

SEPTIEME COLLOQUE SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

NICE du 28 MAI au 2 JUIN 1979

SIMULATEUR DE RESEAUX DE PETRI

M. CHAMBON Philippe

Société SINTRA - 26, Rue Malakoff 92600 ASNIERES

RESUME

Le simulateur de Réseau de Pétri est une C.A.O. d'aide à la description, l'analyse, la validation et l'évaluation de performances de processus complexes en satisfaisant les spécifications fonctionnelles et les contraintes coût-performance-fiabilité-sécurité du système.

Ce simulateur basé sur la description des processus par réseaux de Pétri a été réalisé dans le cadre d'un contrat DRET, sur système Time Sharing FORTRAN CDC 7600 ; il utilise un langage graphique et alpha-numérique évolué de dialogue interactif.

L'intérêt du simulateur de réseaux de Pétri réside dans son aptitude à représenter, visualiser et simuler l'évolution dynamique dans leur contexte externe de processus complexes en traitement du signal tels que :

- processus temps réel,
- systèmes logiques séquentiels et combinatoires,
- systèmes micronumériques à hautes performances,
- processus asynchrones parallèles, concurrentiels et interactifs (optimisation d'algorithmes de traitement du signal en fonction de l'implémentation hardware),
- moniteur d'enchaînement et de coordination des tâches (moniteur câblé).

Il permet également de traiter des domaines plus divers tels que :

- recherche opérationnelle,
- traitement de base de donnée,
- système de gestion,
- automatisme industriel,
- analyse de fiabilité, disponibilité, locavar, dans les systèmes macromodulaires.

Dans ce contexte, le simulateur permet :

- l'optimisation des algorithmes de traitement du signal en fonction de leur implémentation hardware,
- la détection des conflits de processus,
- la détection des blocages partiels ou généralisés des processus,
- la détection des fonctionnements indéterminés,
- l'optimisation de l'enchaînement des tâches (parallélisme, pipe-ligne, ...),
- l'évaluation des synchronisations inter-processus,
- l'évaluation des goulots d'étranglement,
- l'évaluation des priorités,
- l'évaluation et l'optimisation des performances du processus dynamiquement et en fonction de son contexte externe.

SUMMARY

The Petri nets simulator allows modelisation, analysis, validation and performances computation of complex process according to functional specifications and cost-performances-reliability-security of the system.

This simulator based on process description by mean of Petri nets was sponsored by DRET (Direction des Recherches et Etudes Techniques) and the development on FORTRAN CDC 7600 Time Sharing system was managed by SINTRA.

Interest of Petri nets simulator is his ability to design, analyse and simulate dynamic evolution of complex digital signal process in their external environment, like :

- real time process,
- combinational and sequential logic systems
- high performances digital micronumerical systems,
- parallel, asynchronous, competitive and interactive process (digital signal processing optimisation according to hardware implementation),
- hardware monitor.

Likewise, it allows to analyse others problems such as :

- operational research,
- data base processing,
- management systems,
- reliability and locavar analysis,
- industrial automation.

For all this process, the simulator allows us :

- algorithms optimisation in digital signal processing according to hardware implementation,
- process conflicts detection,
- global and partial deadlock detection and localisation,
- undetermined process detection,
- scheduling optimisation,
- crossprocess synchronisation determination,
- priority computation,
- performances computation and optimisation according to dynamical process and external environment.



1 - DESCRIPTION DU SIMULATEUR (SIREP)

Le simulateur de réseaux de Pétri est entièrement écrit en FORTRAN 4 sur système Time Sharing CDC 7600. L'interface entre l'opérateur et le système est réalisé par un terminal graphique TEKTRONIX 4015 (options 40 et 41) et un télétype Anderson Jacobson 832.

Ce simulateur utilise en plus du langage interactif alphanumérique, un langage graphique interactif évolué à deux modes (rafraîchi et permanent) permettant son utilisation simple et efficace par l'opérateur.

1.1 - CARACTERISTIQUES

Ce simulateur "SIREP" possède les caractéristiques suivantes :

- Modularité

- . Organisation structurée modulaire (bibliothèques de 150 modules).
- . Interconnexion des modules composant un système au moyen de variables d'interface.

- Conception ouverte

- . Conception intégrant actuellement toute la classe des réseaux de Pétri généralisés et permettant une extension aux réseaux colorés, aux réseaux auto-modifiables aux places de type "sources et puits".

