



Universidad de Buenos Aires

Facultad de Ingeniería

Tesis de Maestría en Ciencias de la Ingeniería

Desarrollo de circuitos integrados CMOS para aplicaciones de RFID

Andrew Parlane

Director: Dr. Ing. Mariano Garcia-Inza

Codirectores: Ing. Federico G. Zacchigna, Ing. Octavio Alpago

Resumen

Este trabajo presenta el diseño de circuitos integrados digitales para la implementación de un TAG RFID en un chip CMOS. Los requerimientos de los circuitos son definidos por un proyecto de investigación marco cuyo objetivo general es el desarrollo de sensores inalámbricos de radiación ionizante para dosimetría médica. Esto requiere la integración de diferentes subsistemas en un ASIC a fabricar en tecnología CMOS.

Los circuitos presentados en esta tesis consisten en un bloque digital de control, que puede recibir y responder a las tramas definidas en la norma ISO/IEC 14443A, y mediante un protocolo propietario encima de la norma, controlar otros subcircuitos del chip necesarios para la adquisición de la señal dosimétrica y su posterior transmisión inalámbrica. La implementación de la norma es presentada en un núcleo IP genérico. Por lo tanto es apto para uso en otros proyectos y con cualquier lector comercial que trabaje bajo norma. El protocolo propietario consiste en cinco mensajes que permiten el muestreo de hasta quince TAGs de forma sincronizada. El muestreo simultáneo de múltiples TAGs sensores utilizando RFID es una estrategia novedosa desarrollada en esta tesis, el cual representa un potencial avance en el desarrollo de métodos de control dosimétrico en aplicaciones médicas.

El trabajo realizado incluye el diseño de los bloques digitales utilizando HDL SystemVerilog y su verificación funcional. Luego, se utiliza el kit de diseño del proceso (PDK, por sus siglas en Inglés) XH018 (nodo tecnológico de 180 nm) de la foundry XFAB para realizar la síntesis, el place and route y la generación de las máscaras de fabricación (layout).

Las pruebas de verificación llevadas a cabo dan alta confianza en el diseño. Las mismas incluyen: más de cien horas de simulaciones del RTL ejecutando 187 aserciones de SystemVerilog más de cien mil millones de veces y generando informes de cobertura de código, la verificación formal de equivalencia entre lógica RTL y los netlists post síntesis y post implementación, Design Rules Check (DRC), y Layout Vs Schematic (LVS); todas esas pruebas tienen resultados favorables.

El layout final no tiene violaciones de timing, el slack de Setup menor es 1,59 ns y de Hold es 0,02 ns, el área utilizada es 0,087 mm² (295,68 μm por 294,00 μm), y la estimación de consumo de potencia promedio es 256 μW.

Aclaraciones

Aclaraciones Por cuestiones de claridad en esta tesis cuándo se dice que una señal está en ‘1’ o ‘0’ significa que la señal tiene un valor lógico igual a ‘1’ o ‘0’ respectivamente.

El código RTL, los bancos de pruebas y los scripts de síntesis y place & route están publicados de forma abierta en GitHub bajo la licencia GNU v3.0, de manera que cualquier parte de este trabajo pueda ser utilizada en otros proyectos académicos o comerciales. Para facilitar la adopción y modificación de este trabajo por la comunidad internacional, el código y los scripts están escritos con nombres y comentarios en Inglés.

Para facilitar la lectura de esta tesis las señales, variables y parámetros están escritos en *cursiva*, los módulos, clases y funciones están escritos en **negritas**. Además se utilizan paréntesis para nombrar a las funciones, aunque sin sus argumentos, por ejemplo: **compare()**. Finalmente los nombres de las tramas definidas en los protocolos están escritos en MAYÚSCULAS.

Índice

1	Introducción	6
	Motivación y Contexto de Trabajo	6
	ISO/IEC 14443A	8
	ISO/IEC 14443-1: Características Físicas	9
	ISO/IEC 14443-2: Radiofrecuencia Potencia y Señal Interfaz	9
	ISO/IEC 14443-3: Inicialización y Anticolisión	12
	ISO/IEC 14443-4: Protocolo de Transmisión	17
2	Descripción General del Proyecto Marco	20
3	Implementación y Verificación	23
	Interfaces	24
	Marco de Verificación	27
	Transacciones	31
	Controladores	31
	Monitores	31
	Generador y Conversores de Transacciones	31
	Secuencias	31
	Otros	31
	Modelos Analógicos	31
	Estructura de los Bancos de Pruebas	31
	ISO/IEC 14443A núcleo IP	31
	ISO/IEC 14443-2A	31
	subcarrier	31
	bit_encoder	31
	tx	31
	sequence_decode	31
	iso14443_2a	31
	ISO/IEC 14443-3A	31
	frame_decode	31
	deserialiser	31
	FDT	31
	CRC_A	31

crc_control	31
serialiser	31
frame_encode	31
framing	31
routing	31
initialisation	31
ISO/IEC 14443-3A	31
ISO/IEC 14443-4A	31
ISO/IEC 14443A	31
Otros	31
synchroniser	31
active_low_reset_synchroniser	31
pause_n_latch_and_synchroniser	31
Aplicación - Interfaz con el sensor MOSFET de radiación y el ADC	31
Sincronización	31
Protocolo	31
Ejemplos	31
Marco de Verificación Extendido	31
Implementación	31
signal_control	31
adapter	31
radiation_sensor_digital_top	31
4 Síntesis y Place & Route	32
Síntesis	32
Preparación de librerías	32
Design Planning	32
Place & Route	32
LVS / DRC	32
5 Resultados y Conclusiones	33
Recomendaciones para Trabajos Futuros	33

Introducción

Motivación y Contexto de Trabajo

La radioterapia es un tratamiento médico que consiste en utilizar radiación ionizante para eliminar células cancerígenas que forman tumores. Sin embargo, la radiación puede presentar riesgos a tejido sano, especialmente si la dosis aplicada es mayor a la necesaria o mal localizada. Hay varios incidentes registrados donde algunas personas recibieron una sobredosis durante radioterapia, y en algunos casos dosis letales [8]. Por otro lado, una dosis localizada pero demasiado pequeña reduciría la efectividad del tratamiento. Por esas razones es muy importante adoptar consideraciones de QA (Quality Assurance), la cual puede definirse como:

Los procedimientos que aseguren el cumplimiento de las prescripciones médicas con respeto a la dosis entregada al volumen deseado, junto con una dosis mínima a tejido sano, exposición mínima al personal, y el monitoreo adecuado del paciente para determinar el resultado del tratamiento. [15, traducción mía]

Una técnica importante en QA es dosimetría in-vivo (IVD), que es la práctica de medir la dosis recibida durante el tratamiento en tiempo real. La IVD es recomendada para su uso en radioterapia [14][5], con un error menor del 3 % por tratamiento. Existen varios tipos de sensores de radiación que pueden ser utilizados en IVD con radioterapia. Entre ellos se encuentran los sensores MOSFET, los cuales tienen varias ventajas, como por ejemplo que pueden ser leídos en tiempo real o posteriormente, son pequeños y robustos, pero también tienen algunas limitaciones [8]. Los investigadores del Laboratorio de Física de Dispositivos - Microelectrónica de la Facultad de Ingeniería de la Universidad de Buenos Aires han trabajado durante los últimos años en mejorar el desempeño de sensores MOSFET de radiación para uso en IVD [3][2]. Otra ventaja de los sensores MOSFETs es que pueden ser integrados en un mismo chip con circuitos adicionales que permitan su lectura, digitalización y posterior transmisión de los resultados en tiempo real. Esto representa una importante ventaja frente al método usual de lectura post irradiación, ya que el seguimiento en tiempo real permitiría ajustar la dosis durante la ejecución del

tratamiento.

Un sensor pequeño construido con una cantidad mínima de componentes permitiría obtener arreglos de sensores con excelente resolución espacial y así realizar un mapeo dosimétrico de una zona de interés. A la hora de implementar esta solución, los TAGs RFID (Radio Frequency Identification) pasivos son una excelente opción, ya que sólo requieren del circuito integrado y una antena externa. La alimentación y la comunicación pueden realizarse a través del campo electromagnético generado por el dispositivo de lectura. Un sistema de estas características, además de ser más simple, sería más cómodo para el paciente y más práctico para su uso en el campo médico.

