



ATARI France

Manuel de Référence Atari

TT030

Juin 90

ADDENDUM

Veillez trouver ci-joint les quelques corrections à apporter au document:

- page 3** – 2ème ligne de 2ème § : "*dans*" au lieu de "sur"
- page 3** – 3ème ligne de 3ème § : "*carte VME simple Europe*" au lieu de "carte Eurocarte VME"
- page 4** – ligne 14 : "*prioritaire*" au lieu de "propriétaire"
- page 5** – 1ère ligne de §II.1 : "*Motorola MC68030*" au lieu de "Motorola MC680302"
- page 6** – 2ème ligne de §II.4 : "*On utilise 16 DRAM 256Kbit x 4100 nS*" au lieu de "DRAM 16256Kbitx4100nS..."
- page 13** – 4ème ligne de §III.3.1 : remplacer "*carter*" par "*cache métallique*"
- page 14** – 2ème ligne de § III.4.2 : supprimer "*unbuffered*"
- page 15** – 2ème ligne de §III.7 : remplacer "bord" par "*port cartouche*"
- page 16** – 1er § 2ème ligne : remplacer "la palette est comprise dans ..."
par "*la palette est déterminée par des ..*"
- page 16** – 2ème § : "*inverseur*" au lieu de "inversseur"
- page 18** – 1ère ligne de § III.9.1 : "*fait partie*" au lieu de "fait parti"
- page 19** – 1ère ligne de § IV.3 : "*une adrese*" au lieu de "un adresse"
- page 27** – 8ème ligne : "*HSYNC*" au lieu de "HSUNC"
- page 37** – 2ème ligne du 6ème§ : "*nécessaire*" au lieu de "nécessaire"
- page 34** – 1ère ligne du 6ème § : supprimer "*paresseuses*" !!!
- page 38** – § 4 et 5 : "*Malloc*" au lieu de "malloc"
- page 41** – 2ème § : "*Atari Corporation*" au lieu de "Atari"
- page 43** – dernier § : "*Rskonf*" au lieu de "Rsonf"
-
-

Atari Corporation
Manuel de Référence Atari TT030

Juin 1990

Bibliographie

Bus VME :

VMEbus International Trade Association (VITA), *VMEbus Specification Manual*, Revision C.1, October, 1985.

Small Computer Systems Interface (SCSI) :

Adaptive Data Systems, Inc., *SCSI Guidebook*, Issue Number 2, June, 1985.

Spécificités des implantations de périphériques SCSI :

Adapted Inc., *Description of SCSI Command Set for Communications Devices*, Revision 0.91, 1988.

Archive Corporation, *VIPER Product Manual*, SCSI Models 2060S and 2150S, Part No. 21391-001, June, 1988.

Maxtor Corporation, *XT-4000S QEM Manual & Product Specification*, 1014995, 1987.

Quantum Corporation, *Q200 Series Programmer's Manual*, 81-45416, Rev. B, 1987.

Spécificités des principaux composants employés dans le TT :

General Instrument Corp., *AY-3-8910/8912 Programmable Sound Generator Data Manual*, February 1979.

Logic Devices Inc., *L5380/L53C80 CMOS SCSI Bus Controllers*, September 1988.

Motorola, Inc. *MC68030 Enhanced 32-Bit Microprocessor User's Manual*, 2nd. Edition 1989

Motorola, Inc., *MC68881/MC68882 Floating-Point Coprocessor User's Manual*, First Edition, 1987.

Motorola, Inc., *MC68901 Multi-Fonction Peripheral*, January, 1984.

Motorola, Inc., *MC 146818A Real-Time Clock Plus RAM (RTC)*, 1984

Motorola, Inc., *MC 6850 Asynchronous Communications Interface Adapter*.

Western Digital Corp., *WD1772-02 Floppy Disk Formatter/Controller*

MANUEL DEVELOPPEUR

Zilog, Inc., Z80C30 CMOS Z-BUS SCC / Z85C30 CMOS SCC Serial Communications Controller - Preliminary Product Specification, October, 1987.

Zilog, Inc., Z80C30 CMOS Z-BUS SCC / Z85C30 CMOS SCC Serial Communications Controller - Technical Manual, September, 1986.

Les interfaces compatibles ST sont aussi décrites dans :

Atari Corporation, *Engineering Hardware Specification of the Atari ST Computer System*, January 7, 1986.

Atari Corporation, *STE Developer Addendum*

Atari Corporation, *Intelligent Keyboard (ikbd) Protocol*, February 26, 1985.

Atari Corporation, *[ST] DMA Controller*, September 26, 1984.

I. INTRODUCTION

Le TT (*Trente-deux/Trente-deux bits*) est le premier membre d'une nouvelle série d'ordinateurs Atari, version améliorée de la famille des ST et MEGA. Le TT conserve la compatibilité avec l'architecture MEGA/ST, mais est structuré autour du microprocesseur Motorola 68030. Il permet un affichage graphique d'une qualité encore meilleure et offre des caractéristiques sonores plus élaborées. Le TT est conçu pour pouvoir exécuter UNIX (1) sans aucune pénalité de vitesse provenant des limitations de compatibilité du ST.

Le TT est construit autour du processeur haute performance Motorola MC68030, 32 bits, cadencé à 16 MHz. Le 68030 comporte un cache données et un cache instructions sur le composant.

L'architecture comprend également un bus VME au standard de l'industrie, facilitant les extensions. Le système supporte la dernière révision (C.1) des spécifications du bus VME. Le TT peut recevoir une carte Eurocarte VME.

Le TT est conçu pour fonctionner en réseau avec d'autres TT comme en environnement hétérogène. Pour faciliter la connectivité, chaque système possède un port interne permettant la connexion à un réseau local à basse vitesse. Si la connexion réseau local n'est pas utilisée, le port peut être redéfini comme port RS232C. En option, grâce à un contrôleur Ethernet à base SCSI ou bus VME, le TT se connecte à des réseaux Ethernet hétérogènes. En outre, chaque TT a trois ports série RS-232C qui permettent la connexion à des modems, à des terminaux ou à des tables à numériser.

Le TT est destiné à être utilisé sous TOS ou sous UNIX. La version UNIX proposée est basée sur l'UniPlus+ V Release 3 version 1 d'Unisoft qui est compatible avec le System V Interface Definition (SVID) d'AT&T, le Portable Operating System Interface Specification (POSIX) et le X/Open Portability Guide. Le X Window System, réseau transparent à fenêtrage, développé tout d'abord par le MIT, est aussi disponible dans la version initiale. L'interface utilisateur conviviale (fenêtrage) tourne sous X Window.

Les caractéristiques matérielles du TT sont les suivantes :

- Microprocesseur Motorola MC68030 à 16MHz
- Co-processeur à virgule flottante Motorola MC68881/68882
- 2 Mo de RAM double fonction (système/vidéo), extensible par une carte additionnelle de 2 ou 8 Mo supplémentaires. Cette mémoire se décompose en 64 bits pour la logique vidéo et 32 bits pour le reste du système. La logique vidéo du TT réclame un accès à cette mémoire sur une base de temps critique. La logique du système restante (y compris le processeur) a accès à cette mémoire par tranches de 250 nS.
- RAM : 4 Mo (nybble mode) - extensible par une carte additionnelle de 4 ou 16 Mo supplémentaires.

(1) Unix est une marque déposée de AT&T.

- ROM : 4 supports ROM d'1 Mbit, fournissant 512 ko d'espace ROM. Le bus étant un bus 32 bits, les quatre ROM doivent être présentes.

- Modes vidéo internes (versions améliorées des modes vidéo de l'Atari ST.)

Couleur : 320x200x16, 320x480x256, 640x200x4, 640x480x16.

DuoChrome : 640x400.

Monochrome :1280x960.

- Une interface RVB couleur analogique (pour les modes couleur et DuoChrome)

- Une interface moniteur ECL haute-performance (haute-résolution monochrome)

- Un port parallèle E/S (implémenté en utilisant le port parallèle du processeur sonore General Instruments AY-3-8910 / Yamaha YM-2149)

- Un haut-parleur interne qui peut être désactivé directement par le programme.

- 2 ports E/S asynchrones (un sur chaque MFP 68901)

- 2 ports série SDLC haute performance (Zilog 8530 SCC), dont l'un peut être configuré comme interface réseau local comportant un contrôleur DMA simple canal, propriétaire.

- Horloge Temps Réel (RTC) avec 50 octets de RAM non volatile.

- Clavier intelligent, compatible MEGA/ST, comportant un port souris et un port manette de commande (supportant les souris 3 boutons)

- Un canal Atari ACSI DMA (pour disque dur Atari, imprimante laser, CD-ROM, etc.)

- Une interface et un contrôleur de disquette partageant le canal ACSI DMA

- Une interface MIDI

- Port cartouche compatible Atari ST (128 ko)

- Une interface SCSI 25 broches (implémentée sur le contrôleur SCSI NCR 5380 et le contrôleur DMA propriétaire).

- Bus d'extension VME : le TT comporte une Eurocarte A24/D16 et une interface A16/D16 asservie

II. SYSTEME PRINCIPAL

Son architecture fait du TT une plate-forme hautement performante. Grâce au bus VME et à ses caractéristiques (multi-tâche, etc...), le système peut se développer en fonction des besoins futurs.

II.1 Processeur et unité de gestion de mémoire (MMU)

Le TT est construit autour du microprocesseur Motorola MC68030 à 32 bits. Ce mono-composant contient un processeur 68020 amélioré, une unité de gestion de mémoire paginée, un cache données et un cache instructions. Le 68030 est un composant CISC qui étend les instructions du 68000 et améliore les modes d'adressage. Le processeur est cadencé à 16 MHz.

L'unité de gestion de mémoire du 68030 est un sous-ensemble du Motorola MC68851. Le buffer de translation (TLB) a été réduit à 22 entrées, exigeant un soin spécial dans l'assignation mémoire, mais permettant d'éviter des nettoyages de processeur non nécessaire.

Les caches données et les caches instructions embarquées maximalisent le débit du processeur et réduisent en même temps la largeur de bande du bus nécessaire pour alimenter le processeur.

II.2 Co-processeur à virgule flottante

Le TT comporte un co-processeur à virgule flottante sur support Motorola MC68881. Le MC68881 peut être remplacé par un compatible MC68882. Il y a une légère différence au niveau logiciel en ce qui concerne la pile de gestion des exceptions, mais les programmes peuvent être écrits de manière à permettre l'utilisation transparente de l'un ou de l'autre.

Les opérations à virgule flottante sont exécutées en conformité avec le standard IEEE 754 avec des accès externes à simple (32 bits) ou double (64 bits) précision.

Le co-processeur à virgule flottante est cadencé à la même fréquence que le processeur principal. Il se présente comme le co-processeur à virgule flottante ID "standard" dans l'espace d'adressage de l'unité centrale 68030.

III.3 ROM

Ce système comprend un ensemble de quatre supports permettant d'accommoder des ROM de 1Mbit, pour un total de 512 ko d'espace ROM. Le bus d'accès système étant sur 32 bits, les quatre ROM doivent être présentes. Des cavaliers permettent l'utilisation des EPROM 27256, 27512, 27010/27C1001 et 57101/27C1000, en plus des ROM 53100. Les réglages par défaut des cavaliers permettent l'utilisation des EPROM 27512 (pour un total de 256 ko d'espace ROM) ainsi que celle des EPROM 57101/27C1000 ou des ROM 531000 (pour un total de 512 ko de ROM). Des prises 32 broches sont fournies, bien que les EPROM 27256, 27512 et les ROM 53100 n'utilisent que la partie inférieure des 28 broches.

Une copie des 8 premiers octets de la ROM réside dans les 8 premiers octets de la ROM allouée à la compatibilité ST. Ces 8 premiers octets (0x00000000-0x00000007, ou 0xFF000000-0xFF000007 sur l'image) ne sont accessibles *qu'en* mode superviseur.

Toute tentative pour lire cette zone dans un mode utilisateur, ou toute écriture, se solde par une erreur de bus. Le bus VME maître doit créer des accès privilégiés pour lire la ROM à ces endroits. L'intégralité de la ROM se trouve en 0x00E00000 - 0x00EFFFFF (avec une copie sur 0xFFE00000 - 0xFFEFFFFF).

Cette ROM permet, entre autres, d'initialiser le système, d'effectuer les tests de mise en marche, ainsi que les démarrages à froid à partir du lecteur de disquette, d'une unité ACSI ou d'une unité SCSI.

