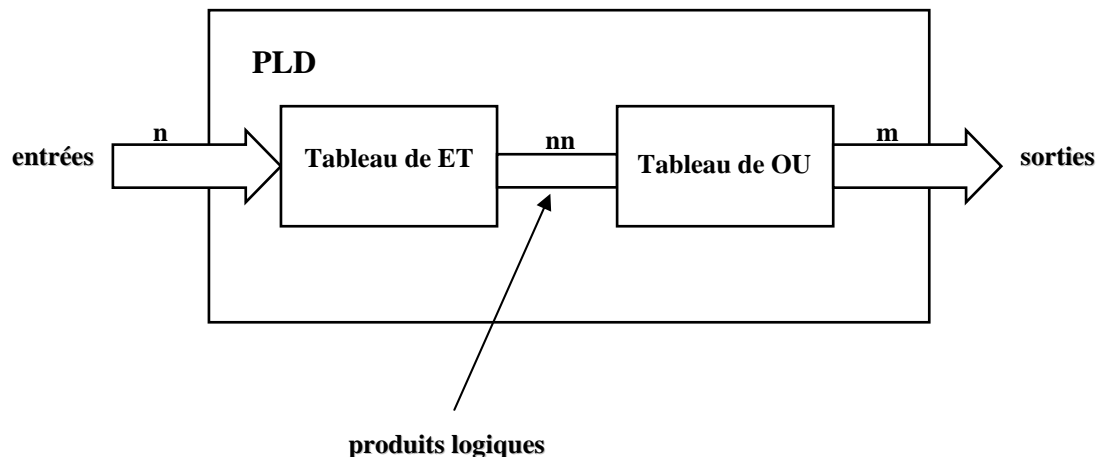


Fig. 3.10. L'architecture d'un SPLD.

PROM

Les premiers circuits programmables à avoir vu le jour sont les PROMs (Programmable Read-Only Memory) (figure 3.11), le terme PROM étant introduit en 1970. Ces circuits sont des mémoires accessibles en lecture, qui contrairement aux ROMs, sont programmables. En effet, une ROM est livrée déjà configurée, et ne peut être accédée qu'en lecture, alors qu'une PROM est une mémoire vide, qui peut être écrite, une seule fois, par l'utilisateur. Après cette unique écriture, une PROM se comporte exactement comme une ROM. Ces circuits, qui ne sont pas reprogrammables, sont répartis en deux sous-classes, les Mask-Programmable Chips, programmés par le fabricant, et les Field-Programmable Chips, qui sont programmés par l'utilisateur.

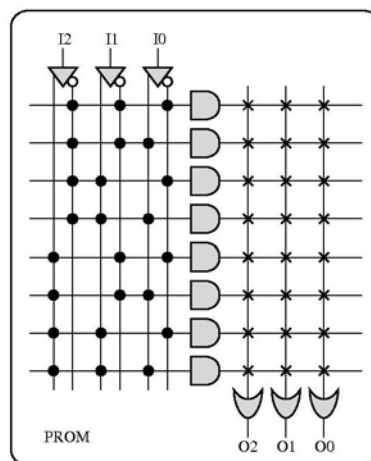


Fig. 3.11. L'architecture fonctionnelle d'une PROM.

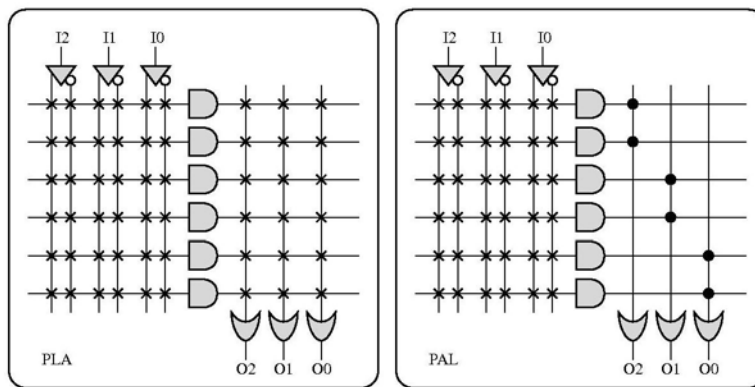
Notons également qu'à l'origine, les PROMs étaient vouées à faire office de mémoire d'instruction pour les ordinateurs. Leur emploi s'est toutefois généralisé puisqu'on les a utilisées pour l'implémentation de fonctions logiques simples tels que des look-up tables ou des machines d'état. Les PROMs se sont très vite imposées dans ce domaine, de par le fait

qu'elles étaient plus petites, moins chères, et moins sujettes aux erreurs que les systèmes composés de plusieurs circuits comportant des portes logiques. De plus, une erreur de design pouvait être rapidement modifiée en programmant une nouvelle PROM, ce qui était nettement plus rapide et plus simple que de modifier un circuit imprimé.

L'avantage des PROMs sur les autres PLDs est leur efficacité pour l'implémentation de fonctions logiques nécessitant un grand nombre de produits, mais elles ont le désavantage de n'accepter qu'un nombre limité d'entrées, étant donné que toutes les combinaisons possibles des entrées sont décodées (la figure 3.11 illustre ce décodage par la partie ET et la programmation de la partie OU).

PLA

Les PLAs (Programmable Logic Array) [40] apparurent aux environs de l'année 1975 dans le but de pallier les limitations des PROMs. En effet, dans un PLA (figure 3.12(a)), contrairement à une PROM, toutes les interconnexions peuvent être programmées, ce qui en fait le PLD le plus général. Il est alors possible de définir les produits et les sommes, les rendant particulièrement efficaces lorsque plusieurs sorties utilisent les mêmes produits. Une amélioration s'accompagnant souvent de désagréments, citons un des désavantages du PLA comparé à la PROM : étant donné que les deux tableaux (ET/OU) sont programmés, le temps de propagation d'un signal de l'entrée à la sortie est nettement plus important que lorsqu'un seul tableau l'est.



(a) L'architecture d'un PLA (b) L'architecture d'un PAL

Fig. 3.12. Architectures d'un PLA et d'un PAL.

PAL

Vers la fin des années 1970, les PALs (Programmable Array Logic) [41] furent introduites afin de contrer le problème de vitesse de propagation des PLAs. Dans un PAL (figure 3.12(b)), les connexions entre les portes ET et OU sont fixes, et les connexions entre les entrées et les portes ET peuvent être programmées. L'avantage des PALs sur les PLAs est donc leur rapidité, mais elles présentent l'inconvénient de n'avoir qu'un nombre limité de produits pour chaque porte OU.

4.2 CPLD

Les SPLDs présentent deux limitations majeures, à savoir l'impossibilité de réaliser des fonctions à plusieurs niveaux et celle de ne pouvoir partager les produits de différentes fonctions. Les CPLDs (Complex Programmable Logic Device), apparus au début des années 80, sont donc le résultat de l'évolution des PLDs. Ils permettent l'implémentation de systèmes nettement plus complexes, et sont composés d'éléments de base programmables, connectés entre eux par un réseau d'interconnexions relativement simple. Ces éléments de base sont du type SPLD, et peuvent, comme dans le cas de la famille MAX3000 d'Altera, être composés d'un tableau de portes ET programmables et de portes OU fixes (une sorte de PAL), ainsi que d'un registre [42]. La technologie de programmation des CPLDs dépend évidemment du constructeur, et peut être de type EPROM, EEPROM, FLASH ou SRAM.

Notons qu'un des avantages des CPLDs sur les FPGAs, que nous présenterons plus loin, est la rapidité. En effet, le réseau d'interconnexions, en étant nettement plus simple, est plus rapide que celui d'un FPGA (figure 3.13). De plus, les connexions se font toujours avec une destination pour une source, et le temps de propagation est donc toujours le même. Le placement d'un design dans un CPLD n'est donc pas critique, et le routage peut être systématisé, sans avoir besoin de tenir compte de contraintes de temps.

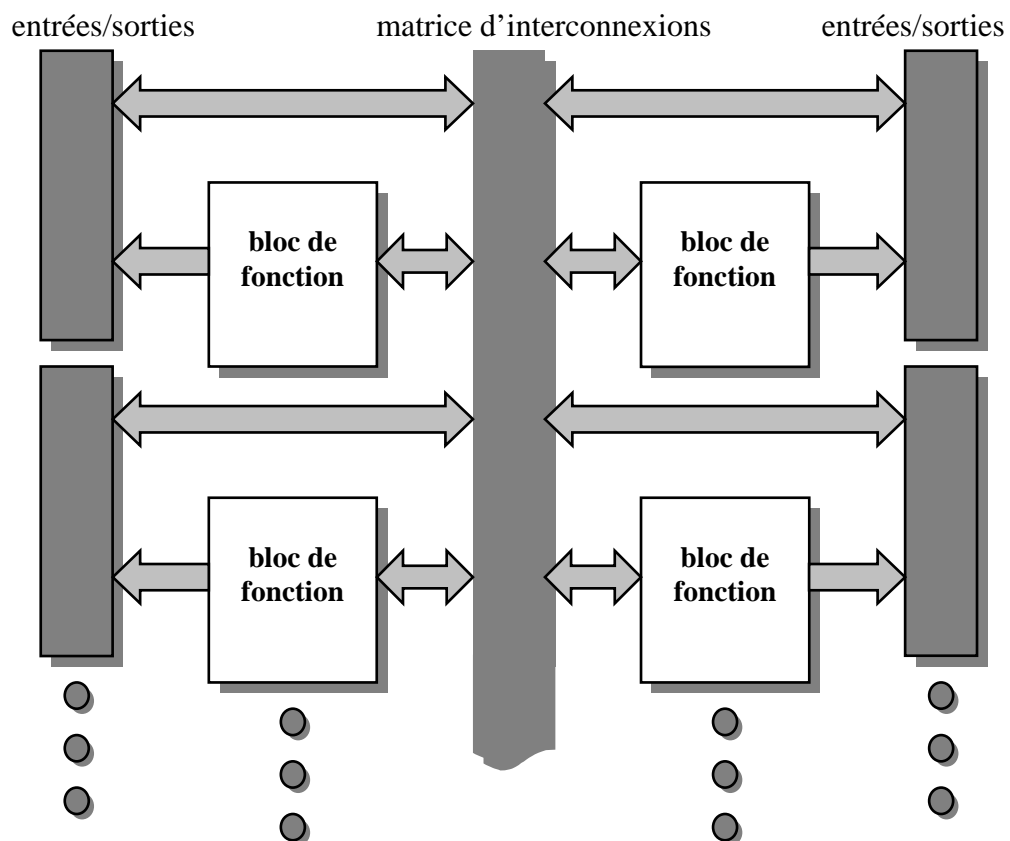


Fig. 3.13. L'architecture d'un CPLD

Le routage des CPLDs commence à être intéressant, puisqu'en plus de définir la fonctionnalité de blocs simples comme les SPLDs, il devient possible de les interconnecter.

CPLD XC9500 de Xilinx

Nous allons présenter ici une famille des circuits CPLD de Xilinx, à savoir la série XC9500 [43]. Les principales caractéristiques des XC9500 sont données comme suit:

- Programmabilité : mémoire FLASH
- Supporte jusqu'à 10000 cycles d'écriture
- 36 - 288 macrocellules (6400 portes)
- Délai de propagation entrée-sortie de 5-10 ns
- Signaux globaux : Reset, horloge et tri-state (haute impédance)

Tout circuit XC9500 est composé d'une multitude de blocs fonctionnels et de blocs d'entrées sorties complètement interconnectés par la matrice de commutation Fast CONNECT™. La figure 3.14 représente l'architecture du XC9500.

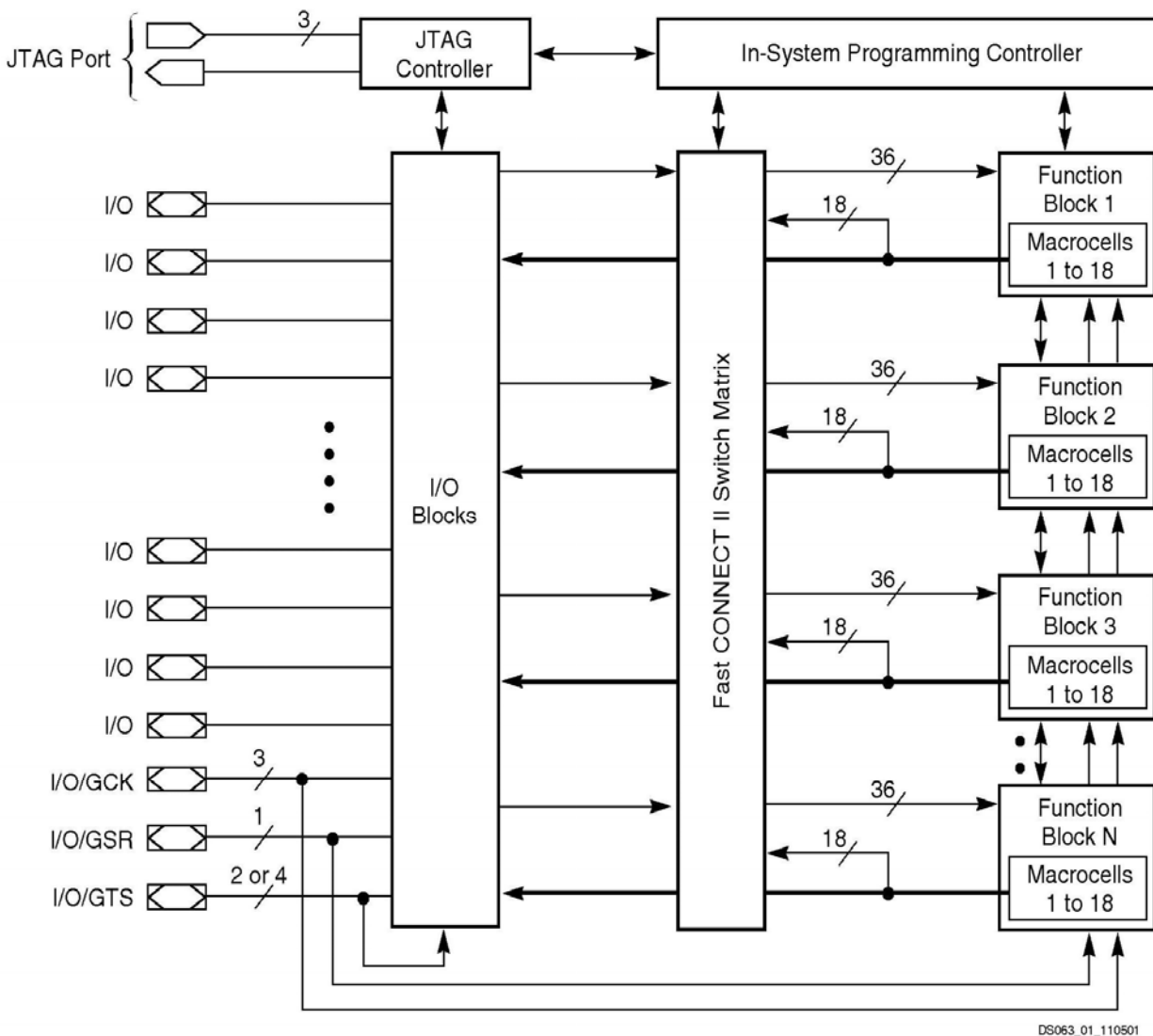


Fig. 3.14. Architecture du XC9500

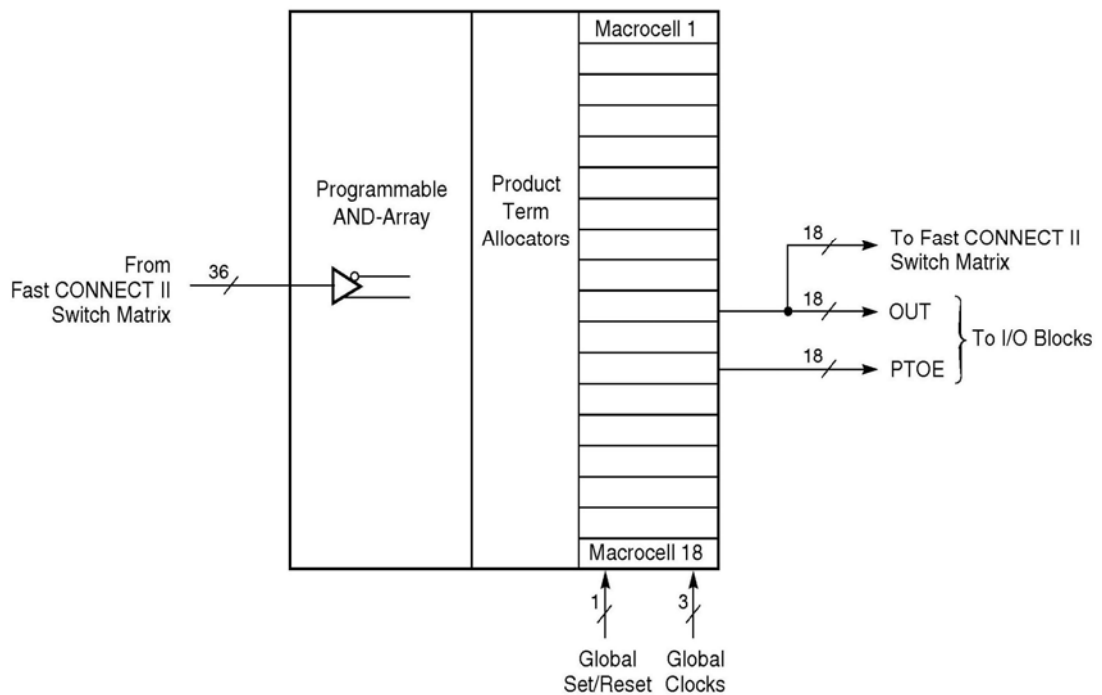
TABLEAU 3.1

Famille des circuits XC9500

	XC9536	XC9572	XC95108	XC95144	XC95216	XC95288
<i>Macrocellules</i>	36	72	108	144	216	288
<i>Portes utilisables</i>	800	1,600	2,400	3,200	4,800	6,400
<i>Registres</i>	36	72	108	144	216	288
<i>TPD (ns)</i>	5	7.5	7.5	7.5	10	15
<i>E/S max</i>	34	72	108	133	166	192
<i>fCNT (MHz)</i>	100	125	125	125	111.1	92.2
<i>fSYSTEM (MHz)</i>	100	83.3	83.3	83.3	66.7	56.6

Bloc fonctionnel du XC9500

Chaque bloc fonctionnel (FB) donne une capacité de programmation logique avec 36 entrées et 18 sorties [44]. Tout bloc fonctionnel comporte 18 macrocellules indépendantes, chacune capable d'implémenter une fonction logique combinatoire ou séquentielle. Les figures 3.15 et 3.16 représentent respectivement le bloc fonctionnel du XC9500 et ses connexions.



DS063_02_110501

Fig. 3.15. Bloc fonctionnel du XC9500

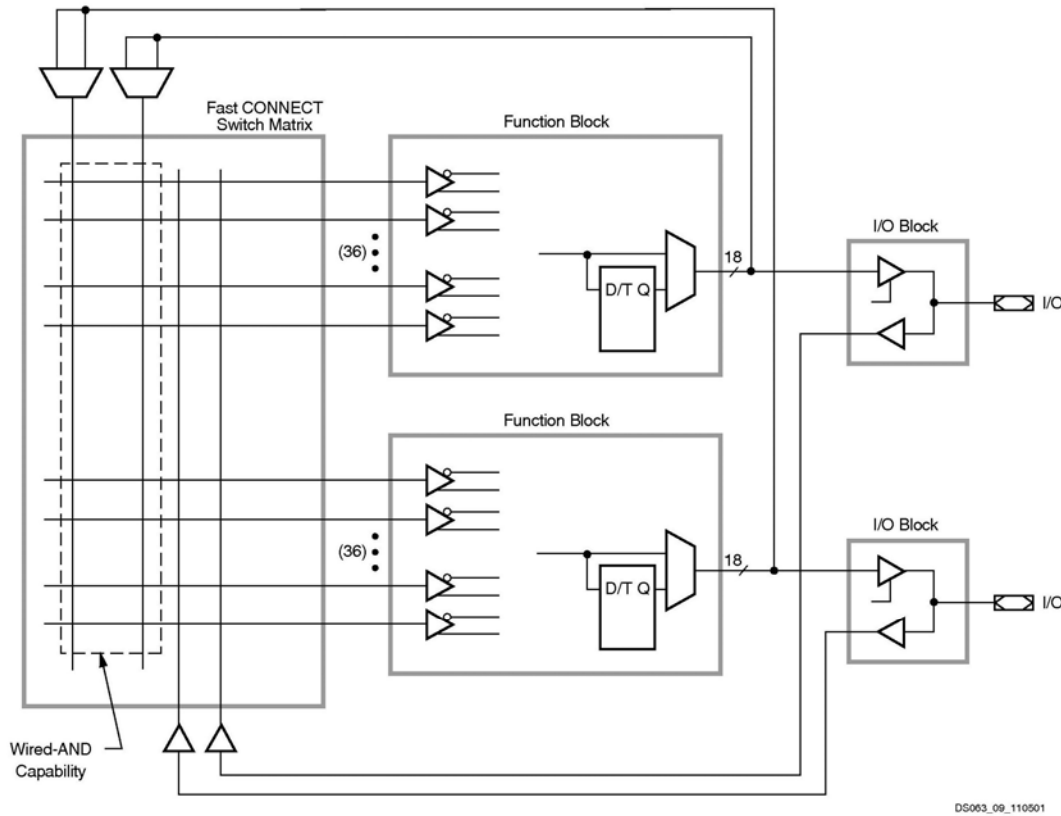


Fig. 3.16. Connexion du bloc fonctionnel du XC9500

4.3. FPGAs

Nous présentons dans cette partie les circuits PLD de type FPGA (Field Programmable Gate Array). Au début des années 80, les développeurs disposaient des circuits de type PLD, facilement configurables, mais ne pouvant contenir des architectures considérablement complexes. Les ASICs, quant à eux, supportaient des systèmes de grande complexité, mais n'avaient pas les propriétés de configuration des PLDs ni celles de la reconfiguration. Il manquait donc un type de circuits permettant la réalisation de systèmes complexes, tout en offrant une reconfiguration rapide et peu coûteuse. C'est pourquoi, Ross Freeman, Bernie Vonderschmitt, et Jim Barnett fondateurs de la compagnie Xilinx (1984), introduisent en 1985 sur le marché le premier FPGA, le XC2064, et offrent ainsi une alternative aux précédentes approches.

Les circuits FPGAs [45,46] permettent d'implémenter des systèmes numériques aussi complexes que ceux réalisés jusqu'alors grâce aux ASICs, tout en ayant le grand avantage de pouvoir être programmés électriquement. Ils sont principalement composés d'un tableau d'éléments plus ou moins complexes pouvant être configurés, ainsi que d'un réseau complexe de connexions également configurables (figure 3.17).

Nous allons présenter ici les circuits de Xilinx, les circuits d'Altera présentant le même type d'architecture. Nous commencerons par le XC2064, le premier FPGA mis sur le marché. Ensuite nous aborderons la famille XC6200 [47, 48, 49] et Nous terminerons par les deux familles Spartan et Virtex, les plus utilisées dans les applications de matériel évolutif.

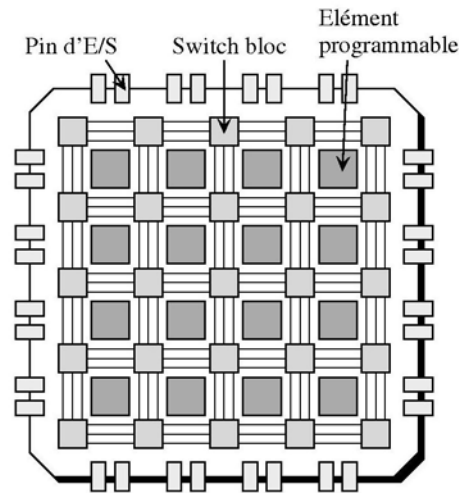


Fig. 3.17. L'architecture générale du FPGA.

Nous verrons également la manière dont les circuits se complexifient en embarquant de plus en plus de composants tels que RAM, multiplicateurs, processeurs, et autres. Afin de ne pas surcharger ce chapitre, nous ne présenterons pas en détail les derniers-nés des FPGAs [50, 51], leur architecture n'étant qu'une variation évoluée du XC2064, mais nous présentons l'architecture générale de la série Spartan-3E et de la série Virtex-II.

4.3.1. XC2000

La série XC2000 de Xilinx représente le premier FPGA commercial [52]. Son architecture est relativement simple en comparaison des énormes circuits actuels, mais sensiblement identique à l'ensemble des FPGAs. Il est basé sur une technologie SRAM, de la même manière que tous les FPGAs de Xilinx, et est donc reprogrammable un nombre illimité de fois, et ce de manière très rapide. Son élément de base, le CLB, pour Configurable Logic Bloc (figure 3.18), est composé d'une bascule et de deux look-up tables de trois entrées qui ont également la possibilité de réaliser une look-up table à quatre entrées (suivant la dénomination introduite par Hill et Woo dans [53]). Il s'agit d'une look-up table à 4 entrées et 2 sorties). Il est intéressant de noter que les deux sorties, X et Y, sont interchangeables, puisque leurs multiplexeurs ont exactement les mêmes entrées. Ce détail est très utile lors de l'opération de routage.

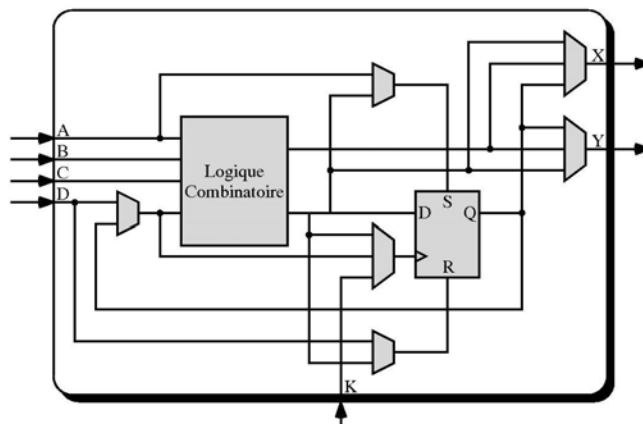


Fig. 3.18. Le CLB d'un XC2000.

Sur le plan des interconnexions, le XC2000 contient des switchboxes, chacun étant connecté à quatre autres switchboxes, comme présenté sur la figure 3.19. Ils sont reliés verticalement par cinq fils, et horizontalement par quatre. Ces switchboxes permettent de relier des CLBs sur de longues distances, au travers du FPGA, les problèmes liés au délai RC des noeuds routés étant évités en plaçant des buffers sur certains fils. Le choix a été fait de diviser le FPGA en neuf parties, et de placer des buffers sur toutes les frontières de ces parties. Cette structure de routage est encore présente dans les FPGAs actuels, dans lesquels nous trouvons des canaux de routage accessibles par les unités fonctionnelles, et qui sont reliés entre eux par des switchboxes.

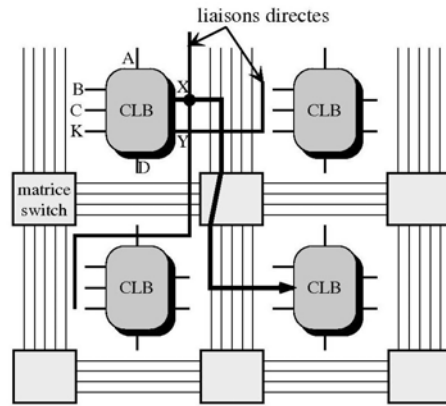


Fig. 3.19. Le schéma d'interconnexions d'un XC2000.

Afin de pouvoir implémenter des designs encore plus efficacement, des liaisons directes sont également disponibles (figure 3.19). Elles permettent de connecter un CLB à ses quatre voisins, sans accéder aux liaisons des switchboxes, et réduisent dès lors le temps de propagation du signal, ainsi que les problèmes de congestion du routage.

La figure 3.20 montre les différents bits de configuration liés à l'interconnexion des blocs. Chaque carré correspond à un élément de mémoire SRAM, relié à un transistor qui connecte ou non les lignes verticales et horizontales. Nous pouvons noter que chacune des deux sorties X et Y est connectée à un nombre réduit de lignes. Si le routeur rencontre des difficultés à acheminer un signal, il peut simplement interchanger les deux sorties, pour avoir de nouvelles possibilités de routage. De même pour les entrées, qui ne sont pas connectées aux mêmes lignes, le routeur peut les interchanger, en modifiant toutefois le contenu de la look-up table de manière à garder la même fonctionnalité.

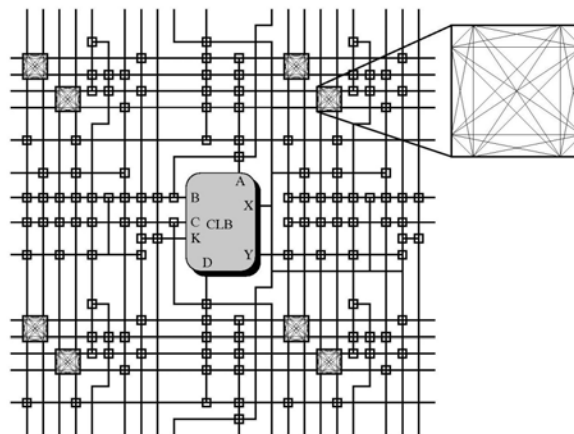


Fig. 3.20. Les connexions d'un bloc d'un XC2000.

Pour terminer, il est important de constater que la configuration du FPGA dans un état correct est cruciale. En effet, il est très facile de créer des courts-circuits en connectant plusieurs sorties de CLB ensemble, ce qui peut être très dommageable pour le circuit.

4.3.2 XC6200

La famille XC6200 de Xilinx [47] fut très importante pour l'ensemble de la communauté intéressée par le matériel évolutif. En effet, son architecture est encore plus simple que celle des XC2000, autant en ce qui concerne les blocs logiques que le réseau de routage. De plus, le point central de son attrait est le fait qu'elle est basée sur une technologie SRAM couplée à des multiplexeurs. Tout court-circuit est alors impossible, rendant aisée l'évolution de systèmes au niveau des portes.

