Rapport de projet de fin d'études

Conception et réalisation d'une carte d'interface pour Ethernet autour d'un microcontrôleur M68HC12

Dates du projet : du 2 Avril au 20 juin

Encadré par : M. KADIONIK Patrice

Moukmir marouane Chara youness	Département : électronique Option :informatique industrielle	Juin 2002

0.TABLE DES MATIERES

1. INTRODUCTION

le but de ce projet ENSEIRB est la réalisation d'une carte basée sur le microcontrôleur 68HC12 de MOTOROLA. Cette carte comprend un microcontrôleur 68HC12BC32 adressé àdes ressources externes RAM / ROM, et dispose d'une liaison série de type RS232 pour la communication de données avec un PC offrant ainsi une multitude de possibilités d'application, ainsi qu'un port de debug ou BDM (Background Debug Mode) qui est un avantage par rapport a la série 68HC11.

La carte mère intégrera une interface Ethernet àl'aide du composant CS8900A de Cirrus Logic qui sera exploité comme un périphérique 16 bits intégrant la couche liaison en Hardware. Après développement et test de la carte, il est prévu de développer les logiciels couplés au noyau Temps Réel uC/OS II afin de permettre un contrôle à distance par Sockets Internet dans un premier temps, puis par le Web dans un deuxième temps.

2. REALISATION DU PROJET

2.1 Phase de documentation

- documentation sur les protocoles TCP, UDP, ARP
- recherche de documents sur le M68HC12
- recherche de documents sur le CS8900A
- réalisation de quelques TP sur la carte 68HC11 de l'enseirb

2.2 Etablissement d'un cahier des charges

Le cahier des charges àété établi suivant les directives de notre tuteur M. Patrice KADIONIK.
le cahier de charge spécifie la réalisation de deux cartes :

- une carte mère : regroupant un composant nommé CS8900A qui sert d'interface avec le réseau Ethernet, des RAM, des ROM pour pouvoir utiliser le microcontrôleur M68HC12 en mode étendu. Cette carte sera réalisée àl'Enseirb et sera connectée àune carte fille.
- Une carte fille : regroupant le microcontrôleur M68HC12 et ses composants annexes. Elle devra disposer de connecteurs adéquats pour la connecter avec la carte mère regroupant les différents périphériques.

Le matériel réalisé lors de ce projet Enseirb devra être programmé en utilisant le noyau temps réel µCos II pour la gestion de taches en temps réel, ainsi que différents outils de développement logiciel dédié au HC12 tels le Compilateur C de Cosmic Software, éditeur de lien, convertisseur d'images hexa, chargeur de programme exécutable…

2.3 Critères de validation du projet

- Le développement du software se fait sous environnement PC, la communication avec le circuit se faisant via une liaison série.
- Les tests seront effectués sur une carte d'évaluation regroupant le microcontrôleur 68HC12 qui sera reliée au réseau via la carte mère.
- Un banc de test complet fournira les éléments nécessaires pour le fonctionnement de la carte d'évaluation.
- La validation d'écriture et de lecture dans les RAM et les ROM du prototype.

2.4 Moyens mis à disposition

- Appareils de mesures : Oscilloscope numérique, multimètres…
- Compilateur Cosmic Software pour 68HC12, éditeur de liens…
- Carte d'évaluation Motorolla EVB, elle comporte :
  - Un M68HC12
  - Une liaison série RS232 pour communication avec PC
  - Les composants annexes au HC12
- Les outils Mentor graphics : Board architect, Layout, Fablink….

2.5 Mise en œuvre de la solution

- conception sous Mentor graphics.
- Routage automatique et manuel.
- Réalisation du circuit imprimé PCB

3. PARTIE HARDWARE

3.1 Présentation

Le support matériel de notre projet s'étend sur deux cartes :

- Une carte mère : une carte mère : regroupant un composant nommé CS8900A qui sert d'interface avec le réseau Ethernet, des RAM, des ROM pour pouvoir utiliser le microcontrôleur M68HC12 en mode étendu. Cette carte sera réalisée àl'Enseirb et sera connectée àune carte fille.

- Une carte fille : regroupant le microcontrôleur M68HC12 et ses composants annexes. Elle devra disposer de connecteurs adéquats pour la connecter avec la carte mère regroupant les différents périphériques. La géométrie du M68HC12 ne nous permet pas de réaliser cette carte fille àl'ENSEIRB. Elle sera fabriquée chez un fabricant de circuit imprimés.
  Cette carte pourra être utilisée en n'importe quelle autre application, il suffira alors de changer la carte mère.

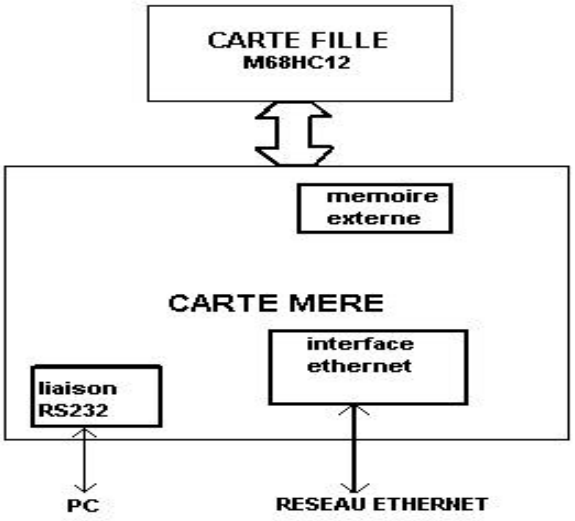


Figure 1:vue générale des deux cartes

3.2 Présentation de la carte fille

La carte fille contient :

- Les connecteurs pour le relier a la carte mère.
- Le connecteur BDM : c'est une nouveauté par rapport au 68HC11, en effet, il s'agit d'un connecteur dédié au développement de l'application en temps réel car il n'a nul besoin d'arrêter le déroulement du programme (cf. annexe)
- Le circuit intégré de reset qui fournit une durée de reset conforme aux spécifications du M68HC12.
- Le 68HC12BC32 de MOTOROLA.

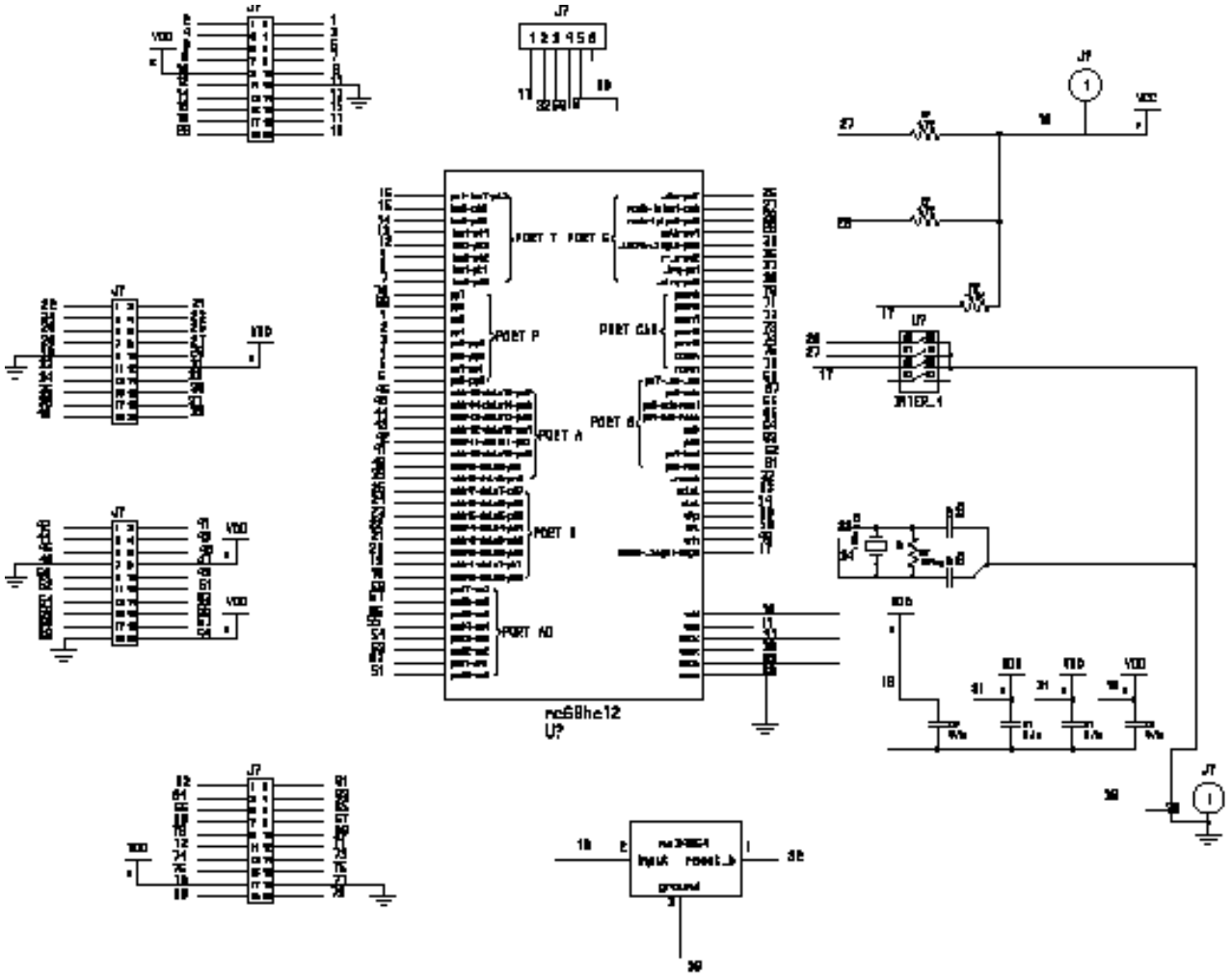


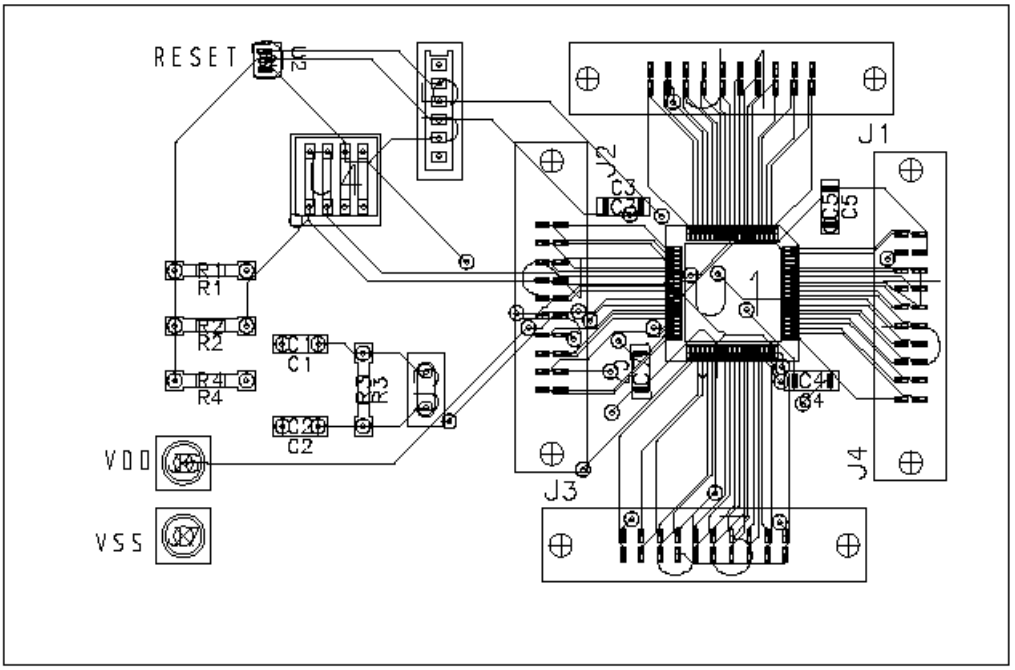
Figure 2 : SCHEMATIQUE DE LA CARTE FILLE

3.3 Routage de la carte fille:

Les 2 schémas qui suivent sont les 2 couches de la carte fille. La majorité des pistes a été travaillée manuellement.