II.4 RAM

Le système de base comprend 2 Mo de RAM double-fonction qui sont utilisés pour la mémoire vidéo et la mémoire système. On utilise des DRAM 16256Kbitx4100 nS, permettant d'obtenir un large bus interne de 64 bits et des accès vidéo haute performance.

L'architecture du bus est similaire à celle du ST dans la mesure où les cycles d'accès à la mémoire sont répartis entre le MPU et le contrôleur vidéo par tranches de 250 nS, permettant à la mémoire vidéo de résider en tant que partie de la mémoire principale. Pendant les cycles de visualisation active, il n'est pas possible pour le processeur d'accéder à la mémoire ; cependant, on lui affecte la prochaine tranche de 250 nS. L'interfaçage du processeur et de cette RAM s'effectue par un bus 32 bits, mais le sous-système vidéo accède lui-même à la mémoire grâce à un bus 64 bits. Le composant vidéo (shifter du TT) embarque une mémoire tampon qui assure une très grande largeur de bandes pour les données.

Des cartes RAM additionnelles spécialisées sont également disponibles en option. En éliminant les contraintes de synchronisation vidéo sur cette RAM, elle peut être conçue de façon à paraître plus rapide au processeur. Les cartes additionnelles sont généralement utilisées avec 32 DRAM de 1 Mbit à 100 nS. Lorsque des DRAM de 4Mbit seront disponibles, il sera possible d'obtenir 16 Mo de RAM spécialisée sur une seule carte additionnelle.

Une mémoire supplémentaire peut être installée en connectant une carte mémoire VME. La RAM VME est légèrement plus lente que la RAM système puisque les accès VME nécessitent une attente supplémentaire par cycle de bus.

Les premiers 0x800 octets (2Ko) de RAM (0x00000008-0x000007FF, ou 0xFF000008-0xFF0007FF sur la copie) ne sont accessibles qu'en mode superviseur. Toute tentative pour lire cette zone ou d'y écrire se solde par une erreur de bus.

II.5 Unité de contrôle du système (SCU)

L'**unité de contrôle du système** (SCU) fournit un niveau d'interruption supplémentaire au système. Elle contient également des registres qui permettent la création logicielle d'interruptions. Tous les registres SCU sont réinitialisés lors de la mise sous tension ou lors d'une action sur le bouton de réinitialisation.

II.5.1 Masque d'interruption et état en cours

Le SCU contient deux registres masques qui permettent de contrôler indépendamment les niveaux d'interruption pouvant être vus par le processeur. Un registre masque les interruptions créées sur la carte système et l'autre masque les sources du bus VME. Ces registres sont réinitialisés lors de la mise sous tension ou du reset, supprimant toutes les interruptions.

Il comporte également des registres de demande d'interruption qui montrent l'état en cours des sept niveaux d'interruption demandés depuis chacune des sources. Ce registre montre le statut physique des interruptions avant qu'elles ne soient combinées dans une porte ET avec le registre masque du SCU.

Les sources de la carte mère pour IRQ5 et IRQ6 peuvent être prises en charge par le 68030 ou par le bus VME maître. L'implantation utilisée spécifie que pour le 68030, IRQ5 et IRQ6 ont la même fonction que les interruptions VME et ne peuvent pas de ce fait être masqués par le registre masquant les interruptions de la carte mère SCU.

II.5.2 Registres de contrôle du système

Le SCU contient également deux registres lecture/écriture qui peuvent être utilisés pour stocker des informations sur la configuration du système.

II.5.3 Générateur d'interruption

Le système peut écrire sur une adresse E/S afin de générer une interruption à priorité basse (niveau 1) pour le 68030. Cette adresse E/S comporte un registre d'état en lecture/écriture, et seulement le bit de poids faible de l'octet de poids faible est configuré. A "1", une interruption auto-vectorisée de niveau 1 est générée. A "0", la demande d'interruption est supprimée.

Le SCU est câblé de façon à ce que:

- seules les interruptions 5 et 6 aient des broches IACK externes et soient capables de générer des interruptions vectorisées sur la carte mère (et, respectivement, IRQ5 et IRQ6 VME).
- IRQ1 et IRQ3 générés par le SCU soient câblés pour les priorités correspondantes et soient toujours auto-vectorisés.
- IRQ1 généré par le SCU ne soit détecté que par le MPU et non par le bus VME
- Le SYSFAIL du bus VME génère un système (carte-mère) IRQ7 vers le MPU, mais ne génère pas IRQ7 vers le bus VME. La seule autre source IRQ7 est une carte de bus VME.

II.5.4 Bus Timer

Le SCU comporte également un système de bus timer. Si le cycle de bus n'est pas conclu dans les 16 microsecondes, le SCU signale une erreur de bus.

II.6 Contrôleurs DMA

Le TT possède trois canaux DMA indépendants : 1) le port réseau à basse vitesse (implémenté par le port série SCC A), 2) le port SCSI et 3) le canal DMA du lecteur de disquette "ACSI" ST. Les niveaux de priorités du DMA sont les suivantes :

<u>Priorité</u>	<u>Fonction</u>
haute	Contrôleur du lecteur de disquette ACSI Canal DMA SCC Canal DMA SCSI
basse	68030

II.6.1 Canaux DMA SCC et SCSI

Les contrôleurs DMA SCC et SCSI assemblent les octets depuis le périphérique par doubles mots pour écrire sur le bus système. Ce dispositif est effectivement mis en place avec deux groupes indépendants de deux mots. De cette façon, lorsque l'un d'eux, déjà rempli, est en attente pour accéder au bus du processeur, l'autre peut être rempli. Si le second groupe de mots est rempli avant que le bus ne soit libéré par le DMA, il est écrit dans la même opération de bus.

On peut envoyer sur le DMA n'importe quel octet, d'où qu'il vienne dans l'espace mémoire. L'accès direct à la mémoire est exécuté dans l'espace adresse physique.

Le modèle de programmation de chacun de ces canaux DMA est composé des éléments suivants :

- un registre d'état sur 16 bits en lecture/écriture qui contient les bits de direction, de validation et d'erreurs de bus.
- quatre octets qui forment un pointeur DMA 32 bits.
- un registre d'entrées partielles qui doit être lu et fusionné avec le contenu de la RAM sous le contrôle du CPU si l'entrée DMA est faite à partir d'un point de l'espace mémoire qui n'est pas dans les 32 bits ou si l'opération DMA n'est pas effectuée en multiple de quatre octets.
- compteur d'octet du DMA de 32 bits (implémenté en quatre octets).

Il y a un contrôleur DMA pour le canal SCC et un pour le canal SCSI. Physiquement, les contrôleurs DMA se trouvent sur deux composants : l'un pour l'interface bus système, et l'autre pour l'interface périphérique - FIFO. Le contrôleur de l'interface bus est connecté extérieurement tant pour le SCSI que pour le SCC.

Le logiciel qui initialise le contrôleur DMA (DMAC) pour les transferts de type DMA doit tenir compte du fait que le DMAC est un périphérique 16 bits qui apparaît sur les octets d'imparité du bus d'adresses. Ainsi soit le 68030 utilise les instructions MOVEP, soit il effectue des permutations circulaires et quatre opérations différentes de sorties d'octets pour extraire un nombre d'octets ou d'adresses à 32 bits.

Registres du contrôleur DMA

Décalage	Largeur	Fonction
0x00	OB	DMA Pointer Upper
0x02	OB	DMA Pointer Upper-Middle
0x04	OB	DMA Pointer Lower-Middle
0x06	OB	DMA Pointer Lower
0x08	OB	Byte Count Upper
0x0A	OB	Byte Count Upper-Middle
0x0C	OB	Byte Count Lower-Middle
0x0E	OB	Byte Count Lower
0x10	W	Data Residue Register High
0x12	W	Data Residue Register Low
0x14	OB	Control Register

Le mot de contrôle est un registre bit-mapped :

Bit	Fonction
0	DMA Direction Out (1 = vers le port)
1	Validé (0 = off, 1 = on)
2-5	<réservé>
6	Byte Count Zero (1 = comptage définitif)
7	Bus Error (1 = erreur de bus durant une transmission de type DMA - sur ce canal)

Pour effectuer une transmission de type DMA :

- 1) initialiser la direction du contrôleur DMA
- 2) initialiser l'adresse de base

- 3) initialiser le périphérique
- 4) initialiser le bit de validation

Les bits de direction et de validation ne doivent pas être établis lors de la même opération.

En DMA, si une entrée/lecture est effectuée sur autre chose qu'un mot double ou si la longueur n'est pas un multiple de 4, les derniers octets ne sont pas écrits dans la RAM système. Le programmeur doit donc lire le *data residue register* et fusionner l'entrée avec le contenu du groupe de deux mots approprié en RAM. (Les deux bits de poids faible du pointeur DMA sont correctement incrémentés, et peuvent être utilisés pour déterminer la portion de Residue Register correcte)

Les transmissions de type DMA peuvent seulement être effectuées sur des ports 32 bits, comme les RAM et les cartes VME D32.

Si une tentative d'opération de type DMA génère une erreur de bus, l'opération DMA est immédiatement invalidée et le bit d'erreur du bus est configuré dans le registre d'état. Les bits d'état d'erreur de bus de chacun des contrôleurs DMA sont transmis vers les bits d'entrée MFP-2 individuels où ils peuvent être lus ou utilisés pour générer une interruption (option). L'état d'erreur de bus pour un canal est automatiquement effacé lors de la lecture du registre de contrôle du canal.

Le registre du compteur d'octets du DMA génère une interruption lorsque le nombre d'octets atteint 0. Le DMA est automatiquement effacé dès qu'il atteint le chiffre défini comme comptage définitif.

L'interface SCSI NCR 5380 ne doit pas être utilisée en mode BLOCKMODE DMA destiné à être utilisé avec les contrôleurs DMA du TT. Le SCC doit être programmé pour utiliser la broche WAIT/*REQ en mode *REQ lors des opérations DMA.

II 6.2 Interface lecteur de disquette/ACSI

Le sous-système lecteur de disquette/ACSI compatible ST s'interface avec la RAM double-fonction et les périphériques compatibles ACSI tels que l'imprimante laser SLM804, les unités de disques durs SHxxx/Megafile, et le CD-ROM Atari. Ce canal DAM est partagé avec le contrôleur de lecteur de disquette interne.

Les opérations DMA entre la RAM et les périphériques ACSI, et entre la RAM et le lecteur de disquette, peuvent seulement être exécutés en utilisant la RAM double fonction. Si un transfert est tenté depuis l'un de ces périphériques vers une RAM système "standard", un transfert à deux phases est nécessaire, et la RAM double fonction est utilisée comme mémoire tampon.

II.7 Horloge temps réel

Le TT comprend une horloge temps réel Motorola MC146818A. Elle donne le jour et l'heure (à la seconde près), et une interruption périodique programmable. L'horloge temps réel est cadencée à 32,768 kHz. Elle est indépendante des autres horloges système.

Les interruptions générées par l'horloge sont transmises par l'une des entrées parallèles du MFP (périphérique multi-fonction).

Le composant comporte une RAM (non volatile) de 50 octets sauvegardés par pile qui est utilisée pour stocker les données de diagnostic et de configuration.

Le composant est sollicité par deux registres sur 16 bits. Les premiers 8 bits sont en écriture seulement, ils permettent de définir l'adresse désirée de l'horloge. Les 8 bits suivants sont en lecture/écriture. Lorsque l'on écrit sur un registre de l'horloge, on peut le faire sur 16 bits ; les 8 premiers bits définissent l'adresse, et les 8 derniers les données.

III. GESTION DES PERIPHERIQUES

L'architecture du TT supporte les sous-systèmes suivants :

- SCSI
- ACSI compatible ST
- des ports série haute vitesse et un port pour réseau à basse vitesse (implémenté sur le composant SCC)
- deux ports série supplémentaires et un port d'interruption externe connecté aux contrôleurs MFP
- un port parallèle (Centronics) implémenté sur le composant sonore Yamaha YM-2149
- une interface clavier, souris et manette de commande compatibles ST/MEGA
- un port supportant les cartouches d'application et de diagnostic

III.1 SCSI

Le TT implémente le bus SCSI complet en utilisant un contrôleur SCSI NCR5380. Le NCR5380 est utilisé lors des transferts de données asynchrones 8 bits jusqu'à 4.0 Mo/seconde.

Le connecteur SCSI assure la connexion des périphériques compatibles SCSI grâce à un connecteur D 25 broches. A l'intérieur, les 50 broches sont utilisées.