- Séparation partie contrôle-partie opérative

- . Chaque module du système est un réseau de Pétri interprété temporellement.
- . La partie contrôle est décrite par le réseau de Pétri et la partie opérative par un ensemble de procédures, chacune représentant le fonctionnement d'un opérateur ou d'une tâche.

- Représentation des synchronisations et du parallélisme

- . Utilisation d'arcs inhibiteurs, arcs valués, prédicats complexes pour la synchronisation des modules au niveau de la partie contrôle.
- . Souplesse d'utilisation de la partie contrôle (évaluation paramétrée des arcs, suppressions-ajouts de marques, arcs, places, ...).

- Analyse du comportement du système dans son contexte

- . Détection des conflits, blocages, verrous,
- . Sensibilité des performances aux variations de différents facteurs.
- . Prise en compte des événements et du contexte externe au système simulé.

- Simulation temps réel

- . Intervention interactive de l'utilisateur lors de la modélisation, analyse, validation de son système pour corriger et influencer sur la simulation.

1.2 - ORGANISATION DU PROGRAMME SIREP

Ce programme est constitué de trois modules principaux :

- module de visualisation,
- module de dialogue et de gestion du simulateur,
- module de simulation.

Leurs fonctions principales sont les suivantes :

- Module de visualisation

Ce module permet :

- . constitution graphique du réseau de Pétri décrivant le système à simuler,

- . constitution des fichiers de description et de chaînage du réseau de Pétri,
- . modifications du réseau et des paramètres liés au réseau,
- . visualisation d'informations relatives au réseau sur commande du module de dialogue,
- . visualisation des marquages du réseau représentant son évolution dynamique,
- . visualisation de tous états remarquables ou critiques et des paramètres associés.

- Module de dialogue et de gestion du simulateur

Ce module permet :

- . le pilotage du simulateur par l'intermédiaire d'un jeu de 25 commandes,
- . commande d'affichage d'un certain nombre d'informations relatives au réseau à simuler,
- . modification de l'état du réseau en cours de simulation (marquage, prédicats, variables générales ou locales, ...),
- . prise en compte d'assertions sur l'évolution du réseau.

- Module de simulation

Ce module permet :

- . le calcul de l'évolution des marquages,
- . la détection des états remarquables,
- . le calcul des évaluations des performances du système décrit,
- . l'historique de l'évolution dynamique du processus décrit,
- . etc.....

2 - EXEMPLE D'APPLICATION DU SIMULATEUR (SIREP)

L'exemple proposé décrit l'utilisation du simulateur (SIREP) pour l'analyse complète qui a été réalisé à la SINTRA d'un opérateur spécialisé de F.F.T. destiné à être couplé au processeur de traitement du signal PPS conçu pour réaliser dans les meilleurs rapports coût-efficacité tous les problèmes de traitement du signal temps réel à fort débit.

L'exemple proposé décrit l'utilisation du simulateur de réseaux de Pétri (SIREP) pour l'analyse complète d'un opérateur spécialisé de FFT. Cet opérateur est une des versions d'opérateurs spécialisés destinés à être couplés au processeur de traitement du signal(2) conçu pour réaliser dans les meilleurs rapports coût-efficacité tous les problèmes de traitement du signal temps réel à fort débit.

Etant donné la complexité du problème traité nous ne présenterons ici que les grandes lignes et résultats de l'analyse à l'aide du simulateur de réseaux de Pétri.

3 - APPLICATION A LA REALISATION D'UN PROCESSEUR SPECIALISE DE TRAITEMENT DE SIGNAL

Cet exemple, volontairement limité dans sa complexité, illustrera l'utilisation du simulateur de réseaux de Pétri à la description et l'analyse d'un processeur de FFT en pipe-line.

Le simulateur permet une modélisation et une analyse structurée du haut vers le bas. La première étape consiste à modéliser le processus dans sa généralité et d'en tirer les spécifications globales du système. La deuxième étape analyse et valide les processus de synchronisation interprocesseurs au niveau macroscopique

Enfin, à l'intérieur d'un processeur élémentaire, à un niveau encore plus fin, nous analyserons le processus au niveau des différents éléments constitutifs du processeur (adressage, traitement des données).

