

Implementación de modelos de fotodiodos en lenguajes de descripción de hardware de señal mixta

Implementing photodiode models in languages for the description of mixed signal hardware

Recibido: 15 abril 2013, aceptado: 09 septiembre 2013

Carlos Bran*, B. Blanco-Filgueira y P. López**

Resumen

El rápido escalamiento de la tecnología CMOS y el desarrollo de procesos CIS (CMOS Image Sensor) optimizados para el diseño de sensores de imagen, en nodos tecnológicos de 90nm, 65nm e inferiores, incrementa grandemente la complejidad de los modelos debido a los efectos físicos que ocurren a esta escala, lo que crea la necesidad de implementar celdas compactas con dichos efectos descritos completamente, para los procesos de simulación usando herramientas CAD. En este artículo se muestra el proceso de implementación de una celda completa de un fotodiodo que incluye los efectos físicos fundamentales presentes en áreas de estas dimensiones, haciendo uso de lenguajes de descripción de hardware; evaluando luego la celda en un circuito típico de sensor de píxel activo con CMOS (APS), simulando el circuito en un entorno mixto que incorpora modelos de dispositivos activos en otros lenguajes para comparar las tendencias de las variables eléctricas del fotodiodo con el comportamiento típico de este componente en los circuitos APS.

Palabras Clave: Sensor de imagen, Fotodiodo, Modelado Simulación a nivel de circuito, Verilog-AMS

Abstract

The rapid scaling of CMOS technology and CIS (CMOS Image Sensor) process development optimized for image sensor design in 90nm technology nodes , 65nm and lower, greatly increases the complexity of the models due to the physical effects that apply at this scale, which creates the need to implement compact cells that fully describe these effects for simulation processes using CAD tools . This article shows the process of implementing a complete cell photodiode, including fundamental physical effects present in areas of these dimensions, using hardware description languages, then assessing cell in a typical sensor circuit pixel active with CMOS (APS), simulating the circuit in a mixed environment that includes active device models in other languages to compare trends in the electrical variables of the photodiode with the typical behavior of this component in APS circuits.

Keywords: image sensor, photodiode, simulated modeling at the circuit level, Verilog-AMS

* Instituto de Investigación e Innovación en Electrónica, Universidad Don Bosco. cbran@udb.edu.sv

** Departamento de Electrónica y Computación, Universidad de Santiago de Compostela

Introducción

El escalamiento de los sensores de imagen implica una reducción significativa del área de captación de las unidades de luz, lo que obliga a que los efectos ópticos de la captación lumínica en el fotodiodo sean modelados de forma completa incorporando no solo los efectos de la luz incidente directa sino también otros que para áreas mucho más grandes se consideran despreciables; por lo tanto los modelos para los fotodiodos en nodos tecnológicos de 90nm o inferiores han ampliado su complejidad de forma significativa. Una dificultad para poder simular estos efectos en circuitos básicos de procesamiento de píxeles, usando herramientas CAD, es la incorporación de las ecuaciones que describen el modelo del fotodiodo en celdas que puedan integrarse con otros componentes activos y pasivos para formar las unidades de sensores CMOS de píxel activo (APS), las cuales representan la unidad básica de procesamiento de imagen de cualquier dispositivo dedicado a esta función.

La tecnología de los APS ha demostrado características de ruido, eficiencia y rendimiento comparable a la tecnología CCD, además de presentar mayores funcionalidades y mucho menor consumo de potencia, por lo que esta tecnología será la base de unidades que integren además de la captura de cada píxel, los circuitos de temporización, control, conversión analógico a digital, etc; con lo que se consigue integrar la cámara en el mismo chip; esta consideración demanda que al crear celdas que describan los efectos ópticos completos en los fotodiodos, éstas tengan la capacidad de poder simularse en entornos de señal mixta, es decir que impliquen circuitos analógicos y digitales, cuyos modelos de componentes pueden estar descritos en múltiples lenguajes, como spice o spectre y simulados de forma integrada en la herramienta CAD; lo que demanda que la descripción de la celda deba desarrollarse en un lenguaje que permita esta habilidad, por lo que la necesidad de lenguajes descriptivos de señal mixta representan la mejor opción para este trabajo, así lenguajes como VHDL-AMS y Verilog-AMS son las opciones naturales para la descripción de estos modelos.