Généralement, le bus SCSI est utilisé pour les unités de mémoire de masse du système. Mais il peut également être utilisé pour des périphériques de stockage amovibles tels que les lecteurs de cartouches Syquest et les dérouleurs de bandes magnétiques. Par défaut, le disque dur du système est paramétré comme unité 0, périphérique 0.

Le bus SCSI peut supporter jusqu'à 7 périphériques (en plus du TT lui-même).

III.2 ACSI

L'interface ACSI du TT est identique à celle du ST. Le plus grand problème est celui de l'utilisation des boucles de temps pour parler simultanément avec le contrôleur et les périphériques. Nous recommandons aux développeurs d'utiliser le Timer A du MFP.

III.3 Ports série haute vitesse

Le TT comprend un Zilog 85C30 SCC, périphérique de transmission de données, multiprotocole à double canal, qui offre deux ports série (ports A et B).

Le port A peut être utilisé soit en tant que port réseau, soit en tant que port RS232C basse vitesse. Le mode réseau local est sélectionné, lorsque le bit 7 du port A du composant sonore GI est à 0. Les E/S du port A sont acheminées vers le connecteur approprié : (1) si le mode RS232C est sélectionné, le port est connecté sur une DB-9 broches ou bien (2) si le port réseau est sélectionné, il est connecté à une mini-DIN 8 broches. Les broches de sortie sur le port non sélectionné demeurent inactives.

Le SCC traite les formats asynchrones et les protocoles orientés octet synchrones tels que HDLC et SDLC (IBM).

Le port B est un port série RS232C basse vitesse qui peut être utilisé pour les connexions modem ou mainframe. Il est raccordé à un connecteur DB-9 et est compatible AT. Les signaux de contrôle du modem sont dérivés directement depuis les lignes de contrôle du port B 85C30. Ce port peut fonctionner à des vitesses (élevées) de transmission et d'émission différentes.

L'entrée PCLK du SCC est cadencée à 8 MHz. L'entrée RTxCA a une horloge à 3,672 MHz. L'entrée du TRxCA provient du connecteur réseau local à basse vitesse. RTxCB est à 2,4576 MHz. TRxCB est généré par la sortie de l'horloge C du second MFP (TT).

III.3.1 Brochage du port SCC RS232

Les ports série SCC RS232 sont branchés sur des connecteurs DB-9 broches de manière à assurer la compatibilité AT. Sur le TT, les connexions RS232 du port A SCC sont acheminées vers une barrette sur la carte mère. Cette barrette peut être connectée par un câble-ruban à un connecteur D à neuf broches situé sur le carter du slot VME.

Brochage du port SCC RS232

Broche	Port A	Port B
	(Mode RS232)	
1	(CD) Détection de porteuse (E)	Détection de porteuse (E)
2	(RD) Donnée réceptrice (E)	Donnée réceptrice (E)
3	(TD) Donnée de transmission (S)	Donnée de transmission (S)
4	(DTR) Terminal de données prêt (S)	Terminal de données prêt (S)
5	Masse	Masse
6	(DSR) Modem prêt (E)	Modem prêt (E)
7	(RTS) Demande pour émettre (S)	Demande pour émettre (S)
8	(CTS) Prêt à émettre (E)	Prêt à émettre (E)
9	--	(RI) Indicateur d'appel (E)

Remarque : Le signal de l'indicateur d'appel (RI) du port B SCC est connecté au bit 6 du port E/S universel MFP-2 (GPIP).

III.3.2 Brochage du connecteur réseau local

Le connecteur réseau local à vitesse modéré est un mini-DIN femelle à 8 broches.

Brochage réseau local SCC (Port A)

Broche	Fonction
1	Handshake (sortie) (DTR, RS423)
2	Handshake (entrée)/Horloge externe
3	Donnée de transmission -
4	Masse
5	Donnée réceptrice -
6	Donnée de transmission +
7	TRxCA/CTSA
8	Donnée réceptrice +

III.4 MFP

Deux contrôleurs périphériques multifonction (MFP) 68901 sont utilisés pour fournir les timers système, des ports série RS232C et un contrôleur d'interruption. L'un deux, appelé MFP-ST, est compatible ST. Il fournit un port série et un contrôle d'interruption. Le second fournit un autre port série basse vitesse, des E/S et des broches d'interruption.

La vitesse du transmetteur et du récepteur série du MFP provient de la sortie du timer D de chaque MFP. Le MFP étant cadencé à 2,4576 MHz, ses ports série supportent des vitesses de transfert allant jusqu'à 19,2 Kbauds.

III.4.1 Brochages des ports série MFP

Les deux ports série MFP sont branchés sur des connecteurs DB-9 broches et sont compatibles AT. Sur le TT, le port série MFP-2 est acheminé vers une barrette de la carte mère. Cette barrette peut être connectée avec un câble-ruban à un connecteur D 9 broches situé sur le carter du slot VME.

L'un des ports série MFP offre un ensemble complet de lignes de contrôle modem compatibles ST, mais sur un connecteur D 9 broches. L'autre port série MFP ne fournit qu'une interface "trois fils".

Brochages de ports série MFP

broche	MFP-ST	MFP-2
1	Détection de porteuse (E)	--
2	Donnée réceptrice (E)	Donnée réceptrice (E)
3	Donnée de transmission (S)	Donnée de transmission (S)
4	Terminal de données prêt (S)	Terminal de données prêt (S) (toujours sur "on")
5	Masse	Masse
6	--	--
7	Demande pour émettre (S)	Demande pour émettre (S) (toujours sur "on")
8	Prêt à émettre (E)	--
9	Indicateur d'appel (E)	--

Le signal de l'indicateur d'appel (RI) est connecté au port universel E/S du MFP-ST (GPIP) 6 bits.

III.4.2 Broches E/S non définies

Les deux bits de poids faible du port universel E/S du MFP-2 ne sont pas habituellement utilisés. Ce sont de simples signaux de niveau TTL unbuffered qui peuvent être utilisés en E/S.

III.5 Port d'imprimante parallèle

L'architecture du TT comprend un port d'imprimante parallèle bi-directionnel 8 bits qui implémente un sous ensemble Centronics. L'interfaçage est effectué par le composant General Instruments AY-3-8910 ou par le générateur sonore programmable (PSG) Yamaha YM-2149. Elle est branchée sur une DB-25S (style AT). Le signal STROBE (Centronics) est généré par une broche PSG. Le signal BUSY (Centronics) de l'imprimante est transmis par une des lignes d'entrée parallèles du MFP pour permettre l'interruption de la gestion de l'imprimante.

Huit bits de données d'écriture/lecture sont traités par le port B E/S du PSG.

III.6 Interface clavier

L'interface clavier du TT est un sur-ensemble compatible, similaire à celui des MEGA/ST. Ce clavier est équipé d'un port combiné souris/manette de commande et d'un port manette de commande. Le clavier transmet un scan code codé des touches, les données souris/trackball, les données manette de commande et l'heure. Un adaptateur d'interface de communications asynchrone (ACIA) MC6850 permet au clavier de recevoir les commandes et de transmettre les données via une transmission de type bidirectionnel implémentée par un ACIA MC6850. La vitesse de transfert de données est de 7812,5 bits/seconde. Toutes les fonctions du clavier telles que la lecture des touches, le déplacement de la souris, l'analyse syntaxique de la commande, etc. sont exécutées par un microprocesseur 8 bits HD6301V1. (Cf. Atari Intelligent Keyboard (ikbd) Protocol, 26 février 1985).

III.6.1 Interface souris et manette de commande

La souris deux boutons d'Atari peut être une souris mécanique, opto-mécanique ou optique à infra-rouge présentant au moins les caractéristiques suivantes : résolution de 100 pas par pouce, vitesse maximale de 10 pouces/seconde, et erreur maximale en phase d'impulsion de 50%. La manette de commande est une manette à quatre directions avec un bouton de mise à feu.

III.7 Cartouche ROM

Le port cartouche du TT est compatible avec les cartouches ST. La connexion se fait par le bord 40 broches de la carte. Les cartouches ROM sont mappées vers une région de mémoire de 128Ko allant de 0x00FA0000 à 0x00FBFFFF (avec une copie de 0xFFFA0000 à 0xFFFBFFFF).

III.8 Sous-système vidéo

Le sous-système vidéo TT est conçu comme une extension des modes ST existant. Des modes supplémentaires sont disponibles sur le TT permettant l'utilisation d'un plus grand nombre de couleurs et d'écrans plus importants.

III.8.1 Configuration vidéo

Les différents modes disponibles sur le TT sont :

Mode ST

Mode	Résolution	Plans	Couleurs	Palette
			(entrées Palette & convertisseurs N/A)	
00	320x200	4	16	512/3-bits
01	640x200	2	4	512/3-bits
10	640x400	1	-	Monochrome

Mode TT

Mode	Résolution	Plans	Couleurs	Palette
				(entrées Palette & convertisseurs N/A)
000	320x200	4	16	4096/4-bits
001	640x200	2	4	4096/4-bits
010	640x400	1	2	4096/4-bits (Duochrome)
011	<réservé>			
100	640x480	4	16	4096/4-bits
101	<réservé>			
110	1280x960	1	-	Monochrome
111	320x480	8	256	4096/4-bits

Comme le tableau l'indique, les modes sont initialisés en fonction du type de registre utilisé (mode ST ou mode TT). Dans le mode ST, la palette est comprise dans des registres de 16 bits. Neuf-bits de couleur sont compris dans chaque entrée : rouge, vert et bleu (3 bits chacun). Il y a ainsi une sélection totale de 512 combinaisons de couleurs possibles (8x8x8) pour chaque entrée.

Le mode 00 (320x200x4) peut utiliser la totalité des seize couleurs de la palette. Le mode 01 (640x200x2) ne peut utiliser que les quatre premières couleurs de la palette (Reg0 - Reg3). Le mode monochrome (10 -640x400x1) est doté, à la place, d'un inverseur pour la vidéo inversée contrôlé par le bit 0 de la palette 0 (ST Reg 0). La palette 0 est utilisée pour définir la couleur du bord en mode multi-plans.

D'autres résolutions sont disponibles en programmant le shifter par le registre de mode shift du TT. Dans ces modes, on obtient un maximum de 256 registres de palette TT contenant chacun 12-bits de couleur : rouge, vert et bleu (4 bits chacun). Cela donne 4096 combinaisons de couleurs possibles (16x16x16). Grâce à la banque de palette ST (4 bits inférieurs du registre de mode shift du TT), l'une des 16 banques peut être choisie depuis la palette TT pour une utilisation dans les modes ST. Cela permet aux modes 000, 001, 010 et 100 de sélectionner apparemment jusqu'à 256 registres en configurant simplement la banque de palette. Il n'y a que le mode 111 (320x480x8) qui peut utiliser directement la totalité des 256 registres.

Le mode Duochrome est une extension du mode monochrome ST d'Atari. Au lieu d'être limité au noir et blanc, le mode Duochrome permet l'affichage de deux couleurs programmables. Le registre 254 de la palette TT est normalement utilisé pour la couleur "0", et le registre 255 est utilisé pour la couleur "1". Exactement comme pour le ST, les couleurs de l'écran peuvent être inversées en configurant D1 dans le registre 0 de palette ST (ou TT).

HyperMono est un mode spécial qui combine deux des sorties du convertisseur N/A pour donner 8 bits de contrôle au niveau des trois canons. La sortie verte de l'entrée de la palette sélectionnée fournit les 4 bits de poids fort et la sortie bleue fournit les 4 bits de poids faible.

III.8.2 Interface RAM vidéo/Contrôleur/Affichage

La mémoire vidéo d'affichage est configurée sous forme de plans logiques (1, 2, 4 ou 8) de 16-bits entrelacés de mémoire consécutive pour former une mémoire tampon d'écran de 32 000 octets (modes ST) ou de 153 600 octets (modes TT) démarrant à n'importe quel 8 octets (seulement sur une RAM double fonction). L'adresse de démarrage de la mémoire d'affichage est chargée sur les registres Vidéo Base High, Vidéo Base Mid ou Vidéo Base Low (l'octet de poids fort des trente-deux adresses binaires est toujours zéro c'est-à-dire

dans la copie ST). Ce registre est chargé sur le compteur d'adresses vidéo (High/Mid/Low) au début de chaque trame. Le compteur d'adresse est incrémenté au fur et à mesure de la lecture du tampon d'écran.

La mémoire tampon d'écran est transférée immédiatement vers la mémoire tampon de la puce vidéo (shifter TT) 64 bits. Le shifter charge ensuite le registre vidéo shift où un bit de chaque plan est décalé et utilisé collectivement comme index (le plan 0 apparaît tout d'abord dans la RAM et fournit le bit de poids faible de chaque pixel) pour le registre spécifique de palettes ST ou TT (dépendant du mode décalage).

