

MODELADO PARA LA GESTIÓN DE RECURSOS DE COMPUTACIÓN EN SISTEMAS SDR

Vuk Marojevic, Xavier Revés, Antoni Gelonch

marojevic@tsc.upc.edu, xavier.reves@tsc.upc.edu, antoni@tsc.upc.edu

Dpto. de Teoría de Señal y Comunicaciones, Universidad Politécnica de Cataluña, C/Jordi Girona 1-3, 08034 Barcelona.

Abstract- Software-defined radio (SDR) is an emerging concept that leverages the design of software-defined and hardware-independent signal processing blocks for radio communications. A future SDR platform will be subject to the dynamic reconfiguration of its radio functionality. An SDR application, which defines this functionality, will require a certain amount of the limited computing resources for real-time processing. This leads to the problem of dynamic computing resource allocation in software-defined radio. To efficiently solve this problem an appropriate system modeling is necessary, which is the subject of this paper. We introduce a mathematical-graphical modeling of SDR systems. A simple computing resource allocation method and realistic case study demonstrate the suitability of the proposal.

I. INTRODUCCIÓN

La tendencia actual en las comunicaciones radio es la transición de las radios definidas completamente en hardware hacia radios definidos por software. Es decir, las implementaciones software del procesamiento de señal para las radiocomunicaciones sustituirán las implementaciones hardware de hoy en día. El concepto *software-defined radio* (SDR) es el que define este planteamiento [1] [2].

Las futuras *plataformas SDR*, tanto los terminales móviles como las estaciones base, contarán con dispositivos reprogramables, como los DSPs, y dispositivos reconfigurables, como las FPGAs. La reconfiguración de estos dispositivos puede cambiar la funcionalidad radio de la plataforma. Para un DSP esto supone el cambio del software que se ejecuta, para una FPGA el traspaso de un nuevo fichero de configuración. Cualquiera de los dos tipos de configuración cambia parcialmente o completamente la funcionalidad de la plataforma.

El concepto SDR no exige que todo el transceptor radio sea definido mediante software. El software que define la funcionalidad radio (o parte de ella) de una plataforma SDR es la *aplicación SDR* que consiste de varias *funciones SDR*. Una función SDR representa un bloque de procesamiento de señal definido en software, como, por ejemplo, un filtro, un descodificador o un receptor RAKE.

Desde nuestro punto de vista y en contraste a las aplicaciones radio tradicionales, las aplicaciones SDR serán similares a las numerosas aplicaciones de propósito general de hoy en día. Es por ello que argumentamos que los métodos de propósito de gestión de la computación general, como es el mapeo y el *scheduling*, deberían ser considerados en el ámbito de software radio. El mapeo describe la asignación de módulos software a recursos hardware. El

scheduling especifica el orden de ejecución de los módulos software previamente mapeados.

Los trabajos previos relativos a esta gestión de la computación, tanto en el ámbito de los procesadores de propósito general como en del software radio, tratan conjuntamente los problemas de mapeo y *scheduling* con el objetivo de minimizar el tiempo de ejecución total de una aplicación [3]–[8]. Sin embargo debe tenerse en cuenta que en el ámbito de software radio es imprescindible cumplir todos los requerimientos del sistema, incluyendo las demandas en tiempo real y los recursos de cómputo limitados. Ello debe suponer el objetivo principal dentro del entorno SDR [9].

Este artículo trata el modelado del sistema SDR como base para resolver el problema de asignación de recursos computacionales. Para ello se debe tener en cuenta que la disponibilidad de recursos de computación de una plataforma SDR, además de ser muy limitados en general, cambia dinámicamente. Por otro lado, la aplicación SDR tiene restricciones temporales estrictas.

El resto del artículo está organizado de la manera siguiente: Después de introducir el modelado del sistema SDR en la parte II, discutimos un método simple de asignación de recursos computacionales, un estudio de casos prácticos y los resultados de las simulaciones correspondientes. Finalmente, en la sección IV derivamos las principales conclusiones.

