

Descripción Sintetizable de Alto Nivel para Sistemas Digitales Asíncronos con Redes de Petri

Luis Felipe Plata Rueda
(luis.plata@umb.edu.co)

Resumen

Este documento presenta una forma de realizar la detección de finalización de procesos, para sistemas digitales asíncronos. Para ello se muestran resultados del estudio del protocolo de dos fases "dual rail". También se detalla alguna información de diferentes casos sintetizados, con una descripción VHDL, en una FPGA (Field Programmable Gate Array). En dichas arquitecturas, se emplean redes de Petri, como sistema de control para un circuito asíncrono. Finalmente se analizan los modelos de sistemas asíncronos, para escoger el más idóneo en el momento de sincronizar el proceso con la red de control (red de Petri).

Abstract

This paper shows a method for detecting completion of processes in asynchronous digital systems. To do that, it is show results about studies of dual rail protocol. It is also present some information on different cases synthesized by a VHDL description on a FPGA (Field Programmable Gate Array). In this architecture, Petri nets are used as a control system for an asynchronous circuit. Finally it is analyzed the asynchronous system models, to choose the best architecture at the moment to synchronize the processes with the control network (Petri net).

Keywords:

FPGA, Red de Petri, VHDL, Asíncrono.

1. Introducción

Este documento presenta la continuación de una investigación exhaustiva [17] en donde se pone en evidencia resultados de simulación y síntesis de sistemas digitales asíncronos. De esta manera, se pretende exponer las ventajas que presenta la implementación de sistemas electrónicos que funcionan sin un reloj global, como lo es su alto desempeño. Se estudiaron las diferentes arquitecturas de circuitos asíncronos y se aplicó el esquema "Delay Insensitive" [3] para sintetizar sistemas digitales en una FPGA (Field Programmable Gate Array). Para lograr la síntesis, se analizan los protocolos de comunicaciones para sistemas asíncronos, empleando una Red de Petri (PN) [1] como sistema de control para el proceso.

CONGRESO

INTERNACIONAL DE

COMPUTACIÓN Y

TELECOMUNICACIONES

COMTEL 2009

La descripción Hardware sintetizable que se realizó, se basó en el análisis de cada una de las condiciones que operación del circuito de una red de Petri cualquiera. Por otro lado, el algoritmo se diseñó para que fuera independiente de la tecnología empleada y de la arquitectura de las redes. Para ello, se definen los componentes de una red de Petri, en donde, un "Lugar" asíncrono se compone de un latch SR, como elemento de almacenamiento, y una lógica combinatoria que establece el marcado del componente, según la secuencia de la red. También se definió una "Transición" como una compuerta AND en donde sus entradas se encuentran relacionadas con una condición externa y las señales de activación de cada uno de los lugares predecesores a dicha Transición.

En un principio, se exponen las principales ventajas del diseño asíncrono, como motivación para el estudio de la comunicación de este tipo de sistemas. Se expone el modelo de comunicación asíncrona "delay insensitive", como el esquema más realístico para el diseño de circuitos sin reloj global. Posteriormente, se presentan las redes de Petri, como el mecanismo de control para los sistemas mencionados. Finalmente, se explican los protocolos de comunicación asíncrona, para entender el sistema de detección "dual rail", que servirá como medio para establecer la finalización de los procesos asíncronos. Se expone la aplicación de un caso, en donde se emplea HiLeS [7], para obtener el algoritmo que se sintetiza en una FPGA.

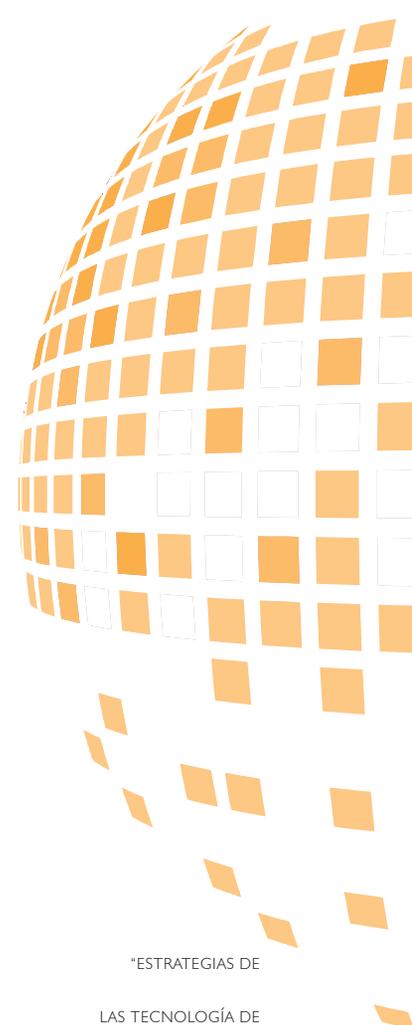
Básicamente se quiere mostrar las principales características de la comunicación asíncrona por medio del protocolo de dos fases. Como marco tecnológico se emplean FPGA de XILINX [4] en donde se sintetiza la lógica asíncrona, se realiza una simulación funcional y posteriormente una simulación temporal con la herramienta Model Sim.

