

# **UNIVERSIDAD NACIONAL DE INGENIERIA**

FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA



## **ESTUDIO DE CONVERTORES ANALOGO DIGITAL ASINCRONO APLICADO A REDES DE SENSORES DISTRIBUIDOS PARA MEDICION DE TEMPERATURA EN RESERVAS NATURALES**

**INFORME DE SUFICIENCIA**

**PARA OPTAR EL TITULO PROFESIONAL DE:**

**INGENIERO ELECTRONICO**

**PRESENTADO POR:**

**RAUL REQUENA TAYPE**

**PROMOCION  
2007-I**

**LIMA – PERU  
2011**

**ESTUDIO DE CONVERSORES ANALOGO DIGITAL ASINCRONO APLICADO A  
REDES DE SENSORES DISTRIBUIDOS PARA MEDICION DE TEMPERATURA EN  
RESERVAS NATURALES**

## **SUMARIO**

El trabajo que se presenta a continuación de título ESTUDIO DE CONVERTORES ANALOGO DIGITAL ASINCRONO APLICADO A REDES DE SENSORES DISTRIBUIDOS PARA MEDICION DE TEMPERATURA EN RESERVAS NATURALES, trata sobre la metodología de diseño digital asíncrono, los principios de las redes de sensores distribuidos y la forma de relación de ambos conceptos que permita conseguir aplicaciones útiles y ventajosas para la toma de datos de variables lentas en escenarios de áreas extensas y difíciles geográficamente como las reservas naturales. En este informe se desarrolla como aplicación un conversor análogo digital sobre la tecnología FPGAs usando técnicas digitales asíncronas para un sensor de temperatura ambiental.

De esta forma, el presente trabajo tiene como fin presentar y desarrollar los conceptos de las redes de sensores inalámbricos, identificando la energía como factor crucial para un funcionamiento óptimo de la red y la metodología de diseño digital asíncrono como posible solución a ser evaluada ante una aplicación específica. En la actualidad existen dispositivos y sistemas bajo estos conceptos y en el ámbito tanto académico como industrial se realizan investigaciones constantes para mejorar sus características.

El estudio está desarrollado en cinco capítulos, que tratan primero sobre el escenario de aplicación con el parámetro temperatura y su importancia a ser monitoreada, luego los conceptos básicos tanto de la redes de sensores inalámbricos como de la metodología de diseño digital asíncrono son tratados, con esta base conceptual se presenta el conversor análogo digital desarrollado en base a FPGAs y finalmente en base a los resultados se dan la conclusiones y recomendaciones.

## INDICE

<b>PROLOGO</b> .....	01
<b>CAPITULO I</b>	
<b>PLANTEAMIENTO DEL PROYECTO</b> .....	03
1.1. Descripción del Proyecto.....	03
1.1.1. Monitoreo de variables ambientales en reservas naturales.....	04
1.2. Objetivos.....	06
1.2.1. Objetivo general.....	06
1.2.2. Objetivos específicos.....	06
1.3. Limitaciones del informe.....	06
<b>CAPITULO II</b>	
<b>MARCO TEORICO CONCEPTUAL</b> .....	07
2.1. Introducción.....	07
2.2. Redes de Sensores Inalámbricos Distribuidos.....	07
2.2.1. Componentes de un nodo perteneciente a una WSN.....	08
2.2.2. Protocolos en redes de sensores distribuidos.....	10
2.2.3. Manejo de la energía en las WSN.....	12
2.2.4. Industria de las WSN.....	19
2.2.5. FPGAs en las WSN.....	20
2.3. Diseño Digital Asíncrono.....	20
2.3.1. Entorno de diseño digital asíncrono.....	20
2.3.2. Protocolos de Comunicación.....	21
2.3.3. Codificación de data.....	23
2.3.4. Estilos de diseño asíncrono.....	25
2.3.5. Token.....	26
<b>CAPITULO III</b>	
<b>CONVERSION ANALOGO DIGITAL ASINCRONO</b> .....	33
3.1. Esquema de Análisis.....	33
3.2. Muestreo no uniforme.....	33
3.3. Modulación Sigma – Delta Síncrona.....	36
3.4. Modulación Sigma – Delta Asíncrona.....	38
3.5. Conversor análogo digital asíncrono.....	39
3.5.1. Parte Análoga: Comparador Regenerativo.....	40

3.5.2. Parte Digital: Lógica Digital Asíncrona.....	44
<b>CAPITULO IV</b>	
<b>ANÁLISIS Y PRESENTACION DE RESULTADOS.....</b>	<b>52</b>
4.1. Introducción.....	52
4.2. Simulación de la sección analógica.....	52
4.2.1. Comparador regenerativo con histéresis.....	52
4.3. Simulación de la sección digital.....	59
4.3.1. Consideraciones particulares para el test de circuitos asíncronos.....	59
4.3.2. Metodología de las mediciones.....	65
<b>CAPITULO V</b>	
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>69</b>
<b>ANEXO A</b>	
<b>Herramientas de especificación para diseños digitales asíncronos.....</b>	<b>71</b>
<b>ANEXO B</b>	
<b>Definición de bloques digitales asíncronos básicos.....</b>	<b>74</b>
<b>BIBLIOGRAFIA.....</b>	<b>78</b>

## PROLOGO

En la última década ha evolucionado el desarrollo de los sistemas embebidos y las comunicaciones inalámbricas. Estos dos factores marcan la tendencia de ubicar microcomputadoras en la mayor cantidad posible de actividades, pero cuidando siempre que sea lo más transparente para el usuario. Una de las actividades donde se puede aprovechar las ventajas de todos estos avances es el monitoreo y la toma de datos. Los sistemas micro electro mecánicos (MEMS) y los circuitos electrónicos integrados digitales han dado paso a la creación de los micro sensores que unidos a un procesador y transmisor adecuados, forman las redes de sensores distribuidos, con características propias diferentes de una red telefonía celular clásica o del internet.