En su tesis de 2018 [13], Arana analizó la relación entre la frecuencia de operación de un TAG RFID y su distancia máxima de la lectura considerando los límites de exposición de humanos a campos electromagnéticos definidos en IEEE C95.1. Los resultados muestran que una frecuencia en el orden de 10 MHz daría el mejor rango de operación mientras manteniendo el campo electromagnético (EM) dentro de límites seguros. Esta frecuencia se encuentra cercana a los 13.56 MHz de RFID HF (High Frequency). Arana muestra en su tesis el diseño de la antena y de circuitos integrados analógicos para un TAG ISO/IEC 14443 tipo A que puede funcionar a una distancia de 30 cm de la lectura. También en una publicación de 2014 [1], Alcalde et al. presentan el diseño y fabricación de un TAG RFID que implementa parte de la norma ISO/IEC 14443 tipo A. El funcionamiento es verificado a través de mediciones experimentales del sistema funcionando en loopback.

Un atributo del protocolo ISO/IEC 14443A es que permite hasta 15 TAGs activos al mismo tiempo. Esto da la posibilidad de obtener muestras de los sensores de forma sincronizada, con la ventaja de medir en múltiples lugares del cuerpo del paciente. Esto contribuye a mejorar la verificación de la ejecución del tratamiento planificado y por lo tanto se alinea con los criterios de QA.

En un artículo publicado en 2016 [18], Villani et al. se desarrolló un sensor de radiación inalámbrico para uso en IVD por radioterapia. El sensor emite una señal RF con frecuencia que depende de la dosis de radiación recibida. La ventaja de este enfoque es la simplicidad del diseño, el circuito integrado no necesita un bloque digital complejo para soportar el protocolo. Las desventajas son: solo es posible usar solo un sensor a la vez, requiere una fuente de alimentación externa (batería), no permite configurar el sensor inalámbricamente, y por no haber sido diseñado dentro de una norma, no es compatible con otros equipos requiriendo de un lector diseñado ad hoc para la aplicación.

Los circuitos diseñados en esta tesis permitirán implementar una red de sensores inalámbricos cuya finalidad es medir dosis de radiación en un tratamiento de radioterapia en diferentes lugares de interés. La adquisición en múltiples puntos tiene como objetivo mejorar el control de la irradiación para ajustar con mayor precisión la dosis entregada por el acelerador a la planificación previa. Típicamente, la dosis entregada por los LINAC no es lineal con el tiempo por lo cual, para que la medición en tiempo real sea útil, se requiere sincronizar el muestreo de los sensores.

ISO/IEC 14443A

La norma ISO/IEC 14443 fue desarrollada por la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (IEC), específicamente por el grupo de tareas 2 del grupo de trabajo 8 de la subcomisión del comité técnico mixto 1. Fue publicado primero en 2001 con dos interfaces distintas: tipos A y B. Tipo A fue desarrollado en colaboración con Mikron (adquirida desde entonces por Phillips), basado en su tecnología Mifare. Esta interfaz fue diseñada como una tarjeta para almacenar datos únicamente. Tipo B fue desarrollado en colaboración con varios operadores de sistemas de transporte públicos de Europa e Innovatron. Principalmente diseñado para pagos de tarifas pero también puede funcionar como billetera electrónica y verificación de identidad. Esta interfaz fue diseñada como una tarjeta que al tener un microprocesador, además de almacenar datos, cuenta con capacidad de procesamiento. Al principio estas dos interfaces fueron complementarias, pero luego de los años sus aplicaciones se ampliaron y diversificaron [16].

La norma viene en cuatro partes:

- ISO/IEC 14443-1: Características Físicas.
- ISO/IEC 14443-2: Radiofrecuencia Potencia y Señal Interfaz.
- ISO/IEC 14443-3: Inicialización y Anticolisión.
- ISO/IEC 14443-4: Protocolo de Transmisión.

La parte una define atributos físicos de la PICC (Proximity Card, la tarjeta o TAG) como las dimensiones y el rango de operación a temperatura ambiente. La parte dos define la interfaz inalámbrica de la transmisión de potencia y comunicaciones bidireccionales entre un PCD (Proximity Coupling Device, la lectura) y la PICC. La parte tres define el formato de los bytes, la estructura de las tramas y los mensajes necesarios para el descubrimiento de todas las PICCs presentes en el campo del PCD y para activarlas.

Finalmente, la parte cuatro define un protocolo de bloques para la configuración de la PICC y la transmisión de mensajes de nivel aplicación.

Un sistema tiene un PCD que es el maestro y una o más PICCs que son los esclavos. Una PICC solo responde a solicitudes, no inicia comunicaciones. Durante el proceso de inicialización múltiples PICCs pueden responder a la misma solicitud, así colisiones son posibles y esperadas. Después del proceso de inicialización, en un sistema correctamente configurado, solo una PICC responde a cada solicitud.

ISO/IEC 14443-1: Características Físicas

ISO/IEC 14443-2: Radiofrecuencia Potencia y Señal Interfaz

El PCD emite un campo electromagnético con frecuencia $f_c = 13,56$ MHz. Las PICCs dentro de este campo se acoplan inductivamente al campo para la transferencia de potencia. El campo es modulado para la comunicación entre los dispositivos. El PCD usa modulación de amplitud para enviar información a las PICCs, y las PICCs envían sus respuestas con modulación de carga mediante un subcarrier. La norma define varias tasas de bits, pero las comunicaciones siempre comienzan con tasa de bits $f_c/128 \approx 106$ Kbps en las dos direcciones. La codificación de los bits y los parámetros de modulación dependen de: la dirección de la comunicación, si las PICCs son tipo A o B, y la tasa de bits. La [Figura 1.1](#) muestra una representación de las dos direcciones de comunicación para PICCs de tipo A y de tipo B.

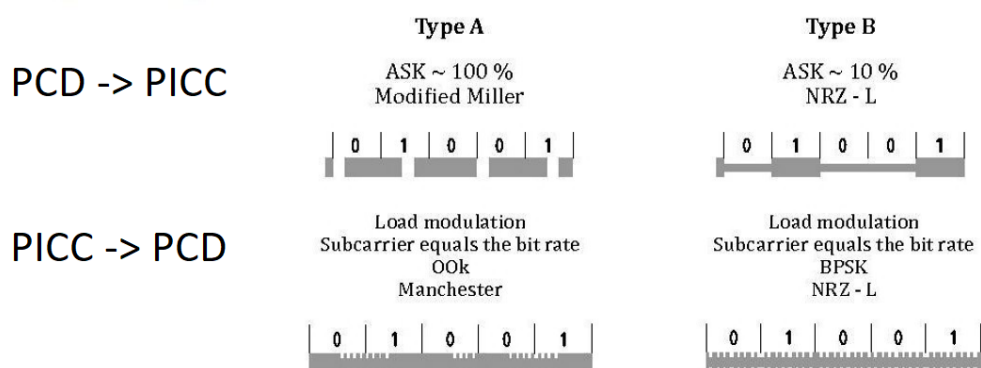


Figura 1.1: Comunicaciones con una tasa de bits de $f_c/128$ [9, Modificaciones Mías]

Para PICCs de tipo A la modulación por comunicaciones desde el PCD hasta la PICC

es ASK (Amplitude Shift Keying). Cuando la amplitud de la portadora baja al 5 % del campo original, la norma define esto como una pausa. Un esquema de la misma se muestra en la [Figura 1.2](#), con valores que se definen en el [Cuadro 1.1](#). Debido a que una PICC pasiva recibe su potencia inalámbricamente desde el campo electromagnético, es responsabilidad de la PICC mantener los rails de alimentación internos a tensiones adecuadas durante las pausas para que el sistema no cambie de estado. La codificación de los bits es Modified Miller. Cada trama comienza con un SOC (Start of Communication), tiene un número de bits de datos y termina con un EOC (End Of Communications). SOC, EOC y los valores lógicos son enviados mediante secuencias. Cada secuencia tiene el largo de una duración de bit, lo que es 128 ciclos de la portadora. La presencia y la ubicación de una pausa dentro de una duración de bit define el tipo de la secuencia, como es mostrado en la [Figura 1.3](#). Una pausa que ocurre al principio de la duración de bit es una secuencia Z, una pausa que ocurre en el medio de la duración de bit es una secuencia X, y una bit sin pausas es una secuencia Y. El SOC es la secuencia Z. Un '1' lógico es una X. Un '0' lógico depende en la última secuencia, si fue una X, se envía una Y, si fue una Y o una Z, se envía una Z. La trama termina con el EOC lo que es un '0' lógico, seguido por una secuencia Y. La [Figura 1.4](#) muestra dos ejemplos de tramas.