2. Organización del tiempo en Hardware y ventajas del diseño asíncrono

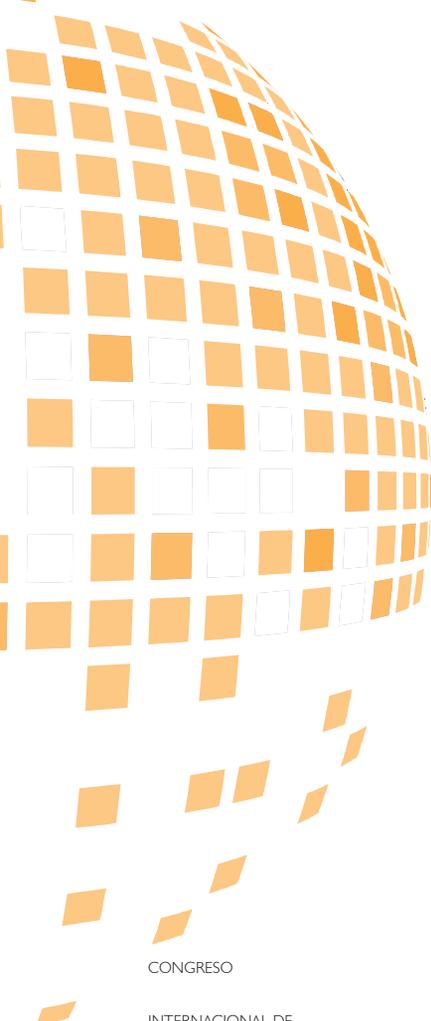
La sincronización es uno de los problemas más importantes para los sistemas digitales. Este término se puede definir como la coordinación de eventos (señales, operaciones o procesos) en tiempo. Al mismo tiempo, se podría decir que asincronía es una relación entre los pasos que hay desde un dato inicial hasta obtener el resultado de un proceso. Esta relación de causa-efecto debe tener un protocolo que rija la comunicación de información.

Los circuitos asíncronos son circuitos digitales que reaccionan a los cambios en sus señales de entrada de acuerdo a la funcionalidad de las compuertas del circuito [15]. En general, cualquier interconexión arbitraria de compuertas es considerada un circuito asíncrono. La complejidad de esto, consiste en generar una interconexión apropiada de compuertas que lleve a una correcta interacción con el entorno de acuerdo con un protocolo específico.

El consumo de energía en los circuitos síncronos es el segundo y más importante problema. Todos los ciclos de reloj son cargados y descargados todo el tiempo. La señal de reloj es distribuida a todos los elementos de



"ESTRATEGIAS DE
LAS TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL"



CONGRESO

INTERNACIONAL DE

COMPUTACIÓN Y

TELECOMUNICACIONES

COMTEL 2009

memoria, inclusive a muchas partes del circuito que son funcionalmente inoficiosas. Es bastante la energía perdida debido a la conmutación inútil de la señal.

Este nivel de consumo es uno de los parámetros críticos en el momento de realizar un diseño eficiente, principalmente en equipos portátiles y en general en los equipos electrónicos que requieren una duración prolongada de la batería y baja disipación de calor. El calor también trae como consecuencia, el deterioro de los chips, obligando a adquirir encapsulados más costosos de cerámica y sistemas de ventilación para el control de la temperatura.

El diseño asíncrono puede ser una alternativa a los múltiples problemas que se generan con los sistemas síncronos. Al no tener un reloj global, estos sistemas reducen el consumo de potencia que puede darse al incrementar la frecuencia en los sistemas síncronos. Por otro lado, al diseñar asincrónicamente se puede evitar restricciones de diseño como lo es la distribución de las señales de reloj en sistemas síncronos complejos.

3. Modelo "Delay Insensitive"

La verificación formal de un sistema requiere que dicho sistema sea descrito por medio de un modelo formal. En este trabajo, el lenguaje de modelamiento considerado, son las redes de Petri.

Múltiples diseños de sistemas digitales son modelados mediante el uso de diferentes herramientas que permiten un análisis complejo del comportamiento funcional y temporal. Además de ello, existen diversas tecnologías de implementación de hardware que facilitan la elaboración de nuevos diseños tecnológicos.

Existen diversos estilos de diseño asíncronos los cuales se establecen por los modelos que emplean [11]. Estas basan sus especificaciones en asunciones de retardos en compuertas y/o cables de interconexión. (Fig. 1).

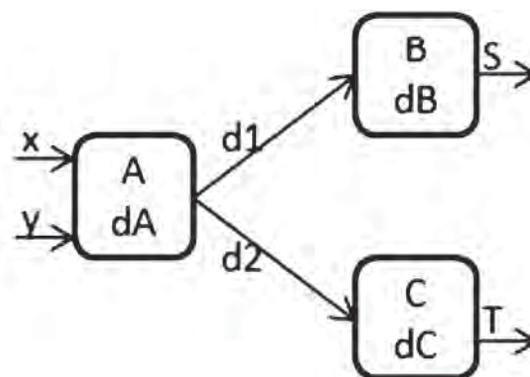


Fig. 1. Diagrama en bloques de un sistema asíncrono.

Para este trabajo, se tomó como base el modelo “delay insensitive”, dadas sus características y aproximación con la realidad en síntesis.