Los trabajos relacionados con la descripción de modelos de fotodiodos en lenguajes hardware, son reducidos y en muchos de los casos se enfocan en la interconexión óptica de las celdas además de otros fenómenos, describiendo el código AMS, pero sin mostrar el proceso de implementación con la herramienta CAD y los resultados del proceso de simulación; por otro lado otros trabajos hacen uso modelos reducidos de los efectos ópticos del fotodiodo, sin consideraciones dimensionales más complejas que permitan simularlos con componentes de nodos tecnológicos particulares, así solo muestran el despliegue del modelo del fotodiodo en lenguajes AMS de forma simple, con pocos parámetros y probado en circuitos que solo incluyen elementos pasivos; por otro lado resulta de gran importancia la habilidad de reutilización de la celda de forma independiente del circuito donde la mayoría de los factores que determinan el comportamiento del modelo puedan ser editables fácilmente.

El trabajo se dividirá en cinco partes. La primera hará una descripción del lenguaje utilizado para la creación de la celda, sus características, estructura y la justificación de su selección; la segunda se enfocará en las herramientas CAD donde se puede implementar la celda y los pasos para su creación, en la tercera parte se usará la celda del fotodiodo para implementar un circuito básico APS, específicamente en la arquitectura estándar 3T-APS, describiendo su proceso de implementación de forma completa; en la cuarta parte se presentarán los resultados obtenidos para distintas dimensiones de área y longitudes de onda, comparando

su tendencia con la respuesta típica de estos circuitos, en la última parte se agregarán algunas conclusiones y consideraciones finales.

HDL-AMS

Los lenguajes de descripción de hardware HDL (Hardware Description Language), son herramientas que nos permiten describir las interconexiones, características y comportamientos de los circuitos, como una alternativa a los diagramas esquemáticos; la principal cualidad de estos lenguajes es su independencia del hardware que describen, potenciando la modularidad y facilitando la técnica de diseño top-down. Originalmente estos lenguajes se usaban para describir bloques digitales, en la actualidad se ha ampliado a la extensión AMS (Analog and Mixed Signals), con la que se pueden describir no solo bloques analógicos sino cualquier otro sistema conservativo, lo que amplía las posibilidades de descripción de sistemas naturales más allá de sistemas puramente eléctricos, además de permitir la simulación del comportamiento de bloques combinados que incluyan módulos digitales, analógicos, mecánicos, térmicos, etc.

Existen múltiples lenguajes HDL, propietarios y estándares como VHDL, Verilog, ABEL, etc; sin embargo los dos que cuentan con extensiones AMS son VHDL y Verilog, los cuales son la elección natural para describir modelos de fotodiodos.

VHDL-AMS es el lenguaje definido por IEEE (ANSI/IEEE 1076-1993), el cual es ampliamente usado en diseño de circuitos digitales, soportado por múltiples herramientas de diseño. Sin embargo, su extensión para bloques analógicos y de señal mixta, es ligeramente menos utilizada para el diseño de circuitos. Por otro lado Verilog-AMS es un lenguaje que fue desarrollado para ser similar en su sintaxis a C, con el propósito de acelerar su aceptación, además de que las extensiones del mismo fueron desarrolladas en partes, así la primera extensión Verilog-A daba soporte a la descripción de bloques analógicos y posteriormente se incluyó la extensión AMS para el soporte a circuitos de señal mixta; al igual que VHDL, Verilog fue estandarizado por IEEE (IEEE 1364-1995) (IEEE 1364-2001-5), con lo que también está disponible en múltiples herramientas de diseño y cuenta con una mayor utilización en el desarrollo de diseños de bloques analógicos y de señal mixta. Si bien ambos lenguajes son comparables en la descripción de los modelos de fotodiodos se optó por Verilog-AMS, por su mayor utilización, en los entornos de diseño.