Le routage a été effectue en utilisant les valeurs théoriques de conception suivantes:

Largeur des pistes : 0.005 inches
Clearance : 0.008 inches



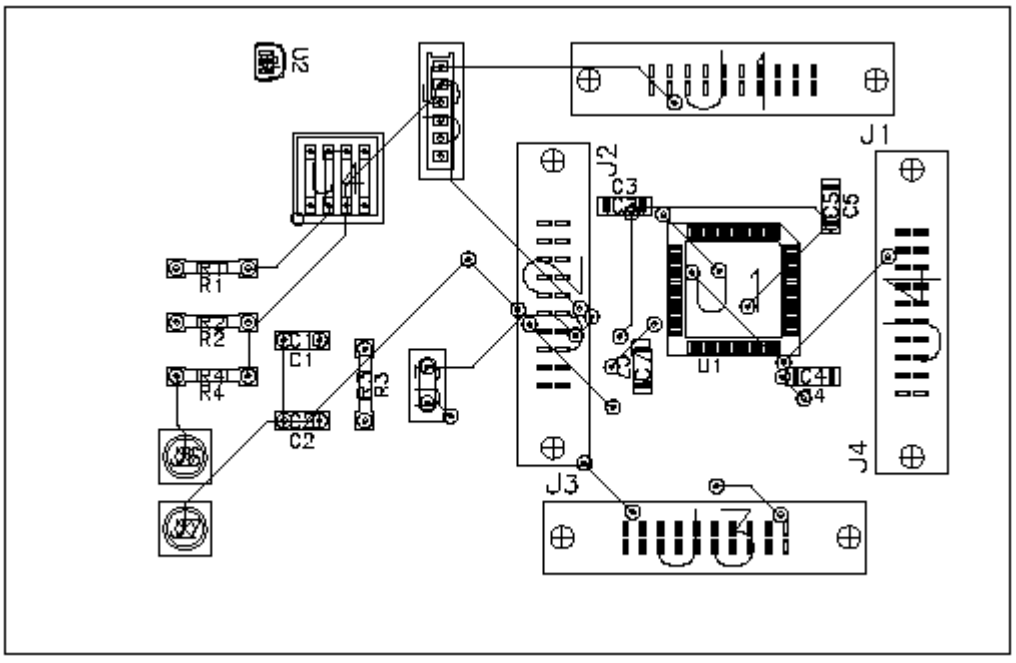


Figure 3 : vue du coté composant (en haut) et du coté cuivre (en bas)

le PCB réalisé lors de ce projet a été exploité pour fabriquer la carte à l'extérieur de l'Enseirb chez un fabricant de circuits imprimés.

3.4 présentation de la carte mère

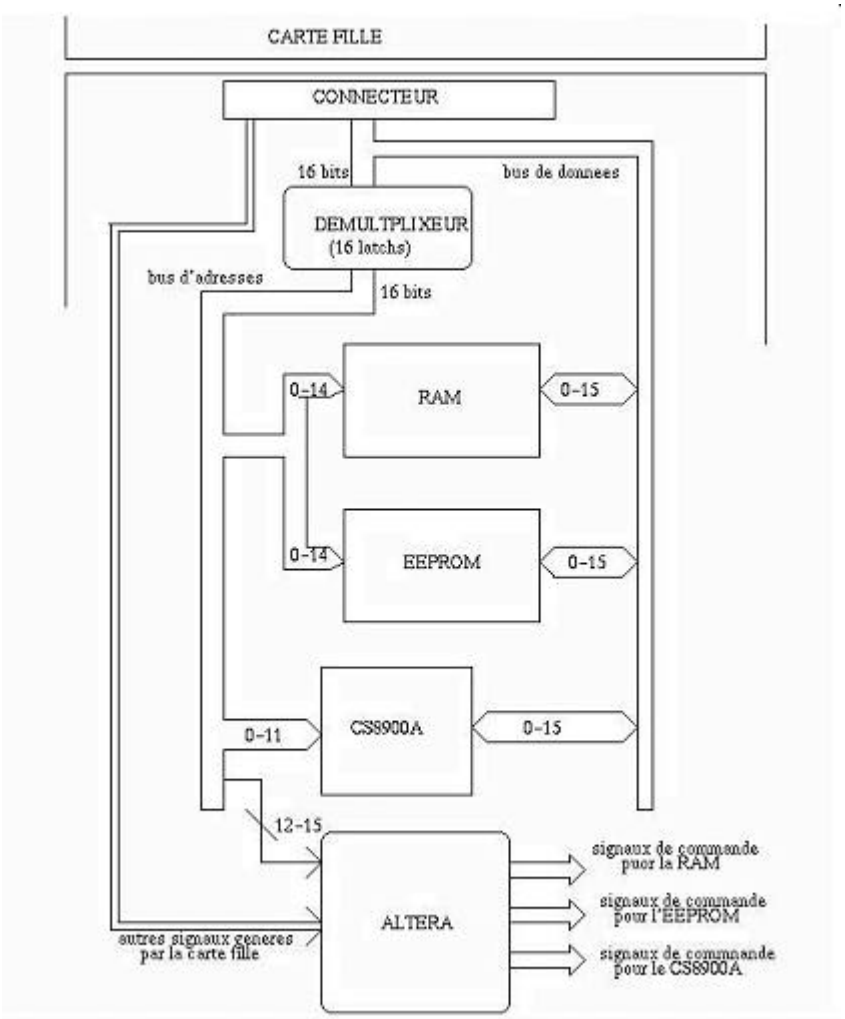


Figure 4 :schéma de principe de la carte mère

La carte mère est articulée autour d'un microcontrôleur 68HC12BC32 de MOTOROLA sur lequel est adressé jusqu'à64K octets (16 Bits d'adresse : A0..A15) de ressources externes pour le mode étendu du processeur. Parmi ces ressources externes, les deux RAM de 8 bits et d'une capacité de 32K chacune, permettent de traiter les données en mode 16 bits , en effet, dans la première on stokes le poids fort, dans la seconde on stock le poids faible.

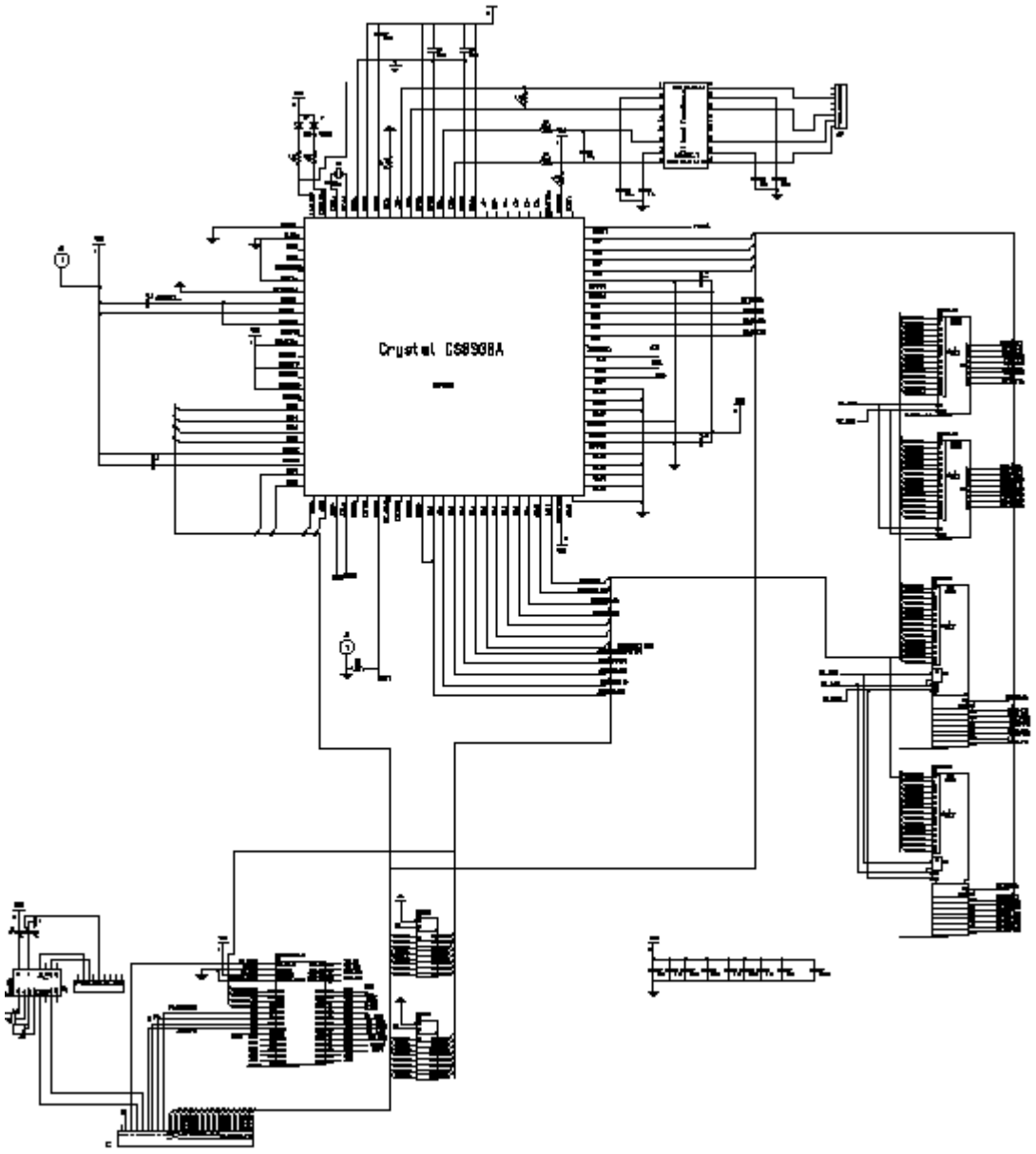


Figure 5: schématique de la carte mère

3.4.1 Démultiplexage d'adresses

Les deux circuits 74HC573 sont des latch qui sont synchronisés sur le niveau logique du signal AS qui provient du microcontrôleur et qui assurent le démultiplexage des ports A et B du 68HC12 sur lequel les adresses (A0..A7, A8...A15) et les données (D0...D7, D8…D15) sont véhiculées.

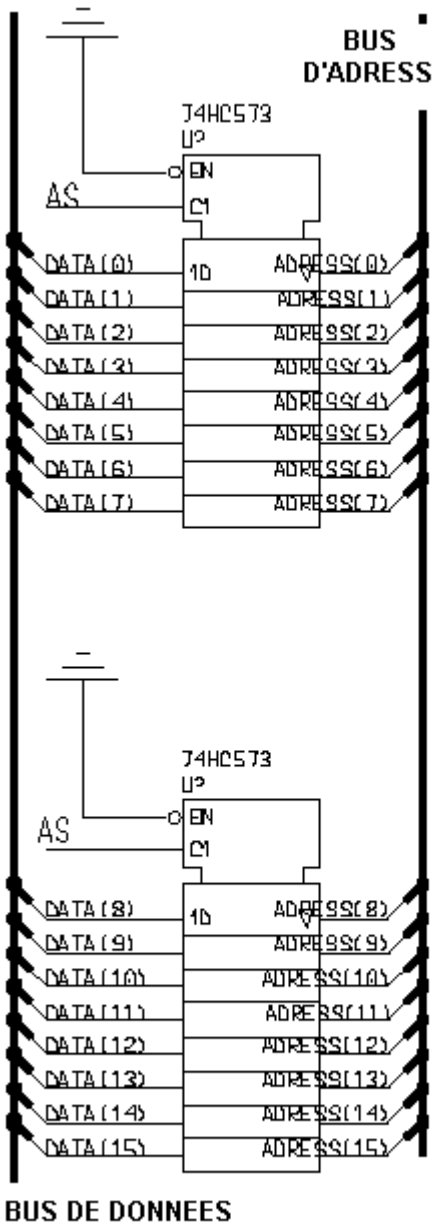


Figure 6: le démultiplexage des ports A et B

3.4.2 Décodage d'adresses

L'ALTERA permet de faire le décodage d'adresse, et suivant le cas, il génère les signaux de sélection permettant au microcontrôleur d'effectuer ses accès mémoire avec le bon composant, il génère aussi le signal d'interruption inverse qui a pour destination la carte fille comprenant le microcontrôleur 68HC12, sans oublier le signal de Reset du 68HC12 qui est actif au niveau bas et qu'on inverse pour qu'il concorde avec celui du CS8900A.

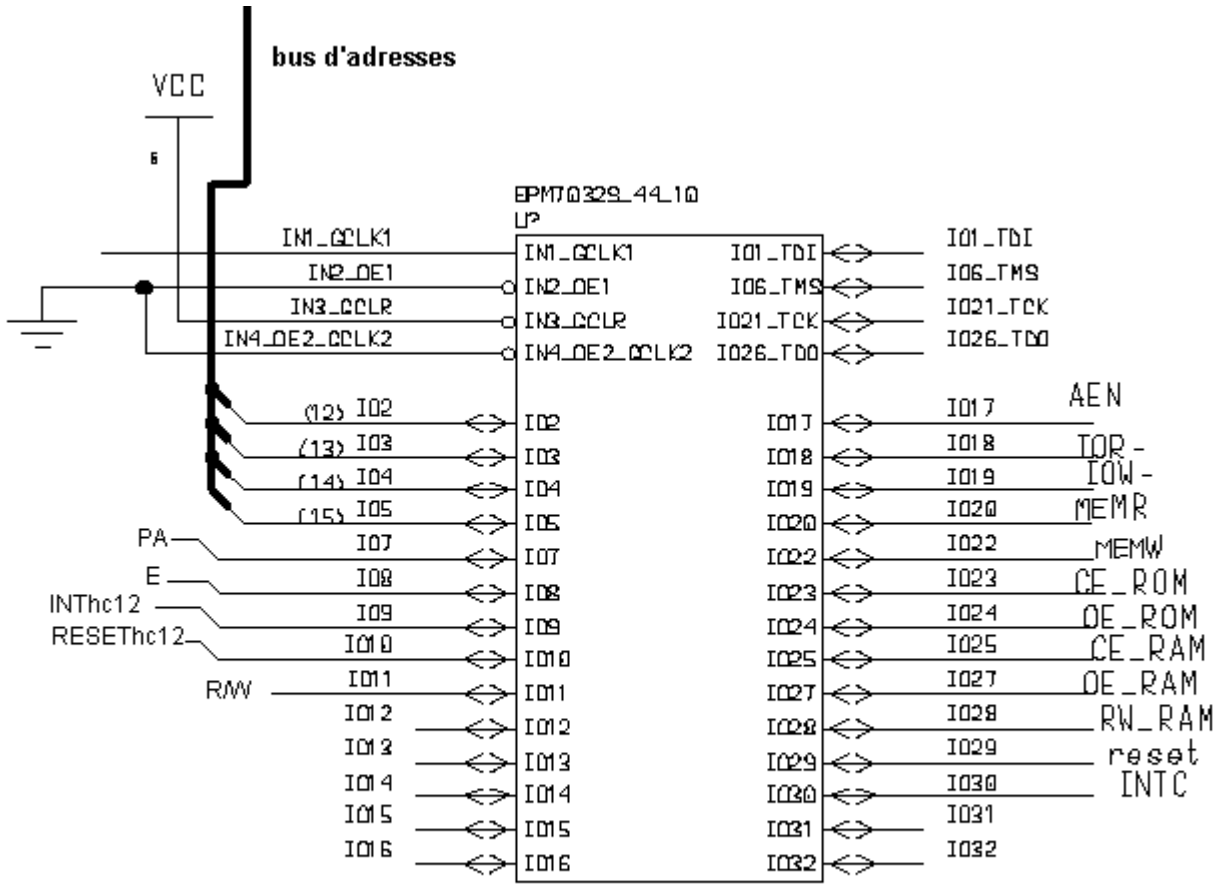


Figure 7: les signaux du FPGA

- Lorsque A15 vaut 0 :
  - Si A14.A13.A12 vaut 111, on accède dans ce cas au CS8900A (chip Ethernet) adressé aux adresses $7XXX.
  - Si A14.A13.A12 est différent de 111 (c'est àdire pour les adresses débutant en $8000) on accède alors àla RAM.
- Lorsque A15 vaut 1, c'est l'EEPROM qui est sélectionnée.