Son bloc de base, présenté par la figure 3.21, ne possède que trois entrées, et sa fonctionnalité n'est pas réalisée par une look-up table, mais par des multiplexeurs, ce qui, dans cette configuration, ne permet pas d'implémenter n'importe quelle fonction à trois variables. Une bascule est placée de la même manière que dans un XC2000, et l'unique sortie peut donc être combinatoire ou séquentielle.

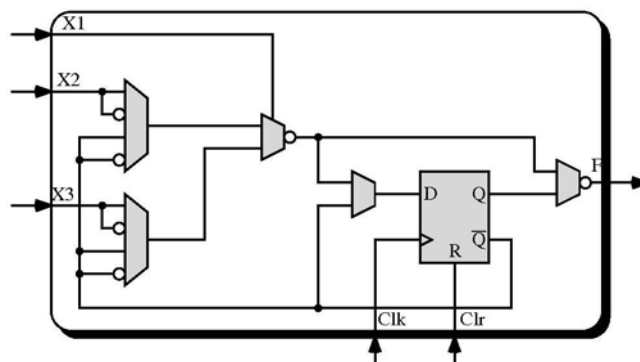


Fig. 3.21. L'unité fonctionnelle d'un XC6200.

Concernant les interconnexions, la figure 3.22 détaille la manière dont les entrées et la sortie de l'unité fonctionnelle sont connectées aux unités voisines. Nous pouvons observer que les valeurs des quatre voisins immédiats peuvent directement être récupérées par l'unité fonctionnelle. Outre ces liaisons courte-distance, il existe des lignes d'une longueur de quatre cellules, seize cellules, ou parcourant le circuit entier. Toutefois, leur nombre est très réduit en comparaison de tous les autres types de FPGAs. Un circuit de la famille XC6200 est donc efficace pour des systèmes ne demandant que peu de ressources de routage de distance plus importante que le simple voisinage.

4.3.3 Comparaison des architectures MUX et LUT

Deux types d'architectures sont utilisés par les fabricants, certains basant les éléments logiques sur des multiplexeurs (MUX), à la manière de la XC6200, et d'autres sur des look-up tables (LUT), comme dans la XC2000 [54].

Dans l'approche MUX, la fonctionnalité est réalisée en connectant les entrées et le signal de sélection des multiplexeurs. De la logique peut également être ajoutée, les multiplexeurs sélectionnant alors une parmi plusieurs fonctions.

L'implémentation des éléments logiques grâce à une ou des LUTs permet quant à elle de réaliser n'importe quelle fonction, en programmant correctement les bits de configuration de la LUT. Cette approche a été choisie par Altera et Xilinx, les deux plus gros fabricants. Cette alternative permet d'implémenter des registres à décalage et des mémoires, en tirant parti des bits de configuration des LUTs, ce que les multiplexeurs ne peuvent faire. Son désavantage est toutefois de n'être que peu efficace pour les applications réalisées par un grand nombre de fonctions logiques simples et indépendantes. En effet, une fonction à 2 entrées nécessite la réquisition d'une LUT entière, alors qu'elle pourrait n'être implémentée que sur un multiplexeur. Concernant la taille optimale d'une LUT, les recherches ont montré que l'optimum se situait à 4 entrées [53, 55, 56].

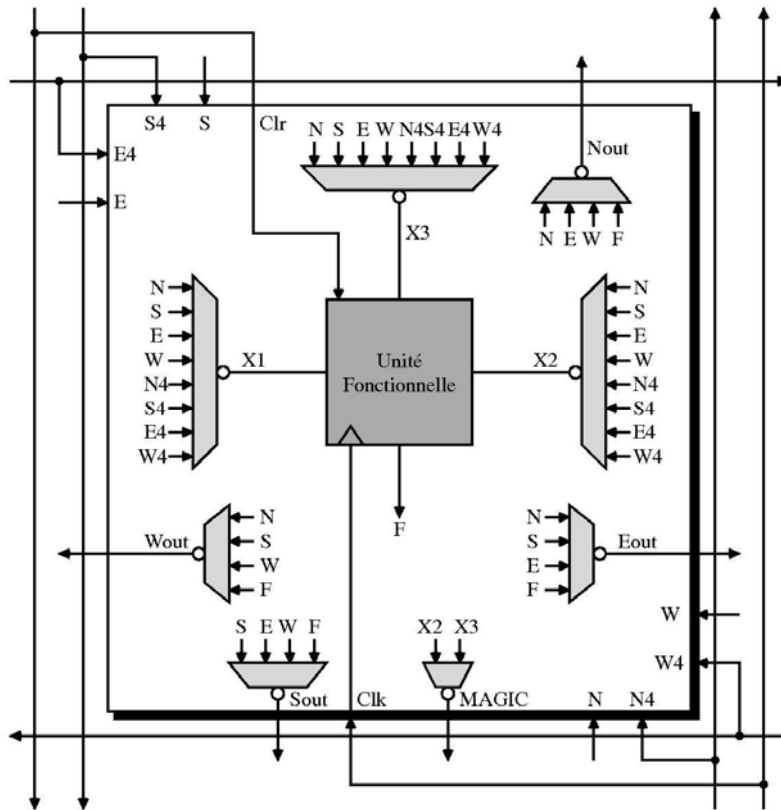


Fig. 3.22. La cellule de base du XC6200.

4.3.4 Spartan™ et Virtex™

Actuellement, Xilinx, le premier fabricant de FPGAs, propose principalement deux familles, Virtex et Spartan, toutes deux de type SRAM. Les deux familles contiennent des multiplieurs des blocs de RAM, pouvant être utilisés en single ou dual port. La figure 3.23 présente l'architecture générale de la série Spartan-3E [57].

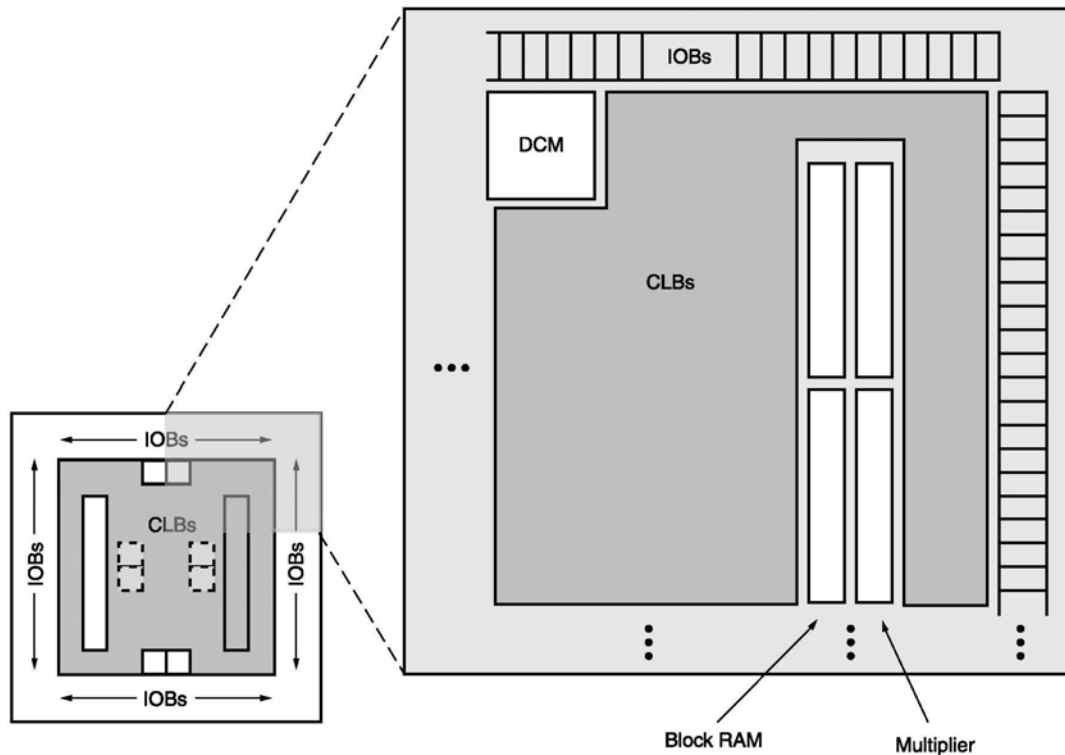


Fig. 3.23. L'architecture générale de la série Spartan-3E.

- Les CLBs (Configurable Logic Blocks) possèdent des "look-up tables"(LUTs) qui permettent d'implémenter n'importe quelle fonction logique.
- Les blocs DCM (Digital Clock Manager Blocks) offrent un auto-calibrage, temporisation, multiplication, division et décalage de phase des signaux d'horloge.
- Les Blocs d'entrée-sortie (IOBs) permettent de contrôler l'échange des données entre les pins d'entrée-sortie et les blocs logiques internes. Chaque IOB supporte un échange de données bidirectionnelles et la logique 3 états. Les registres DDR (Double Data-Rate) sont inclus.
- Les blocs de RAM fournissent un stockage de données sous forme de blocs 18-Kbit avec un accès double.
- Les blocs de multiplicateurs acceptent deux nombres binaires de 18 bit comme entrées et assurent le calcul du produit.

Le Virtex-II offre la même architecture avec des technologies nouvelles. La figure 3.24 présente l'architecture générale de la série Virtex-II. Les Virtex-4 [58] et Virtex-5 [59] représentent les deux dernières versions des FPGA de Xilinx.

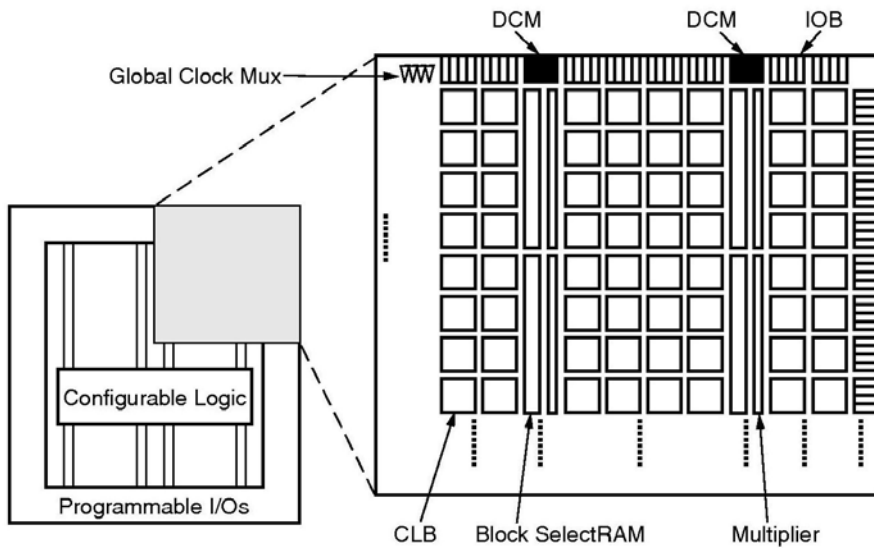


Fig. 3.24. L'architecture générale de la série Virtex-II

Les Virtex-4 [60] sont fabriqués avec une technologie de 90 nm et contiennent des blocs multiplieurs accumulateurs (MAC) de 18×18bits et des blocs de RAM de 18 Kbits. Cette série possède jusqu'à 512 "500 Mhz DSP48 Slice" [61].

La technologie de 65 nm est employée pour la fabrication des FPGA de la série Virtex-5 [62]. Ces derniers contiennent des blocs multiplieurs accumulateurs (MAC) de 25×18bits, des blocs de RAM de 36 Kbits et jusqu'à 640 "550 Mhz DSP48E Slice". La figure 3.25 illustre l'architecture du "Virtex-5 SXT DSP48E Slice" [63].

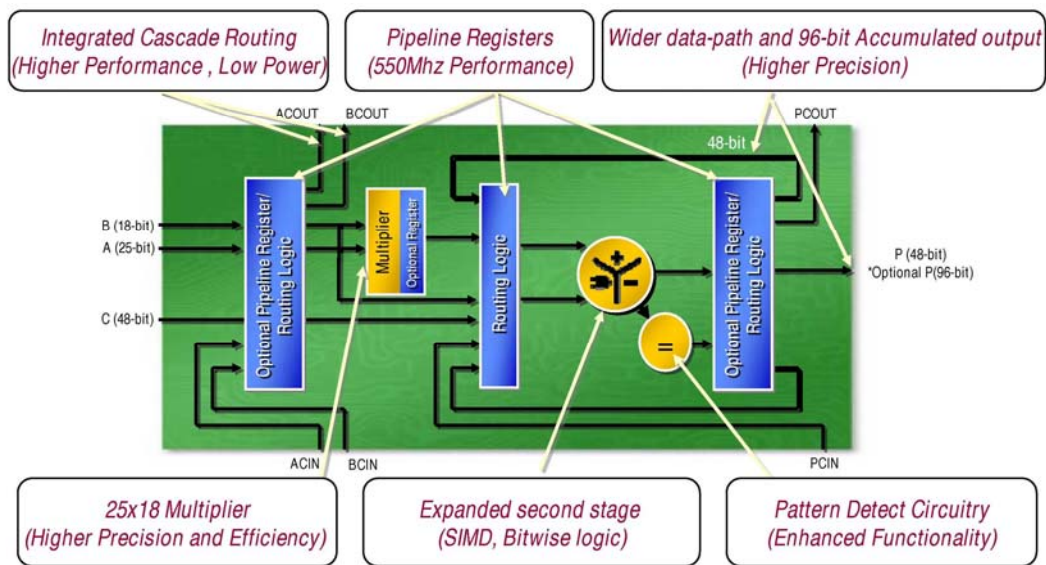


Fig. 3.25. L'architecture du "Virtex-5 SXT DSP48E Slice"

5. Implémentation de l'Adressage Physique Etendu

5.1. Implémentation schématique de l'Adressage Physique Etendu

Pour tester le système de l'Adressage Physique Etendu, nous avons réalisés [64] une implémentation sur **CPLD** avec le **ISE** de **Xilinx** "Integrated Software Environment". Le **ECS** de **Xilinx** "Engineering Capture System" est exploité pour la production schématique [65,66]. Un circuit **XC9572-7PC84** est utilisé pour une première implémentation [60]. Nous avons utilisé une architecture d'un Adressage Physique Etendu d'ordre deux. Dans cet exemple, le bus système utilisé est composé d'un bus de données de 8 bits, d'un bus d'adresses de 12 bits et d'un bus de contrôle. Ce dernier est composé de AEN, IOR et IOW. Ce dernier est composé de AEN, IOR et IOW.

Un model d'une implémentation schématique est représenté sur la figure 3.26. **BUFE8** est un amplificateur de bus, interne à 3 états activé à l'état haut. **LD8** est un **Latch** multiple de données. Le dec301_2_3 symbolise le décodeur d'adresses.

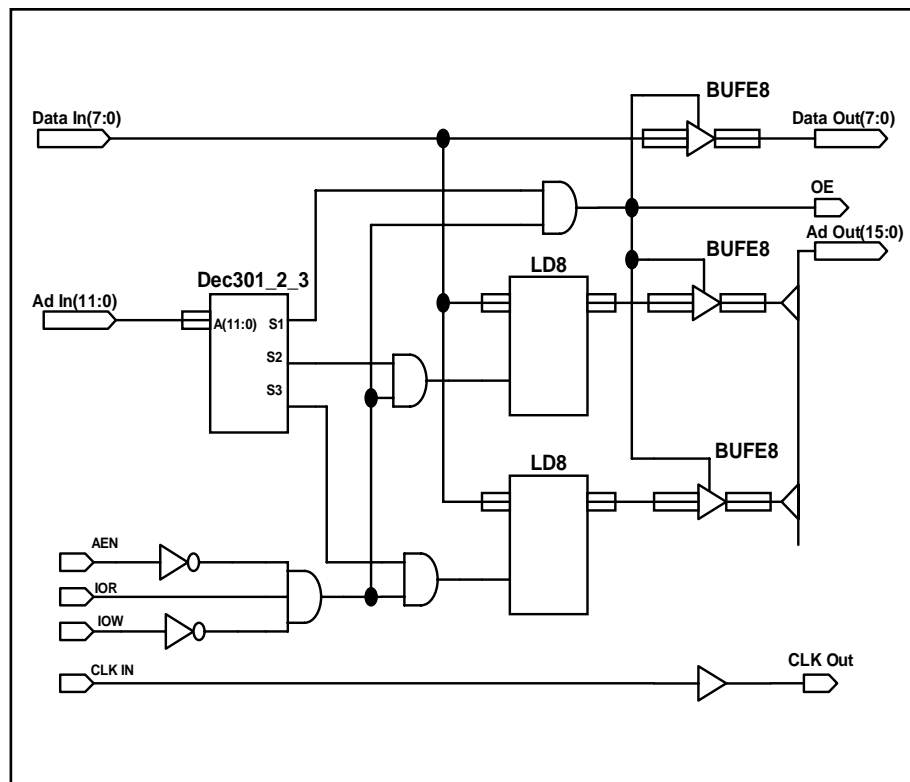


Fig. 3.26. Schéma de l'Adressage Physique Etendu avec ECS

5.2. Implémentation VHDL de l'Adressage Physique Etendu

L'instance et déclaration VHDL du composant est présenté sur la figure 3.27 [68]. La description VHDL de l'entité et de l'architecture du décodeur d'adresses **dec301_2_3** est donnée par la figure 3.28.

```

Vhdl instantiation template
COMPONENT epa01_sch
PORT( Ad_In : IN STD_LOGIC_VECTOR (11 DOWNT0 0);
      AEN : IN STD_LOGIC;
      Data_In : IN STD_LOGIC_VECTOR (7 DOWNT0 0);
      IOR : IN STD_LOGIC;
      IOW : IN STD_LOGIC;
      OE : OUT STD_LOGIC;
      Data_Out : OUT STD_LOGIC_VECTOR (7 DOWNT0 0);
      Ad_Out : OUT STD_LOGIC_VECTOR (15 DOWNT0 0);
      CLK_In : IN STD_LOGIC;
      CLK_Out : OUT STD_LOGIC);
END COMPONENT;
UUT: epa01_sch PORT MAP(
  Ad_In =>,
  AEN =>,
  Data_In =>,
  IOR =>,
  IOW =>,
  OE =>,
  Data_Out =>,
  Ad_Out =>,
  CLK_In =>,
  CLK_Out =>
);

```

Fig. 3.27. "VHDL instantiation template" de l'Adressage Physique Etendu.

```

entity dec301_2_3 is
  port (A : in std_logic_vector (15 downto 0);
        S1 : out std_logic;
        S2 : out std_logic;
        S3 : out std_logic);
end dec301_2_3;

architecture DESCRIPTION of dec301_2_3 is
begin
  S1 <= '1' when A = x"0301" else '0';
  S2 <= '1' when A = x"0302" else '0';
  S3 <= '1' when A = x"0303" else '0';

end DESCRIPTION;

```

Fig. 3.28. La description VHDL de l'entité et de l'architecture, du décodeur d'adresses dec301_2_3.

La figure (3.29) montre la fenêtre du "Test Bench Waveform" ouverte avec le "Project Navigator" de Xilinx. Le "Test Bench Waveform" présente la forme des signaux et les valeurs attribués au bus de données et d'adresses du bus système. Le bus d'adresses est piloté par les lignes Ad_In(11:0) et le bus de données par les lignes Data_In(8:0).

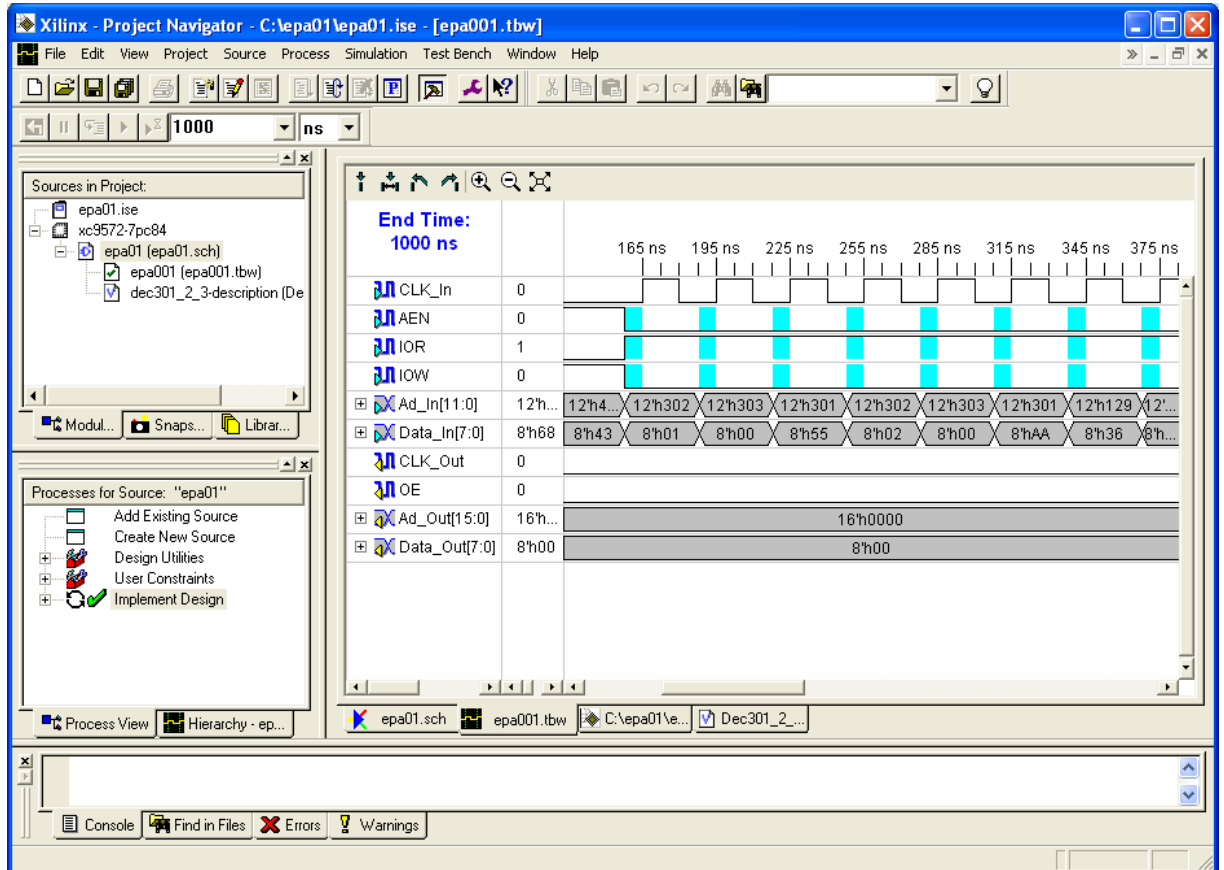


Fig. 3.29. Les signaux "Test Bench Waveform" pour l'Adressage Physique Etendu.

Les valeurs 55H et AAH (valeurs exprimées en hexadécimal) seront activées respectivement, sur le nouveau bus de données, avec les valeurs 0001H et 0002H sur le nouveau bus d'adresses (Figure 3.30).

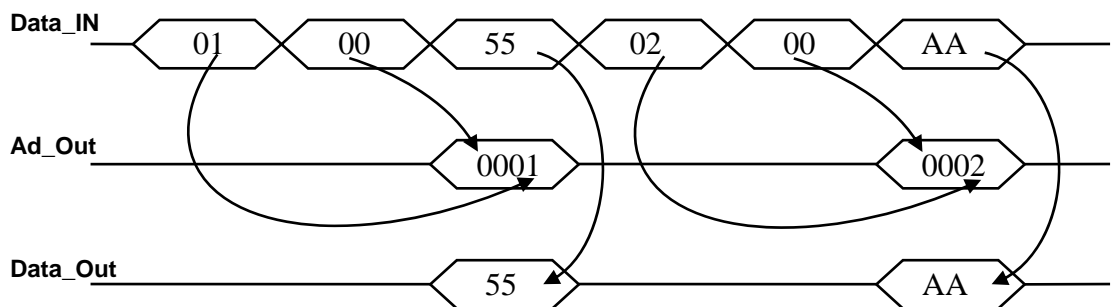


Fig. 3.30. Les phases de la conversion données-adresses.

La simulation est effectuée par le simulateur "ModelSim" version SE de "Mentor Graphics Corporation". Les valeurs, affectés dans la fenêtre du "Test Bench Waveform", fournissent les résultats de la fenêtre "Waveform" du "ModelSim" (figure 3.31).

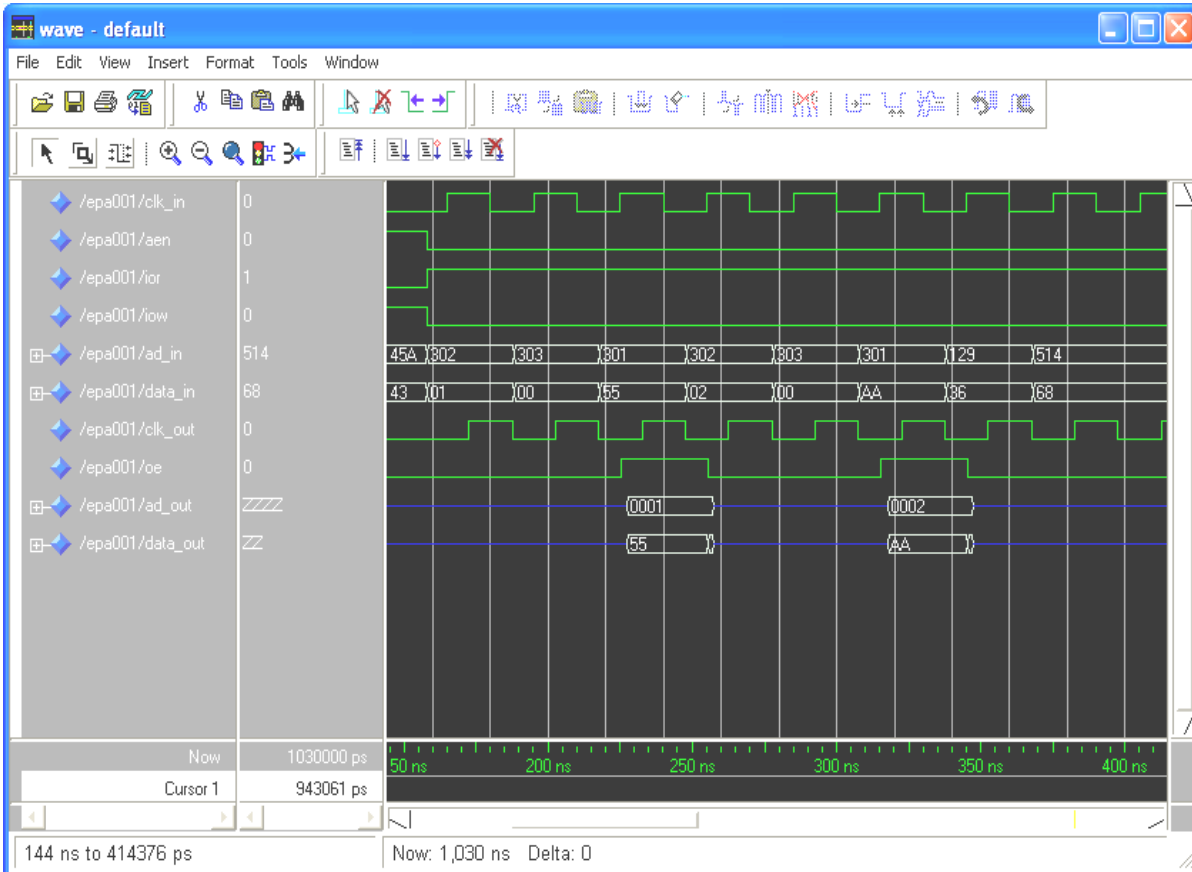


Fig. 3.31. Le résultat de la simulation de l'Adressage Physique Etendu

6. Implémentation de l'Adressage Physique Rapide

6.1. Implémentation schématique de l'Adressage Physique Rapide

L'implémentation proposée avec la logique programmable de l'Adressage Physique Rapide en utilisant le ECS de Xilinx est présentée avec un schéma de base sur la figure 3.32 [69,70]. Une implémentation sur CPLD avec le ISE "Xilinx Integrated Software Environment" de Xilinx est réalisée sur un circuit XC95108-10PC84 [71]. Nous avons implémenté un exemple d'une architecture d'un Adressage Physique Rapide à 16 bits. Le bus système utilisé est composé d'un bus de données de 16 bits et d'un bus d'adresses de 16 bits. La ligne DA, du bus système, assure l'autorisation de la DMA par le système à microprocesseur.

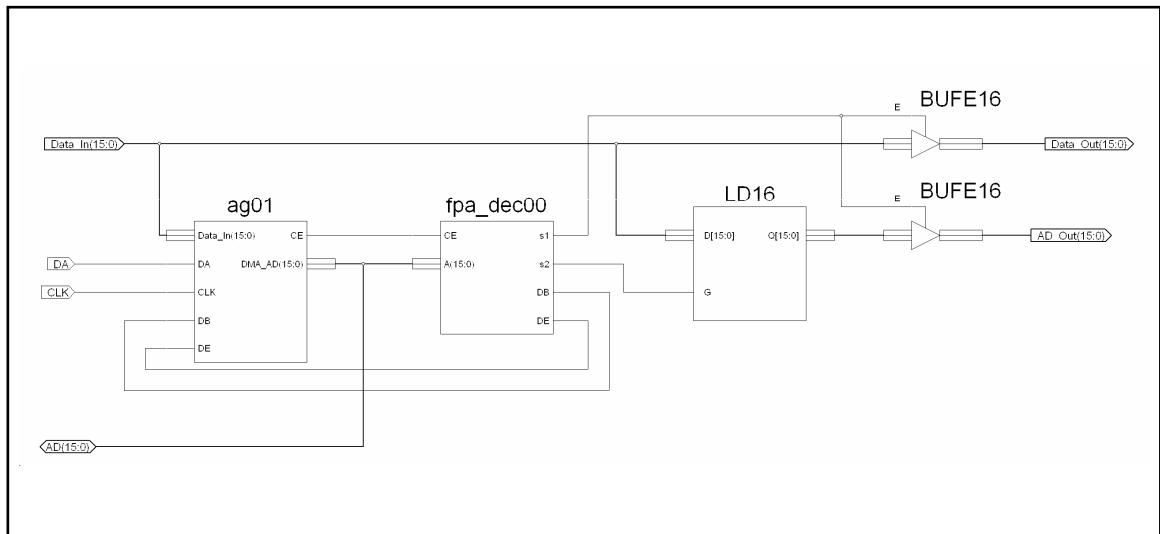


Fig. 3.32. Schéma de l'Adressage Physique Rapide avec ECS

BUFE16 est un amplificateur de bus, interne à 3 états, activé à l'état haut et LD16 est un **Latch** multiple de données. Le circuit **fpa_dec00** représente le décodeur d'adresses de l'Adressage Physique Rapide et **ag01** représente le Générateur d'Adresses. Le schéma du Générateur d'Adresses proposé est illustré sur la figure 3.33.