Verilog-AMS constituye un lenguaje unificado con estructuras semánticas para bloques tanto digitales como analógicos, separando la descripción de cada módulo de la descripción de su comportamiento y de la interconexión de éstos, teniendo cuidado de solo asociar las señales analógicas y mixtas a los nodos, no así las señales puramente digitales. Para la descripción de componentes monolíticos como el fotodiodo no es necesario describir interconexiones, por lo que será suficiente con la declaración del modulo en sí y la especificación de su comportamiento, que es dónde se implementará el modelo, por lo que la estructura de la descripción del fotodiodo se basará en el siguiente formato:

```
include "Incluir los archivos de cabecera de constantes"  
include "Incluir los archivos de cabecera de disciplinas"  
module <nombre de la celda> (<Nombre de los pines>);  
// Declaracion de los parámetros editables  
parameter <parámetros editables>;
```

```

...
// Declaración de las constantes no incluidas en el archivo de cabeceras, o parámetros no
editables.
    real <nombre de la constante>;    ...

// Establecimiento de la naturaleza de los pines de la celda, los que pueden ser eléctricos,
mecánicos, térmicos, etc. además del sentido de su interacción con el módulo, así ser de
entrada, salida o entrada-salida
    inout <Nombre de los pines>;
    electrical <Nombre de los pines>;
// Descripción del comportamiento del módulo
                                                                    analog begin
                                                                    <sintaxis de las ecuaciones que describen el
comportamiento>
                                                                    ....
                                                                    end

endmodule

```

En negrilla se representan las palabras reservadas del lenguaje, la figura 1 muestra un ejemplo de código de descripción de una celda simple de fotodiodo.

```

//Verilog-AMS HDL for "vamslib", "Fotodiodo1" "verilogams"

`include "constants.vams"
`include "disciplines.vams"

module Fotodiodo1 (a,c);
    parameter real is=10f from (0:inf); // Corriente de saturación (A)
    parameter real r=0 from [0:inf];    // Resistencia serie (Ohms)
    parameter real q=1.6e-19;
    parameter real Popt=0; // Potencia de iluminacion <175000>
    parameter real Tx=0.6;
    parameter real lambda=450e-9; //Longitud de onda entre 450 y 700nm
    parameter real h=6.63e-34;
    parameter real v_luz=3e8;
    parameter real area=6.25e-12; // Area del fotodiodo

    inout a, c;
    electrical a, c;

    analog
        if (V(a,c) > 0)
            I(a,c) <+ is*(limexp((V(a,c) - r*I(a,c))/$vt) - 1);
        else
            I(a,c) <+ -1.0*(area*q*Popt*Tx*lambda)/(h*v_luz);
endmodule

```

Figura 1: Código Verilog-AMS de un fotodiodo simple

Los modelos completos de fotodiodos descritos presentan ecuaciones que describen su comportamiento con expresiones más complejas, las cuales implican más presión sobre los métodos numéricos que las resuelven, por lo que es necesario hacer ajustes en los márgenes de tolerancia absoluta que controlan los ciclos iterativos de estos métodos, así para que el sistema tenga solución es necesario modificar el archivo de cabecera original donde se declaran estos márgenes; el parámetro que fue necesario modificar es el **“abstol”**, el cual especifica la medida de tolerancia que usa el simulador para determinar cuando el cálculo de un potencial o flujo ha alcanzado la convergencia, el valor de esta tolerancia está en el rango de 1000 a 1000000 veces más pequeña que el valor típico de la señal más grande que la celda procesará; este parámetro y otros incluidos en el archivo de cabecera “disciplines.vams” se vuelven críticos al trabajar con nodos tecnológicos de dimensiones muy pequeñas, sin embargo los valores por defecto funcionan muy bien para nodos más grandes. Esta necesidad se podrá identificar cuando la herramienta de simulación de código envíe mensajes de que no es posible resolver la celda porque se han superado o no se han alcanzado los márgenes de tolerancia especificados en dicho archivo de cabecera.

Por otro lado, es importante considerar que los métodos numéricos, usados para el cálculo del comportamiento **“analog”**, pueden hacer que las herramientas no alcancen la convergencia, por lo que es conveniente separar cálculos de variables o expresiones en bloques previos y dejar solo las ecuaciones específicas que describen el comportamiento en este bloque; para esos cálculos previos puede hacerse uso de bloques de tipo **“initial”**, para que el método numérico los trate como procesos previos a la descripción del comportamiento el cual se desarrolla en tiempo real en los procesos de simulación.

Herramientas CAD

Existen múltiples herramientas que soportan celdas de componentes creadas en Verilog o VHDL, sin embargo son bastante reducidas las que soportan la extensión AMS y que además permiten combinar las celdas con otros dispositivos descritos en lenguajes distintos, como hspice o spectre, por tanto las herramienta evaluadas únicamente fueron CADENCE y SMASH que cumplían con estos requerimientos.