La primera aplicación se dio en el campo militar, como sucede con la mayoría de avances tecnológicos, con el nombre de redes de sensores distribuidos (DSN), pero poco a poco se fue difundiendo al uso civil en áreas como la salud personal, agricultura, monitoreo del tráfico en las ciudades, mantenimiento de tuberías, seguimiento de variables ambientales, excavaciones mineras, entre otras, bajo el nombre de redes de sensores inalámbricos (WSN por sus siglas en ingles).

La implementación de una WSN está directamente relacionada con el escenario y condiciones de aplicación, es así que se presentan varios retos a superar como son: la calidad de comunicación, el procesamiento de data, la escalabilidad y la duración del tiempo de vida de la energía que acompaña a cada dispositivo.

En toda la etapa de evolución de este tipo de redes se han presentando diversas soluciones en su mayoría tomando como base a los microcontroladores para cada tipo de caso, tratando siempre de tener la máxima calidad de comunicación y procesamiento usando el mínimo de energía. Una alternativa válida para abordar la dificultad del tiempo de vida de los nodos de una red son los FPGAs que unidos a los sistemas y métodos de diseño digital asíncronos representan una tecnología que no presenta las dificultades del diseño digital síncrono predominante (clock skew, latencia, emisión electromagnética, etc.) y por su propia naturaleza traen algunos beneficios que se pueden aprovechar en alguna etapa del procesamiento, específicamente este informe desarrolla un conversor análogo digital asíncrono, dirigido por evento, de bajo consumo de energía y alta velocidad en base al FPGA STRATIX III de Altera

En el primer capítulo se plantea el caso de la variable a monitorear, así como los parámetros y objetivos a seguir en el desarrollo del trabajo.

El segundo capítulo es el marco teórico que describe las consideraciones de diseño de una red de sensores inalámbricos, haciendo énfasis en el manejo de energía, así también trata la teoría del diseño digital asíncrono, que incluye, protocolos de comunicación, transmisión de data y alternativas de implementación.

El tercer capítulo es la descripción del conversor análogo digital, que como primer elemento tiene un comparador con histéresis, luego una etapa de procesamiento de data netamente asíncrona que produce la modulación sigma delta asíncrona que es interpretada por un contador digital ascendente – descendente para tener el código binario final.

En el cuarto capítulo se analiza e interpreta los resultados realizados en base a las simulaciones usando como lenguaje de programación al VHDL y VERILOG en el simulador QUARTUS II de Altera, que son herramientas conocidas y muy difundidas entre los diseñadores. Si bien estas herramientas son de carácter síncrono permite ser usadas de modo adaptativo para diseños digitales de naturaleza asíncrona. Entonces se comprueba la adaptabilidad de los FPGAs, ya que las herramientas específicas para síntesis de alto nivel hasta el nivel físico de circuitos asíncronos están en plena evolución, no son actualmente de fácil acceso y son un factor del porque todavía no hay un uso masivo de esta metodología.

Finalmente se dan las Conclusiones y recomendaciones sobre el trabajo así como la dirección de futuras investigaciones.

## **CAPITULO I PLANTEAMIENTO DEL PROYECTO**

### **1.1. Descripción del Proyecto**

El Perú cuenta con áreas naturales protegidas, entre parques nacionales, reservas de biosfera, reservas nacionales, santuarios históricos, bosques de protección, zonas reservadas, reservas paisajistas y comunales. Algunas de ellas reconocidas por la Unesco como patrimonio mundial como: La Reserva del Manu, Reserva del Huascarán y la Reserva del Noroeste. Cada una de las áreas protegidas cuenta con características geográficas y atmosféricas propias y con necesidades distintas. Dentro de estas zonas extensas se desarrollan muchas actividades de carácter biológico y ambiental que influyen tanto a nivel local como al resto del país y el mundo.

La declaración como reserva natural de un área de manera oficial no es suficiente para garantizar el normal desarrollo de las especies vivas involucradas, es necesario el monitoreo y vigilancia constante de los factores que afectan el hábitat propio de cada región y que pueden alterar el normal desarrollo de las especies. El estado peruano tiene como obligación velar por el no deterioro de las zonas protegidas, prevenir posibles desequilibrios con consecuencias catastróficas (inundaciones, huaycos, incendios forestales, etc.) y promover, según sea el caso, un uso racional de los recursos protegidos por parte de las comunidades que están alrededor de estas áreas especiales.

Con este fin la comunidad científica especializada realiza investigaciones para lo cual es necesario la recolección de data por un periodo de tiempo desde puntos pre establecidos, mas aun las aplicaciones de carácter ambiental se caracterizan por recolectar datos desde muchos puntos, lo cual requiere la implementación de varios nodos de toma de datos. La Implementación de una red de monitoreo para reservas naturales implica un estudio detallado de cada escenario en particular, como factor común estas zonas comparten la característica de que son de gran extensión, de difícil acceso y no pueden ser dañadas en la medida de lo posible para no alterar el equilibrio natural. Bajo estas condiciones un seguimiento a los factores ambientales y biológicos se debe hacer de una manera externa o ser lo menos invasiva posible, dentro de esta categoría las redes de sensores inalámbricos junto con la captura de imágenes por satélite, son sin duda alternativas tecnológicas a tomar en cuenta, es más, el Instituto de Investigación de la Amazonia Peruana ha realizado un proyecto de nombre BIODAMAZ



usando imágenes satelitales para analizar los cambios en la selva peruana, como parte de un proyecto de diversidad biológica [1]. Proyectos como este muestran la preocupación creciente por el monitoreo de carácter ambiental no solo por parte de nuestro país si no a nivel mundial y es responsabilidad de la tecnología brindar el soporte necesario para las aplicaciones.

