



Controlador empotrado para PBX de ocho abonados

Jaime Fontanella Lauzardo

Víctor Marín Contreras

René Yañez de la Rivera

RESUMEN / ABSTRACT

Como parte del Proyecto Ramal del MIC “Plataforma de Conmutación de Paquetes”, previsto para el desarrollo de productos y tecnologías de origen cubano en el campo de la telefonía, se ha trabajado en el desarrollo de una central telefónica privada (PBX) basada en el diseño digital con lógica programable. En el presente artículo se propone el diseño de un controlador de llamada empotrado para la atención de ocho abonados que constituye uno de los bloques fundamentales de dicha PBX. El controlador, desarrollado en VHDL y empotrado en un FPGA Cyclone II de Altera, permite a través de dos tarjetas de abonado previamente desarrolladas, generar toda la señalización de abonado y establecer hasta cuatro conversaciones simultáneas.

Palabras claves: **Conmutación digital, PBX, VHDL**

Embedded Controller for eight subscribers PBX

As part of the Project “Digital Switching Platform”, forecasted for the development of products and technologies of Cuban origin in the telephony field, some work has been done for developing a Private Branch Exchange (PBX) based on logic programmable digital design. The present article proposes an embedded controller design in charge of eight subscribers. This block is one of the main components of the PBX. The controller developed on VHDL and embedded into an Altera’s Cyclone II FPGA, allow us to generate all the subscribers signaling and establish up to four conversations simultaneously. The controller carries out its function by means of two previously designed subscribers’ boards.

Key words: Digital Switching, PBX, VHDL

Introducción

El presente trabajo forma parte del proyecto ramal del MIC “Plataforma de Conmutación de Paquetes” previsto para el desarrollo de productos y tecnologías de origen cubano en el campo de la telefonía. Dicho proyecto tiene como propósito el desarrollo de un **Gateway** de acceso de Voz sobre IP (**VoIP**) para soluciones de digitalización con redes de nueva generación (NGN). El diseño que se propone en este artículo se basa en la experiencia acumulada por el grupo de trabajo en la implementación de una central telefónica privada (**PBX**, Private Branch Exchange) para cuatro abonados¹, en las tarjetas de abonados que con ese fin fueron desarrolladas y se inserta a su vez en una propuesta más integradora para el desarrollo final del **Gateway** de **VoIP** basado en un sistema basado en el microprocesador (softcore) **Nios II** de **Altera Corporation**.

Se persigue, además, la fabricación de un producto que sea capaz de funcionar con independencia de la PC y que garantice soberanía tecnológica. Es por ello que resulta muy factible para implementar el diseño la utilización de dispositivos lógicos programables. En este caso, se recurre a la tarjeta **Nios II Development Kit** de **Altera Corporation**².

Los objetivos de este trabajo son: expandir las potencialidades del diseño existente elevando la capacidad de la **PBX** hasta ocho abonados. Adaptar la trama E1 con que trabaja la central al estándar establecido en la recomendación **G.732** de la Unión Internacional de Telecomunicaciones (**UIT**)³. Lograr que la central atienda tantas llamadas de forma simultanea como lo permita el hardware existente. Emplear el **FPGA Cyclone II** del **Nios II Development Kit** de **Altera** a fin de trabajar con el mismo dispositivo donde luego se pretende empotrar la aplicación final del **Gateway** (este diseño sería uno de los periféricos del **Nios II**).

El controlador diseñado está compuesto por dos bloques básicos que se conectan en el fichero top level del diseño según se aprecia en la figura 1. El primer bloque (**divider**) que aparece a la izquierda se adaptó del diseño anterior para dotarlo de las potencialidades necesarias para el nuevo diseño. El segundo bloque (**control_llamada**) fue implementado completamente en **VHDL** y constituye el componente fundamental del controlador. Este controlador se programa en el **FPGA (Cyclone II)** y se conecta a través de los conectores **J11** y **J16** del kit de desarrollo al conector **J3** de cada una de las dos tarjetas de cuatro abonados que conforman la central⁴.

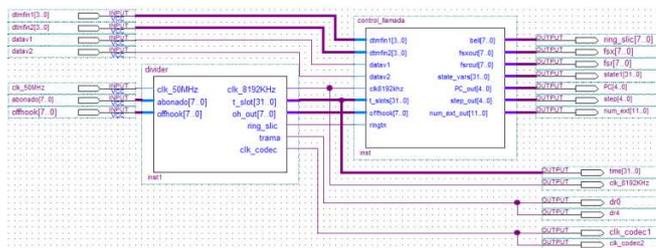


Figura 1: Diagrama en bloques del controlador empotrado.

Tarjeta de abonados

En la figura 2 se muestra una vista superior de la tarjeta de abonados. En ella se pueden apreciar los cuatro canales que la conforman. Un detalle interesante es que aparece un solo **DTMF** mientras que el resto de los componentes (**Relevador de Timbre, SLIC y CODEC**) están repetidos y ubicados en cada uno de los canales. Esto se debe a un análisis de costo y rendimiento cuyo resultado condujo a compartir el **DTMF** entre los cuatro canales¹.

