

## RESUMEN Y COMPARATIVA ENTRE LOS ALGORITMOS DE MARCADOR Y TOMASULO

P. Pérez-Romero,  
J. N. Alba-Juárez,  
J. A. Rodríguez-Meza  
R. Macias-Igari

Instituto Politécnico Nacional, CIDEFEC.  
Área de Arquitecturas Avanzadas y Paralelas.  
Unidad Profesional Adolfo López Mateos. C.P.  
07700, México, D. F., MÉXICO.

### RESUMEN

En este artículo se presenta de manera resumida el funcionamiento y comparativa entre los algoritmos Scoreboard, (en lo sucesivo marcador) y el algoritmo conocido como Tomasulo; los cuales son utilizados para la planificación dinámica de instrucciones. Primeramente, se presenta una descripción sintetizada del funcionamiento del algoritmo del marcador. Seguido a esto se muestra también de manera simplificada, el proceso de ejecución realizado por el algoritmo de Tomasulo para la planificación de instrucciones. Aunado a esto se realizó una comparativa de ambos, con el propósito de identificar ventajas y desventajas entre ellos.

### 1. Introducción

El uso de la planificación dinámica de instrucciones se comenzó a desarrollar y a utilizar cuando los procesadores y compiladores requerían evitar riesgos, peligros o paradas de rendimiento debidos a las dependencias de datos, ya que al ser secuencial el orden de las instrucciones, se provocaban estas incidencias; de esta manera con el objetivo de resolver estas problemáticas y minimizar las paradas se utilizaron los algoritmos del marcador y Tomasulo desde hace ya algunas décadas. Con esto se logró que las instrucciones pudieran ejecutarse fuera de orden, es decir, no de forma secuencial sino por medio de planificación dinámica. La primera implementación del algoritmo del marcador fue en 1966 y Robert Tomasulo implementó su algoritmo por primera vez en 1969, para el procesador de IBM 360/91. 2.

### 2. Algoritmo del Marcador

El método o algoritmo del marcador, ver Figura 1(a), fue implementado en la computadora CDC 6600. El objetivo principal era optimizar una instrucción por ciclo de reloj, es decir este algoritmo ejecuta una instrucción lo más rápido posible, siempre y cuando no existan riesgos estructurales; si una instrucción se detiene, otras instrucciones pueden ser ejecutadas. Esto solo puede realizarse si y sólo si, dichas instrucciones no dependen de alguna instrucción que este activa o detenida en ese instante de tiempo. Para salvaguardar lo anteriormente mencionado se hace uso de un marcador, el cual tiene la función de manejar la emisión, la ejecución y la detección de riesgos de las instrucciones. El algoritmo del marcador consta de un pipeline de cuatro etapas. Además, se requiere de estructuras de datos, en donde se van registrando la dependencia de datos entre los operandos de las instrucciones que han entrado en el flujo. Es importante recalcar que cada instrucción pasa por las cuatro etapas que a continuación se describen en forma de resumen, ver Figura 1(b):

Emisión: Se comprueba que la instrucción necesitada por la unidad funcional esté libre y además se verifica que alguna otra instrucción activa tenga el mismo registro destino. En el caso de ocurrir esto

último (riesgo tipo WAW, Write After Write, por sus siglas en Inglés), el marcador tiene la tarea de emitir la instrucción, actualizando de esta forma su estructura interna de datos. Si existe un riesgo estructural o un riesgo WAW, la emisión de la instrucción se detiene hasta que desaparezcan dichos riesgos.

**Lectura de datos:** El marcador tiene la tarea de comprobar si los operandos de entrada están disponibles. Un operando se dice estar libre si no existe alguna instrucción activa, ya emitida, que vaya a escribir en tal operando. Una vez que los operandos están a disposición, el marcador le dice a la unidad funcional que lea los operandos de los registros correspondientes para posteriormente iniciar la ejecución.

**Ejecución:** En esta etapa el marcador recibe los operandos y comienza la ejecución de ellos. Una vez finalizada la ejecución el marcador recibe una notificación de que el resultado ahora está disponible.

**Escritura de resultados:** La unidad funcional asignada a la instrucción finaliza la operación, en este momento el marcador recibe esta información y comprueba que no existan riesgos del tipo WAR (Write After Read). Si tal riesgo existiese, el marcador detiene la escritura del resultado hasta solventar el riesgo. Finalmente, el marcador le indica a la etapa de escritura que escriba el resultado en el registro destino.