D' où le tableau suivant :

A15	A14.A13.A12	RAM	CS8900A	EEPROM

0	**de 0 0 0 à 1 1 0**	**1**	**0**	**0**
0	**1 1 1**	**0**	**1**	**0**
1	**X X X**	**0**	**0**	**1**

Figure 8: table de vérité pour le décodage

La cartographie mémoire qui en résulte est :

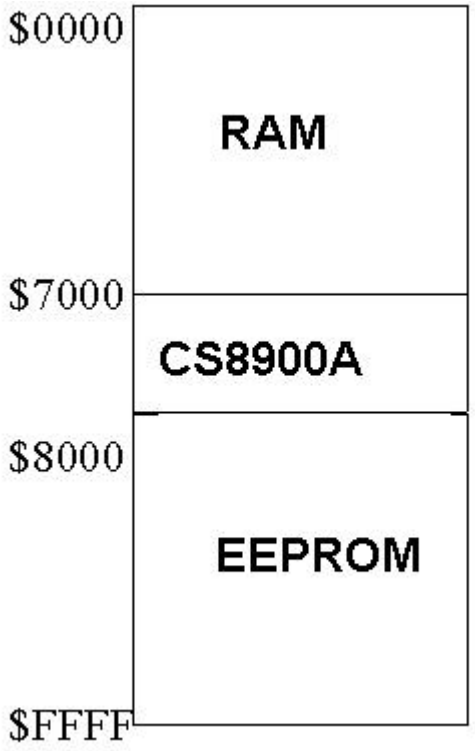


Figure 9: Cartographie de la mémoire

On a alors les équations suivantes :

- cs = A15*.(A14.A13.A12)
- ram = A15*.(A14.A13.A12)*
- eeprom = A15.

Les sorties CE (*Chip Enable*) sont les combinaisons des signaux d'adressage et de E qui est un signal de synchronisation des échanges de données du 68HC12 en mode étendu.

D'où :

- CE_RAM* = ram.E
- AEN* = cs.E
- CE_EEPROM * = eeprom.E

Les sorties OE *(Ouput Enable*) évitant les problèmes de contention sur le bus de données sont générées àpartir des CE et du signal de lecture.

D'où :

- OE_RAM* = R/W.ram.E
- OE_EEPROM* = R/W.eeprom.E
- R/W_RAM* = (R/W)*.E

Suivant l'état du signal PA (provenant du port E du 68HC12), ce sont soit les signaux MEM soit les signaux IO qui sont générés pour les accès en écriture ou en lecture du CS8900A

D'où :

- IOR* = cs.PA*(R/W).E
- IOW* = cs.PA*(R/W)*.E
- MEMR* = cs.PA(R/W).E
- MEMW* = cs.PA(R/W)*.E

Ces équations logiques sont représentées par le schéma logique suivant :

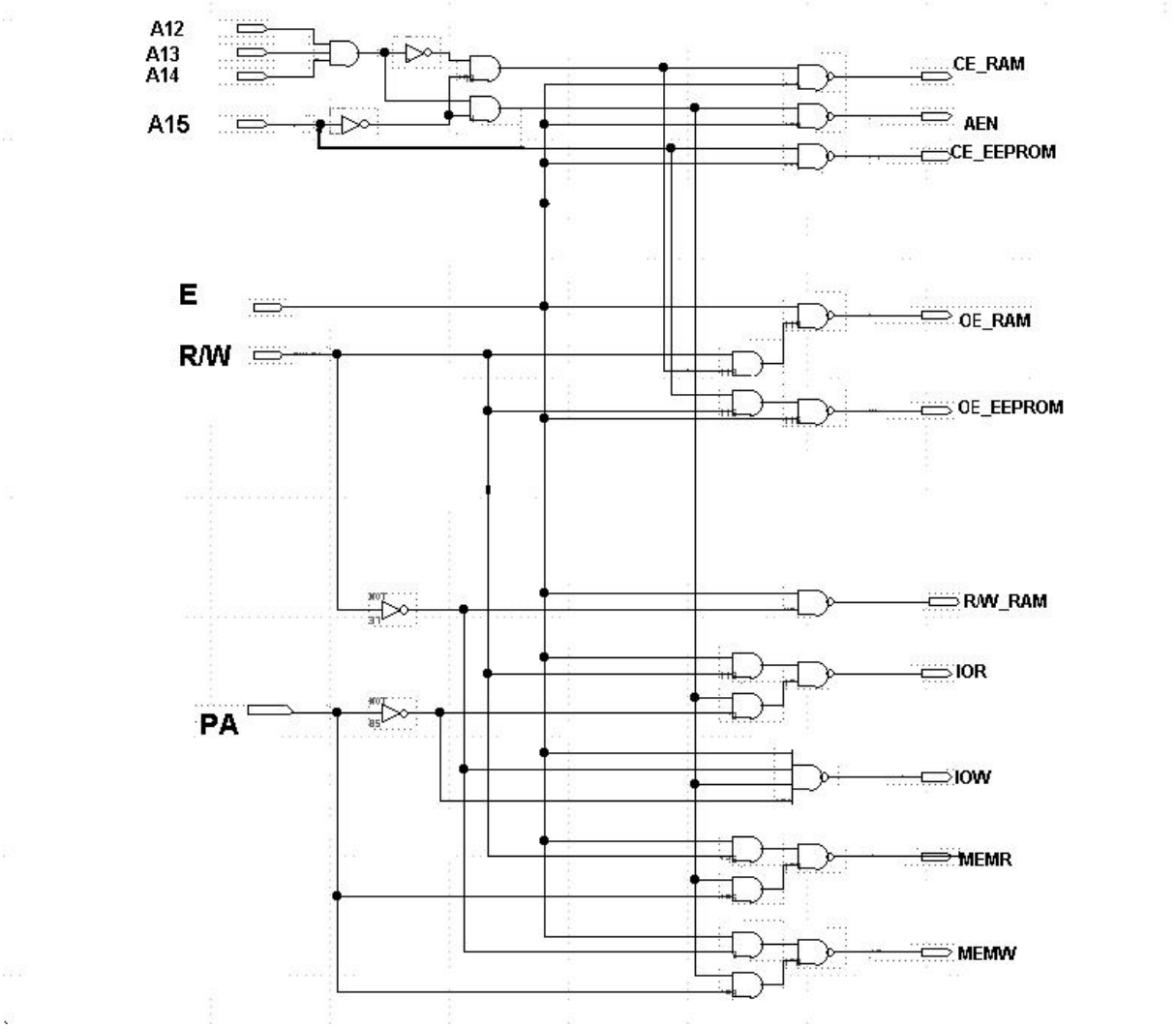


Figure 10: Schéma logique du décodage d'adresses

3.4.3 Utilisation du CS8900A :

Le chip CS8900A est optimisé dans sa conception pour un câblage sur des bus de type ISA Concernant notre cas de figure, le CS8900A doit pouvoir fonctionner en mode 16 bits), et connecté àbus système qui n'est pas ISA. Il faut donc trier les signaux susceptibles d'interfacer correctement le chip avec le microcontrôleur 68HC12 :

- Les 8 bits de données du chip SD0…SD7 sont reliés au port A du 68HC12.
- Les 8 bits de données du chip SD8…SD15 sont reliés au port B du 68HC12.
- Les adresses SA0…SA11 (adressage jusqu'à0FFF comprend facilement les 4Ko de données du chip Ethernet) connectées au bus d'adresse du 68HC12. (voir Packet Page Adresses)
- SA12…SA19 sont àla masse.
- ELCS* àla masse car LA non utilisé.

- EECS, EESK et EED Out non utilisés car pas de ROM.
- EEDI*, CHIPSEL* la masse.
- DMACK* en VCC car pas de DMA.
- CSOUT* non utilisé car pas de ROM de Boot.
- DO+/-, CI+/-, DI+/- non utilisés car l'AUI non utilisé.
- SLEEP àVCC car pas de Hardware Sleep ni Standby.
- RESET actif bas donc nécessité d'un inverseur àl'entrée. C'est le FPGA qui génère ce signal
- SBHE* est relie àSA0 étant donné qu'il nécessite après chaque reset une transition (0 à1 puis 1 à0) pour qu'on puisse passer au mode 16 bits.
- IOCHRDY* non utilisé.
- INTRQ0 utilisé pour le mode MEM.
- AEN* sert de chip select du CS8900A. C'est le EPM 7032 qui génère ce signal.
- MEMW*, MEMR*, IOR*, IOW* sont les signaux de lecture et d'écriture sur le chip suivant son mode de fonctionnement. C'est le EPM 7032 qui génère ces signaux.
- LANLED* et LINKLED* sont reliés àdes leds lumineuses.
- XTAL 1,XTAL2 sont les broches pour relier le quartz de 20MHz.
- MEMCS16* et IOCS16*sont des sorties non utilisés.
- RES connecté àune résistance de 4.99K vers la masse.
- BSTATUS* non utilisé
- TEST* non utilisé car pas de Boundary Scan Test.
- DVDD à5V numérique
- DVSS àla masse numérique.
- AVDD à5V analogique.
- AVSS àla masse analogique

3.5 Routage de la carte mère

Les 2 schémas de l'annexe 4 sont les 2 couches de la carte mère. La majorité des pistes a été travaillée manuellement.

Le routage a été effectue en utilisant les valeurs théoriques de conception suivantes:

Initialement:

Largeur des pistes : 0.012 inches
Clearance : 0.010 inches

Au voisinage de CS8900A: (en effet il impose un routage fin)

Largeur des pistes : 0.005 inches
Clearance : 0.008 inches

En effet, le CS8900A impose un routage fin étant donne la distance entre les broches.

3.6 Réalisation de la carte mère

Deux essais de réalisation de la carte mère comportant le CS8900A ont été fait au sein de l'Enseirb, en faisant appel au matériel dédié, malheureusement il s'est avéré impossible de réaliser cette carte étant donne la finesse de certaines piste qui sont déjààla limite du routage en ce qui concerne la largeur acceptée par le CS8900A.
Toutefois, il est possible d'exploiter le PCB réalisé lors de ce projet, pour fabriquer la carte àl'extérieur de l'Enseirb chez un fabricant de circuits imprimés, mais étant donne les contraintes de temps allant jusqu'à4 semaines, cela n'est pas possible lors de ce projet. Cela pourra être fait ultérieurement.
Pour cette raison, nous avons du renoncer àl'achèvement de cette carte mais en la remplaçant par une version simplifiée destinée àessayer la programmation du M68HC12 en intégrant tous les composants de la carte mère excepter le CS8900A et ses fournitures, et donc pas d'essais d'émission-réception de trames Ethernet qui seront remplacés par la validation des accès mémoire sur les RAM et les ROM une fois que le prototype sera achevé.

4. PARTIE SOFTWARE

4.1 Généralités

Durant ce projet nous avons vu une multitude de notions relatives aux réseaux, tels les réseaux Ethernet, nous avons également vu de près quelques notions enrichissantes sur le temps réel àtravers plusieurs ouvrages qui sont cités en annexe, toutefois, les quelques paragraphes qui suivent constituent une petite synthèse, qui résume le plus important.