#### **1.1.1. Monitoreo de variables ambientales en reservas naturales**

Entre las diferentes variables que se pueden extraer de una zona declarada como reserva natural, es muy importante el monitoreo de factores ambientales como la temperatura, humedad, presión, radiación solar, viento y precipitación. Dentro de estos actores la temperatura o mas generalmente denominado el clima, es sin duda, el factor más influyente para el ecosistema a corto y mediano plazo.

La variable temperatura y más aun cuando se trata de la temperatura ambiental, es una magnitud de variación lenta. Uno de los sensores para captar la temperatura ambiental es el termistor que tiene una variación de 6% de su valor inicial por grado centígrado y un alcance de -100 °C a 300°C.

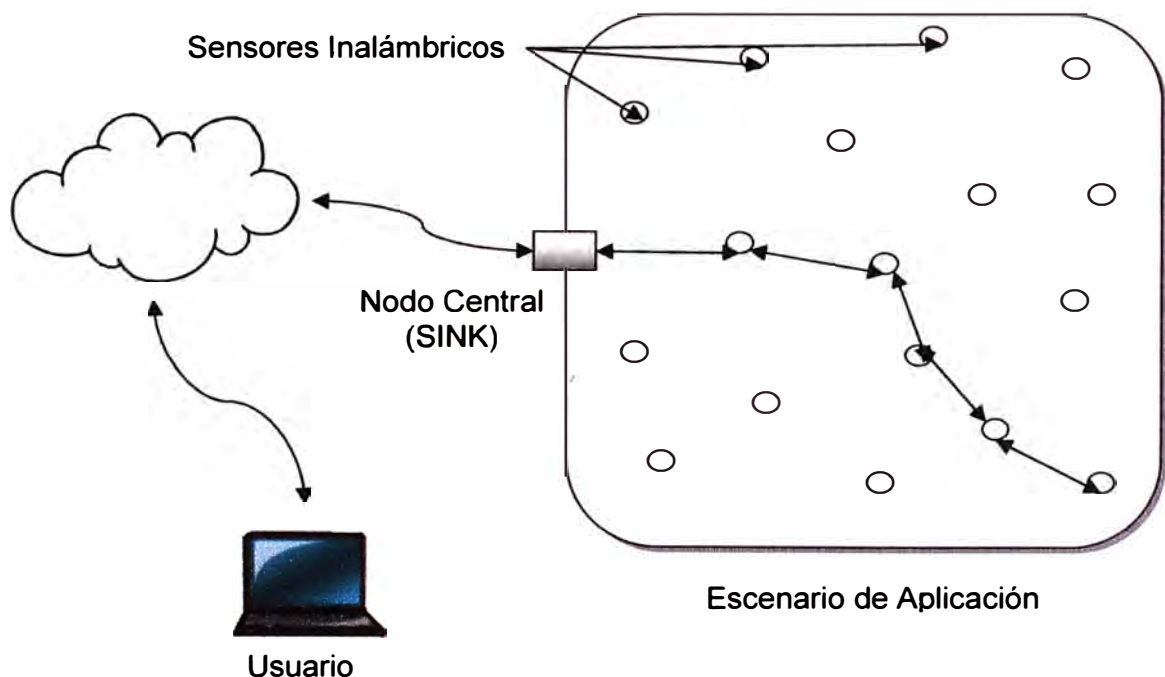
En los ecosistemas la temperatura ambiental como variable principal que indica el estado del clima, está sujeta a diferencias de latitud y de nivel con respecto al mar. A mayor altura sobre el nivel del mar, la temperatura desciende, de manera que las zonas son ambientes fríos o frescos. A partir de estas variaciones los ecosistemas cuentan con plantas, animales y que han desarrollado estrategias para soportar el frío, el calor o la mayor radiación solar según la zona en la que estén. Variaciones producidas por factores globales o locales traen consecuencias para todos los elementos del ecosistema, afectando no solo la vegetación y fauna, sino también la densidad de la población humana, sus necesidades y cultura.

El monitoreo de factores ambientales esta dentro de las aplicaciones de las redes de sensores inalámbricos que incluye aplicaciones en al ámbito militar, salud, domesticas e industriales. Los objetivos de tener una red de monitoreo ambiental es poder atender las demandas de previsión de cambios atmosféricos para actividades regionales de carácter conservativo, a la par de tener especialistas calificados que puedan interpretar los datos y poder responder de manera eficiente ante los fenómenos ambientales, para una correcta conservación de la flora y fauna, como se hace en otros países: se puede citar como aplicación la prevención de avalanchas en Suiza tomando datos de la densidad del suelo, el monitoreo ambiental de glaciares en Noruega, la prevención de incendios forestales en España y en países como Brasil, Estados Unidos, también existen aplicaciones de carácter preventivo y de investigación.

Todos estos proyectos coinciden en que un sistema de monitoreo útil para el seguimiento de factores ambientales deberá tener las siguientes características:

- El sistema deberá ser capaz de cubrir la mayor cantidad de área y funcionar correctamente por un largo periodo de tiempo, porque se espera que los puntos de toma de datos estén en lugares de difícil acceso y un probable mantenimiento o reemplazo, sería una tarea muy difícil.
- Los datos que el sistema aporta serán en tiempo real, con enlaces de comunicación fiables entre los nodos. Así mismo los dispositivos serán físicamente robustos para evitar cualquier daño por parte de algún factor externo.
- La red de sensores será capaz de adaptarse a cambios del escenario de ejecución y también podrá aceptar nodos adicionales, es decir, debe ser escalable para permitir una mayor densidad de sensores.
- En el proceso de transferencia se deberá asegurar la integridad y seguridad de la información.
- La red tendrá uno o varios nodos centrales (sink o cabecera), donde todos los datos recolectados serán pre procesados y estarán disponibles para que el usuario pueda acceder a ellas por cualquier red estándar conocida.
- El periodo de reporte y retardo de transmisión no son factores críticos, puesto que la toma de datos es para un análisis posterior y las variables ambientales como temperatura, humedad, intensidad de luz, etc son de cambio lento, típicos reportes son de 1 a 15 minutos. Esto implica la implementación de protocolos y mecanismos que permitan periodos activos y pasivos.