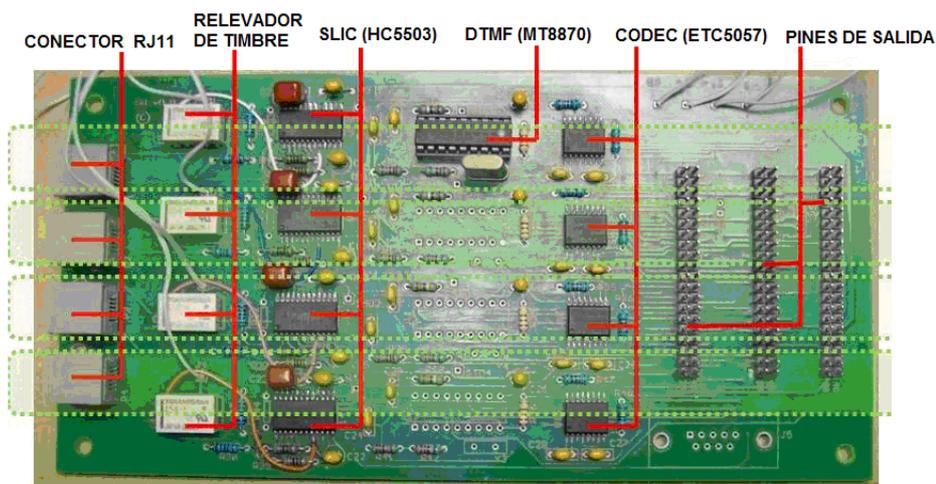


Figura 2: Tarjeta de abonados.

En el **SLIC** (circuito integrado **HC5503**), la señal de salida **SHD** y de entrada **RC** (activas a nivel bajo) indican en el caso de **SHD** si el abonado se encuentra colgado o descolgado, mientras que por **RC** se recibe el comando de timbre cuando el abonado en cuestión está siendo llamado⁵. El **DTMF** (circuito integrado **MT8870**) entrega en la salida **StD** un uno lógico, cada vez que detecta y valida que se ha oprimido un dígito del número llamado, colocando en las salidas **Q1** a **Q4** los cuatro bits correspondientes al código binario del dígito⁶. Por último el **CODEC** (circuito integrado **ETC5057**), tiene dos entradas **FSX** y **FSR** de habilitación de transmisión y recepción respectivamente; la entrada **DR** que recibe los datos de voz codificada procedentes de un abonado remoto, mientras que la salida **DX** transmite la señal de voz, también codificada, procedente del abonado que pertenece al canal⁷.

Bloque Divider

Este bloque es el encargado de entregar las diferentes señales de reloj que necesita el sistema, las señales que permiten generar los tonos de señalización de la central (tono de discar, ocupado, timbre y alarma), eliminar el efecto de rebote del **offhook** proveniente de la **SLIC**, conformar la trama de 32 canales³ y proveer al controlador de las señales, que estando en fase con los canales de abonado y servicio, permiten establecer adecuadamente la comunicación y hacer llegar a los abonados los diferentes tonos de señalización. Este bloque está compuesto a su vez por tres bloques fundamentales que le permiten cumplir con las funcionalidades antes descritas: **freq_divider**, **rebote** y **timeslot**.

Bloque Timeslot

El bloque **timeslot** es un registro de desplazamiento de 32 bits que produce a partir de una frecuencia de reloj de 256 kHz, las 32 señales de muestreo de 8KHz que el sistema utiliza para cada canal de la trama. Las señales serán por tanto de igual frecuencia y ciclo útil y solo difieren en el desfase relativo necesario para multiplexar en tiempo los 32 canales de trabajo y formar una trama periódica con las muestras de voz de los ocho abonados y de las cinco señales de servicio. Los canales de abonado fueron colocados consecutivamente a partir del canal 1 en la parte baja de la trama. El canal 0 se deja libre para funciones de señalización en un posible sistema multitrama a implementarse en futuros diseños. Los canales de servicio donde se colocan las señales de tono de discar, ocupado, timbre y alarmas fueron colocados en la parte alta de la trama a partir del canal 27.

Aunque en el diseño actual solo se utilizan 13 canales se decidió adaptar este bloque a exigencias futuras del proyecto donde serán necesarios los canales restantes hasta completar 32 como lo estipula la recomendación **G.732**. El bloque fue implementando expandiendo el registro de desplazamiento de 8 bits proporcionado como megafunción de Altera (74164) utilizado en diseños previos del grupo hasta alcanzar los 32 bits necesarios en esta etapa de desarrollo.

Bloque Rebote

Este bloque permite eliminar el rebote mecánico que se produce en el teléfono al colgar o descolgar. La señal **offhook** (activa a nivel bajo) debe mantener su valor al menos por tres ciclos de reloj (500 Hz) para ser detectada por el sistema. El tiempo mínimo necesario para ser detectado será de 667 μ s el cual ha demostrado ser suficiente en diseños anteriores. La implementación se realiza a través de un registro de desplazamiento de 3 bits cuyas salidas se conectan a una compuerta **OR**, dicha compuerta solo dará un cero lógico cuando la señal de entrada se haya mantenido en nivel bajo al menos los tres ciclos de reloj requeridos.

Bloque Freq_divider

Cyclone II cuenta con cuatro (phase-locked loops) **PLL** internos que se encuentran ubicados en las esquinas del circuito integrado⁸. En este caso se utilizó uno de ellos en su régimen normal de trabajo con el fin de obtener a partir de los 50 MHz de reloj que brinda el Kit de desarrollo una frecuencia que fuera múltiplo de los 2,048 MHz que se necesitaban para sincronizar las tramas del flujo E1. A través del bloque **altpll0** se configura el **PLL** según se aprecia en la figura 3. La frecuencia final a la salida del **PLL** (**f_{out}**) dependerá de la frecuencia de entrada (**f_{in}**), del coeficiente **n** (divisor de pre-escala) y del multiplicador **m**, según la fórmula (1)

$$f_{out} = \frac{f_{in} \times m}{n} \quad (1)$$

Se tomó como señal de entrada **CLK0**, reloj interno del kit de desarrollo que entrega 50 MHz con un ciclo útil del 50 %. La salida del **PLL** se conecta a un bloque **LPMCounter** que proporciona **Altera** en las bibliotecas del **Quartus II** a fin de obtener las diferentes frecuencias con las que opera el sistema. El contador se configura para un bus de salida de 27 bits, lo cual permite obtener en su bit más significativo una señal que varía a una frecuencia de 0.25 Hz, la más baja que necesita el sistema y que es utilizada en la conformación del tono de timbre.

