

A.- INTRODUCCION.

Los sistemas multiprocesadores constituyen una solución para la obtención de sistemas informáticos que permiten obtener un alto grado de paralelismo en la ejecución de los algoritmos.

INTERCONEXIÓN DE PROCESADORES Y MEMORIAS

EN SISTEMAS MULTIMICROPROCESADORES

por

DEPARTAMENT D'ARQUITECTURA DE COMPUTADORS
DE LA FACULTAT D'INFORMÀTICA DE BARCELONA

Nosotros nos centraremos en las redes de interconexión del tipo primero.

Parámetros característicos de estas redes de interconexión son los siguientes:

- Degradación.

Una red no produce degradación si permite conectar, en cada momento, los procesadores a todos los módulos de memoria diferentes.

II Jornadas Hispano-Francesas de Informática
Teórica y Programación.

- Costo.

S. Sebastian, 15-17 de Enero de 1982.

Como en todo diseño, el costo es un parámetro importante que puede decidir el tipo de red a emplear en un sistema multiprocesador. El costo de una red de interconexión tiene dos componentes: la circuitería necesaria para realizar las conexiones y los circuitos necesarios para decidir qué conexiones se establecen en cada momento (árbitro).

- Arbitraje.

Vista la red como un sistema compuesto por enlaces virtuales procesadores-memorias, se denomina árbitro al mecanismo

A.- INTRODUCCIÓN.

Los sistemas multiprocesadores constituyen una solución para la obtención de sistemas informáticos que permiten obtener un alto grado de paralelismo en la ejecución de los algoritmos.

Uno de los muchos problemas encontrados en el diseño de tales sistemas, con elevado número de procesadores, es la interconexión entre los procesadores y los módulos de memoria. Al mecanismo de unión se le denomina red de interconexión. Básicamente existen dos soluciones extremas: la primera de ellas es aquella en la que todos los procesadores ven a toda memoria de la misma forma (Ejemplo, C.mmp) y la otra es aquella en la que los módulos de memoria son accedidos de forma diferente por los procesadores (Ejemplo, Cm*).

Nosotros nos centraremos en las redes de interconexión del tipo primero.

Parámetros característicos de estas redes de interconexión son los siguientes:

- Degradación.

Una red no produce degradación si permite conectar, en cada momento, los procesadores a todos los módulos de memoria diferentes que son solicitados. (Ejemplo: crossbar).

- Costo.

Como en todo diseño, el costo es un parámetro importante que puede decidir el tipo de red a emplear en un sistema multiprocesador. El costo de una red de interconexión tiene dos componentes: la circuitería necesaria para realizar las conexiones y los circuitos necesarios para decidir qué conexiones se establecen en cada momento (árbitro).

- Arbitraje.

Vista la red como un sistema compuesto por enlaces virtuales procesadores-memorias, se denomina árbitro al mecanismo

que asigna dichos enlaces virtuales a enlaces físicos. Dos son los parámetros fundamentales que caracterizan dichos árbitros: Complejidad (función de la red, número de procesadores y memorias y algoritmo de arbitraje) y tiempo necesitado en establecer los enlaces físicos.

- **Tiempo de tránsito.**

Se denomina tiempo de tránsito al tiempo necesitado por las señales en atravesar la red de interconexión. Parámetro asociado es el número de etapas que posee la red (Ej: 1 etapa en el crossbar y varias etapas en las redes barajadas).

- **Modularidad.**

Entendemos aquí por modularidad la propiedad que deben poseer las redes de interconexión para permitir aumentar el número de procesadores, módulos de memoria y canales virtuales a un costo lineal.

- **Fiabilidad.**

Considera la red como un conjunto de enlaces virtuales procesadores-memorias, entendemos por fiabilidad aquella característica que le permite continuar estableciendo conexiones entre todos los procesadores y módulos de memoria aún cuando fallen un número determinado de dichos enlaces virtuales. Lógicamente, una red es tanto mas fiable cuanto mayor es la independencia entre los enlaces virtuales y los enlaces físicos.