4.2 Les Protocoles Internet

La partie Software a pour but l'implémentation des différentes couches réseau tels qu'elles sont décrites dans le modèle OSI (Open System Interconnection) en couches qui sont comme suit :

- La couche physique : qui représentée le réseau Ethernet.
- La couche liaison :sa fonction de signalisation d'erreurs de transmission par vérification du CRC est effectuée par le CS8900A.
- La couche réseau : dont les fonctions sont le routage, l'adressage, la détection et la correction d'erreurs sera implémentée par le protocole IP (Internet Protocol), elle comprend aussi le protocole ICMP (Internet Control Message Protocol).
- La couche transport :elle assure tout d'abord une communication de bout en bout en faisant abstraction des machines intermédiaires entre l'émetteur et le destinataire, elle s'occupe de réguler le flux de données et assure le transport de données selon deux possibilités :
    - Le protocole TCP (Transmisssion Control Protocol) : le transport est fiable car il s'agit d'un mode connecté.
    - Le protocole UDP (User Datagramme Protocol) : le transport et non fiable car il s'agit d'un mode non connecté. Dans ce cas c'est la couche application qui doit vérifier que le datagramme est arrivé àbon port.
- La couche session : elle fournit aux entités de la couche présentation les moyens d'organiser et synchroniser les dialogues et les échanges de données.
- La couche présentation :elle s'occupe de la syntaxe et de la sémantique des informations transportées en se chargeant notamment de la représentation des données.
- La couche application :elle donne au processus d'application le moyen d'accéder à l'environnement OSI et fournit tous les services directement utilisables par l'application :
    - Le transfert d'informations
    - L'allocation des ressources
    - L'intégrité et la cohérence des données accédées
    - La synchronisation des application coopérantes

  La couche application gère le programme utilisateur qui sera dans la mémoire du M68HC12. au niveau client, elle sera représentée par des programmes utilisateurs comme *telnet*, FTP, HTTP

Pour aboutir àce résultat, l'utilisation d'un noyau temps réel est très recommandée. Dans le paragraphe qui suit nous expliquerons succinctement le fonctionnement d'un noyau temps réel.

4.3 Le noyau temps réel µC/OSII

Le noyau temps réel µC/OSII(Micro Controller Operating System version II) a la particularité d'être distribué gratuitement. Il est portable sur un grand nombre de microprocesseurs de microcontrôleurs et de DSP. Ce noyau temps réel est programmé en C ANSI.
Il est prévu d'être implanté dans une ROM, et donc parfaitement adapté au but du projet.
C'est un noyau :

- Préemptif : il est possible d'interrompre une tache par une interruption et de reprendre son exécution une fois que la routine d'interruption est terminée.
- Déterministe : parce que les différentes taches ont des temps d'exécution biens définis.
- A Ordonnancement des taches : selon leurs priorités respectives qui peuvent varier dynamiquement.

De plus, il peut gérer jusqu'à62 taches .
La figure suivante montre les différents états que peut prendre une tache :

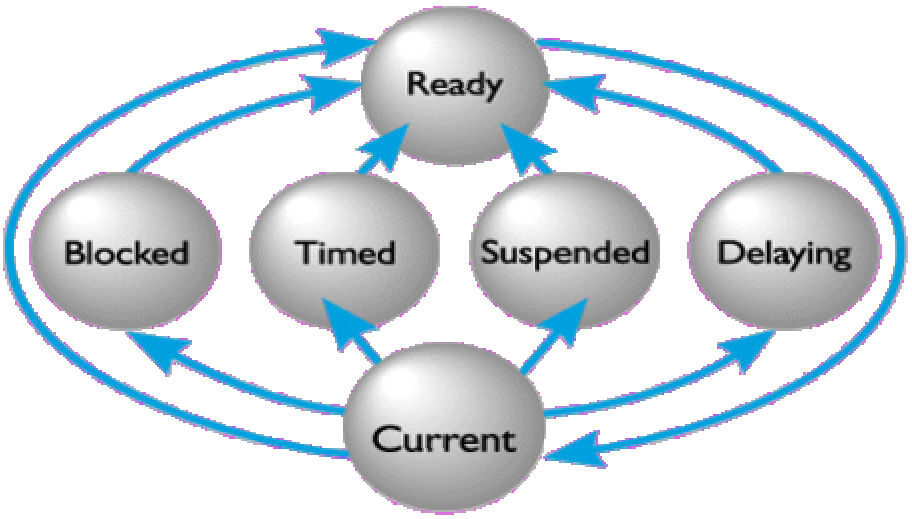


Figure 11: changement d'état d'une tache

4.4 Le temps réel en général :

Une des caractéristiques du temps réel est qu'il permet d'assurer un certain déterminisme entre deux utilisations successives d'un processeur par un processus. Il serait inacceptable pour un processus ayant encore besoin de temps machine après l'expiration de son quantum de temps, de disposer ànouveau du processeur après un délai incompatible avec le traitement qu'il àeffectuer.

La prise en compte de ce type de contraintes conduit àassocier une priorité au processus. La file d'attente des processus éligibles est alors triée par ordre de priorité croissante ou décroissante et le processus de plus forte priorité est élu. Ainsi un processus pourra s'assurer de la disponibilité du processeur tant qu'aucun processus de priorité supérieur ne le réclamera.

Généralement, les tâches les plus prioritaires sont associées aux traitements de défauts du système (panne, reconfiguration, alarme, etc..). Ensuite sont considérées les tâches associées aux événements matériels tels que les exceptions ou les interruptions. Enfin restent les tâches associées àla gestion des périphériques (tâches de service) et les tâches "utilisateur". Parmi ces dernières une plus forte priorité est généralement associée aux tâches les plus courtes. Par ailleurs, une règles de base selon cette approche est de minimiser les temps de traitements des tâches les plus prioritaires.

A la notion de priorité est attachée celle de rang de processus. Le seul aspect important est généralement la priorité relative des processus et non leur priorité absolue. Exécuter deux processus P1 et P2 de priorités 10 puis les exécuter de nouveau avec des priorités 100 conduit au même fonctionnement (si ces processus sont les seuls existant sur le système). Deux grandes méthodes sont universellement employées en environnement temps réel pour le partage du temps CPU entre processus :

La première méthode est basée sur la modification dynamique des priorités. Le principe est d'assurer que tout processus, quelle que soit sa priorité, finira àun moment ou àun autre par s'exécuter. Un circuit horloge est dans ce cas souvent nécessaire pour la gestion des incrément de priorités. L'avantage d'une telle approche est de procurer une certaine souplesse dans la gestion des tâches. Elle permet d'affiner cette gestion de telle sorte que l'on puisse par exemple exécuter une tâche trois fois plus souvent qu'une autre. En environnement multi-utilisateur, une telle méthode a encore l'avantage de moduler facilement les différents traitement des utilisateurs. Son inconvénient majeur réside dans le fait que l'exécution d'une tâche est fonction de la charge du système, c'est àdire du nombre et de la priorité des processus évoluant simultanément avec cette même tâche. Ceci nuit au critère déterministe du système. Il convient alors d'utiliser des palliatifs tels que les techniques de préemption (réquisition du processeur) pour répondre àce critère.

La deuxième méthode, plus simple dans sa philosophie, met justement l'accent sur l'aspect déterministe en stipulant que c'est toujours la tâche de plus haute priorité qui est exécutée. Ceci impose une gestion plus suivie et plus contraignante des tâches. Par définition, le déterminisme implique l'exécution d'un traitement dans un temps donné. Par conséquent, une tâche de priorité élevée, effectuant un long travail, gardera le processeur aussi longtemps qu'elle désire, bloquant ainsi toutes les autres tâches actives de priorité inférieure.

4.5 Réseaux Ethernet

Ethernet est le nom donné àune des technologies les plus utilisées pour les réseaux locaux en bus. Elle a été inventée par Xerox au début des années 70 et normalisée par l'IEEE (*Institute for Electrical and Electronics Engineers)* vers 1980 sous la norme IEEE 802.

Tout d'abord, il existe plusieurs technologies physiques pour établir un réseau Ethernet.

- *10 base 5 ou thick Ethernet* est un réseau àbase de câble coaxial de 1,27 cm de diamètre, d'une longueur de 500 m maximum et terminé àchaque extrémité par une résistance. Chaque ordinateur est relié, par un cordon AUI *Attachment Unit Interface*), àun boîtier appelé *transceiver* lui-même connecté au câble par l'intermédiaire d'une prise « vampire ». Le transceiver est capable de détecter si des signaux numériques transitent sur le câble et de les traduire en signaux numériques àdestination de l'ordinateur, et inversement.

- *10 base 2 ou thin Ethernet* est un réseau àbase d'un câble coaxial plus fin et plus souple, moins résistant aux perturbations électromagnétiques que le 10 base 5, mais d'un coût inférieur. Le transceiver et le câble AUI ne sont plus utiles car l'ordinateur est relié directement au câble par l'intermédiaire d'une prise BNC en T intégrée àla carte Ethernet de l'ordinateur.

- *10 base T ou twisted pair Ethernet* est un réseau dans lequel chaque ordinateur est relié, par un câble de type paire torsadée, àun point central appelé *hub* qui simule l'effet d'un transceiver et de son câble AUI. La connexion des câbles se fait par l'intermédiaire d'une prise RJ45 et les hubs doivent être alimentés électriquement. Ils simulent ainsi le fonctionnement d'un bus alors que la topologie physique du réseau est une étoile.

Majoritairement, les réseaux Ethernet ont un débit de 10Mbit/s et les informations sont transmises sur le bus sans garantie de remise. Chaque transceiver capte toutes les trames qui sont émises sur le câble et les redirige vers le contrôleur de l'ordinateur qui rejettera les trames qui ne lui sont pas destinées et enverra au processeur celles qui le concernent, c'est-àdire celles dont l'adresse de destination est égale àcelle de la carte réseau. Comme il n'y a pas d'autorité centrale qui gère l'accès au câble, il est possible que plusieurs stations veuillent émettre simultanément sur le câble. C'est pourquoi chaque transceiver écoute le câble pendant qu'il émet des données afin de détecter des éventuelles perturbations. Si une collision est détectée par le transceiver, celui-ci prévient le coupleur qui arrête d'émettre et attend un laps de temps aléatoire compris entre 0 et une certaine durée δ avant de re-émettre ses données. S'il y a encore un problème de collision, alors un nouveau temps d'attente est tiré au sort entre 0 et $3*\delta$, puis entre 0 et $4*\delta$, etc. .. jusqu'àce que la trame soit émise. Ce principe est justifié par le fait que si une première collision se produit, il y a de fortes chances que les délais d'attente tirés au sort par chacune des 2 stations soient très proches, donc il ne sera pas surprenant d'avoir une nouvelle collision. En doublant àchaque fois l'intervalle des délais d'attente possibles on augmente les chances de voir les retransmissions s'étaler sur des durées relativement longues et donc de diminuer les risques de collision. Cette technologie s'appelle CSMA/CD (*Carrier Sense Multiple Access with Collision Detect)*. Elle est efficace en

générale mais a le défaut de ne pas garantir un délai de transmission maximal après lequel on est sûr que la trame a été émise, donc cela ne permet pas de l'envisager pour des applications temps réel.
Les adresses physiques Ethernet sont codées sur 6 octets (48 bits) et sont censées être uniques car les constructeurs et l'IEEE gère cet adressage de manière àce que deux coupleurs ne portent pas la même adresse. Elles sont de trois types

- *unicast* dans le cas d'une adresse mono-destinataire désignant un seul coupleur
- *broadcast* dans le cas d'une adresse de diffusion générale (tous les bits à1) qui permet d'envoyer une trame àtoutes les stations du réseau
- *multicast* dans le cas d'une adresse multi-destinataire qui permet d'adresser une même trame àun ensemble de stations qui ont convenu de faire partie du groupe que représente cette adresse multipoint.

On voit donc qu'un coupleur doit être capable de reconnaître sa propre adresse physique, l'adresse de multicast, et toute adresse de groupe dont il fait partie.
Au niveau des trames, la normalisation IEEE 802 définit un format de trame légèrement différent de celui du véritable Ethernet. Ainsi, le RFC 894 définit les trames Ethernet et le RFC 1042 définit celles des réseaux IEE 802 comme illustré dans la figure suivante :

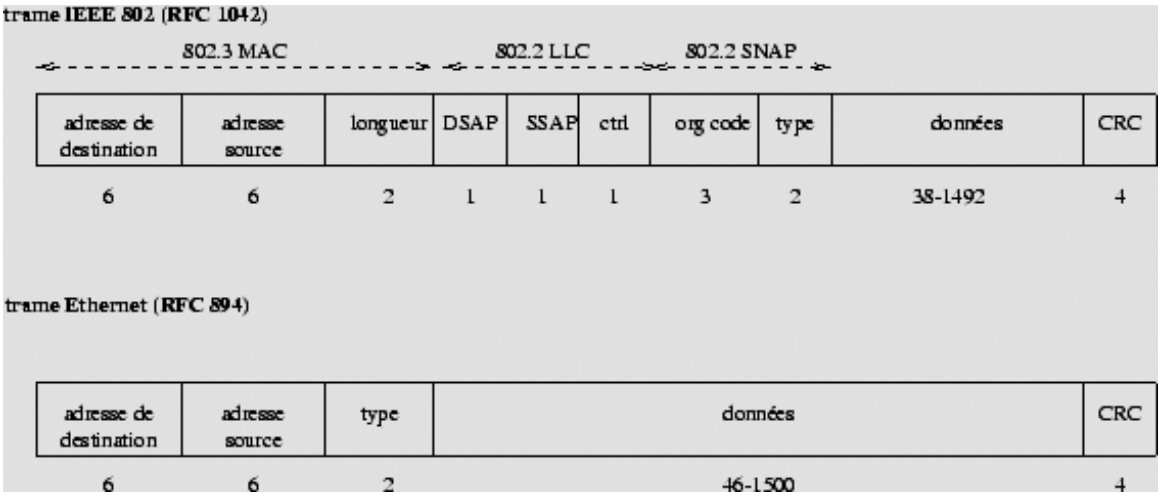


Figure 12:la trame Ethernet

Mais la variante la plus usitée est l'Ethernet.
Les deux trames utilisent des adresses matérielles source et destination de 6 octets (adresse Ethernet) et un CRC de 4 octets mais différent sur les points suivants.