II. MODELADO DEL SISTEMA SDR

Antes de iniciar el modelado del sistema SDR debemos asumir la existencia de capas de abstracción a nivel de hardware y software que proporcionan la información necesaria sobre los recursos actualmente disponibles y que además facilitan la ejecución de aplicaciones SDR de manera solapada (*pipeline*). A base de estas abstracciones introducimos el modelado del sistema en que implícitamente se considera el tiempo de computación como el recurso de base. En consecuencia, una asignación de recursos computacionales que cumpla con todas las restricciones del sistema indica que la aplicación SDR se podrá ejecutar dentro de la ventana temporal permitida.

Un sistema SDR en general consiste de una plataforma SDR y una o más aplicaciones SDR. En los siguientes párrafos se describe el modelado matemático-gráfico que planteamos.

A. Modelado de plataformas SDR

Los recursos de computación de una plataforma SDR son las capacidades de procesamiento de los N procesadores heterogéneos y los anchos de banda para la transmisión de información entre ellos. Un valor de N de entre 1–3 puede ser válido para un terminal de usuario y entre 10–20 para una estación de base, la cual se comparte entre múltiples usuarios. Un procesador representa un dispositivo genérico dentro del ámbito de software radio, típicamente un DSP o una FPGA. La mencionada capa de abstracción esconde este tipo de heterogeneidad y proporciona la información sobre las capacidades de procesamiento en MOPS (Millones de Operaciones Por Segundo) y los anchos de banda en MBPS (Mega-Bits Por Segundo) disponibles en cada momento.

Las capacidades de procesamiento se resumen en

$$C = (C_1, C_2, \dots, C_N) \text{ [MOPS]}, \quad (1)$$

donde C_k representa la capacidad de cómputo del procesador P_k . La capacidad total de la plataforma SDR es

$$C_{\text{total}} = C_1 + C_2 + \dots + C_N \text{ [MOPS]}. \quad (2)$$

Asignamos los procesadores en orden descendente de sus capacidades de procesamiento, así que $C_1 \geq C_2 \geq \dots \geq C_N$.

El ancho de banda entre el procesador P_i y procesador P_j es B_{ij} ($i, j \in 1, 2, \dots, N$). Asumimos una memoria compartida de suficiente capacidad para facilitar la comunicación interna inmediata, con lo cual la matriz B se puede escribir como

$$B = \begin{pmatrix} \infty & B_{12} & \dots & B_{1N} \\ B_{21} & \infty & \dots & B_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N1} & B_{N2} & \dots & \infty \end{pmatrix} \text{ [MBPS]}. \quad (3)$$

El ancho de banda acumulado de una plataforma SDR es el B_{total} ,

$$B_{\text{total}} = \sum_{i=1}^N \sum_{j=1, j \neq i}^N B_{ij} \text{ [MBPS]}. \quad (4)$$

Nótese que B_{total} acumula los anchos de banda externos.

Asimismo los grafos dirigidos (*directed graphs*) pueden modelar gráficamente las plataformas SDR. En este caso los nodos representan los procesadores y los arcos las comunicaciones unidireccionales.

La figura 1 ejemplifica el modelado expuesto, ilustrando una plataforma SDR parcialmente y dinámicamente reconfigurable. Esta plataforma representa un futuro terminal móvil SDR y consiste de tres procesadores heterogéneos todos ellos interconectados entre sí de forma bidireccional. La capacidad de cómputo total es de $30\,000 \cdot S_C$, mientras que el ancho de banda total es de $6000 \cdot S_B$. S_C y S_B son dos factores de escalado individuales en unidades de MOPS y MBPS que se usarán para simular la reconfigurabilidad dinámica de la plataforma.