Una red de sensores inalámbricos por sus propias características y arquitectura general, como muestra la figura 1.1, tiene la capacidad de afrontar los requisitos mencionados.



**Figura 1.1 - Arquitectura general de WSN**

Dentro de las dificultades pendientes a superar, uno de los mayores es el consumo de energía que requiere cada nodo, por ejemplo, para una implementación promedio con tiempos de activación de 250mseg, transmisión de datos cada minuto, consumo de corriente en reposo de  $10\mu\text{A}$  y en estado activo de  $15\text{mA}$ ; usando una pila alcalina se obtiene, en promedio 1 día y medio de duración, usando una pila de litio con las mismas condiciones la duración es cercana a los 2 días [2]. Estos datos muestran que la energía disponible en un nodo inalámbrico es escasa, la mayor cantidad de energía se gasta en la transmisión y recepción inalámbrica, por lo tanto es necesario identificar data significate para no desperdiciar los recursos disponibles transmitiendo data no relevante, se han planteado varias soluciones tanto a nivel de red como a nivel de nodo, como son protocolos energéticamente eficientes o pre-procesamientos a nivel local, en este punto la metodología de diseño asíncrona se plantea como alternativa válida.

## **1.2. Objetivos**

### **1.2.1. Objetivo general**

El objetivo de este informe es desarrollar un conversor análogo digital, empleando la metodología de diseño digital asíncrono, como parte de un sistema completo desarrollado en base a FPGAs como alternativa para conseguir un uso más eficiente de la energía disponible en un nodo perteneciente a una red de sensores inalámbricos destinado a la recolección de data de baja frecuencia, es decir, con características de cambio lento, como la temperatura y otras variables ambientales.

### **1.2.2. Objetivos específicos**

- Elaborar una revisión previa tanto global como individual de una red de sensores inalámbricos. También estudiar y comprobar las técnicas de diseño digital asíncrono, con el fin de comprender su funcionamiento y características.
- Analizar los pasos que realiza cada nodo para su funcionamiento, con el propósito de identificar procesos donde se pueden implementar modificaciones para ahorrar la energía y extender el tiempo de funcionamiento.
- Finalmente desarrollar un conversor análogo digital (elemento básico en todo sistema de toma de data) asíncrono de alta velocidad y bajo consumo de potencia, tomando como plataforma los FPGAs validando la metodología de diseño digital asíncrono en esta tecnología.

## **1.3. Limitaciones del informe**

La principal limitación del presente informe es la implementación del módulo conversor análogo digital asíncrono y su adaptación en una red de sensores inalámbricos, debido a la poca accesibilidad a dicha tecnología y también por el alto costo que significan, haciendo énfasis en la interpretación del código termométrico y formación del código binario final.

## **CAPITULO II MARCO TEORICO CONCEPTUAL**

### **2.1. Introducción**

El presente capítulo contiene los conceptos básicos de las redes de sensores distribuidos y el diseño digital asíncronos relacionados con el tema del presente trabajo, permite establecer también la nomenclatura y abreviaciones que se usarán en los siguientes capítulos.

### **2.2. Redes de Sensores Distribuidos**

En forma general una red de sensores distribuidos hace referencia a una red de sensores inalámbricos (WSN – por sus siglas en ingles) y está formada por dispositivos de tamaño muy pequeño (algunos los llaman “motas”), capaces de desarrollar ciertas tareas de manera autónoma, constan básicamente de varios sub-sistemas como: una alimentación autónoma (mayormente baterías), radio transmisores y receptores para que se pueden comunicarse entre si y un elemento de procesamiento digital básico. Si los dispositivos no son del mismo tipo se dice que es una red heterogénea, si los dispositivos son del mismo tipo es una red homogénea.

Los nodos que conforman una red de sensores inalámbricos por lo general son “nodos inteligentes”, porque el pequeño procesador que poseen, no solo es usado para controlar los demás componentes del dispositivo si no que también interpreta la información de los nodos vecinos para formar un mapa global del escenario, también en algunos casos realizan pre procesamientos de la data capturada y administran los recursos energéticos con la que cuenta [3]. Este tipo de red presenta algunas ventajas y desventajas que a continuación se enumeran.

Ventajas de las WSN:

- a.- La red es de una gran densidad, es decir, se puede cubrir prácticamente el 100% del área a monitorear, con capacidad de crecimiento y adaptabilidad.
- b.- Los nodos se auto organizan de manera dinámica, intercambiando mensajes según el protocolo e identifican las rutas de transmisión más adecuadas hacia el nodo principal o nodo ‘SINK’.
- c.- Son de uso diverso según el tipo de sensor que lleven (pueden incluir sensores de vibración, temperatura, humedad, presión, así como químicos, biológicos y nucleares), puesto que los datos que se transfieren son independientes de la gestión de red.

Desventajas de las WSN:

a.- Por el tamaño del escenario de aplicación se deben de usar cientos de nodos por lo que la comunicación con el nodo colector (SINK) no se puede lograr en un solo salto o enlace.

b.- A pesar de la existencia de mecanismos de auto-alimentación, no es suficiente y el uso de baterías es imprescindible lo cual lleva a implementar un manejo adecuado de la energía.

c.- Los recursos de cómputo son limitados.