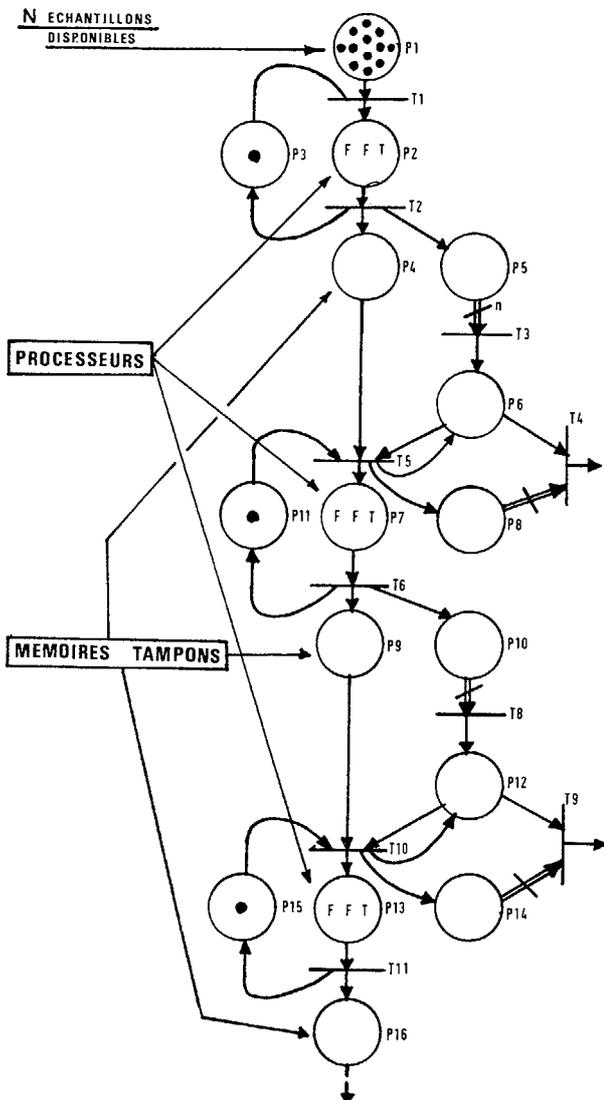
On pourra, donc, après validation par le simulateur, implémenter, par la CAO IMPLA, directement le système de commande sur PLA par exemple.

3.1 - ETUDE ET MODELISATION DU PROCESSUS PIPE-LINE DE FFT SUR K PROCESSEURS

Cette première analyse au niveau le plus macroscopique sert à déterminer et valider les contraintes globales, ainsi qu'à optimiser un certain nombre de variables caractéristiques du processus. Ici l'un des éléments essentiels est le dimensionnement des mémoires tampons interprocesseurs.

Il s'agit non seulement de minimiser le volume mémoire nécessaire au processus pipe-line, mais encore de vérifier que le processus se synchronise bien sur les données d'entrées du système et fonctionne sans blocage et sans perte de temps.

Le réseau de Pétri généralisé suivant décrit le processus pipe-line au sens de flots de données et gestion des ressources disponibles plus qu'au sens de processus de commande.



3.1.1 - Description du réseau

- Initialisation

La place P1 représente les ressources à l'entrée du système à modéliser, soit dans notre cas n échantillons complexes disponibles (n jetons de la place P1).

La place P3 représente par la présence ou non d'un jeton la disponibilité ou non du processeur de données typique représenté par la place P2.

La place P4 représente la mémoire tampon entre deux processeurs typiques.

L'ensemble des places P5, P6, P8, représente l'évolution d'un compteur destiné à gérer l'ensemble des ressources entre processeurs.

La contrainte essentielle fixée par le cahier des charges est qu'un processeur typique (réalisant un étage de FFT) ne peut être déclenché que par la présence de n données disponibles en entrée.

3.1.2 - Analyse et résultats de simulation

Partant de l'initialisation décrite précédemment (n données disponibles à l'entrée du système), le simulateur a permis :

- le calcul du dimensionnement des mémoires tampons par l'analyse du nombre maximum de jetons présents dans les places P4, P9, ..., au cours de l'évolution du processus,
- de vérifier par les analyses statistiques sur l'ensemble des transitions du réseau que le processus pipe-line est optimum et ne présente aucune perte de temps,
- d'optimiser le volume des mémoires tampons en fonction de plusieurs hypothèses différentes sur les processeurs de données (notamment sur les processus élémentaires d'adressage),
- de déterminer en fonction de ces hypothèses les contraintes technologiques sur les mémoires (temps d'accès notamment) pour optimiser le processus,
- de vérifier l'absence de blocages, de conflits et permettre ainsi une optimisation de la gestion des ressources et donc une optimisation du matériel,
- d'optimiser les algorithmes de calcul d'adresse et de traitement des données (papillon) en fonction des différentes implémentations possibles.