@@@

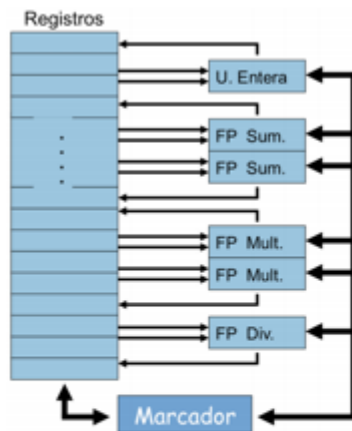


Figura 1(a). Marcador.



Figura 1(b). Etapas del marcador.

Figura 1. Método del marcador.

Por otro lado, el estado de los objetos que controla el marcador está dividido en tres estructuras de datos los cuales se mencionan a continuación:

**Estado de las instrucciones:** Básicamente indica la etapa en que se encuentra cada una de las instrucciones activas.

**Estado de las unidades funcionales:** Es una tabla que indica precisamente el estado de cada unidad funcional, dicha tabla consta de nueve campos.

**Estado de los registros:** Indica a la unidad funcional que deberá escribir en cada uno de sus registros, destino de alguna instrucción activa.

### 3. Algoritmo de Tomasulo

El algoritmo de Tomasulo, es otro método para implementar la planificación dinámica. Fue inventado por Robert Tomasulo en 1967 para permitir a los procesadores ejecutar instrucciones fuera de orden, ver Figura 2. La principal diferencia entre el algoritmo del marcador y el algoritmo de Tomasulo es que este último, permite el renombramiento de registros solucionando el problema WAW y WAR

que se presenta en el algoritmo del marcador. El mayor aporte del algoritmo de Tomasulo es que permite renombrar registros por medio de hardware, estaciones de reserva para todas las unidades de ejecuciones y un bus de datos común en donde todos los valores procesados pueden ser transmitidos a todas las estaciones de reserva que pudieran necesitarlos.

□@|@

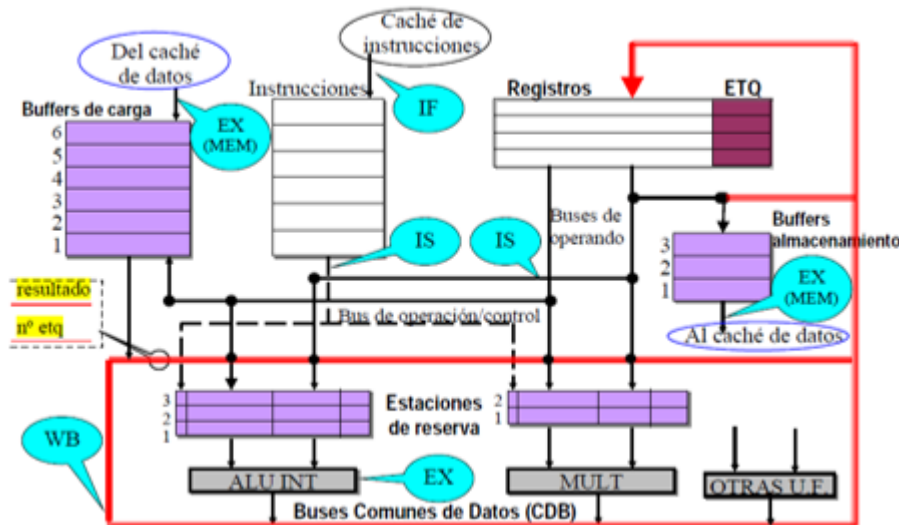


Figura 2. Diagrama esquemático del algoritmo Tomasulo

Las siguientes definiciones son necesarias para la implementación del algoritmo de Tomasulo:

**Unidad de almacenamiento y decodificación:** Es la encargada de emitir las instrucciones a las estaciones de reserva, donde las instrucciones esperan para su posterior ejecución en las unidades funcionales. Esta unidad es igual a la utilizada en la planificación estática.

**Unidades Funcionales:** Son aquellos lugares donde se ejecutan las instrucciones, pueden ser unidades de punto flotante, unidades aritméticas/lógicas o unidades de carga/almacenamiento.

**Estaciones de Reserva:** Pueden ser una o varias dependiendo del tipo de unidad funcional, se encargan de almacenar las instrucciones hasta que puedan ejecutarse en las unidades funcionales.

**Bus de datos común (CBD, por sus siglas en inglés Common Bus Data):** Es el encargado de establecer los canales de comunicación entre las estaciones de reserva y las unidades funcionales. Según Tomasulo "Preserva la precedencia fomentando al mismo tiempo la concurrencia".