### **2.2.1. Componentes de un nodo perteneciente a una WSN.**

**a.- Sensor.-** Puede ser uno o varios y es el elemento por el cual el dispositivo se distingue de otros sistemas embebidos, el sensor provee información de algún tipo de variable del mundo físico. Se puede decir que está compuesto por un elemento sensor propiamente dicho y un conversor (ADC) análogo digital que recibe las señales del sensor, las convierte a un formato digital y alimenta a la unidad de procesamiento. La tendencia es producir sensores cada vez más pequeños, con mayor capacidad de procesamiento pero a bajo costo

**b.- Unidad de procesamiento.-** Es la unidad de control del nodo inalámbrico porque maneja todos los demás componentes, consta de una memoria y un procesador pequeños, maneja los datos medidos y la comunicación con los demás componentes de la red.

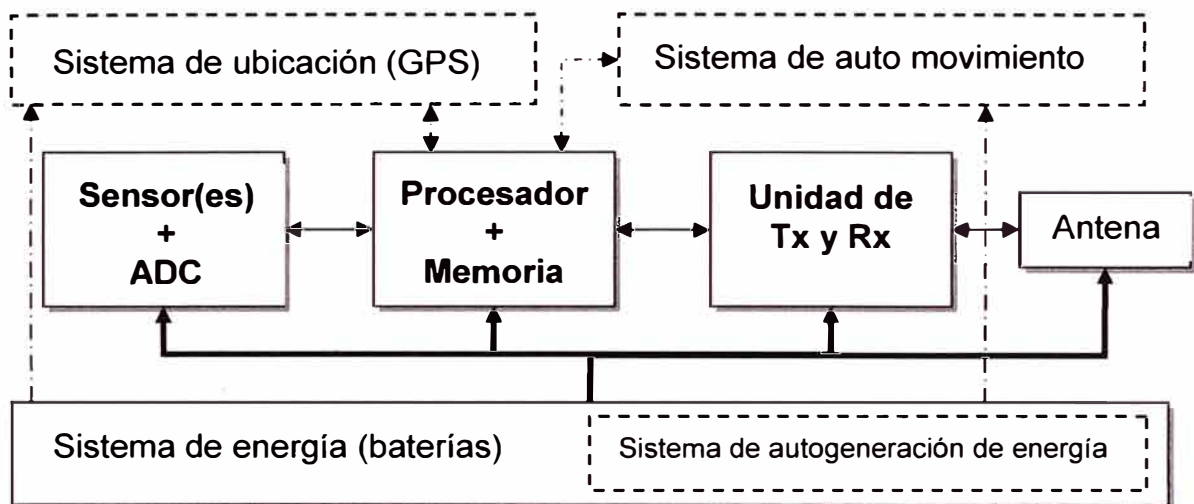
**c.- Unidad de transmisión.-** Es la unidad que permite la conexión del nodo a la red de manera inalámbrica convirtiendo la data a transmitir de un formato de bits a ondas de radio frecuencia, tanto de manera directa (transmisión) como inversa (la recepción), algunos transmisores tienen sus propios procesadores para el manejo de data en la capa física. Cabe señalar que es en este punto donde más energía se consume, por eso la mayoría de dispositivos comerciales cuentan con un control de cambio entre varios estados de operación: activo – en espera – reposo.

**d.- Unidad de alimentación o energía.-** Usualmente se usan baterías, aunque también se puede implementar algún tipo de generador propio, dependiendo de las facilidades del escenario de aplicación, esta unidad influye directamente en la capacidad que va a tener el nodo inalámbrico, tanto de procesamiento como en cobertura y tiempo de vida. El subsistema encargado de la alimentación provee energía DC al resto de elementos vía sus componentes activos como osciladores y amplificadores, más aun, consta de convertidores DC-DC para que cada parte del sistema obtenga el valor exacto de energía que requiere.

Adicionalmente se pueden mencionar otros componentes que pueden acompañar a un nodo inalámbrico, como se describe en la figura 2.1. Algunos dispositivos llevan un

sistema de posicionamiento (GPS) muy útil para formar las topologías de control, también se pueden implementar generadores autónomos de energía basados en principios como la energía solar, vibración, etc. Y en algunas aplicaciones se incluye un movilizador para darle desplazamiento controlado al dispositivo [4].

**e.- Sistema Operativo.-** El sistema operativo debido a las limitaciones en los procesadores embebidos y memoria, deberá ser del tipo dirigido por eventos, poco complejos y con tamaño de código reducido. Entre los más usados están: TinyOS, Contiki, SOS y PalOS; el primero es el más ampliamente usado, es de fuente abierta, tiene una arquitectura que permite un uso fino de energía que incluye protocolos, drivers y herramientas de adquisición, está desarrollado en base a un lenguaje llamado nesC, que es una extensión del lenguaje C y es el marco de trabajo para los otros sistemas operativos y futuros desarrollos. Contiki es de características similares al TinyOS e incorpora además IPv4 e IPv6. Los dos últimos están escritos en lenguaje C y están en proceso de estandarización [5].



**Figura 2.1 -** Arquitectura general de un nodo sensor

El principal obstáculo para un funcionamiento eficaz de una WSN es la energía disponible, en la mayoría de casos se usan baterías, lo cual limita el tiempo de operación de la red, convirtiendo a la energía en un recurso escaso. Por lo tanto cada nodo deberá economizar al máximo su energía disponible y desde un punto de vista global una red deberá tener una distribución alta de nodos para crear redundancia y no perder conectividad. Entre los elementos mencionados es el transmisor-receptor de radio frecuencia quien consume mayor energía, porque requiere de todo un procesamiento previo para la transmisión que incluye la modulación, filtros, demodulación y multiplexación sin embargo no se puede prescindir de él, porque nos da conectividad a la red y permite el flujo de datos. Como solución más común se busca tener básicamente varios estados (pasivo, en espera y activo) para el transmisor-receptor y es el micro procesador embebido quien decide los estados de cada uno de los componentes.