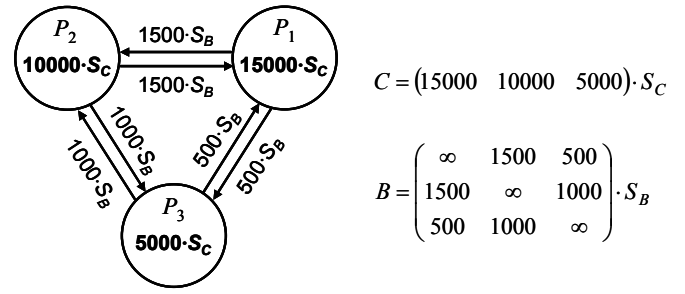


Fig. 1. Modelado gráfico y matemático de una plataforma SDR.

B. Modelado de aplicaciones SDR

Una aplicación SDR se modela como un conjunto de M funciones SDR f_1, f_2, \dots, f_M que procesan y traspasan datos entre ellos. Una función f_i ($i \in 1, 2, \dots, M$) pertenece a una cadena de dos o más funciones. Los grafos dirigidos y acíclicos (*directed acyclic graphs* o DAGs) modelan estas cadenas de funciones SDR, donde un nodo representa una función SDR y un arco una transferencia de datos. Las funciones SDR se numeran según el concepto de numeración lógica: si f_i envía datos a f_j , entonces $i < j$ [10].

El modelado matemático de una aplicación SDR incluye

$$c = (c_1, c_2, \dots, c_M) \text{ [MOPS]}, \quad (5)$$

que absorbe los demandas de procesamiento, y donde

$$b = \begin{pmatrix} 0 & b_{12} & \dots & b_{1M} \\ 0 & 0 & \dots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \text{ [MBPS]} \quad (6)$$

especifica los requerimientos de ancho de banda. Las demandas totales de computación y ancho de banda se definen como

$$c_{\text{total}} = c_1 + c_2 + \dots + c_M \text{ [MOPS]} \quad (7)$$

$$b_{\text{total}} = \sum_{i=1}^M \sum_{j=i+1}^M b_{ij} \text{ [MBPS]}. \quad (8)$$

Como ejemplo modelamos el procesamiento digital de señal de la capa física de un receptor UMTS en el lazo ascendente. La figura 2 contiene el modelado gráfico de esta aplicación SDR indicando los diferentes bloques de procesamiento. Los diferentes requerimientos se han obtenido de [11] [12], entre otros, y de implementaciones reales.

C. Modelado de la relación entre aplicaciones y plataformas

El parámetro *c-load*,

$$c\text{-load} = c_{\text{total}} / C_{\text{total}}, \quad (9)$$

representa la relación entre el requerimiento total de procesamiento de una aplicación SDR (7) y la capacidad de computación disponible de una plataforma SDR (2). Igualmente,

$$b\text{-load} = b_{\text{total}} / B_{\text{total}} \quad (10)$$

captura la relación entre el requerimiento total de ancho de banda de una aplicación SDR (8) y la capacidad total de ancho de banda de una plataforma SDR (4).