A continuación daremos algunos ejemplos de redes de interconexión así como una breve descripción de sus características.

a) **Unibus.**

Este tipo de red está constituido por un solo bus que conecta todos los procesadores a todos los módulos de memoria. El costo es bajo, necesita un árbitro N-usuarios, 1-servicio de baja complejidad y tiempo de arbitraje pequeño, su modularidad es alta; sin embargo su fiabilidad es nula y su degradación es altísima. Dicha red es empleada

en sistemas con bajo número de procesadores (Pulsar) o para las conexiones de procesadores fuertemente conectados localmente dentro de un sistema con varias agrupaciones del tipo anterior (Cm \times).

b) Crossbar.

En este tipo de red, los procesadores se conectan a las memorias a través de una matriz completa de conexiones. Si las solicitudes de los procesadores en un momento determinado es una permutación de módulos de memoria, todos acceden, por lo que la red no produce degradación. El costo es proporcional a N^2 (N =número de procesadores y módulos de memorias siendo prohibitivo para N grandes). La red es modular y poco fiable (si falla una columna de la matriz de conexión no se podrá acceder al módulo de memoria asociado. Para establecer las conexiones se necesitan N árbitros del tipo N -usuarios, 1-servicio equivalentes cada uno al necesitado en el unibus. Esta red se ha empleado en C.mmp siendo su costo, aproximadamente, el 30% del sistema total.

c) Multipuerta.

Esta red es funcionalmente equivalente al crossbar solo que el arbitraje y conmutadores se han distribuido en los módulos de memoria.

d) Multibus.

Esta red está formada por B buses ($1 \leq B \leq N$) que conectan los procesadores a los módulos de memoria. Si $B = 1$ se obtiene el Unibus y si $B=N$ se obtiene una red equivalente al crossbar en cuanto a degradación nula, pero con un costo mayor y con mejor fiabilidad.

Esta red produce degradación en aquellos instantes en los que existen solicitudes a mas de B módulos diferentes. El número de puntos de conexión necesitados es de $2BN$ (mayor que en el crossbar si $B > N/2$). Para asignar los buses es necesario un árbitro N -usuarios, B -servicios que posee una complejidad mayor que los N árbitros N -usuarios, 1-servicio necesitados por el crossbar y que, además, necesita un --

tiempo de arbitraje mucho mayor (este tiempo es función del número de módulos de memoria, N , del sistema). Estas dos características han originado, tal vez, el hecho de que no haya sido empleado el multibus en sistemas con procesadores rápidos relativamente a las velocidades de las memorias. Así pues, prácticamente no existen trabajos conocidos sobre el multibus de forma que en nuestra charla intentaremos describir algunos de los resultados obtenidos en nuestras investigaciones sobre el tema.

e) Redes de interconexión barajadas.

Constituyen como el multibus, una solución intermedia entre el unibus y el crossbar para el diseño de redes de interconexión. El diseño de dichas redes está formado por módulos elementales tipo crossbar de $Z \times Z$ entradas-salidas. El caso más estudiado es aquel en que $Z=2$ por lo que para $N=2^n$ procesadores y módulos de memoria la red está formada por $N/2 \times n$ módulos distribuidos en una matriz de $N/2$ filas y n columnas. Constituyen pues, redes de n etapas.

En cada columna de la red se emplea un bit de la dirección para discriminar las solicitudes que pasan a la siguiente columna o etapa: en cada módulo elemental pasan las dos solicitudes si tienen los bits diferentes y solo uno de ellos cuando son iguales. Cada módulo necesita un árbitro muy elemental y cuatro puntos de conexión, el tiempo de tránsito de ida y vuelta es proporcional a $2n$ y como hemos visto, producen una cierta degradación.

2: Diseño de un árbitro para la topología multibus.

Tal como hemos apuntado anteriormente, la red multibus necesita un árbitro de tipo N -usuarios, N -servicios. El diseño

B.- TEMAS A DESARROLLAR EN LA CHARLA.

Tal como hemos dicho en la introducción, hablaremos fundamentalmente de la red de interconexión tipo multibus. Dentro del tema general, podemos dividirlo en los subapartados siguientes:

- 1: Reducción de conexiones en el multibus y topologías asociadas.

Denominamos red multibus completa a aquella en la que cada uno de los B buses se conecta a cada procesador y a cada módulo de memoria. Siendo N el número de éstos últimos, el número de puntos de conexión necesitados es de $2 \cdot B \cdot N$. Dicho número es excesivo (mayor que los necesitados en el crossbar, N^2 , para $B > N/2$). Sin embargo, existen conexiones redundantes de forma que es posible eliminar algunas de ellas obteniéndose redes multibus que permiten el mismo ancho de banda que el multibus completo.