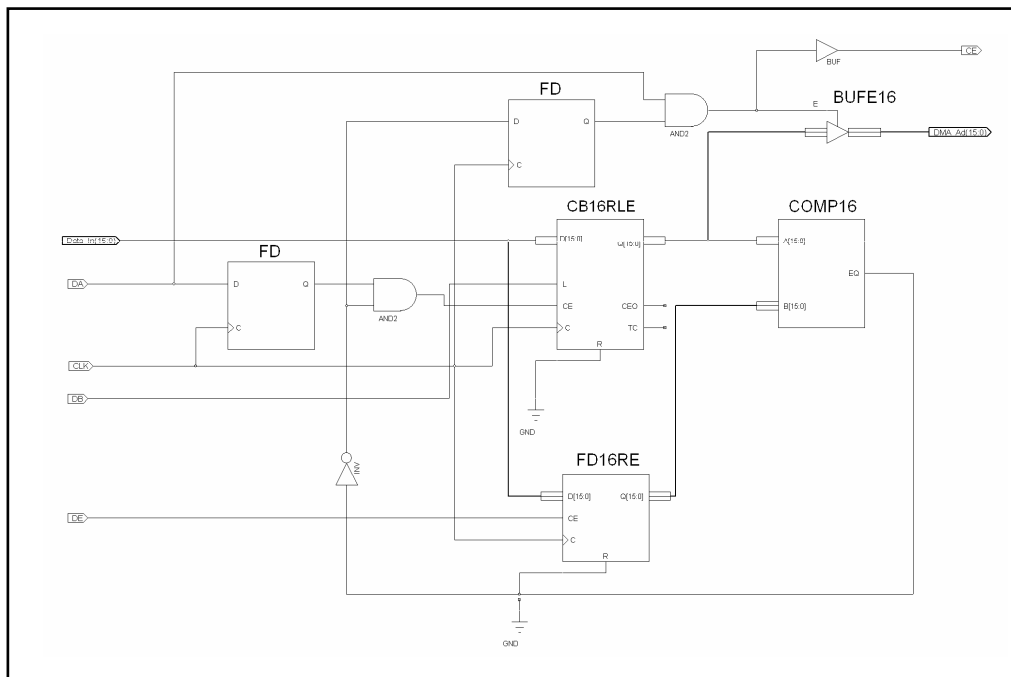


Fig. 3.33. Schéma du Générateur d'Adresses proposé

6.2. Implémentation VHDL de l'Adressage Physique Rapide

L'instance et déclaration VHDL [72] de composant est présenté sur la figure 3.34. La description VHDL de l'entité et de l'architecture est donnée par la figure 3.35.

```

Vhdl instantiation template
COMPONENT fpa00
PORT(AD : INOUT STD_LOGIC_VECTOR (15 DOWNTO 0);
      Data_In : IN STD_LOGIC_VECTOR (15 DOWNTO 0);
      DA : IN STD_LOGIC;
      CLK : IN STD_LOGIC;
      Data_Out : OUTSTD_LOGIC_VECTOR (15 DOWNTO 0);
      AD_Out : OUT STD_LOGIC_VECTOR (15 DOWNTO 0));
END COMPONENT;

UUT: fpa00 PORT MAP(
      AD =>,
      Data_In =>,
      DA =>,
      CLK =>,
      Data_Out =>,
      AD_Out =>
);

```

Fig. 3.34. "VHDL instantiation template" de l'Adressage Physique Rapide

```

entity fpa_dec00 is
  port(
    CE : in std_logic;
    A : in std_logic_vector(15 downto 0);
    DB : out std_logic;
    DE : out std_logic;
    s1 : out std_logic;
    s2 : out std_logic);
end fpa_dec00;
architecture DESCRIPTION of fpa_dec00 is
begin
  process(CE,A)
  begin
    if ((CE='0' and A=x"0301") or (CE='1' and A(0)='1')) then
      s1 <= '1';
    else
      s1 <= '0';
    end if;
    if((CE='0' and A=x"0302") or (CE='1' and A(1)='1'))then
      s2 <= '1';
    else
      s2 <= '0';
    end if;
    if(CE='0' and A=x"0303") then
      DB <= '1';
    else
      DB <= '0';
    end if;
    if(CE='0' and A=x"0304") then
      DE <= '1';
    else
      DE <= '0';
    end if;
  end process;
end DESCRIPTION;

```

Fig. 3.35. La description VHDL de l'entité et de l'architecture du décodeur d'adresses.

La description VHDL de l'entité du Générateur d'Adresses est illustrée dans figure 3.36.

```

entity ag01 is
  port (CLK   : in std_logic;
        DA    : in std_logic;
        Data_In : in std_logic_vector (15 downto 0);
        DB    : in std_logic;
        DE    : in std_logic;
        CE    : out std_logic;
        DMA_AD : out std_logic_vector (15 downto 0));
end ag01;

architecture BEHAVIORAL of ag01 is
begin
end BEHAVIORAL;

```

Fig. 3.36. Entité du Générateur d'Adresses.

Le "Test Bench Waveform" de la figure 3.37 présente la forme des signaux et les valeurs attribués au bus de données. Le processus d'accès direct à la mémoire est activé par le signal DMA du bus système.

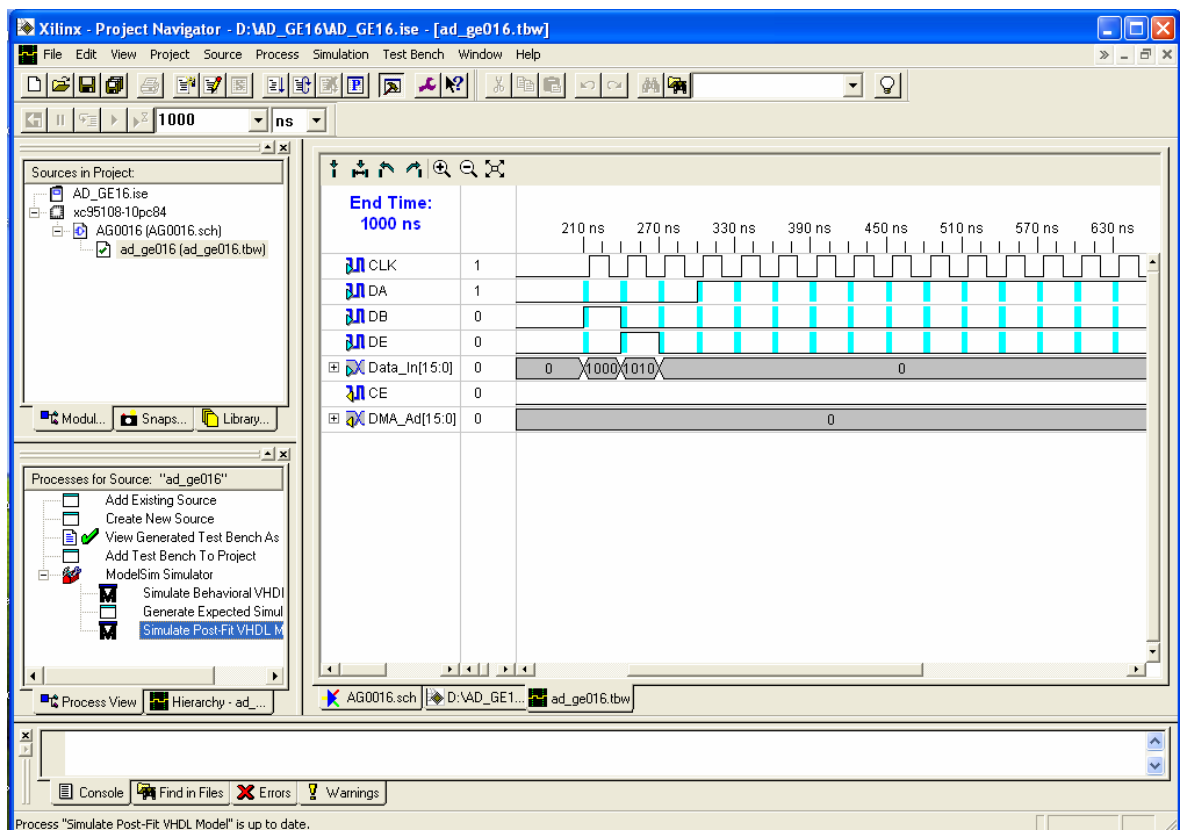


Fig. 3.37. Les signaux "Test Bench Waveform" d'activations de la DMA.

Les valeurs 1000 et 1010 (valeurs exprimées en décimal) fournissent les deux positions, début et fin, de la zone mémoire autorisée à être accessible par notre système (Figure 3.38). Le résultat de la simulation est donné par la figure 3.39.

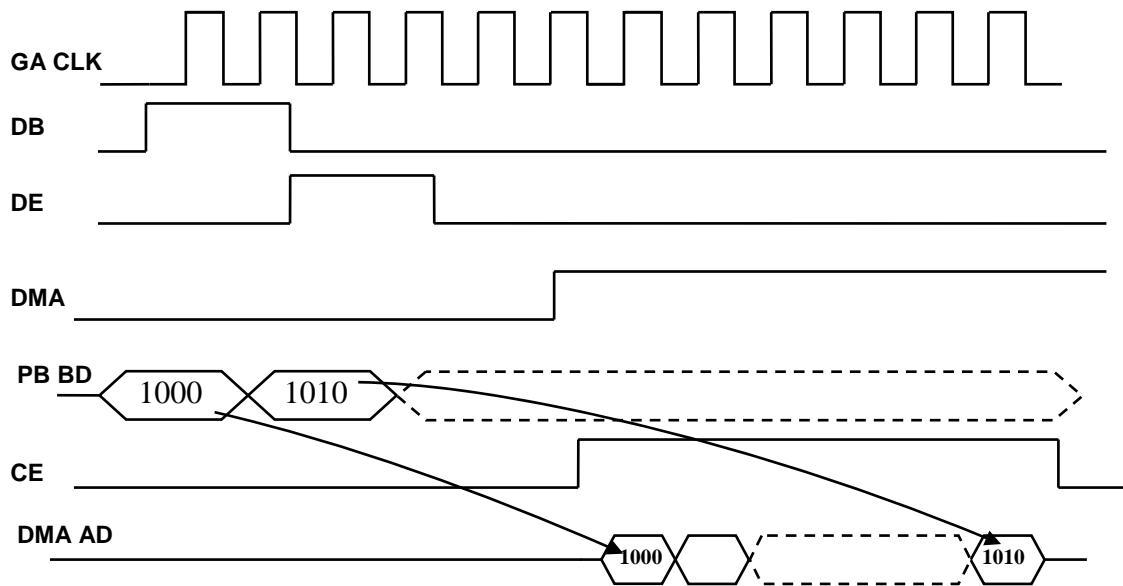


Fig. 3.38. Les phases de la génération d'adresses

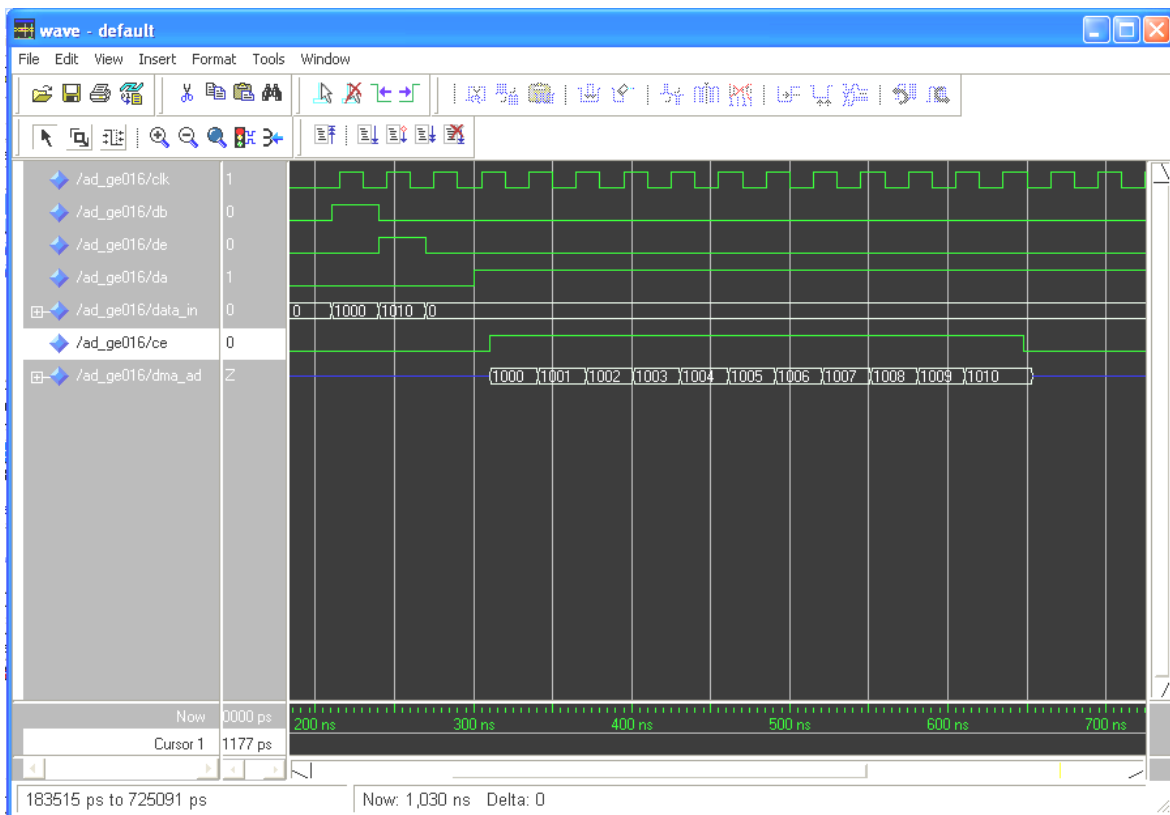


Fig. 3.39 Le résultat de la simulation du Générateur d'Adresses

L'application pratique de cette version sur un bus système qui travaille avec un bus de données de N bits et une fréquence maximale F , donne un nouveau bus de N bits de données, d'une capacité d'adressage physique de 2^N et qui travaille à une fréquence maximale égale à Fn . Le tableau 3.2 représente les principales caractéristiques testées de l'Adressage Physique Rapide.

TABLEAU 3.2

Caractéristiques testées de l'Adressage Physique Rapide

FI	NDS	NAS	NDN	NAN	FO
F	8 bits	02	8 bits	256	Fn
F	16 bits	02	16 bits	64 K	Fn
F	32 bits	02	32 bits	4 G	Fn

NAN : Nombre d'adresses physiques disponibles sur le nouveau bus, NDN : Taille des données sur le nouveau bus.
 NAS : Nombre d'adresses physiques occupées sur le bus système, NDS : Taille des données sur le bus système.
 FI : Fréquence de travail du bus système, FO : Fréquence de travail du nouveau bus.

La fréquence maximale du nouveau bus est obtenue par la formule suivante :

$$Fn = \frac{1}{2 \times Ta} \quad (3.1)$$

Ta : Temps d'accès de la mémoire utilisée par le système à microprocesseur durant le processus de l'accès direct à la mémoire.

L'Adressage Physique Rapide est appliqué sur un bus système qui fonctionne à une fréquence de 33 MHz, un bus de données de 32 bits et un temps d'accès de la mémoire utilisée dans la DMA égale à 20 ns, le nouveau bus fonctionne avec une capacité d'adressage physique de 4 G et une fréquence de travail maximale de 50 MHz

7. Conclusion

Nous avons présenté un survol des circuits reprogrammables, prêtant une attention particulière aux CPLDs et FPGAs, qui sont les circuits reconfigurables les plus flexibles et offrant le plus de fonctionnalités. Ils sont les meilleurs candidats pour la réalisation rapide de prototype, de par leur capacité à être reconfiguré un très grand nombre de fois. Nous avons également vu que leur configuration est lancée par un agent externe, typiquement un petit contrôleur accédant à une mémoire qui contient les bits de configuration du CPLD ou du FPGA. De plus le placement et le routage des éléments sont calculés par un ordinateur, et ne peuvent être modifiés durant l'exécution d'une application.

Ensuite, nous avons présenté une implémentation schématique et VHDL de la solution de l'Adressage Physique Etendu et de l'Adressage Physique Rapide en utilisant le **ECS** de **Xilinx** et le **ISE** " Xilinx Integrated Software Environment ". Les deux nouveaux systèmes, que nous avons exposé, sont implantés dans des CPLD de la famille **XC9500** de **Xilinx** et sur la carte de développement **DS-BD-V2MB1000 (Virtex-II)**. L'un des principaux avantages des CPLDs sur les FPGAs est la rapidité. Le réseau d'interconnexions des CPLD est plus rapide que celui d'un FPGA. De plus, les connexions sont toujours réalisées avec une destination pour une source, le temps de propagation est donc toujours le même. Le placement d'un design dans un CPLD n'est donc pas critique, et le routage peut être systématiser, sans avoir besoin de tenir compte des contraintes de temps.

Dans le chapitre suivant nous présenterons l'utilisation des CPLD pour la réalisation des nouvelles architectures d'interfaçages graphiques proposées et aussi pour des réalisations qui fonctionnent sur des ordinateurs IBMTM et compatibles.

Chapitre 4

Nouvelles architectures de génération d'images

1. Introduction

La construction des images vidéo dans les systèmes à microprocesseur est assurée par les cartes vidéo ou les cartes graphiques. Une carte vidéo se compose de plusieurs unités de fonctions travaillant en collaboration. Le point de départ de la réalisation d'une image réside dans la RAM vidéo. Cette dernière est une mémoire RAM figurant sur la carte vidéo et contenant des informations sur l'image à afficher [73].

Pour construire une image vidéo sur la base de la RAM vidéo, les divers composants de la carte vidéo achèvent de lire la totalité de sa mémoire avec une fréquence de répétition d'image pouvant atteindre 75 fois par seconde (figure 4.1). Pendant la durée de cet accès de lecture, aucun accès simultané n'est autorisé. Ce qui donne la possibilité d'accès d'écriture seulement dans les périodes de retour ligne et trame. Dans ces conditions de fonctionnement classique, la réalisation d'un dispositif d'affichage, pour des systèmes de traitement d'images en temps réel et qui nécessite un temps d'enregistrement important reste très compliqué.

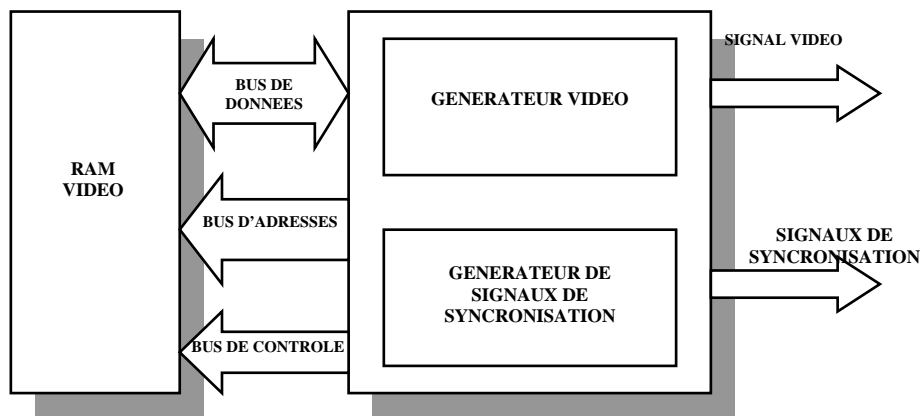


Fig. 4.1. Synoptique d'une carte vidéo

Dans ce chapitre nous présentons deux nouvelles techniques de production des séquences d'images vidéo dans un système à microprocesseur. La première technique de production d'image est basée sur une architecture mixte logicielle/matérielle. Elle utilise le système d'Adressage Physique Etendu comme interface [74] et avec un accès mémoire classique.

La deuxième solution est élaborée pour élargir la période d'accès réservée à l'enregistrement mémoire. Cette solution consiste à concevoir un système d'affichage qui permet un accès simultané de lecture/écriture. L'architecture proposée est constituée d'une partie matérielle et d'une partie logicielle. La partie matérielle est basée d'une part sur la décomposition de l'espace mémoire de la RAM vidéo en des parties avec des bus séparés et d'autre part sur l'association d'un circuit de sélection simultanée. La partie logicielle assure le transfert des données vers le système d'affichage et la synchronisation de l'enregistrement avec l'état de lecture de la RAM vidéo.

2. Construction d'image dans les cartes vidéo

L'image, qui représente les signaux de sortie des cartes vidéo, est le résultat d'une répétition de lecture de la RAM vidéo avec une conversion numérique analogique des données validées de cette dernière.

En VGA standard, l'écran contient 640 par 480 **pixels** qui doivent être balayés à une fréquence supérieure à 30 Hz. En général, la fréquence de rafraîchissement sur les PC est entre 60 Hz et 80 Hz (60 à 80 images par seconde). La trame VGA étant découpée en 525 lignes, une fréquence image de 60 Hertz entraîne une fréquence ligne de 31,5 kHz. Le balayage vertical se décompose en une synchro (2 lignes) suivi de 32 lignes noires suivi des 480 lignes de pixel suivies de 11 lignes noires (figure 4.2).

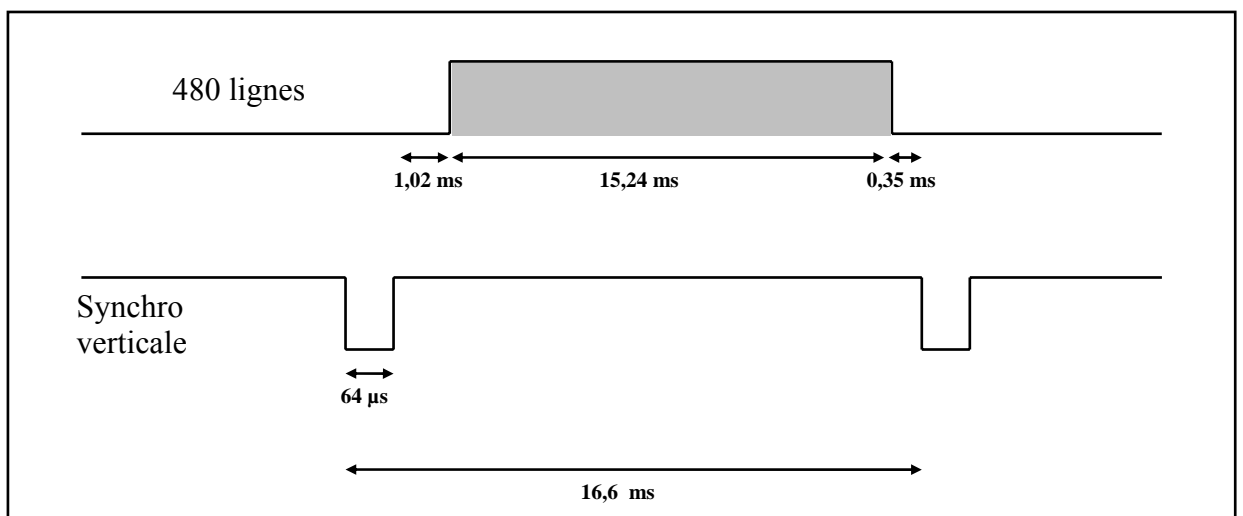


Fig. 4.2. Synchro verticale

Le balayage horizontal se décompose en un signal de synchronisation (95 points) un pallier avant noir (43 points), les 640 pixels et un pallier arrière (16 points) (figure 4.3).

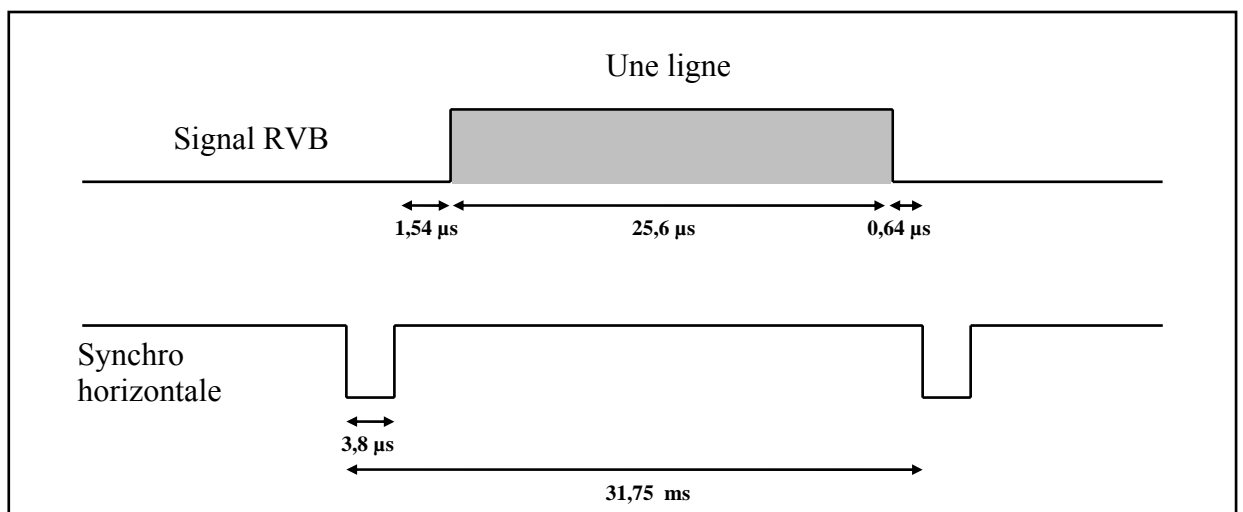


Fig. 4.3. Synchro horizontale

Le temps maximum possible pour l'écriture sur cette RAM, durant une trame, est donné par la relation suivante.

$$T_w = T_{fr} + T_{lr} \times N_l \tag{4.1}$$

T_{fr} : Le temps de retour trame.

T_{lr} : Le temps de retour ligne

N_l : Le nombre de lignes par trame

Le rapport du temps d'écriture sur le temps total de la trame vidéo varie généralement entre 10% à 15% et peut s'écrire sous la forme suivante.

$$W_r = \frac{(T_{fr} + T_{lr} \times N_l)}{T_f} \tag{4.2}$$

T_f : Le temps total de la trame.

3. Video RAM et Window RAM

Afin d'accroître le rapport du temps d'écriture sur le temps total de la trame vidéo, la mémoire vidéo VRAM (Video RAM) a été développée. La VRAM est une mémoire vive pour la vidéo, fonctionnant à double entrée (dual-ported), donc plus rapide que les autres types de mémoire. Ce type de mémoire admet une lecture écriture simultanée partielle. La figure 4.4 montre une architecture interne d'une VRAM [75].

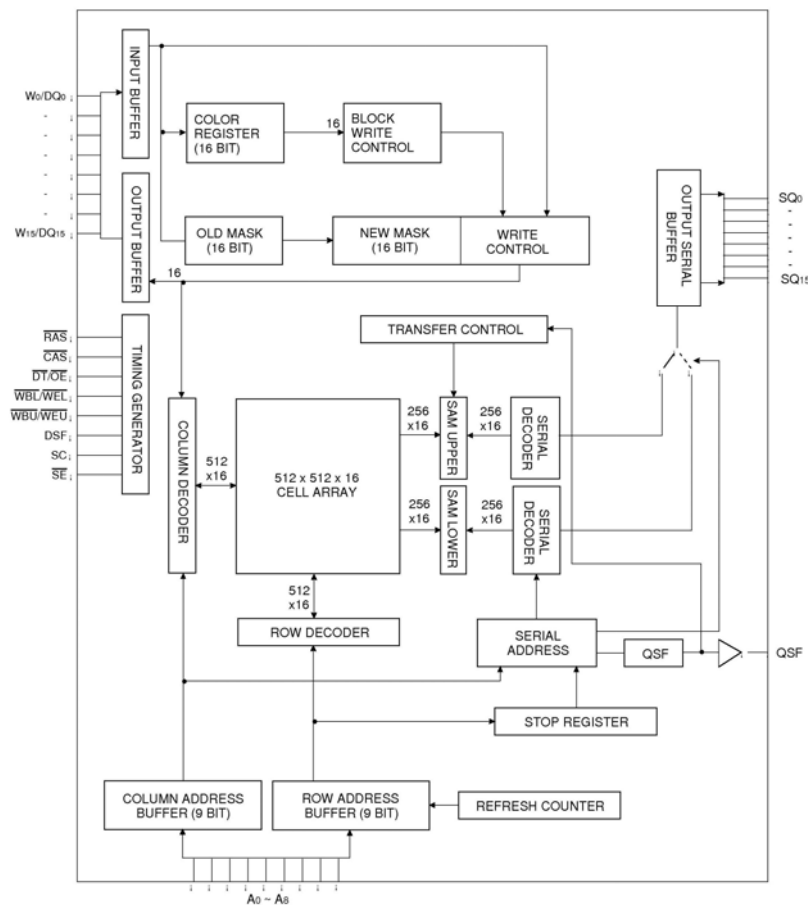


Fig. 4.4. Schéma bloc de KM4216C256

WRAM (Window Random Access Memory ou Window RAM) est une mémoire VRAM "dual-ported" de nouvelle génération disposant de fonctions graphiques spécialisées, conçue par Samsung. Cette VRAM est optimisée pour obtenir des taux de transfert maximum pour les opérations les plus courantes, comme le remplissage de formes simples. L'architecture de la WRAM (Window RAM) offre une amélioration jusqu'à 25% sur le taux global de transfert. Avec cette architecture le rapport Wr sera multiplié par un facteur de 1,25. La figure 4.5 présente une architecture interne d'une WRAM de Samsung [76].

