

Réalisation d'un système de traitement de données radio-astronomiques pour la recherche de nouveaux pulsars rapides.

Thierry Fayard

CNES
18 av BELIN 31401 TOULOUSE cédex

thierry.fayard@cnes.fr

Résumé - Le CNES a eu pour mission de réaliser un système capable d'enregistrer numériquement et de traiter un signal reçu en station de radio astronomie dans le cadre d'un programme de recherche de nouveaux pulsars rapides du Bureau International des Poids et Mesures et de l'Observatoire de Paris Meudon. Pour optimiser le rapport signal à bruit, très faible, nous traitons le signal brut issu du récepteur, par transformée de Fourier, dédispersion et repliement. Le volume de donnée élevé a nécessité la réalisation d'un ordinateur parallèle à base de DSP SHARC d'Analog Device. L'usage de tel système devrait se répandre étant donné les applications possibles en particulier dans la lutte contre les parasites en radioastronomie.

Abstract - Involved in a pulsar search program, CNES had to provide Bureau International des Poids et Mesures and the Observatoire de Paris Meudon with a digital recorder and a high performance data processor. To optimise the weak signal to noise ratio we process the data directly issued from the receivers, by Fourier transform, dedispersion and folding algorithms. The high data flow required the set up of a parallel computer based on the SHARC DSP from Analog Device. Such system should be in use in a near future to handle parasite problems in radio astronomy.

1. Introduction

Un programme de recherche de nouveaux pulsars rapides a été initié sous la direction scientifique de Gérard Petit du Bureau International des Poids et Mesures. En 1998 près de 400 observations ont été faites à l'Observatoire de radio astronomie de Nançay. Chaque observation correspond à 4 minutes x 4 bandes de 6.4 Mhz à 16 Méga échantillons/s x 1 bit, soit au total 2 Giga octets de données par observation. Celles-ci devront être analysées pour tenter de découvrir un pulsar. Un système complet a été conçu à cette fin. Les deux problèmes techniques que nous avons du résoudre ont été d'une part la grandeur des débits utiles à l'acquisition des données sur support informatique, et d'autre part la puissance de calcul nécessaire au traitement. L'enregistrement des signaux en station se fait par un système dédié à la radio astronomie.

2. Description des différents éléments du système

2.1 L'interface d'entrée de l'enregistreur

Développé en interne (6 mois + 50 KF), Il s'agit de 4 convertisseurs 2 bits travaillant à 16 Méga échantillons/s. Le codage d'un échantillon sur 2 bits est choisi de façon à simplifier les traitements logiciels ultérieurs. L'interface est connectée d'une part à la chaîne de réception de la station radio qui lui délivre 4 signaux analogiques de 6.4 Mhz de largeur de bande. D'autre part elle est connectée à l'enregistreur par une liaison ECL.

2.2 L'enregistreur numérique

Acheté au Canada il est dédié à la radio astronomie et est le moins cher à ce niveau de performance: 128 Mbits/s pendant 4 Heures soit 2 Téra bits. Le coût de ce système est de 350 KF.

2.3 L'interface de sortie vers un PC

Développée en interne en 1998 (8 mois + 90 KF), c'est un des points durs du système. Les données doivent être stockées sur un disque dur de PC avant les traitements qui seront fait en temps différé. Le taux de transfert doit être en permanence supérieur à 4 Mcoctets/s (2 Mcoctets/s sur 1 bit) car l'enregistreur numérique ne peut pas restituer les données à un débit inférieur. L'interface est connectée à l'enregistreur par une liaison ECL et au PC par une connexion SCSI. Afin d'éviter d'avoir à gérer le bus PCI nous avons utilisé une carte SCSI du commerce dont le processeur a été reprogrammé pour faire l'acquisition brute des données sans utiliser le protocole SCSI. Sous Windows95 il a été impératif d'écrire un driver pour gérer la mémoire tampon, celle-ci doit être attribuée physiquement à l'application. La gestion des accès disques par Windows95 provoque des blocages atteignant plusieurs centaines de millisecondes, voire plus dans certains mode d'écriture disque. Avec 4 Mcoctets de mémoire tampon l'interface n'a jamais perdu de données.