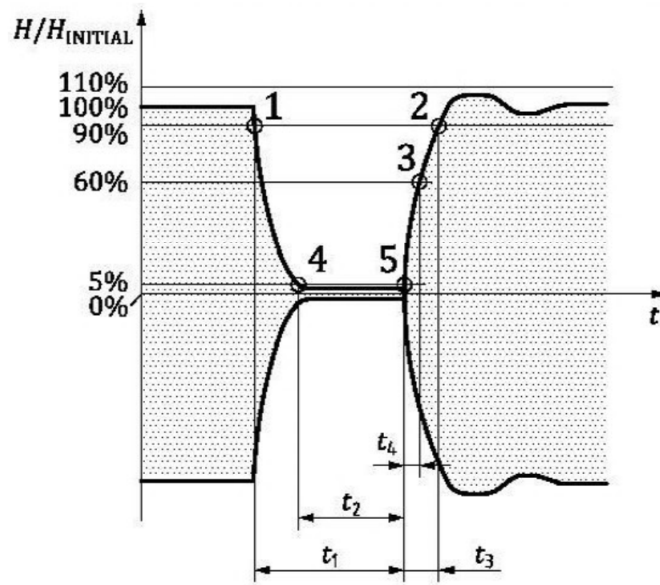


Figura 1.2: El formato de una pausa tipo A [9]

Parámetro	Condición	Mínimo	Máximo
t_1		$28/f_c$	$40,5/f_c$
t_2	$t_1 > 34/f_c$	$7/f_c$	t_1
	$t_1 \leq 34/f_c$	$10/f_c$	
t_3		$1,5 \cdot t_4$	$16/f_c$
t_4		0	$6/f_c$

Cuadro 1.1: Parámetros de timing para una pausa [9]

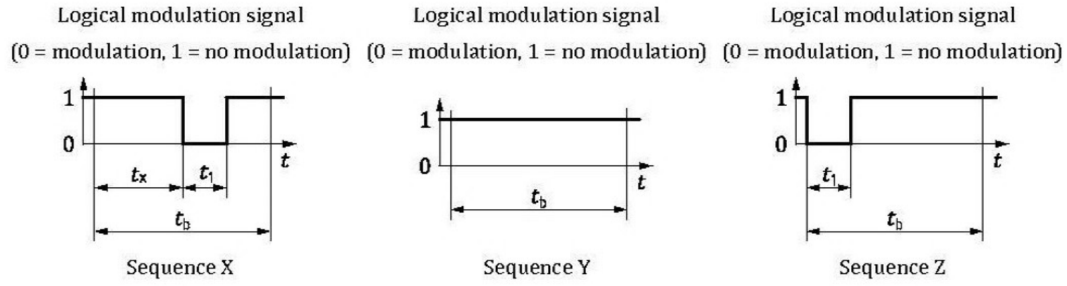


Figura 1.3: Codificación Modified Miller [9]

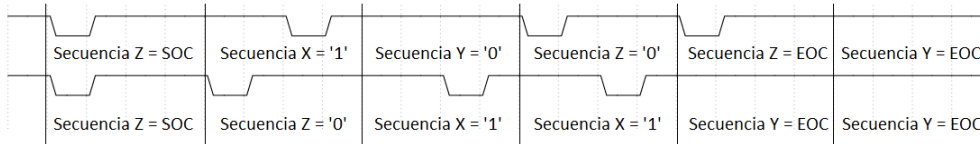


Figura 1.4: Ejemplos de tramas PCD → PICC tipo A

Para responder las PICCs tipo A usan modulación de carga para generar una subportadora. La frecuencia de la subportadora (f_s) depende de la tasa de bit. Para una tasa de bit de $f_c/128$ la frecuencia es $f_c/16 \approx 848 \text{ kHz}$. Como en el caso de comunicaciones del PCD a la PICC la duración de un bit es $128/f_c = 8/f_s$. La portadora debe ser solo modulada con la subportadora cuando la PICC está enviando datos. Los bits a transmitir tienen codificación Manchester, para representar un '1' lógico la señal es en '1' por la primera mitad de la duración de bit, y en '0' por la segunda mitad, para representar un '0' lógico es al revés. La señal que maneja el modulador de carga es la operación lógica AND entre la señal de codificación Manchester y la subportadora como se muestra en la [Figura 1.5](#). Una trama comienza con un SOC lo que es un '1' lógico, y termina con un EOC lo que es una duración de bit sin modulación, eso es a decir que la señal al modulador de carga es en '0' por toda la duración de bit. La [Figura 1.6](#) muestra un ejemplo de una trama que ingresa a la compuerta AND con la subportadora.

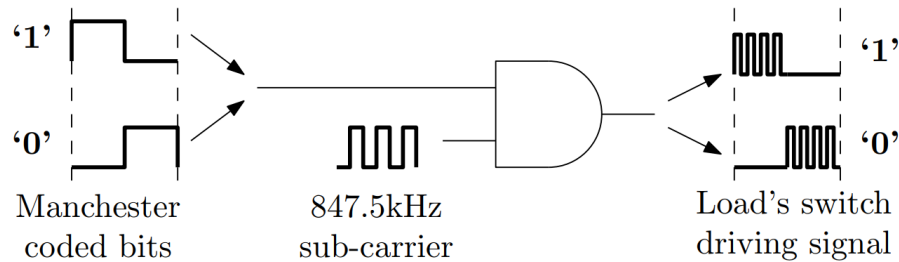


Figura 1.5: Codificación de bits y modulación de la portadora con la subportadora para comunicaciones de la PICC a PCD [1]

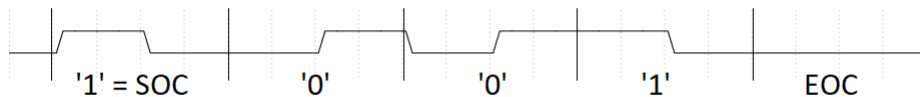


Figura 1.6: Ejemplo de una trama por comunicaciones PICC tipo A \rightarrow PCD.

ISO/IEC 14443-3: Inicialización y Anticolisión

La información presentada en esta sección es válida considerando una tasa de bit de $f_c/128$ en cada dirección y PICCs tipo A.

La parte 3 de la norma define un FDT (Frame Delay Time), lo que es el tiempo entre dos tramas transmitidas en direcciones opuestas. Los tiempos del FDT definidos en la

norma son especificados en números de ciclos de la portadora, y son medidos entre el último flanco de modulación de la primera trama y el primer flanco de modulación de la segunda trama. El Cuadro 1.2 muestra los FDTs especificados en la norma. En el caso del FDT entre una trama del PCD y una de la PICC, el FDT usado depende en el último bit lógico de la trama, y los valores especificados son absolutos para tramas de inicialización y tiempos mínimos para otras tramas.

Primera Trama	Segunda Trama	Último Bit	FDT
PCD → PICC	PICC → PCD	'0'	1172
		'1'	1236
PICC → PCD	PCD → PICC		1272

Cuadro 1.2: Valores del FDT en número de ciclos de la portadora.

Una trama comienza con un SOC, después se envían los datos empezando por el bit menos significativo, y termina con un EOC. Cada 8 bits de data hay un bit de paridad impar, así que el número de los 1s en cada byte más su bit de paridad es impar. Hay tres tipos de tramas definidas:

Tramas Cortas

Tienen siete bits de datos sin bit de paridad.

Tramas Estándares

Tienen un número de bytes enteros, cada uno con un bit de paridad.

Tramas Anticolisión orientada a bits

Tienen siete bytes cada uno con un bit de paridad, y están divididas en dos: la primera parte es enviada desde el PCD y la segunda parte desde la PICC. La partición puede ser después de un byte entero, incluyendo su bit de paridad (Figura 1.7 arriba), o en medio de un byte (Figura 1.7 abajo). Estas tramas son usadas durante el proceso de inicialización para que el PCD pueda detectar todas las PICCs en su campo electromagnético.

Unas de las tramas definidas en la norma terminan con un CRC16 (Cyclic Redundancy Check de 16 bits). El CRC16 permite el receptor de la trama determinar si el contenido fue corrompido. El polinomio del CRC es $P(x) = x^{16} + x^{12} + x^5 + 1$, y el valor inicial es 0x6363 [10][12].