Cette analyse et ces résultats nous ont donc permis de déterminer et d'optimiser les principales spécifications du système à un niveau macroscopique.

Bien évidemment en fonction des résultats obtenus par la suite à des niveaux plus fins, nous pourrions toujours, à l'aide de l'effet zoom du simulateur, introduire toutes modifications sur le système et vérifier en temps réel leurs impacts sur le comportement global du système.

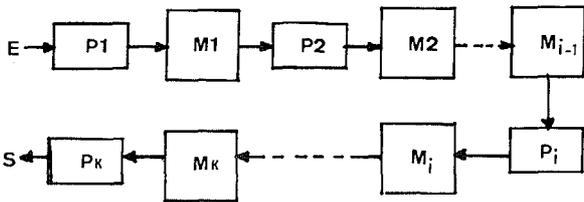
Cette première étape permettant d'établir les contraintes générales qui seront reportées dans toute la suite de l'analyse jusqu'aux niveaux les plus microscopiques du système afin d'en déterminer non seulement les caractéristiques et contraintes, mais également les performances et l'implémentation directe de la partie commande (sur PLA par exemple).

3.2 - ETUDE DE LA SYNCHRONISATION DE K PROCESSEURS DE FFT EN PIPE-LINE

Le système global étant synchronisé sur les entrées nous pouvons schématiser le système par :

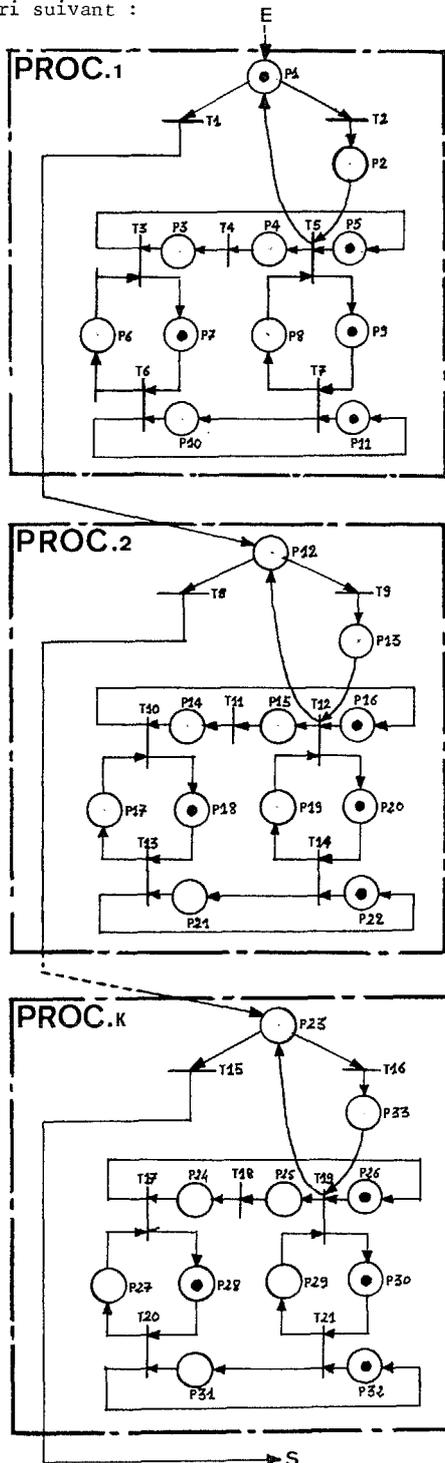


SIMULATEUR DE RESEAUX DE PETRI



P_i = processeur élémentaire i
 M_i = mémoire tampon i
 E = (N = 2K) échantillons disponibles
 S = sortie des échantillons du spectre.

La modélisation par réseau de Pétri est donnée par le réseau de Pétri suivant :



- Description des actions liées au réseau :

Soit ($V_i = 1$) la valeur du compteur i (initialisé à 1) est $\leq N/2$.