2.4 Le calculateur parallèle

Développé en interne en 1997(3 mois + 40KF pour 5 cartes), c'est la pièce maîtresse du système. Sa puissance et son architecture doivent permettre de traiter les données en un temps raisonnable et pour un coût raisonnable également. L'architecture générale choisie est de type mémoire distribuée avec connexion en point par point. Chaque processeur a sa mémoire propre et est relié à quelques voisins (jusqu'à 6) par une liaison dédiée. Ce choix est justifié par la puissance de calcul des processeurs disponibles et la puissance totale nécessaire qui implique un nombre de processeurs supérieur à 20 et pouvant atteindre quelques centaines. Les systèmes à mémoire partagée ou à bus partagés subissent des contraintes lorsque le nombre de processeurs atteint ces valeurs. Lors de la conception nous avons souhaité simplifier au maximum la réalisation matérielle tout en

ayant un niveau de performance élevé. L'architecture retenue évite les complications liées à un partage de ressources matérielles, mémoire ou bus principalement. Ainsi, la conception du système se restreint à la conception d'une carte dite nœud de calcul qui comporte un DSP (digital signal processor), de la mémoire et des liens de communication. Les nœuds de calculs sont reliés entre eux par un simple câble pouvant atteindre 1.5 mètre de longueur.

Le choix du processeur DSP s'est fait avec les mêmes soucis de simplicité matérielle. Nous avons donc opté pour le SHARC 2106x qui associe excellentes performances sur nombre flottant 40 bits et 6 liens de communication à 20 Mcoctets/s soutenu et vérifié.

Nous avons adopté de la mémoire statique pour la simplicité de mise en œuvre et les performances également. Evidemment ce choix limite la capacité mémoire disponible, nos nœuds de calcul en comportent 1.25 Mcoctets en externe auquel se rajoute la mémoire interne pouvant atteindre 500 Kcoctets suivant les modèles. Tant la mémoire externe que le cache interne sont accessibles en un cycle d'horloge. Aucun contrôleur d'adresse n'a été nécessaire. Une mémoire morte est également présente pour le programme de lancement décrit ci dessous. Un petit EPLD (Altera 7032) gère des entrées sorties comme par exemples 5 leds présentes en façade. Ces leds sont un auxiliaire très utiles pour rechercher des erreurs logicielles, pourtant on n'en trouve pas sur les systèmes du commerce.

La carte d'essai a fonctionné du premier coup. Le seul problème rencontré par le sous-traitant a été le soudage du SHARC sur le circuit. Une carte a du être refaite à cause de faux contacts et sur une autre carte, un des liens de communication ne fonctionne pas.

2.5 L'interface entre le PC et les SHARC

Développé en interne en 1997 (2 mois), l'interface entre le PC et le calculateur parallèle relie directement un lien SHARC au bus ISA du PC hôte. Ses faibles performances en terme de débit sont celle que l'on peut obtenir avec le bus ISA étendu, c'est à dire à peu près 800 Kcoctets/s. L'adaptation se fait avec un EPLD (Altera 7128). Le choix du bus ISA a été fait pour des raisons de simplicité. L'application ne nécessitait pas de débit élevé avant que l'on ne modifie l'algorithme.

2.6 Le logiciel de lancement automatique

Développé en interne (5 mois), c'est l'élément du calculateur qui a été le plus complexe à mettre au point. Présent dans une mémoire morte dans chaque nœud de calcul il commence par se charger en mémoire vive. Puis il va scruter les différents liens de communication et, de proche en proche, il construit la base de données qui renseigne la géométrie du calculateur. Cette base de données est transmise à l'ordinateur hôte ce qui permet ensuite le chargement du programme dans tout le calculateur. Il s'auto détruit à la fin du chargement du programme principal.