Cada PICC tiene un UID (Unique Identifier) que el PCD puede utilizar para enumerar todas las PICCs en su campo electromagnético y elegir cuáles activar. A pesar de que el nombre indique que el ID es único, no tiene que ser así. Está permitido usar un ID

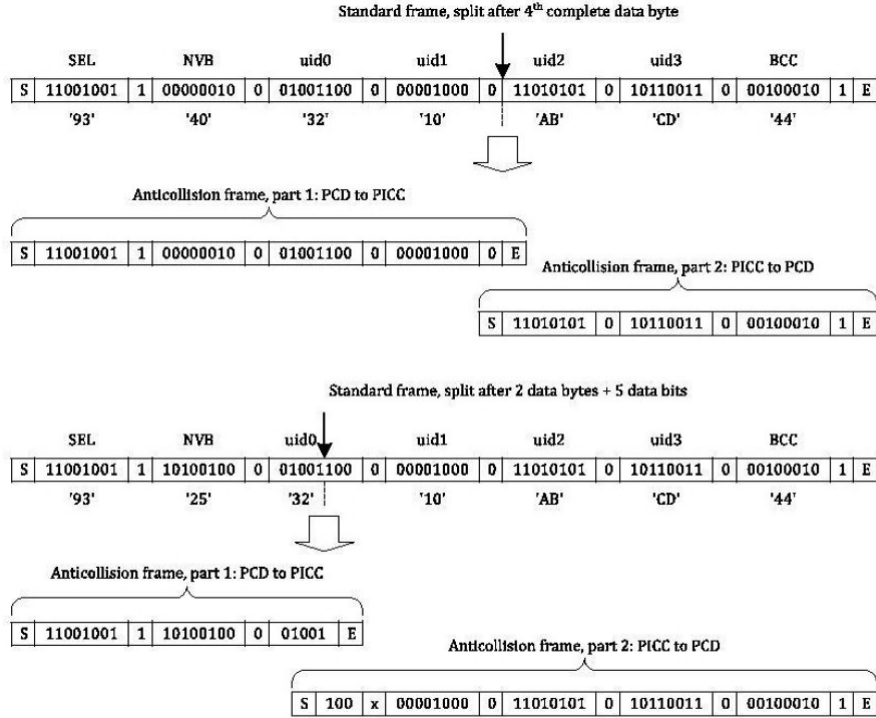


Figura 1.7: Trama Anticolisión orientada a bits [10, Modificaciones Más]

aleatorio (RID), o un ID no único (NUID). Los UUIDs pueden ser: simples (4 bytes), dobles (7 bytes), o triples (10 bytes). Es requerido que todas las PICCs presentes en el campo de un PCD tienen UUIDs únicos. Cuando se usan PICCs con NUIDs o RIDs es posible que haya conflictos. Debido a la gran cantidad de NUIDs posibles, la probabilidad de conflictos es mínima en un sistema bien diseñado.

Para identificar las PICCs presentes y activarlas, la norma define cinco comandos y sus respuestas, mostrados en el Cuadro 1.3. La Figura 1.8 especifica cómo una PICC debería responder cuando recibe un comando dependiendo de su estado actual.

PCD → PICC			PICC → PCD		
Solicitud	Tipo	CRC	Respuesta	Tipo	CRC
REQA	Corta	No	ATQA	Estándar	No
WUPA	Corta	No	ATQA	Estándar	No
ANTICOLLISION	Anticolisión	No	ANTICOLLISION	Anticolisión	No
SELECT	Estándar	Sí	SAK	Estándar	Sí
HLTA	Estándar	Sí	Sin Respuesta		

Cuadro 1.3: Solicitudes y respuestas definidas en ISO/IEC 14443-3A

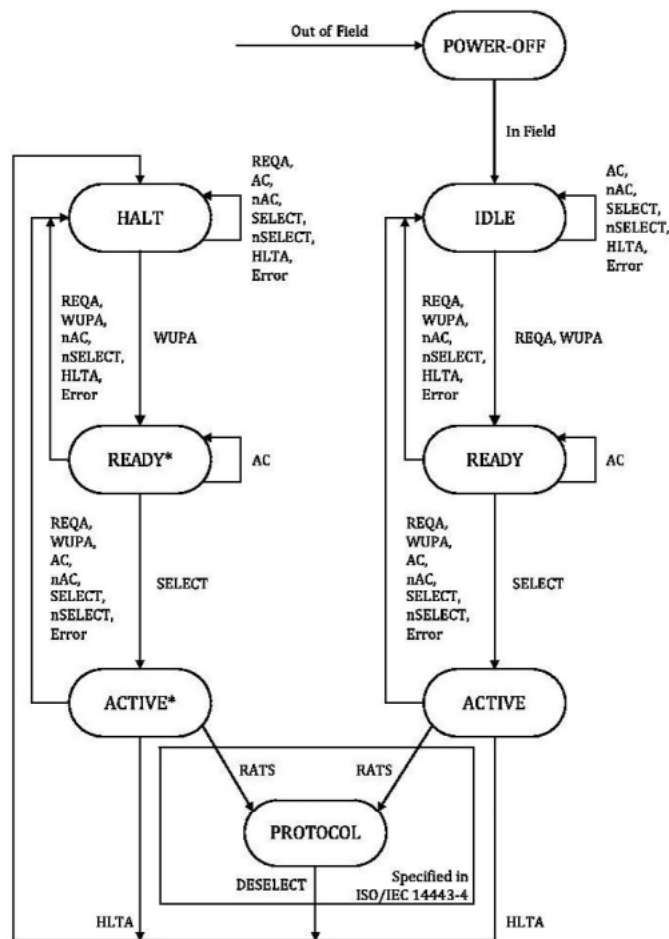


Figura 1.8: Diagrama de transiciones de estados [10]

Cuando una PICC entra en un campo electromagnético, comienza en el estado IDLE. Un comando REQA o WUPA cambia el estado a READY. En ese estado el PCD puede enviar comandos de ANTICOLLISION para determinar el UID de una de las PICCs presentes. Con ese UID el PCD puede enviar un SELECT para mover la PICC al estado ACTIVE. Desde aquí el PCD puede enviar un RATS para terminar activando la PICC. En el caso de recibir un error, o un comando no esperado, la PICC vuelve al estado IDLE. Hay tres otros estados también: HALT, READY* y ACTIVE*. Las diferencias únicas entre estos estados y IDLE, READY y ACTIVE, son: WUPA es el único comando aceptado en HALT y el comando REQA es ignorado, y un error o un comando no esperado recibido en READY* o ACTIVE* causa la PICC volver a HALT en vez de IDLE. La ventaja de esto es que si el PCD decide que no quiere activar una PICC particular, puede ponerla en el estado HALT, y enviando un nuevo REQA comenzará el proceso de inicialización de nuevo en todas las PICCs menos aquellos que están en HALT.

En el caso de una PICC con UID doble el proceso de ANTICOLLISION y SELECT tiene que repetir dos veces antes de que la PICC se mueva al estado ACTIVE. En el primer lazo, los mensajes usan los primeros tres bytes de su UID junto con una etiqueta cascada: CT (Cascade Tag). La PICC responde al SELECT especificando que su UID no está completo todavía, y el lazo comienza de nuevo, esta vez usando los últimos cuatro bytes del UID. Por una PICC con UID triple este proceso tiene tres lazos. La [Figura 1.9](#) muestra este proceso.

El comando de ANTICOLLISION funciona de la siguiente forma: el PCD envía un UID parcial, y todas las PICCs en el campo electromagnético cuyas UIDs corresponden con la parte enviada, responden con los demás bits de sus UIDs. Debido al FDT fijo todas las PICCs comienzan responder de forma sincronizada. Las respuestas tienen codificación Manchester, por lo tanto cuándo dos PICCs envían valores lógicos diferentes el PCD puede detectar la colisión porque hay modulación durante todo el tiempo de bit. De esta manera el PCD sabe qué parte del UID es compartido entre todas las PICCs, y usando una búsqueda binaria puede determinar el UID completo de una de las PICCs presentes. Esa PICC entonces puede ser activada o puesta en el estado HALT, y después el PCD puede repetir el proceso para enumerar todas las demás PICCs.