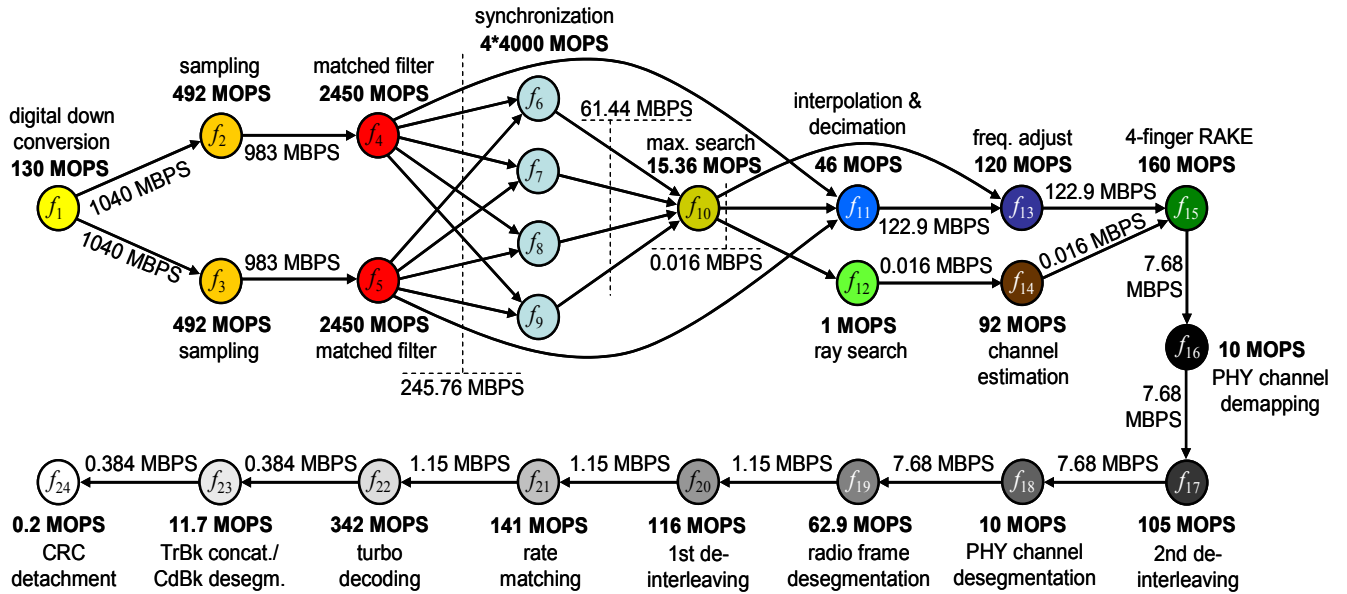


Fig. 2. Modelado gráfico de un aplicación SDR con $c_{\text{total}} = 23\,247.16$ MOPS y $b_{\text{total}} = 7030.18$ MBPS.

III. SIMULACIONES

A. Método de asignación de recursos computacionales

Para la correcta asignación de los recursos computacionales hace falta especificar un método adecuado que mediante la utilización de un algoritmo adecuado permita minimizar una función de coste. Con N procesadores y M funciones SDR, el número de posibilidades es N^M . En estas condiciones el problema de encontrar una solución óptima es NP completo [4] y por ello, son comunes las soluciones subóptimas. Sin embargo el objetivo del presente artículo es demostrar la utilidad del modelado propuesto, por lo cual consideramos un algoritmo básico y una simple función de coste.

El algoritmo básico *g-mapping* mapea una función SDR a cada instante de decisión empezando con f_1 y acabando con f_M , asignando la función SDR a ejecutar al procesador que lleve asociado con el mínimo coste según la función de coste utilizada. Con otras palabras, el algoritmo mapea f_i ($i \in 1, 2, \dots, M$) a P_1, P_2, \dots, P_N y añade esta decisión al mapeo de los primeros $i-1$ funciones SDR.

El coste de los N opciones de mapeo en cada punto de decisión se calcula según

$$\text{cost}(k, i) = \text{cost}_{\text{comp}}(k, i) + \text{cost}_{\text{com}}(k, i). \quad (11)$$

El término $\text{cost}(k, i)$ representa el coste de mapear f_i a P_k ($i \in 1, 2, \dots, M$; $k \in 1, 2, \dots, N$) y es, para $i > 1$, una función de las decisiones anteriores. El coste de computación $\text{cost}_{\text{comp}}(k, i)$ se obtiene como el cociente entre el requerimiento en capacidad de procesado c_i de la función SDR f_i y la capacidad restante del procesador P_k . La suma de hasta $i-1$ cocientes entre los anchos de banda necesarios (para las transferencias de datos entre f_1 y f_i, f_2 y f_i, \dots, f_{i-1} y f_i) y los correspondientes anchos de banda actualmente disponibles define el coste de comunicación $\text{cost}_{\text{com}}(k, i)$.