Un circuito “delay insensitive” debe funcionar correctamente según su especificación cuando es modelado con un retardo en las compuertas (dA, dB, dC), como en los cables (d1, d2), asumido positivo, acotado pero desconocido. Por lo tanto, el receptor de un dato comunica al emisor que se ha recibido la información. Esta comunicación se realiza mediante un circuito de detección de finalización que se ubica en el receptor. Al recibir el emisor esta señal por parte del transmisor, es posible hacer la transferencia de un nuevo dato, de lo contrario, el transmisor no realiza ninguna operación.

Este modelo, idealmente permitiría la más robusta implementación en circuitos asíncronos, ya que ninguna hipótesis es realizada sobre la magnitud y distribución de los retardos.

4. Redes de Petri (Descripción Hardware del Control)

Las redes de Petri (Petri Nets) tienen su origen en 1962 cuando C.A. Petri estableció por primera vez su concepto para representar sistemas complejos mediante un modelo simple [15].

Una red de Petri (PN) es una tupla de cinco elementos, $PN = (P, T, F, W, MO)$ donde:

1. $P = \{p_1, p_2, \dots, p_m\}$ es un conjunto finito de lugares.
2. $T = \{t_1, t_2, \dots, t_n\}$ es un conjunto finito de transiciones
3. $F \subseteq (P \times T) \cup (T \times P)$ es un conjunto de arcos.
4. $W : F \rightarrow \{1, 2, 3, \dots\}$ es una función de peso
5. $MO : P \rightarrow \{0, 1, 2, 3, \dots\}$ es el marcado inicial

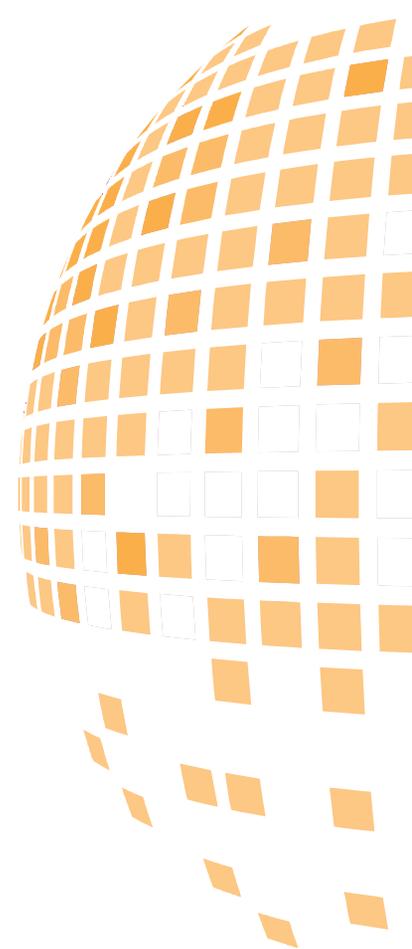
Un grafo red de Petri tiene dos tipos de nodos (lugares y transiciones) y arcos que los conectan. Esto significa que es un grafo bipartito en donde los arcos no pueden conectar directamente nodos del mismo tipo. Es decir, los arcos conectan nodos de lugares a nodos de transiciones y nodos de transiciones a nodos de lugares.

Los lugares pueden contener un conjunto de fichas (tokens), representadas por un punto negro, que indican que el lugar se encuentra activo. La estructura de una red de Petri puede ejecutarse mediante las siguientes reglas:

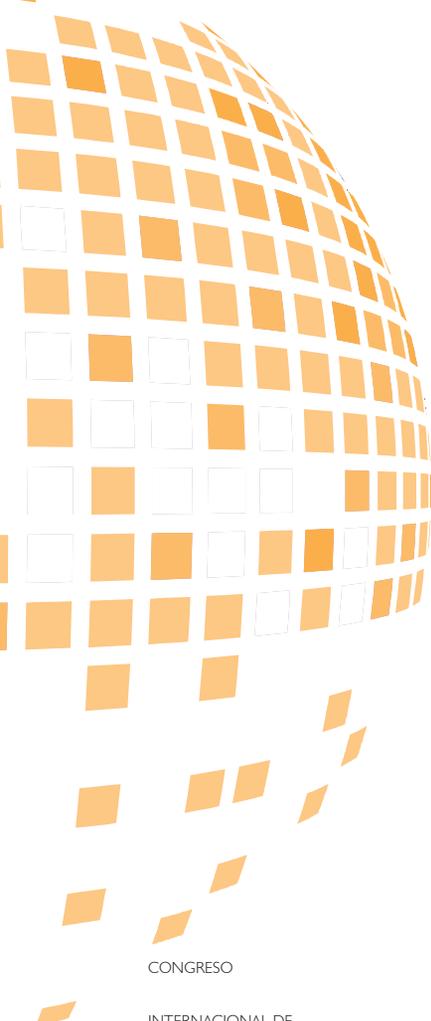
Una transición es habilitada si y solo si existe un token en cada lugar antecesor.

Cuando se dispara una transición, se remueve un token de cada lugar de entrada y se coloca un token en cada lugar de salida a la transición.

Posterior al disparo de la transición, se debe evaluar nuevamente la red.



“ESTRATEGIAS DE
LAS TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL”



CONGRESO

INTERNACIONAL DE

COMPUTACIÓN Y

TELECOMUNICACIONES

COMTEL 2009

Existen varios documentos en donde se da la representación combinacional de una red de Petri asíncrona. Básicamente, siguiendo la definición de un grafo de red de Petri y observando su comportamiento, se podría decir que dichas redes se basan generalmente de un elemento de almacenamiento (flip-flop ó latch) y una lógica combinacional que traducen los lugares y las transiciones en una red.