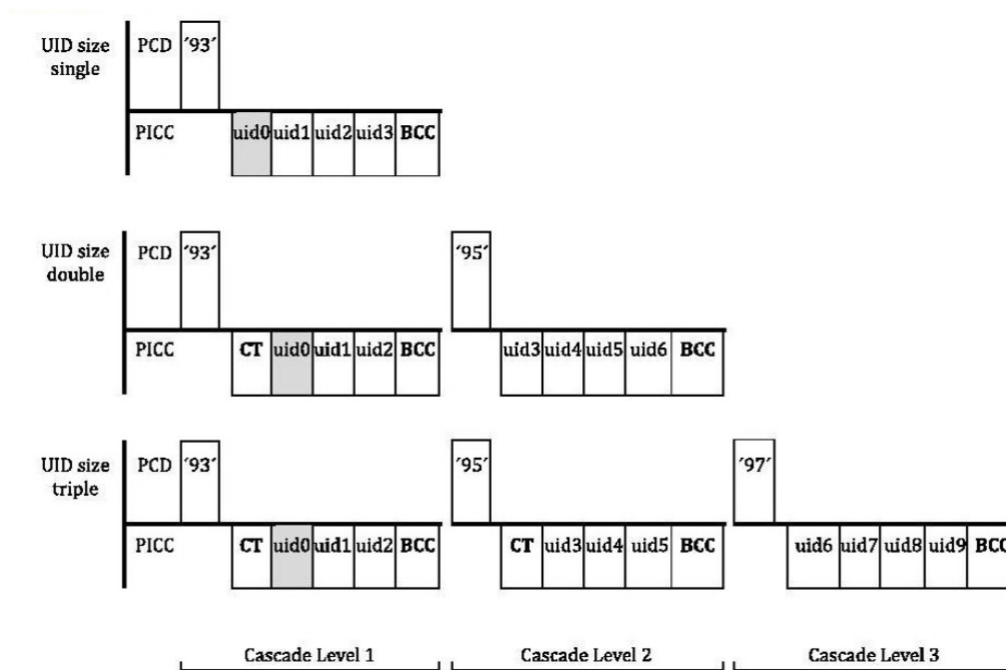


Figura 1.9: Niveles cascadas para UIDs simples, dobles y triples [10]

ISO/IEC 14443-4: Protocolo de Transmisión

La mayor parte de esta subsección es común para PICCs tipo A y B, sin embargo hay dos comandos extras para terminar activando el protocolo para PICCs de tipo A. Estos comandos son RATS y PPS. El PCD debería enviar un RATS como el primer comando después de que una PICC entra en el estado ACTIVE o ACTIVE*. En el RATS el PCD asigna un CID (Card Identifier) a la PICC, esto es un identificador que el PCD puede usar para direccionar un mensaje a una PICC en particular, así que el CID debe ser único para cada PICC activa. En diferencia al UID, el CID solo tiene cuatro bits y son asignados al momento de la inicialización y no en la de fabricación. Este CID es válido sólo hasta que la PICC es desactivada. La respuesta al RATS es la ATS lo que contiene información sobre la capacidad de la PICC, por ejemplo las tasas de bits y el tamaño máximo de una trama que la PICC puede recibir. El comando PPS es opcional, y solo puede ser enviado inmediatamente después de que el PCD recibe la ATS. El PPS permite al PCD configurar las tasas de bits de comunicaciones en cada dirección.

La norma define dos formatos de bloques: estándares y aumentados. Bloques estándares están enviados en tramas estándares con el CRC16 presente. Bloques aumentados

comienzan con el largo del bloque, y terminan con un CRC32, y el campo INF contiene códigos Hamming para la corrección de errores. La [Figura 1.10](#) muestra los dos. Comunicaciones comienza con bloques estándares hasta que el PCD configura lo contrario.

Prologue field				Information field	Epilogue field
PCB	[CID]	[NAD]		[INF]	CRC16
1 byte	1 byte	1 byte			2 bytes

Length field	Prologue field			Information field	Epilogue field
LEN	PCB	[CID]	[NAD]	[INF]	CRC32
2 bytes	1 byte	1 byte	1 byte		4 bytes

Figura 1.10: El formato de un bloque estándar (arriba) y uno aumentado (abajo) [11]

Hay tres tipos de bloques: I (Information), R (Receive Ready) y S (Supervisory). El campo PCB indica que tipo de bloque es. El CID es la dirección de la PICC y solo está presente si la PICC lo soporta. El campo CID en respuestas también contiene dos bits con información sobre el nivel de potencia recibido, el PCD puede usar esa información para controlar la potencia transmitida por el campo. La NAD (Node Address) permite una PICC tener más de una aplicación, direccionado por este campo, también solo está presente si la PICC la soporta. Finalmente el campo INF contiene la información del bloque, y solo es presente por bloques tipos I y S.

Bloques-S son usados para información de control. Hay tres comandos definidos:

S(WTX)

Waiting Time eXtension. Si la PICC no está lista para responder a una solicitud en el tiempo permitido, puede responder con un S(WTX) pidiendo más tiempo.

S(DESELECT)

Este comando es enviado por el PCD cuándo quiere desactivar la PICC.

S(PARAMETERS)

Este comando está usado para leer o setear la configuración de la PICC. Por ejemplo, para cambiar la tasa de bits, o cambiar entre bloques estándares y aumentados.

Bloques-I son usados para transmitir información al nivel de aplicación. El protocolo de la aplicación no está definido en esta norma. Estos bloques pueden ser encadenados para permitir el envío de un mensaje más grande que el soportado por el destino, partiendo el mensaje en partes de tamaños soportados.

Bloques-R son usados para reconocer la recepción de un bloque: R(ACK) o indicar errores: R(NAK). En el caso de bloques-I encadenados un R(ACK) es enviado para pedir la siguiente parte del mensaje. También el PCD puede enviar un R(ACK/NAK) para pedir la PCCC retransmita su última respuesta, esto puede ser usado para recuperar de errores. Finalmente el PCD puede enviar un R(NAK) para verificar la presencia continua de la PCCC.

Descripción General del Proyecto Marco

Este trabajo se enmarca en un proyecto que tiene como objetivo general desarrollar un circuito integrado capaz de tomar muestras de un sensor MOSFET de radiación y transmitirlos a un dispositivo externo mediante un protocolo propietario implementado encima de la norma ISO/IEC 14443A. Este proyecto marco requiere cinco bloques como se muestra en la [Figura 2.1](#):

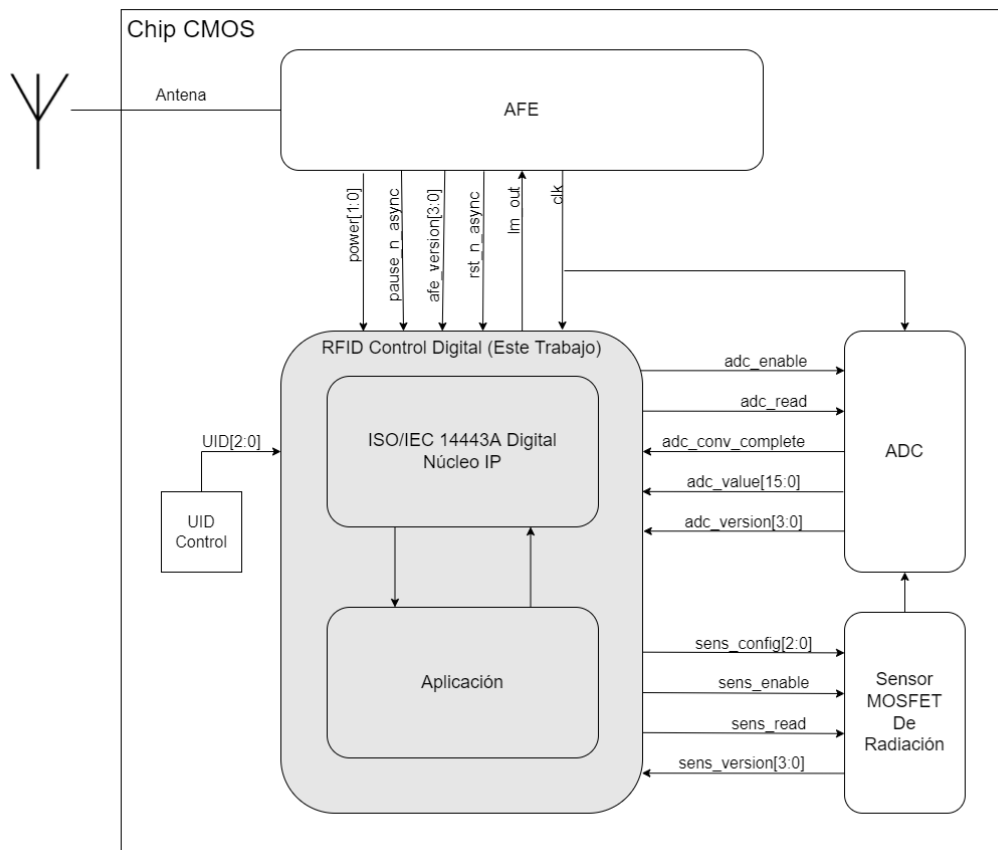


Figura 2.1: Diagrama general del proyecto marco y su relación con el bloque a diseñar en esta Tesis.

El Analogue Front End (AFE)

Es la parte analógica de la norma ISO/IEC 14443A-2.