- Dans le format Ethernet le troisième champ contient le *type* de données transmises selon que c'est un datagramme IP, une requête ou réponse ARP ou RARP. Puis, viennent les données transmises qui peuvent avoir une taille allant de 46 à1500 octets. Dans le cas de données trop petites, comme pour les requêtes et réponse ARP et RARP (voir la sous-section) on complète avec des *bits de bourrage* ou *padding*.

- Dans le format IEEE 802, le troisième champ indique le nombre d'octets de la trame sans compter le CRC. Étant donné qu'aucune des valeurs possibles pour le champ type de la trame Ethernet ne peut représenter une longueur de trame, ce champ peut permettre de distinguer les encapsulations. Pour la sous-couche LLC le champ DSAP (*Destination Service Access Point*) désigne le ou les protocoles de niveau supérieur àqui sont destinées les données de la trame et le champ SSAP (*Source Service Access Point*) désigne le protocole qui a émis la trame. Ici leur valeur hexadécimale est AA, c'est-à-dire la valeur désignant le protocole SNAP (*Sub-Network Access Protocol*). Le champ de contrôle *ctrl* est mis égal à3 et les 3 octets du champ *org code* sont mis à0. Ensuite, on trouve le champ *type* qui a la même signification que celui de la trame Ethernet.

De nombreux équipements matériels interviennent dans la constitution physique d'un réseau Ethernet, ce paragraphe décrit quelques uns de ceux qui interviennent aux niveaux 1 et 2 du modèle OSI.

- Un *répéteur* opère de manière physique uniquement, donc au niveau de la couche 1 du modèle OSI. Il se contente de retransmettre et d'amplifier tous les signaux qu'il reçoit, sans aucun autre traitement. Un « hub » est un répéteur 10 base T multiport qui renvoie donc le signal qu'il reçoit par l'un de ses ports vers tous ses autres ports.

Un *pont* est un équipement qui intervient dans l'architecture d'un réseau en reliant deux segments disjoints de ce réseau. Le pont appartient àla couche 2 du modèle OSI car il va filtrer les trames du réseau en fonction de leur origine et destination, mais il ne se préoccupe pas du logiciel réseau de niveau supérieur (TCP/IP, DECNet, IPX, ...).

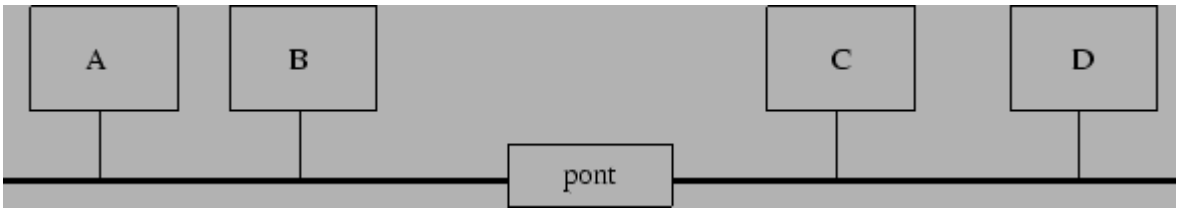


Figure 13: Fonctionnement d'un pont

- Dans la configuration de la figure le pont sera capable de déterminer que les ordinateurs A et B sont sur le segment 1 et les ordinateurs C et D sur le segment 2. Il peut obtenir ces informations car il « voit passer » toutes les trames provenant des ordinateurs appartenant aux deux segments qu'il relie et grâce aux adresses d'origine contenues dans les trames, il peut se construire une table d'adresses mémorisant la cartographie du réseau. Ainsi, si une trame est envoyée de A vers B, ou de C vers D, elle ne franchira pas le pont car celui-ci aura détecté que c'est inutile. Mais si la trame provenant de A est destinée àC ou D, elle le traversera sans aucun autre traitement.

- L'utilisation d'un pont peut ainsi améliorer le débit d'un réseau car toutes les trames ne sont pas transmises sur tout le réseau. D'autre part, cela peut permettre d'augmenter la confidentialité du réseau en isolant certains ordinateurs des autres de manière àce que certaines trames soient impossibles àcapturer par des ordinateurs « espions » collectionnant toutes les trames qui circulent sur le réseau, même celles qui ne lui sont pas destinées.
- Un *commutateur* est en fait un pont multiport qui va aiguiller chacune des trames qu'il reçoit vers le segment sur lequel se trouve l'ordinateur de destination de la trame. Cependant, chacun de ses ports est habituellement relié àun segment contenant un nombre restreint d'ordinateurs, voire àun seul s'il s'agit par exemple d'un serveur très sollicité.

4.6 Configuration

4.6.1 Transmission d'une trame

4.6.1.1 Utilisation des registres

Avant tout fonctionnement du CS8900A, on effectue une initialisation des registres de contrôle de transmission RxCTL, TestCTL et LineCTL.

Une fois tout ces registres configurés, on écrit dans le registre TxCMD la valeur 0x00C0, ce qui permet la transmission après que tout le bits soient écrit dans le mémoire tampon du CS8900A, on écrit ensuite la longueur de la trame àtransmettre dans le registre TxLENGTH dont l'adresse est 0x7306. On configure le registre BusST pour permettre la transmission, et enfin on effectue une série d'écriture sur les registres 0x7300 puis 0x7301 de la trame àtransmettre. Une fois finie, le composant effectue la transmission automatique de trame.

4.6.2 Réception d'une trame

4.6.2.1 Utilisation des registres

Avant tout fonctionnement du CS8900A, on effectue une initialisation des registres de contrôle de transmission RxCTL, TestCTL, LineCTL et BufCFG.

Une fois tout ces registres configurés, on scrute le registre RxEvent indiquant l'état du CS8900A (attente/réception), puis ensuite on lit le statut/longueur_trame/données par une suite de lecture 8bits des registres d'adresses 0x7300 et 0x7301.

4.6.3 Transmission/Réception en utilisant l'interruption du microcontrôleur 68HC12

Le principe consiste àse mettre dans une première phase dans un mode de transmission pendant lequel le CS8900A envoie une trame dans le réseau Ethernet, puis ensuite une deuxième phase dans un mode de réception. Cette dernière phase est la plus intéressante puisqu'elle fonctionne sous interruption. En effet une fois le CS800A, ayant reçu une trame il envoie un signal d'interruption vers le microcontrôleur pour ainsi exécuter le programme associé au mode de réception.

4.7 Un étude comparative entre 68HC11 et 68HC12

Un des buts importants visés lors de la conception du 68HC12 était de maintenir une compatibilité avec le 68HC11 ce qui permet une migration « en douceur » vers le HC12. cette compatibilité a été réalisée en conservant :

- le même model de programmation :
  - Accumulateur A (8 bits)
  - Accumulateur B (8 bits)
  - Accumulateur D (16 bits)
  - Registre d'index IX (16 bits)
  - Registre d'index IY (16 bits)
  - Pointeur de pile SP (16 bits)
  - Compteur Programme (16 bits)
  - Registre CCR S X H I N Z V C
- la même définition fonctionnelle de toutes les instructions du 68HC11.

5. CONCLUSION

ce stage ENSEIRB nous a permis de mieux comprendre le fonctionnement d'une carte à base microcontrôleur surtout qu'il s'agit d'un M68HC12 qui est la nouvelle génération de microcontrôleurs 16 bit qui viennent remplacer les microcontrôleurs 8 bits, certes efficaces, mais offrant moins d'avantages que se soit en puissance ou en souplesse. Ce stage nous a permis de revoir des notions et des techniques importantes tels le routage, soudure, impression sur cuivre de circuit imprimé…
Il s'agit donc d'un apprentissage précieux pour l'avenir, surtout que la conception de systèmes autonomes à base de microcontrôleur semble être vouée à une ascension certaine. De plus, ce stage nous a permis de revoir et mieux comprendre les protocoles internet.
Le seul regret est que les délai de réalisation du matériel nous ont empêché de terminer entièrement le projet, toutefois, le reste du projet, c'est à dire la partie logicielle proprement dite pourrais constituer un sujet de stage pour l'avenir.

6. GLOSSAIRE

BDM	Background Debug Mode
TCP	Transmission Control Protocol
UDP	User Datagramme Protocol
IP	Internet Protocol
ICMP	Internet Control Message Protocol
OSI	Open System interconnection
CRC	Cyclic Redundancy Check
DSP	Digital Signal Processor
PCB	Printed Circuit Board
ISA	Industry Standard Architecture
LAN	Local Area Network
CAN	Controller Area Network
CSMA/CD	Carrier sense Multiple Access with Collision Detection
MAC	Media Access Control

7. BIBLIOGRAPHIE

- www.HC12Web.de
- www.enseirb.fr/~kadionik
- Le temps réel : cours de l'école nationale supérieure d'informatique.
- CIRRUS LOGIC : Datasheet du CS8900A.
- MOTOROLA : Datasheet du M68HC12B.
- Cours de réseaux de l'université d'Angers.
- Application Note 83-3.

8. ANNEXES

Annexe1 : Le MOTOROLA M68HC12

Annexe 2: Nomenclature de la carte mère

Annexe 3 : Le CS8900A

Annexe 4 : Le PCB de la carte mère

Annexe 1: LE MOTOROLA 68HC12

Generalités

Le 68HC12 inclut :

- une CPU12 traitant les donnes en mode 16 bits :
  - le jeu d'instruction est compatible avec celui des 68HC11
  - la pile àinterruptions ainsi que le model de programmation est identique àcelui du M68HC11
  - Une unité arithmétique et logique (ALU) de 20 bits.
  - Une file d'instructions
  - Un adressage indexé plus performant
  - Des instructions Fuzzy Logic
- Un bus multiplexé :
  - Single chip ou extended
  - 16 bits par 16 bits en mode large ou bien 16 bits par 8 bits en mode étroit.
- Mémoire :
  - 32 Koctets de FLASH contenant 2Koctets de EEPROM
  - 32Koctets de ROM
  - 768 octets de EEPROM
  - 1 Koctet de RAM avec un seul cycle d'accès pour les lectures écritures alignées ou non alignées.
- 8 canaux pour conversion Analogique Numérique sur 10 Bits.
- 8 canaux de module timer standard
- Enhanced capture timer ECT
- Accumulateur d'impulsions 16 bit :
  - Comptage d'événements externes
- Modulateur àlargeur d'impulsions PWM
  - 8 bit avec 4 canaux ou bien 16 bit avec 2 canaux
- Interface Série :
  - Interface de communication série asynchrone (SCI)
  - Interface périphérique de communication série asynchrone (SPI)
  - Bus de terrain CAN : Contoller Area Network
- Un Timer Watchdog
- Diviseur d'horloge pour mode lent
- Boîtier plat avec pattes coudées à80 broches.
- Jusqu'à63 ports d'entrées sorties
- Port (BDM) : Background Debug Mode
- Points d'arrêts câblés en hardware