SMASH es una herramienta desarrollada por Dolphin Integration que permite ejecutar simulaciones multi-nivel de circuitos de señal mixta, lo que implica que no solo se limita al nivel de modelado, sino que también puede manejar niveles de comportamiento, eléctrico, estructural y funcional; además del soporte de múltiples lenguajes de descripción de las celdas que incluyen VHDL-AMS, Verilog-AMS, SPICE y ABCD (Lenguaje-C), este último propietario pero con la capacidad también de describir comportamientos de celdas, cuenta además con un entorno gráfico para el soporte de los circuitos esquemáticos llamado SLED, el cual automáticamente se integra con SMASH para los procesos de diseño. Por último esta herramienta también cuenta con conectores para poder acoplarse con CADENCE y garantizar la portabilidad de las celdas entre ambos entornos. Una característica importante es la poca demanda de recursos para su funcionamiento, ya que puede ejecutarse en una estación personal con todas sus características y soportar diseños complejos con una demanda relativamente baja de recursos.

CADENCE por otro lado es un conjunto de herramientas CAD para el diseño de circuitos, que incluye los entornos de soporte para esquemáticos además de los motores para simulación y desarrollo. Para el trabajo con el lenguaje de Verilog-AMS se requiere del componente AMS-Designer que brinda el entorno para la simulación de circuitos de señal mixta, además del uso de SimVision para el manejo de los gráficos obtenidos

de las simulaciones. Este entorno al igual que SMASH es muy completo, pero al usar múltiples componentes durante el proceso de simulación hace demandas mayores de recursos.

Para el trabajo se optó por CADENCE, principalmente por ser la herramienta de uso más extendida tanto en entornos industriales como académicos y que además permite integrar librerías completas de componentes para el nodo tecnológico de 90nm, las cuales no estaban disponibles en SMASH en el entorno de evaluación, sin embargo es una muy buena opción para el trabajo en las estaciones personales.

Para el desarrollo de las celdas usando Verilog-AMS en Cadence se llevan a cabo los siguientes pasos:

Creación de la celda con un editor en Verilog-AMS. Se puede trabajar con cualquier editor, copiando luego el código en el editor por defecto que proporciona CADENCE. Es importante asociarle o crear el símbolo hasta que el código Verilog-AMS no tenga errores sintácticos o lógicos.

Una vez no se presenten errores en el código Verilog-AMS, se procede a crear o editar el símbolo de la celda usando el editor de símbolos de Virtuoso.

A continuación se verifica la accesibilidad de los parámetros editables del código de la celda Verilog-AMS usando para ello un esquemático; es importante que cada vez que se desarrollen cambios en el código éstos se guarden usando el editor por defecto de CADENCE, de lo contrario los cambios no se asociarán al símbolo en cuestión.

Por último se puede hacer una prueba de la celda en un circuito sencillo, junto con elementos pasivos para verificar que ésta responde a las ecuaciones descritas en Verilog-AMS.

Con estos pasos se completa la creación de la celda la cual se puede agregar a una librería de trabajo nueva o a una ya existente para que el componente pueda ser utilizado para la creación de otras celdas o circuitos; es importante recordar que si se usa esta celda en cualquier circuito, el proceso de simulación deberá de ejecutarse con el AMS-Designer. Si se usan otros motores de simulación disponibles en CADENCE, éstos no podrán interpretar el código de dicha celda, devolviendo un error, al inicio del proceso.

Celda de prueba

El rápido escalamiento de las tecnologías CMOS y el desarrollo de procesos de fabricación optimizados para los sensores de imagen no ha estado acompañado de un estudio profundo de los fenómenos físicos que determinan el comportamiento de los fotodiodos en estos nodos tecnológicos. Desde el punto de vista industrial, la existencia de modelos de comportamiento de estos dispositivos, traducidos a un lenguaje de simulación hardware, es esencial para trabajar con herramientas de simulación CAD para el desarrollo de nuevos productos.

En la literatura encontramos diversos modelos eléctricos de fotodiodos, pero no abundan modelos de comportamiento que permitan estudiar la dependencia con parámetros que pueden ser determinantes como el tamaño, la potencia lumínica o los asociados a una determinada tecnología.