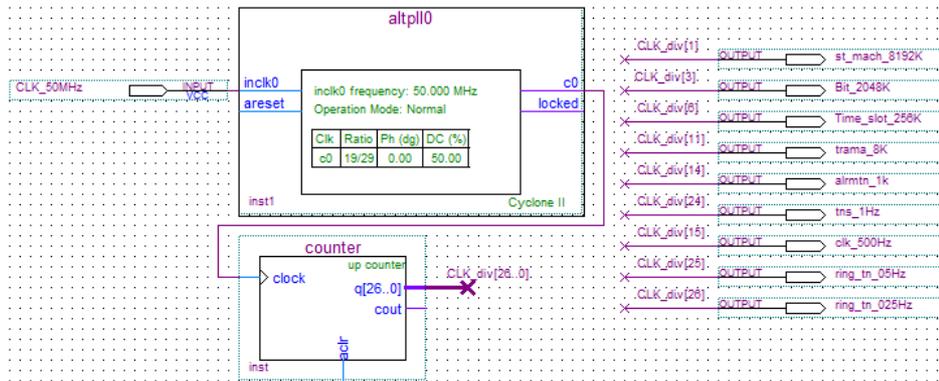


Figura 3: Diagrama en bloques del divisor de frecuencia

Se probaron 24 combinaciones (realizables) para el divisor pre-escala y el multiplicador buscando obtener una frecuencia lo más cercana posible a un múltiplo de 2.048 MHz. Los mejores resultados se obtuvieron para $m = 19$ y $n = 29$, dando una frecuencia base de 32,758 MHz a la salida del **PLL**. Esta frecuencia se divide entre 8 tomando la cuarta salida del contador obteniéndose una frecuencia final de 2,0474 MHz. El error que se comete con esta combinación es de 0,587 kHz lo cual representa un error del 0.0286%. Las otras frecuencias necesarias en el sistema se obtienen de diferentes pines de salida del contador. Por ejemplo la señal de reloj de la máquina de estados del controlador (8.196 MHz) se logra en el pin 2, la señal de reloj del códec (2048MHz) del pin 4, la del bloque **timeslot** (256kHz) del pin 7 y las restantes frecuencias para la conformación de los tonos del pin 12 al 27.

Controlador de la central.

El bloque **control_llamada**, implementado completamente en **VHDL**, está compuesto por una máquina de estados externa que modifica de forma ascendente el valor de la señal **step** correspondiente a cada uno de los cuatro pasos que sigue el controlador en la atención de cada abonado. Al llegar al último paso vuelve a comenzar el conteo para atender al próximo abonado. En el tercero de estos pasos otra máquina de estados genérica se encarga, según el valor actual de la señal **PC**, de actualizar los valores de los elementos de la matriz de estados (**PBX_GKT**) cuya estructura interna se explica a continuación. Se define un tipo de dato (**est_abn**) con los estados por los que puede transitar cada abonado.

TYPE est_abn IS (colg, desc, keyp, disc, keyp2, disc2, keyp3, comu, conv, ocup, f_ser, ll_ent, cong, n_inc, rf_ser);

VHDL permite declarar tipos compuestos por más de un valor. Un tipo compuesto es un conjunto o agrupación de valores que se trata como una unidad. Los tipos compuestos en **VHDL** pueden ser arreglos (**array**) o records (**record**). La diferencia fundamental entre ambos radica en que un arreglo es una colección de valores donde todos los elementos son del mismo tipo mientras que un record es una colección de valores cuyos elementos no tienen que ser todos del mismo tipo⁹. En este diseño se emplean ambos tipos compuestos. Se define como tipo **abonado** un record que resume toda la información necesaria para que la central controle el proceso de comunicación. Un primer campo de tipo **est_abn** que corresponderá al estado en que se encuentre el abonado, en los dos campos siguientes se almacenan los tres dígitos del número de la extensión del propio abonado y del abonado con el que se intenta comunicar (**num_ext**, **est_dest**), este último dato solo se utiliza en el estado de comunicación. También recoge el canal (**own_tslot**) propio asignado al abonado así como el que posee el abonado con el que se establece la comunicación (**other_tslot**), cuando no se está comunicando con nadie este campo toma el valor 0 que no corresponde con ningún abonado por estar este canal reservado a la señalización. El campo **caller** le sirve a la central de bandera para determinar si el abonado en cuestión fue quien realizó la llamada lo cual entre otros privilegios le permite terminarla. Por último en el campo **state_time** se almacena el tiempo transcurrido en un estado, este dato es de suma importancia en los estados que tienen restricciones temporales. La implementación del record de abonado se muestra en el siguiente fragmento de código:

TYPE abonado IS RECORD

```

estado      : est_abn;
num_ext     : std_logic_vector(11 downto 0);
ext_dest    : std_logic_vector(11 downto 0);
own_tslot  : natural RANGE 0 TO 30;
other_tslot : natural RANGE 0 TO 30;
caller     : boolean;
state_time  : natural;
    
```

END RECORD;

El tipo de datos **central** no es más que un arreglo de abonados, en este caso serán 8 aunque la capacidad máxima es de 32 y está limitada por la frecuencia máxima de reloj de 32,76 MHz que se obtiene del PLL. La capacidad máxima pudiera ser de 1024 si solo se actualizara al final de la trama. Se define luego una señal de tipo central (PBX_GKT) de donde el controlador de la central obtendrá en todo momento la información necesaria de los abonados y hacia donde luego de haber tomado las decisiones pertinentes actualizará los campos de dato de los abonados implicados en cada operación. Los números de las extensiones con los que se inicializa la central sirvieron para dar seguimiento a cada abonado en el proceso de depuración de errores pues su primer dígito coincide con el canal correspondiente de cada abonado. Los restantes campos de dato se inicializan a sus valores por defecto, **state_time** y **other_tslot**: 0, **est_abn**: colgado y **caller** en false.