S ($V_i = 0$) la valeur du compteur i (initialisé à 1) est $> N/2$.

Les transitions du réseau traduisent les actions suivantes :

- t1, t8, t15 Franchissables si ($V_i = 0$)
- t2, t9, t16 Franchissables si ($V_i = 1$)
- t5, t12, t19 Lecture des données
- t4, t11, t18 Calcul du papillon FFT
- t3, t10, t17 Ecriture des résultats
- t6, t13, t20 Calcul d'adresse d'écriture
- t7, t14, t21 Calcul d'adresse de lecture

De plus, à chaque transition ou place du réseau est associée une temporisation permettant ainsi d'introduire les contraintes technologiques au niveau hardware, et donc de conditionner la simulation du réseau en fonction de ses implémentations possibles.

3.2.1 - Description du processus

Un processeur typique P_i effectuant un étage de FFT se compose de deux sous-processeurs se synchronisant entre eux :

- l'un (le processeur d'adresse) est chargé du calcul des adresses pour la lecture et/ou l'écriture (t5, t12, t19/t3, t10, t17),

- l'autre (processeur de données) :

- . lit les données (stockées en mémoire tampon M_{i-1} aux adresses calculées par le sous-processeur d'adresses,
- . range les données transformées en mémoire tampon M_i aux adresses calculées par le processeur d'adresses.

Ceci représente le cycle de base du fonctionnement du processeur P_i. Cycle de base que la simulation nous permettra d'optimiser par le simulateur afin d'obtenir les meilleurs ratios de performances.

- Contraintes sur les processus

On remarque sur le réseau de Pétri descriptif du processus que la lecture (t5, t12, t19) et l'écriture (t3, t10, t17) ne peuvent être faites qu'après les calculs d'adresses (t7, t14, t21) et (t6, t13, t20) respectivement.

De même, on ne recommence les calculs d'adresses (6, t12, t20) et (t7, t14, t21) qu'après que la lecture ou l'écriture soit adressée respectivement.

Pour N = 2K échantillons, chaque étape de FFT se fait en N/2 cycles de base.

Chaque processeur est muni d'un compteur modulo N/2 comptant le nombre de cycles effectués par P_i.

Une variable booléenne V_i est donc associée au compteur i (t1, t2, t8, t9, t15, t16) pour indiquer l'état du compteur.

V_i = 1 si la valeur du compteur est inférieure ou égale à N/2.

V_i = 0 dans le cas contraire.

Les transitions auxquelles est associée cette condition booléenne ne peuvent être mise à feu que si elles sont tirables et si la condition associée est satisfaite.

SIMULATEUR DE RESEAUX DE PETRI

Ainsi t_2, t_9, t_{16} sont franchies tant que $V_i = 1$ (le processeur P_i n'a pas terminé un étage de FFT).
 Quand $V_i = 0, t_1, t_8, t_{15}$ sont franchies le processeur P_i a terminé un étage de FFT et déclenche le processeur suivant P_{i+1} pour entamer un autre étage de FFT (processus pipe-line parallèle).

3.2.2 - Simulation et résultats

Etant donné la limitation de place, nous ne donnerons dans ce papier que les principaux résultats de simulation. Une démonstration complète du simulateur sur console TEKTRONIX 4015 est prévue lors du colloque.

La simulation du réseau précédent a permis :