III.8.3 Connecteur du moniteur

La sortie vidéo s'effectue sur un connecteur à 3 rangées de 15 broches similaire à celui qui est utilisé sur les cartes VGA.

Broche	Fonction
1	Rouge
2	Vert
3	Bleu
4	Sortie Monochrome + haute résolution
5	Masse
6	Retour Rouge
7	Retour Vert
8	Retour Bleu
9	Détection Monochrome (entrée)
10	Masse
11	Ouvert
12	Ouvert
13	Synchronisation horizontale
14	Synchronisation verticale
15	Sortie Monochrome - haute résolution

III.9 Sous-système sonore

L'architecture du TT est une extension du sous-système sonore disponible actuellement sur les ST/MEGA. Le TT combine la sortie du système sonore ST PSG avec un nouveau système à double canal numérique/analogique commandé par le DMA. Le TT comprend un haut-parleur interne commandé par ces deux sources pour les simples "bip", mais il peut être connecté à un amplificateur stéréo externe si l'on désire obtenir un son haute-fidélité.

Le TT est également équipé d'une interface MIDI qui permet la transmission sérielle à haute vitesse des données sonores en provenance et en direction de synthétiseurs.

III.9.1 Générateur sonore programmable (PSG)

Le système sonore du ST (à base de General Instruments AY-3-8910 et de PSG) fait parti du TT. Le PSG YM-2149 produit une synthèse musicale, des effets sonores et un feedback. Cadencé à 2 MHz, le PSG fournit une gamme de fréquences de réponses située entre 30 Hz (audible) et 124 KHz (post-audible). Le générateur effectue la plus grande partie du traitement, ne laissant au système principal (agissant comme séquenceur) que le minimum. Le générateur peut fonctionner en utilisant trois voies indépendantes. Les trois sorties des canaux sonores sont combinées et envoyées à la section de contrôle du volume et de la tonalité.

(extrait de la page 10 du guide "*Reference Engineering Hardware Specification of the Atari ST computer System*")

III.9.2 Son échantillonné

Identique au STE.

III.9.3 Interface MIDI

L'interface MIDI permet de combiner le TT avec des synthétiseurs, des séquenceurs, des drums, ou tout appareil possédant une interface MIDI. La transmission en série (31,25 Kbaud) des données clavier et des données de programmation est effectuée par deux ports, MIDI OUT et MIDI IN (MIDI OUT comprend également les signaux MIDI THRU). L'interface MIDI communique avec le bus système grâce à l'adaptateur d'interfaces de communications asynchrone (ACIA) MC6850. Le taux de transfert des données est constant, à 31,25 Kbaud de données asynchrones 8-bits. (Pour plus d'informations sur l'interface MIDI et l'adaptateur d'interface ACIA, consultez les pages 11 et 17 du guide "*Reference Engineering Hardware Specification of the Atari ST Computer System*").

IV. Bus VME

Avec le bus VME au standard de l'industrie (révision C.1), le TT permet des extensions supplémentaires. Le TT propose une carte VME simple hauteur, fond de panier. L'espace mémoire est subdivisé pour permettre au 68030 d'accéder aux cartes A24/D16 et A16/D16.

IV.1 Le contrôleur système

La carte système joue le rôle de contrôleur système de bus VME (slot 1). Elle offre les fonctions suivantes :

- arbitreur de bus VME - simple niveau (niveau trois)
- gestionnaire de connexion en guirlande IACK*
- horloge système globale SYSCLK (16 MHz, indépendant de la vitesse du processeur)
- le time out global du bus VME gère le BERR*

L'arbitrage à trois niveaux est conçu pour répondre aux impératifs techniques du bus VME.

Le gestionnaire de connexion en guirlande IACK* est conçu pour répondre aux impératifs techniques du bus VME.

SYSRESET* est forcé à l'état bas en cas (1) de mise sous-tension, (2) d'une action sur le bouton de réinitialisation ou (3) d'envoi par le 68030 du signal RESET*.

IV.2 Partitionnement des adresses

L'interface A24/D16 du bus VME du TT est fixée aux emplacements : 0xFE000000-0xFEFEFFFF. L'espace A16/D16 occupe 0xFEFF0000-0xFEFFFFFF.

IV.3 Interrupteur VME

Le système peut écrire sur un adresse E/S pour générer une interruption niveau 3 du bus VME. Il peut surveiller un registre d'état qui indique le moment où cette interruption a été autorisée et effectuée. Une adresse E/S contient un registre d'état en écriture/lecture, et seulement le bit de poids faible de l'octet de poids faible est défini. A "1", il génère une interruption de niveau 3 du bus VME. Une fois supprimée, la demande d'interruption est effacée.

Remarquez que l'interruption niveau 3 doit être masquée (soit en établissant la procédure de chargement initial du processeur, soit en masquant l'interruption dans le contrôleur système) ou bien le 68030 sera immédiatement interrompu.

La carte système répond à un cycle d'accusé de réception de l'interruption du bus VME avec le statut d'identification à 0xFF.

Cette caractéristique permettra la compatibilité avec les TT à venir.

V CARTE MEMOIRE, E/S, & INTERRUPTION

Abréviations utilisées :

DW 32 bits

W 16 bits

OB Octet impair (Un port 16 bits qui apparaît dans l'octet de poids faible des mots définis. L'octet de poids fort des mots est indéfini. Si on le désire, on peut accéder à ces ports en tant qu'octets en ajoutant 1 aux adresses de mots spécifiés).

EB Octet pair (Un port 16 bits qui apparaît dans l'octet de poids fort des mots définis. L'octet de poids faible des mots est indéfini).

CARTE MEMOIRE (du 68030)

Adresse	taille	cache possible	utilisation
00000000-00FFFFFF ROM	DW	oui	RAM ST (double fonction),
00F00000-00F7FFFF	W	non	<réservé TT E/S>
00F80000-00FFFFFF	W	non	E/S ST & TT
01000000-013FFFFFF	DW	oui	RAM rapide TT (option)
01400000-FDFFFFFF	--	--	<réservé>
FE000000-FEFFFFFF	W	non	bus VME A24 : D16
FEFF0000-FEFFFFFF	W	non	bus VME A16 : D16
FF000000-FFFFFF	--	--	copie compatible ST

(une écriture en FFD000x initialise la vitesse de régénération rapide de la RAM ; et génère simultanément une erreur de bus.)

Copie compatible ST (Adresse de base 00000000 ou FF000000)

adresse	taille	cache possible	utilisation
000000-000007		DW	oui ROM (image des huit premiers octets de la ROM principale, mode superviseur, lecture uniquement)
000008-9FFFFFFF		DW	oui RAM "double fonction" (la mémoire dans la série 000008-0007FF est seulement accessible en mode superviseur)
A00000-DFFFFFFF		-	oui <réservé>
E00000-EFFFFFFF		DW	oui ROM principale
F00000-F9FFFF		W	non <réservé>
FA0000-FBFFFF		W	non Cartouche ROM
FC0000-FF7FFF		-	non <réservé>
FF8000-FFFFFFF		W	non Espace E/S ST & TT

CARTE E/S ST/TT (Décalage de la copie ST FF8000)

(Adresse de base 00FF8000 OU FFFF8000)

décalage	taille	utilisation
8000-8001	OB	Contrôleur de mémoire
8002-81FF	-	<réservé>
8200-8263	OB	Sous-système vidéo TT
8264-83FF	-	<réservé>
8400-85FF	W	Palette TT
8600-86FF	W	Accès direct mémoire et FCD ST
8700-8715	OB	Contrôleur DMA SCSI
8716-877F	-	<réservé>
8780-878F	OB	Contrôleur SCSI
8790-87FF	-	<réservé>
8800-8803	EB	Composant sonore ST
8804-88FF	-	<réservé>
8900-891F	OB	Contrôle sonore DMA
8940-895F	-	<réservé>
8960-8963	OB	Horloge temps réel et RAM non volatile
8964-8BFF	-	<réservé>
8C00-8C15	OB	Contrôleur DMA SCC
8C16-8C7F	-	<réservé>
8C80-8C87	OB	SCC
8C88-8DFF	-	<réservé>
8E00-8E1F	OB	Unité de contrôle du système (SCU)
8E20-91FF	-	<réservé>
9200-9201	EB	Sélecteurs de configuration
9202-9FFF	-	<réservé>
A000-A3FF	W	Extension périphérique de la carte principale TT
A400-F9FF	-	<réservé>
FA00-FA3F	OB	MFP-ST
FA40-FA7F	-	<réservé>
FA80-FABF	OB	MFP-2
FAC0-FBFF	-	<réservé>
FC00-FC03	EB	Interface IKBD
FC04-FC07	EB	MIDI ACIA
FC08-FFFF	-	<réservé>

E/S DES PERIPHERIQUES

SOUS-SYSTEME VIDEO ST/TT

8200	RW	----	----	xxxx	xxxx	Vidéo Base High
8202	RW	----	----	xxxx	xxxx	Vidéo Base Mid
8204	RO	----	----	xxxx	xxxx	Compteur d'adresses vidéo high
8206	RO	----	----	xxxx	xxxx	Compteur d'adresses vidéo mid
8208	RO	----	----	xxxx	x000	Compteur d'adresses vidéo low
820A	RW	----	--0x			Mode de synchronisation ST (défini à 1)
820B	WO			0000	0000	<réservé>
820C	RW	----	----	xxxx	x000	Vidéo Base Low
8240	RW	----	Rrrr	Gggg	Gbbb	Reg0 palette couleur ST
8242	RW	----	Rrrr	Gggg	Bbbb	Reg1 palette couleur ST
...						
825E	RW	----	Rrrr	Gggg	Bbbb	Reg15 palette couleur ST

Remarque : Les lettres capitales "R"(rouge), "G"(vert), "B"(bleu) indiquent le bit de poids faible dans la valeur effective de la couleur.

8260	RW	----	--ss	----	----	ST shift mode
			(ss		00	320x200, 4 plans
			01		640x200,	2 plans
			10		640x400,	1 plan
			11		<réservé>	
)			
8262	RW	s--h	-mmm	----	bbbb	TT shift mode
			(s			échantillonnage et maintien)
			(h			mode hyper mono)
			(mmm			
			000			320x200x4
			001			640x200x2
			010			640x400x1
			100			640x480x4
			110			1280x960x1
			111			320x480x8)
			(bbbb			banque de palette ST)

SOUS-SYSTEME VIDEO TT

8400	RW	----	rrrR	gggG	bbbB	Reg0 palette TT
8402	RW	----	rrrR	gggG	bbbB	Reg1 palette TT
...						
85FE	RW	----	rrrR	gggG	bbbB	Reg255 palette TT

DMA ACSI ST

8600					<réservé>	
8602					<réservé>	
8604	RW	----	----	xxxx	xxxx	Disk Data Path (WDC)
8606	RO	----	----	----	-xxx	Etat DMA
8606	WO	----	---x	xxxx	xxxx	Mode DMA (WDL)
8608	RW	----	----	xxxx	xxxx	Pointeur DMA High
860A	RW	----	----	xxxx	xxxx	Pointeur DMA Mid
860C	RW	----	----	xxxx	xxx0	Pointeur DMA Low

SCSI DMA

8700	RW	----	----	xxxx	xxxx	Pointeur DMA Upper
8702	RW	----	----	xxxx	xxxx	Pointeur DMA Upper-Mid
8704	RW	----	----	xxxx	xxxx	Pointeur DMA Lower-Mid
8706	RW	----	----	xxxx	xxxx	Pointeur DMA Lower
8708	RW	----	----	xxxx	xxxx	Byte Count Upper
870A	RW	----	----	xxxx	xxxx	Byte Count Upper-Middle
870C	RW	----	----	xxxx	xxxx	Byte Count Lower-Middle
870E	RW	----	----	xxxx	xxxx	Byte Count Lower
8710	RO	xxxx	xxxx	xxxx	xxxx	Data Residue Register High
8712	RO	xxxx	xxxx	xxxx	xxxx	Data Residue Register Low
8714	RW	----	----	bz00	00ed	Control Register

(
 b - erreur de bus pendant le DMA
 (lecture seulement, effacée à la lecture)
 z - nombre d'octets zéro
 (lecture seulement, effacé à la lecture)
 e - validation du DMA 0=off, 1=on
 d - direction du DMA :
 0=entrée depuis le port
 1=sortie vers le port
)