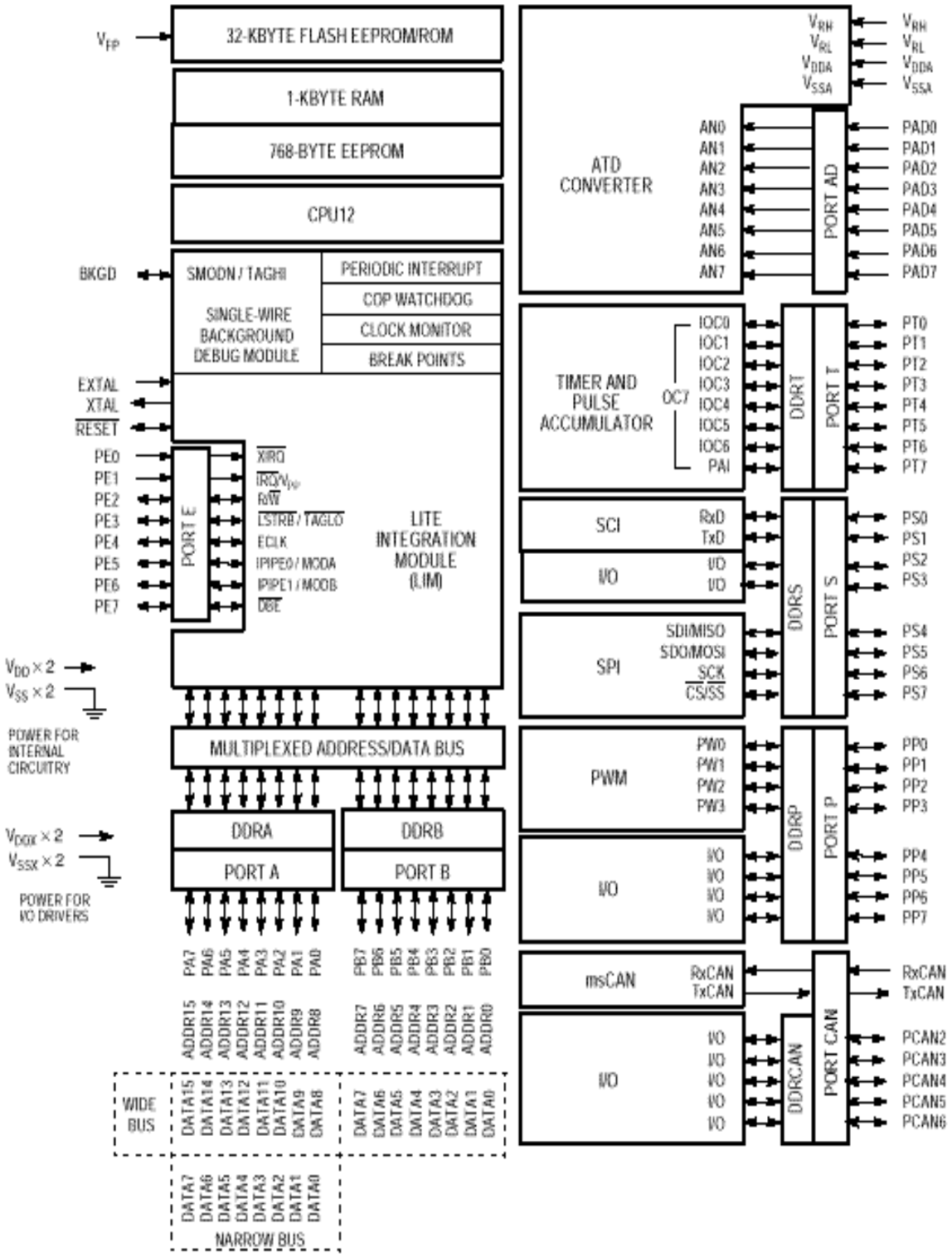


Figure 14: diagramme de blocs pour le M68HC(9)12BC32

Brochage

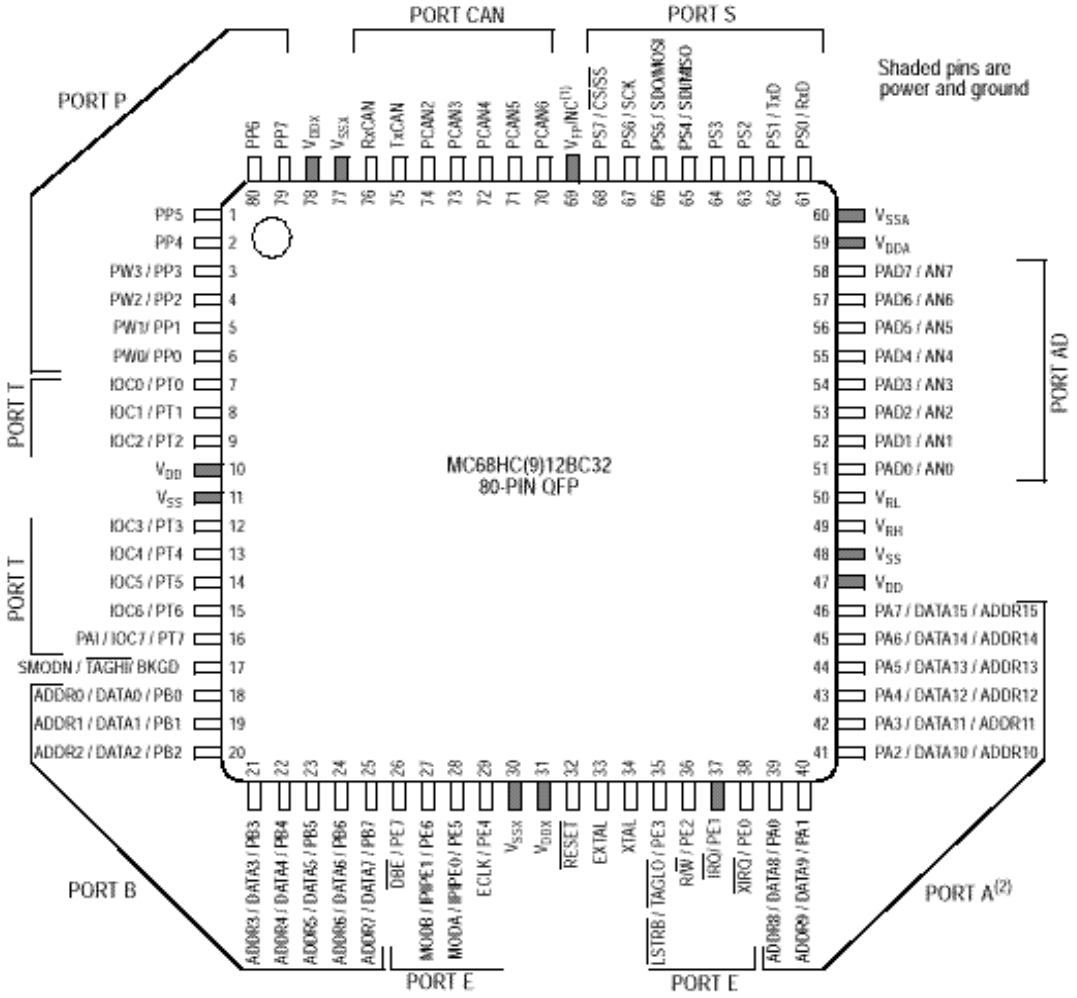


Figure 15 :Assignation des broches pour le MC68HH(9)12BC32

MOD A (28) et MOD B (27) BKGD (17): Les niveaux sur ces trois broches permettent d'initialiser le mode de fonctionnement du uC après le Reset. Par default, si on ne les impose pas les valeurs sont les suivantes : MOD A=0 MOD B=0 BKGD=1. dans les autres cas le tableau suivant résume les différents modes opératoires.

BKGD	MOD B	MOD A	MODE	Port A	Port B
0	0	0	Special Single Ship	I/O	I/O
0	0	1	Special expanded narrow	ADDR[15:8] ou DATA[7:0]	ADDR[7:0]
0	1	0	Special Peripheral	ADDR ou DATA	ADDR ou DATA
0	1	1	Special expanded wide	ADDR ou DATA	ADDR ou DATA
1	0	0	Normal Single Ship	I/O	I/O
1	0	1	Normal expanded narrow	ADDR[15:8] ou DATA[7:0]	ADDR[7:0]
1	1	0	Réservée	--	--
1	1	1	Normal expanded wide	ADDR ou DATA	ADDR ou DATA

Figure 16: les modes de fonctionnement du M68HC12

On dispose ainsi de 3 modes de fonctionnement car les modes spéciaux « special » sont utilisés lors des tests en usine. Ainsi on dispose des modes suivants :

- Normal Single Ship
- Normal expanded narrow ou mode étendu étroit
- Normal expanded wide ou mode étendu large

Mapping mémoire

Selon les modes de fonctionnement, on a les cartographies mémoire suivantes :

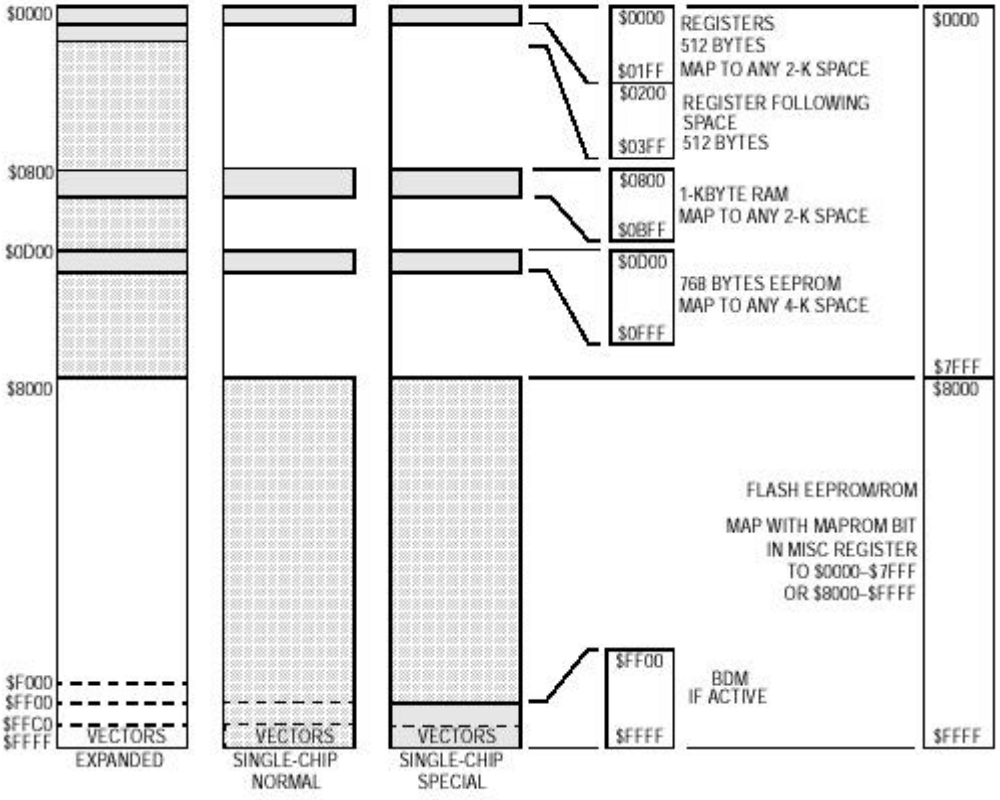


Figure 17: mapping de la mémoire du M68HC12 selon les modes

PA0…PA7 (39 …46) : Ce sont des sorties lorsque le uC fonctionne en Single Chip.

En Etendu, ces broches véhiculent le poids fort (A8…A15) du bus d'adresses et les 8 bits de données (D8…D15) du bus de données. (Technique de multiplexage, le démultiplexage est confié au circuit Latch 74LS573)

PB0…PB7 (18…25) : Ce sont des sorties lorsque le uC fonctionne en Single Chip.

En Etendu, ces broches véhiculent le poids faible du bus d'adresses (A0…A7). et les 8 bits de données (D0…D7) du bus de données. (Technique de multiplexage, le démultiplexage est confié au circuit Latch 74LS573)

PE0…PE (38…35, 29…26) : Ce port comporte 8 entrées soties.

IRQ* (37) : Entrée d'interruptions matérielles masquables.

XIRQ* (38) : Entrée d'interruptions matérielles non masquables..

ECLK (29), XTAL (34) et EXTAL (33) : les Broches concernant l'horloge.

ECLK est une sortie horloge (fréquence du Quartz / 2) permettant la synchronisation des échanges avec des composants extérieurs en mode Etendu. elle est aussi utilisée comme référence de temps pour démultiplexer les données et les adresses.

R/W*(38): En Mode Etendu c'est le signal R/W* sur le bus de données.

En Single Chip, c'est une entrée de ligne d'échange permettant la synchronisation des échanges sur le port C ou comme une entrée de détection d'évènements sensible sur front générant une requête d'interruption.

LSTRB (35) : dans tous les modes, cette broche peut être utilisée comme entrée sortie.

DBE* (26):en mode étendu, si il est utilisé, permet le contrôle du bus de données.

RESET* (32) : Reset sur un niveau bas du microprocesseur.

PS0…PS7 (61…68) :il s'agit d'une interface 8 bits pour l'interface série standard qui consiste en l'interface de communication série (SCI) et l'interface périphérique série (SPI). Elle peuvent aussi être utilisées comme entrée-sorties parallèles.

RxD (61) : Broche de Réception lorsque l'interface SCI est utilisée.

TxD (62) : Broche de Transmission lorsque l'interface SCI est utilisée.

MISO (65) : Lorsque l'interface SPI est utilisée, cette broche est configurée soit en entrée mode maître ou sortie mode esclave.

MOSI (66) : Lorsque l'interface SPI est utilisée, cette broche est configurée soit en sortie mode maître ou entrée mode esclave.

CS*/SS* (68) : Lorsque l'interface SPI est utilisée, cette broche permet de sélectionner les esclaves.

SCK (67) : Lorsque l'interface SPI est utilisée, cette broche véhicule le signal d'horloge fourni par le maître. Cette broche devient une entrée sur les esclaves.

VrH (49) et VrL (50) : Entrées de référence de tension (entre Vdd et Vss) utilisées par le CAN.

Vrl est le niveau bas de référence.

Vrh est le niveau haut de référence et doit être de 3V supérieur àVrl

PAD0…PAD7(51…58) : Ce port comporte 8 entrées pour véhiculer des signaux analogiques et les convertir en signaux numériques par échantillonnage.

PCAN[6 :2] : sont des broches I/O, ils font partie du bus CAN . il a été conçu pour être utilise comme un bus de donnée série, il englobe les spécificités de certains domaines comme :

- Traitement en temps réel
- Efficacité opératoire dans un environnement comportant des interférences électromagnétiques comme dans un véhicule par exemple.
- Efficacité des coûts
- Bande passante requise

Le msCAN 12 ou Motorola scalable Controller Area Network, simplifie les applications logicielles en utilisant un arrangement de tampons(buffer) avancé, ce qui conduit àun comportement en temps réel prédictible.