A lo largo del proceso de mapeo las capacidades de procesado y ancho de banda restantes se actualizan dinámicamente. De esta forma el algoritmo destaca y descarta cualquiera asignación no factible, lo que es una asignación que reserve más de 100% de cualquier recurso computacional [9].

B. Estudio de casos prácticos

Asumimos el marco siguiente: Un terminal móvil reconfigurable (Fig. 1) está trabajando en un modo, por ejemplo GSM. Debido a un cambio en el comportamiento de otros abonados móviles en la célula, la tecnología de acceso radio (*radio access technology* o RAT) UMTS se vuelve disponible y ofrece una calidad de servicio más alta que la RAT en uso. El terminal multimodo incorpora la parte de RF de un transceptor UMTS incluyendo los conversores A/D y D/A. La cadena de procesado digital de señal correspondiente se tiene que bajar (*software download*) e implementar. El terminal negocia la transmisión de este software, que incluye la cadena de procesado del receptor UMTS a nivel físico (Fig. 2).

El terminal móvil dispone de una cierta cantidad de recursos de cómputo para implementar dicha aplicación SDR. Además consideramos que otras capas del protocolo y el transmisor correspondiente se implementan separadamente. El conjunto (*c-load, b-load*) define C_{total} y B_{total} ((9), (10), y Fig. 2), lo cual especifica el factor de escalado S_C y S_B y los recursos computacionales disponibles (Fig. 1). Los diferentes valores para *c-load* y *b-load* pueden ser debido a diferentes capacidades inicial de los terminales móviles o debido a otras tecnologías de acceso radio que pueden residir dentro de los terminales de tipo multimodo.

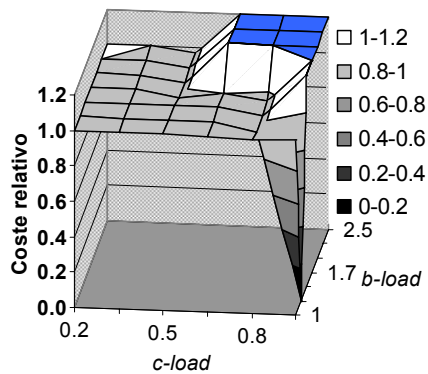


Fig. 3. Resultados.

C. Resultados y discusión

Para cada instancia de $(c\text{-load}, b\text{-load})$, donde $c\text{-load} \in 0.2, 0.35, \dots, 0.95$ y $b\text{-load} \in 1, 1.25, \dots, 2.5$, analizamos la asignación de recursos computacionales según dos algoritmos: el $g\text{-mapping}$ y el $opt\text{-mapping}$. El $opt\text{-mapping}$ examina las 3^{24} posibilidades para encontrar el mapeo de coste mínimo. Relacionamos los resultados del $g\text{-mapping}$ con los del $opt\text{-mapping}$ correspondientes. Más precisamente, calculamos el cociente entre el coste del $opt\text{-mapping}$ y el coste del $g\text{-mapping}$ para cada instancia de $(c\text{-load}, b\text{-load})$.

Una asignación puede ser factible o no factible. Una asignación factible tiene un coste finito, una asignación no factible un coste infinito. Una solución de coste infinito utilizando el $opt\text{-mapping}$ indica un escenario no factible y se caracteriza mediante un coste relativo de 1.2. Una solución no factible del $g\text{-mapping}$ en un escenario factible tiene un coste relativo de 0. La figura 3 ilustra los resultados.

La figura expone que para $(c\text{-load}, b\text{-load}) = (0.95, 1.25)$, a pesar de más de $3 \cdot 10^6$ soluciones factibles, la propuesta de $g\text{-mapping}$ no satisface todas las restricciones del sistema. Este caso particular requiere un algoritmo o método más sofisticado. En 11 casos no existe ninguna solución factible. Para todos los demás casos el algoritmo logra entre 96 y 100% del resultado óptimo. Es decir, excepto a $(c\text{-load}, b\text{-load}) = (0.95, 1.25)$, el método de asignación de recursos computacionales es adecuado para resolver el problema dado.