El sensor MOSFET de radiación

Desarrollado por el Laboratorio de Física de Dispositivos-Microelectrónica (LFDM) de la facultad de ingeniería de la UBA.

El Conversor Analógico Digital (ADC, por sus siglas en inglés)

Encargado de convertir la salida analógica del sensor a una señal digital.

El RFID Control Digital

Es el objetivo de esta Tesis.

El UID Control

Permite asignar la parte variable del UID de la PICC.

Algunos de estos bloques no se han desarrollado todavía, por lo que es necesario asumir las especificaciones de los mismos.

El AFE es responsable de:

- Recibir y regular potencia desde el campo electromagnético. La implementación de esta tesis se realiza considerando la tecnología de fabricación de XFAB 180 nm, la cual utiliza celdas estándares que requieren una tensión de 1.8 V, por lo que el AFE debe producir esa tensión, y mantenerla durante las pausas.
- Recuperar la señal de reloj a partir de la portadora. La dificultad en este punto reside en el hecho de que el PCD envía información a la PICC utilizando pausas en la portadora, por lo cual se espera que el reloj se detenga durante estas pausas. Es posible utilizar un PLL para producir un reloj continuo. El circuito de esta tesis es elaborado con habilidad para funcionar con un reloj continuo o con un reloj que se detenga durante las pausas. En la [Sección `sequence.decode`](#) se detalla el número máximo de flancos del reloj que es posible perder conservando una correcta decodificación de las secuencias.
- Manejar la señal de reset (activa baja). Los demás bloques se deben mantener en un estado de reset hasta que la tensión de alimentación sea estable.
- Detectar las pausas enviadas desde el PCD.
- Permitir la transmisión de respuestas al PCD mediante un modulador de carga. El modulador debería ser manejado directamente desde una entrada del bloque, lo cual sería conectado a la salida *lm_out* del bloque RFID Digital Control. La señal se obtiene como el AND lógico de la codificación Manchester y la subportadora.

- Generar una señal indicando el nivel de potencia recibida a través del campo electromagnético (opcional). Si se tiene esta información, la misma puede ser enviada al PCD en el campo CID de mensajes de nivel protocolo. El PCD puede usarla para ajustar la intensidad del campo electromagnético.
- Proveer una salida de cuatro bits que indique la versión del hardware del AFE. Este valor forma parte de la información enviada al PCD como respuesta al mensaje de protocolo IDENTIFY.

El sensor de radiación tiene tres entradas: *sens_config[2:0]*, *sens_enable* y *sens_read*. Para leer el sensor, primero se debe establecer el valor de *sens_config[2:0]* de manera que se elija la configuración del sensor deseado por el usuario. Posteriormente se debe establecer *sens_enable* en '1' para activar el sensor. Finalmente después del tiempo deseado por el usuario, llevar *sens_read* a '1' configura al circuito del sensor en modo lectura para su muestreo. Por otro lado, el sensor tiene dos salidas, una analógica que está conectada al ADC, y una señal de cuatro bits que indica la versión del sensor. Este valor forma parte de la información enviada al PCD como respuesta al mensaje de protocolo IDENTIFY.

Por su parte el ADC, además de la entrada analógica proveniente desde el sensor, tiene otras dos entradas: *adc_enable* y *adc_read*. El flanco ascendente de la señal *adc_read* indica el momento de muestreo de la salida del sensor.. Respecto de las salidas, hay tres: *adc_conversion_complete*, *adc_value[15:0]* y *adc_versión[3:0]*. Cuando el ADC completa la conversión la señal *adc_conversion_complete* toma el valor '1' durante un ciclo del reloj. La señal *adc_value* debe ser estable antes del pulso en *adc_conversion_complete*.

El bloque de UID Control es responsable de especificar los tres bits más bajos del UID de la PICC. Este valor puede ser configurado mediante una memoria no volátil, pero en esta tesis la intención es configurarlo utilizando wire bonding o resistores pull up/down.

El objetivo de esta tesis es implementar el bloque del RFID Control Digital, que consiste en dos sub-bloques: Una implementación del parte digital de la norma ISO/IEC 14443A y un protocolo a nivel aplicación que permita la lectura del sensor y envíe esta información al PCD. Un requisito de este bloque es que la entrada de UID se mantenga estable mientras el bloque no esté en un estado de reset.

Implementación y Verificación

La implementación de todos los módulos y los bancos de prueba se escriben con el HDL (Hardware Description Language) SystemVerilog, el cual es definido en IEEE 1800 [6]. SystemVerilog es basado en Verilog, formalmente definido en IEEE 1364 [7]. SystemVerilog es conocido principalmente como un HDL para verificación debido a sus extensiones sustanciales a verilog en ese ámbito, por ejemplo en la adición de conceptos de programación orientados a objetivos (como clases), y aserciones. Además SystemVerilog tiene varias ventajas a Verilog para uso en síntesis [17], por ejemplo:

Enumeraciones (enum) y estructuras (struct)

Como en el lenguaje de programación C.

always_comb y always_ff

Estos permiten al ingeniero especificar su intención a implementar lógica combinatoria o secuencial respectivamente. Las herramientas pueden verificar que el circuito inferido cumple con esa intención, por ejemplo, que los bloques combinatorios no contienen latches.

Interfaces

Colecciones de señales que son frecuentemente usadas juntas para reducir la replicación de código.

La implementación de esta tesis está dividida en dos: 1) El núcleo IP genérico para ISO/IEC 14443A que es la lógica digital necesaria para recibir, decodificar y actuar sobre los mensajes definidos en la norma, y construir, codificar y transmitir las respuestas adecuadas. Este núcleo IP está dividido en tres partes principales, uno para cada parte de la norma (excepto ISO/IEC 14443-1). 2) El sistema de control del sensor y del ADC. La [Figura 3.1](#) muestra todos esos bloques, y el flujo de datos entre ellos.

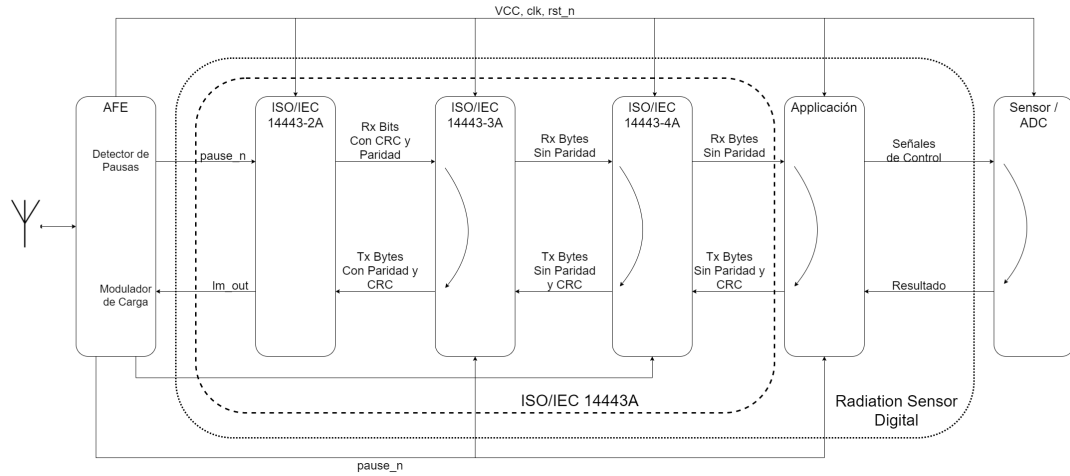


Figura 3.1: Los bloques principales de la implementación.

Interfaces

La norma IEEE de SystemVerilog define un interfaz como:

A su nivel más bajo, un interfaz es un conjunto nombrado de nets o variables. La interfaz es instanciada en un diseño y puede ser accedida por un puerto como un ítem sencillo; las nets o variables componentes [pueden ser] referidos cuando sea necesario. Una proporción significativa de un diseño verilog frecuentemente consiste en listas de puertos y listas de conexiones, los que son simplemente repeticiones de nombres. La habilidad de reemplazar un grupo de nombres con un nombre simple puede reducir significativamente el tamaño de una descripción y mejorar su mantenibilidad. [6, traducción mía]

Además de esas ventajas, una interfaz puede contener funcionalidad, sea para síntesis o verificación, por ejemplo se puede agregar aserciones que verifican el comportamiento de los nets internos en vez de tener que duplicar esas pruebas en cada sitio que la interfaz es usada. Una interfaz puede tener una o más modports, los que especifican las direcciones de las señales. Esos modports pueden ser usados en la lista de puertas por un módulo en vez de especificar cada señal y su dirección individualmente.