En este trabajo se probaron diferentes modelos en lenguaje Verilog-AMS para fotodiodos de unión p-n+. La celda de test utilizada se corresponde con una arquitectura de píxel estándar en sensores de imagen CMOS denominada 3T-APS. Ésta consiste en un fotodiodo y tres transistores: uno de reset Q1 que actúa como interruptor, un seguidor de fuente que hace las veces de buffer Q2 y un selector de fila Q3, además de la carga activa Figura 2; la capacidad de este circuito de solo activarse durante el proceso de lectura, hace que el consumo de potencia se reduzca significativamente además de las mejoras del rendimiento del circuito respecto a los sensores pasivos,

La celda 3T-APS se implementa en el editor esquemático de CADENCE y haciendo uso de cualquiera de los fotodiodos descritos con Verilog-AMS; es importante contar con los modelos Spice de los transistores del nodo tecnológico donde se pretende hacer la simulación ya que será necesario incorporarlos, durante la configuración del proceso.

Para poder simular la celda completa se usa el editor jerárquico de CADENCE, al cual se le incorpora el plugin del AMS-Designer, además del esquemático del circuito y de los modelos de los transistores usados. Este paso es importante ya que el AMS-Designer no incorpora automáticamente los modelos spice de los transistores, por lo que debemos de forma manual indicar la ruta de los mismos. (Figura 3)