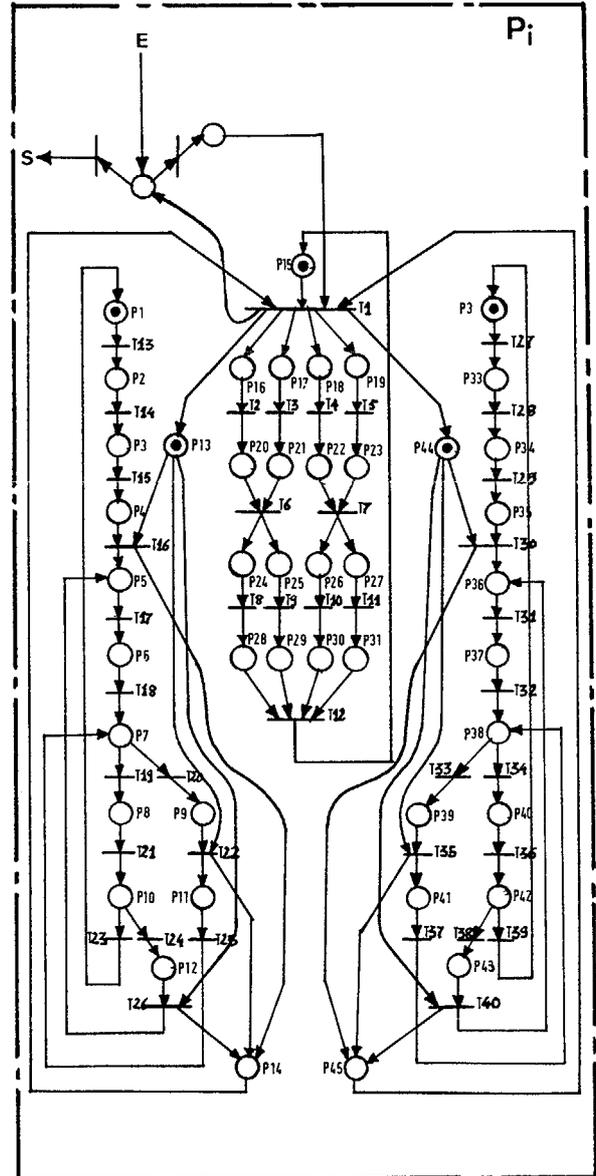
- la vérification de la cohérence des synchronisations interprocesseurs,
- l'optimisation du parallélisme de K processeurs fonctionnant en pipe-line,
- la détection de conflits et de goulots d'étranglements permettant les modifications du réseau résolvant les conflits de processus,
- l'évaluation des performances globales du système pour plusieurs implémentations possibles (temporisations diverses), et l'optimisation du rapport coût/performances,
- l'évaluation par l'analyse des statistiques liées aux places et transitions des vitesses de traitement et des débits maximum du processeur global,
- de déterminer les périodicité des différents processus et d'évaluer ainsi les charges des différents module constitutifs du système. Ce dernier point permettant une optimisation des ratios de performances des différents sous-systèmes.

3.3 - ETUDE D'UNE PROCESSEUR TYPIQUE P_i

Après avoir étudié et validé le processus d'enchaînement et de synchronisation interprocesseurs, nous pouvons descendre au niveau plus microscopique du processus interne d'un processeur typique P_i .

Ceci dans le but d'optimiser les rapports coût/performances et de rendre cohérent la puissance des processeurs élémentaires avec les contraintes technologiques du cahier des charges (débit maximum des bus, encombrement, coûts, type de mémoires, etc...).

Le fonctionnement et les synchronisations des processeurs de données et d'adresses constitutifs du processeur typique P_i sont décrits par le réseau de Pétri suivant :





SIMULATEUR DE RESEAUX DE PETRI

Ce réseau correspond à la version optimisée obtenue par le simulateur en fonction du cahier des charges.

Ce réseau de Pétri se décompose en trois parties représentant le fonctionnement en détail du processeur P_i :

- le calcul d'adresse est géré par deux sous-programmes analogues fournissant au processeur de données l'adresse des deux mots à lire pour alimenter le processeur de données,
- la partie centrale du réseau de Pétri représente le fonctionnement du processeur de données réalisant l'opération papillon,
- l'ensemble du réseau de Pétri permet de voir en détail la synchronisation des différents processeurs.

3.3.1 - Description du réseau de Pétri

Notons qu'ici encore, à chaque transition est affectée une temporisation représentant le temps d'exécution de l'action associée.

Les places P_1 et P_{32} (marquées à l'initialisation) indiquent la disponibilité des sous-processeurs de calcul d'adresse.

La place P_{15} représente la disponibilité du sous-processeur de données.

Les places P_{13} , P_{14} , P_{44} , P_{45} représentent les synchronisations nécessaires entre le sous-processeur de traitement de données (papillon) et les sous-processeurs d'adressage des données alimentant le sous-processeur de traitement.