Los mapeos no factibles indican que serán necesarios más recursos de computación. Mediante la eliminación de funciones o aplicaciones SDR inactivas, por ejemplo, se liberarían recursos modificando $(c\text{-load}, b\text{-load})$ con el fin de conseguir una solución factible.

IV. CONCLUSIONES

Este artículo ha introducido un modelado del sistema SDR como base para resolver el problema de asignación de recursos computacionales en ámbitos de software radio. Los diferentes tipos de matrices y grafos dirigidos describen plataformas y aplicaciones SDR. Además, dos parámetros sencillos capturan la interrelación entre una aplicación SDR y una plataforma SDR. Las simulaciones basadas en un método simple de asignación de recursos computacionales y un estudio de casos prácticos han demostrado la conveniencia del modelado propuesto.

AGRADECIMIENTOS

Este trabajo ha sido financiado por la CYCIT (Consejo de Ciencia Nacional Española) bajo la asignación TIC2003-08609, que parcialmente está financiado por la Comunidad Europea a través del programa FEDER, la DURSI de la Generalitat de Catalunya y los fondos sociales europeos.

REFERENCIAS

- [1] J. Mitola, "The software radio architecture," *IEEE Commun. Mag.*, vol. 33, no. 5, pp. 26–38, May 1995.
- [2] E. Buracchini, "The software radio concept," *IEEE Commun. Mag.*, vol. 38, iss. 9, pp. 138–143, Sept. 2000.
- [3] S.-Y. Lee, J. K. Aggarwal, "A mapping strategy for parallel processing," *IEEE Trans. Comput.*, vol. C-36, no. 4, pp. 433–442, April 1987.
- [4] V. Chaudhary, J. K. Aggarwal, "A generalized scheme for mapping parallel algorithms," *IEEE Trans. on Parallel Distrib. Syst.*, vol. 4, iss. 3, pp. 328–346, March 1993.
- [5] A. H. Alhusaini, V. K. Prasanna, C. S. Raghavendra, "A framework for mapping with resource co-allocation in heterogeneous computing systems," *Proc. 9th IEEE Heterogeneous Computing Workshop (HCW 2000)*, Cancun, Mexico, May 2000, pp. 273–286.
- [6] H. Topcuoglu, S. Hariri, Min-You Wou, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE Trans. on Parallel and Dist. Syst.*, vol. 13, iss. 3, pp. 260–274, March 2002.
- [7] R. Bajaj, D. P. Agrawal, "Improving scheduling of tasks in a heterogeneous environment," *IEEE Trans. Parallel Distr. Syst.*, vol. 15, no. 2, Feb. 2004.
- [8] A.-R. Rhiemeier, F. Jondral, "Mathematical modeling of the software radio design problem," *IEICE Trans. Commun.*, vol. E86-B, no. 12, Dec. 2003.
- [9] V. Marojevic, X. Revés, A. Gelonch, "Computing resource management for SDR platforms," *Proc. 16th IEEE Int'l Symp. Personal, Indoor and Mobile Radio Communications (PIMRC 2005)*, Berlin, Sept. 11–14, 2005.
- [10] D. F. Robinson, L. R. Foulds, *Digraphs: Theory and Techniques*. Gordon and Breach Science Publisher Inc., 1980.
- [11] Technical Specification Group Radio Access Network (3GPP), "TS 25.212 V6.4.0 – multiplexing and channel coding (FDD)," March 2005, www.3gpp.org
- [12] T. Faber, M. Schönle, "DSP-platform target report," SLATS Consortium, Project no. 27016, Deliverable D23, Dec. 1999.