TxCAN (75) : broche de transmission

RxCAN (76) : broche de réception

Annexe 2: Nomenclature de la carte mère

```
# Board Station BOM file
# date : Sunday June 16, 2002; 18:28:53
```

ITEM_NUMBER	COMPANY PART NO.	GEOMETRY	COUNT	DESCRIPTION	REFERENCE
1	PN-EPM7032SLC44-10-PLCC	PLCC44	1	EPM7032S_44_10	U2
2	PN-MD27C256-20-DIP	DIP28_C	2	27C256_20	U10 U11
3	PN-SN74HC573N-DIP	DIP20_P	2	74HC573	U5 U6
4	PN-TC55257BSPL-10-DIP	DIP28_P	2	55257B_10	U8 U9
5	pn-10baset	DIP16_P	1	10BASE-T	U7
6	pn-CS8900A	tqfp100	1	CS8900A	U3
7	pn-cap_1u	ck01	21	CAPACITOR, 0.01	C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15 C16 C17 C18 C24 C25 C26 C27 C28 C29 C30
8	pn-chim_11x5mm	c_chim_2	4	POL_CAPACITOR, 1u	C1 C2 C3 C4
9	pn-conn1x8_s	conn1x8_s	1	conn8	J3
10	pn-conn1x9_s	conn1x9_s	1	conn9	J1
11	photodiode	do50	2	DIODE	D1 D2
12	pn-douille_2mm_ci	plk1x240	2	plot	J4 J5
13	pn-g-subd25f_2.54	subd25f_2.54	1	conn25	J2
14	pn-max232	DIP16_P	1	max232	U1
15	pn-quartz_01	hc18	1	quartz, 20MHz	U4
16	pn-res_01_4u_1/4W	RC05	8	RESISTOR, 4.7k	R1 R2 R3 R4 R5 R6 R7 R8

Moukmir marouane Chara youness	Département : électronique Option :informatique industrielle	Juin 2002

REFERENCE	ITEM_NUMBER	COMPANY PART NO.	GEOMETRY	DESCRIPTION
C1	8	pn-chim_11x5mm	c_chim_2	POL_CAPACITOR, 1u
C2	8	pn-chim_11x5mm	c_chim_2	POL_CAPACITOR, 1u
C3	8	pn-chim_11x5mm	c_chim_2	POL_CAPACITOR, 1u
C4	8	pn-chim_11x5mm	c_chim_2	POL_CAPACITOR, 1u
C5	7	pn-cap_1u	ck01	CAPACITOR, 1u
C6	7	pn-cap_1u	ck01	CAPACITOR, 1u
C7	7	pn-cap_1u	ck01	CAPACITOR, 0.1u
C8	7	pn-cap_1u	ck01	CAPACITOR, 0.1u
C9	7	pn-cap_1u	ck01	CAPACITOR, 0.1u
C10	7	pn-cap_1u	ck01	CAPACITOR, 68p
C11	7	pn-cap_1u	ck01	CAPACITOR, 0.1u
C12	7	pn-cap_1u	ck01	CAPACITOR, 0.1u
C13	7	pn-cap_1u	ck01	CAPACITOR, 1u
C14	7	pn-cap_1u	ck01	CAPACITOR, 1u
C15	7	pn-cap_1u	ck01	CAPACITOR, 0.1u
C16	7	pn-cap_1u	ck01	CAPACITOR, 0.1u
C17	7	pn-cap_1u	ck01	CAPACITOR, 0.01
C18	7	pn-cap_1u	ck01	CAPACITOR, 0.01u
C24	7	pn-cap_1u	ck01	CAPACITOR, 0.01
C25	7	pn-cap_1u	ck01	CAPACITOR, 0.01
C26	7	pn-cap_1u	ck01	CAPACITOR, 0.01
C27	7	pn-cap_1u	ck01	CAPACITOR, 0.01
C28	7	pn-cap_1u	ck01	CAPACITOR, 0.01
C29	7	pn-cap_1u	ck01	CAPACITOR, 0.01
C30	7	pn-cap_1u	ck01	CAPACITOR, 0.01
D1	11	photodiodediode	do50	DIODE
D2	11	photodiodediode	do50	DIODE
J1	10	pn-conn1x9_s	conn1x9_s	conn9
J2	13	pn-g-subd25f_2.54	subd25f_2.54	conn25
J3	9	pn-conn1x8_s	conn1x8_s	conn8
J4	12	pn-douille_2mm_ci	plk1x240	plot
J5	12	pn-douille_2mm_ci	plk1x240	plot
R1	16	pn-res_01_4u_1/4W	RC05	RESISTOR, 680
R2	16	pn-res_01_4u_1/4W	RC05	RESISTOR, 680
R3	16	pn-res_01_4u_1/4W	RC05	RESISTOR, 10K

R4	16	pn-res_01_4u_1/4W	RC05	RESISTOR, 4.99K
R5	16	pn-res_01_4u_1/4W	RC05	RESISTOR, 100
R6	16	pn-res_01_4u_1/4W	RC05	RESISTOR, 23.4
R7	16	pn-res_01_4u_1/4W	RC05	RESISTOR, 23.4
R8	16	pn-res_01_4u_1/4W	RC05	RESISTOR, 4.7k
U1	14	pn-max232	DIP16_P	max232
U2	1	PN-EPM7032SLC44-10-PLCC	PLCC44	EPM7032S_44_10
U3	6	pn-CS8900A	tqfp100	CS8900A
U4	15	pn-quartz_01	hc18	quartz, 20MHz
U5	3	PN-SN74HC573N-DIP	DIP20_P	74HC573
U6	3	PN-SN74HC573N-DIP	DIP20_P	74HC573
U7	5	pn-10baset	DIP16_P	10BASE-T
U8	4	PN-TC55257BSPL-10-DIP	DIP28_P	55257B_10
U9	4	PN-TC55257BSPL-10-DIP	DIP28_P	55257B_10
U10	2	PN-MD27C256-20-DIP	DIP28_C	27C256_20
U11	2	PN-MD27C256-20-DIP	DIP28_C	27C256_20

Annexe3: Présentation du CS8900A

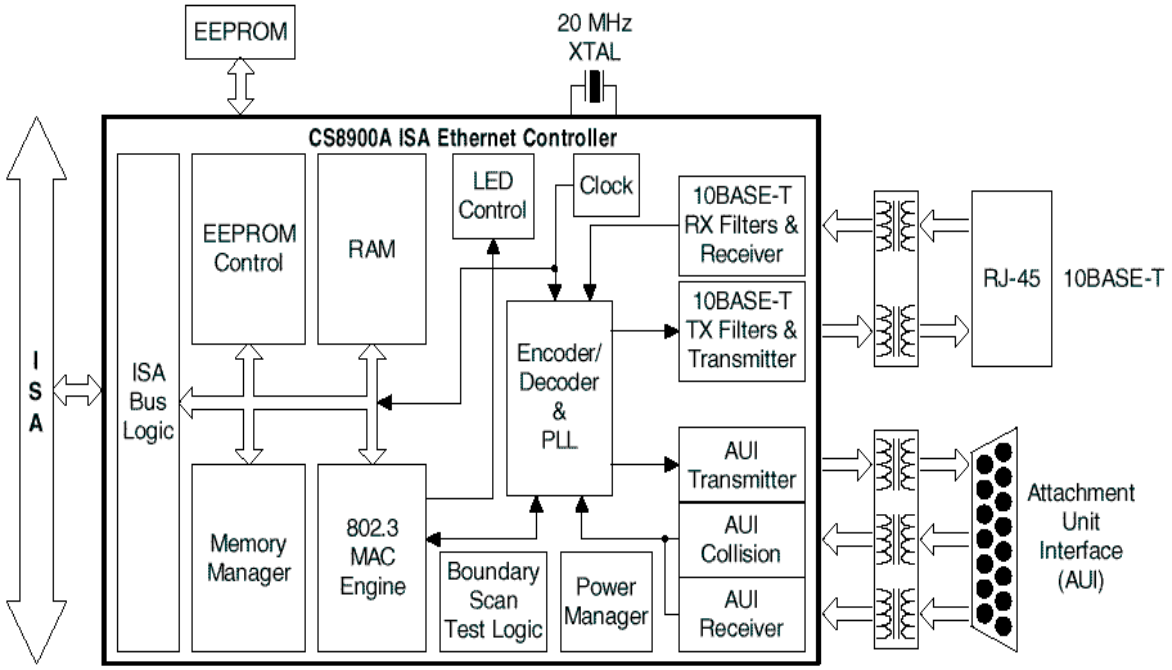


Figure 18: Schéma bloc du CS8900A

Interface complète pour bus ISA (Industry Standard Architecture).

4 lignes d'interruption et 3 canaux DMA .

Contrôleur Ethernet 16 bits.

Modes de fonctionnement en espace I-O ou PacketPage.

4Ko de RAM intégré dans le Chip.

Unité MAC (Media Access Control) incorporée gérant les transmissions et réceptions de trames (collision, erreurs).

Interface EEPROM permettant une configuration personnalisée au démarrage sur EEPROM externe.

Interface bout de ligne entièrement analogique avec 10Base-T et AUI (Attachment Unit Interface).

Package TQFP permettant le design de cartes occupant une surface inférieure à10cm carré.

Design possible du chip sur carte mère ou carte d'adaptation.

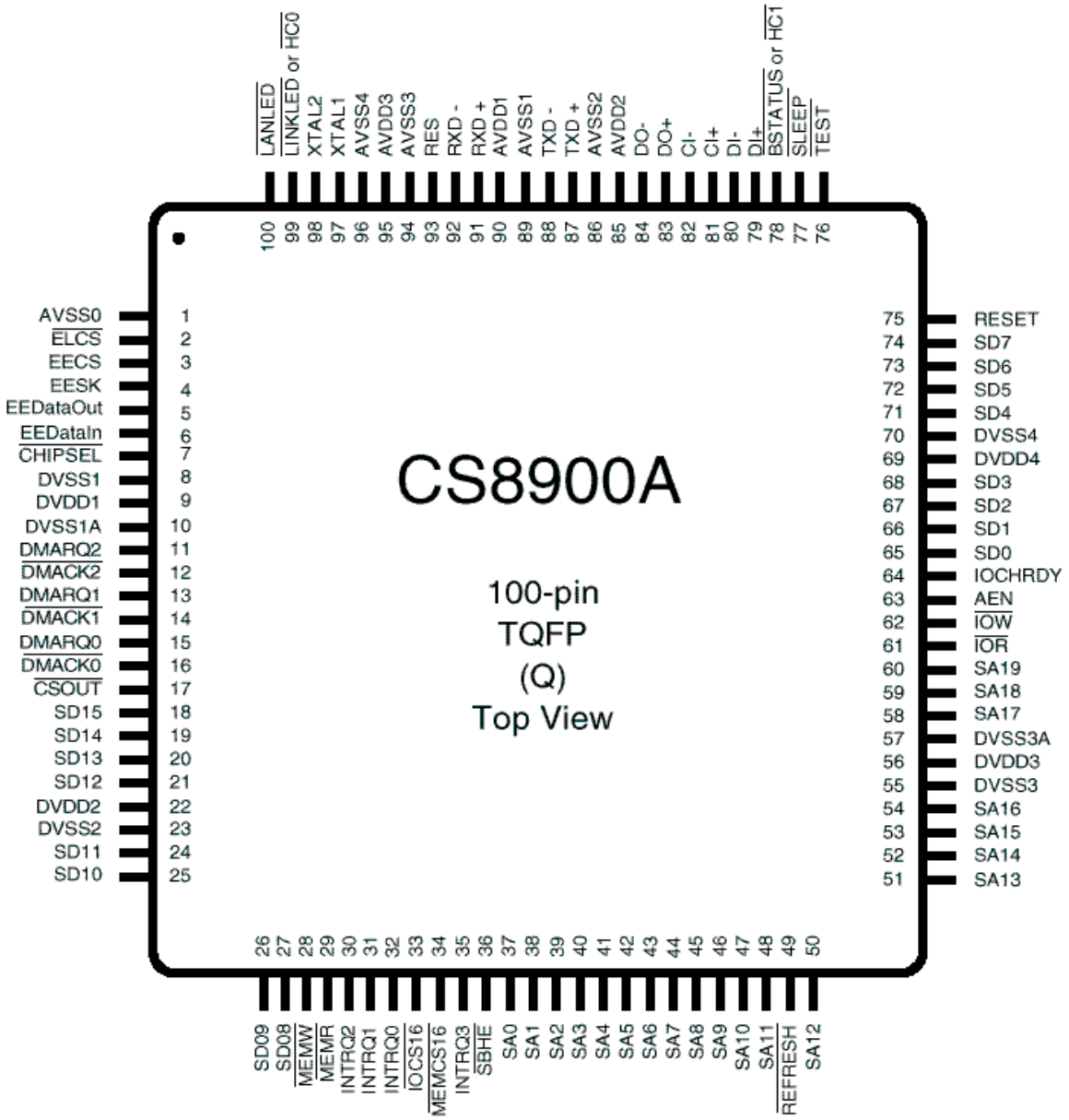


Figure 19: Assignation des broches du CS8900A

Le point supérieur gauche du composant réel avec l'angle cassé permet d'orienter le chip.