Contrôleur SCSI (5380)

8780	OB	Data Register
8782	OB	Initiator Command Register
8784	OB	Mode Register
8786	OB	Target Command Register
8788	OB	ID Select/SCSI Control Register
878A	OB	DMA Start/DMA Status Register
878C	OB	DMA Target Receive/Input Data
878E	OB	DMA Initiator Receive/Reset

GENERATEUR DE SON PROGRAMMABLE (PSG)

(fournit aussi un port imprimante parallèle bi-directionnel et un loquet pour sorties diverses)

8800	RO	xxxx	xxxx	----	----	PSG Read Data
8800	WO	0000	xxxx	----	----	PSG Register Select
8802	WO	xxxx	xxxx	----	----	PSG Write Data

Affectation des bits du port A

- 7 Sélection *LAN (0 achemine le port A SCC vers le connecteur réseau local)
- 6 *Désactivation du haut-parleur (0 désactive le haut-parleur interne)
- 5 Signal de validation du port imprimante

- 4 *DTR (port série MFP-ST)
- 3 *RTS(port série MFP-ST)
- 2 Sélection de *Floppy 1
- 1 Sélection de *Floppy 0
- 0 Sélection de *Floppy Side 0

Affectation des bits du port B

7-0 Bits 7-0 - port imprimante

SOUS-SYSTEME SONORE DMA

8900 RW ---- ---- 0000 00re Contrôleur sonore DMA

- (
- r - Répétition
 - 0 = Trame unique
 - 1 = Répétition
- e - Mise en service
 - 0 = Off (reset)
 - 1 = On

-)
- 8902 RW ---- ---- xxxx xxxx Frame Base Adresse (high)
- 8904 RW ---- ---- xxxx xxxx Frame Base Adresse (med)
- 8906 RW ---- ---- xxxx xxxx Frame Base Adresse (low)
- 8908 RW ---- ---- xxxx xxxx Frame Address Counter (high)
- 890A RW ---- ---- xxxx xxxx Frame Address Counter (med)
- 890C RW ---- ---- xxxx xxxx Frame Address Counter (low)
- 890E RW ---- ---- xxxx xxxx Frame End Address (high)
- 8910 RW ---- ---- xxxx xxxx Frame End Address (med)
- 8912 RW ---- ---- xxxx xxxx Frame End Address (low)
- 8920 RW 0000 0000 a000 00bb Sound Mode Control

- (
- a - Mode
 - 0 = Stéréo (reset)
 - 1 = Mono
- bb - Echantillonnage
 - 00 = 6258 Hz
 - 01 = 12517 Hz
 - 10 = 25033 Hz
 - 11 = 50066 Hz

-)
- 8922 RW xxxx xxxx xxxx xxxx MICROWIRE Data Register
- 8924 RW xxxx xxxx xxxx xxxx MICROWIRE Mask Register

HORLOGE TEMPS REEL (MC146818A)

- 8960 OB RTC Address Register
- 8962 OB RTC Data Register

SCC DMA

- 8C00 RW ---- ---- xxxx xxxx Pointeur DMA Upper
- 8C02 RW ---- ---- xxxx xxxx Pointeur DMA Upper-Middle
- 8C04 RW ---- ---- xxxx xxxx Pointeur DMA Lower-Middle
- 8C06 RW ---- ---- xxxx xxxx Pointeur DMA Lower
- 8C08 RW ---- ---- xxxx xxxx Byte Count Upper
- 8C0A RW ---- ---- xxxx xxxx Byte Count Upper-Middle

8C0C RW ---- ---- xxxx xxxx Byte Count Lower-Middle
 8C0E RW ---- ---- xxxx xxxx Byte Count Lower
 8C10 RO xxxx xxxx xxxx xxxx Data Residue Register High
 8C12 RO xxxx xxxx xxxx xxxx Data Residue Register Low
 8C14 RW ---- ---- bz00 00ed Control Register

(
 b - erreur de bus pendant DMA
 (lecture seulement, effacé à la lecture)
 z - comptage du nombre d'octets à zéro
 (lecture seulement, effacé à la lecture)
 e - DMA activé 0=off, 1=on
 d - direction DMA
 0=entrée depuis le port
 1=sortie vers le port
)

8530 SCC

8C80 OB SCC1 A (commande)
 8C82 OB SCC1 A (donnée)
 8C84 OB SCC1 B (commande)
 8C86 OB SCC1 B (donnée)

SCU

8E00 OB System Interrupt Mask (B7 -B1; B0 inutilisé)
 8E02 OB System Interrupt State (lecture seulement; avant le registre masque)
 8E04 OB System Interrupter (B0 = interruption 1 générée)
 8E06 OB VME Interrupter (B0 = interruption VME IRQ3 générée)
 8E08 OB SCU General Purpose Register 1 (reset à la mise sous tension)
 8E0A OB SCU General Purpose Register 2 (reset à la mise sous tension)
 8E0C OB VME Interrupt Mask (B7 - B1 ; B0 inutilisé)
 8E0E OB VME Interrupt State (lecture seulement; avant registre masque)

MFP-ST (compatible ST)

FA00 OB GPIP
 FA02 OB AER
 FA04 OB DDR
 FA06 OB IERA
 FA08 OB IERB
 FA0A OB IPRA
 FA0C OB IPRB
 FA0E OB ISRA
 FA10 OB ISRB
 FA12 OB IMRA
 FA14 OB IMRB
 FA16 OB VR
 FA18 OB TACR
 FA1A OB TBCR
 FA1C OB TCDCR

FA1E OB TADR
FA20 OB TBDR
FA22 OB TCDR
FA24 OB TDDR
FA26 OB SCR
FA28 OB UCR
FA2A OB RSR
FA2C OB TSR
FA2E OB UDR

MFP2

FA80 OB GPIP
FA82 OB AER
FA84 OB DDR
FA86 OB IERA
FA88 OB IERB
FA8A OB IPRA
FA8C OB IPRB
FA8E OB ISRA
FA90 OB ISRB
FA92 OB IMRA
FA94 OB IMRB
FA96 OB VR
FA98 OB TACR
FA9A OB TBCR
FA9C OB TCDCR
FA9E OB TADR
FAA0 OB TBDR
FAA2 OB TCDR
FAA4 OB TDDR
FAA6 OB SCR
FAA8 OB UCR
FAAAOB RSR
FAACOB TSR
FAAEOB UDR

ikbd ACIA

FC00 EB Clavier ACIA (commande)
FC02 EB Clavier ACIA (donnée)

MIDI ACIA

FC04 EB MIDI ACIA (commande)
FC06 EB MIDI ACIA (donnée)

Remarque : IOCS1 et IOCS2, deux broches du processeur glue donne le décodage offset dans l'espace d'E/S de 0xA000-0x00A1FF et 0xA200-0xA3FF. Ces broches minimisent le décodage lorsque des périphériques sont ajoutés à la carte principale du TT (à venir).

AFFECTATION D'INTERRUPTION

<u>int</u>	<u>système</u>	<u>vecteur</u>	<u>VME</u>	<u>vecteur</u>
7	bus VME SYSFAIL	Autovecteur	IRQ7	programmable
6	aucun	-	MFP & IRQ6	programmable
5	aucun	-	SCC & IRQ5	programmable
4	VSYNC	Autovecteur	IRQ4	programmable
3	(Note 3)	-	Interrupteur VME IRQ3	Autovecteur programmable
2	HSUNC	Autovecteur	IRQ2	programmable
1	Inter. du système	Autovecteur	IRQ1	programmable

Remarque 1 : Dans chaque niveau, l'interruption du système est prioritaire sur l'interruption du VME. Et dans les interruptions partagées du niveau 5 et du niveau 6, la partie sur la carte mère est prioritaire sur l'interruption du VME.

Remarque 2 : Les interruptions VME utilisent aussi bien leurs octets d'interruption d'état que leur vecteur d'interruption.

Remarque 3 : Le masque d'interruption du système de niveau 3 doit être validé pour l'interruption du VME de niveau 3 pour être effectivement généré.

Affectations d'interruption MFP

MFP-ST	(compatible ST)
int	fonction
GPIP7	Détection monochrome du moniteur / IRQ son DMA
GPIP6	Indicateur d'appel
TimerA	
RxRDY	
RxERR	
TxEMPTY	
TxERR	
TimerB	
GPIP5	ACSI / Interruption FDC
GPIP4	MIDI / Interface clavier
TimerC	
TimerD	
GPIP3	<réservé>
GPIP2	CTS
GPIP1	DCD
GPIP0	BUSY (Centronics)
MFP	
GPIP7	IRQ Contrôleur SCSI (actif à "1")
GPIP6	IRQ RTC (actif à "0", effacé par la lecture du registre RTC 0x0C)
TimerA	
RxRDY	
RxERR	
TxEMPTY	
TxERR	
TimerB	
GPIP5	Interruption DMAC SCSI (actif à "0")
GPIP4	<réservé>
TimerC	

MANUEL DEVELOPPEUR

TimerD	
GPIP3	Indicateur d'appel (SCC B)
GPIP2	Interruption SCC DMAC (actif à "0")
GPIP1	Broche E/S fonction générale
GPIP0	Broche E/S fonction générale

PRIORITES DMA/BUS MASTER

PRIORITES DMA

priorité	fonction
haute	Contrôleur de lecteur de disquettes/ACSI
	Contrôleur DMA SCC
	Contrôleur DMA SCSI
basse	

Atari TT030
- Extension VME -

Le slot d'extension VME du TT030 accueille une carte VME simple hauteur (format 3U Eurocarte), compatible A24/D16 ou A16/D16. Ce slot supporte uniquement des cartes asservies. Les dimensions mécaniques du connecteur et du module sont conformes à la spécification du bus VME VITA C.1.

L'emplacement est compatible électriquement avec les spécifications VME, mais veuillez noter les points suivants :

Aucun arbitrage de bus n'est supporté. BR0*, BR1*, BR2*, et BR4* sont connectés les uns avec les autres et forcés à l'état haut par une résistance de 1K vers le VCC. BG0IN*, BGG1IN*, BG2IN*, et BG3IN* sont connectés ensemble et forcés à l'état haut par une résistance de 1K vers le VCC. BBSY* et BCLR* sont forcés à l'état haut par une résistance de 1K vers le VCC mais ne sont pas commandés autrement. BG0OUT, BG1OUT, BG2OUT, et BG3OUT ne sont pas connectés.

Les lignes d'interruption IRQ1* à IRQ7* peuvent chacune être utilisées et sont forcées à l'état haut par une résistance de 1K vers le VCC. IRQ3*, IRQ5* et IRQ6* peuvent être forcés à l'état bas par le système. Le signal SYSFAIL* est forcé à l'état haut par une résistance de 1K vers le VCC. Il peut générer une interruption du système niveau 7 lorsqu'il est excité par une carte. IACK* et IACKIN* sont commandés par le système. Une carte ne devrait pas commander ces signaux. IACKOUT* n'est pas connecté. Le mot d'état fourni par la carte pendant le cycle d'interruption autorisé est utilisé comme vecteur d'interruption 68030. Pour une compatibilité avec les produits Atari, le vecteur alimenté ne doit pas être OxFF. Toutes les interruptions du bus VME et du système sont indépendamment masquables dans le SCU.

Dans le TT030, SYSCLK est généré par une horloge cadencée à 16,107953 MHz. Les signaux SERCLK et SERDAT* ne sont pas connectés. Le signal ACFAIL* est forcé à l'état bas par le système lorsque l'alimentation électrique n'est pas stable. ACFAIL* est excité 1Ms avant que l'alimentation ne quitte la gamme réglée. Il est forcé à l'état haut par une résistance de 1K.

AM0, AM1, AM2, et AM4 sont générés par le système. AM3 et AM5 sont connectés l'un à l'autre et forcés à l'état haut par une résistance de 1K vers le VCC. Cette implémentation permet des accès non privilégiés et le contrôle standard des données et des programmes ainsi que des accès non privilégiés et de contrôle court. Les transferts par blocs ne sont pas supportés. LWORD* est forcé à l'état haut par une résistance de 1K mais n'est pas commandé autrement.

Les signaux BERR* et SYSRES* sont connectés directement aux signaux de reset et d'erreur du bus système. Le timer d'erreur de bus implémenté sur la carte système dépasse le temps imparti et génère une erreur de bus si la carte n'excite pas DTACK* dans les 255 cycles de 16MHz après l'arrêt du signal VME AS*. Les signaux SYSRES* générés lorsque le 68030 exécute un reset doit durer au plus 16uS. Ils sont forcés à l'état haut par des résistances de 1,2K et peuvent être forcés à l'état bas tant par le système que par la carte. Le signal +5VSTDBY est à +5V. DTACK* est forcé à l'état haut par une résistance de 1K.