Un lugar de una red de Petri cualquiera, es un flip-flop (si es una red síncrona) o un latch SR (si es una red asíncrona), que coloca o retira un "token" dependiendo de la activación o no de las transiciones anteriores o posteriores a ese lugar. Una transición se puede resumir en una compuerta lógica "AND", en cuyas entradas se reúnen todas las condiciones necesarias para adicionar un "token" al lugar posterior a la transición, y retirar un "token" al lugar anterior a la transición. En la Fig. 2 se puede observar el circuito combinacional que representa una red de Petri asíncrona, en donde la señal "init" representa la inicialización de la red, estableciendo los marcados iniciales en los lugares correspondientes. C_T_1, C_T_2 y C_T_3 corresponden a las condiciones externas que habilitan la continuidad del proceso. Así mismo, se observa que la salida de la transición es la señal que activa el "token" en el siguiente lugar y al mismo tiempo lo desactiva del lugar anterior. Como elementos de almacenamiento, se emplean latch SR.

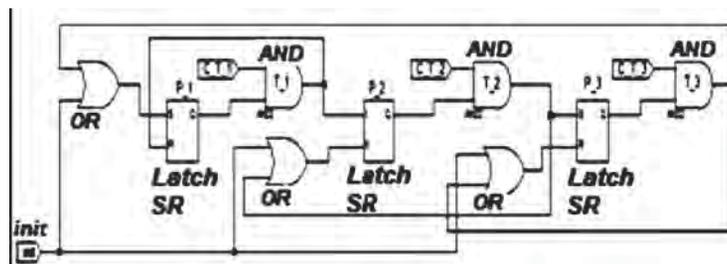


Fig. 2. Lógica de una red de Petri asíncrona cíclica de tres lugares y tres transiciones [17]

En conclusión, una forma práctica de describir una red de Petri en VHDL, sería establecer dos tipos de componentes (lugares y transiciones) y realizar una descripción general de la red, en donde se establezcan las relaciones entre cada uno de los componentes que allí se encuentran.

5. Comunicación de Sistemas Asíncronos

Un elemento fundamental dentro de la arquitectura de un sistema asíncrono es la comunicación entre datos y control de las distintas partes del circuito. Al no existir un reloj global que controle el flujo de datos de un bloque a otro, nace el concepto de "handshake", el cual es un mecanismo para sincronizar dos unidades de un sistema asíncrono [2]. Una unidad se encuentra activa e inicia el "handshake" emitiendo una señal de "request" a la otra unidad. Esta segunda unidad es pasiva y recibe la señal de petición, respondiendo (cuando la unidad se encuentre lista) con una señal de "acknowledgement". Un "handshake" el cual consiste en el intercambio de solo dos tokens (un "request", un "acknowledge") es la forma más simple de sincronización.

La estructura circuital básica para el manejo de "handshakes" se basa en elementos combinacionales, tales como compuertas "AND", "OR" y "NOT", y elementos de almacenamiento activos por nivel como los "latches".

Los circuitos asíncronos realizan la comunicación entre bloques por medio de dos protocolos fundamentales (Fig. 3): Dos Fases y Cuatro Fases<cite>ortega</cite>, que a su vez se extienden dependiendo del nivel de detección de cada señal.

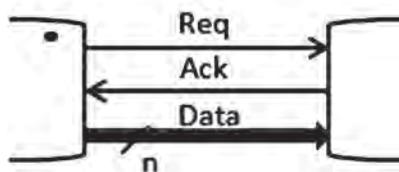


Fig. 3. Comunicación Asíncrona

A continuación se presentan algunas características de los protocolos de comunicación para sistemas asíncronos:

Protocolo Bundled-data (4 Fases Single rail)

El término "Bundled-data" se refiere a que los datos en las señales usan niveles Booleanos para codificar la información. Por otro lado, cables separados (Request, Acknowledge), son relacionados con las señales de datos.

El término Cuatro Fases, se refiere al número de acciones para realizar una comunicación. Este protocolo describe su funcionamiento de la siguiente forma (Fig. 4):

- El emisor establece un dato a la entrada de una transformación y envía una señal de petición en alto al receptor.
- El receptor inicia la transformación de esos datos y envía una señal de reconocimiento en alto al emisor.
- El emisor responde estableciendo la señal de petición en bajo.
- El receptor reconoce este cambio colocando la señal de reconocimiento en bajo.