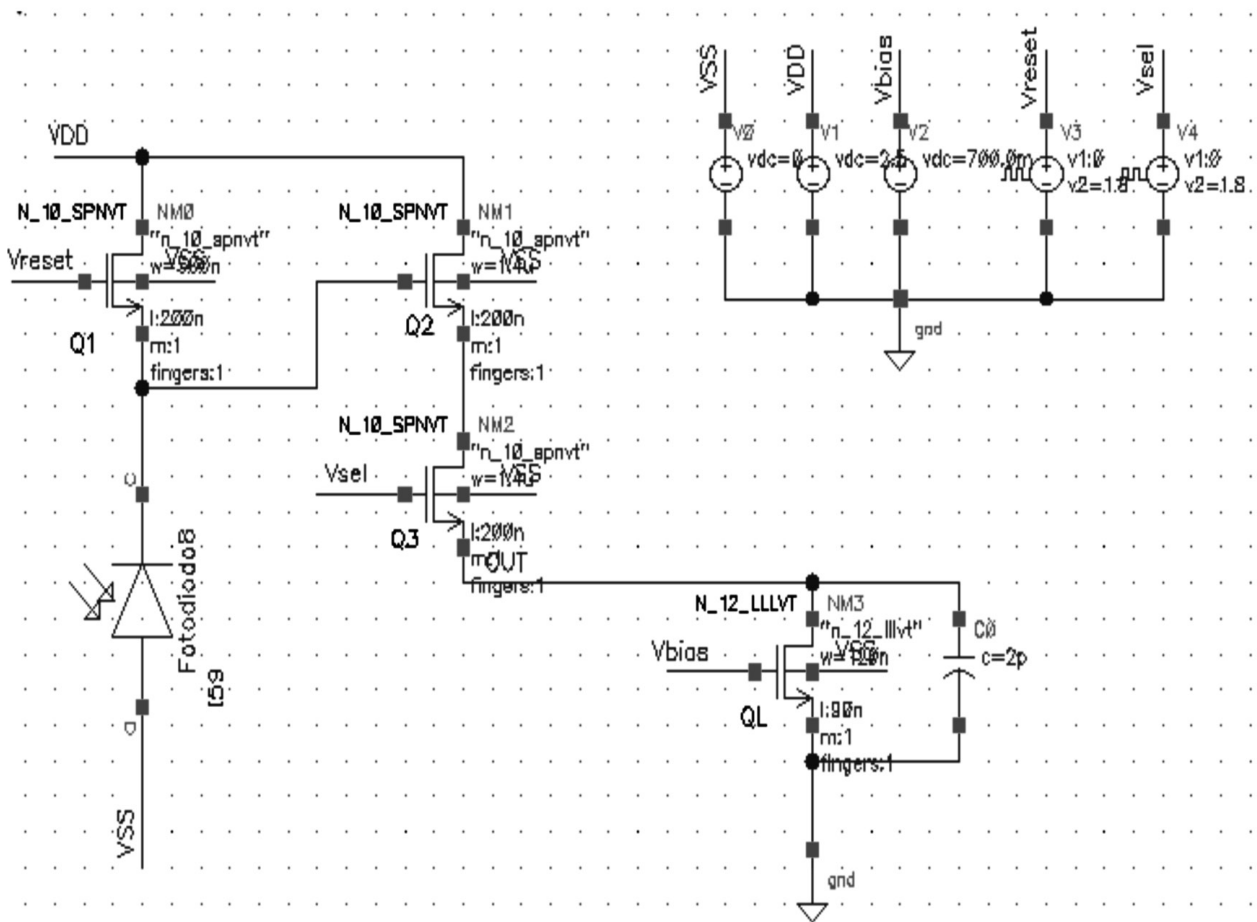


Figura 2: Esquemático de la celda de prueba 3T-APS

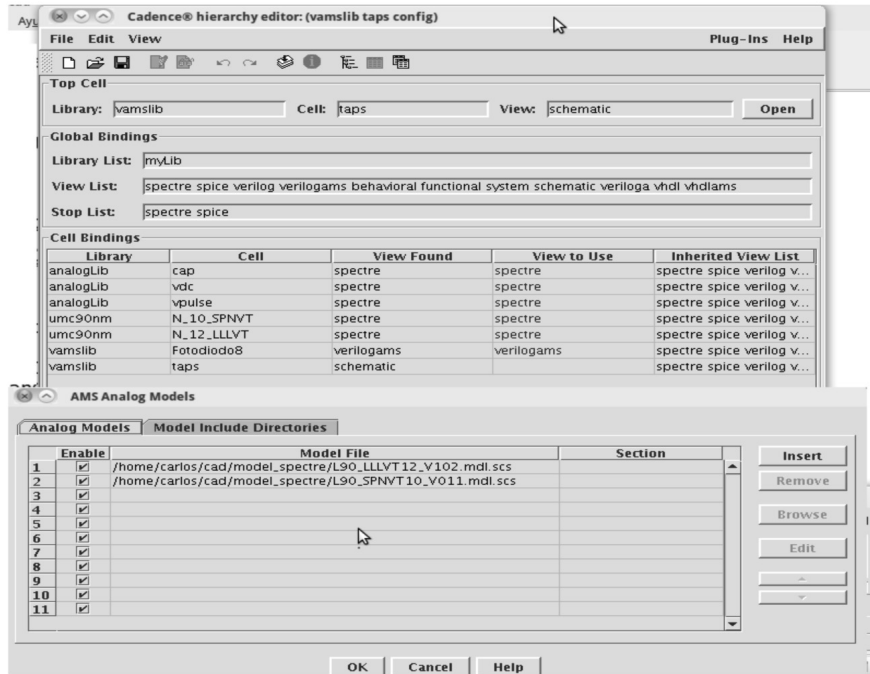


Figura 3: Editor Jerárquico de CADENCE

Por último se configuran las opciones de compilación donde se establece la ruta para el archivo de cabecera de hdl.var, el cual es necesario para incorporar las variables con las rutas que permitirán que el simulador pueda leer los códigos HDL, además se agregan las opciones de la simulación que se desea ejecutar y las variables que se visualizarán con el SimVision, entre otras cosas. La figura 4 muestra el gráfico de la salida de voltaje para una simulación con una longitud de onda (λ) y un área particular, junto con las señales de reset y selección, la pendiente del gráfico nos permite calcular la sensibilidad del fotodiodo, la cual varía con los valores de longitud de onda y área del mismo, por otro lado se pueden exportar los datos de los gráficos a archivos de texto, para el desarrollo de diversos análisis con otras herramientas.