L'écriture d'un programme pour le calculateur se fait en écrivant le processus qui tourne sur chaque processeur. Chaque processus est compilé séparément avec un nom de fichier qui lui appartient. Le programme de l'ordinateur hôte appelle une fonction qui

initialise le calculateur parallèle, reçoit la base de données sur sa géométrie et envoie dans l'ordre qui convient les processus vers les SHARC. Ainsi l'utilisateur n'a pas à se soucier du chargement, il doit simplement indiquer le numéro du processeur à affecter à chaque processus.

2.7 Les bibliothèques de communications

Développées en interne (5 mois). C'est un ensemble de fonctions de communication entre les processeurs SHARC ou, entre un processeur et l'ordinateur PC hôte, qui permettent d'utiliser très simplement le calculateur. Ces fonctions écrites en assembleur permettent d'envoyer simplement (ou recevoir) un entier, un flottant ou un tableau sur un lien de communication du SHARC. A charge du programmeur de vérifier si à l'envoi d'une donnée correspond une réception en face. Ces bibliothèques existent également pour l'ordinateur hôte.

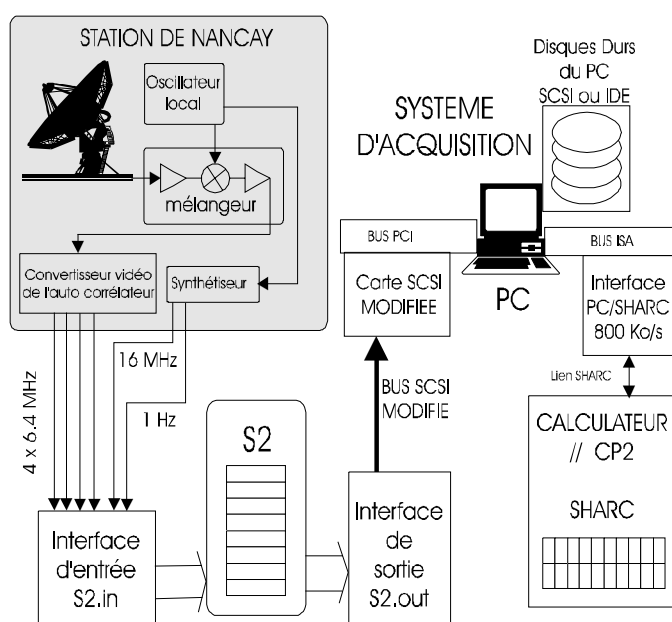


Fig.1 Le système dans l'état actuel en juin 1999. La zone grisée représente la station de radio astronomie de Nançay. Les données sont enregistrées sur des cassettes de 3 heures avec l'enregistreur S2. Puis les observations sont restituées du S2 vers le disque dur du PC. Enfin les données sont traitées par le calculateur parallèle, 5 SHARC actuellement.

3. Le logiciel de traitement

3.1 Le signal à traiter

Le signal à traiter est le tableau de valeurs numérisées sur 1 ou 2 bits représentant la tension reçue au récepteur. Un signal émis par un pulsar est caractérisé par un bruit gaussien multiplié par une fonction périodique créneau. Lorsque la terre se trouve dans le pinceau émissif du pulsar il reçoit un pulse. Celui-ci est d'une largeur moyenne constante de typiquement quelques dizaines de micro secondes. La période est elle aussi constante sur l'intervalle de temps considéré mais, dans le cas des pulsars binaires (un

pulsar tournant autour d'un autre corps céleste), un effet Doppler fait apparaître une variation. Les périodes des pulsars connus varient entre 1.5 et plusieurs centaines millisecondes. Le pulse, en voyageant à travers le milieu interstellaire, est dispersé au sens optique, les fréquences élevées voyageant plus vite arrivent en premier. Cet effet devra être corrigé pour permettre la détection. A ce signal va se rajouter le bruit gaussien lui aussi et des dizaines de milliers de fois plus fort que le signal à détecter. Ainsi il faut accumuler, en repliant le signal sur lui-même à la période du pulsar, des millions d'échantillons, pour permettre la détection. On montre que n repliements augmente le rapport signal sur bruit de racine de n . Cette détection se caractérise par la présence d'un pic dans le signal comme on le voit en figure 2.