La estrategia de control que sigue el controlador es la siguiente: en el tiempo comprendido entre un canal y el próximo se obtiene a partir del valor de **PC** todos los datos relacionados con el abonado que será atendido en este intervalo de tiempo, se procesa la información obtenida y se modifican de ser necesario los campos que permiten actualizar el estado del abonado que está siendo atendido y el posible segundo abonado con el que intenta establecer o tiene ya establecida una comunicación. Para la atención de cada abonado se siguen cuatro pasos fundamentales:

- Se incrementa el contador de programa (**PC**) que hace la función de identificador de abonado. Esto es posible porque el valor de PC coincide con la posición que ocupa cada abonado en la central. Los accesos se hacen luego de manera genérica (**PBX_GKT(PC)**).
- Se obtienen los datos necesarios del abonado que está siendo atendido a través del valor actual de **PC**. Esta información se asigna a señales temporales que de forma general manejan la información del abonado activo en el proceso de atención.
- Las señales temporales son procesadas por una máquina de estados que concentra la lógica de la central y que actualiza los valores de dichas señales.
- Por último se actualizan los campos de la central a partir de las señales temporales actualizadas en el paso anterior.

Esta división fue necesaria porque el valor de las señales solo se actualiza al recibirse el nuevo frente de subida del reloj y de esta manera se garantiza estar trabajando en cada momento con el valor actual del contador de programa y de los datos que a través de él se obtienen. Los pasos de atención de cada abonado se pueden seguir a través de la señal **step** como se aprecia en figura 4. La señal **num_ext** nos permite corroborar que se entró en la máquina de estados y que se está trabajando con el abonado adecuado. Nótese que entre los pasos 0 y 1 se actualiza el valor de **PC** y que entre 2 y 3 se actualiza el valor de **num_ext**.

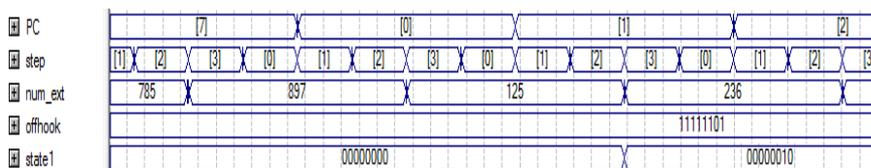


Figura 4: Simulación del proceso de atención de los abonados.

En las otras dos transiciones se lee y escriben los datos desde y hacia la central lo cual no produce modificación alguna en las señales antes mencionadas. Se aprecia además que el cambio en el vector de estados, que se produce al recibir la señal **offhook** (descolgado) del abonado 2, ocurre en el segundo valor de **PC** donde dicho abonado está siendo atendido.

Codificación de los estados

Se genera una codificación de los estados que permite seguir cada abonado en el proceso de depuración de errores. Esta codificación se emplea a través de un vector de estados donde cada posición representa a un abonado en específico. El vector, que es del tipo **std_logic_vector**, posee 32 bits en los que se asignan en orden ascendente 4 a cada abonado. De esta forma un cambio en el estado de un abonado representa un cambio en el vector de estados en las posiciones reservadas a dicho abonado. Por ejemplo si todos los abonados están en el estado de colgado, el vector de estados tiene el valor 00000000, pero si los abonados 0 y 3 pasan a descolgado manteniendo los otros su estado anterior, cambia el vector al valor 00001001. Así se logra registrar en todo momento de la simulación el estado en que se encuentra cada abonado siendo esto de gran ayuda en el proceso de depuración de errores. A continuación se relacionan los estados por los que pueden transitar los abonados, su codificación hexadecimal que es la que observamos en la simulación, una breve descripción de los mismos y de cómo se accede o sale de ellos.

Colgado (**colg**, 0)

En este estado el abonado se encuentra inactivo y puede pasar al estado de llamada entrante (**ll_ent**) si otro abonado intenta comunicarse con él, a descolgado (**desc**) si descuelga (**offhook(abonado) = 0**) y el **DTMF** de su tarjeta de abonados está libre o a gestión (**cong**) si al descolgar el **DTMF** de su tarjeta de abonados está siendo utilizado por otro abonado.

Descolgado (**desc**, 1)

En este estado el abonado recibe tono de discar (500 Hz) y permanece esperando por el primer dígito por un período máximo de 15 segundos (este tiempo se censa a partir del **state_time** del abonado). Si la central recibe un dígito válido (**datav = 1**) se pasa al estado de tecla presionada (**keyp**). Si el abonado cuelga se regresa al estado colgado (**colg**). Transcurrido el lapso de 15 segundos, de no presentarse alguna de las dos condiciones anteriores, se pasa automáticamente al estado ocupado. Esto permite liberar el **DTMF** para ser usado por otros abonados de la misma tarjeta. Se debe destacar que siempre que el abonado pasa al estado de ocupado o de número incorrecto, el **DTMF** de su tarjeta de abonados es liberado, hacer un uso lo más racional posible del **DTMF** es el objetivo que persiguen todas las restricciones temporales antes expuestas. Cada abonado puede hacer uso exclusivo del **DTMF** por aproximadamente un minuto en el peor de los casos. Los valores de las restricciones temporales de los estados que hacen uso del **DTMF** se tomaron siguiendo los valores de las centrales comerciales que emplea Etecsa. Cada vez que el abonado pasa a un nuevo estado la máquina de estados reinicia su **state_time** para recomenzar el conteo en el nuevo estado. Esta variable se incrementa continuamente y su valor crítico va a depender del estado en el que se encuentre el abonado y la frecuencia del reloj de la máquina de estados.