A chaque transition sont associées les actions suivantes (pour $N = 2K$ et $i = n^\circ$ du processeur typique dans le pipe-line) :

t13 (t27) ; $K - i + 1 \rightarrow P(P')$
 t14 (t28) ; $i \rightarrow IT(IT')$
 t15 (t29) ; $2iT(iT') - 1 \rightarrow CT(CT')$
 t16 (t30) ; $2P(P') \rightarrow AD_1(AD_2)$
 t17 (t31) ; $AD_1(AD_2) \rightarrow T_1(T_2)$
 t18 (t32) ; $2P(P') \rightarrow X(X')$
 t19 (t34) ; Condition associée $X(X') = 0$
 t20 (t33) ; Condition associée $X(X') \neq 0$
 t21 (t36) ; $CT(CT') - 1 \rightarrow CT(CT')$
 t22 (t35) ; $AD_1(AD_2) + 2 \rightarrow AD_1(AD_2)$
 t23 (t39) ; Condition associée $CT(CT') = 0$
 t24 (t38) ; Condition associée $CT(CT') \neq 0$
 t25 (t37) ; $X(X') - 1 \rightarrow X(X')$
 t26 (t40) ; $T_1(T_2) + 2P(P') + 1 \rightarrow AD_1(AD_2)$

où : AD_1, AD_2 : variables représentant les adresses de lecture

X, X' : Compteurs
 CT, CT' : Compteurs

T_1, T_2 : variables d'adresse (tampon).

Pour la partie traitement de données nous avons :

t1 : lecture des données stockées aux adresses AD_1 et AD_2 , Q_{1r} , Q_{1i} , Q_{2r} , Q_{2i} , $\cos \theta$, $\sin \theta$,
 t2 : $Q_{2r} \times \cos \theta$; t3 : $Q_{2i} \times \sin \theta$; t4 : $Q_{2i} \times \cos \theta$
 t5 : $Q_{2r} \times \sin \theta$;
 t6 : $Q_{2r} \cos \theta + Q_{2i} \sin \theta$; t7 : $Q_{2i} \cos \theta - Q_{2r} \sin \theta$
 t8 : $Q_{1r} + (Q_{2r} \cos \theta + Q_{2i} \sin \theta)$;
 t9 : $Q_{1r} - (Q_{2r} \cos \theta + Q_{2i} \sin \theta)$;
 t10 : $Q_{1i} + (Q_{2i} \cos \theta - Q_{2r} \sin \theta)$;
 t11 : $Q_{1i} - (Q_{2i} \cos \theta - Q_{2r} \sin \theta)$;
 t12 : Ecriture en mémoire des résultats.

Nous voyons que par exemple dans le cas où $N = 23$ (et supposant que chaque échantillon est stocké en 2 mots (32 bits)), les adresses générées sont :

AD1	AD2
0 2 4 6	8 10 12 14
0 2 8 10	4 6 12 14
0 4 8 12	2 6 10 14

3.3.2 - Résultats de simulations

A ce niveau de l'étude, nous avons rapporté toutes les contraintes et résultats obtenus dans ce qui précède.

Ceci permet donc de valider complètement le processus et de tenir compte de toutes les composantes du système au niveau le plus microscopique.

La simulation nous a permis :

- de mettre au point et d'optimiser les différents sous-processeurs, à l'action élémentaire près,
- de vérifier le bon fonctionnement et les synchronisations entre sous-processeurs,
- de valider le processus de commande et de l'implémenter à l'aide de la CAO (IMPLA) directement sur PLA,
- de calculer et optimiser les performances globales du système,
- de spécifier jusqu'au niveau le plus fin toutes les spécifications, performances et contraintes liées au système global.

4 - CONCLUSION

Cet exemple, décrit très succinctement ici, nous a permis de montrer la puissance et l'utilité du simulateur de réseaux de Pétri pour l'analyse de processus complexes, notamment dans le domaine des micro-systèmes numériques à très hautes performances.

Par sa souplesse, sa puissance, ce simulateur permet donc de décrire, analyser et modéliser en temps réel des processus très complexes, et d'en assurer l'optimisation au niveau des performances, des coûts et donc de faire une sorte de preuve de l'implémentation optimale de processus et de bon fonctionnement.

Références

- Le Préprocesseur de signal PPS (GRETSI 79)
- Programmes de conception des micro-systèmes numériques à hautes performances (GRETSI 79)
- Ensemble d'outils pour la conception assistée par ordinateur de systèmes numériques à hautes performances (GRETSI 79).
- Travaux de MM. BERTHELOT - CAPLAIN - MOALLA - GHERBI PAPACOSTAS.