Tous les autres signaux des connecteurs sont conformes aux spécificités VME, mais avec les restrictions électriques quant à l'unité courante et au connecteur de terminaison. Il n'y a pas terminaison dans le système que les états forcés mentionnés ci-dessus. Toutes les sorties ont au moins une capacité de 1 LS TTL et aucune entrée ne présente plus de deux charges LS TTL.

Sur le TT030, après les 16 Mo A24, il reste de l'espace disponible :

FEOO OOOO - FFFF FFFF	A24:D16, la carte VME voit les 24 bits de poids faible
FEFF OOOO - FFFF FFFF	poids de l'adresse A16:D16

Les sociétés tierces devront concevoir leurs cartes afin d'obtenir des adresses, des priorités d'interruption, et des vecteurs d'interruption paramétrables par cavalier ou par programme. C'est une bonne pratique de conception que de décoder l'espace d'adresse aussi précisément que possible. L'espace A16 est particulièrement sensible.

Les sociétés tierces devront fournir des octets d'état d'interruption dans la gamme 0x80-0xBF (c'est-à-dire les vecteurs entre 0x200 et 0x2FF)

Alimentation de la carte VME :

Tension	Activité maximale	Régulation	Ondulation/Parasite (crête-à-crête)
+5V	2.0A	+5%/-4%	60 mV
+12V	50MA	+/-5%	120mV
-12V	50MA	+/-10%	120mV

ATARI TT030

REMARQUES A L'INTENTION DES DEVELOPPEURS

Ce document contient une première série d'informations sur le TT030, à l'intention des développeurs. Ces données s'ajoutent à la documentation ST et STe existante.

GENERALITES

Un TT030 est un ST à base de 68030 cadencé à 16MHz, avec les changements suivants (en termes généraux) :

- Une extensibilité RAM supplémentaire, avec une RAM rapide, en mode "nybble".
- Transferts SCSI de type DMA (disques durs SCSI et autres périphériques).
- Son stéréo numérique échantillonné (comme sur le STe).
- Co-processeur à virgule flottante 68881/68882
- Modes vidéo supplémentaires (voir plus loin)
- SCC 8530 (contrôleur de transmissions sérielles), permettant ainsi d'obtenir un connecteur Locatalk (avec DMA) et un port série supplémentaire, ou bien deux ports série supplémentaires.
- Un haut-parleur interne.

Toutes ces nouveautés exigeaient certains changements au niveau du logiciel interne, particulièrement au niveau du BIOS. GEMDOS a également été modifié. Les nouvelles résolutions ont entraîné des changements au niveau AES et VDI, ainsi qu'au niveau du bureau pour permettre les contrôles cache.

La chose la plus importante à retenir à propos du TT est qu'il est compatible ST et donc compatible GEM. En conséquence, pour la majorité des applications, TOUTES les procédures habituelles effectuées sous GEM restent inchangées. Les documentations en votre possession tant sur AES, VDI, GEMDOS, XBIOS ou BIOS sont toujours valables et s'appliquent au TT. Vous trouverez ci-après une description générale des éléments ci-dessus permettant une adaptation parfaite aux spécificités matérielles.

LE NOUVEAU MATERIEL

LE 68030

Le 68030 possède deux mémoires cache : l'une d'instructions et l'autre de données. Elles sont expliquées en détails dans la documentation du 68030. Le TOS démarre avec ces mémoires cache inhibées mais il existe une option sur le bureau pour les valider (exactement comme le Blitter du Mega). Normalement, l'utilisation des mémoires cache est transparente (les programmes étant seulement plus rapides !). Les programmes qui modifient la mémoire, puis l'exécutent (par exemple le code auto-modificateur) devront être vérifiés. Les programmes qui utilisent directement le DMA (opposés à ceux effectuant les appels BIOS ou XBIOS) doivent invalider les mémoires cache après l'exécution de l'opération DMA, avant d'accéder à la mémoire qui peut avoir été changée par le DMA.

Le 68030 a un bus d'adresses 32-bits complet, et non le bus 24-bits du 68000. Les programmes qui utilisent les 8 bits supérieurs d'adresses pour une raison ou une autre (par exemple une zone de caractères pour un pointeur) ne fonctionnent pas sur le TT030.

Bien entendu, le 68030 16MHz avec un bus de données 32-bits est plus rapide que le 68000 8MHz avec un bus de données 16-bits. Les programmes qui reposent sur la synchronisation d'instructions ne sont pas correctement exécutés.

Ceux d'entre vous qui font de la programmation ou de la mise au point de langage d'assemblage peuvent obtenir plus d'informations dans le manuel d'utilisation du 68030 édité par Motorola : MC68030UM/AD REV 1, "Enhanced 32-Bit Microprocessor User's Manual, Second Edition". Le 68030 exécute les programmes "mode utilisateur seulement" du 68000, mais le déplacement depuis l'instruction SR est désormais privilégié. Certains compilateurs, notamment l'Alcyon C4.14, utilisent cette instruction. Le BIOS possède un programme de traitement pour l'exception du privilège d'infraction, qui recherche les erreurs éventuelles dans cette instruction : elle remplace l'instruction en un déplacement depuis le CCR et essaie de l'exécuter à nouveau. Cela est suffisant pour la plupart des programmes.

Les auteurs de programmes qui exécutent des opérations de contrôle (par exemple répondent aux interruptions) doivent être informés d'un autre changement capital : le format des piles d'exceptions a été changé. Il y a un mot supplémentaire sur la pile après le retour SR et PC. Cela signifie que si une interruption a été effectuée depuis le mode superviseur, les arguments que le demandeur a justifiés se trouvent un mot plus loin du sommet de la pile que dans le 68000. C'est le cas pour tous les processeurs 680x0 sauf le 68000. Il existe une nouvelle variable du système, `_longframe`, à \$59e : si ce mot est différent de zéro, vous êtes sur un CPU qui utilise le format long. (Consultez les documentations Cookie Jar pour savoir sur quel CPU vous travaillez).

MEMOIRE

Il y a trois sortes de mémoire dans un TT030

- o La RAM ST est une RAM double fonction, partagée par le CPU et toutes les unités ST (vidéo, DMA ACSI, son échantillonné). Le TT a 2Mo de RAM ST en standard, avec une extension possible de 2Mo supplémentaires.
- o La RAM TT est une RAM spécialisée : elle n'est pas partagée par les unités compatibles ST. Les accès ont été accélérés puisque le CPU ne doit jamais attendre son tour. Elle est aussi plus rapide grâce à l'adoption du mode "nybble". Elle n'est pas visible pour les transferts DMA ACSI, vidéo ou son échantillonné. Elle est visible pour le CPU et pour les transferts de type DMA SCSI. Le TT030 supporte une carte RAM TT de 4 Mo (ou 16Mo avec des barrettes de 4Mo).
- o La RAM VME est une mémoire qui se trouve sur le bus VME. Celui-ci étant seulement de 16 bits, la RAM VME a à peu près la même vitesse que la RAM ST. Elle est visible seulement pour le CPU. (Elle n'est pas visible pour les transferts de type DMA SCSI qui exigent un chemin de données de 32-bits).

DMA SCSI

Un bus SCSI est disponible sur le TT030. Pour les développeurs et les utilisateurs, cela signifie que pratiquement toutes les unités de disque dur peuvent être connectées au TT030. Les utilitaires du disque dur Atari peuvent reconnaître, formater, partitionner et installer (rendre amorçable) pratiquement tous les périphériques de mémoire de masse SCSI. (La norme SCSI, bien que parfaitement définie, n'est pas toujours implémentée de la même façon par tous les fabricants, et il est possible que certaines unités dites "SCSI" ne fonctionnent pas avec le TT030).

Les disques durs ne sont pas les seuls périphériques SCSI. Vous pouvez connecter également des dérouleurs de bandes, des noeuds de réseaux, etc... Ils se connectent parfaitement au TT030, mais les fabricants, VAR ou utilisateurs devront définir les programmes de gestions correspondants.

Le TT030 possède une connexion SCSI interne sous la forme d'un connecteur 50 broches pour câble-ruban, et il peut accueillir une unité 3"1/2. Il a, de plus, un connecteur SCSI externe 25 broches compatible avec les autres connecteurs SCSI 25 broches de l'industrie (lire "Mac"). L'utilisateur peut donc acheter l'un de ces périphériques et le brancher directement à un TT030.

SON ECHANTILLONNE, HAUT-PARLEUR INTERNE

Les caractéristiques d'échantillonnage du TT030 sont exactement les mêmes que celles du STe. Les registres matériel sont aux mêmes adresses et ont les mêmes fonctions. Les sons utilisés en mode DMA et ceux du processeur de son compatible ST sont (option) mixés par le contrôleur de volume/tonalité et envoyés vers le haut-parleur interne et vers les prises RCA droite/gauche à l'arrière de la machine. Un interrupteur permet de désactiver le haut-parleur interne (bit 6, port A sur le registre sortie fonction générale du PSG).

68881

Le TT030 est fourni avec un co-processeur à virgule flottante Motorola 68881. Celui-ci peut être utilisé par des programmes pour réaliser des calculs à virgule flottante. Il n'est pas prévu que ce co-processeur soit partagé, il ne doit donc pas être utilisé par des accessoires ou depuis des interruptions. (Les systèmes multi-tâche devront sauvegarder et restaurer l'état du 68881 exactement comme ils sauvegardent et restaurent l'état des registres CPU lorsqu'ils passent d'une opération à l'autre).

VIDEO

Le TT030 supporte les trois modes vidéo ST, ainsi que trois nouveaux modes.

ST LOW	320x200	16 couleurs
ST MEDIUM	640x200	4 couleurs
ST HIGH	640x400	2 couleurs (pas seulement noir & blanc)
TT LOW	320x480	256 couleurs
TT MEDIUM	640x480	16 couleurs
TT HIGH	1280x960	noir et blanc

Les résolutions ST, et les deux résolutions couleur TT utilisent le même type de moniteur. La résolution TT HIGH (1280x960) nécessite un moniteur spécifique. Atari tient à disposition une liste des modèles compatibles et de leur fabricants.

La palette est identique à celle du STe : rouge, vert et bleu, (4 bits chacun), et offre un total de 4096 couleurs. Dans la palette compatible ST à \$FFFF8240 (identique au ST et STe), le bit de poids fort de chaque quartet est le bit de poids faible de la valeur du canon correspondant pour le rouge, le vert et le bleu. De plus, à une nouvelle adresse (\$FFFF8400), vous trouvez une palette 256 entrées avec les bits en ordre naturel. Dans les modes 16 couleurs (et le mode 4 couleurs), la palette est divisée en 16 "banques", seule une de ces "banques" est active à la fois. C'est la banque active qui est visible dans la palette 16 entrées compatible ST. Le changement de banque demande seulement une écriture sur le composant vidéo, vous pouvez utiliser le système de banques pour changer immédiatement l'ensemble des 16 couleurs. De nouveaux appels XBIOS permettent d'accéder aux palettes et aux registres du mode shifter.

La résolution ST HIGH est désormais appelée "Duochrome". Vous pouvez, en effet, afficher n'importe quel groupe de deux couleurs (et non simplement le noir et le blanc). Les deux couleurs qui sont affichées sont les deux dernières dans la palette 256 entrées. De plus, un bit est utilisé pour inverser l'affichage, exactement comme dans le mode haute résolution du ST. Le bit en question est le bit 1 (et non le bit 0) de la première entrée de la palette.

L'existence de nouveaux modes vidéos révélera les paresseuses pratiques de programmation de certains développeurs qui ont parfois des idées préconçues quant aux écrans, telles que leur résolution, le nombre de couleurs disponibles, ou la taille de l'image en mémoire. Il a toujours été possible d'utiliser les fonctions d'enquête VDI, ou même d'observer l'espace variable Ligne A, et de déterminer les caractéristiques de l'affichage. Même l'écriture du code "indépendant de la résolution" qui appelle Getrez() n'est pas suffisamment bonne, étant donné que Getrez() renvoie des valeurs pour les nouveaux modes qui étaient impossibles (et par conséquent non anticipés) sur un ST. Utilisez le VDI pour obtenir les informations dont vous avez besoin.