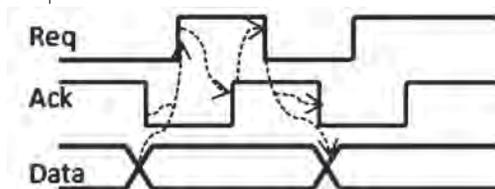
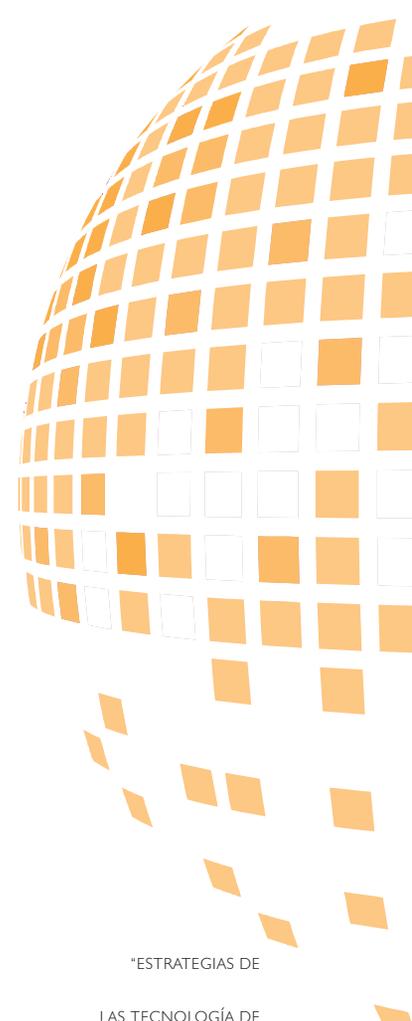


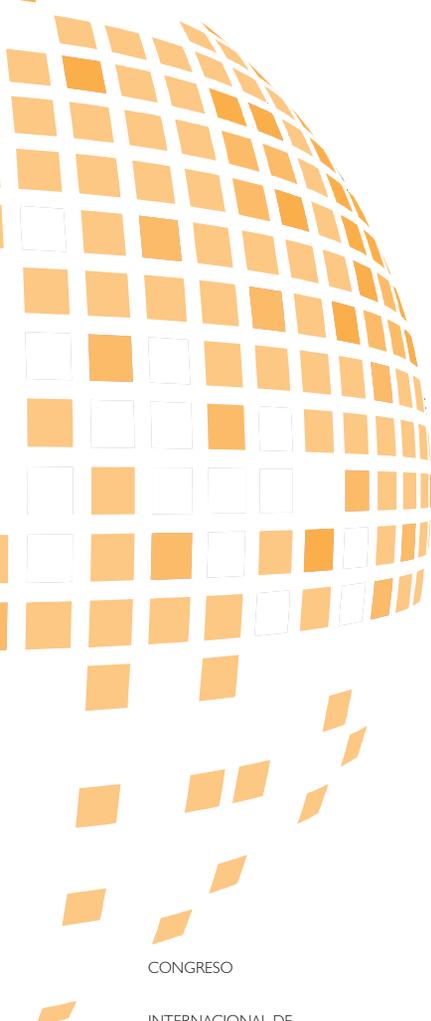
Fig. 4. Señalización 4 Fases (Single rail)

Protocolo Bundled-data (2 Fases Single rail)

Este protocolo también emplea las señales de "Request" y "Acknowledge" para realizar transferencias de información con la diferencia que en cada transición se produce un evento. Es decir, no existe diferencia entre transiciones de 0 a 1 y de 1 a 0, ya que ambas producen una señal de evento.



"ESTRATEGIAS DE
LA TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL"



CONGRESO

INTERNACIONAL DE

COMPUTACIÓN Y

TELECOMUNICACIONES

COMTEL 2009

Este protocolo describe su funcionamiento de la siguiente forma (Fig. 5):

- El emisor establece un dato a la entrada de una transformación y envía una señal de petición al receptor.
- El receptor inicia la transformación y cuando éste finaliza, envía una señal de reconocimiento al emisor.

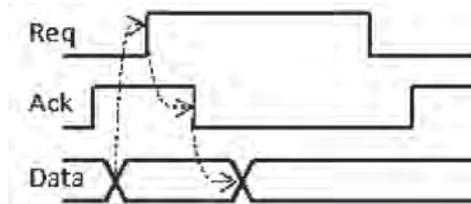


Fig. 5. Señalización 2 Fases (Single rail)

Protocolo de 4 Fases Dual rail

Este protocolo codifica la señal "Request" en las señales de datos, usando dos cables por bit de información que va a ser comunicada (Fig. 6).

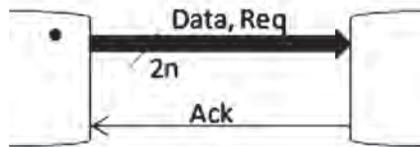
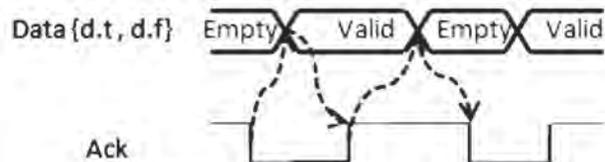


Fig. 6. Comunicación Dual rail

El par de cables $\{x.f, x.t\}$ representa un código de palabra, en donde:

- $\{x.f, x.t\} = \{1,0\}$ y $\{x.f, x.t\} = \{0,1\}$ representan un dato válido ("0" lógico y "1" lógico respectivamente).
- $\{x.f, x.t\} = \{0,0\}$ representa "no data" ó "empty value".
- $\{x.f, x.t\} = \{1,1\}$ no es usado.



	d.t.	d.f.
Empty ("E")	0	0
Valid "0"	0	1
Valid "1"	1	0
Not used	1	1

Fig. 7 Comunicación 4 Fases Dual rail

Este protocolo describe su funcionamiento de la siguiente forma (Fig. 7):

1. El emisor coloca un "codeword" válido.
2. El receptor toma el "codeword" y activa la señal de "acknowledge" en alto.
3. El emisor responde colocando un "empty value".
4. El receptor reconoce esto, colocando la señal de "acknowledge" en bajo.