Una vez comprendidas las definiciones anteriores, el siguiente paso para la implementación del algoritmo de Tomasulo es entender las etapas que lo conforman. En la Figura 3 se muestra un diagrama esquemático con las etapas del algoritmo de Tomasulo. Las tareas que debe realizar cada etapa se describen a continuación:

**Emisión:** Esta etapa es la encargada de comprobar si hay una estación de reserva libre (correspondiente al tipo de operación a realizar) o un buffer de carga/almacenamiento y si es así, se envía la instrucción. Es importante mencionar que al final de esta etapa la instrucción ya queda en la instrucción de reserva. Si se diera el caso en que la estación de reserva no está desocupada, se produce un riesgo estructural y se detiene la instrucción hasta que dicha estación se libere. En esta etapa también se realiza el renombramiento de registros, eliminando los problemas WAR y WAW presentados en el algoritmo del marcador.

**Ejecución:** En esta etapa las operaciones correspondientes a cada instrucción son llevadas a cabo. Todas las instrucciones son retrasadas en este paso hasta que todos sus operandos están disponibles, eliminando problemas RAW.

Escritura de resultados: La última etapa es donde se lleva a cabo el almacenamiento de los resultados obtenidos en la etapa de ejecución. La escritura es realizada escribiendo en CDB, y de ahí, a los registros generales y a las estaciones de reserva o buffer de almacenamiento que estén esperando dicho resultado.

[]@|@



Figura 3. Etapas del algoritmo de Tomasulo

#### 4. Comparación de los algoritmos del marcador y Tomasulo

Primeramente, el algoritmo del marcador presenta cuatro etapas para desarrollar la planificación dinámica descritas en este artículo, mientras que, el algoritmo de Tomasulo únicamente requiere de tres etapas para realizar dicha planificación. Por otro lado, el control de flujo de datos e instrucciones para el algoritmo del marcador es centralizado, es decir, es el encargado de verificar que todas las etapas que conforman a dicho algoritmo se realicen de manera satisfactoria. Mientras que, el algoritmo de Tomasulo almacena las instrucciones haciendo uso de estaciones de reserva. Además, el algoritmo del marcador utiliza una memoria de registros para las instrucciones, a diferencia del algoritmo de Tomasulo que utiliza apuntadores a las estaciones de reserva; asimismo, permite el renombramiento de registros de memoria, evitando así los riesgos WAR y WAW, los cuales son una limitante importante en el algoritmo del marcador. Del mismo modo, Tomasulo cuenta con un bus de datos común (CDB), lo que permite difundir todos los resultados a todas las unidades funcionales, algo con lo que no cuenta el algoritmo del marcador. Finalmente, la complejidad del algoritmo de Tomasulo es mayor a la complejidad del algoritmo del marcador.

#### 5. Conclusiones

En este artículo se ha presentado una comparación entre el algoritmo del marcador y el algoritmo de Tomasulo. Para llevar a cabo la comparación se describieron los puntos fundamentales que hay que comprender para poder llevar a cabo dichos algoritmos, así como las diferencias que existen entre los mismos. Podemos concluir que los dos algoritmos cumplen con la planeación dinámica de instrucciones, pero el algoritmo de Tomasulo resulta ser una mejora del algoritmo del marcador ya que sólo tiene tres etapas mientras que el algoritmo del marcador tiene cuatro. Además, el algoritmo de Tomasulo permite renombrar registros para evitar los errores de WAR y WAW algo que no es posible con el algoritmo del marcador. Por último es importante mencionar que los dos algoritmos realizan la escritura de datos en su última etapa, por lo que ambos son funcionales para la planificación dinámica de instrucciones.

#### 6. Referencias

- [1] Dynamic Scheduling Techniques: Consultado el 6 de Octubre de 2015. Disponible en: <http://home.deib.polimi.it/silvano/FilePDF/ARC-MULTIMEDIA/Dynamic%20Scheduling.pdf>
- [2] Planificación Dinámica de Instrucciones. Consultado el 5 de Octubre de 2015. Disponible en: <http://www.dia.eui.upm.es/asignatu/Arq.com/AC%20Grado/Paco%20Aylagas/4-Planificacion%20Dinamica.pdf>
- [3] Scoreboarding and Tomasulo Algorithm. Consultado el 3 de Octubre de 2015. Disponible en: [http://users.utcluj.ro/~sebestyen/.Word\\_docs/Cursuri/SSC.course\\_5.Scoreboard.ex.pdf](http://users.utcluj.ro/~sebestyen/.Word_docs/Cursuri/SSC.course_5.Scoreboard.ex.pdf)