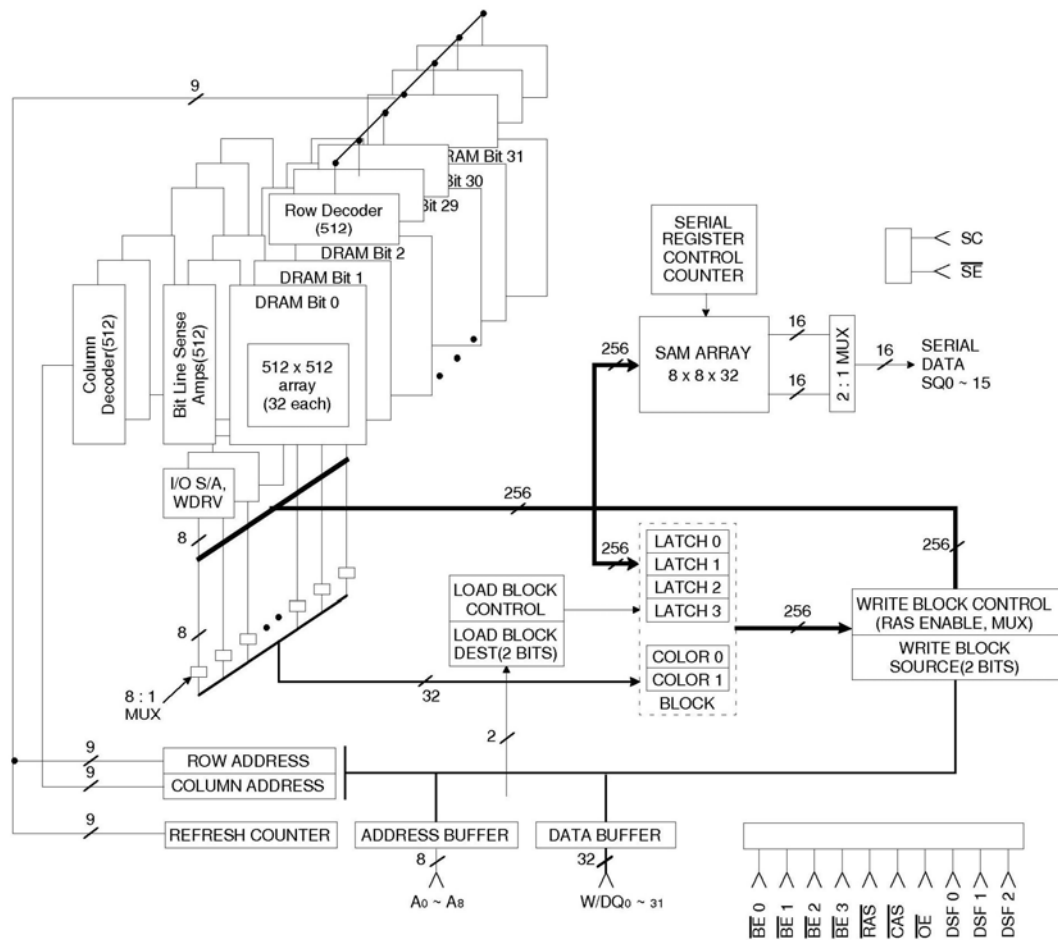
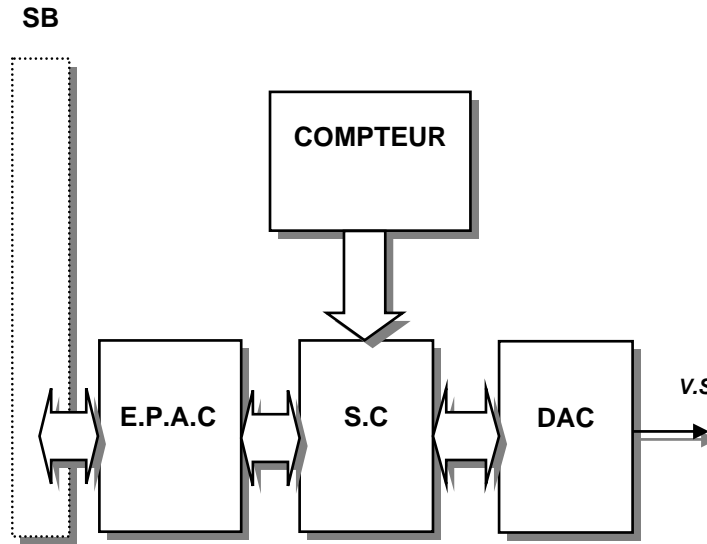


Fig. 4.5. Schéma bloc de KM4232W259A

4. Génération d'Images avec l'Adressage Physique Étendu

Dans cette partie nous présentons une nouvelle technique de production des séquences d'images vidéo dans un système à microprocesseur. Cette technique est basée sur une architecture mixte logicielle/matérielle et qui utilise le système d'Adressage Physique Étendu comme interface [74]. L'architecture proposée est constituée d'une partie matérielle et d'une partie logicielle, la première est composée principalement d'une interface à base d'Adressage Physique Étendu, d'une mémoire vidéo, d'un compteur d'adresses et d'un convertisseur numérique analogique (figure 4.6).



E.P.A.C: Circuit de l'Adressage Physique Etendu.
 S.B: Bus Système. V.S: Signal Vidéo.
 S.C: Circuit de Stockage.

Fig. 4.6. Synoptique de la production d'images avec l'Adressage Physique Etendu

La partie logicielle assure le transfert des données et le basculement entre les standards d'affichage. L'utilisation de l'Adressage Physique Etendu assure une écriture directe sur la mémoire vidéo et diminue la plage mémoire occupée dans le système à microprocesseur.

L'intégration d'un circuit d'Adressage Physique Etendu du premier ordre, dans ce système, donne un espace mémoire vidéo d'une capacité égale à 2^N où N est la taille des données utilisées sur le bus système. La taille maximale de notre image sera égale 2^N ce qui réclame une utilisation d'une mémoire vidéo avec un temps d'accès maximum égal à A_{Tmax} .

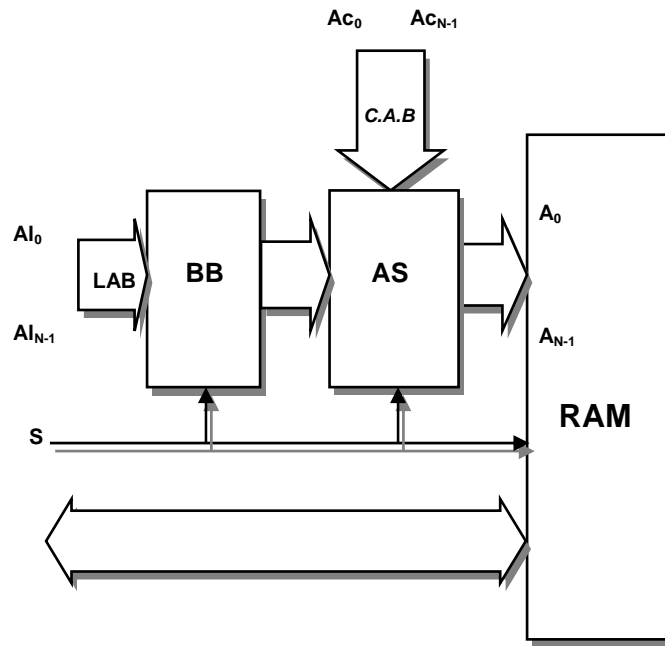
$$A_{Tmax} = \frac{1}{I \times 2^N} \tag{4.3}$$

Où I est le nombre d'image par seconde.

4.1. Unité de Stockage

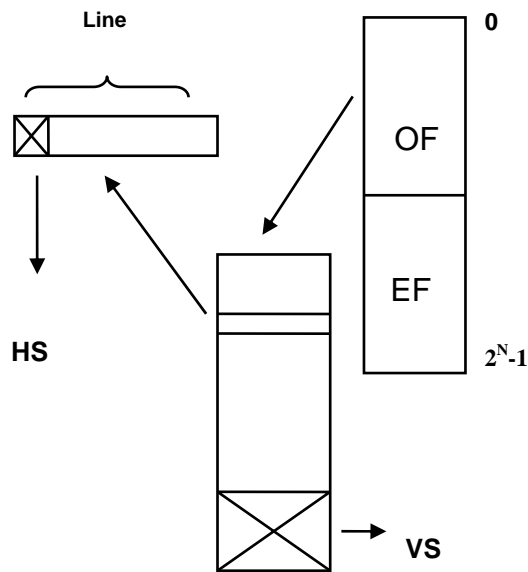
Le module de stockage sert, d'une part au stockage sur la RAM des données binaires qui représentent l'intensité numérique du signal vidéo. D'autre part, il assure la validation des données pour la conversion analogique à l'aide d'un convertisseur numérique analogique. La figure 4.7 présente les éléments essentiels qui constituent l'unité de stockage.

Le sélecteur d'adresse assure la commutation entre les adresses (Ali) et les adresses (Aci) en fonction du signal S qui indique la phase d'écriture. Si le signal S est actif, le sélecteur d'adresses commute sur les lignes Ai0-AiN-1, et la mémoire de notre système réalise un cycle d'écriture. Dans le cas contraire le sélecteur commute sur les lignes Ac0-AcN-1, et la mémoire est validée en lecture.



LAB: BUS D'ADRESSES LOCALES, BB: AMPLI DE BUS.
 CAB: BUS D'ADRESSES DU COMPTEUR
 AS: SELECTEUR D'ADRESSES.

Fig. 4.7. Unité de stockage.



VS: INFORMATION DE SYNCHRONISATION TRAME.
 HS: INFORMATION DE SYNCHRONISATION LIGNES.
 OF: TRAME IMPAIRE, EF: TRAME PAIRE.

Fig. 4.8. Structure de stockage du fichier image.

Le signal vidéo composite généré est le résultat d'une conversion numérique-analogique du fichier image stocké dans la RAM statique de notre carte. L'image numérique stockée dans la RAM est composée des deux trames paire et impaire, et chaque trame est composée d'une partie supérieure qui représente les lignes visibles de la trame et d'une partie inférieure non visible qui représente l'information de synchronisation trame. Les lignes visibles sont composées d'une information de synchronisation ligne et d'une information vidéo.

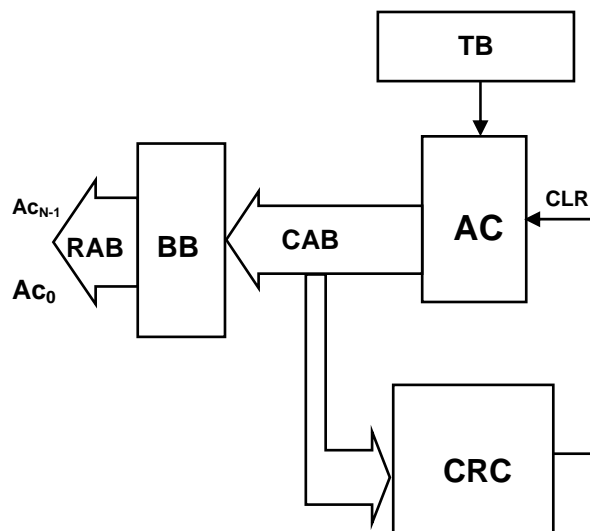
4.2. Unité de Lecture

Pour pouvoir adresser la mémoire d'image en lecture, nous avons besoin d'un générateur d'adresses (compteur). La taille des images que nous stockons est de 2^N points, ce qui nécessite N bits d'adresses. Il faut ajouter au circuit de comptage un circuit pour la remise à zéro du compteur à chaque arrivée à l'adresse 2^N (figure 4.9).

Le changement d'adresse de la valeur 0 à la valeur 2^N assure un balayage d'image de gauche vers la droite et de haut vers le bas.

L'unité de lecture est composée de trois parties essentielles :

- Un compteur 2^N bits
- Un décodeur d'adresse de fin lecture
- Une base de temps



CRC: CIRCUIT DE REMISE A ZERO DU COMPTEUR.
 AC: COMPTEUR D'ADRESSES, TB: BASE DE TEMPS.
 BB: AMPLI DE BUS, RAB: BUS D'ADRESSES DE LECTURES.
 CAB: BUS D'ADRESSES DU COMPTEUR

Fig. 4.9. Synoptique du circuit de lecture

Remarque : Le module de lecture est validé pendant la phase de lecture et ses sorties sont en haute impédance pendant la phase d'écriture.

La base de temps alimente l'entrée d'horloge du compteur d'adresses. Les lignes d'adresses, générées par le compteur, sont connectées au décodeur de fin de lecture, qui assure une remise à zéro du compteur à chaque arrivée à la fin de la zone mémoire réservée à l'image. Les lignes d'adresses sont amplifiées pour commander les lignes d'adresses de la RAM.

4.3. Emission d'image

Dès la validation du fichier image par le système, le programme teste l'existence et la forme de ce fichier, ensuite récupère la dimension et la position des valeurs des pixels qui forment l'image.

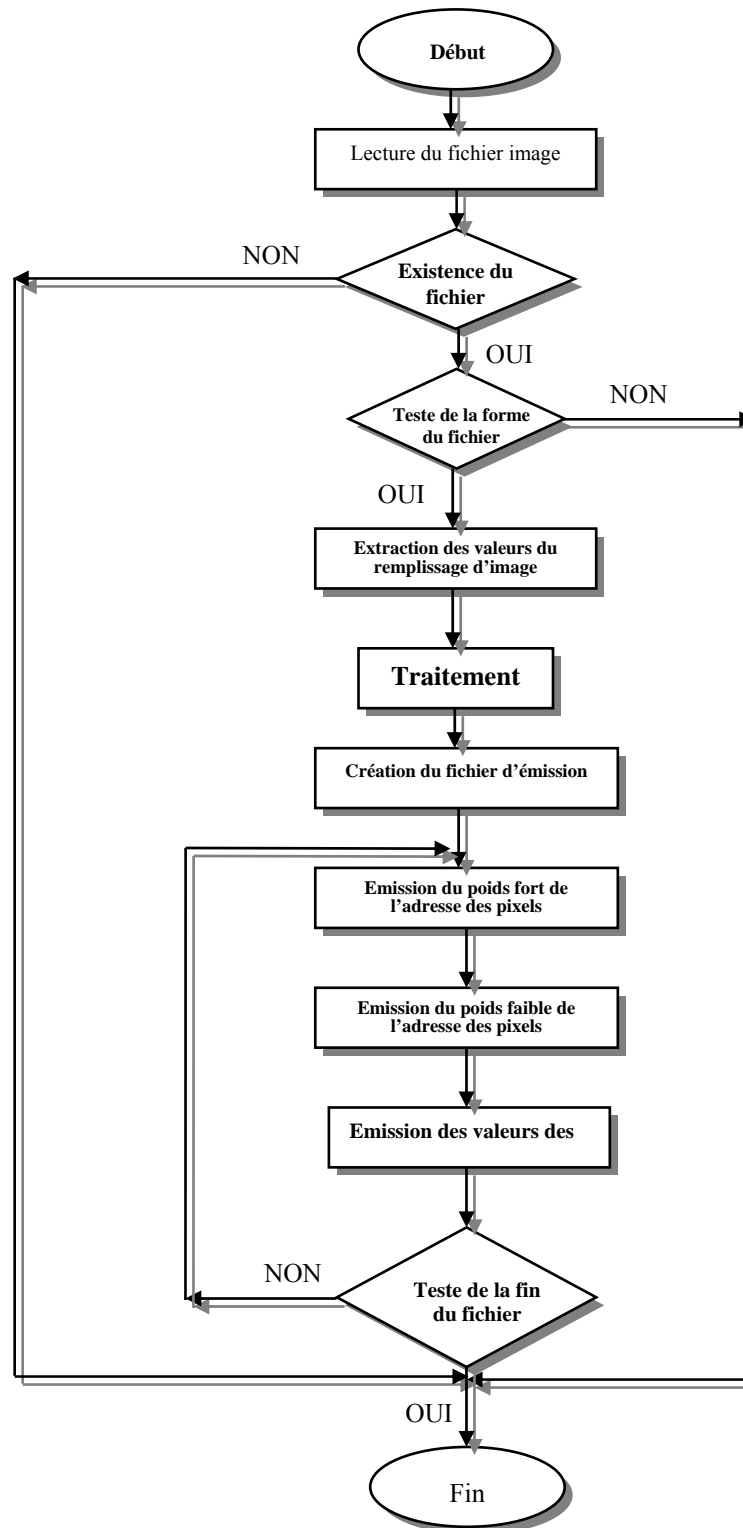


Fig. 4.10. Organigramme de l'émission des fichiers image

Les valeurs récupérées seront traitées afin de construire le fichier qui sera émis à notre carte. Avant l'émission d'un octet vers la carte, le logiciel envoie les deux octets du poids fort et du poids faible de l'adresse d'écriture sur la mémoire de notre carte. A la fin de l'envoi de tous les octets du fichier d'émission le programme affiche le message suivant « **Le travail est accompli avec succès** ».

4.4. Traitement numérique de base réalisé

4.4.1. Requantification

Avec une étude étalée des différences entre les images analogiques (signal vidéo composite) et les images numériques, nous pouvons déduire qu'une conversion analogique-numérique ne suffit pas pour produire des images TV à partir des données numériques stockées dans les fichiers image.

La première différence entre les deux images est que, l'une est représentée par un signal vidéo continu, entrelacé ou progressif (non entrelacé) et la deuxième est une pile de données numériques. Pour une image numérique monochrome codée sur 8 bits (256 niveaux de gris), la valeur « 0 » représente le niveau noir et la valeur 255 représente le niveau blanc. Pour les images analogiques l'information est divisée en deux parties, la partie supérieure représente le niveau de gris et occupe 70 % et la partie inférieure représente les informations de synchronisation et occupe les 30 %.

Pour cela le programme accomplit des procédures de traitement pour permettre au convertisseur A/D de la carte de générer un signal vidéo composite.

L'image numérique, $I = f(i, j)$, est écrite dans un tableau de « hauteur*largeur » cases, donnée par (4.4).

- $0 \leq i \leq (largeur - 1)$
- $0 \leq j \leq (hauteur - 1)$

$$I = \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,L-1) \\ f(1,0) & f(1,1) & & & f(1,L-1) \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ f(H-1,0) & \dots & & & f(H-1,L-1) \end{pmatrix} \quad (4.4)$$

La première opération de traitement est la multiplication de la matrice I par un facteur A pour avoir une matrice I' (4.5).

$$I = A \times \begin{pmatrix} f(0,0) & f(0,1) & \dots & \dots & f(0,L-1) \\ f(1,0) & f(1,1) & & & f(1,L-1) \\ \dots & \dots & & & \dots \\ \dots & \dots & & & \dots \\ f(H-1,0) & \dots & & & f(H-1,L-1) \end{pmatrix} \quad (4.5)$$

- $A = \frac{7}{10}$

La deuxième opération de traitement est l'addition de la matrice I' avec la matrice B pour avoir une matrice I'' (4.6).

$$I'' = I + \begin{pmatrix} b(0,0) & \dots & \dots & \dots & b(0,L-1) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ b(H-1,0) & \dots & \dots & \dots & b(H-1,L-1) \end{pmatrix} \quad (4.6)$$

- $b(i, j) = 77$ (représente les 30 %)
- $i \in [0, \text{hauteur}]$
- $j \in [0, \text{largeur}]$
- $L = \text{largeur}$
- $H = \text{hauteur}$

La troisième opération de traitement est l'addition des impulsions lignes et des impulsions trames. La figure 4.11 représente l'organigramme du traitement d'image effectué par le programme pour un affichage entrelacé.

La figure 4.12 représente l'organigramme du traitement d'image effectué par le programme pour un affichage progressif.

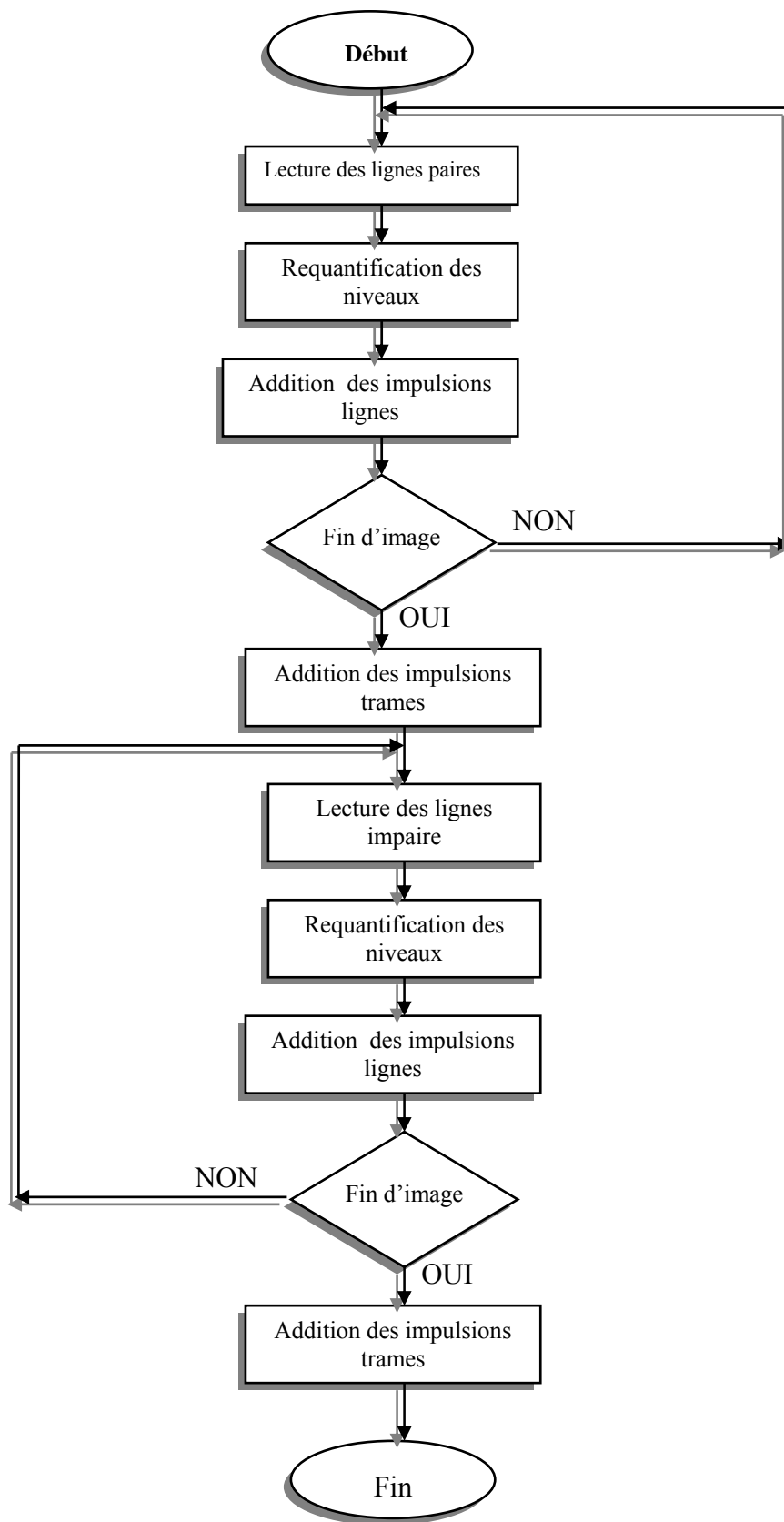


Fig. 4.11. Organigramme du traitement pour un affichage entrelacé

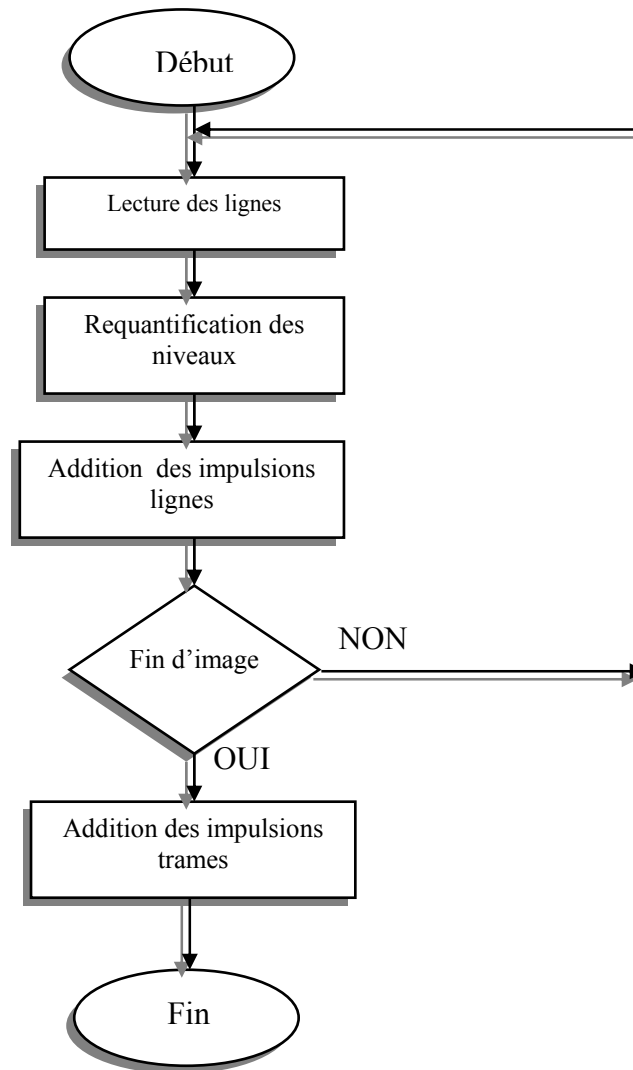


Fig. 4.12. Organigramme du traitement pour un affichage progressif

4.4.2. Changement de la définition horizontale

Dans les cas de nécessité d'avoir une bande passante limitée du signal vidéo à cause du temps de réponse des RAM utilisées ou dans l'obligation d'avoir des faibles définitions analogiques, un changement de définition horizontale numérique (information de synchronisation y compris) est exigé. La figure 4.13 représente le principe de ce traitement qui comprend un filtrage passe bas et un sous échantillonnage.

Pour générer un signal vidéo de bande passante élevée, il faut utiliser des RAM et des convertisseurs plus rapides.

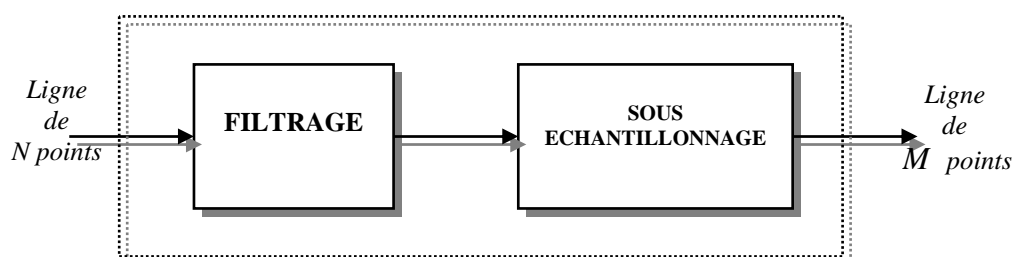


Fig. 4.13. Principe du changement de la définition horizontale

4.4.3. Addition des impulsions de synchronisation

Dans les images analogiques une ligne est précédée par une impulsion de synchronisation qui représente 15 % à 20 % de la période de la ligne, ce qui impose l'addition des données représentatives des impulsions lignes aux fichiers images.

Si l_N et $l_{N'}$ (4.7 et 4.8) représentent respectivement la ligne dans le fichier image de départ et après un changement de définition, l_A représente la ligne d'émission pour la ligne analogique (4.9).

$$l_N = \begin{pmatrix} d_0 \\ d_1 \\ \dots \\ \dots \\ \dots \\ d_{84} \\ d_{85} \end{pmatrix} \quad (4.7)$$

$$l_{N'} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ \dots \\ \dot{0} \\ d_0 \\ d_1 \\ \dots \\ \dots \\ d_{83} \\ d_{84} \end{pmatrix} \quad (4.8)$$

$$l_A = l_{N'} + Tl \quad (4.9)$$

$$Tl = \begin{pmatrix} Top_0 \\ Top_1 \\ \dots \\ \dots \\ \dots \\ Top_{18} \\ 0 \\ 0 \\ 0 \\ \dots \\ \dots \\ 0 \end{pmatrix} \quad (4.10)$$

Avec :

Top_i représente la valeur de l'impulsion ligne

Le même principe est appliqué pour les impulsions trames, mais dans ce cas l'addition des impulsions trames est réalisée sur des images complètes.

5. Génération d'Images avec Lecture/Ecriture Simultanée

5.2. Construction d'image avec accès simultané à deux niveaux

5.2.1. Partie Matérielle

Le système de construction d'image avec accès simultané à deux niveaux représente l'architecture de base de notre système [77]. Cette architecture de visualisation utilise une RAM vidéo composé de deux RAM. Les deux RAM peuvent être sélectionnées séparément, une peut être sélectionnée en lecture et la deuxième en écriture et inversement. La figure 4.14 représente le schéma bloc de la partie matérielle de ce système.

La partie matérielle de notre système est constitué de quatre unités, unité d'interface, unité de sélection, unité de lecture et unité de conversion numérique analogique. L'unité d'interface peut utiliser deux types d'adressage, l'adressage physique étendu ou l'adressage physique rapide [78]. L'unité de sélection est connectée à toutes les unités et à la RAM vidéo. La connexion avec l'interface est constituée d'un bus d'état, d'un bus de contrôle, d'un bus d'adresses et d'un bus de données. La connexion avec l'unité de lecture est composée d'un bus de contrôle et d'un bus d'adresses. La connexion avec l'unité de conversion est assurée par un bus de données unidirectionnel qui transfère les données de l'unité de sélection vers l'unité de conversion.