Protocolo de 2 Fases Dual rail

Este protocolo también emplea dos cables por bit {d.t , d.f} (Fig. 8), pero la información es codificada como transiciones.

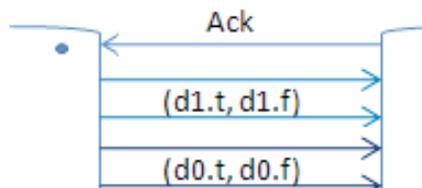


Fig. 8. Protocolo de 2 Fases Dual rail

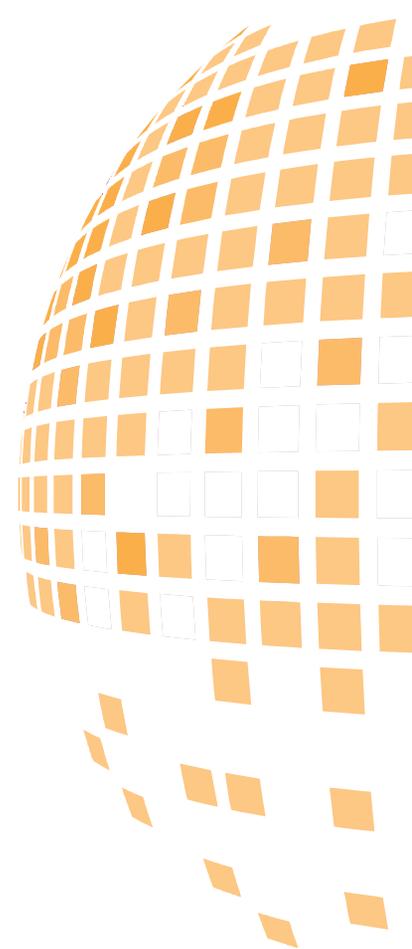
El valor "empty" (00) no existe, por lo tanto, el receptor envía una señal de "acknowledge" únicamente cuando recibe un código válido por parte del emisor.

El protocolo de dos fases, idealmente generaría problemas en la activación del latch, ya que el protocolo se basa en la generación de eventos en cada transición de sus señales, y el elemento de almacenamiento se activa con un solo tipo de transición. Pero en una red de Petri, el token pasa de un lugar a otro. Dado a estas características es la más aplicable por las razones anteriores.

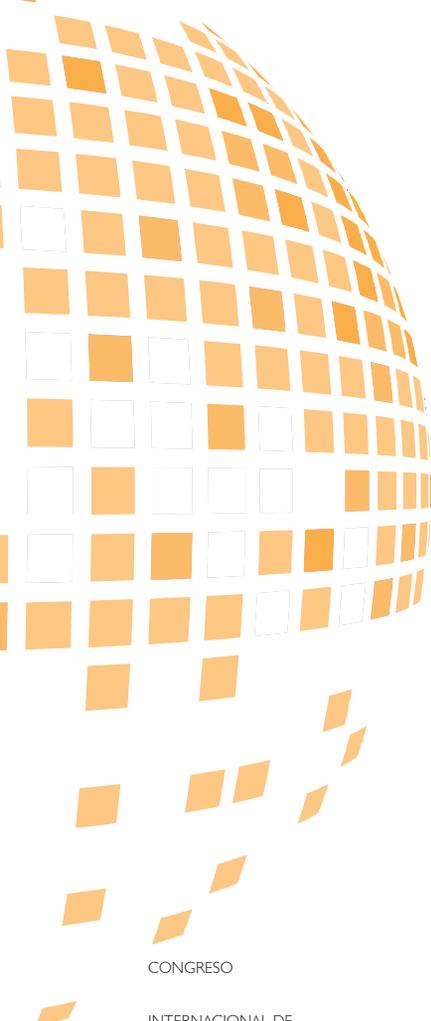
Este protocolo es muy eficiente, ya que la información se puede comunicar sin importar el retardo en el sistema, por eso es comúnmente llamado "delay insensitive".

6. Detección de finalización de procesos asíncronos (Dual rail)

Anteriormente se mencionó que las redes de Petri serían el módulo de control para un sistema asíncrono. Por otro lado, se debe realizar una sincronización entre el bloque que realiza el proceso asincrónicamente y el control. Para ello, es necesario determinar el momento exacto en que finaliza el proceso. En un sistema síncrono, se determina el retardo más largo de todos los procesos, y con base en ello, se establece un periodo de reloj suficientemente largo, el cual dará un pulso cuando el sistema se encuentre estable. Dado un sistema asíncrono, se debe realizar un sistema de detección de finalización de procesos, ya que no existe un reloj global. Esta detección, después de un estudio, se realiza mediante señales de "acknowledge" [3], las cuales le indicarán a las "Transiciones" de la red de petri, el momento en que finaliza el proceso y por lo tanto, el sistema puede



"ESTRATEGIAS DE
LAS TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL"



CONGRESO

INTERNACIONAL DE

COMPUTACIÓN Y

TELECOMUNICACIONES

COMTEL 2009

continuar. Mediante el empleo del concepto "dual rail", se logran establecer las señales del "Acknowledge", en donde se emplean dos cables por cada bit de información y mediante una compuerta "XOR" y las reglas empleadas en la Fig. 7, enviará o no, una señal que active o no la "Transición" que dará continuación al proceso. Es decir, únicamente existirá un dato válido cuando $d.t = 0$ y $d.f=1$ ($d.t$ ="data true", $d.f$ ="data false"), o también será un dato válido para el caso $d.t=1$ y $d.f=0$.