Este trabajo consiste en varios módulos que contienen uno o más sumideros para recibir

tramas desde otros módulos, y una o más fuentes para enviar tramas a otros módulos. Por ejemplo el módulo: **frame_decode** recibe tramas desde el módulo:

La *rx_interface* contiene:

- *soc*: Un indicador que indica el comienzo de una trama.
- *eoc*: Un indicador que indica el fin de una trama.
- *data*: Un bit / byte de la trama.
- *data_valid*: Un indicador que indica si los datos en *data* son válidos.
- *data_bits*: La cantidad de bits válidos. Por una interfaz configurada a bytes, esta señal permite la recepción de tramas cortas que tienen solo 7 bits, o de tramas anticolisión orientada a bits que pueden terminar con entre uno y ocho bits.
- *error*: Indica la detección de un error en la trama, por ejemplo: un bit de paridad equivocado.

La [Figura 3.2](#) muestra una simulación de la recepción de una trama corta con una *rx_interface* de bits, y su conversión a una serie de bytes. Arriba están las señales en una *rx_interface* de bits, *data_valid* está en ‘1’ siete veces durante la trama, indicando que el dato recibido es: 1,1,1,1,0,1,0. Debido a que el bit menos significativo es enviado primero, este serie representa 7'b0101111. Abajo están las señales en una *rx_interface* de bytes representando la misma trama. El dato tiene valor 7'b0101111 cuando *data_valid* tiene valor lógico igual a ‘1’.

La *rx_interface* incluye varias aserciones para verificar el comportamiento de las señales en la interfaz. Las aserciones consideran: Las señales son correctas en el estado de reset, los indicadores nunca son desconocidos, *soc* y *eoc* no están en ‘1’ en el mismo ciclo, y que solo están en ‘1’ por la duración de un solo ciclo del reloj a la vez.

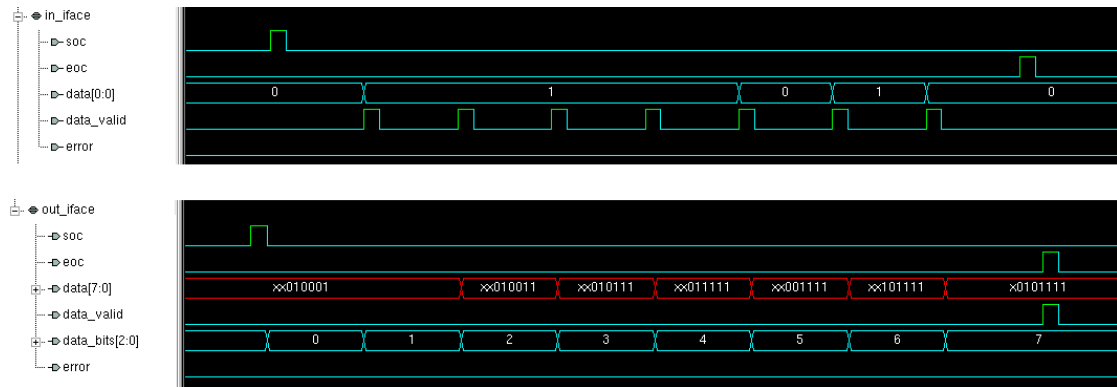


Figura 3.2: Una simulación de la recepción de una trama corta representada con una *rx_interface* de bits (arriba) y su conversión a una *rx_interface* de bytes (abajo).

La *tx_interface* contiene:

- *data*: Un bit / byte de la trama.
- *data_valid*: Un indicador que indica que *data* es válida.
- *data_bits*: La cantidad de bits válidos. Para una interfaz configurada a bytes, esta señal permite la transmisión de tramas de anticollisión orientada a bits, las que pueden comenzar con un byte parcial.
- *last_bit_in_byte*: Esta señal solo existe en interfaces de bits. Indica que el bit actual es el último bit en un byte. Es usado para conocer dónde agregar bits de paridad dentro de una serie de bits.
- *req*: El sumidero usa esta señal para pedir que la fuente envíe el siguiente bit / byte de la trama. La norma ISO/IEC 14443A-2 define la duración de bits como 128 ciclos de la portadora, por lo tanto esta señal es necesaria para limitar la tasa de envío de datos.

Cuándo una fuente está lista para transmitir una trama se establece el primer bit / byte de la trama en la señal *data* y fija *data_valid* en '1'. El sumidero puede utilizar la señal *data_valid* para ver si la fuente tiene data enviar. Después de haber leído la primera bit / byte de data, el sumidero fija la señal *req* en '1' durante un ciclo del reloj. La fuente detecta ese pulso y si hay más data a enviar, se actualiza la *data* con el siguiente bit / byte, dejando *data_valid* en '1'. Este proceso repite hasta que la fuente no tiene más data a enviar, y después del último pulso en la *req*, se fija *data_valid* en '0'. La [Figura 3.3](#) muestra una simulación de la transmisión de una trama de 16 bits con una *tx_interface* de bytes, y su conversión a una serie de bits. Arriba están las señales en la *tx_interface*

de bytes enviando la data 0xAE, 0x42. Abajo están las señales en una *tx_interface* de bits representando la misma trama.

La *tx_interface* también incluye tres aserciones que verifican: Las señales son correctas en el estado de reset, *req* es ‘1’ solo para la duración de un ciclo del reloj, y que *data*, *data_valid*, *data_bits* y *last_bit_in_byte* solo cambian estado cuando *data_valid* está ‘0’ o en los cuatro ciclos después de un pulso de *req*. La última aserción es para verificar que la fuente puede proveer datos cuándo es pedido con la señal *req* antes de que el sumidero la necesita.

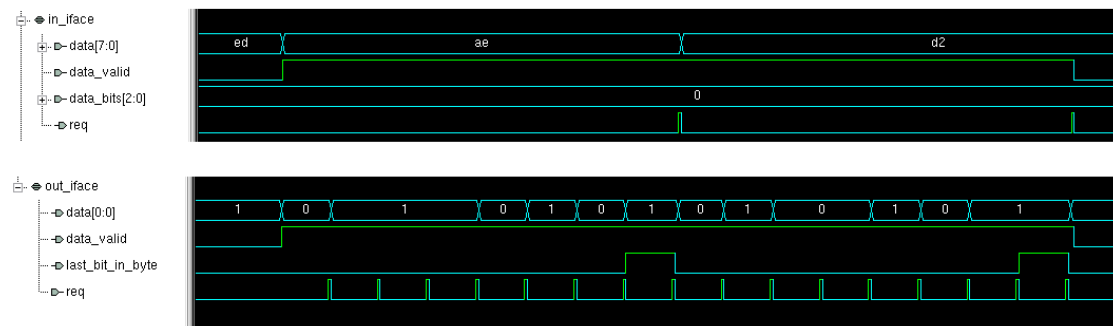


Figura 3.3: Una simulación de la transmisión de una trama de 16 bits representada con una *tx_interface* de bytes (arriba) y su conversión a una *tx_interface* de bits (abajo).

Marco de Verificación

Simulación y verificación es una parte muy importante del diseño digital. En un informe de 2020 sobre las tendencias de diseño y verificación de ICs y ASICs [4], los autores encontraron que en promedio más de 50 % del tiempo de un proyecto es utilizado en verificación. También encontraron que en promedio hay un ingeniero de verificación por cada ingeniero de diseño, además los ingenieros de diseño gastan aproximadamente la mitad de su tiempo verificando sus diseños.

En esta tesis se utiliza verificación funcional mediante simulaciones con la herramienta VCS de Synopsys. Las simulaciones son completamente automatizadas y son ejecutadas con un comando sencillo utilizando un Makefile, por ejemplo: “make serialiser_tb”.

UVM (Universal Verification Methodology) es una metodología de verificación funcional mediante un conjunto de clases de SystemVerilog. Casi 80 % de proyectos mundiales son

verificados con UVM [4]. Una de las ventajas principales de UVM es la modularidad. El código es partido en bloques separados facilitando la reutilización de los componentes en varios bancos de pruebas sin tener que duplicar código. La desventaja principal es la complejidad, requiere mucho código para armar un banco de prueba. UVM no es usada en esta tesis porque su implementación sería más compleja que el propio diseño. Sin embargo el marco de verificación está basado en las técnicas de UVM.