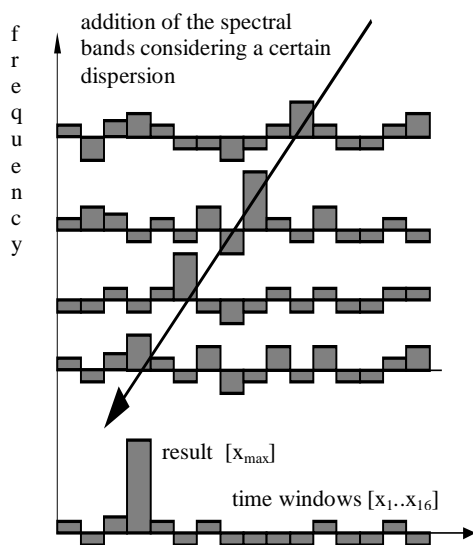


Fig.2 L'addition des bandes spectrales à la bonne valeur de dispersion fait ressortir le pulse.

3.2 L'algorithme de traitement

Quand un pulsar est connu, sa période l'est également. Mais, comme nous cherchons à découvrir un pulsar dans l'observation d'une portion de ciel, la méthode que nous utilisons consiste à tester toutes les périodes possibles comprises entre deux valeurs extrêmes. Celles que nous avons retenues pour notre application sont 0.6 et 3 millisecondes. La dispersion n'est pas connue également, elle est liée principalement à la distance du pulsar à la terre et au milieu traversé. Donc, toutes les dispersions comprises entre deux valeurs réalistes sont testées. Si pour une période et une dispersion particulière on observe un pulse dans le signal (figure 2) on retient l'échantillon pour une analyse plus fine.

Pré traitement : Il consiste à calculer des spectres sur 32 bandes de fréquence puis de les accumuler sur une durée choisie. On obtient ainsi un spectre dynamique dont la résolution temporelle est paramétrable. Si la résolution est trop faible le volume de données est très élevé. Si elle dépasse trop la largeur des pulses ceux-ci seront noyés dans le bruit et la détection sera impossible. Le temps de calcul lié à cette étape est faible comparativement à l'étape suivante.

Repliement : A ce stade, le spectre dynamique est replié (accumulé) avec une période variant entre deux valeurs extrêmes. Il est également dédispersé avec une valeur variant également.

C'est le très grand nombre de valeur de période à essayer, qui accroît considérablement le temps de calcul. Ce nombre est grand parce que la précision sur la valeur de la période doit être tel que le pulse se trouve toujours dans la même fenêtre temporelle après les repliements. Cette précision est proportionnelle à la durée totale du signal à traiter et à l'inverse du carré de la période. Ainsi, pour une observation de 4 minutes plus de 2 500 000 valeurs de période doivent être testées.

Détection automatique : pour chaque couple de valeur de période et dispersion on obtient une figure caractéristique. Les propriétés de cette figure peuvent indiquer la présence d'un pulsar (figure 3) ou d'un parasite (figure 4). Une discrimination simple se fait en comparant le rapport signal sur bruit local et global. Dans le cas d'un parasite le signal sur bruit local est mauvais.