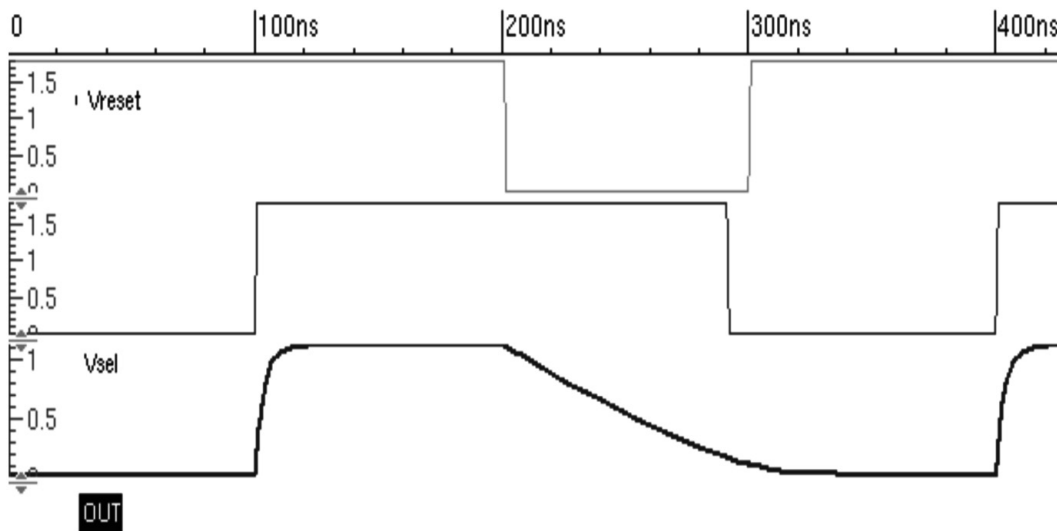


Figura 4: Salida para Lamda de 650nm y área de 608.4 fm cuadrados

Resultados

Los resultados que se presentan se basan en una celda 3T-APS que incluye un fotodiodo de unión p-n+ descrito en lenguaje Verilog-AMS. De este modo, se estudia la influencia del área del fotodiodo y de la longitud de onda de la fuente de iluminación en la salida del píxel. Así mismo, la descripción del fotodiodo permitiría el análisis de la respuesta del píxel en función de muchos otros parámetros. Las curvas presentadas solo reflejan la variación de estos dos parámetros área y longitud de onda, sin embargo las celdas descritas, incluyen los efectos de muchos más, por lo que pueden ser usados para otro tipo de estudios.

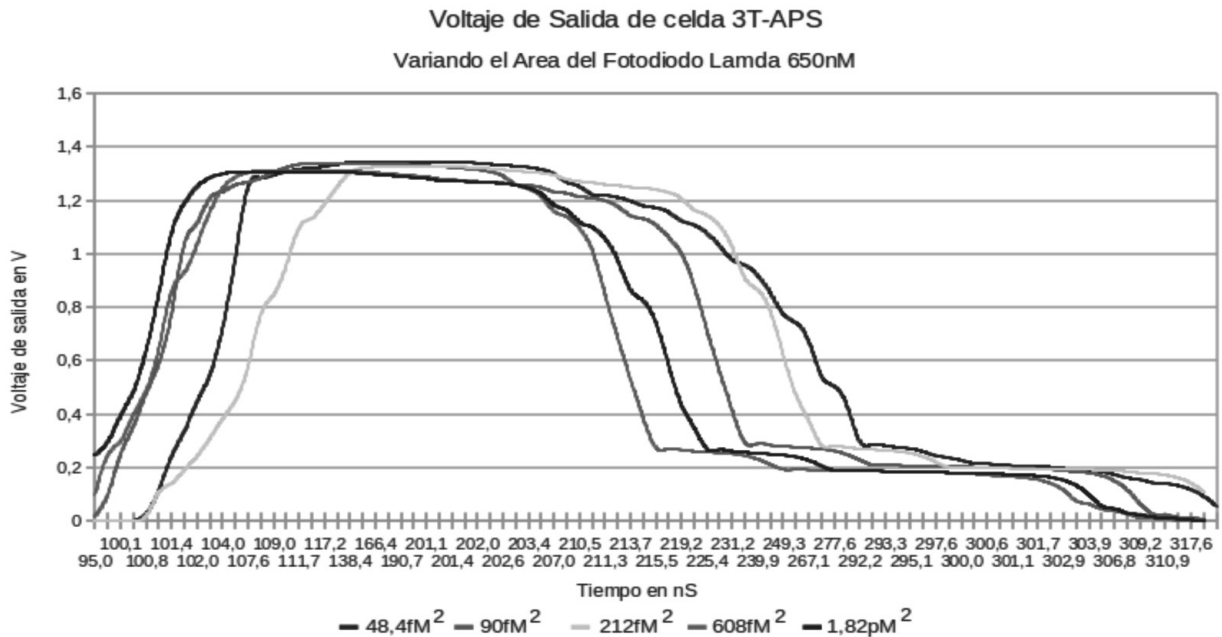


Figura 5: Curvas de respuesta de voltaje de salida al variar λ .