### 2.2.2. Protocolos en redes de sensores inalámbricos

Los protocolos dividen el proceso en capas similar al modelo OSI. Las capas son: Física, Data (MAC), Red, Transporte y Aplicación, todo esto en 06 planos, como muestra la tabla 2.1: localización, manejo de energía, sincronización, manejo de movilidad, topología y manejo de tareas.

La capa física define técnicas de generación de la frecuencia, modulación, encriptación, transmisión y recepción de manera segura. La capa de enlace (MAC) maneja la multiplexación del flujo de data evitando colisiones y errores, controla el acceso al medio de comunicación, permite enlaces punto - punto y punto - multipunto. La capa de red se encarga del correcto direccionamiento de la data, dado que no todos los nodos (que por lo general son cientos) pueden comunicarse directamente con el nodo "sink" o "gateway", esto sumado a que cada nodo tiene un rango de comunicación limitado, hace necesario que la comunicación sea través de múltiples saltos donde existen nodos puente y cada nodo, preferentemente, lleva como ID su ubicación geográfica, lo cual resulta en una configuración más completa. Como función adicional esta capa se encarga de la interface con otros tipos de redes, como el internet. La capa de transporte tiene como función principal mantener el flujo de data que requiere la capa de aplicación, controlando la congestión de data, pero está limitado por los recursos de energía y procesamiento del nodo, es por eso que se basa más en la capacidad de conjunto que tiene la red y en los recursos de procesamiento que tiene el nodo "SINK". Finalmente, la capa de aplicación contiene el manejo de las funciones del nodo, como el proceso de consulta e interacción en la red.

**Tabla 2.1 - Protocolos y planos para WSN**

<b>Capa de aplicación</b>	<b>Plano de localización</b>	<b>Plano de manejo de energía</b>	<b>Plano de sincronización</b>	<b>Plano del manejo de movilidad</b>	<b>Plano del manejo de topología</b>	<b>Plano del manejo de funciones</b>
<b>Capa de transporte</b>						
<b>Capa de red</b>						
<b>Capa de enlace (MAC)</b>						
<b>Capa física</b>						

Los planos adicionales que incluye el modelo son para que el nodo pueda desarrollar sus funciones de una forma más eficiente. El plano manejo de energía controla el uso de la energía del nodo identificando procesos no útiles durante su funcionamiento dando paso a estados de reposo en algunos de sus componentes. El plano manejo de movilidad, localización y topología mantienen actualizada la ubicación relativa del nodo con respecto a sus vecinos y al área a monitorear, esto con el fin de poder balancear la

energía necesaria para una posible transmisión y tener datos más exactos de la variable a monitorear, algunas configuraciones incluyen técnicas de rastreo.

El plano de manejo de tareas y de sincronización controla los tiempos de realización de toma de data por parte del nodo, para que en un área determinada no todos los nodos presentes realicen la misma tarea al mismo tiempo y que la data entregada sea correctamente ordenada en el tiempo y así poder llegar a un mayor tiempo de vida del sistema.

En general las diferentes capas y planos interactúan entre ellos y son flexibles (lo cual dificulta la estandarización), porque admiten modificaciones según la aplicación con el fin de conseguir una comunicación robusta pero con un gasto racional de energía.

Diferentes esfuerzos se han desarrollado para generar un estándar para este tipo de redes, dichos esfuerzos provienen tanto del sector industrial como académico, con este fin surgió la norma IEEE 802.15.4, que fue desarrollado para normalizar la transmisión inalámbrica de baja complejidad, velocidad reducida y paquetes de data ligeros. La capa MAC provee comunicación para topologías tipo estrella, mesh y cluster. El rango de transmisión varía entre 10 y 100 metros, la velocidad de transmisión va de los 20 a 250Kbps, actualmente es el protocolo Zigbee como evolución de IEEE 802.15.4, el estándar por defecto para la capa física y MAC esto permite una mayor integración de varias plataformas con diferentes capacidades.

### **Protocolo ZIGBEE**

Las comunicaciones por medio de redes inalámbricas como los celulares, radio, Wi-Fi han experimentado un gran avance y el estudio de redes de este tipo se ha ampliado, específicamente a lo que compete a las redes de sensores inalámbricos, el desarrollo del protocolo ZigBee/802.15.4. surgió como resultado de un esfuerzo conjunto de más de quinientas empresas y representa el estándar para este tipo de redes. Este protocolo es actualmente el más usado puesto que está pensado para un bajo consumo de energía, alta seguridad y confiabilidad.

El estándar ZigBee aporta una capa de red que gestiona las tareas de enrutamiento y mantenimiento de los nodos de la red; así también define un entorno de aplicación que proporciona una subcapa de aplicación (APS) que establece una interfaz para la capa de red y los objetos de los dispositivos tanto de ZigBee como del diseñador cuando implementa un caso específico [6].

Entre las principales características de este protocolo, se mencionan las siguientes:

- Bandas de trabajo: 2.4GHz(16Canales), 915Mhz(10 Canales), 868 MHz (1 Canal)
- Tasas de transferencia: 250 Kb/s, 40 Kb/s, 20 Kb/s
- Topologías: estrella y p2p (punto-a-punto)
- Direccionamiento MAC recortado (16 bits) y extendido (64 bits)



- Métodos de acceso al canal: CSMA-CA Soporta redes slotted (QoS) y non-slotted
- Bajo consumo energético
- Gran densidad de nodos por red
- Direccionamiento a nivel de red (16 bits)
- Soporte para enrutamiento de paquetes
- Permite topología de malla, gracias a las posibilidades de enrutamiento
- Dispositivos FFD (coordinador, router y dispositivo final) y RFD (dispositivo final)
- Tiene un alcance de aproximadamente 70 hasta unos 300 metros.