En este punto, ya se puede realizar aplicaciones de síntesis para sistemas digitales asíncronos.

7. Aplicación: $Y=(X1+X2)(X3+X4)(X5+X6)$

Para validar el proceso de diseño y síntesis, se realizaron pruebas a diferentes casos. En primer lugar se observaron las propiedades de diferentes multiplicadores asíncronos y síncronos de 32 bits [17], como lo son la velocidad y el espacio ocupado en un dispositivo. En dichas pruebas, se comienza a identificar las ventajas de velocidad de los sistemas asíncronos.

La aplicación realizada en este documento, coloca en evidencia una de las ventajas de las redes de Petri, la cual es los procesos paralelos. En este caso, se realizan simultáneamente las operaciones $(X1+X2)$, $(X3+X4)$ y $(X5+X6)$, cuyas entradas son vectores de 32 bits cada uno (Fig. 9). Las salidas de dichas operaciones, son las entradas de un multiplicador de Booth [6], que según la bibliografía estudiada, es uno de los multiplicadores más rápidos del momento. Finalmente, el vector "Y", corresponde al resultado de la operación. Los "Lugares" PI_1 , PI_3 , PI_5 y PI_6 , son "testigos" de las operaciones que ocurren en los bloques de funciones. El diagrama de la Fig. 9 fue realizado en el formalismo HileS Designer [7] y la tabla de la Fig. 10 muestra las características de síntesis del sistema.

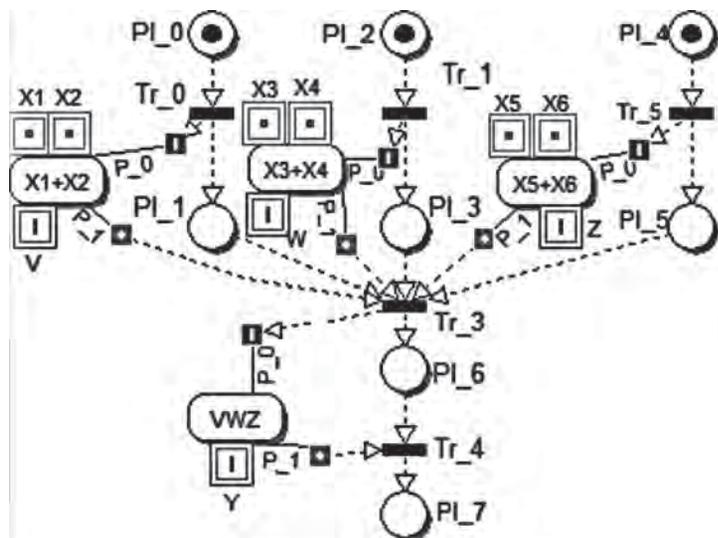


Fig. 9. Caso $Y=(X1+X2)*(X3+X4)*(X5+X6)$ desarrollado en HileS [7]

	Asíncrono	Síncrono
Slices	2321	2280
LUTS	4572	4295
Retardo (ns)	85,2	148,2ns

Fig. 10. Resultados de la aplicación desarrollada en este documento.

Como se observa en la Fig. 10, al no existir un reloj global en un sistema asíncrono, el espacio ocupado se incrementa en relación con su contraparte síncrona. También se puede observar las ventajas de velocidad del sistema asincrónico, quien ejecuta la operación (Fig. 9) en 85,2ns; el cual es casi la mitad del tiempo que demora en realizar el mismo proceso, pero sincrónicamente (148,2ns). Se observa una diferencia de 63ns, dando una ventaja en velocidad a los sistemas que no manejan un reloj global.

8. Conclusiones

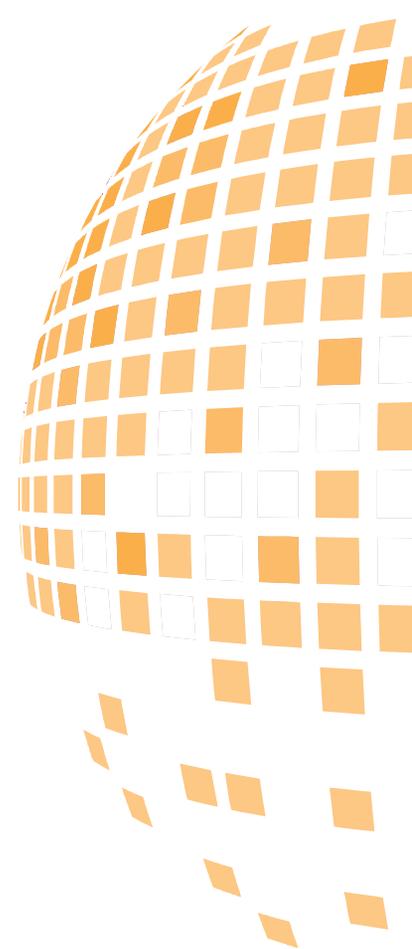
El concepto de "dual rail" facilitó la realización de un modelo de detección de finalización de procesos asíncronos. Esta etapa del diseño tiene gran importancia, al no existir un reloj global, que determine el tiempo en que un proceso ha terminado. Con este mecanismo se sabe con certeza, el momento en que son válidos los datos de salida de un proceso digital asíncrono. Para ello, el sistema emplea dos cables por bit de operación (Fig. 8).