Matemáticamente, demostraremos que el número máximo de conexiones redundantes que se pueden eliminar sin degradar el ancho de banda es $B(B-1)$.

Partiendo de este resultado presentaremos redes basadas en el multibus completo y que hemos denominado: trapezoidal, rómbica, escalera, balanceada y cíclicas. Para cada una de ellas, presentaremos sus características de conexión, algoritmos de arbitraje, fiabilidad si se produce el fallo de un bus así como la modularidad en cuanto al crecimiento en el número de buses, procesadores y módulos de memoria.

- 2: Diseño de un árbitro para la topología multibus.

Tal como hemos apuntado anteriormente, la red multibus necesita un árbitro de tipo N-usuarios, B-servicios. El diseño

de dicho árbitro puede ser un parámetro crítico de la red. En este apartado presentaremos el diseño de un árbitro para un sistema multibus en el que los procesadores están sincronizados y donde se asignan los buses a los módulos de memoria solicitados de manera cíclica. Una vez que un módulo de memoria ha obtenido un bus, se asigna dicho bus a uno de entre los procesadores que solicitan dicha memoria por cualquiera de las técnicas ya empleadas en los árbitros N-usuarios, 1-servicio descritos ampliamente en la literatura técnica. Presentaremos tres diseños de menor a mayor complejidad y de mayor a menor tiempo necesitado por el arbitraje: el primero completamente iterativo y los dos últimos con uno y dos niveles de anticipación.

3: Obtención del ancho de banda del multibus mediante simulación. Redes multibus con buses partidos.

Aspecto importante de cualquier sistema multiprocesador es calcular el ancho de banda (número de accesos a memoria por ciclo de memoria). Dicho valor da una buena idea de la eficiencia del sistema y es función de la distribución de las peticiones a los módulos de memoria, del tiempo de servicio de las memorias, de los conflictos o degradación producidos por la red de interconexión, del tiempo necesitado para atravesar la red y del tiempo entre dos peticiones consecutivas de cada procesador.

Hemos dicho que el multibus produce degradación en aquellos instantes en los que existen mas de B peticiones a diferentes módulos de memoria. Importante es pues, conocer la degradación introducida en el ancho de banda.

Para calcular el ancho de banda del multibus, podríamos usar el modelo desarrollado en UCLA y basado en una cadena de Markov continua (exacta para B y N pequeños y aproximada para valores grandes); sin embargo, las hipótesis son restrictivas debido a que suponen distribución exponencial en las peticiones de los procesadores y en los tiempos de servicio de las memorias.

Así pues presentaremos los resultados obtenidos por simulación con las hipótesis siguientes:

- Los procesadores están sincronizados.
- Las peticiones de los procesadores son independientes y uniformemente distribuidos en los módulos de memoria.
- El tiempo de ciclo de las memorias es constante e igual en todas ellas.
- Cuando un procesador obtiene servicio de memoria en cada un ciclo, vuelve a solicitar servicio en el ciclo siguiente con probabilidad $p \leq 1$.
- Los tiempos de arbitraje de los buses y de propagación forman parte del tiempo de ciclo de la memoria.
- En cada ciclo, los buses son asignados de manera cíclica a los módulos de memoria que tienen al menos una petición. Para un módulo al que se le asigna un bus, se selecciona de manera aleatoria uno de los procesadores que solicitan dicha memoria.

Para $p=1$ se obtiene que el multibus con $B = N/2 + 1$ produce una degradación respecto al ancho de banda obtenido con $B = N$ (equivalente al crossbar) menor del 5%. Para $p < 1$ la degradación relativa es menor.

Estos resultados junto con los obtenidos en la reducción del número de conexiones nos llevan a la conclusión de que para $B = N/2 + 1$, el número de conexiones necesitadas en cualquier red multibus con conexiones no redundantes es

$$(N/2 + 1)(2N) - (N/2 + 1)(N/2) = 3/4 N^2 + 3/2 N$$

Para $N=6$, obtenemos 36 que coincide con las necesitadas por el crossbar. Para N menores el crossbar necesita menor número de conexiones y para N mayores el multibus necesita menos conexiones.

Basados en los resultados anteriores investigamos en redes de interconexión basadas en el multibus (con o sin reducción de conexiones) con los objetivos de reducir todavía más el número de conexiones así como el tiempo de asignación de los buses, a costa de disminuir en pequeña medida, el ancho de banda en comparación con el multibus. A la red resultante la hemos denominado multibus con buses parciales.