En général, seuls les programmeurs les plus prudents évitent les idées reçues dans ce domaine. Les programmes qui utilisent exclusivement AES/VDI fonctionnent souvent dans les nouveaux modes, permettant à leurs utilisateurs de profiter d'un espace d'écran plus grand, des couleurs, etc. D'autres programmes auront besoin d'être modifiés, ou seront limités aux modes compatibles ST.

SCC ET AUTRES PORTS SERIE

Le TT030 a trois nouveaux ports série. Deux d'entre eux viennent du SCC 8530 (contrôleur de transmissions sérielles) : ils ont tous les signaux de contrôle modem. L'un d'eux est partagé avec le connecteur réseau local compatible Locatalk : vous pouvez utiliser le port réseau local ou bien le port série, mais pas les deux à la fois.

L'autre nouveau port vient du nouveau MFP 68901. Il fonctionne exactement comme le port compatible ST, à la différence qu'il n'a que les signaux de transmission, de réception et de masse : il n'y a pas de signaux de contrôle modem sur ce port.

Le BIOS supporte tous ces ports, y compris les primitives XON/XOFF ou RTS/CTS, tout en conservant la compatibilité avec les programmes existants. Reportez-vous à la documentation concernant les appels Bconmap. (Le nouveau port 68901 ne supporte pas les signaux RTS/CTS, puisqu'ils ne sont pas disponibles).

CHANGEMENTS CONCERNANT LE TOS

Le numéro de version du TOS TT (premier release) est 3.0 (TOS 3.0). Jusqu'à ce que cette ROM soit terminée, toutes les ROM du TT portent le numéro 3.0. Des distinctions plus précises seront effectuées avec le code date sur l'en-tête du système d'exploitation.

LES MODIFICATIONS DE GEMDOS

DEUX SORTES DE RAM

Cette section introduit le concept de "RAM alternative", en général tout d'abord, puis ensuite plus spécifiquement par rapport au TT030.

Le TT030, comme les autres modèles de la gamme ST à venir, comporte deux sortes différentes de RAM. La RAM ST, compatible ST, et la "RAM alternative" qui n'est pas compatible. Cette non-compatibilité varie selon la machine et le type de RAM. Premièrement, le composant vidéo ne peut afficher que des données d'écran provenant de la RAM ST, et le composant sonore DMA ne dispose que des données stockées dans la RAM ST. Deuxièmement, d'autres composants qui ont accès à la mémoire, comme l'ACSI DMA (disques durs ST et autres unités) et le SCSI DMA (périphériques SCSI), ne pourront peut-être pas accéder directement à la mémoire alternative. La plupart des programmes ne sont pas affectés puisqu'ils utilisent les appels BIOS et GEMDOS pour ce genre de transfert, et c'est au programme de gestion de périphériques que revient la responsabilité de prendre les données "ici" et de les laisser "là", ceci de manière transparente, où que se trouvent "ici" et "là".

Les "règles d'éligibilité" pour l'exécution d'un programme en RAM alternée sont :

- (1) On ne doit pas essayer d'initialiser l'adresse de base de l'écran en RAM alternative, ou d'y placer un son échantillonné.
- (2) On ne doit pas essayer de faire un driver vers le DMA, à moins que le driver connaisse les différences entre la RAM ST et la RAM alternative.
- (3) Le programme ne doit pas tenter lui-même d'opérations DMA (il n'y a que les drivers spécifiques qui fassent cela).

Le deuxième point est un peu difficile : il se réfère au fait que les drivers DMA ne connaissent pas les restrictions de la RAM alternative.

Les programmes ayant été écrits avant l'apparition du concept "RAM alternative", ils ne savent pas s'ils respectent ou non les règles. C'est donc à vous d'informer GEMDOS du fait que le programme peut utiliser la RAM alternative, ou bien doit utiliser la RAM ST. Pour une distinction plus précise, vous pouvez sélectionner l'éligibilité du chargement du programme et les appels Malloc() séparément. Un programme dont le Malloc est un tampon d'écran devrait être encore éligible pour charger en RAM alternative, mais ses appels Malloc() doivent être satisfaits depuis la RAM ST.

SPECIFICITES SUR LES DEUX SORTES DE RAM

Comme avec le TOS Rainbow, un des mots longs réservés dans l'en-tête des fichiers exécutables (PRG, TTP, TOS) a acquis un sens : les bits contrôlent la façon avec laquelle GEMDOS traite ce programme. (Le bit de poids faible de ce mot long (bit0), quand il est initialisé, signifie que GEMDOS n'a pas besoin d'effacer la totalité de la RAM lors du chargement de ce programme, seul ce programme est déclaré BSS. Cela accélère le chargement du programme).

Les deux bits suivants ont été assignés à un sens en relation avec la RAM alternative. Le bit 1, lorsqu'il est effacé, signifie que le programme doit être chargé en RAM ST ; le bit 2, lorsqu'il est effacé signifie que les appels Malloc de ce programme doivent être satisfaits en utilisant la RAM ST.

Quand un de ces bits est initialisé, l'opération correspondante (chargement du programme, appel Malloc) doit être satisfaite depuis la RAM "alternative". En général, la RAM alternative est préférable à la RAM ST. Si un programme remplit les conditions pour l'éligibilité en RAM alternative, il est souhaitable de placer ces bits dans son en-tête.

Si la RAM TT est éligible mais que l'espace RAM n'est pas assez grand, la demande va provenir de la RAM ST. Si l'espace n'est, là non plus, pas assez important, la demande échoue.

Pour charger un programme, le concept de RAM "suffisante" est tout à fait relatif. Pour certains, il est plus important de tourner rapidement que d'avoir beaucoup de mémoire. On peut ainsi dire qu'une RAM "suffisante" est une RAM de 256Ko de plus que ses propres exigences déclarées (texte+données+bss). Pour un autre programme, il est plus important d'avoir beaucoup de RAM, même si cela implique une certaine lenteur d'exécution.

Un nouveau secteur dans l'en-tête du programme, TPAsize, indique la mémoire minimum nécessaire. Si le bit "chargement du programme" du programme est effacé (signifiant qu'il doit charger en RAM ST), ce secteur est ignoré, et le programme est chargé en RAM ST. Si le programme peut être chargé en RAM alternative, et qu'il y a plus de RAM alternative que de RAM ST, le secteur est ignoré et le programme est chargé en RAM alternative. Le secteur est seulement vérifié si la RAM alternative est éligible pour charger le programme, et que la RAM ST disponible est plus importante. Dans ce cas, TPAsize communique la quantité de RAM alternative considérée comme "suffisante". Si cet espace mémoire existe, le programme s'y charge ; sinon, le programme se charge en RAM ST.

TPAsize indique l'espace de RAM alternative considéré comme suffisante, par incréments de 128Ko. Ce chiffre est ajouté à la capacité déclarée (texte+données+bss) du programme. Si la quantité disponible est inférieure à cette somme, le programme est chargé en RAM ST. Le secteur peut contenir quatre bits, et se trouve dans les quatre bits supérieurs du mot long des indicateurs de programme. Le total est égal à la valeur du secteur multipliée par 128Ko, plus 128Ko. En conséquence une valeur de zéro, ce que les programmes ont souvent, signifie un total de 128K. La valeur peut aller jusqu'à 15, soit un total de 2Mo.

EXEMPLE :

SETUP :

Un bit de chargement en RAM alternative est initialisé. TPAsize est configuré à 512Ko. La somme de texte+données+bss est de 110Ko.

RESULTAT :

S'il y a plus de RAM alternative que de RAM ST, le programme se charge en RAM alternative.

S'il y a plus de RAM ST, TPAsize est pris en compte. S'il y a 622Ko de RAM alternative disponible, ou plus, le programme est chargé en RAM alternative. Sinon, il est chargé en RAM ST.

Dans cet exemple, il est possible qu'il n'y ait pas non plus 622Ko de RAM ST disponible. S'il y en a plus de 110Ko, le programme est chargé et exécuté ; TPAsize n'est pas considéré comme un minimum absolu pour le chargement du programme. 110Ko est la capacité texte+données+bss déclarée du programme : cela, plus l'espace pour une petite pile initiale, représente le minimum indispensable.

Rappelez-vous que TPAsize ne reflète pas la capacité maximale ou minimale TPA que le programme va finalement obtenir. Cette valeur représente seulement le circuit de départage lorsqu'il y a plus de RAM ST que de RAM alternative, afin que GEMDOS sache où placer le programme.

QUELLE EST LA SIGNIFICATION DE TOUT CELA ?

L'Atari ST dispose de deux modèles de mémoire communs pour les programmes. Pour l'un, l'utilisateur ou la bibliothèque doit définir une "taille de pile" au moment de la compilation ou de l'édition de lien. Le démarrage du runtime déplace le pointeur de pile vers la fin du programme plus la taille de la pile et utilise Mshrink pour rendre le reste du TPA à GEMDOS. Puis, lorsque le programme appelle la bibliothèque malloc(), il utilise Malloc pour reprendre la mémoire au système d'exploitation. Pour ce genre de programme, TPAsize devrait représenter au moins autant d'espace que la "taille de la pile" utilisée par le démarrage. MWC, GCC, Turbo C, et beaucoup d'autres environnements utilisent ce modèle de mémoire.

L'autre modèle conserve une certaine quantité de mémoire, et cette mémoire est utilisée comme un en-tête -- la pile s'y étend du haut vers le bas, et les appels de la bibliothèque malloc() utilisent la mémoire du bas vers le haut. Pour ce genre de programme, TPAsize doit correspondre à la capacité raisonnable minimale de pile+en-tête. Alcyon C utilise ce modèle de mémoire.

Vous vous demandez certainement pourquoi ces secteurs font partie de l'en-tête du programme, et ne sont pas contrôlés, disons, par de nouveaux appels GEMDOS, ou de nouveaux paramètres vers le Pexec. La raison est qu'ils font, à proprement parler, partie du programme. Les caractéristiques de l'espace RAM alternative et les exigences de la mémoire d'un programme sont inhérentes à son comportement. Elles ne sont pas basées sur la structure arborescente et celle-ci ne doit pas s'intéresser à elles de façon à ce que GEMDOS prenne des décisions intelligentes.

Etant donné que les informations se trouvent dans l'en-tête du programme, elles peuvent être remplacées par un programme utilitaire extérieur ne possédant pas une connaissance particulière de la structure du programme. Si un programme ne produit pas de "screen-flipping" ou ne s'adresse pas directement au composant DMA, il peut

probablement être exécuté en RAM alternative, et vous pouvez définir ses attributs convenablement.

APRES LE CHARGEMENT DE VOTRE PROGRAMME

Le bit dans l'en-tête du programme qui contrôle l'éligibilité de la RAM alternative pour les appels Malloc est destiné à la compatibilité, ainsi les programmes existants qui n'ont pas connaissance de la RAM alternative peuvent bénéficier d'une vitesse plus rapide et d'une capacité supplémentaire.

Les nouveaux programmes, écrits après la publication de ces informations, peuvent utiliser un nouvel appel, Mxalloc() :

GEMDOS CALL 0x44 : Mxalloc

LONG Mxalloc (amount, mode)

LONG amount;

WORD mode;

Cet appel agit comme Malloc(), mais prend un paramètre supplémentaire : un attribut signalant où prendre la mémoire.

MODE	SIGNIFICATION
0	seulement RAM ST
1	seulement RAM alternative
2	l'une ou l'autre, préférence pour la RAM ST
3	l'une ou l'autre, préférence pour la RAM alternative

Si "amount" = -1L, la taille du plus grand bloc unique du caractère spécifié par "mode" est communiquée. Dans ce cas, les valeurs 2 et 3 de "mode" sont identiques, et la taille du plus grand bloc de l'un ou l'autre caractère est communiquée.

Si "amount" est différent de -1L, et qu'un bloc de cette taille est disponible dans le(s) caractère(s) de mémoire spécifiés par "mode", le bloc est affecté et son adresse de début d'implantation est communiquée.

Il est clair que l'appel Malloc() revient à un appel Mxalloc() avec une valeur "mode" de 0 ou 3, dépendant de l'état du bit d'éligibilité Malloc dans le programme de l'en-tête.

NOUVEAUX APPELS XBIOS

XBIOS 0x2a : lecture DMA
0x2b : écriture DMA

```
long DMAread(secteur,count,buffer,devno)
long secteur;
word count;
void *buffer;
word devno;
```

Lit les secteurs depuis l'unité vers la mémoire. Fonctionne pour les unités ACSI et SCSI. Pour SCSI, il n'utilise pas en fait le DMA : il relie les octets entre eux. Les numéros des unités sont :

\$0-\$7	unités ACSI	\$0-\$7
\$8-\$f	unités SCSI	\$8-\$f
autre	réservé pour une utilisation future	

Retourne un code d'erreur BIOS.