### **2.2.3. Manejo de la energía en las WSN.**

En cada implementación el objetivo es hallar un estado de equilibrio entre una transmisión segura y un gasto de energía necesaria.

El manejo eficiente de la energía se hace identificando procesos innecesarios y redundantes, esto se puede ver desde dos puntos de vista, una es a nivel global, desarrollando protocolos y topologías energéticamente eficientes y la segunda es a nivel local, identificando la mayor cantidad de procesos internos no trascendentales para la operación del nodo individualmente.

#### **1. Manejo de la energía en una topología (nivel global) identificando procesos no útiles.**

Una topología de control de una red de sensores inalámbricos se forma a partir de los rangos de transmisión de cada nodo, el fin es aprovechar al máximo las capacidades de la red pero usando la menor cantidad de recursos.

Se establece que un enlace entre dos nodos existe si y solo si la potencia recibida es mayor al umbral de sensibilidad ' $\beta$ ' del receptor, este umbral depende de las características de la unidad del transmisión-recepción del nodo y de las condiciones de transmisión como la velocidad, tamaño y cantidad de data. La potencia recibida es igual a la potencia transmitida menos la pérdida en la trayectoria, de estos factores el valor conocido es la potencia de transmisión, para la pérdida en la trayectoria se necesita usar un modelo.

Según la teoría de radio propagación existen tres fenómenos que se pueden presentar en un enlace, estos son: la difracción, la reflexión y la obstrucción, en base a esto se han elaborado modelos válidos para estimar la pérdida en la trayectoria de un radio enlace, entre los modelos más conocidos tenemos: Propagación en espacio libre, el modelo de dos rayos, el modelo de trayectoria a gran distancia con sus variaciones a gran y baja escala.

La estimación del rango de cobertura de un nodo dentro de una distribución se hace tomando como referencia el modelo de trayectoria a gran distancia, que es una combinación de métodos analíticos y experimentales, este modelo establece que la

pérdida de potencia en la trayectoria es proporcional a la distancia de separación elevado a un exponente 'α', conocido como exponente de pérdida por trayectoria [6].

$$P_{rx} = \frac{P_{tx}}{d^\alpha} \quad (2.1)$$

Donde:

$P_{rx}$ : Potencia en el nodo receptor.

$P_{tx}$ : Potencia de transmisión.

α: Coeficiente de pérdida por trayectoria.

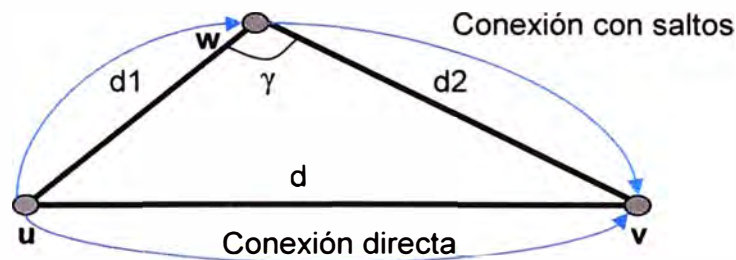
d: distancia entre los nodos Transmisor y receptor.

El valor de α depende de las condiciones ambientales y ha sido evaluado en diferentes condiciones, como se resume en la tabla 2.2.

**Tabla 2.2** – Coeficiente de pérdida 'α' para diferentes áreas.

Escenario	α
Espacio libre	2
Área urbana	2.7 – 3.5
indoor	1.6 – 1.8
outdoor	4 - 6

Se asume α = 2 a modo de referencia y según la distribución de la figura 2.2, donde:  $d_1 < d$  y  $d_2 < d$ ; la pérdida de energía entre el nodo 'u' y 'v' usando la conexión directa es directamente proporcional a  $d^2$  y tomando como "nodo puente" a 'w' la pérdida de energía es proporcional a  $d_1^2 + d_2^2$



**Figura 2.2** - Comparación de enlace directo y enlace usando un nodo puente.

Trayectoria directa:

$$P_{rx} = \frac{P_{tx}}{d^\alpha} \quad (2.2)$$

Trayectoria usando el nodo 'w' como puente:

$$P_{rx} = \frac{P_{tx}}{d_1^\alpha} + \frac{P_{tx}}{d_2^\alpha} \quad (2.3)$$

Y por fórmula geométrica tenemos:

$$d^2 = d_1^2 + d_2^2 - 2d_1 \times d_2 \times \cos \gamma \quad (2.4)$$

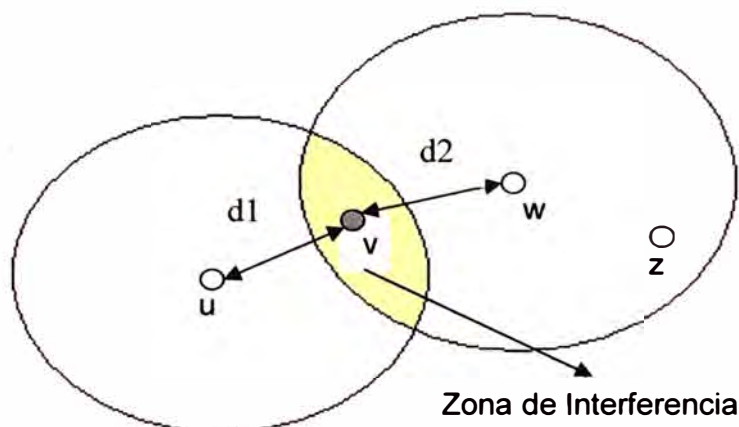
Por el tipo de distribución  $90 \leq \gamma \leq 180$ , por lo tanto  $-1 \leq \cos \gamma \leq 0$

De esta forma:

$$d^2 \geq d_1^2 + d_2^2 \quad (2.5)$$

De las ecuaciones (2.2), (2.3), (2.4) y (2.5), la transmisión directa entre 'u' y 'v' es más costosa que si se usa como puente el nodo 'w', es decir, es mejor la comunicación entre dos nodos, desde el punto de vista de ahorro de energía, usando saltos múltiples de pequeños tramos que de manera directa. Entonces al formar una topología el objetivo es identificar las líneas de comunicación ineficientes (líneas directas) y removerlas.