Mediante el uso de redes de Petri, se realiza el control del sistema, sincronizando de una manera idónea cada una de las señales y procesos del mismo. Se realiza la síntesis del sistema en una FPGA [4], comenzando por la elaboración de una red de Petri como el mecanismo de control. Esto se valida mediante la realización de pruebas a nivel de simulación e implementación (Spartan-3E de Xilinx).

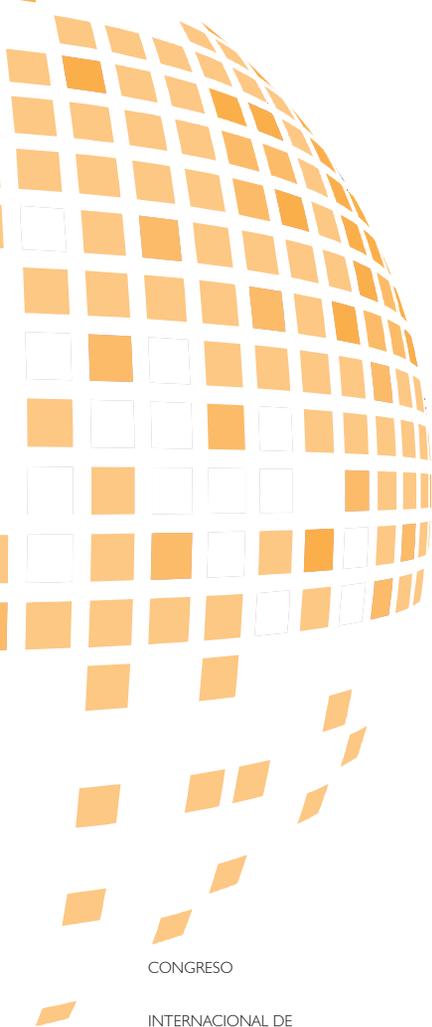
La finalidad de esta investigación, no es pretender afirmar que los sistemas asíncronos son mejores que los síncronos. Se propone investigar e implementar sistemas combinados, como lo son los GALS (Globally Asynchronous Locally Synchronous) [11].

9. Referencias

- [1] Cassandras, C. G., & Lafortune, S., Introduction to Discrete Event Systems, Libro, Boston: 2001.
- [2] Bardsley, A., Implementing Balsa Handshake Circuits. Manchester: 2000.
- [3] J. Sparso, Asynchronous circuit design, a Tutorial, Technical University of Denmark: 2006.



"ESTRATEGIAS DE
LAS TECNOLOGÍA DE
LA INFORMACIÓN Y
COMUNICACIÓN EN
EL CONTEXTO DE LA
CRISIS MUNDIAL"



CONGRESO

INTERNACIONAL DE

COMPUTACIÓN Y

TELECOMUNICACIONES

COMTEL 2009

- [4] Xilinx, Spartan-3E FPGA Family: Complete Data Sheet. www.xilinx.com : 2008.
- [5] Ortega, S. Sincronización Self-Timed: Protocolo de 4 Fases. Mexico: 2006
- [6] A. D. Booth, Electronic Digital Calculators, Society at the Science Museum, 1991.
- [7] HileS Designer, Tool Manual, LAAS-CNRS, Universidad de los Andes.
- [8] W. Vogler and R. Wollowski, Decomposition in Asynchronous Circuit Design, Augsburg, 2002.
- [9] Poliakov, I., Mokhov, A., & Sokolov, D. (2008). Automated Verification of asynchronous circuits using circuit Petri nets. Newcastle: Asynchronous Systems Group.
- [10] F. Jimenez, Specification et Conception de Micro-systemes bases sur des Circuits Asynchrones, PhD Thesis, Institut National des Sciences Appliquees, LAAS-CNRS, Toulouse, France: 2000.
- [11] V. Varshavsky, GALA (Globally Asynchronous - Locally Arbitrary) Design, Neural Network Technologies Ltda. Israel: 2002.
- [12] Huffman, D. A. The synthesis of sequential switching circuits. J. Franklin Institute: 1954.
- [13] Muller, D. E., & Bartky, W. C., A theory of asynchronous circuits. In Annals of Computing Laboratory of Harvard University: 1959.
- [14] G. Birtwistl and A. Davis, Asynchronous Digital Circuit Design, British Computer Society, 1995.
- [15] Brzozowski, J., & Seger, C., Asynchronous Circuits, Libro, Springer_Verlag: 1995.
- [16] Unger, S. H. (1969). Asynchronous Sequential Switching Circuits. Wiley Interscience.
- [17] Fernando Jiménez, Carlos Villarraga, Luis Plata. Síntesis de Sistemas Digitales modelado con redes de Petri a través del formalismo Hiles Designer. ISBN 978 - 987 - 9486 - 10 - 8. Colombia, 2009