Tecla Presionada (**keyp**, 2)

En este estado el abonado recibe el tono de tecla presionada de frecuencia 1 kHz y permanece esperando que la tecla sea liberada por un período máximo de 10 segundos. Si es liberada la tecla y el dígito es válido (alguno de los abonados lo tiene como dígito más significativo) se pasa al estado discando (**disc**). Si el dígito no es válido se pasa al estado número incorrecto (**n_inc**). Si se supera el tiempo de 10 segundos sin liberar la tecla (**datav = 0**) se pasa al estado de ocupado.

Discando (**disc**, 3)

En este estado el abonado no recibe tono alguno y permanece esperando por el segundo dígito por un período máximo de 10 segundos. Si se recibe un dígito válido (**datav = 1**) se pasa al estado de tecla presionada 2 (**keyp2**). Si el abonado cuelga se regresa al estado colgado, esto ocurre en la mayoría de los estados a excepción del de llamada entrante donde ya se encuentra colgado y el de conversación cuando usted no es el que llama y por ende no tiene el privilegio de terminar la llamada. Luego de 10 segundos, de no presentarse alguna de las dos condiciones anteriores, se pasa al estado ocupado.

Tecla Presionada 2 (**keyp2**, 4)

El abonado recibe tono de tecla presionada (1 kHz) y permanece esperando que la tecla sea liberada por un período máximo de 10 segundos. Si es liberada la tecla y el dígito es válido (alguno de los abonados lo tiene como segundo dígito) se pasa al estado discando (**disc2**). Si el dígito no es válido se pasa al estado número incorrecto (**n_inc**). Si se supera el tiempo de 10 segundos sin liberar la tecla se pasa al abonado al estado de ocupado.

Discando 2 (disc2, 5)

En este estado el abonado no recibe tono alguno y permanece esperando por el tercer dígito por un período máximo de 10 segundos. Si se recibe un dígito válido, pasa al estado de tecla presionada 3 (**keyp3**). Si el abonado cuelga se regresa al estado colgado. Si luego de 10 segundos no se cumple alguna de las dos condiciones anteriores, se pasa al estado ocupado.

Tecla Presionada 3 (keyp3, 6)

El recibe tono de tecla presionada (1 kHz) y permanece esperando que la tecla sea liberada por un período máximo de 10 segundos. Si es liberada la tecla se libera el **DTMF** de la tarjeta a la que pertenece el abonado para que pueda ser utilizado por otro abonado que intente marcar. Si el número de extensión compuesto por los tres dígitos tecleados por el usuario y almacenados por la señal en **ext_dest** es válido o no (alguno de los abonados lo tiene como su número de extensión), en función del estado del abonado destino, se toma una de las decisiones siguientes:

- Si está colgado se pasa al estado comunicación (**comu**) y al abonado destino a llamada entrante (**ll_ent**).
- Si está fuera de servicio se pasa al estado receptor fuera de servicio (**rf_serv**).
- Si el número es incorrecto se pasa al estado número incorrecto (**n_inc**).
- Si no se libera la tecla en los 10 segundos establecidos se pasa al estado ocupado (**ocup**).

Comunicación (comu, 7)

En este estado el abonado recibe tono timbre atrás. Si el abonado destino no descuelga en un máximo de 45 segundos, se pasa a dicho abonado al estado de colgado lo cual le retira el aviso de timbre al abonado destino, y al abonado que llama se pasa al estado de ocupado. Si el que llama cuelga, ambos pasan al estado de colgado. Si el abonado destino descuelga a tiempo pasan ambos al estado de conversación (**conv**).

Conversación (conv, 8).

En este estado el abonado destino queda retenido hasta tanto el que llama no cuelgue, si el abonado destino cuelga la comunicación sigue estando activa. Durante el mismo se combinan adecuadamente las señales de recepción y transmisión del **CODEC (FSR y FSX)** de cada canal de abonado, para que pueda establecerse la conversación. Cuando el que llama cuelga este pasa al estado de colgado y el abonado destino pasa al estado de ocupado (**ocup**).

Ocupado (ocup, 9).

En este estado el abonado recibe tono de ocupado. Si en 60 segundos el abonado no cuelga, se pasa al estado de fuera de servicio (**f_ser**). Al colgar pasa al estado de colgado.

Fuera de servicio (f_ser, A).

En este estado al abonado se le retira tono de ocupado y no recibe tono alguno. Estará en este estado hasta tanto no cuelgue, no consume ningún recurso de su tarjeta de abonados pero tampoco puede recibir llamadas.

Llamada entrante (ll_ent, B).

El abonado en este estado recibe la señal de timbre a través del **SLIC**. Se mantiene en este estado por un máximo de 45 segundos. Si descuelga ambos pasan al estado de conversación como se analizó anteriormente, Si el que llama cuelga la central le pasa automáticamente al estado de colgado. Si no descuelga a tiempo también es regresado al estado de colgado.

Congestión (cong, C).

A este estado se accede al tratar de utilizar el **DTMF** de su tarjeta de abonados y estar este ocupado por otro abonado de la misma tarjeta. El carácter compartido de este recurso limita a la central a tener como máximo a dos abonados marcando.

Número incorrecto (n_inc, D).