Esta red consiste también de B buses para conectar los procesadores a los módulos de memoria. Cada bus se conecta a cada procesador; sin embargo, los módulos de memoria se dividen en g grupos de N/g módulos cada uno y de forma que cada uno de los módulos de cada grupo se conecta a cada uno de los buses que forman un mismo grupo de B/g buses. En esta red, el número de conexiones es proporcional a $B(N + N/g)$ sin reducción de conexiones necesitando g árbitros del tipo N -usuarios, B/g -servicios que son más sencillos de implementar que el árbitro necesitado por el multibus y mucho más rápidos ya que gestionan el acceso a N/g módulos de memoria. La desventaja fundamental es que introducen un nivel adicional en la degradación del ancho de banda, frente al multibus, ya que cada subgrupo de N/g memorias permite B/g accesos como máximo.

Para evaluar esta nueva degradación hicimos de nuevo simulaciones. Como conclusiones breves diremos que para valores de N considerados (4, 8, 12, 16) y $p=1$ la configuración con $g=2$ y $B=N/2 + 2$ (cada uno de los dos subgrupos estaba conectado a $N/4 + 1$ buses) el ancho de banda obtenido era mayor que con el multibus con $B=N/2 + 1$, de forma que los costos de la red y tiempos de arbitraje necesarios eran menores. Con $p < 1$ se cumple todavía más la afirmación anterior.

Basados en los resultados anteriores investigamos en redes de interconexión basadas en el multibus (con o sin reducción de conexiones) con los objetivos de reducir todavía más el número de conexiones así como el tiempo de asignación de los buses, a costa de disminuir en pequeña medida, el ancho de banda en comparación con el multibus. A la red resultante la hemos denominado multibus con buses parciales.

Esta red consiste también de B buses para conectar los procesadores a los módulos de memoria. Cada bus se conecta a cada procesador; sin embargo, los módulos de memoria se dividen en g grupos de N/g módulos cada uno y de forma que cada uno de los módulos de cada grupo se conecta a cada uno de los buses que forman un mismo grupo de B/g buses. En esta red, el número de conexiones es proporcional a $B(N + N/g)$ sin reducción de conexiones necesitando g árbitros del tipo N -usuarios, B/g -servicios que son más sencillos de implementar que el árbitro necesitado por el multibus y mucho más rápidos ya que gestionan el acceso a N/g módulos de memoria. La desventaja fundamental es que introducen un nivel adicional en la degradación del ancho de banda, frente al multibus, ya que cada subgrupo de N/g memorias permite B/g accesos como máximo.

Para evaluar esta nueva degradación hicimos de nuevo simulaciones. Como conclusiones breves diremos que para valores de N considerados (4, 8, 12, 16) y $p=1$ la configuración con $g=2$ y $B=N/2 + 2$ (cada uno de los dos subgrupos estaba conectado a $N/4 + 1$ buses) el ancho de banda obtenido era mayor que con el multibus con $B=N/2 + 1$, de forma que los costos de la red y tiempos de arbitraje necesarios eran menores. Con $p < 1$ se cumple todavía más la afirmación anterior.



4: Modelos matemáticos aproximados.

Para cualquiera de las redes de interconexión descritas en la introducción existen modelos matemáticos exactos o aproximados que evalúan el ancho de banda del sistema, el tiempo medio de espera, ... bajo diferentes hipótesis de funcionamiento del sistema.

Para el caso concreto del multibus parece muy difícil, debido al excesivo número de estados, encontrar un modelo basado en una cadena de Markov discreta o continua que nos permite obtener los valores de las magnitudes anteriores. Tal como hemos dicho, conocemos un trabajo que obtiene una cadena de Markov continua bajo hipótesis muy restrictivas.

Una técnica muy usada en la evaluación de las redes de interconexión es encontrar una fórmula que nos evalúa las magnitudes anteriores bajo las hipótesis de procesadores sincronizados, tiempos de servicio de las memorias constantes e iguales, distribución binomial de las peticiones así como distribución uniforme a todos los módulos de memoria y consideración del sistema como si no tuviera memoria. (Al final de cada ciclo los procesadores que no obtuvieron acceso vuelven a pedir de nuevo; hipótesis claramente que no se ajusta a la realidad). En todas las redes estudiadas con estas hipótesis se obtienen resultados optimistas (ancho de banda mayor que el real) pero con un margen de error razonable ($< 5\%$).