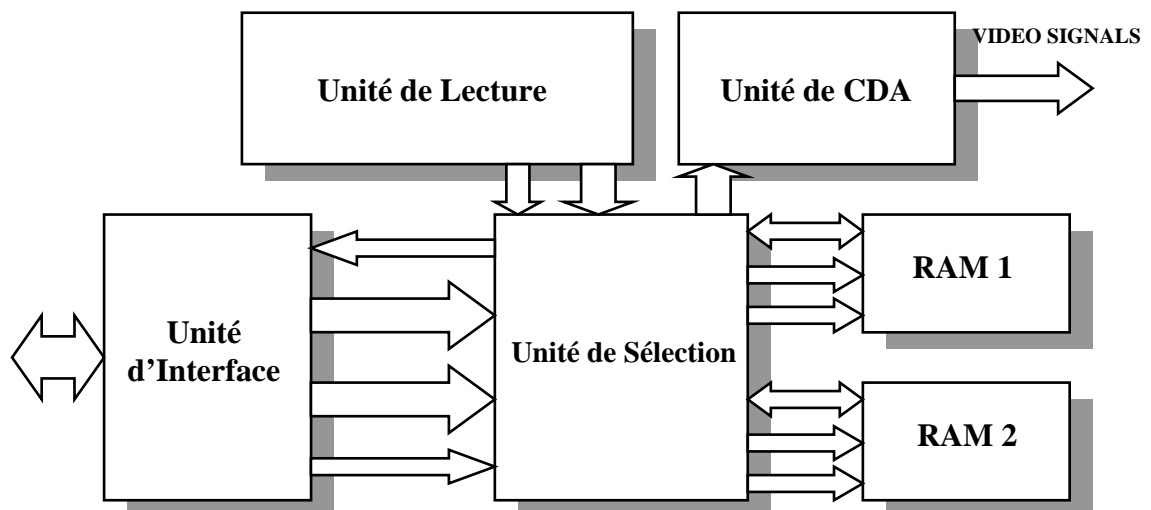


Fig. 4.14. Schéma bloc du Système avec Accès Simultané à Deux Niveaux

Dans cette architecture d'accès simultané, le rapport temps d'écriture sur le temps total de la trame vidéo varie de 100% à une valeur minimale W_{r2} . Le rapport W_{r2} est donné par la relation suivante.

$$W_{r2} = \frac{1}{2} + \frac{W_r}{2} \tag{4.11}$$

En utilisant la relation (4.2), l'expression de W_{r2} s'écrit sous la forme suivante :

$$W_{r2} = \frac{1}{2} + \frac{(T_{fr} + T_{lr} \times N_l)}{2T_f} \tag{4.12}$$

En utilisant les valeurs de W_r de la section 2, le rapport W_{r2} varie entre 55% et 57,5%

5.2.2. Partie Logicielle

La partie logicielle est conçue pour réaliser un processus d'enregistrement permanent [79] sur la RAM vidéo de cette architecture [80].

Dans le cas d'un accès simultané à deux niveaux, le fichier image est décomposé en deux parties, f1 et f2. Le fichier f1 représente la partie supérieure de l'image et le fichier f2 représente la partie inférieure. Le logiciel permet le basculement entre l'envoi des données du fichier f1 et les données du fichier f2. La synchronisation de l'envoi des données est liée à l'état de sélection des deux RAM de notre système. La figure 4.15 illustre l'organigramme principal de la partie logicielle.

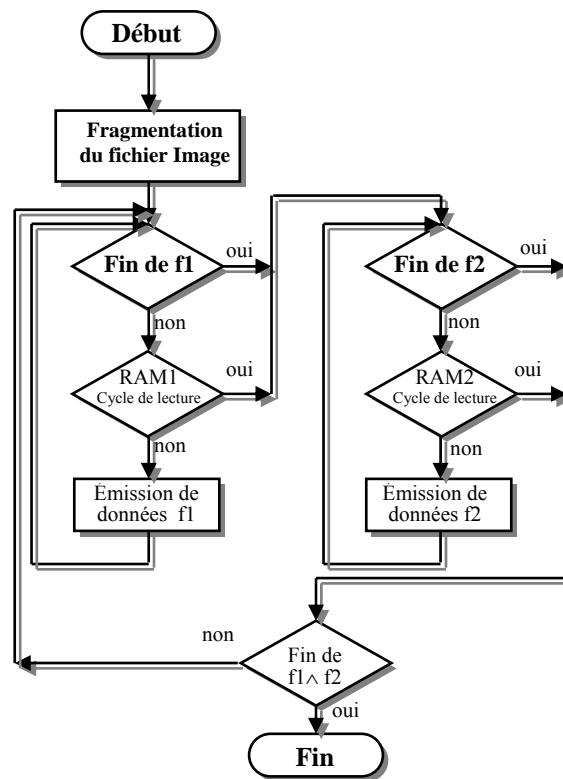


Fig. 4.15. Organigramme principal de la partie logicielle

5.3. Construction d'image avec accès simultané à plusieurs niveaux.

Le principe de la construction d'image avec accès simultané à plusieurs niveaux est similaire au principe de construction à deux niveaux. Dans cette structure de visualisation, notre système utilise une RAM vidéo composée de plusieurs RAM.

5.3.1. Partie Matérielle

La figure 4.16 représente le schéma block de la partie matérielle de ce système. Cette dernière est constituée de quatre unités, unité d'interface, unité de sélection, unité de lecture et unité de conversion numérique analogique.

La connexion entre les unités utilise la même structure que le système à deux niveaux. La largeur du bus d'état est dépendante du nombre de RAM utilisées. La communication entre l'unité de sélection et les RAM utilisées est assurée par plusieurs bus, chaque bus est composé d'un bus de données, d'un bus d'adresses et d'un bus de contrôle.

Cette architecture donne un rapport du temps d'écriture sur le temps total de la trame vidéo, variant de 100% à une valeur minimale W_{rN} . Le rapport W_{rN} est donné sous la forme suivante.

$$W_{rN} = \left(1 - \frac{1}{N}\right) + \frac{W_r}{N} \quad (4.13)$$

N : Représente le nombre de RAM utilisées dans notre système.

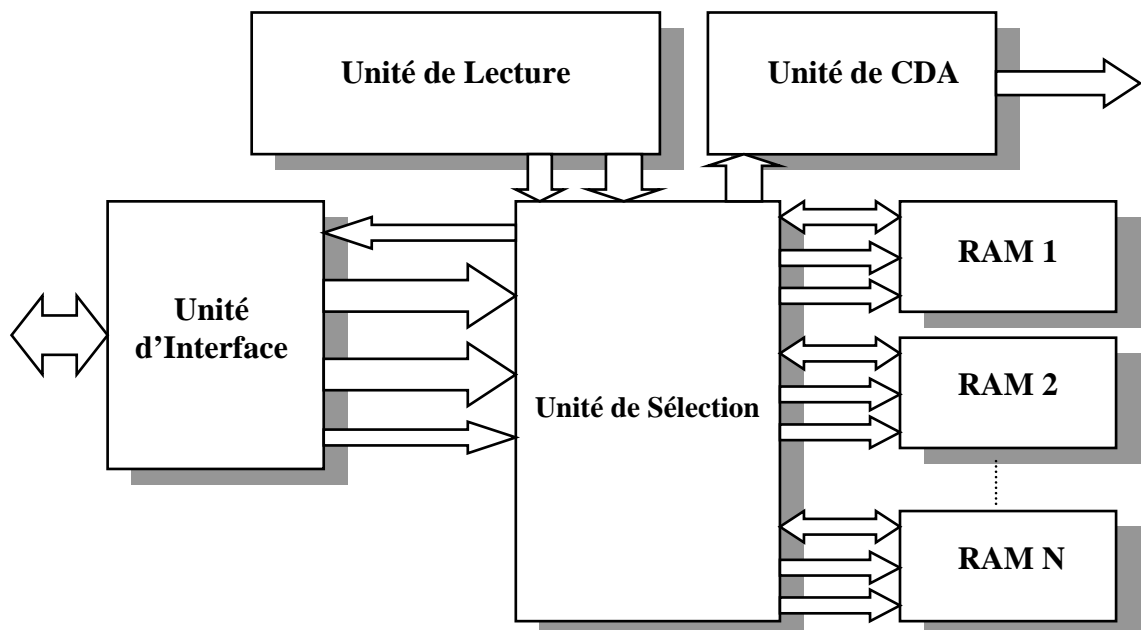


Fig. 4.16. Schéma bloc du Système avec Accès Simultané à N Niveau

En utilisant la relation (4.2), l'expression de W_{rN} s'écrit sous la forme suivante.

$$W_{rN} = \left(1 - \frac{1}{N}\right) + \frac{(T_{fr} + T_{lr} \times N_l)}{T_f \times N} \tag{4.14}$$

En utilisant la formule (4.14) et une valeur de W_r égale à 10%, W_{rN} peut prendre plusieurs valeurs en fonction de N . Le tableau 4.1 et la figure 4.17 présentent les valeurs W_{rN} pour N variant de 1 à 100.

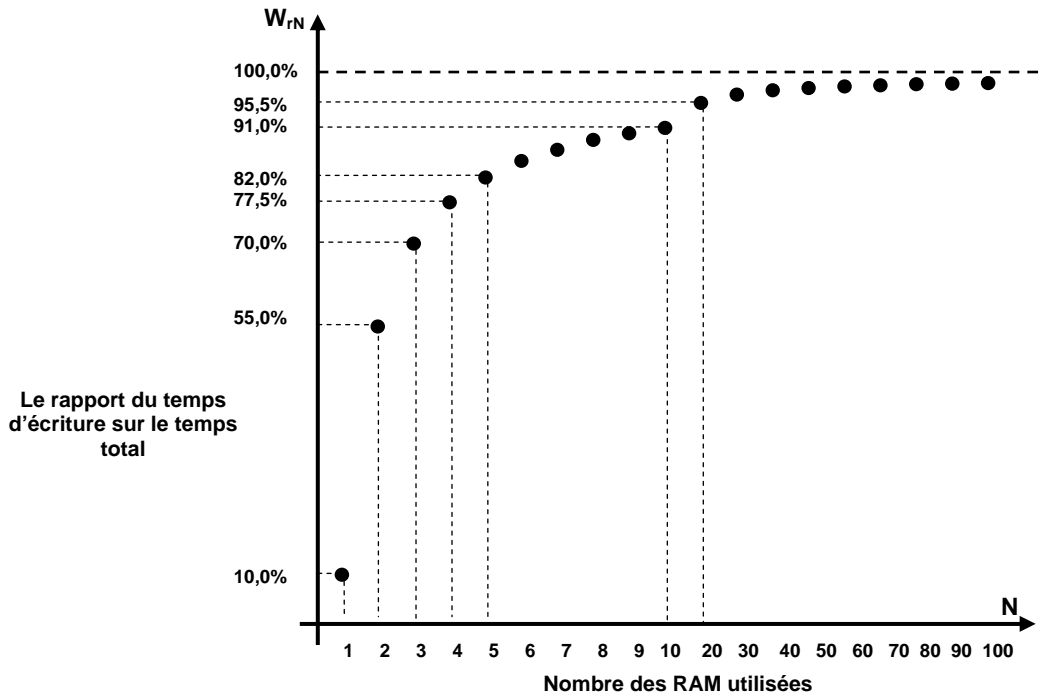


Fig. 4.17. Variation de W_{rN} avec le nombre des RAM

TABEAU 4.1

Les valeurs de W_{rN}

N	1	2	3	4	5	6	7	8	9
W_{rN}	10%	55%	70%	77,5%	82%	85%	87,1%	88,75%	90%
N	20	30	40	50	60	70	80	90	100
W_{rN}	95,5%	97%	97,75%	98,2%	98,5%	98,7%	98,87%	99%	99,1%

5.3.2. Partie Logicielle

La partie logicielle de l'accès simultané à plusieurs niveaux décompose le fichier image en plusieurs parties. Le résultat donne N fichiers ($f_1, f_2, f_3, \dots, f_N$). Le basculement de l'envoi des données ce fait entre les N fichiers. La synchronisation de l'envoi des données est liée à l'état de sélection des N RAM qui constituent la RAM vidéo de notre système. La figure 4.18 présente l'organigramme principal de la partie logicielle du système à plusieurs niveaux.

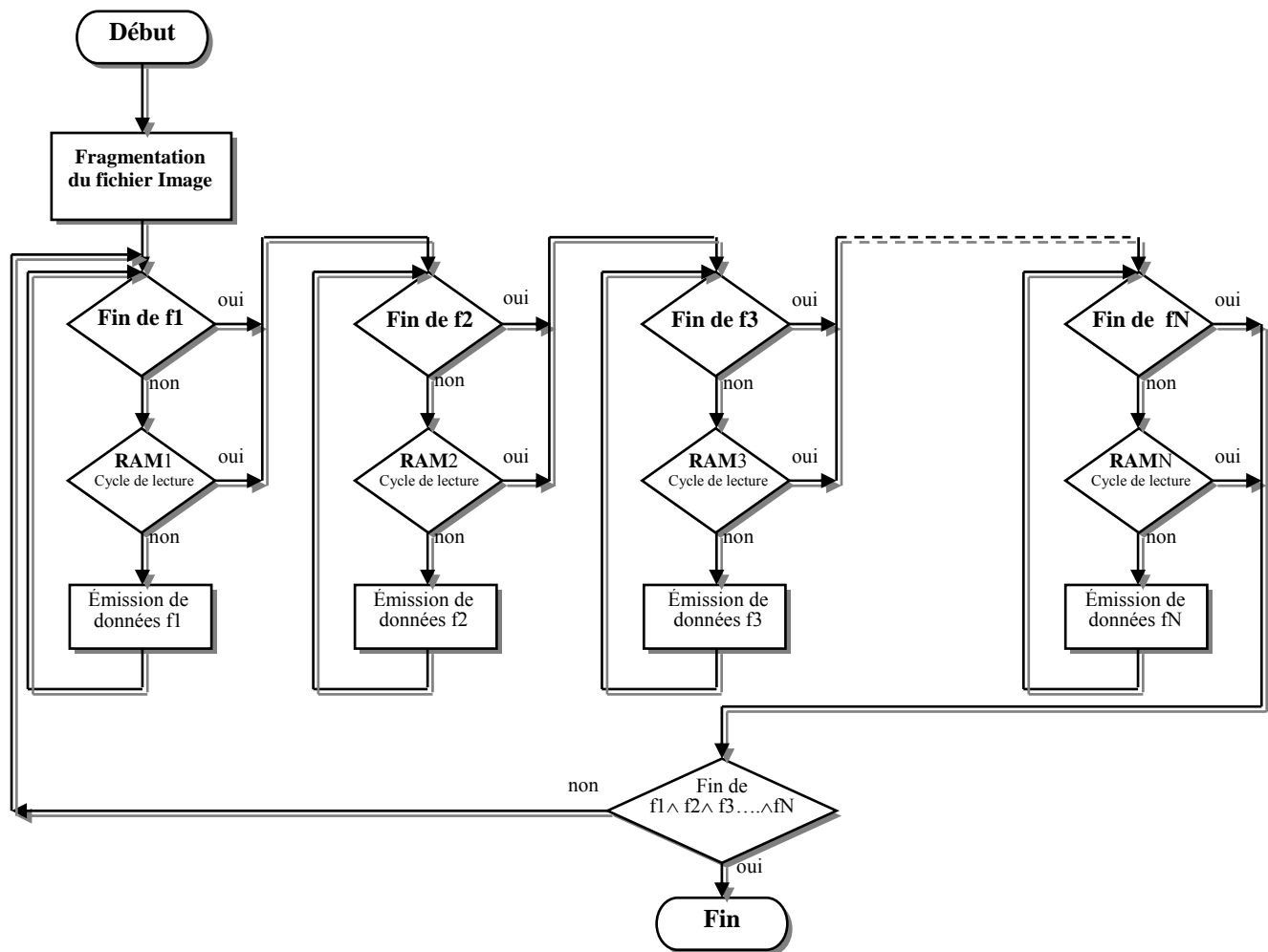


Fig. 4.18. Organigramme principal de la partie logicielle du système à plusieurs niveaux.

6. Stockage et Conversion

6.1. Conversion numérique analogique

La conversion numérique analogique est effectuée par un convertisseur classique DAC à 8 bits, le TDA8702 de Philips, recommandé pour les applications vidéo [81]. Le convertisseur offre un temps de conversion inférieur à 8 ns et un temps de propagation des bits maximum de 1 ns. Le TDA8702 possède deux sorties complémentaires avec une vitesse de conversion de 30 MHz.

Le schéma synoptique du convertisseur numérique-analogique (DAC) TDA 8702 de Philips est représenté par la figure 4.19. 100 nF est la valeur recommandée pour le découplage de la broche 1.

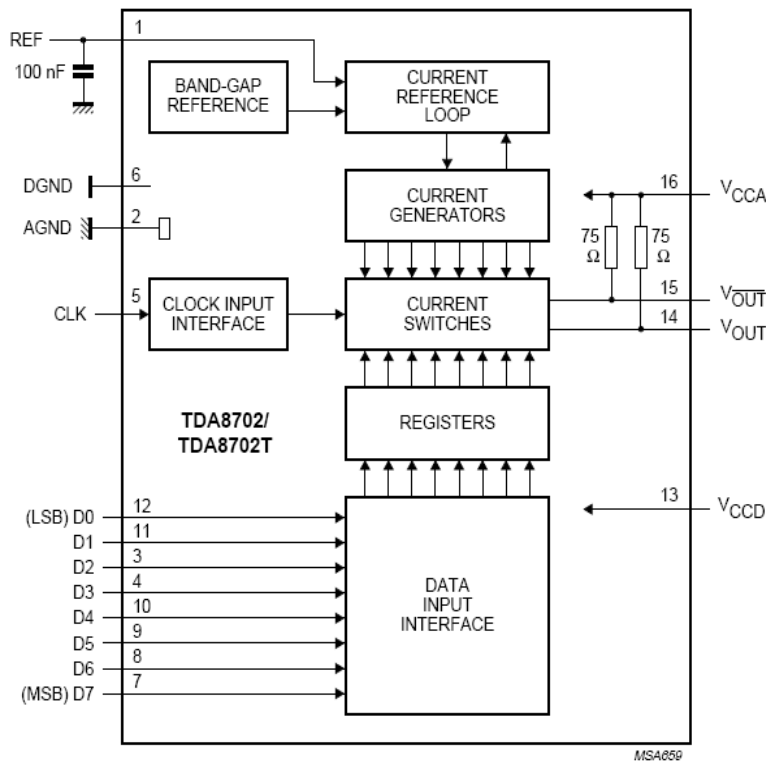


Fig. 4.19. Schéma synoptique du TDA8702

Les caractéristiques du TDA8702 sont :

- Vitesse de conversion maximale : 30 MHz.
- Entrées compatibles TTL.
- Générateur de tension de référence interne.
- Deux sorties analogiques complémentaires.
- Pas de nécessité d'horloge en entrée.
- Charge de sortie interne de 75Ω (connectée à la l'alimentation analogique).
- Faible dissipation.
- Alimentation analogique : $V_{CCA} = 5\text{ V}$.
- Alimentation numérique : $V_{CCD} = 5\text{ V}$.

Les pins du convertisseur :

REF : tension de référence (découplage)

AGND : masse analogique

D2: data input; bit 2

D3: data input; bit 3

CLK: horloge d'entrée

DGND : masse numérique

D7: data input; bit 7

D6: data input; bit 6

D5: data input; bit 5

D4: data input; bit 4

D1: data input; bit 1
D0: data input; bit 0
VCCD : alimentation numérique pour les circuits numériques (+5 V)
V_{OUT} : tension de sortie analogique
V_{OUT} : tension de sortie analogique complémentaire
VCCA : alimentation analogique pour les circuits analogiques (+5 V)

6.2. Stockage et mémoire vidéo.

Cette partie de notre système est réalisée avec des circuits de mémoires SRAM (RAM statique) de 1M bits. Nous avons utilisé deux versions de SRAM, μ PD431000 de NEC [81] et IS61C1024 de ISSI [83] (Integrated Silicon Solution, Inc). La figure 4.20 présente le schéma block du μ PD431000.

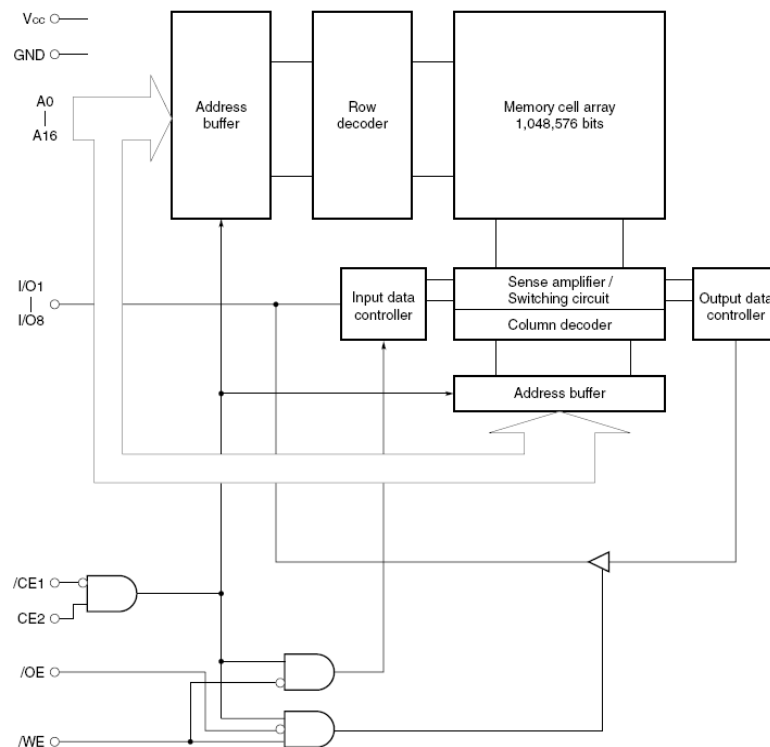


Fig. 4.20. Schéma synoptique du μ PD431000

Les pins de la SRAM :

A0~A16 : Entrées des adresses
WE : Validation d'écriture
CE : Validation du circuit
OE : Validation des sorties
D0~D7 : Entrées-sorties des données
Vcc : Alim. +5 V
GND : Masse

Pour pouvoir écrire sur les RAM statiques il faut que les signaux de contrôles remplissent les conditions d'écritures du constructeur de la RAM statique. Les figures 4.21 et 4.22 présentent des exemples de chronogrammes de lectures et d'écritures proposés par NEC. La logique de **CE** gère le processus de sélection du chip dans sa totalité, **OE** gère les données au niveau des amplis de sortie et **WE** gère la lecture ou l'écriture.

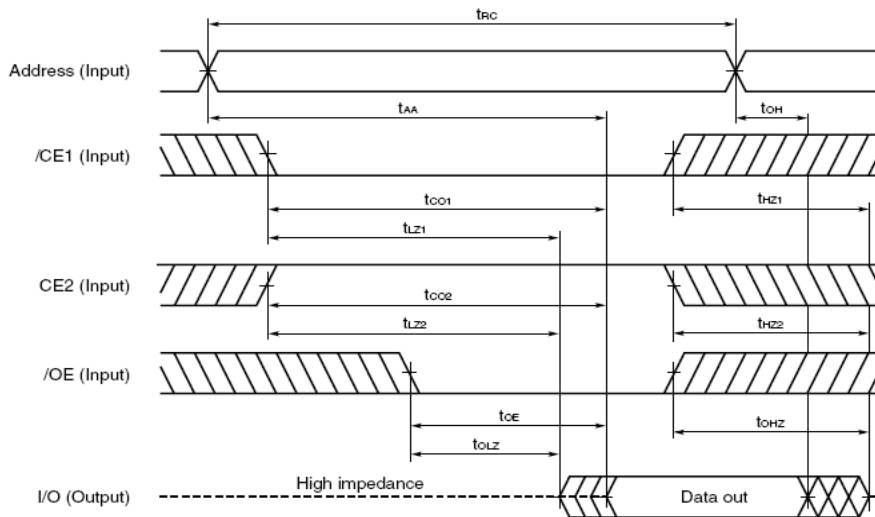


Fig. 4.21. Cycle de lecture

En lecture, le temps d'accès par rapport à CE donne une idée de la vitesse d'accès. Le temps de cycle est légèrement plus long car il doit tenir compte de la désactivation de l'adresse courante en préparation d'un prochain accès. En écriture, après avoir activé CE et amené les données valides sur le bus de données de la RAM statique, l'écriture en mémoire se produit sur le front montant de WE.

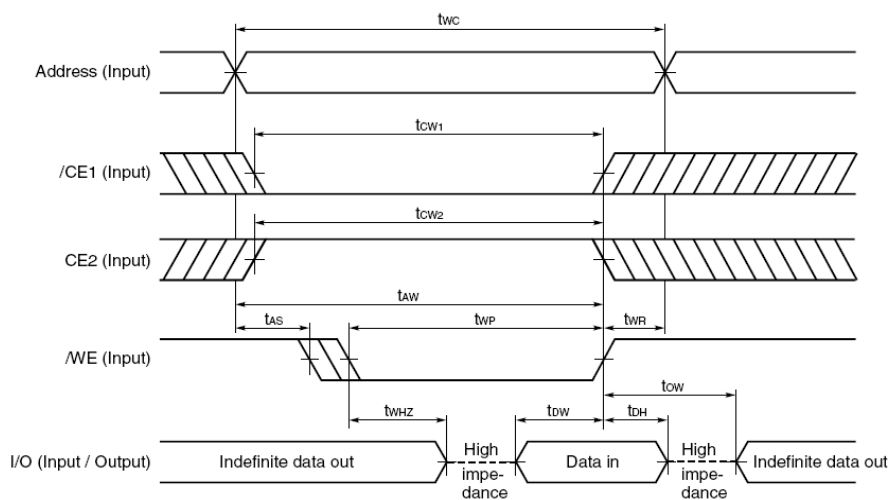


Fig. 4.22. Cycle d'écriture

La figure 4.23 présente le schéma block du IS61C1024. La SRAM de ISSI propose les mêmes broches de NEC.

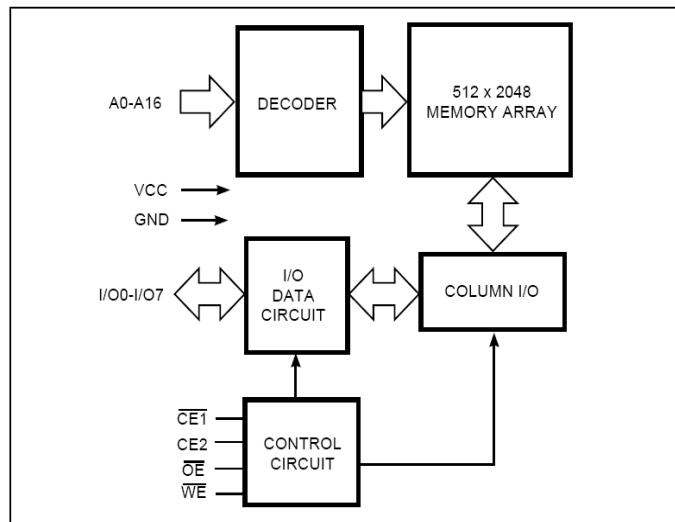


Fig. 4.23. Schéma synoptique du IS61C1024

Les figures 4.24 et 4.25 présentent des exemples de chronogrammes de lecture et d'écriture proposés par ISSI (Integrated Silicon Solution, Inc).

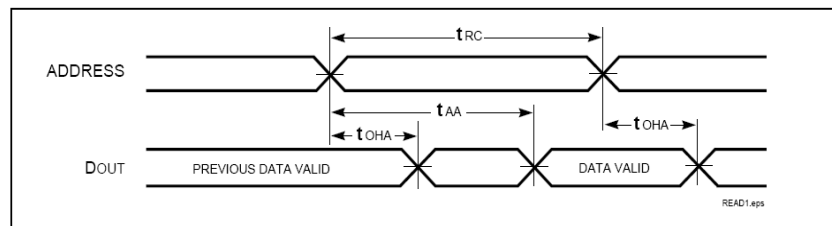


Fig. 4.24. Cycle de lecture

Le chronogramme de lecture de la figure 4.24 est applicable pour les SRAM de NEC. Il faut mettre en évidence l'existence d'un temps d'accès t_{AA} qui peut être mesuré par rapport aux divers signaux de référence. Les conditions de la figure 4.25 sont applicables pour un grand nombre de RAM statiques.

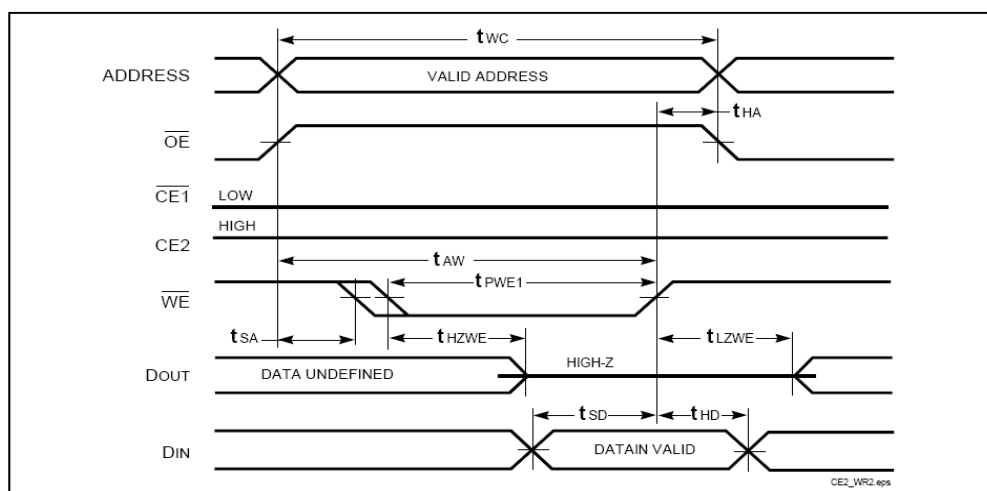


Fig. 4.25. Cycle d'écriture

7. Conclusion

Les architectures que nous avons proposées dans ce dernier chapitre constituent une solution pour l'utilisation optimale des composants de la logique programmable qui répondent bien aux contraintes du traitement vidéo temps réel. L'idée majeure est d'introduire les nouveaux systèmes d'interfaçages physiques et aussi d'implanter le principe de l'accès simultané pour la gestion de la mémoire vidéo.