A este estado se accede al presionar un dígito no válido en las dos primeras posiciones del número del abonado o porque la combinación de los tres sea incorrecta. Con esto se persigue limitar el uso del **DTMF** cuando la combinación de marcado ya está destinada a ser incorrecta.

Receptor fuera de servicio (rf_ser, E)

A este estado se accede al tratar de comunicar con un abonado que se encuentra en fuera de servicio.

En la figura 5 se muestra un diagrama de estados simplificado de la máquina de estados genérica que se menciona con anterioridad. Esta máquina maneja en cada momento la ubicación del abonado que atiende en su propio diagrama de estados. Las condiciones de transición pueden ser consultadas en la explicación de los estados antes descrita. El cambio de estados solo se produce en el período en que está siendo atendido el abonado o cuando se recibe una llamada de un abonado remoto.

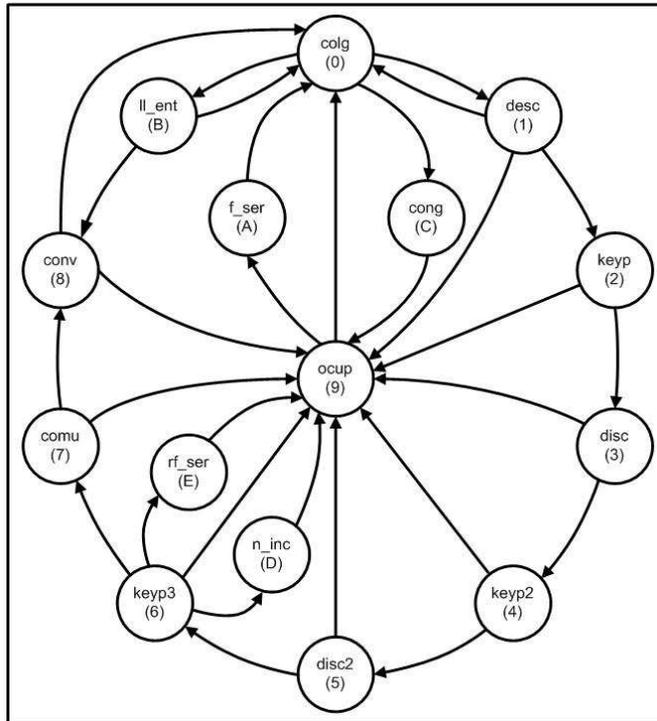


Figura 5. Diagramas de estados del controlador de llamadas.

Resultados de simulación.

Con el objetivo de corroborar la validez del diseño se realizaron múltiples pruebas de simulación de los bloques individuales que lo componen así como del sistema completo, en este artículo se muestra una prueba donde se establecen cuatro comunicaciones que involucran a los ocho abonados y que una vez establecidas son atendidas de forma simultánea.

Las frecuencias que representan las señales de voz de los abonados fueron tomadas para la simulación de forma descendente para lograr que se aprecie en la trama la ubicación de cada uno de los abonados. La señal de mayor frecuencia corresponde al abonado 0 y se ubica en el canal 1 y la de menor frecuencia que corresponde al abonado 7 se ubica en el canal 8. La codificación del vector de los canales es tal que el incremento ocurre en potencias consecutivas de 2, el primer canal de la izquierda en la figura (canal 31) corresponde con un 1 en la posición 31 del vector y en este caso está vinculado a uno de los canales de servicio, a continuación aparece el canal 0 cuya codificación corresponde con un 1 en la posición 0 del vector. En la figura 11 se aprecia claramente lo antes expuesto.

Las señales de activación de los canales quedan desfasadas de forma tal que el fin de un canal coincide con el comienzo del canal siguiente, en el caso del último canal (31) se vuelve a comenzar por el canal 0. La línea de tiempo de la figura 6 permite corroborar lo antes expuesto. Esta simulación corresponde al bloque **time_slot** explicado anteriormente.

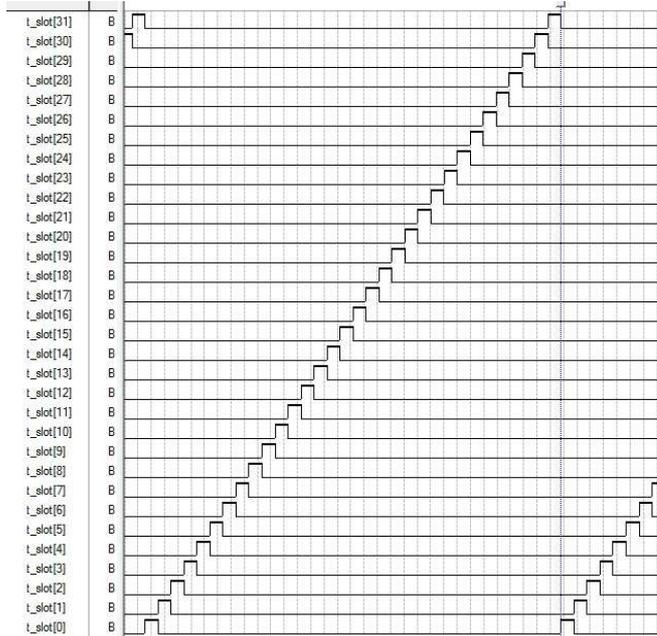


Figura 6. Simulación del bloque time_slot.

Según se observa en la figura 7, en un inicio todos los abonados se encuentran en el estado de colgado (0) como lo refleja el vector **state1**. Esto se corresponde con el vector **offhook** que relaciona las señales de igual nombre proveniente de la **SLIC** de cada abonado. Cuando el abonado 1 descuelga se le pasa al estado descolgado (1) y se mantienen a los demás abonados en el mismo estado en que se encontraban. Al abonado que se encuentra descolgado se le da tono de discar, la señal de habilitación de recepción del **CODEC fsr(1)** se activa en fase con el canal de servicio que brinda dicho tono.