DMAwrite est identique, mais écrit des secteurs. Ces appels supposent que la mémoire "buffer" peut effectivement être atteinte par le bus sur lequel se trouve l'unité. En conséquence, DMAread depuis une unité ACSI vers une RAM alternative ne fonctionne pas.

Appel 0x2e XBIOS : NVMAccess

```
WORD NVMAccess(op,start,count,buffer)
WORD op,start,count;
BYTE *buffer
```

Cet appel traite la mémoire non volatile (NVM) de l'horloge du TT. Il y a là 50 octets de mémoire, parmi lesquels deux sont réservés pour une vérification du reste des données. Cet appel valide les données de vérification en lectures, recalcule les données de vérification en écritures, et initialise les données de vérification si vous le désirez.

OPCODE	SIGNIFICATION
0	READ : copie les données depuis NVM vers le buffer
1	WRITE : copie les données depuis le buffer vers NVM
2	INIT : met NVM à zéro et initialise les données de vérification

"start" spécifie le premier emplacement à lire ou écrire
"count" spécifie le nombre d'octets à transférer.

Retour à zéro en cas de succès. EBADRQ (-5) en cas d'erreur dans les arguments. EGENRL (-12) si la vérification NVM n'est pas cohérente avant une lecture ou une écriture. Dans le cas de la lecture, les données sont transférées de toute façon.

L'utilisation de NMV doit être dictée par Atari. Nous accepterons les suggestions et les applications pour les affectations d'octets, mais l'utilisation d'octets ou de valeurs dont nous ne publions pas les significations entraînera assurément des problèmes dans le futur.

AUTRES REMARQUES IMPORTANTES

L'interface de traitement graphique ligne-A est mise à jour pour la compatibilité en amont avec les programmes ST *existants*. **On ne doit pas l'utiliser pour de nouveaux programmes.** Elle ne pourra pas se conformer aux modifications à venir des équipements et des logiciels. On devra utiliser le VDI.

Ce document décrit le nouvel appel XBIOS, Bconmap(), qui rend les nouveaux ports série du TT030 accessibles aux programmes qui ont été écrits lorsqu'il n'y avait qu'un seul port série.

Nouveaux numéros d'unité Bconin/out/stat/ostat du TT :

devno	signification
0	PRN
1	Port série actuellement mappé (voir ci-après)
2	CON
3	MIDI
4	IKBD
5	RAW
6	Port série compatible ST (appelé Modem 1, défaut).
7	Canal B SCC (Modem 2 à l'arrière du TT).
8	Port série TTMFP (3 fils, Série 1).
9	Canal A SCC (handshake complet, Série 2).

Les appels Bcon sur l'unité 1 (normalement AUX) devraient effectivement se référer à n'importe laquelle de ces unités, ou à une unité installée par l'utilisateur (avec un devno encore plus élevé). Vous utilisez Bconmap pour changer la mappe de l'unité 1. Bconmap change également la mappe des appels Rscnf(), et des appels Iorec avec un numéro d'unité 0 Iorec.

Les affectations de port ci-dessus concernent uniquement le TT ; d'autres machines avec des ports série "supplémentaires" auront d'autres affectations. Cependant, le port 6 est toujours destiné à être un port compatible ST. D'autres unités pourront être installées par des circuits de commande au moment de l'amorçage (ou même après).

Appel XBIOS 44 (0x2c)

```
LONG Bconmap(devno)
WORD devno;
```

Cet appel mappe "devno" comme unité 1 Bcon*. Il retourne l'ancien mappage. Si devno est -1, il n'y a pas de changement; le mappage en cours est simplement retourné. Si devno = -2, un pointeur est retourné (voir ci-après). Les valeurs légales sont -1 (pour aucun changement), -2 (pour le pointeur), ainsi que les valeurs 6 et au-delà.

Vous pouvez savoir que vous êtes dans un système qui ne supporte pas Bconmap en faisant cet appel et en vérifiant le retour : si la valeur de retour est 44 (0x2c, la même chose que le numéro d'appel XBIOS) alors Bconmap n'est pas disponible.

En plus de tout ce qui a été dit auparavant, si devno = -2, un pointeur pour la mappe de l'unité est fourni. Ceci est utilisé par les programmes qui doivent installer de nouvelles entrées mappables. Il contient également le nombre d'unités mappées. La valeur légale la plus élevée de devno pour les appels Bcon (Bconmap compris) est ce nombre plus 5.

Les valeurs illégales (0-5, ou plus élevées que la valeur légale la plus élevée, ou une valeur négative sauf -1 ou -2) ne changent rien et retournent 0.

Le mappage est accompli par l'écriture sur la table vectorielle publiée en mémoire basse. De plus, de nouveaux pointeurs sont utilisés pour rendre Iorec et Rsconf indirects. Par conséquent, les programmes qui utilisent la table vectorielle en RAM basse et/ou Iorec et Tsconf pourront voir l'unité "actuellement mappée" quand ils feront des appels Bcon avec devno=1, et quand ils feront des appels Iorec et Rsconf.

Les programmes connaissant Bconmap peuvent utiliser les valeurs les plus élevées de devno pour atteindre un port spécifique, peu importe la composition de la mappe courante. Ils ont encore besoin d'utiliser Bconmap pour "mapper" le port désiré avant d'effectuer les appels Iorec et Rsconf.

INSTALLATION DES NOUVEAUX DRIVERS BCON

Bconmap(-2) retourne un pointeur à une structure. La structure ressemble à cela :

```
struct bconmap {
    LONG *maptab ;      /* ptr à mappe (voir ci-après) */
    WORD maptabsz ;    /* nombre de lignes de la mappe */
};
```

La mappe contient une ligne pour chaque unité. Chaque ligne contient des pointeurs vers les routines Bconstat, Bconin, Bconstat et Bconout, un pointeur vers la routine Rsconf; plus un pointeur vers Iorec pour cette unité. La capacité de la table (le nombre d'unités) est en maptabsz. Maptabsz est utilisé par tous les appels Bcon pour contrôler les limites du nombre d'unités. La valeur légale la plus haute pour les appels Bcon (Bconmap compris) est ce nombre plus 5.

Un circuit de commande traitant Bconmap doit avoir les adresses de lancement Bconstat, Bconin, Bconstat, et Bconout, iorec, plus un pointeur fonction Rsconf. Vous l'installez en copiant la table désignée par maptab sur une zone plus étendue et en ajoutant l'entrée de votre driver (cinq pointeurs de procédure et un pointeur iorec), puis en changeant maptab et maptabsz.

Dans le cas improbable où votre programme se trouverait en train d'installer lui-même la 38ème unité, c'est-à-dire celle qui aurait le numéro 44 de Bconmap, vous devriez en fait allouer DEUX nouvelles rangées à maptab, installer votre unité dans la SECONDE, et doubler maptabsz. Sinon, la mappe courante de l'unité 44 ne serait pas différenciable du cas où Bconmap n'est pas disponible du tout. On ne devrait jamais laisser un programme utiliser le numéro d'unité 44.

BCONMAP ET RSCONF

Rsonf est mal documenté depuis un certain temps. Il retourne en fait un double mot. Celui-ci est un ensemble de quatre octets. Ces quatre octets sont les registres UCR, RSR, et TSR du MFP, plus un octet non utilisé. Le registre UCR est dans l'octet supérieur retourné, suivi par RSR, puis TSR, et l'octet non utilisé dans l'octet de poids faible.

Ces octets sont les valeurs de ces registres AVANT les changements dictés par les arguments Rsconf.

De plus, depuis le Rainbow TOS, Rsconf(-2, -1, -1, -1, -1, -1) retourne la dernière valeur de débit initialisée avec Rsconf. Si le premier argument est -2, tous les autres arguments sont ignorés.

Dans le monde Bconmap, les arguments Rsconf doivent être interprétés un peu différemment. Les unités ne sont plus exclusivement des 68901 MFP. Pour les nouvelles unités, les bits cohérents sont utilisés, les autres sont supprimés.

Les programmeurs devraient utiliser Rsconf(-1, -1, -1, -1, -1, -1) pour lire les valeurs courantes, puis changer les bits qu'ils désirent changer et appeler Rsconf de nouveau avec les nouvelles valeurs. Le changement de bits dans des registres non listés ci-dessous est maintenant considéré comme illégal. (Dans un VERITABLE MFP : le TSR contient, entre autres choses, le bit de validation du transmetteur ; si vous écrivez 0x08 pour déclencher une interruption, vous invalidez le transmetteur !)

Les bits dans les args Rsconf et la valeur de retour que les unités traitant Bconmap doivent émuler sont les suivantes :

UCR : bits 6-5 :	nombre de mots (00=8, 01=7, 10=6, 11=5)
bits 4-3 :	bits d'arrêt : (01=1, 10=1.5, 11=2 ; 00 est invalide)
bit 2 :	parité (0=non, 1=oui)
bit 1 :	parité (0=impair, 1=pair, significatif seulement si bit 2 est 1)

RSR : aucun

TSR : bit 3 : interruption (envoie l'interruption pendant 1)

SCR : aucun

Les programmes qui utilisent des modes synchrones parlent probablement directement à leur unité, désormais, cela ne sert donc à rien de l'"émuler" ainsi. Si une valeur légale est inopportune (tels que les bits d'arrêt 1.5 avec une unité qui ne le supporte pas, ou un débit que vous ne pouvez pas supporter) vous pouvez l'ignorer : les utilisateurs vont s'habituer aux restrictions imposées par votre unité et votre driver.

Le canal A SCC est partagé entre le DB9 à l'arrière du TT030 (label Serie 2) et un connecteur réseau local sur le côté gauche. Initialement, il est défini pour utiliser le connecteur DB9.

La sélection peut être effectuée de la façon suivante :

Connecteur LAN :	Offgibit(0x7f);	/* effacer bit 7 (seulement) */
Connecteur DB9 :	Ongibit(0x80);	/* établir bit 7 (seulement) */

- (83) `_EsetColor`
`WORD _EsetColor(colorNum, color)`
`WORD colorNum, color;`
Initialise l'entrée absolue "colorNum" dans la palette TT pour la couleur donnée. La couleur est immédiatement initialisée. La fonction retourne toujours l'ancienne couleur. Si la "couleur" est négative, le registre de l'unité n'est pas modifié.
- (84) `_EsetPalette`
`VOID _EsetPalette(colorNum, count, palettePtr)`
`WORD colorNum, count;`
`LONG palettePtr`
Initialise le contenu d'un ensemble contigu de registres de palettes TT avec les mots désignés par "palettePtr". "palettePtr" doit tomber sur une limite de mots. L'ensemble des registres chargés commence avec le registre "colorNum" de la palette et calcule le nombre de mots ("count"). La fonction initialise immédiatement la palette.
- (85) `_EgetPalette`
`VOID _EgetPalette(colorNum, count, palettePtr)`
`WORD colorNum, count;`
`LONG palettePtr;`
Copie le contenu d'un ensemble contigu de registres de palette TT commençant avec le registre "colorNum" dans le secteur désigné par "palettePtr". Le nombre de mots ("count") est transféré dans ce secteur. "palettePtr" doit tomber sur une limite de mots.
- (86) `_EsetGray`
`WORD _EsetGray(switch)`
`WORD switch;`
Initialise l'interprétation de la palette par le dispositif de visualisation. Une valeur "switch" de zéro pousse le dispositif de visualisation à interpréter les données de la palette comme une couleur, avec 4 bits pour chacun des trois composants. Avec une valeur "switch" différente de zéro, l'octet supérieur de l'entrée de la palette est ignoré et seul l'octet inférieur représente un des 256 niveaux de gris. La fonction retourne toujours l'ancienne valeur de la commutation (une valeur différente de zéro signifie que "switch" est configuré). Lors d'une entrée, si "switch" est configuré sur une valeur négative, le registre du dispositif n'est pas modifié.
- (87) `_EsetSmear`
`WORD _EsetSmear(switch)`
`WORD switch`
Initialise le mode vidéo smear. Une valeur "switch" de zéro indique un mode d'affichage normal, alors qu'une valeur différente de zéro demande au dispositif d'affichage de répéter (smear) la dernière couleur rencontrée différente de zéro chaque fois que les valeurs zéro sont retrouvées. La fonction retourne toujours l'ancienne valeur de la commutation (une valeur différente de zéro signifie que "switch" est configuré). Lors d'une entrée, si "switch" est configuré sur une valeur négative, le registre du dispositif n'est pas modifié.