Nous avons exposé une nouvelle architecture basée sur l'utilisation de l'Adressage Physique Étendu. Cette architecture occupe une plage mémoire réduite et offre la possibilité de générer des images de haute résolution. Ensuite, nous avons présenté le principe de base de l'accès simultané avec une structure à deux niveaux puis avec une structure à plusieurs niveaux.

Les systèmes proposés présentent l'avantage de pouvoir accepter des mises à jours matérielles. Une nouvelle description VHDL ou schématique permettra d'implanter de nouvelles de circuit. Le concepteur doit tenir compte de la capacité du composant programmable ainsi que de la fréquence maximale de fonctionnement.

Les premiers tests pratiques ont donné un taux d'enregistrement de 95% pour le système à deux niveaux et un taux de 98% pour le système à trois niveaux. Les structures de niveau supérieur à trois donnent des taux inférieurs. Cette dégradation du taux d'enregistrement à des niveaux supérieur à trois est due à la complexité de la circuiterie et au temps de traitement logiciel. Il est donc inutile de recourir à des structures de niveau supérieur à trois. Les structures optimales sont obtenues pour les niveaux deus et trois. Les résultats obtenus permettent en premier lieu d'envisager des structures similaires avec des circuits plus performants permettant d'aborder des fréquences plus élevées [84,85].

Conclusion

Nous nous sommes intéressés dans ce travail à la conception de nouvelles architectures d'interfaçage graphiques dans les systèmes à microprocesseur avec une utilisation optimale des composants de la logique programmable sous les contraintes du traitement vidéo temps réel. Les architectures présentées sont basées sur l'implantation des nouveaux systèmes d'interfaçages physiques proposés mais aussi sur l'application de notre procédé d'accès simultané pour la gestion de la mémoire vidéo. Les quatre nouveaux systèmes d'interfaçages physiques sont l'Adressage Physique Étendu, l'Adressage Physique Accéléré, l'Adressage Physique Rapide et l'Adressage Physique Avancé.

Au cours de nos travaux, nous avons abordé le problème du débit pratique dans les interfaces et particulièrement dans les interfaces graphiques. L'écart entre le gain théorique et le gain pratique est très important (parfois supérieur à 95%). Cette situation impose une nouvelle réflexion architecturale pour l'interfaçage graphique temps réel afin de prendre en considération l'augmentation de la bande passante pratique avec la réduction des temps d'attentes.

L'état de l'art des interfaces d'entrées-sorties et le fonctionnement général des bus actuels, commençant avec le bus ISA d'IBM et allant au bus AGP et PCI express, que nous avons présenté dans le chapitre 1, montre la nécessité d'introduire de nouveaux systèmes d'interfaçages. L'exécution séquentielle, l'allocation rigide des plages d'entrées-sorties et les temps d'attentes dans les transactions entre les contrôleurs d'interfaces et les bus d'extensions sont les principales sources de ralentissement pour l'échange de données et donc pour le débit pratique. Nous avons donc analysé ces sources de ralentissement et leurs architectures associées afin de proposer nos nouveaux systèmes.

Les quatre nouveaux systèmes sont exposés dans le chapitre trois. Nous avons exposé le système d'Adressage Physique Étendu d'ordre un qui représente le système que nous avons proposé dans sa version de base. En suite nous avons exposé l'Adressage Physique Accéléré, l'Adressage Physique Rapide et l'Adressage Physique Avancé. Ces nouveaux systèmes présentent des remèdes à l'inconvénient de la basse fréquence des débits pratiques et à la capacité d'adressage physique.

Dans la troisième partie, nous avons présenté un survol des circuits reprogrammables où nous avons proposé une implémentation schématique et VHDL de la solution de l'Adressage Physique Etendu et de la variante de l'Adressage Physique Rapide en utilisant le ECS de Xilinx et le ISE "Xilinx Integrated Software Environment".

Une nouvelle architecture d'interfaçage graphique, basée sur l'utilisation conjointe de l'Adressage Physique Etendu et des mémoires vidéo classique, est exposée dans la première partie du dernier chapitre. Nous avons ensuite exposé le principe de base de notre architecture d'accès mémoire simultané avec la structure à deux niveaux et la structure à plusieurs niveaux. Les interfaces des deux architectures occupent des plages d'entrées-sorties réduites et offrent la possibilité de générer des images de haute résolution. Notre deuxième architecture basée sur l'accès simultané est particulièrement adaptée pour assurer l'affichage dans un système de traitement vidéo temps réel. Cette deuxième architecture offre des mécanismes d'accélération non honorés par les architectures graphiques actuelles.

Les premiers tests pratiques ont donné un taux d'enregistrement de 95% pour le système à deux niveaux et un taux de 98% pour le système à trois niveaux. Les structures de niveau supérieur à trois donnent des taux inférieurs. Cette dégradation du taux d'enregistrement à des niveaux supérieur à trois est due à la complexité de la circuiterie et au temps de traitement logiciel. Il est donc inutile de recourir à des structures de niveau supérieur à trois. Les structures optimales sont obtenues pour les niveaux deux et trois. Les systèmes proposés présentent l'avantage de pouvoir accepter des mises à jours matérielles. Une nouvelle description VHDL ou schématique permettra d'implanter de nouvelles de circuit. Le concepteur doit tenir compte de la capacité du composant programmable ainsi que de la fréquence maximale de fonctionnement.

Les perspectives à l'issue de cette thèse sont nombreuses, nous pouvons en citer à titre d'exemple l'implantation des structures développées dans ce travail ou des structures similaires sur c des circuits plus performants. Ceci permettra d'aborder des fréquences plus élevées.

Nous pouvons également citer comme perspective le développement de variantes grand public par l'addition d'un pont "South Bridge". Le pont, dont la circuiterie dépendra de la variante du bus PCI utilisé, assurera l'interconnexion entre ce dernier et l'interface de notre système.

Bibliographie

- [1] PLUG AND PLAY ISA SPECIFICATION, Intel Corporation and Microsoft Corporation, USA, 1994.
- [2] Tom Shanley and Don Anderson, EISA SYSTEM ARCHITECTURE, Addison Wesley Publishing Company, USA, 1995
- [3] PCI LOCAL BUS SPECIFICATION, REVISION 2.1, PCI Special Interest Group, June 1, USA, 1995.
- [4] PCI LOCAL BUS SPECIFICATION, REVISION 2.2, PCI Special Interest Group, December 18, USA, 1998.
- [5] ACCELERATED GRAPHICS PORT INTERFACE SPECIFICATION, REVISION 2.0, Intel Corporation, USA, 1998
- [6] AGP3.0 INTERFACE SPECIFICATION, REVISION 1.0, Intel Corporation, USA, 2002
- [7] N. Ishibashi, DIGITAL IMAGE TRANSFER CONTROLLER, United States Patent US006948022B2, 20 Septembre 2005.
- [8] Mostafa Abd-El-Barr and Hesham El-Rewini, FUNDAMENTALS OF COMPUTER ORGANIZATION AND ARCHITECTURE, 2005 by John Wiley & Sons, Inc. USA, 2005.
- [9] Scott Mueller et Craig Zacker, Le PC Architecture, maintenance et mise à niveau, CAMPUS PRESS, France, 1999.
- [10] William Buchanan, Computer busses: DESIGN AND APPLICATION, CRC Press, USA, 2000.
- [11] Intel® 845G/845GL/845GV Chipset Datasheet, Intel Corporation, USA, 2002.
- [12] MindShare, Inc , Ravi Budruk, Don Anderson and Tom Shanley, PCI Express System Architecture, Addison Wesley, USA, 2003.

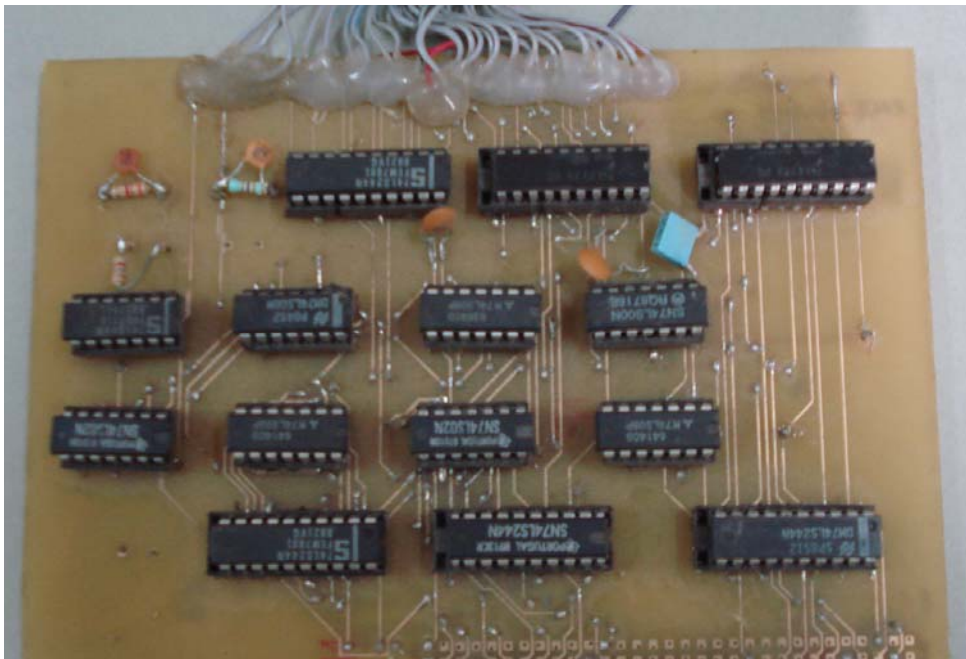
-
- [13] Intel® Intel® 915G/915GV/915GL/915P/915PL/910GL Express Chipset Datasheet, Intel Corporation, USA, 2005.
- [14] Universal Serial Bus Specification, REVISION 1.1, Compaq Computer Corporation, Intel Corporation, Microsoft Corporation, NEC Corporation, USA, 1998.
- [15] Universal Serial Bus Specification, REVISION 2.0, Compaq Computer Corporation, Hewlett-Packard Company, Intel Corporation, Lucent Technologies Inc, Microsoft Corporation, NEC Corporation, Koninklijke Philips Electronics N.V, USA, 2000.
- [16] Hesham El-Rewini and Mostafa Abd-El-Barr, ADVANCED COMPUTER ARCHITECTURE AND PARALLEL PROCESSING, 2005 by John Wiley & Sons, Inc. USA, 2005.
- [17] Arnold S. Berger, Hardware and Computer Organization, Elsevier Inc, USA, 2005.
- [18] M. Tischer, LA BIBLE PC, EDITIONS MICRO APPLICATION, France, 1997.
- [19] IA-32 INTEL ARCHITECTURE SOFTWARE DEVELOPER'S MANUAL, Intel Corporation, USA, 2002.
- [20] M.Maamoun et G.Zerari. ADRESSAGE MATERIEL DANS LES SYSTEMES A MICROPROCESSEUR AVEC UN ADRESSAGE PHYSIQUE ETENDU. 2001 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE01). IEEE Canada. Toronto, Ontario, Canada. 2001.
- [21] G.F.Marchiro. DECOUPAGE TRANSFORMATIONNEL POUR LA CONCEPTION DE SYSTEMES MIXTES LOGICIEL/MATERIEL. Thèse de Doctorat. Institut National Polytechnique de Grenoble. France. 1998.
- [22] M.Maamoun, A.Benbelkacem, D.Berkani. INTERFAÇAGE DANS LES SYSTEMES A MICROPROCESSEUR AVEC UN ADRESSAGE PHYSIQUE ACCELERE. 2002 IEEE Canadian Conference on Electrical and Computer Engineering (IEEE-CCECE02). Winnipeg, Manitoba, Canada. 2002.
- [23] Mark Balch, COMPLETE DIGITAL DESIGN, The McGraw-Hill Companies, USA, 2003.
- [24] William K. Lam, Hardware Design Verification: Simulation and Formal Method-Based Approaches, Prentice Hall PTR, USA, 2005.
- [25] S.Leibson. MANUEL DES INTERFACES. McGRAW-HILL. France. 1984.
- [26] M.Maamoun, A.Benbelkacem, D.Berkani, A.Guessoum. INTERFACING IN MICROPROCESSOR-BASED SYSTEMS WITH A FAST PHYSICAL ADDRESSING, The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications. Calgary, Alberta, Canada. 2003.
- [27] Phillip A. Laplante, REAL-TIME SYSTEMS DESIGN AND ANALYSIS, IEEE Press, USA, 2004.
- [28] VOJIN G. OKLOBDZIJA, THE COMPUTER ENGINEERING HANDBOOK, CRC Press, USA, 2002.

-
- [29] M.Maamoun, B.Laichi, A.Benbelkacem, D.Berkani. INTERFACING IN MICROPROCESSOR-BASED SYSTEMS WITH AN ADVANCED PHYSICAL ADDRESSING. The 4th IEEE International Workshop on System-on-Chip for Real-Time Applications, Banff, Alberta – Canada. July 19 - July 21, 2004.
- [30] Marc Defossez, Interfacing a 64-Bit DDR Memory Bus to a 32-Bit Microprocessor Bus, Xilinx, Inc, USA, 2006.
- [31] Nagesh Gupta and Maria George, Creating High-Speed Memory Interfaces with Virtex-II and Virtex-II Pro FPGAs, Xilinx, Inc, USA, 2004.
- [32] Robert Dueck, Digital Design with CPLD Applications and VHDL, Thomson Delmar Learning, 2005.
- [33] Xilinx Virtex-II Platform FPGAs: Complete Data Sheet, Xilinx, Inc, USA, 2005.
- [34] Xilinx Virtex-II Platform FPGA User Guide, Xilinx, Inc, USA, 2005.
- [35] Ricardo Reis and Jochen A.G. Jess, DESIGN OF SYSTEM ON A CHIP, Kluwer Academic Publishers, USA, 2004.
- [36] Brian Matas and Christian de Suberbasaux, "ROM, EPROM, & EEPROM Technology", Integrated Circuit Engineering Corporation, USA, 1997.
- [37] Brian Matas and Christian de Suberbasaux, Flash Memory Technology, Integrated Circuit Engineering Corporation, USA, 1997.
- [38] Brian Matas and Christian de Suberbasaux, SRAM Technology, Integrated Circuit Engineering Corporation, USA, 1997.
- [39] P.K. LALA. Digital System Design Using Programmable Logic Devices, Chapitre 5, pages 114–166. Computer Engineering. Prentice Hall, New Jersey, USA, 1990.
- [40] S. BROWN, R. FRANCIS, J. ROSE et Z. VRANESIC. Field Programmable Gate Arrays. Kluwer Academic Publishers, 1992.
- [41] A. BARNA et D.I. PORAT. Integrated Circuits in Digital Electronics, pages 413–420. John Wiley & Sons, New York, 1973.
- [42] ALTERA. MAX 3000A Programmable Logic Device Family, juin 2003.
- [43] Xilinx XC9500 In-System Programmable CPLD Family, Xilinx, Inc, USA, 2006.
- [44] "Designing with XC9500 CPLDs", Xilinx, Inc, USA, 1998.
- [45] Clive Maxfield, The Design warriors guide to FPGA, Elsevier Inc, USA, 2004.
- [46] Gregory Ray Goslin. A Guide to Using Field Programmable Gate Arrays (FPGAs) for Application-Specific Digital Signal Processing Performance, Xilinx, Inc, USA, 1995.
- [47] XC6200 Field Programmable Gate Arrays, Xilinx, Inc, USA, 1997.
- [48] Interfacing XC6200 to Microprocessors, Xilinx, Inc, USA, 1996.

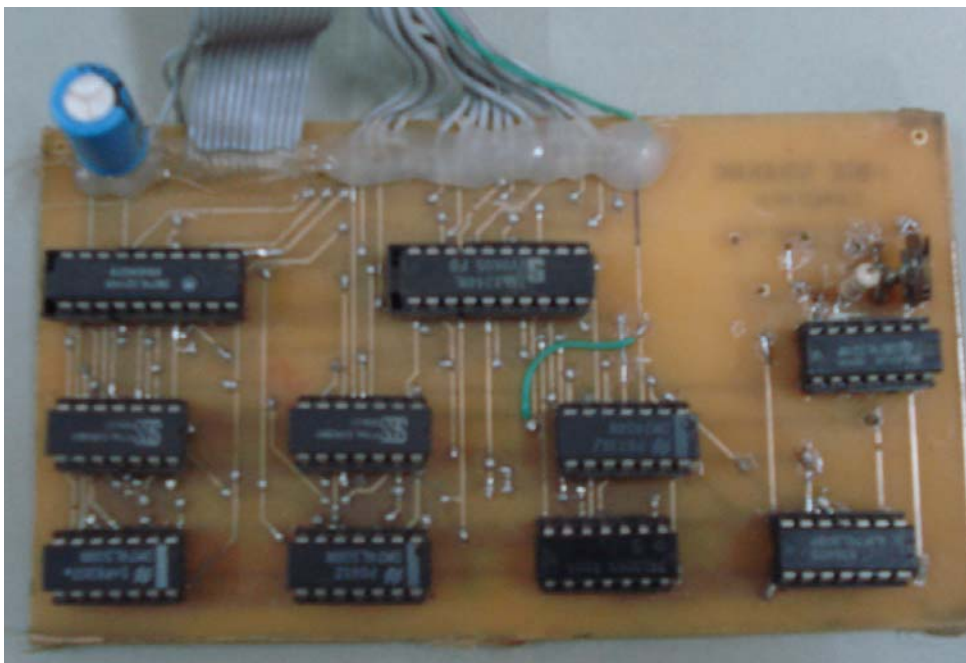
-
- [49] Reiner W.Hartenstein, Michael, Frank Gilbert, Designing for Xilinx XC6200 FPGAs, Proceedings of 8th International Workshop on Field-Programmable Logic and Applications, FPL'98, Tallinn, Estonia, August 31- September3, 1998.
- [50] Programmable Logic Design Quick Start Handbook, Xilinx, Inc, USA, 2006.
- [51] Using the Virtex SelectI/O Resource, Xilinx, Inc, USA, 2005.
- [52] Xilinx XC2000 Logic Cell Array Families, Xilinx, Inc, USA.
- [53] D. HILL et N.-S. WOO. The Benefits of Flexibility in Look-up Table FPGAs. Dans W. MOORE et W. LUK, éditeurs, Proc. Oxford 1991 International Workshop on Field Programmable Logic and Applications, pages 127–136, Abingdon, England, 1991. Abingdon EE&CS Books.
- [54] The Programmable Logic Data Book 2000, Xilinx, Inc, USA, 2000.
- [55] J. ROSE, R. FRANCIS, D. LEWIS et P. CHOW. “Architecture of Field-Programmable Gate Arrays: the Effect of Logic Block Functionality on Area Efficiency”. Solid-State Circuits, IEEE Journal of, 25(5) :1217–1225, October 1990.
- [56] S. SINGH, J. ROSE, P. CHOW et D. LEWIS. The Effect of Logic Block Architecture on FPGA Performance. Solid-State Circuits, IEEE Journal of, 27(3) :281–287, mars 1992.
- [57] Spartan-3E FPGA Family: Complete Data Sheet, Xilinx, Inc, USA, 2006.
- [58] Virtex-4 Family Overview, Xilinx, Inc, USA, 2007.
- [59] Virtex-5 Family Overview LX, LXT, and SXT Platforms, Xilinx, Inc, USA, 2007.
- [60] Virtex-4 Data Sheet: DC and Switching Characteristics, Xilinx, Inc, USA, 2006.
- [61] Virtex-4 User Guide, Xilinx, Inc, USA, 2007.
- [62] Virtex-5 Data Sheet: DC and Switching Characteristics, Xilinx, Inc, USA, 2007.
- [63] Virtex-5 SXT Platform Technical Backgrounder, Xilinx, Inc, USA, 2007.
- [64] ISE Quick Start Tutorial, Xilinx, Inc, USA, 2006.
- [65] Development System Reference Guide, Xilinx, Inc, USA, 2005.
- [66] Xilinx ISE Software.
- [67] Xilinx XC9572 In-System Programmable CPLD, Xilinx, Inc, USA, 2006.
- [68] Embedded Instrumentation Using XC9500 CPLDs, Xilinx, Inc, USA, 1997.
- [69] Data to Clock Phase Alignment, Xilinx, Inc, USA, 2002.
- [70] High Performance Multi-Port Memory Controller, Xilinx, Inc, USA, 2004.
- [71] Xilinx XC95108 In-System Programmable CPLD, Xilinx, Inc, USA, 2006.

-
- [72] Volnei A. Pedroni, *CIRCUIT DESIGN WITH VHDL*. MIT (Massachusetts Institute of Technology) Press, USA, 2004.
- [73] Keith Jack, *Video Demystified: A Handbook for the Digital Engineer*, Elsevier Inc, USA, 2005.
- [74] M.Maamoun, A.Benbelkacem, D.Berkani, *VIDEO SIGNAL GENERATION USING A NEW INTERFACING TECHNIQUE FOR COMPUTER SYSTEM*. 2004 First International Symposium on Control, Communications and Signal Processing, Hammamet, Tunisia, March 21-24, 2004.
- [75] *SAMSUNG KM4216C256 Data Sheet*, Samsung Electronics, Korea, 1998.
- [76] *SAMSUNG KM4232C259 Data Sheet*, Samsung Electronics, Korea, 1997.
- [77] M.Maamoun, B.Laichi, A.Benbelkacem, D.erkani. *REAL-TIME IMAGE GENERATION WITH SIMULTANEOUS VIDEO MEMORY READ/WRITE ACCESS AND FAST PHYSICAL ADDRESSING*. *WSEAS TRANSACTIONS on CIRCUITS AND SYSTEMS*, Issue 3, Volume 3, May 2004.
- [78] M.Maamoun, B.Laichi, D.Berkani. *APPLICATION DE L'ADRESSAGE PHYSIQUE RAPIDE A L'ARCHITECTURE DE VISUALISATION DANS LES SYSTEMES A MICROPROCESSEUR*. 2004 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE04), Sheraton Fallsview, Niagara Falls, Ontario, Canada, May 2-5, 2004.
- [79] Maria George, *Memory Interface Application Notes Overview*, Xilinx, Inc, USA, 2006.
- [80] Gerald W. Collins, PE, *FUNDAMENTALS OF DIGITAL TELEVISION TRANSMISSION*. A Wiley-Interscience, Publication, John Wiley & Sons, Inc. USA, 2001.
- [81] *Philips TDA8702 Datasheet*, Philips Semiconductors Product Specification, 1996.
- [82] *NEC μ PD431000 Datasheet*, NEC Electronics Corporation, Japan, 2006.
- [83] *ISSI IS61C1024 Datasheet*, Integrated Silicon Solution, Inc, USA, 1999.
- [84] Maria George, *DDR2 SDRAM Memory Interface for Virtex-II Pro FPGAs*, Xilinx, Inc, USA, 2005.
- [85] *Connecting Virtex-II Devices to a 3.3V/5V PCI Bus*, Xilinx, Inc, USA, 2004.

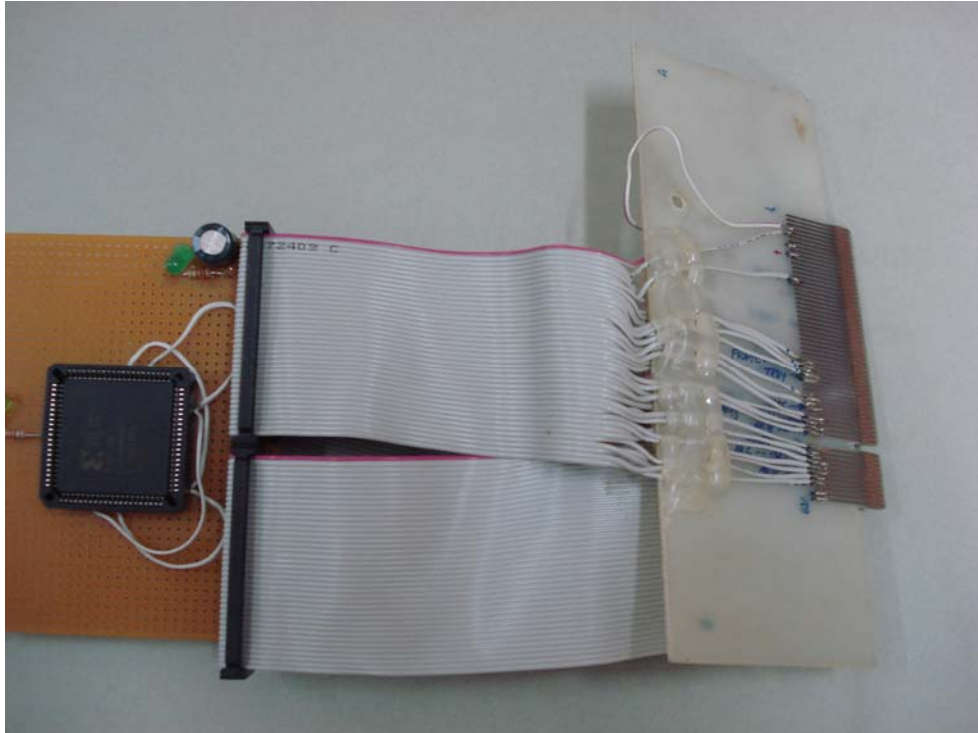
Annexes



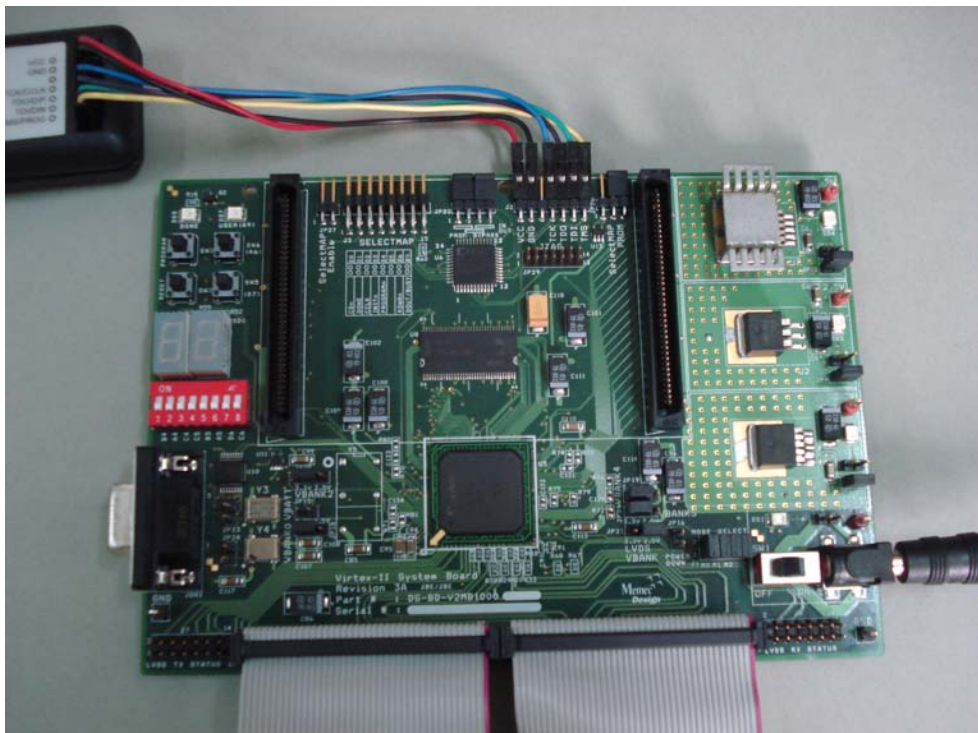
Circuit de l'Adressage Physique Etendu avec logique câblée.



Circuit de lecture avec logique câblée.



Interface de test PCI avec le XC 95108 (CPLD de Xilinx).



Carte de développement FPGA DS-BD-V2MB1000 (Virtex-II de Xilinx)



US006948022B2

(12) **United States Patent**
Ishibashi

(10) **Patent No.:** **US 6,948,022 B2**
(45) **Date of Patent:** **Sep. 20, 2005**

(54) **DIGITAL IMAGE TRANSFER CONTROLLER**

(75) Inventor: **Norio Ishibashi, Kanagawa (JP)**

(73) Assignee: **Sony Corporation, Tokyo (JP)**

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 315 days.