El proceso de verificación es estimular las entradas del DUT (Design Under Test) y verificar que las salidas son las esperadas. Frecuentemente no es posible verificar un diseño para todas las combinaciones y secuencias posibles de las entradas, por lo tanto es común usar estímulo aleatorio. Con estímulo aleatorio siempre hay el riesgo de no verificar parte del diseño, por la posibilidad de no elegir una de las combinaciones o secuencias de entradas necesarias para estimular esa parte del diseño. Una técnica para ayudar a mitigar esto, es: aleatorio restringido (constrained random), esta técnica permite la generación de estímulo aleatorio mediante constraints para restringir el estímulo a un rango interesante en particular. Por ejemplo, en vez de generar tramas completamente aleatorias, pueden ser limitadas: a tramas válidas, una colección de tramas en particular, o tramas con errores.

Una técnica para asegurar que todas las partes de un diseño son verificadas suficientemente, es generar informes de cobertura. La herramienta VCS está habilitada para generar, de forma automática, informes de cobertura de código con varias métricas, y un resultado total de qué proporción del diseño fue verificado. Estos informes pueden ser analizados por el diseñador para verificar cuáles partes de su diseño fue suficientemente estimulado y cuáles partes necesitan más trabajo. Las métricas de cobertura de código habilitadas en VCS son:

- Línea: Muestra las líneas del RTL que fueron ejecutadas.
- Condición: Muestra las sub expresiones booleanas que fueron evaluados a verdadero y falso. Por ejemplo en la declaración: `res = (A == 0) ? B : C`, el informe de cobertura indicará si la expresión `"A == 0"` fue evaluado al menos una vez a verdadero y al menos una vez a falso.
- Cambio de Estado (Toggle): Muestra cuáles señales y puertas cambiaron de estado en las dos direcciones.
- Branch: Muestra cuáles branches fueron tomados.
- FSM (Finite State Machine): Muestra cuáles estados en un FSM fueron utilizados, y las transiciones entre ellos.

- Aserción: Muestra cuáles aserciones fueron: ejecutadas, aprobadas y falladas, y cuántas veces por cada uno.

El marco de verificación es implementado con varios componentes, la mayoría son clases de SystemVerilog. Los componentes pueden ser divididos en siete grupos distintos:

- Transacciones: Una transacción representa una trama, puede ser una trama de bits, de bytes o de secuencias (como definido en ISO/IEC 14443-2).
- Controladores: Un controlador envía una transacción sobre una interfaz.
- Monitores: Un monitor monitoriza una interfaz, y construye transacciones representando las tramas detectadas.
- Generador de transacciones: Genera transacciones de bytes para representar las tramas definidas en la norma. También puede generar transacciones aleatoriamente.
- Convertidor de transacciones: Produce una transacción en un formato desde una transacción en otro formato. Por ejemplo puede convertir una transacción de bytes a una transacción de bits, opcionalmente agregando los bits de paridad.
- Secuencias: Código compartido para verificar diseños que reciben tramas, actúan sobre ellos, y generan las respuestas. Por ejemplo: cuándo el DUT está en el estado READY, responde a un SELECT con un SAK con los valores esperados, y se transiciona al estado ACTIVE. Ese ejemplo es una prueba utilizada para verificar el comportamiento de cuatro módulos: **initialization**, **iso14443_3a**, **iso14443a**, **radiation_sensor_digital_top**.
- Otros: Este grupo incluye: modelos de los bloques analógicos, una clase para guardar la dirección de una PICC (UID, CID y NAD), y unas interfaces que solo están usadas para verificación.

Transacciones

Controladores

Monitores

Generador y Conversores de Transacciones

Secuencias

Otros

Modelos Analógicos

Estructura de los Bancos de Pruebas

ISO/IEC 14443A núcleo IP

ISO/IEC 14443-2A

subcarrier

bit_encoder

tx

sequence_decode

iso14443_2a

ISO/IEC 14443-3A

frame_decode

31

deserialiser

FDT

CRC_A

crc_control

Síntesis y Place & Route

Síntesis

Preparación de librerías

Design Planning

Place & Route

LVS / DRC

Resultados y Conclusiones

Recomendaciones para Trabajos Futuros

Bibliografía

- [1] Fabricio Alcalde, Octavio Alpago y José Lipovetsky. «CMOS design of an RFID interface compatible with ISO/IEC-14443 type a protocol». En: (2014). DOI: [10.1109/EAMTA.2014.6906077](https://doi.org/10.1109/EAMTA.2014.6906077).
- [2] M. Garcia-Inza y col. «6MV LINAC characterization of a MOSFET dosimeter fabricated in a CMOS process». En: *Radiation Measurements* 117 (2018), págs. 63-69. ISSN: 1350-4487. DOI: [10.1016/j.radmeas.2018.07.009](https://doi.org/10.1016/j.radmeas.2018.07.009). URL: <https://www.sciencedirect.com/science/article/pii/S1350448718301586>.
- [3] Mariano Garcia-Inza y col. «Radiation Sensor Based on MOSFETs Mismatch Amplification for Radiotherapy Applications». En: *IEEE Transactions on Nuclear Science* 63.3 (2016), págs. 1784-1789. DOI: [10.1109/TNS.2016.2560172](https://doi.org/10.1109/TNS.2016.2560172).
- [4] Wilson Research Group. «2020 Wilson Research Group functional verification study – IC/ASIC functional verification trend report». En: (2020). URL: https://uobdv.github.io/Design-Verification/WilsonResearchGroupFunctionalVerificationStudy2020-WRGFV-Study/ic-asic-trend-report_2020-wilson-research-verification-study_hfoster.pdf.
- [5] ICRP. *ICRP publication 86. Prevention of Accidents to Patients Undergoing Radiation Therapy*. Annals of the ICRP. London, England: Elsevier Health Sciences, 2000.
- [6] *IEEE Standard for SystemVerilog–Unified Hardware Design, Specification, and Verification Language*. en. Standard IEEE 1800-2009. 2009. URL: <https://standards.ieee.org/ieee/1800/3989/>.
- [7] *IEEE Standard Verilog Hardware Description Language*. en. Standard IEEE 1364-2001. 2001. URL: <https://standards.ieee.org/ieee/1364/2052/>.
- [8] Dr. Ing. Mariano Garcia Inza. «Diseño de sensores CMOS de radiación ionizante». 2019.
- [9] ISO Central Secretary. *Identification cards – Contactless integrated circuit cards – Proximity cards – Part 2: Radio frequency power and signal interface*. en. Standard ISO/IEC 14443-1:2016. International Organization for Standardization, International Electrotechnical Commission, 2016. URL: <https://www.iso.org/standard/66288.html>.

- [10] ISO Central Secretary. *Identification cards – Contactless integrated circuit cards – Proximity cards – Part 3: Initialization and anticollision*. en. Standard ISO/IEC 14443-3:2016. International Organization for Standardization, International Electrotechnical Commission, 2016. URL: <https://www.iso.org/standard/70171.html>.
- [11] ISO Central Secretary. *Identification cards – Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol*. en. Standard ISO/IEC 14443-4:2016. International Organization for Standardization, International Electrotechnical Commission, 2016. URL: <https://www.iso.org/standard/70172.html>.
- [12] ISO Central Secretary. *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures*. en. Standard ISO/IEC 13239:1997. 1997. URL: <https://www.iso.org/standard/21474.html>.
- [13] Arana Leandro Javier. «Interfaz RFID para dosimetría MOS de aplicación médica». Universidad de Buenos Aires, 2018.
- [14] Gerald J. Kutcher y col. *Comprehensive QA for Radiation Oncology*. Inf. téc. 1994. DOI: [10.37206/45](https://doi.org/10.37206/45).
- [15] *Quality Assurance in Radiotherapy*. Geneva, Switzerland: World Health Organization, dic. de 1988. ISBN: 9241542241.
- [16] CampusIDNews Staff. *IS THE DEBATE STILL RELEVANT? An in-depth look at ISO 14443 and its competing interface types*. 2003. URL: <https://www.campusidnews.com/is-the-debate-still-relevant-an-in-depth-look-at-iso-14443-and-its-competing-interface-types/>.
- [17] Stuart Sutherland y Don Mills. «Synthesizing SystemVerilog – Busting the Myth that SystemVerilog is only for Verification». En: (2013). URL: https://sutherland-hdl.com/papers/2013-SNUG-SV_Synthesizable-SystemVerilog_paper.pdf.
- [18] E.G. Villani y col. «A monolithic 180 nm CMOS dosimeter for wireless In Vivo Dosimetry». En: *Radiation Measurements* 84 (2016), págs. 55-64. ISSN: 1350-4487. DOI: [10.1016/j.radmeas.2015.11.004](https://doi.org/10.1016/j.radmeas.2015.11.004). URL: <https://www.sciencedirect.com/science/article/pii/S1350448715300755>.