Basados en el conocimiento real del sistema (obtenido por simulación) presentaremos modelos matemáticos aproximados para el multibus y multibus con buses parciales con $p < 1$. La mayoría de los resultados obtenidos tienen un margen de error inferior al 5% en relación con los obtenidos por la simulación.

C.- CONCLUSIONES.

Para el caso de que los procesadores sean microprocesadores y junto a las características relativas de las memorias actuales, la red multibus y asociadas parecen una solución como redes de interconexión para los sistemas multimicroprocesadores.

Los trabajos presentados en la charla están orientados a minimizar las desventajas presentadas por el multibus frente al crossbar:

- En cuanto al número de conexiones, hemos visto que se pueden ahorrar $B(B-1)$ reduidantes. Para $N=6$ y $B=4$, la degradación producida por el multibus es menor del 5% en relación con el crossbar necesitándose el mismo número de conexiones. Sin embargo, el multibus admite funcionamiento en modo degradado cuando falla un bus (característica no admitida por el crossbar).
- Con una degradación adicional pequeña, se ha presentado una red de interconexión denominada multibus con buses parciales que necesita de un número menor de conexiones así como de árbitros mas sencillos en su complejidad y que ejecutan su algoritmo de arbitraje en menor tiempo.
- Se ha presentado el diseño de un árbitro para el multibus completo y que sirve lógicamente para arbitrar cada uno de los grupos de buses en el multibus can bases parciales. El diseño con dos niveles de anticipación permite aprovechar al máximo los niveles de integración actuales, minimizando el tiempo de arbitraje.
- Por último se presentan resultados de simulación y de modelos analíticos aproximados que permiten evaluar el ancho de banda del multibus y multibus con buses parciales para los casos en los que los procesadores poseen o no procesado interno.

Los resultados presentados parecen satisfactorios y están siendo continuados en los siguientes aspectos:

- Obtención de redes basadas en el multibus y con reducción de conexiones totalmente equilibradas. Dichas redes permitirán mejorar las características eléctricas de los circuitos drivers de los buses.
- Obtención de árbitros para dichas redes de interconexión, ya que cuanto menor es la información en las conexiones (redes balanceadas o cíclicas) mayor es la dificultad para realizar árbitros que no degraden por su algoritmo de asignación, al ancho de banda del sistema.
- Referente al diseño del árbitro, hemos de incorporar mecanismos adicionales que permitan la realización de secciones críticas.
- En relación con los modelos matemáticos parece realmente difícil encontrar nuevos modelos que nos den una mejor información del Sistema; más en concreto, parece difícilísimo encontrar formulas cerradas que nos permitan conocer los valores de la degradación en función de B y N.
- Por último creemos que las redes multibus, así como ocurre en las redes barajadas, pueden ser usadas en modo paquete para aumentar su eficiencia. Normalmente y tal como hemos entendido en nuestro trabajo, cuando un procesador adquiere un bus, éste es de su propiedad hasta que se termina de realizar la operación de lectura o escritura con la memoria; durante todo ese tiempo, el bus no puede ser utilizado. Normalmente el tiempo de tránsito a través del bus es pequeño comparado con el tiempo de ciclo de las memorias por lo que es posible pensar en interconexiones que funcionen tipo paquetes: cuando un procesador obtiene un bus, lo toma y envía su información al módulo de memoria correspondiente de forma que cuando dicha información llega al módulo, el bus queda liberado para otra nueva operación. Cuando la memoria

El GALILEO es un lenguaje gráfico, un conjunto de herramientas, y una metodología desarrolladas para el diseño, desarrollo y programación de los sistemas de tiempo real.

F. VIDONIA, M. KRAS,
E. LOPEZ,
J. J. GIBO,
Centro de I+D+i

haya realizado la operación, deberá obtener un bus para informarle al procesador correspondiente. Nuestros últimos trabajos sobre el multibus se están orientando por este camino.