3.3 Optimisation de l'algorithme de repliement.

Typiquement le nombre d'addition à effectuer pour le repliement à une période déterminée est $240 \cdot 10^6$ qui multiplié par le nombre de périodes à tester fait environ 10^{15} opérations ! C'est trop élevé et l'algorithme de traitement a été amélioré. Au lieu de replier les spectres sur la totalité de l'observation on replie sur un intervalle temporel réduit appelé bloc, le nombre de période à tester est réduit d'autant. Puis on additionne des spectres repliés à différentes valeurs de période provenant de différents blocs, le traitement de l'ensemble des combinaisons possible est presque équivalent au traitement précédent. Le nombre d'opération est réduit d'un facteur supérieur à 100 ce qui correspond à la réduction de l'intervalle à replier. En répétant ce procédé une seconde fois on gagne encore un facteur 10. La perte évaluée de rapport signal sur bruit est inférieure à 5% dans les mauvais cas. Cela représente encore quelques 10^{12} opérations à effectuer. En outre, le gain est notablement réduit par les opérations de stockage intermédiaire comme nous allons le voir dans la mise en œuvre. Cet algorithme assez complexe, par son efficacité, rend possible le traitement.

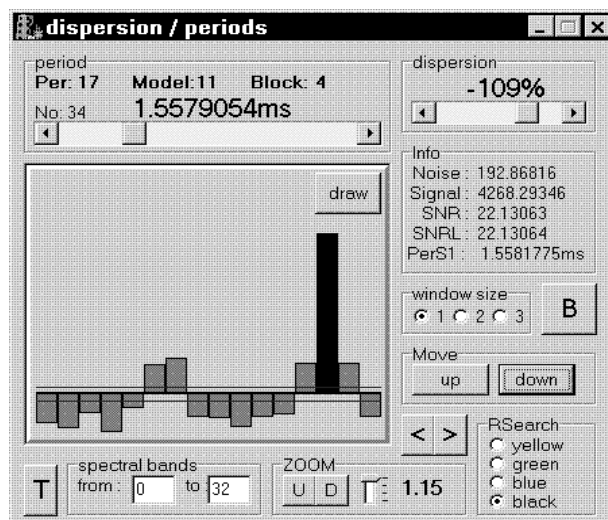


Fig.3 La détection du pulsar 1937+214

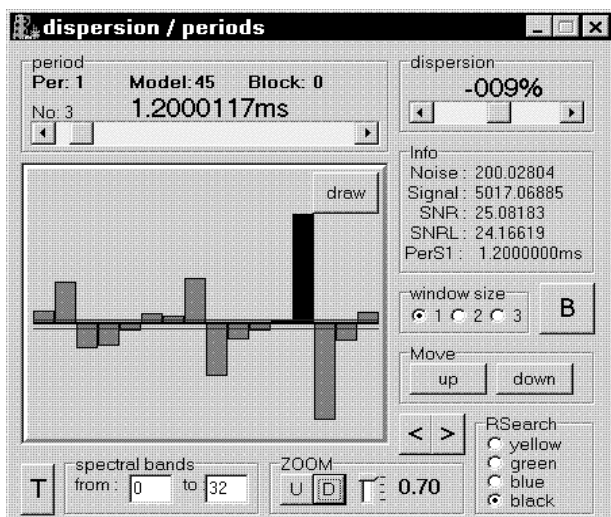


Fig.4 La détection d'un parasite

4. La mise en oeuvre

4.1 Les premières détections

Une fois compilé en C et optimisé en assembleur (outils Analog Device), les logiciels ont été pleinement validés par la détection de pulsars connus. Des comparaisons avec un système basé sur un réseau de PC ont montré une excellente qualité des résultats de détection qui valide les logiciels et les algorithmes utilisés (figure 3). Une augmentation du rapport signal sur bruit a été observée, elle s'explique sans doute par l'utilisation de flottant dans les calculs au lieu d'entier 16 bits.