(21) Appl. No.: **10/239,330**

(22) PCT Filed: **Jan. 25, 2002**

(86) PCT No.: **PCT/JP02/00568**

§ 371 (c)(1),
(2), (4) Date: **Feb. 5, 2003**

(87) PCT Pub. No.: **WO02/060175**

PCT Pub. Date: **Aug. 1, 2002**

(65) **Prior Publication Data**

US 2003/0158978 A1 Aug. 21, 2003

(30) **Foreign Application Priority Data**

Jan. 25, 2001 (JP) 2001-016820
Jan. 26, 2001 (JP) 2001-017934

(51) **Int. Cl.**⁷ **G06F 13/14**; G06F 15/00;
G06T 1/00; G09G 5/38

(52) **U.S. Cl.** **710/305**; 710/308; 345/204;
345/418

(58) **Field of Search** 710/305, 306,
710/308, 313, 22, 62; 345/204, 531, 418,
501, 562; 463/40, 1; 348/514, 46; 701/211;
340/286.01; 370/402; 711/101

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,891,760 A * 1/1990 Kashiwazaki et al. 701/200

5,576,736 A * 11/1996 Miwa et al. 345/534
5,877,708 A * 3/1999 Hijikata 340/995.19
5,941,968 A * 8/1999 Mergard et al. 710/308
6,557,682 B2 * 5/2003 Imamura 192/48.5

FOREIGN PATENT DOCUMENTS

JP 6-125488 5/1994

OTHER PUBLICATIONS

“Design and implementation of multimedia functional module for digital TV” by Kim et al. (abstract only) Publication Date: Aug. 2004.*

“Video signal generation using a new interfacing technique for computer system” by Maamoun et al. (abstract only) Publication Date: 2004.*

* cited by examiner

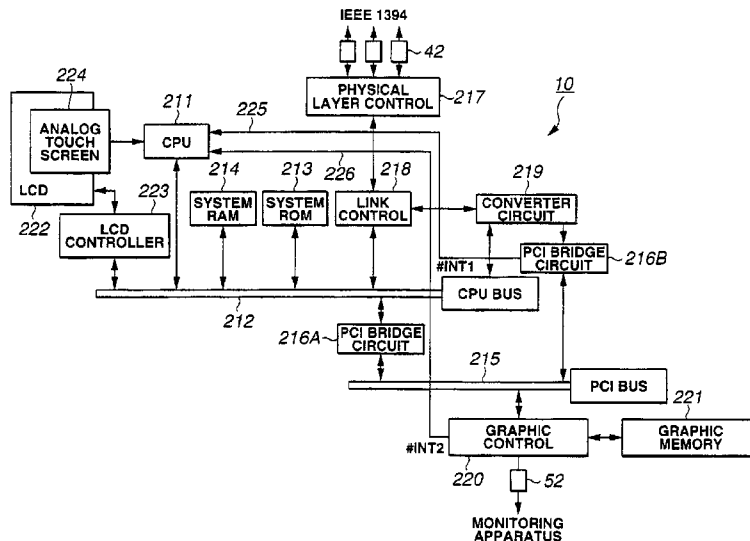
Primary Examiner—Gopal C. Ray

(74) *Attorney, Agent, or Firm*—Frommer Lawrence & Haug LLP; William S. Frommer; Samuel S. Lee

(57) **ABSTRACT**

A data transfer device (10) converts the data stream input to it by IEEE 1394 isochronous transmission from an image processing device (20) connected to it into data in a color signal mode adapted to PCI transfer by means of a converter circuit (219), while controlling an address of PCI bridge circuit (216B) as bus master, transmits the image data from the converter circuit (219) to a graphic memory (221) by DMA transmission by way of a PCI bus (215) and writes the data in the graphic memory (221) by means of a graphic control circuit (220). It then reads the image data from the graphic memory (221) by means of the graphic control circuit (220) and transfers the data to an electronic device (30).

11 Claims, 7 Drawing Sheets



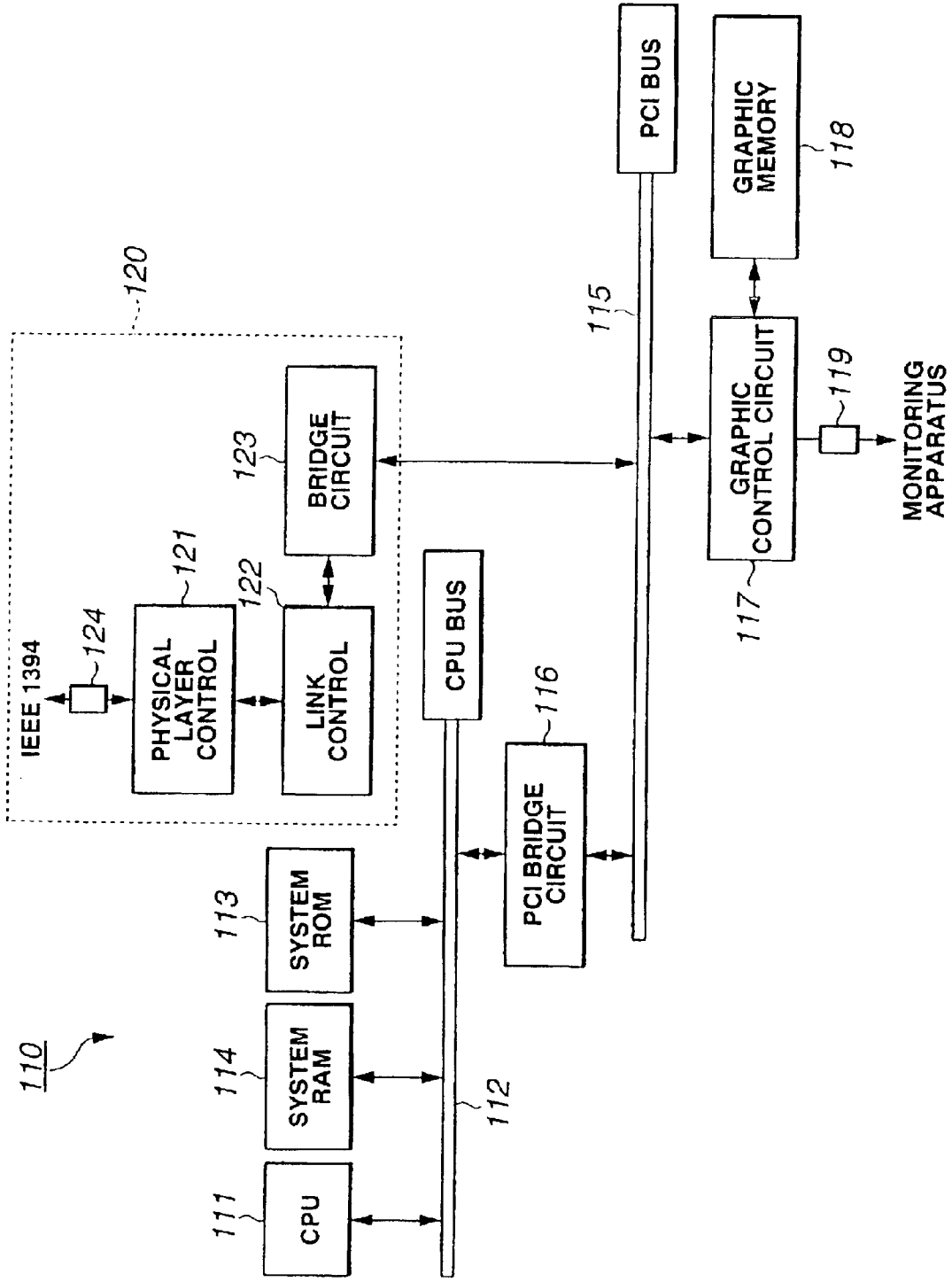


FIG. 1

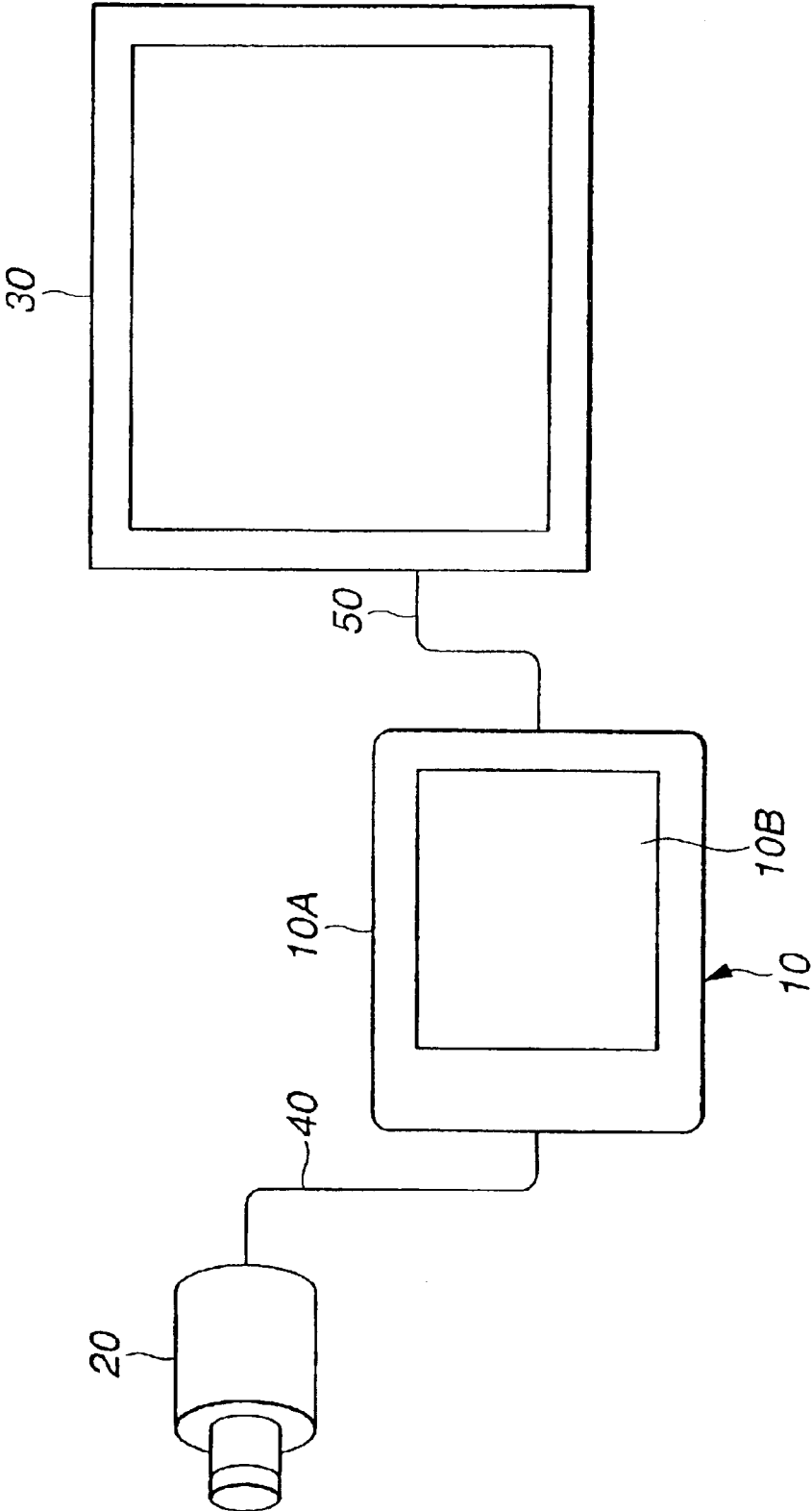


FIG.2

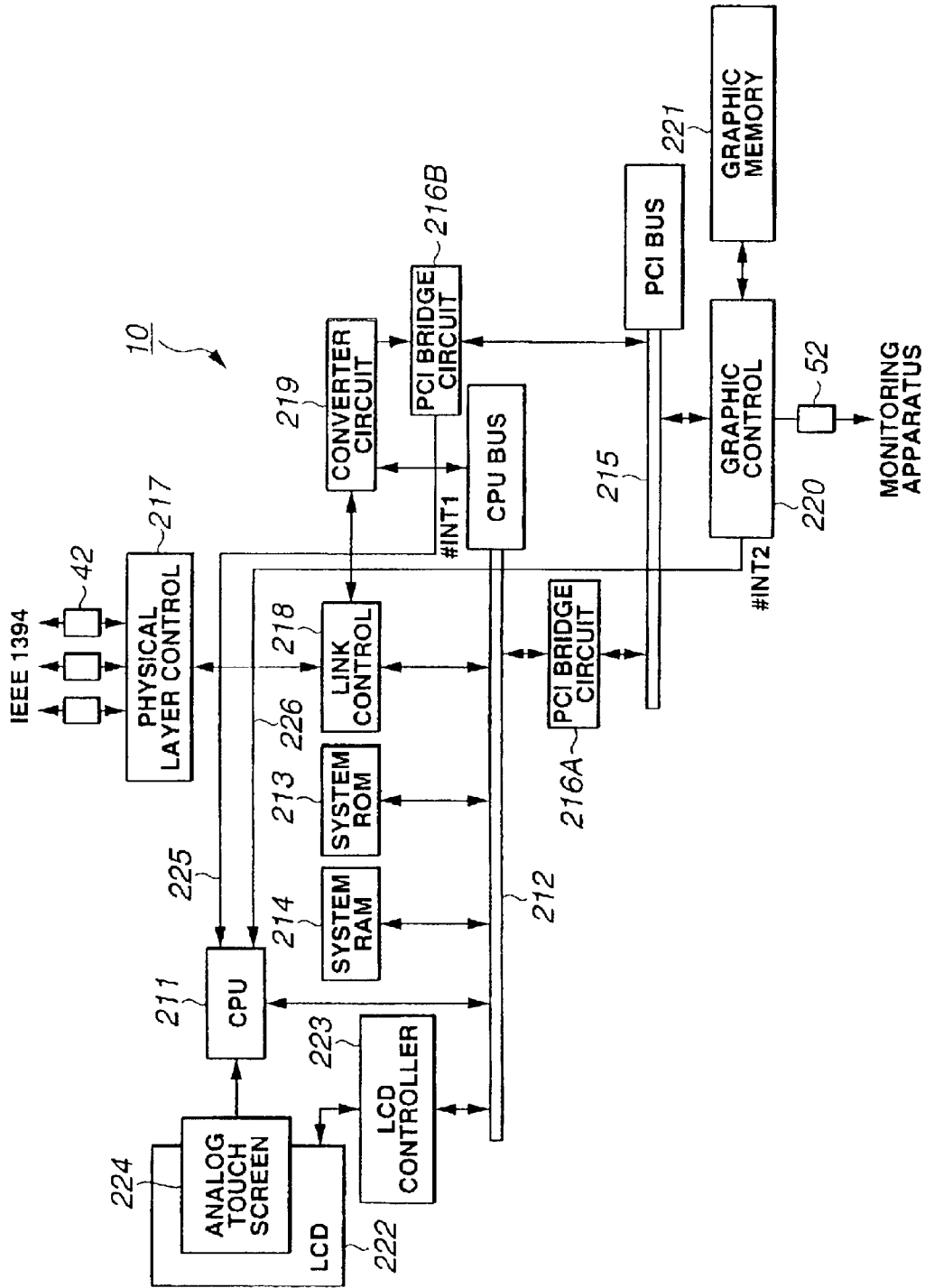


FIG.3

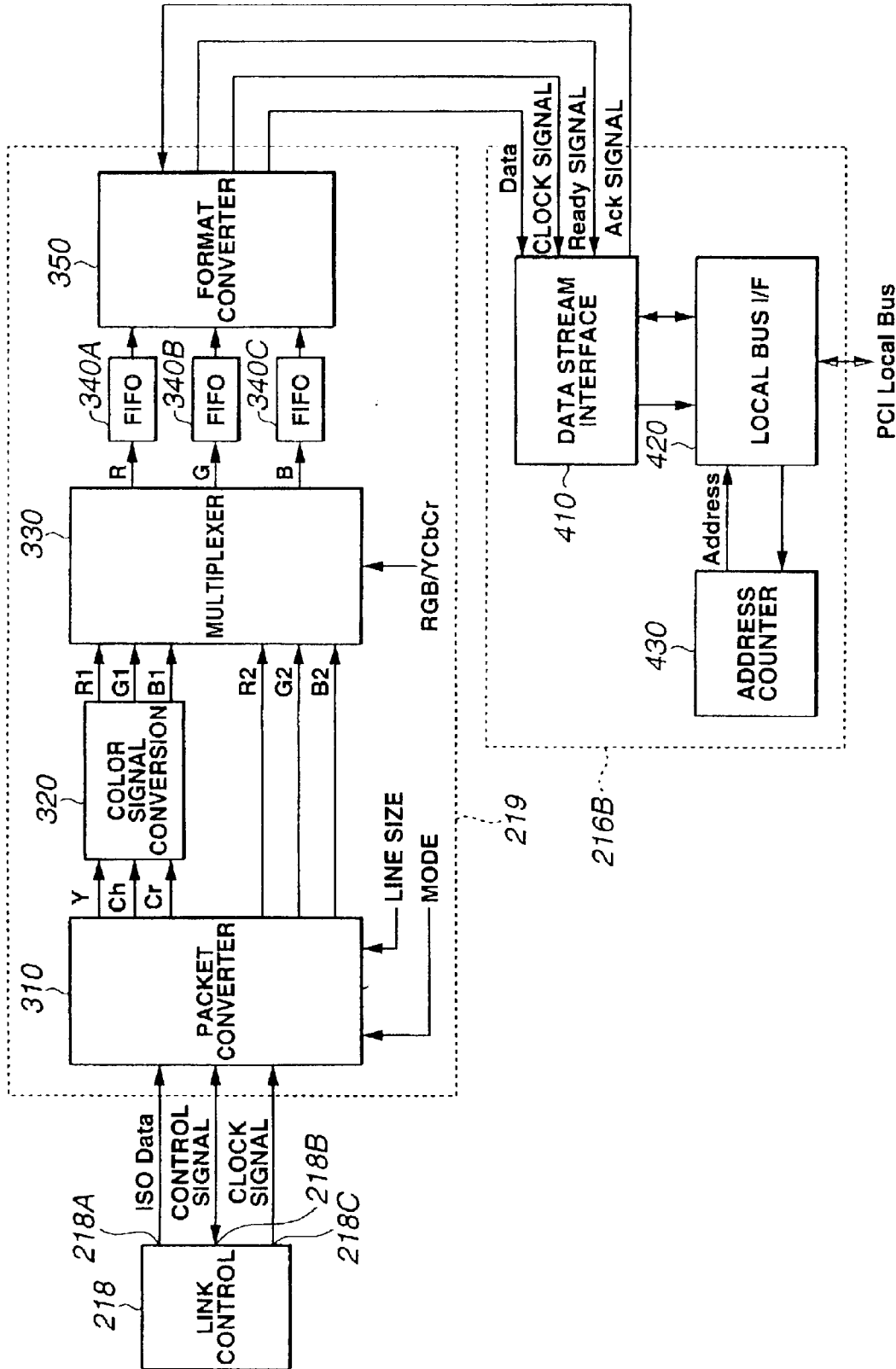


FIG. 4

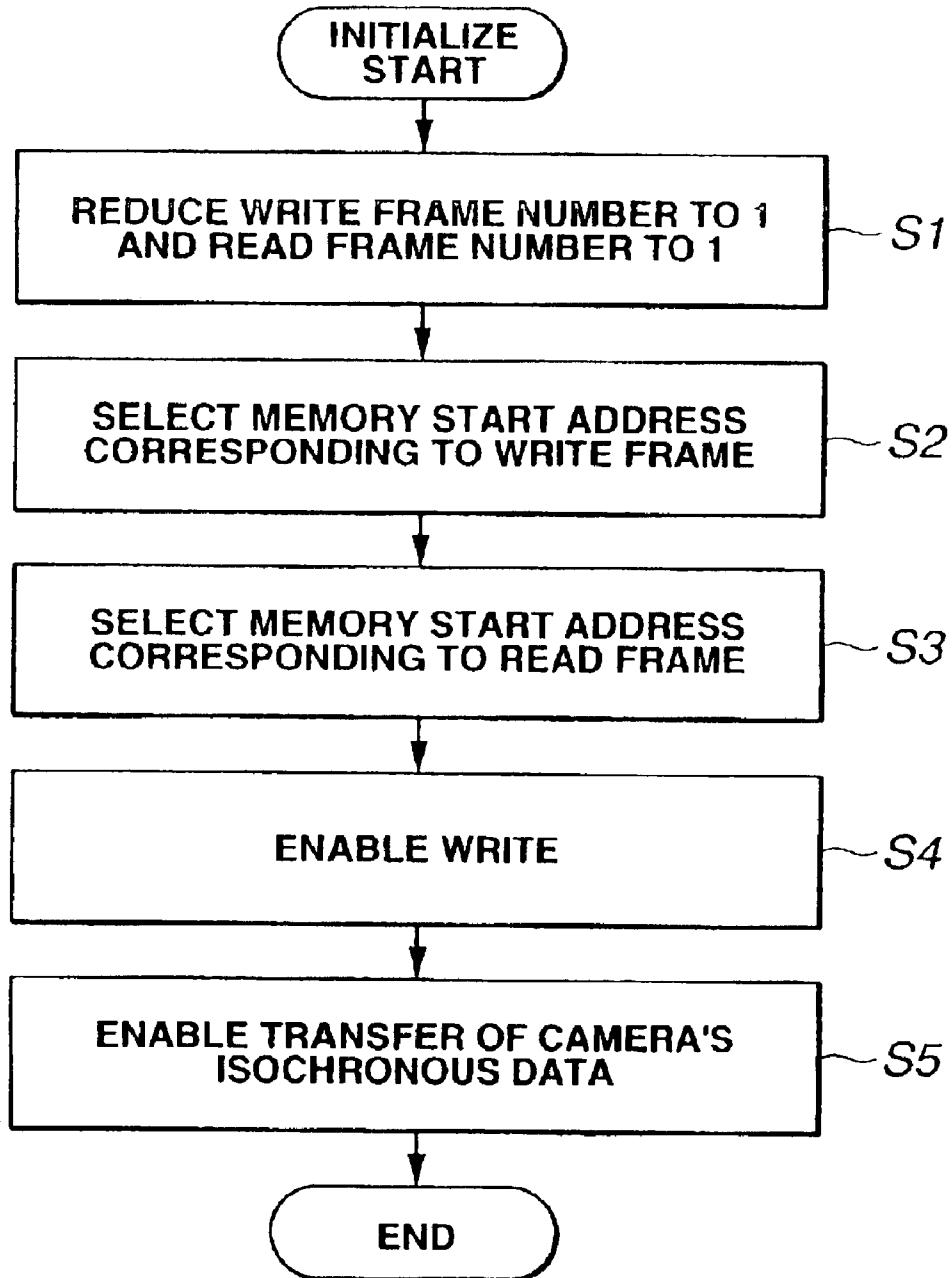


FIG.5

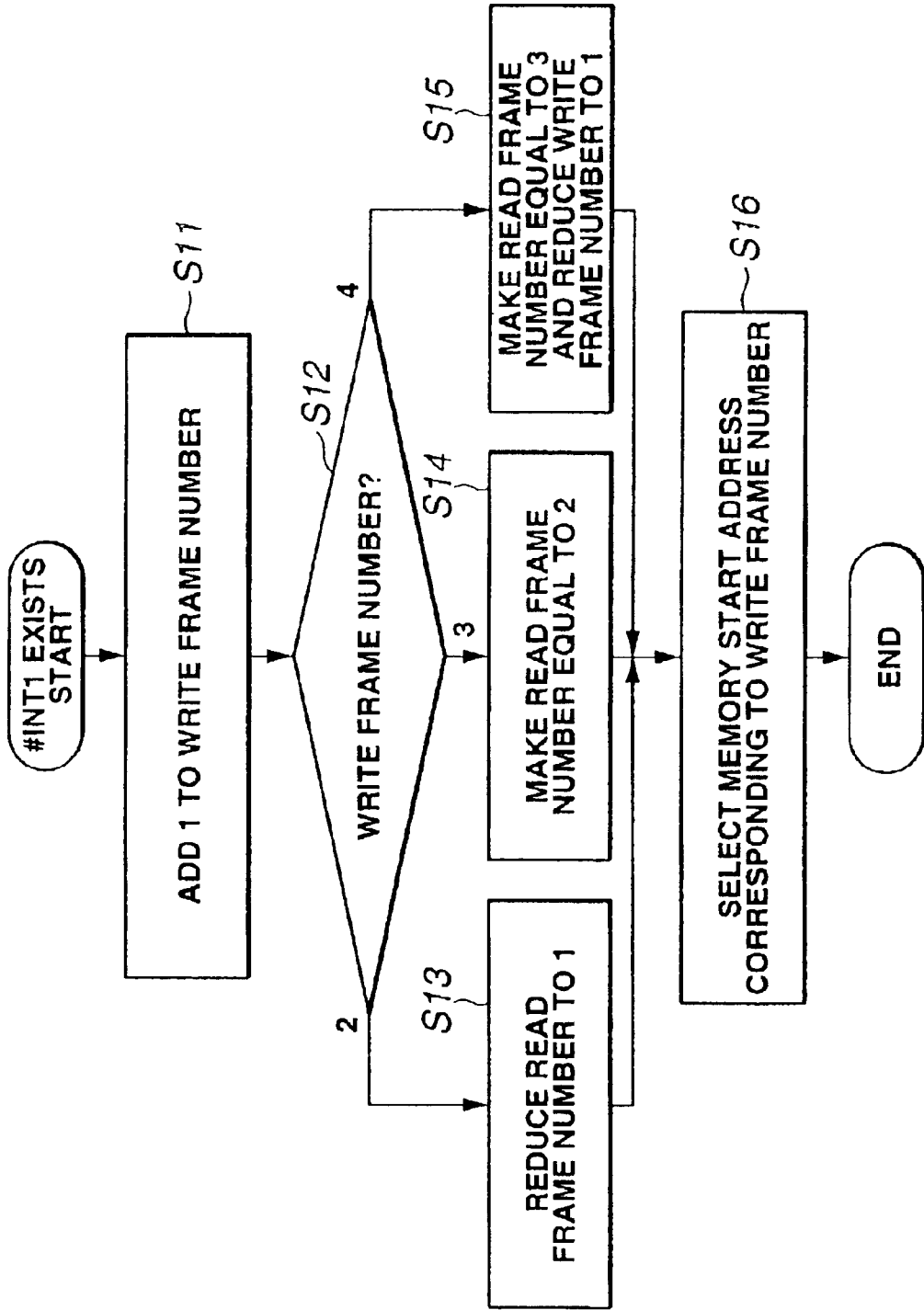


FIG. 6

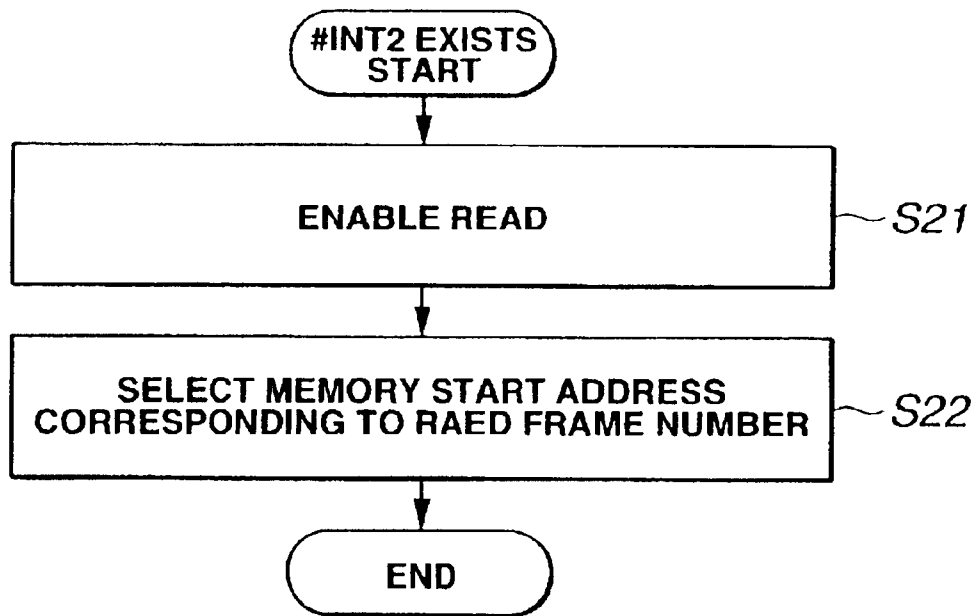


FIG.7

DIGITAL IMAGE TRANSFER CONTROLLER

TECHNICAL FIELD

This invention relates to a data transfer device adapted to perform control operations for outputting image data input from an image processing device to an electronic device.

BACKGROUND ART

When displaying an image picked up by a digital camera on the display screen of a monitoring apparatus, it is necessary to convert the data format of the image data of the digital camera into a format adapted to display the image on the screen of the monitoring apparatus. Normally, the data format of the image data of the digital camera is converted by inserting a converter board having a data format converting feature between the digital camera and the monitoring apparatus. Such a converter board can be built in a personal computer, so as to perform a processing operation on the image data transmitted from the digital camera for format conversion and transfer the image data to the monitoring apparatus. With this arrangement, it is possible to provide a system that allows smooth image pickup operations.

FIG. 1 of the accompanying drawings schematically illustrates the circuit configuration of a converter board **110** having such a data format convening feature. Referring to FIG. 1, the converter board **110** is connected to a digital camera adapted to output image data in a format conforming to the IEEE 1394 Standard and comprises a CPU **111**, a CPU bus **112**, a system ROM **113**, a system RAM **114**, a PCI bus **115**, a PCI bridge circuit **116**, a graphic control circuit **117** and a graphic memory **118**. An IEEE 1394 control board **120** for controlling data transmission between the converter board **110** and the digital camera (not shown) is also connected to the PCI bus **115**.

The CPU **111** controls the component sections of the converter board **110** by way of the CPU bus **112** according to the programs stored in the system ROM **113** and executes various processing operations. The system ROM **113** typically comprises a flash ROM and stores various programs and fixed data necessary for the operations of the CPU **111**. The system RAM **114** typically comprises an SDRAM and temporarily stores data necessary for various control operations of the CPU **111**. The PCI bus **115** is used for transmitting data between the CPU **111** and peripheral elements. The PCI bridge circuit **116** controls the connection between the PCI bus **115** and the CPU bus **112**.

The graphic control circuit **117** writes image data in and reads image data from the graphic memory **118** having a storage area for storing data of a frame on a time division basis, the data originating from the digital camera. In other words, the graphic control circuit receives the data transmitted from the CPU **111** by way of the PCI bridge circuit **116** and the PCI bus **115** and writes the data in the graphic memory **118**. The graphic control circuit **117** also reads the image data stored in the graphic memory **118** and transmits them to the monitoring apparatus by way of the connector **119** and a cable (not shown).

The IEEE 1394 control board **120** comprises an IEEE 1394 physical layer control circuit **121**, a link control circuit **122** and a PCI bridge circuit **123**.