Introducción

Los sistemas de control de comutación actuales (por ejemplo MENTA-CONTA* 2000, Sistema 12) son sistemas de tiempo real con un alto grado de concurrencia y paralelismo. Desgraciadamente los lenguajes de diseño y programación existentes, no son adecuados para representar la concurrencia, las interacciones de sincronización, y el estado de las actividades que pueden dar origen a bloqueos. Existe, por lo tanto, una necesidad de encontrar una metodología adecuada para el diseño, la representación y documentación de estos sistemas.

Inicialmente, las redes PERT se utilizaron para representar el paralelismo que existía en los procesos y tareas de tales sistemas, y las relaciones de precedencia (comienzo-comienzo, comienzo-terminación, terminación-comienzo, terminación-terminación). Al mismo tiempo, la utilización de marcas o "tokens" sirvió para representar el estado del sistema y la secuencia de las actividades de los procesos. A partir de entonces se realizó una investigación en la búsqueda y evaluación de otros métodos alternativos que pudieran eventualmente ayudar en la problemática que presenta el diseño de sistemas de tiempo real complejos. Un artículo de Tilak Agrwala [1] nos llevó a la investigación de las redes de Petri debido a su aparente potencia, simplicidad y soporte matemático.

Se realizó un estudio profundo de las redes de Petri y se aplicaron para representar ciertos ejemplos clásicos de sistemas de tiempo real asíncronos y concurrentes. Sin embargo, aunque las redes de Petri resultaron ser una poderosa herramienta para representar estos sistemas, para ciertas situaciones simples, resultaban crínicas y difíciles de aplicar. Por otra parte, y basados en la necesidad de encontrar un método para la documentación de la programación en tiempo real, que permitiera producir documentos comprensibles, se realizó un estudio en profundidad para representar un sistema operativo y se establecieron los siguientes objetivos:

- Evaluar la potencia de representación e identificar los puntos débiles.
- Crear un catálogo de funciones y operadores en forma de módulos o bloques arquitectónicos que permitiera el diseño de sistemas complejos con control distribuido, considerando bloques tomados de este catálogo.
- Resolver el problema de comprensión en la legibilidad de programación.

Las redes de Petri no alcanzaron los objetivos y, finalmente, fueron sustituidas por un lenguaje basado en Petri

y descrito por Mazurkiewicz [2]. El lenguaje de Mazurkiewicz es un lenguaje gráfico textual, que representa la transición mediante cajas, permitiendo la inclusión de ellas de un texto explicativo relacionado con el proceso acción, o escribiendo una instrucción o varias de un lenguaje de programación en cada una de estas cajas.

El catálogo de bloques arquitectónicos y la experiencia ganada durante esta investigación, fueron las bases de nuevo lenguaje: el GALILEO. Este lenguaje ha sido especialmente desarrollado para ayudar a los diseñadores producir diagramas claros, no ambiguos, e independientes de las implantaciones en equipo/procesador. Es especialmente adecuado para el tratamiento de la concurrencia, sincronización, sincronismo, y otras características encontradas en los sistemas distribuidos y de tiempo real [3, 4, 5, 7].

Lenguaje GALILEO para el diseño de sistemas

Este lenguaje de diseño permite representar los sistemas en forma abstracta, en el sentido de que sea lo más independiente posible de la implantación; el modelo de datos puede ser verificado y reutilizado, y su comportamiento dinámico simulado.

El GALILEO es un lenguaje de diseño y no de programación. Permite separar el control y el flujo de datos y presentar los procesos secuenciales y concurrentes en su forma abstracta independiente de su implantación. Especialmente es un lenguaje gráfico que se usa para dibujar diagramas consistentes en una red con un texto complementario. Este texto puede ser expresado utilizando un lenguaje de transición, un pseudolenguaje o un lenguaje de programación; por ejemplo, el pseudolenguaje usado en este artículo está basado en el Pascal.

Los elementos gráficos del lenguaje GALILEO son representados en la figura 1 y se utilizan para construir diagramas que representan la red de acciones y flujo de datos de un bloque funcional de diseño o bloque G. Dicho blo que G puede estar formado por otros bloques componentes y, en última instancia, permitir representar un modelo sistema.

Los elementos básicos del lenguaje son lugares, transiciones y arcos. El lugar contiene datos; la transición contiene las acciones que van a ser realizadas; y el arco entre un lugar con una transición y establece que las acciones una transición operarán con los datos suministrados por lugar al cual está conectada la transición. Los lugares representan los lugares de control y lugares de datos, dependiendo del tipo de datos que contengan.

*Nota registrada del sistema IIT