Brochage

Description rapide des Broches du CS8900A pour la conception Hardware:

Interface du Bus ISA :

SA0…SA19 : Bus d'adresse du système pour décodage des accès aux espaces I/O ou Memory.

SD0…SD15 : Bus bidirectionnel de données format 16 bits.

RESET : Broche d'initialisation asynchrone actif sur état HAUT. – maintien minimal de 400ns –

AEN : Entrée Adresse Enable.

Si Test* est HAUT, cette entrée indique au CS8900Aque le contrôleur du système DMA prend le contrôle du bus ISA .

MEMR* : Memory Read. Entrée active sur état BAS indique que l'on exécute une opération de lecture.

MEMW* : Memory Write. Entrée active sur état BAS indique que l'on exécute une opération d'écriture.

MEMCS16* : Sortie pour le mode MEM 16bits

REFRESH* : Entrée pour indiquer qu'un cycle de rafraîchissement DRAM est en progression.

Lorsque REFRESH* est BAS, MEMR*, MEMW*, IOR*, IOW*, DMACK1* et DMACK2* sont ignorées.

IOR* : I/O Read. Indique une sortie du contenu du registre I/O 16 bits sur le bus de données du système lorsqu'une adresse valide est détectée.

Lorsque REFRESH* est BAS, IOR* est ignorée.

IOW* : I/O Write. Le CS8900 écrit les données du bus de données vers le registre I/O 16 bits lorsqu'une adresse valide est détectée.

Lorsque REFRESH* est BAS, IOW* est ignorée.

IOCS16* : Le CS8900A génère cette sortie BAS lorsqu'il reconnaît une adresse du bus ISA qui correspond àl'espace I/O assigné sinon sortie en 3 e état.

SBHE* : Entrée indiquant un transfert sur l'octet fort du bus de données (SD8…SD15), en effet au Reset le Chip se place dans une configuration 8 bits pour passer en mode 16 bits il faut au moins une transition sur ce signal (0→1 puis 1→0).

INTRQ0…INTRQ2 : Ces sorties indiquent la présence d'un événement d'interruption.

DMARQ0…DMARQ2 : Sortie HAUT indiquant que le CS8900A demande un transfert DMA. Seule une sortie est utilisée àla fois, les autres sont en 3 e état.

DMACK0…DMACK2 : Entrée indiquant la reconnaissance par l'hôte de la correspondance de la sortie de demande DMA

CHIPSEL* : Entrée générée par un décodeur logique d'adresse latchable externe lorsque une adresse mémoire valide est présente sur le bus ISA.

Si le Mode Memory n'est pas utilisé, CHIPSEL* doit être tirée BAS.

CHIPSEL* est ignorée en mode I/O et DMA du CS8900A.

Interface EEPROM et PROM de Boot : (non utilkisée lors de ce projet)

EESK : Horloge Série utilisée pour les transferts de données vers l'EEPROM.

EECS : Sortie pour selectionner l'EEPROM

EEDataIN : Entrée Série pour la réception des données de l'EEPROM connectée àla broche DO de l'EEPROM.

Permet également de sonder la présence de l'EEPROM.

ELCS* : Signal bidirectionnel pour configurer un décodeur logique d'adresse externe latchable.

Si le LA n'est pas utilisé, ELCS* doit être tiré vers BAS.

EEDataOut : Sortie série pour l'envoie de données vers l'EEPROM, connectée àla broche DI de l'EEPROM.

Si TEST* est BAS, elle devient la sortie pour le Boundary Scan Test.

CSOUT* : Sortie pour sélectionner une PROM de Boot externe lorsque le CS8900A décode une adresse mémoire valide de la PROM de Boot.

Interface 10 Base-T :

TXD+, TXD- : Paire différentielle de sortie 10Mb/s pour la transmission de données codées en Manchester vers le 10Base-T.

RXD+, RXD- : Paire différentielle d'entrée 10Mb/s pour la réception de données codées en Manchester vers le 10Base-T.

InterfaceAUI (Attachment Unit Interface)

DO+, DO- : Paire différentielle de sortie 10Mb/s pour la transmission de données codées en Manchester vers l'AUI.

DI+, DI- : Paire différentielle d'entrée 10Mb/s pour la réception de données codées en Manchester vers l'AUI.

CI+, CI- : Paire différentielle d'entrée connectée àla paire de collision de l'AUI.

Pins Généraux :

XTAL1…XTAL2 : Pour connexion d'un quartz de fréquence 20MHz.

Si un signal de 20MHz est utilisé au lieu du quartz, il est relié àXTAL1 et XTAL2 libre.

SLEEP* : Entrée permettant l'activation des modes de mise en veille hardware (Suspend et Hardware Standby).

LINKLED* ou HC0* : Sortie pour la détection d'impulsions de connexion valide.

BSTATUS* ou HC1* : Sortie indiquant une activité sur le bus ISA.

LANLED* : Sortie indiquant l'arrivée, la transmission de paquets pour une collision.

TEST* : Entrée utilisée pour mettre le CS8900A en mode Boundary Scan Test.

Pour une utilisation normale, cette pin est laissée HAUT.

RES : Cette entrée est reliée àune résistance de 4.99K +/- 1% pour

DVDD1…DVDD4 : Fournit le 5V +/- 5% aux circuits numériques du CS8900A.

DVSS1…DVSS4 : Masses numériques.

AVDD1…AVDD3 : Fournit le 5V +/- 5% aux circuits analogiques du CS8900A.

AVSS0…AVSS4 : Masses analogiques.

Annexe 4: le PCB de la carte mère

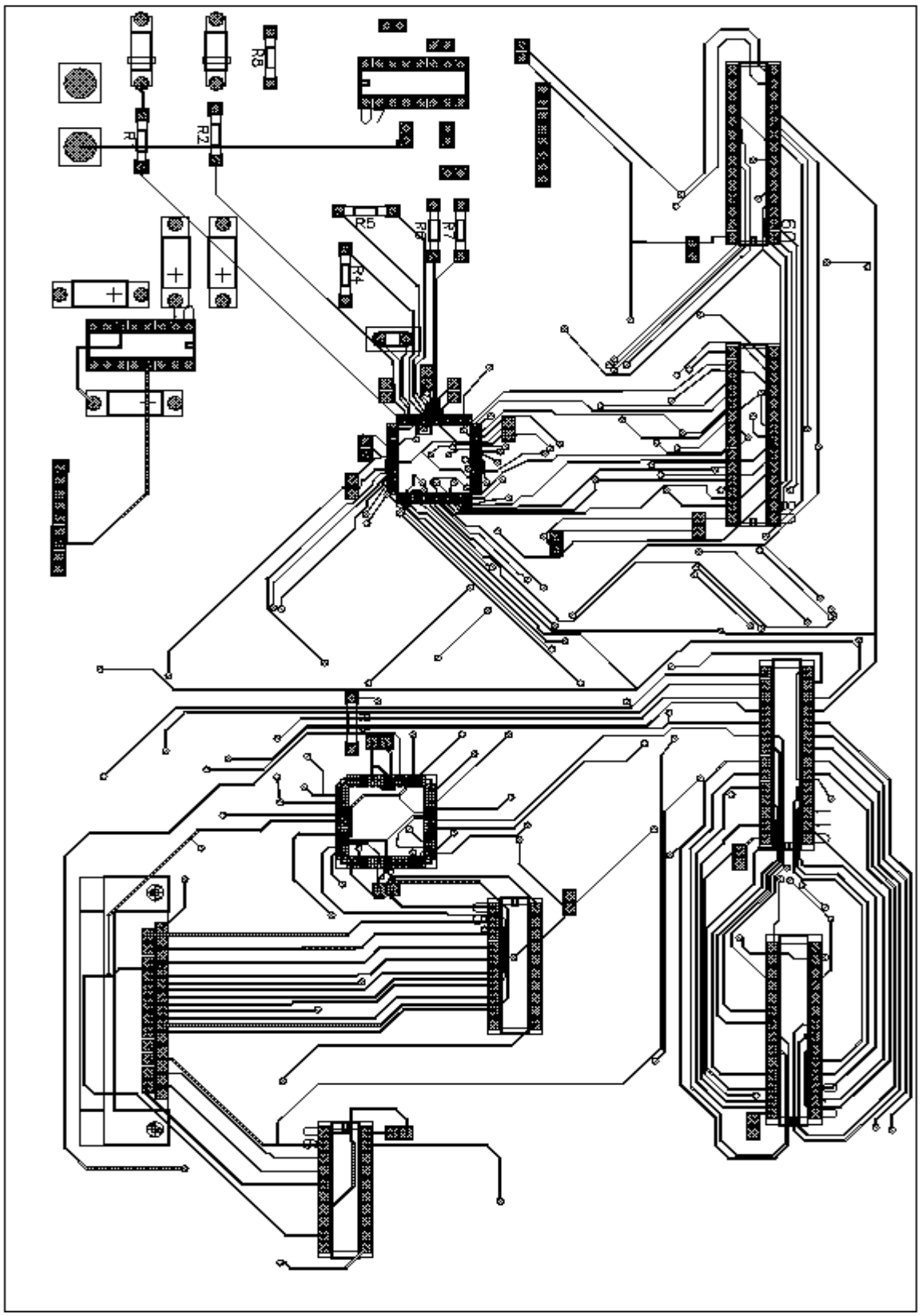


Figure 20: Artwork1 coté composant

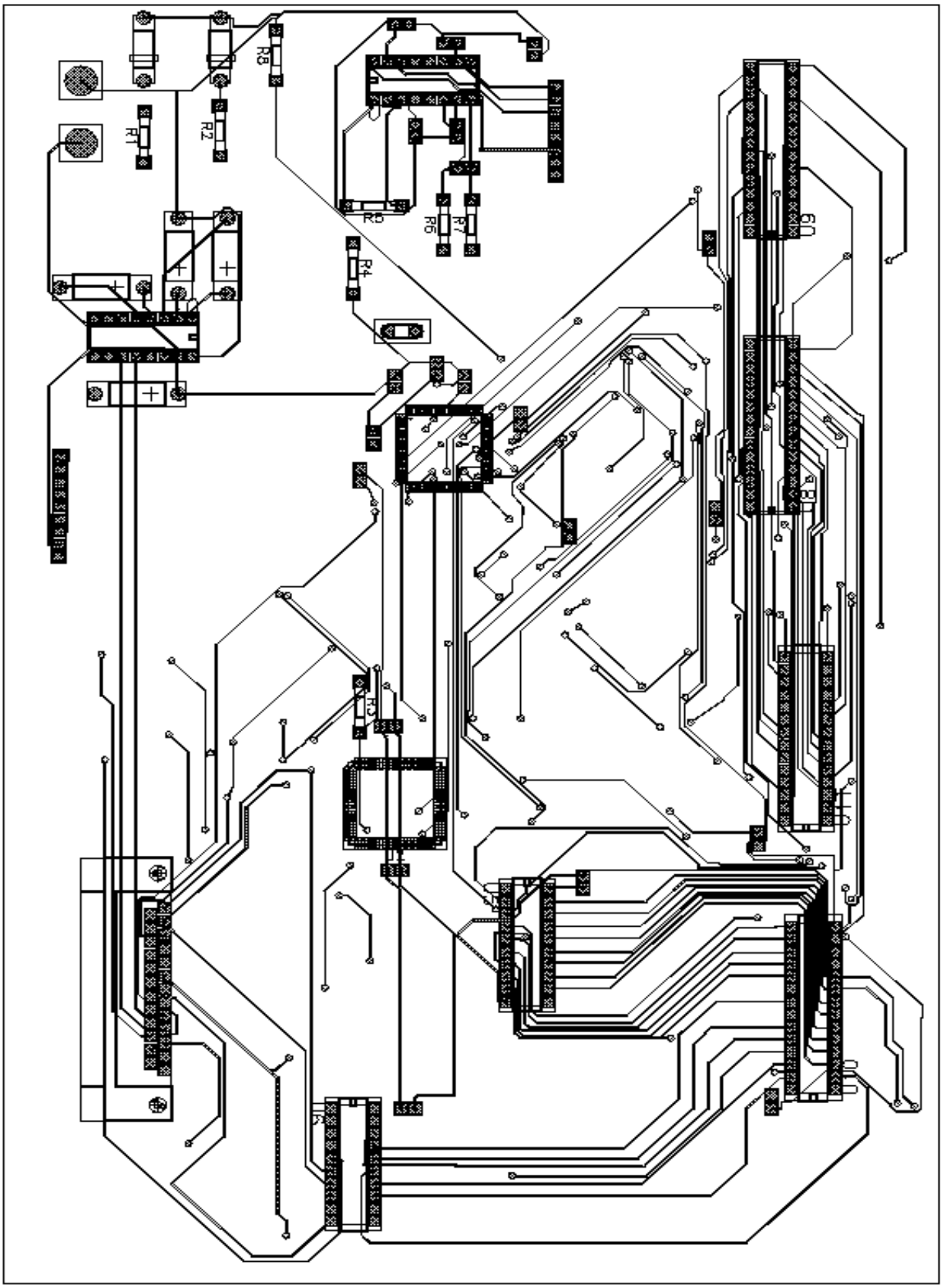


Figure 21:Artwork2 coté cuivre

9. INDEX DES FIGURES