La figura 5 muestra la variación del voltaje de salida del circuito 3T-APS, variando el área de exposición del fotodiodo, desde 48.4fm² hasta 1.82pm², manteniendo la longitud de onda de la luz incidente en a 650nm y a una potencia de iluminación de 175kW por metro cuadrado, lo cual muestra como la celda responde a las pequeñas variaciones en área del fotodiodo.

La figura 6 muestra la variación del voltaje de salida del circuito 3T-APS, variando la longitud de onda de la iluminación aplicada al fotodiodo desde 550nm a 800nm, manteniendo fija el área del fotodiodo a 48.4fm² y a una potencia de iluminación de 175kW por metro cuadrado, lo cual destaca que la celda está respondiendo a la variación de este parámetro.

La celda descrita usando Verilog-AMS cuenta con 21 parámetros editables y dependiendo de la complejidad del modelo a describir éstos pueden aumentar, sin que sea un problema para el lenguaje o la herramienta, por lo que el ajuste de los diseños al simular estas celdas es a nivel muy fino.

Conclusiones

Los lenguajes de descripción de hardware describen adecuadamente el comportamiento de componentes que a escalas dimensionales muy pequeñas, resultan muy útiles para simular la respuesta de circuitos y su ajuste durante los procesos de diseño y síntesis de los mismos. La posibilidad que brindan los HDL de incluir múltiples parámetros, los cuales son resultados de la complejidad de la respuesta de los dispositivos a modelos más detallados, permite poder evaluar múltiples comportamientos de los circuitos y hacer los ajustes con una mayor granularidad en los productos que se diseñan. Por otra parte, aplicar a estas celdas análisis pseudoaleatorios como las técnicas de Montecarlo permitiría manipular muchos de estos parámetros para simular sus efectos en el comportamiento de los circuitos de los que forman parte.

Es recomendable hacer la síntesis y fabricación de la celda para poder hacer mediciones experimentales que permitan ajustar los modelos y su descripción en HDL, así como compactarlos y asegurar la confiabilidad de las celdas, para su aplicación en otros circuitos; por otro lado resulta interesante en trabajos futuros evaluar el soporte de las herramientas CAD para simular la interacción de variables no eléctricas acopladas, como el intercambio de potencia, óptica, térmica o mecánica entre otras, las cuales son soportadas por la extensión AMS y podrían facilitar el desarrollo de diseños tipo MEMS o NEMS.

Por último a medida que los diseños escalen en nodos tecnológicos mas pequeños, el modelado de los efectos físicos en los componentes será más detallado, por lo que el uso de lenguajes HDL-AMS se volverá el entorno más óptimo y rápido para el desarrollo de librerías de componentes para los procesos de simulación, análisis y diseño de circuitos electrónicos.

Referencias

- B. Blanco-Filgueira, P. López, J. Döge (1997) "Analytical Model For P-N Junctions Under Point Source Illumination"; *IEEE Trans. Electron Devices* 1997 vol. 44 no. 10 1689-1698.
- H.S. Wong (1996) "Technology and device scaling considerations for CMOS imagers" *IEEE Trans. Electron Devices* 1996 vol. 43 no. 12 2131-2142.
- Eric R. Fossum (1997) « CMOS Image Sensors:Electronic Camera-On-A-Chip". *IEEE Trans. Electron Devices* 1997 vol. 44 no. 10 1689-1698.
- Michiel De Wilde, Olivier Rits, Ronny Bockstaele (2003) "A circuit-level simulation approach to analyse system level behaviour of VCSEL-based optical interconnects" *VCSEL and Optical Interconnects* 2003 vol. 4942 no. 247.
- Kenneth S. Kundert, Olaf Zinke (2004) *The Designer`s Guide to Verilog-AMS*; Kluwer Academic Publishers, primera edición, ISBN 1-4020-8044-1.
- Accellera (2009) *Verilog-AMS Language Reference Manual; Accellera version 2.3.1* Junio 2009
- Dolphin (2011) *SMASH & SCROOGE User Manual; Accellera version 2.3.1* Enero 2011