The physical layer control circuit **121** is a so-called PHY chip for controlling IEEE 1394 physical layers and adapted to convert the analog signal input from an IEEE 1394 cable

(not shown) by way of the connector **124** into digital data. The link control circuit **122** is a so-called LINK chip for controlling IEEE 1394 link layers and adapted to control the interface between the physical layer control circuit **121** and the PCI bridge circuit **123**. The PCI bridge circuit **123** controls the connection of the link control circuit **122** and the PCI bus **115**.

For transferring the image data input from the digital camera to the monitoring apparatus by means of the converter board **110**, firstly the image data received by the IEEE 1394 control board **120** is taken into the CPU **111** by way of the PCI bridge circuit **123**, the PCI bus **115**, the PCI bridge circuit **116** and the CPU bus **112** and its format of the image data is converted into the format to be used for displaying an image on the display screen of the monitoring apparatus by means of a processing operation using the software stored in the CPU **111**.

The image data whose format is converted by the CPU **111** is then transmitted to the graphic control circuit **117** by way of the CPU bus **112**, the PCI bridge circuit **116** and the PCI bus **115** and finally sent to the monitoring apparatus by means of the graphic control circuit **117** and the graphic memory **118**.

Meanwhile, when converting the format of the image data by means of the software stored in the CPU **111**, the processing operation of the CPU **111** is slow relative to the data transmission rate for IEEE 1394 isochronous data to give rise to a problem of a reduced frame rate for the image to be displayed. Additionally, there is a problem of a reduced processing rate of the entire system because of a lowered operational capacity of the system due to the increased processing load of the CPU **111** caused by the data transmission.

Furthermore, with the above described known converter board, the transmission rate of the image data input from the digital camera and that of the image data transmitted to the monitoring apparatus can be different. In most cases, this difference is caused by the difference between the rate of writing data in and that of reading data from the graphic memory. Therefore, when the image data of a frame is written into and read from the storage area of the graphic memory for a frame and there arises a difference between the rate of writing image data and that of reading image data, there can arise a phenomenon that the lines being read out can outrun the lines being written in. Such a phenomenon can result in line noise (outrun scanning noise) that appear on the image being displayed on the screen of the monitoring apparatus. Particularly, when a moving image is continuously transmitted and displayed on a real time basis, such an outrun scanning noise can periodically occur to degrade the image quality.

The above identified problems need to be dissolved for any systems adapted to transmit image data input from a digital camera to a monitoring apparatus and those adapted to transmit image data input from an image processing device to an electronic device.

DISCLOSURE OF THE INVENTION

In view of the above described circumstances, it is therefore an object of the present invention to provide a data transfer device that can avoid the problem of a reduced frame rate and a reduced operation speed without causing the CPU to bear an excessive load when converting the format of image data of a terminal device such as digital camera into that of image data to be displayed and transmitting the image data to an electronic device such as monitoring apparatus.

Another object of the present invention is to provide a data transfer device that can avoid the problem of generation of outrun scanning noise due to the difference between the rate of writing image data into and that of reading image data from a graphic memory.

In an aspect of the invention, the above objects and other objects are achieved by providing a data transfer device for transferring the image data input from an image processing device to an electronic device, said device comprising: a conversion processing section for performing a predetermined conversion processing operation on the image data input from said image processing device; a graphic memory for temporarily storing the image data; a graphic control circuit for writing the image data transmitted from said conversion processing section to said graphic memory or reading the image data stored in said graphic memory and transmitting it to said electronic device; and a CPU for selecting a first data transmission route for transmitting the image data from said conversion processing section to said graphic control circuit or a second data transmission route for transmitting a control signal to said graphic control circuit; said conversion processing section being adapted to convert the stream of image data input from said image processing device into an output format.

In another aspect of the invention, there is provided a data transfer device for transferring the image data input from an image processing device to an electronic device, said device comprising: a conversion processing section for performing a predetermined conversion processing operation on the image data input from said image processing device; a graphic memory for temporarily storing the image data; a graphic control circuit for writing the image data transmitted from said conversion processing section to said graphic memory or reading the image data stored in said graphic memory and transmitting it to said electronic device, and a CPU for selecting a first data transmission route for transmitting the image data from said conversion processing section to said graphic control circuit or a second data transmission route for transmitting a control signal to said graphic control circuit; said conversion processing section being adapted to convert the stream of image data input from said terminal device into a format suited for displaying the image; said graphic memory having an image data storage area for a plurality of frames; said graphic control circuit being adapted to sequentially select the image data storage area for a plurality of frames and control the operation of writing and reading image data on a time division basis.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic block diagram of a converter board having a functional feature of converting the format of data, illustrating the circuit configuration thereof.

FIG. 2 is a schematic illustration of a digital camera controller connected to a digital camera and a multi-scan monitor.

FIG. 3 is a schematic block diagram of a digital camera controller realized by applying the present invention, illustrating the internal configuration thereof.

FIG. 4 is a schematic block diagram of a converter circuit and a PCI bridge circuit arranged in the digital camera controller, illustrating the configurations thereof.

FIG. 5 is a flow chart of the start of a data writing operation of the digital camera controller.

FIG. 6 is a flow chart of an interrupt processing operation of the CPU when the CPU detects detection signal #INT1.

FIG. 7 is a flow chart of an interrupt processing operation of the CPU when the CPU detects detection signal #INT2.

BEST MODE FOR CARRYING OUT THE INVENTION

Now, the present invention will be described by referring to the accompanying drawings that illustrate preferred embodiments of the invention.

FIG. 2 is a schematic illustration of a digital camera controller realized by applying the present invention and connected to a digital camera and a multi-scan monitor. As shown in FIG. 2, the digital camera controller 10 is arranged between a digital camera 20 and a multi-scan monitor 30 and adapted to convert the image data transmitted from the digital camera 20 into image data to be used for displaying the image on the multi-scan monitor 30 and output it.

The digital camera controller 10 is provided with a liquid crystal display 10B comprising a liquid crystal display element and arranged on the front surface of the device cabinet 10A.

Note that, in FIG. 2, the digital camera controller 10 and the digital camera 20 are connected to each other by way of a cable conforming to the IEEE 1394 Standard (IEEE 1394 cable) and the digital camera controller 10 and the multi-scan monitor 30 are connected to each other by way of a Dsub15pin cable 50.

The digital camera controller may be connected to a plurality of digital cameras 20 and a plurality of multi-scan monitors 30.

The digital camera 20 is adapted to pick up both still images and moving images. A plurality of resolutions are provided to correspond to available image modes (VGA, SVGA, XGA, SXGA, etc.) and any of the resolutions may be selected for operation. The digital camera 20 outputs image data in a signal format conforming to the IEEE 1394 Standard. The digital camera 20 may be controlled directly by the user operating it or remotely by way of the digital camera controller 10.

A number of sets of parameters are provided for the operation of the multi-scan monitor 30. The image input through the cable 50 is displayed by using a selected set of parameters corresponding to the selected image mode (VGA, SVGA, XGA, SXGA, etc.). The parameters include resolution (image size), frame rate, horizontal frequency and pixel frequency. The multi-scan monitor 30 may be controlled directly by the user operating it or remotely by way of the digital camera controller 10.

Now, the internal configuration of a digital camera controller realized by applying the present invention will be described below.

FIG. 3 is a schematic block diagram of a digital camera controller realized by applying the present invention, illustrating the internal configuration thereof. As shown in FIG. 3, the digital camera controller 10 comprises a CPU 211, a CPU bus 212, a system ROM 213, a system RAM 214, a PCI bus 215, PCI bridge circuits 216A, 216B, a physical layer control circuit 217, a link control circuit 218, a converter circuit 219, a graphic control circuit 220, a graphic memory 221, an LCD 222, an LCD controller 223 and an analog touch screen 224.

The CPU 211 controls the sections of the digital camera controller 10 by way of the CPU bus 212 according to the programs stored in the system ROM 213. The CPU 211 receives interrupt signal #INT1 that indicates the end of an operation of writing the data of a frame (image to be displayed) from the PCI bridge circuit 216B and also interrupt signal #INT2 that indicates the end of an operation of reading the data of a frame from the graphic control

circuit **220**. The CPU **211** controls operations of writing image data to and reading image data from the graphic memory **221** that are performed by the graphic control circuit **220** according to the received interrupt signals #INT1, #INT2. Interrupt signals #INT1, #INT2 are typically transmitted by way of dedicated interrupt signal lines **225**, **226**.

The system ROM **213** is typically a flash ROM and stores various programs and fixed data necessary for the operation of the CPU **211**. The system RAM **214** is typically an SDRAM and temporarily stores data necessary for various control operations of the CPU **211**. The system RAM **214** is provided with a read frame register and a write frame register for storing frame numbers that are used wherein the CPU **211** manages the storage areas to be used for reading image data and those to be used for writing image data of the graphic memory **221** according to the interrupt signals #INT1, #INT2.

The PCI bus **215** is used for transmission of data between the CPU **211** and peripheral elements. The PCI bus **215** can transmit data streams by means of isochronous transfer (realtime data) conforming to the IEEE 1394 Standard.

The PCI bridge circuit **216A** is a device for controlling the connection between the PCI bus **215** and the CPU bus **212**. The PCI bridge circuit **216B** is a device for transmitting the image data input from the converter circuit **219** to the graphic control circuit **220** by way of the PCI bus **215**. Each time the PCI bridge circuit **216B** transmits the image data of a frame to the graphic control circuit **220**, it outputs an interrupt signal #INT1 that indicates the end of an operation of writing the data of a frame to the graphic memory **221** to the CPU **211**. The PCI bridge circuit **216B** has a data counter (not shown) to be used for outputting the interrupt signal #INT1. The PCI bridge circuit **216B** may replace the graphic control circuit **220** and perform the operation of writing image data to the graphic memory **221**. For this purpose, the PCI bridge circuit **216B** has a function of selecting a write address for writing image data to the graphic memory **221** and transmitting image data to the graphic memory **221**. Thus, with the system configuration of FIG. 3, graphic processing operations are controlled by the PCI bridge circuit **216B** and the graphic control circuit **220**.

The physical layer control circuit **217** is a so-called PHY chip for controlling physical layers defined by the IEEE 1394 Standard and converts the analog data input from the IEEE 1394 cable **40** by way of the connector **42** into digital data. The link control circuit **218** is a so-called LINK chip for controlling link layers defined by the IEEE 1394 Standard. It controls the interface between the physical layer control circuit **217** and the CPU **211** and the interface between the physical layer control circuit **217** and the converter circuit **219**. The converter circuit **219** converts the format of the image data transmitted from the digital camera **20** into that of image data suited for displaying the image.

The graphic memory **221** is a memory for temporarily storing the image data transmitted by way of the PCI bus **215** and has storage areas for three frames (multi-frame area) of the multi-scan monitor **30**. Hereinafter, the storage areas for three frames are referred to as first frame storage area, second frame storage area and third frame storage area respectively.

The graphic control circuit **220** receives the image data transmitted from the digital camera **20** by way of the connector **42**, the physical layer control circuit **217**, the link control circuit **218** and the converter circuit **219** and also through the PCI bridge circuit **216B** and the PCI bus **215**.

The graphic control circuit **220** receives the control data transmitted from the CPU **211** through the CPU bus **212**, the PCI bridge circuit **216A** and the PCI bus **215**. The graphic control circuit **220** writes the received image data into the graphic memory **221**. It also reads image data stored in the graphic memory **221** and transmits the data to the monitor **30** by way of the connector **52** and the cable **50**. The graphic control circuit **220** has a read data counter (not shown) to be used for outputting interrupt signal #INT2 that indicates the end of an operation of reading out the data of a frame to the CPU **211**.

The LCD **222** displays various pieces of information on the LCD display screen **10B** under the control of the CPU **211**. The LCD controller **223** controls the display operation of the LCD **222**.

The analog touch screen **224** is arranged on the display screen **10B** of the LCD **222** and operates as touch panel to be used for detecting the user operation that is performed in response to the contents displayed on the LCD **222**.

There are two routes for transmitting image data and control data to the graphic control circuit **220** as described below. The first data transmission route is used to transfer the image data input by way of the connector **42** to the graphic control circuit **220** through the physical layer control circuit **217**, the link control circuit **218**, the converter circuit **219**, the PCI bridge circuit **216B** and the PCI bus **215**. The second data transmission route is used to temporarily take in the image data that are input by way of the connector **42** to the CPU **211** (system RAM **214**) by way of the physical layer control circuit **217**, the link control circuit **218** and the CPU bus **212** and transfer the data to the graphic control circuit **220** by way of the CPU bus **212**, the PCI bridge circuit **216A** and the PCI bus **215**.

The digital camera controller **10** transmits the image data normally through the first data transmission route so as to alleviate the load of the CPU **211**. The digital camera controller may use the second data transmission route for transferring the image data processed by the CPU **211** to the graphic control circuit **220** and writing the data into the graphic memory **221** or inversely for reading the image data stored in the graphic memory **221** to the CPU **211** by way of the graphic control circuit **220**. The control data from the CPU **211** are input to the graphic control circuit **220** by way of the second data transmission route.

FIG. 4 is a schematic block diagram of the converter circuit **219** and the PCI bridge circuit **216B** of the digital camera controller **10** of this embodiment, illustrating the configuration thereof.

Referring to FIG. 4, the link control circuit **218** outputs IEEE 1394 isochronous data (ISO data) from data output terminal **218A** that is independent from the interface of the CPU **211**. More specifically, the link control circuit **218** outputs isochronous data from the data output terminal **218A** in synchronism with clock signal CLK from clock output terminal **218C** and supplies the data to the converter circuit **219**.

Transmission/reception of control signals takes place between the control terminal **218B** and the converter circuit **219**. The control signal contains a signal synchronized with the leading packet of a frame of image data and a signal indicating effective data (image itself) in the packet.

The converter circuit **219** comprises a packet converter **310**, a color signal conversion (YCbCr→RGB) circuit **320**, a multiplexer **330**, FIFO buffers **340A**, **340B**, **340C** and a PCI format converter **350**.

Data on the format corresponding to the image pickup mode and the pixel size of the digital camera **20** are written

in advance in the converter circuit **219** by the CPU **211** along with other data.

The packet converter **310** comprises a line size register and a mode register (not shown) as internal registers. These internal registers are made to store data on line size and mode in advance by the CPU **211**.

The packet converter **310** receives control signals from the link control circuit **218**. The packet converter **310** resets the converter circuit **219** and detects the leading packet of a frame according to the signal synchronized with the leading packet of the frame and contained in the received control signal. Additionally, the packet converter **310** performs a processing operation corresponding to the line size data stored in the line size register and the mode data (indicating the color signal format) stored in the mode register according to the signal indicating the effective data in the packet and contained in the control signal.

Specific conversion processing methods are predefined for the modes including 8-bit Mono, YCbCr 4:1:1, YCbCr 4:2:2, YCbCr 4:4:4 and RGB, the data of which are stored in the mode register. For example, for each of the modes of YCbCr 4:1:1, YCbCr 4:2:2 and YCbCr 4:4:4, CbCr is converted to YCbCr 4:4:4 by linear interpolation and the obtained signals are transmitted to the color signal conversion (YCbCr→RGB) circuit **320**, which converts the signals into RGB signals. As a result, color signals **R1**, **G1**, **B1** are output from the color signal conversion circuit **320** to the multiplexer **330**. On the other hand, Y=R=G=B is used for the 8-bit Mono mode, whereas color signals **R2**, **G2**, **B2** are output from the packet converter **310** to the multiplexer **330** without any conversion in the RGB mode.

The line size register stores the number of pixels of a line (line size) for the purpose of processing the tail end of each line.

The multiplexer **330** selects color signals **R1**, **G1**, **B1** coming from the color signal conversion circuit **320** in each of the modes of YCbCr 4:1:1, YCbCr 4:2:2 and YCbCr 4:4:4. In the 8-bit Mono mode or the RGB mode, the multiplexer **330** selects color signals **R2**, **G2**, **B2** coming from the packet converter **310**. Then, the multiplexer **330** outputs the selected color signals to the FIFO buffers **340A**, **340B**, **340C** as output signals **R**, **G**, **B**.

The FIFO buffers **340A**, **340B**, **340C** take a role of absorbing the difference between the transfer rate of isochronous data from the digital camera **20** and the PCI transfer rate. The data of the FIFO buffers **340A**, **340B**, **340C** are sequentially output to the PCI format converter **350**.

The PCI format converter **350** converts the 8-bit RGB data (data of a total of 24 bits) into a data of 32 bits to be used for PCI transfer. The 32-bit data obtained by the conversion is transferred to the PCI bridge circuit **216** in synchronism with the clock.

The PCI bridge circuit **216B** comprises a data stream interface **410**, a PCI local bus interface **420** and a PCI address counter section **430**.

The data stream interface **410** receives a data stream of clock signal CLK and image data from the converter circuit **219** and controls the interface between the converter circuit **219** by means of handshake of Ready signal and ACK signal. Note that the above described data stream is formed by causing the isochronous data transfer rate and the PCI transfer rate to match each other by means of handshake of Ready signal and ACK signal.

The PCI local bus interface **420** receives as input the data stream and the clock signal CLK that are received by the

data stream interface **410**. Then, the PCI local bus interface **420** transmits the data stream to the PCI bus **215** according to the clock signal CLK.

The PCI address counter section **430** controls the address to be used for transferring the data stream by way of the PCI bus **215**. The PCI address counter section **430** also controls the address to be used for writing the data stream in a predetermined area of the graphic memory **221** by way of the graphic control circuit **220**.

More specifically, for controlling the DMA transmission, the PCI bridge circuit **216B** takes the role of so-called PCI master, while the graphic control circuit **220** plays the part of so-called PCI slave.

Note that, when the operation of writing data for a frame in an operation of writing a data stream in the graphic memory **221** without resorting to software, the next image data needs to be overwritten on the same area of the graphic memory **221** from the leading address. The PCI address counter section **430** performs the address control operation for overwriting the image data. The PCI address counter section **430** comprises a total number of DMA data register for storing the total number of image data of a frame, a DMA data counter (down counter) for counting DMA data, a PCI address counter for counting PCI addresses and a start address register for storing the start address, or the leading address, of the predetermined storage area.

When the DMA data counter counts down and the reading of the counter becomes equal to "0", the PCI address counter section **430** presets the DMA data counter and the PCI address counter, using the respective registered values of the total number of DMA data register and the start address register.

Then, the PCI address counter section **430** decrements (-1) the DMA data counter and increments (+1) the PCI address counter each time a unit data (32 bits (4 bytes)) is output from the PCI bridge circuit **216B**. The address value counted by the PCI address counter is output to the PCI bus **215**. The graphic control circuit **220** writes the image data in the predetermined storage area of the graphic memory **221** by using the address value.

With the above described hardware arrangement, the circuit elements of the converter circuit **219** and the PCI bridge circuit **216B** are initialized each time an image data of a frame is transferred so that it is possible to automatically write image data in the predetermined respective storage areas without resorting to software.

Then, the image data are read out from the storage areas storing the image data by means of the graphic control circuit **220** and subjected to necessary processing operations such as D/A conversion within the graphic control circuit **220** before they are output to the multi-scan monitor **30** by way of the cable **50**. As a result, the image taken by the IEEE 1394 camera can be displayed automatically at the frame rate of the selected mode.

Since conversion-related processing operations are conducted in this way by means of hardware according to the invention, it is now possible to carry out the operations at high speed if compared with processing operations that are carried out by the CPU **211** using software. Therefore, the present invention easily accommodates high speed data streams and provides an improved processing capacity.

Thus, the digital camera controller **10** realized by applying the present invention converts the data stream input from the digital camera **20** by IEEE 1394 isochronous transfer of the converter circuit **219** into image data frames to be used for PCI transmission and the image data frames are then

transferred to the graphic control circuit 220 by DMA transfer of the PCI bridge circuit 216B. With this arrangement, the processing operation for transmitting image data is conducted at high speed and the entire system is made to show a high processing capacity. In other words, the digital camera controller 10 realized by applying the present invention uses the PCI bus 215 as extension bus and controls the DMA transmission by using the PCI bridge circuit 216 as bus master. Once a DMA transmission is started, the CPU 211 is released from the operation of transmitting image data and hence can perform other processing operations by way of the CPU bus 212. As a result, the load of the CPU 211 is reduced to improve the efficiency of operation of the entire system.

Now, the operation of the graphic control circuit 220 of the digital camera controller 10 realized by applying the present invention will be discussed below in detail.

Note that the first, second and third frame storage areas of the graphic memory 221 are provided respectively with frame numbers 1, 2 and 3.

FIG. 5 is a flow chart of the start of an image data writing operation of the CPU 211. Firstly, in Step S1, the CPU 211 initializes the read frame register and the write frame register of the system RAM 214 by storing frame number 1 there.

Then, the CPU 211 selects the memory start address that corresponds to the write frame for the PCI bridge circuit 216B by way of the CPU bus 212, the PCI bridge circuit 216A and the PCI bus 215 (Step S2). Thereafter, the CPU 211 selects the memory start address that corresponds to the read frame for the graphic control circuit 220 by way of the CPU bus 212, the PCI bridge 216A and the PCI bus 215 (Step S3). Note that the memory start address indicates the leading address of each frame storage area (the first frame storage area here) of the graphic memory 221.

Subsequently, the CPU 211 initializes (enables) the converter circuit 219 and the link control circuit 218 by way of the CPU bus 212 for the purpose of writing data and, at the same time, it initializes the PCI bridge circuit 216B by way of the CPU bus 212 and the converter circuit 219, and also the graphic control circuit 220 by way of the CPU bus 212, the PCI bridge circuit 216A and the PCI bus 215 (Step S4). Thereafter, the CPU 211 permits (enables) isochronous data transfer of the digital camera 20 (Step S5). At this time, the number of data of a frame is set in the write data counter of the PCI bridge circuit 216B and the counter value is decremented (-1) thereafter each time an image data is transferred. Detection signal #INT1 is output to the CPU 211 when the counter value becomes equal to 0. With this arrangement, the PCI bridge circuit 216B operates as bus master independently from the CPU 211.

FIG. 6 is a flow chart of an interrupt processing operation of the CPU 211 when it detects detection signal #INT1.

Referring to FIG. 6, in this interrupt processing operation, the CPU 211 firstly adds 1 to the write frame number (Step S11) and finds out the write frame number after the addition (Step S12). If the CPU 211 finds that the write frame number is equal to 2, it reduce the read frame number to 1 (Step S13), whereas it makes the read frame number equal to 2 if the write frame number is equal to 3 (Step S14) but makes the read frame number equal to 3 and reduces the write frame number to 1 if the write frame number is equal to 4 (Step S15). Then, in Step S16, the CPU 211 provides the PCI bridge circuit 216B with the memory start address that corresponds to the read frame number after the addition or the write frame number reduced to 1 and moves out of the

interrupt processing operation. As a result, the write frame storage area is switched and an operation of writing image data in a new area starts.

FIG. 7 is a flow chart of an interrupt processing operation of the CPU 211 when it detects detection signal #INT2.

Referring to FIG. 7, in this interrupt processing operation, the CPU 211 firstly enables the graphic control circuit 220 to perform a read operation by way of the CPU bus 212, the PCI bridge circuit 216A and the PCI bus 215 (Step S21). Then, the CPU 211 provides the graphic control circuit 220 with the memory start address that corresponds to the read frame number by way of the CPU bus 212, the PCI bridge circuit 216A and the PCI bus 215 (Step S22) to start an operation of reading data from the graphic memory 221.

As shown in FIG. 6, the digital controller 10 realized by applying the present invention appropriately switches the write frame storage area according to the read frame number and conducts a processing operation in the sequence shown in FIG. 7 so that a frame storage area that is not being used for any write operation is selected as frame storage area to be used for reading image data. Thus, generation of outrun scanning noise is prevented from taking place.

Additionally, the workload of the CPU 211 can be alleviated because the PCI bridge circuit 216B can carry out the transfer processing operation for which the CPU 211 is normally responsible as it takes the role of bus master.

It will be appreciated that the scope of application of the present invention is not limited to the above described embodiment of digital camera controller 10.

While the digital camera controller is adapted to transfer the image data input from a digital camera to a multi-scan monitor in the above description, the present invention is applicable to various data transfer devices adapted to transfer the image data input from an image processing device to an electronic device.

While storage areas for three frames are provided in the graphic memory in the above description (if M frame storage areas are provided and image data is written in the N-th frame, M=3 and N=1, 2, 3 in the above description) in the above description, it is also possible to control in a similar manner if storage areas for two frames or four or more than four frames are provided.

While the PCI bridge circuit 216B is provided with the read address control function for image data stored in the graphic memory 221 in the above description, the graphic control circuit 220 may alternatively be provided with such a function.

While a multi-scan monitor 30 is used as monitoring apparatus for displaying images in the above description, it may be replaced by some other monitoring apparatus or an electronic device that can change the resolution (image size) according to the external input.

Finally, while a PCI bus is used as extension bus and made to operate as bus master for the purpose of controlling DMA transmissions in the above description, a bus other than PCI bus may be selected as extension bus.

What is claimed is:

1. A data transfer device for transferring the image data input from an image processing device to an electronic device, said device comprising:

- a conversion processing section for performing a predetermined conversion processing operation on the image data input from said image processing device;
- a graphic memory for temporarily storing the image data;
- a graphic control circuit for writing the image data transmitted from said conversion processing section

11

into said graphic memory or reading the image data stored in said graphic memory and transmitting it to said electronic device; and

- a CPU for selecting a first data transmission route for transmitting the image data from said conversion processing section to said graphic control circuit or a second data transmission route for transmitting a control signal to said graphic control circuit; said conversion processing section being adapted to convert the image data input from said image processing device into an output format,

wherein said CPU is provided with a CPU bus for controlling the component sections, an extension bus for connecting itself to peripheral elements including said graphic control circuit and a bus bridge circuit for controlling said extension bus; and said bus bridge circuit controls said extension bus as bus master and controls DMA transmissions of transmitting image data from said conversion processing section to said graphic control circuit.

2. The device according to claim 1, wherein said extension bus is a PCI bus and said bus bridge circuit is a PCI bridge circuit.

3. The device according to claim 1, wherein said bus bridge circuit operating as bus master for controlling DMA transmissions includes an interface section for controlling the interface between the converter circuit and the extension bus and an address management section for managing the address of the image data being transmitted.

4. A data transfer device for transferring the image data input from an image processing device to an electronic device, said device comprising:

a conversion processing section for performing a predetermined conversion processing operation on the image data input from said image processing device;

a graphic memory for temporarily storing the image data; a graphic control circuit for writing the image data transmitted from said conversion processing section into said graphic memory or reading the image data stored in said graphic memory and transmitting it to said electronic device; and

a CPU for selecting a first data transmission route for transmitting the image data from said conversion processing section to said graphic control circuit or a second data transmission route for transmitting a control signal to said graphic control circuit; said conversion processing section being adapted to convert the image data input from said image processing device into an output format.

wherein said conversion processing section includes a color signal format conversion section for converting the format of the image data transferred from said image processing device into a color signal format suited for displaying images and a transfer format conversion section for converting the format of the data converted by said color signal format conversion section further into a data format suited for data transfer, and

wherein said color signal format conversion section includes a plurality of conversion processing sections adapted to perform different respective conversion processing operations corresponding to the color signal formats of image processing devices and a selection section for selecting one of said plurality of conversion processing sections according to the color signal format of said image processing device and outputting the

12

output data of said selected conversion processing section to said transfer format conversion section.

5. A data transfer device for transferring the image data input from an image processing device to an electronic device, said device comprising:

a conversion processing section for performing a predetermined conversion processing operation on the image data input from said image processing device;

a graphic memory for temporarily storing the image data; a graphic control circuit for writing the image data transmitted from said conversion processing section into said graphic memory or reading the image data stored in said graphic memory and transmitting it to said electronic device; and

a CPU for selecting a first data transmission route for transmitting the image data from said conversion processing section to said graphic control circuit or a second data transmission route for transmitting a control signal to said graphic control circuit; said conversion processing section being adapted to convert the image data input from said image processing device into an output format,

wherein said conversion processing section includes a color signal format conversion section for converting the format of the image data transferred from said image processing device into a color signal format suited for displaying images and a transfer format conversion section for converting the format of the data converted by said color signal format conversion section further into a data format suited for data transfer, and

wherein a buffer section is provided between said color signal format conversion section and said transfer format conversion section in order to absorb the difference of the transfer rate of the image processing device and that of the graphic control circuit.

6. A data transfer device for transferring the image data input from an image processing device to an electronic device, said device comprising:

a conversion processing section for performing a predetermined conversion processing operation on the image data input from said image processing device;

a graphic memory for temporarily storing the image data; a graphic control circuit for writing the image data transmitted from said conversion processing section into said graphic memory or reading the image data stored in said graphic memory and transmitting it to said electronic device; and

a CPU for selecting a first data transmission route for transmitting the image data from said conversion processing section to said graphic control circuit or a second data transmission route for transmitting a control signal to said graphic control circuit; said conversion processing section being adapted to convert the image data input from said image processing device into an output format,

wherein said graphic memory has image data storage areas for a plurality of frames and said graphic control circuit sequentially selects the image data storage areas for the plurality of frames and controls image data write operations and image data read operations on a time division basis.

7. The device according to claim 6, wherein said CPU judges the storage area where an image data is being written and determines the storage area to be used for reading an image data on the basis of its judgment.

13

8. The device according to claim 7, wherein said CPU detects that the image data of a frame has been written in said graphic memory and that the image data of a frame has been read out from said graphic memory and switches the storage area for writing image data and the storage area for reading image data according to the detection signals from said write timing detection circuit and said read timing detection circuit.

9. The device according to claim 8, wherein said graphic memory has M frame storage areas, each being adapted to store the image data of a frame, and said graphic control circuit switches to the N+1-th frame storage area to write the

14

image data of the N+1-th frame after writing the image data in the storage area for the N-th frame ($N \leq M$, N, M: positive integer) and also switches to the N-th frame storage area to read the image data of the N-th frame after reading the image data in the storage area for the N-1-th frame.

10. The device according to claim 6, wherein said image data are real time moving image data.

11. The device according to claim 6, wherein said image data are adapted to be converted into a data stream of a predetermined format before being transmitted.

* * * * *