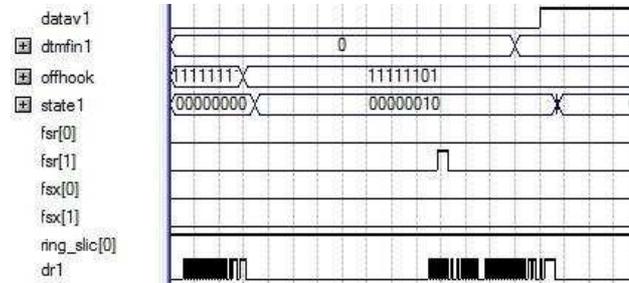


Figura 7. Primera etapa del proceso de llamada.

Se observa en la figura 8 que el abonado 1 permanece en descolgado hasta tanto no oprime el primer dígito lo cual es censado a través de la entrada **datav1** del **DTMF** al que pertenece dicho abonado. Al recibir un 1 en **datav1** se pasa al abonado (se está hablando en este caso del abonado 1 pues los demás permanecen colgados) al próximo estado (**keyp**) cuyo código es 2. Se mantiene en este estado hasta tanto no libera el dígito, esto garantiza que solo se tome el valor del dígito una vez. El abonado va transitando por los estados de discado hasta llegar al estado **keyp3** (código 6) donde se han recibido los tres dígitos de la extensión destino y se encuentra en condiciones de establecer la comunicación.

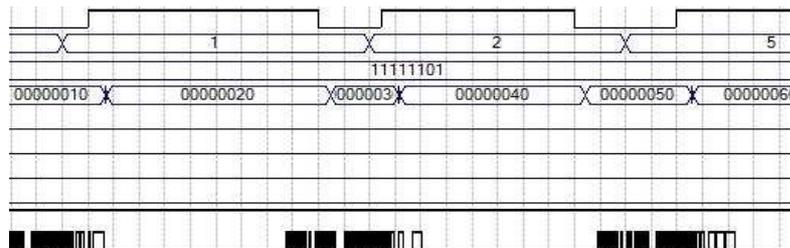


Figura 8. Simulación del proceso de discado.

Cuando es liberada la tecla correspondiente al dígito menos significativo y teniendo en cuenta que los dígitos tecleados corresponden al número 125 del abonado 0 que se encontraba colgado, se pasa al abonado 1 al estado de comunicación (8) y al abonado 0 al estado de llamada entrante (B). Al primero se le da señal de timbre atrás lo cual se aprecia en la habilitación del **fsr(1)** en fase con el correspondiente canal de servicio, al abonado 0 se le da señal de timbre a través de la salida **ring_slic(0)** conectada al **RC** del **SLIC** de dicho abonado. Al descolgar el abonado 0 ambos pasan al estado de conversación, donde se les retira a ambos la señal de timbre y se colocan en fase con el canal de cada abonado su señal de habilitación de transmisión (**fsx**) y la de habilitación de recepción del otro abonado (**fsr**), lo cual permite que la conversación se establezca. En la figura 9 apreciamos el paso de los abonados 0 y 1 al estado de conversación.

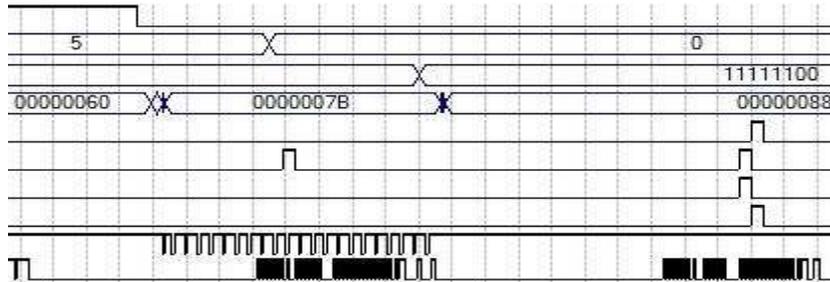


Figura 9. Simulación del bloque time_slot.

En la figura 10 se aprecia con más detalle cómo se alinean las señales de habilitación de recepción y transmisión con el canal correspondiente en la trama. El **fsx(0)** y el **fsr(1)** se encuentran alineados con el canal 1 donde se coloca en la trama las señales de voz codificada del abonado 0. El **fsx(1)** y el **fsr(0)** se encuentran en fase con el canal 2 que corresponde al abonado 1. Además se aprecia que en el canal 0 no se coloca señal alguna por estar reservado a otras funciones. La central es capaz de manejar satisfactoriamente cuatro llamadas internas donde están implicados todos los abonados de la misma.

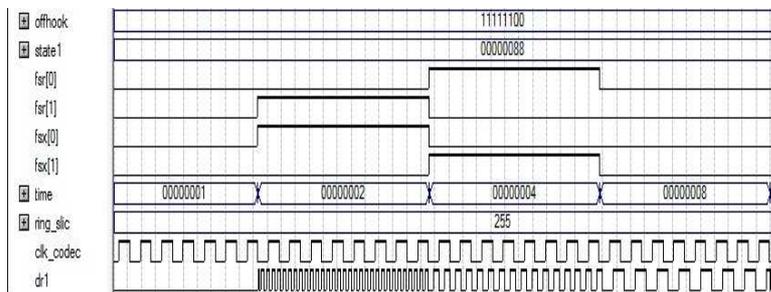


Figura 10. Simulación del bloque time_slot.