4.2 facteurs limitant le système actuel

Actuellement le système de test possède 5 cartes SHARC et le PC hôte a un disque dur de 6 Goctets qui contient également les données à traiter. Il n'est donc pas encore possible de traiter une observation complète en un seul passage ce qui nous oblige à estimer les temps de calculs finaux. Cependant nous constatons déjà que le point faible du système est, avec l'algorithme optimisé, le passage des données entre le calculateur parallèle et le PC hôte. Le débit entre SHARC ne pose aucun problème car il tient bien les 20 Mcoctets/s. Techniquement le problème posé par l'algorithme optimisé est le stockage temporaire sur disque dur des données avant leur combinaison. Ce volume atteint 8 Goctets pour une observation typique de 240 secondes à lire et écrire deux fois (une par étape supplémentaire). Cette liaison se fait par l'intermédiaire du bus ISA du PC à un débit maximum de 800 Koctets par secondes. Donc les transferts nécessitent 11 heures. Or un SHARC est capable d'effectuer une addition flottante et de ranger le résultat en un cycle de 25 nano secondes. Cela représente donc en théorie 2 heures pour le traitement par les 5 SHARC d'une observation. Même si en pratique le temps de calcul est beaucoup plus long, près de 10 heures, on voit que le facteur limitant sera les transferts de données. Ce surcroît de temps de calcul n'est pas encore totalement expliqué, il provient surtout de la limitation en taille du disque dur qui implique des manipulations de données supplémentaires.

4.3 Optimisation de la version finale du système

Des disques durs Ultrawide SCSI de 18 Goctets de capacité sont en cours d'acquisition. Avec une interface entre le calculateur SHARC et le PC hôte sur bus PCI, le taux de transfert sur ces disques doit atteindre les 6 ou 7 Mcoctets/s, ce qui ferait tomber le temps de transfert à 1 heure et 20 minutes. La limitation de taille du disque disparaît également ce qui permettra de se rapprocher des temps de calculs théoriques. Si on souhaite ramener le temps de calcul à une valeur voisine du temps de transfert on voit qu'il faudrait 20 à 30 SHARC pour le système final, ceux-ci sont en cours d'acquisition.

4.4 Considérations économiques

Le coût d'un nœud, boîtier et alimentation compris, est de l'ordre de 4 KF. Le coût total du calculateur version 25 nœuds est de l'ordre de 120 KF avec un an de développement, essentiellement pour les logiciels. Un système équivalent "sur étagère" aurait coûté au moins 600 KF en 1997 avec les logiciels. Mais la mise en œuvre n'est pas toujours une chose. On peut douter que le simple portage d'un source C sur un système prêt à l'emploi donne de bons résultats. Une très bonne connaissance du processeur est de toute façon utile pour l'optimisation assembleur des boucles internes, nous avons noté des facteurs de 10 par rapport au C. Cependant l'évolution rapide des processeurs limite un peu l'intérêt de cette connaissance. Faut-il continuer à développer ? J'aurai tendance à répondre non pour des systèmes inférieurs à 10 processeurs, et oui pour plus de 50. Le débat reste ouvert.

5. Conclusion

L'objectif qui est de traiter une observation complète, soit 4 observations élémentaires de 4 minutes, par jour est tout à fait réalisable avec le système après modifications. Ce qui est remarquable dans cette application c'est que l'optimisation de l'algorithme implique une modification importante du système pour accélérer les accès aux disques durs. Plutôt que de porter l'interface entre le calculateur SHARC et le PC sur le bus PCI, il aurait peut être été plus élégant d'utiliser une carte SHARC SCSI pour accéder aux disques durs. Mais cette solution aurait induit des modifications logicielles importantes.

Une autre application envisageable, et très prometteuse avec un tel système serait la détection de parasites. Un simple spectre à haute résolution permet de détecter des parasites dans un signal radio-astronomique. Il est alors possible de nettoyer les spectres avant intégration, même en temps réel avec 30 SHARC.

Références

Rapport de stage de M. Olivier Danet, CNES juin 1997
Rapport de stage de M. Sebastian Karl, CNES mars 1999

Remerciement

Tous mes remerciements à M. Gérard Petit, responsable scientifique de cette expérience. Mes remerciements les plus vifs également à M. O. Danet pour sa contribution exceptionnelle à la réalisation matérielle et logicielle du calculateur, et à M. S. Karl pour sa mise en œuvre avec les algorithmes décrits précédemment. Merci à M. R. Reich pour me relire. Les figures 2,3 et 4 sont issues du rapport de M. S. Karl.