En la figura 11 correspondiente a la simulación de las cuatro llamadas simultáneas, se observa que todos los abonados están descolgados (**offhook = 00000000**) y en estado de conversación (**state1 = 88888888**).

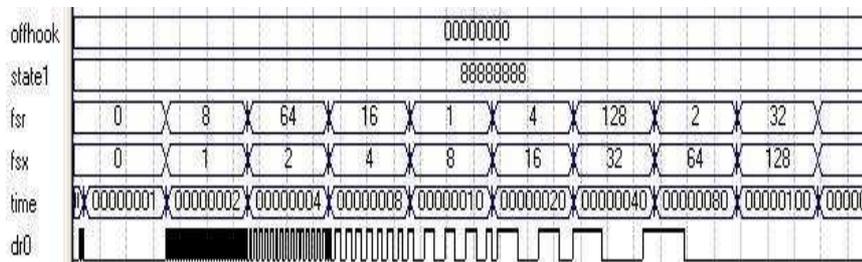


Figura 11. Simulación de cuatro conversaciones simultáneas.

Las señales de **fsx** y **fsr** activan la transmisión y recepción de dos abonados por canal en el estado de conversación. Cada una de ellas coincide con la potencia de 2 correspondiente al canal del abonado que representan. Por ejemplo en el caso de la conversación entre 0 y 3, en el canal 1 (abonado 0) se activa el **fsx** de 0 ($2^0=1$) y el **fsr** de 3 ($2^3=8$) mientras que en el canal 4 (abonado 3) se activa el **fsx** de 3 y el **fsr** de 0. Esto permite que siempre el abonado hable en su propio canal y escuche en el canal del abonado con que se comunica. En esta simulación se encuentran conversando al mismo tiempo el abonado 0 con el 3, el 1 con el 6, el 2 con el 4 y el 5 con el 7.

CONCLUSIONES

Se logró obtener un diseño de controlador empotrado para una PBX de ocho abonados utilizando el software de desarrollo Quartus II de Altera.

El diseño permite expandir las potencialidades de diseños anteriores realizados en el proyecto, reutiliza parte del código de dichos diseños lo cual demuestra la factibilidad del desarrollo con lógica programable y VHDL.

Se comprueba el funcionamiento de la central a través de pruebas de simulación, observando la capacidad de la misma de manejar hasta cuatro llamadas internas donde se ven involucrados los ocho abonados que la componen.

Este diseño sirve de base para un proyecto más ambicioso donde este bloque será un periférico de un sistema basado en Nios II de un Gateway de VoIP el cual constituye el objetivo final del proyecto.

REFERENCIAS

- [1] **V. Marín Contreras, et al.**, "Interfaz digital de abonados" en *Ingeniería Eléctrica, Automática y Comunicaciones*. 2007
- [2] **M. García Montoya, et al.**, "Implementación de los protocolos de comunicación para VoIP: RTP/RTCP sobre FPGAs de Altera," presentado en el III Simposio Internacional de Electrónica: diseño, aplicaciones, técnicas avanzadas y retos actuales, 2011.
- [3] **UIT-T**, "Características del equipo múltiplex MIC primario que funciona a 2048 Kbit/S," ed. Libro Azul: Recomendación UIT-T G.732, 1993.
- [4] **Altera**, "Nios Development Board Cyclone II Edition Reference Manual", Octubre 2006.
- [5] **Intersil**, HC5503PRC data sheet. "Low Cost SLIC For Large Telecom Switches", Agosto 2003.
- [6] **Mitel**, MT8870D/MT8870D-1 data sheet. "Integrated DTMF Receiver", Mayo 1995.
- [7] **STMicroelectronics**, "ETC5054 - ETC5057 data sheet. SERIAL INTERFACE CODEC/FILTER", Marzo 2000.
- [8] **Altera**, "Cyclone II Device Handbook," vol. 1, Febrero 2007.
- [9] **G. Güichal**, "Diseño Digital Utilizando Logicas Programables" ed. Universidad Tecnológica Nacional. Facultad Regional Bahía Blanca, 2005.

AUTORES

Jaime Fontanella Lauzardo

Ingeniero en Automática y Computación. Desde el año 2008 se desempeña como Profesor Instructor del Departamento Docente del Centro de Investigaciones en Microelectrónica de la Facultad de Eléctrica de la CUJAE. Vinculado al proyecto "Plataforma de Conmutación de paquetes" ha venido trabajando desde 2008 en este diseño que forma parte de su tesis de maestría en diseño de sistemas electrónicos.

e-mail: jaime.fl@electronica.cujae.edu.cu

Víctor Marín Contreras

Ingeniero en Telecomunicaciones. Doctor en Ciencias Técnicas. Se desempeña como Profesor Titular del Departamento Docente del Centro de Investigaciones en Microelectrónica de la Facultad de Ingeniería Eléctrica de la CUJAE. Dirige el proyecto "Plataforma de Conmutación de Paquetes" y ha sido tutor de varias tesis y diplomas dentro de las temáticas de diseño digital aplicado a la telefonía IP.

e-mail: victor@gkt.cu

René Yañez de la Rivera

Ingeniero en Telecomunicaciones. Doctor en Ciencias Técnicas. Se desempeña como Profesor Titular del Departamento de Telecomunicaciones de la Facultad de Ingeniería Eléctrica de la CUJAE. Al mismo tiempo, es asesor de la presidencia de la Empresa Mixta GKT S.A. y sus intereses de investigación han estado dirigidos hacia las redes móviles y las comunicaciones inalámbricas. Esta vinculado al proyecto "Plataforma de Conmutación de Paquetes" y ha sido tutor de varias tesis y diplomas dentro de esa temática.

e-mail: yanez@gkt.cu