Otro aspecto importante es que al ser redes inalámbricas, los nodos comparten el mismo canal de transmisión, lo cual puede llevar a interferencias. Según la figura 2.3. Si un nodo 'u' quiere alcanzar a un nodo 'v', regula su potencia de transmisión, pero si 'v' está dentro del rango de transmisión de otro nodo 'w', surgen interferencias si  $d_2 < d_1$ , entonces la potencia recibida por 'v' por parte de 'u' es menor que la potencia recibida por 'v' por parte de 'w', para tener el enlace correcto el nodo 'u' tiene que elevar su potencia para superar la interferencia producida por 'w', lo cual lleva a un mayor consumo de recursos y da paso a una posible interferencia con otros nodos. Se presenta así una disyuntiva entre el punto de vista individual (donde una potencia de transmisión grande asegura enlaces confiables) y la del sistema completo (donde potencias de transmisión bajas son deseables para evitar interferencias y asegurar la conectividad total de la red).



**Figura 2.3** – Escenario de interferencia entre nodos en una WSN.

Para resolver el dilema anterior es necesario formar las topologías usando protocolos y modelos de interferencias. Uno de los modelos más simples (Gupta and Kumar) establece que la región de interferencia es un círculo centrado en el receptor, cuyo radio depende de las características de la unidad de comunicación inalámbrica y de la distancia del nodo que quiera entablar un enlace [7].

Sea 'u' y 'v' dos nodos en una distribución, con nodos intermedios, como muestra la figura 2.4, la zona de interferencia es un círculo de radio:

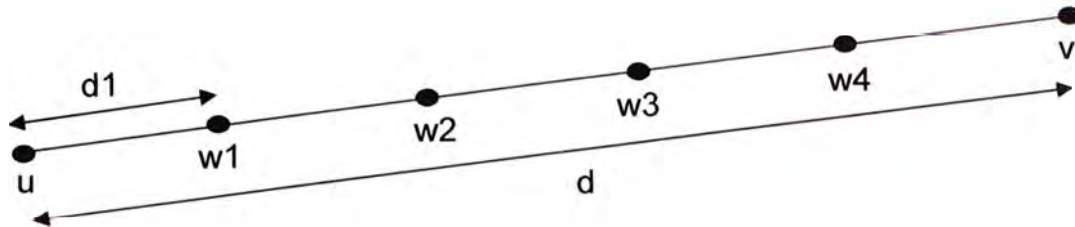
$$\text{Radio de interferencia} = (1 + \eta)\delta(u, v) \quad (2.6)$$

Donde:

$\eta > 0$ : Constante que depende de las características del transmisor

$\delta(u,v)$ : Distancia entre los nodos.

Si entre los puntos 'u' y 'v', existen otros nodos del sistema la interferencia de la conexión directa y por saltos sería:



**Figura 2.4** – Rango de interferencia por enlace directo y enlace con múltiples saltos.

La región de interferencia directa es:

$$\text{Región de interferencia directa} = \pi \times d^2(1 + \eta)^2 \quad (2.7)$$

La región de interferencia por saltos es la suma de las regiones de cada nodo intermedio:

$$\text{Región de interferencia de nodo intermedio} = \pi \times d_i^2(1 + \eta)^2 \quad (2.8)$$

Si la distancia entre los nodos intermedios es una fracción de la distancia total, la expresión (2.8) sería:

$$\text{Región de interferencia de nodo intermedio} = \pi \times \left(\frac{d}{k+1}\right)^2(1 + \eta)^2 \quad (2.9)$$

Usando la desigualdad de Holder:

$$\sum_{i=1}^{k+1} \left(\frac{d}{k+1}\right)^2 = (k+1) \left(\frac{d}{k+1}\right)^2 < \left(\sum_{i=1}^{k+1} \frac{d}{k+1}\right)^2 = d^2 \quad (2.10)$$

Se deduce de las ecuaciones (2.6), (2.7), (2.8), (2.9) y (2.10) que la comunicación vía multi-saltos es mejor que la comunicación directa, desde el punto de vista de interferencia y capacidad de la red. Esta última observación y la anterior nos lleva a concluir que la regulación de la energía va relacionada con la capacidad de tráfico y conectividad que la red puede brindar [7].

Según las soluciones que se plantean para lograr una buena distribución según el escenario de uso, las topologías se clasifican en:

**a.- Topologías homogéneas.**- todos los nodos tienen el mismo rango de transmisión, conocido como RTC (rango de transmisión crítico), el objetivo es encontrar el RTC apropiado, según los factores que influye en el cálculo del RTC tenemos:

- Basada en ubicación.- La ubicación de cada nodo es esencial, se puede conocer usando módulos GPS.
- Basada en dirección.- Se debe tener al menos un estimado de la dirección relativa de los vecinos de un nodo.
- Basada en los nodos vecinos.- Cada nodo posee cierta información de sus nodos vecinos que le sirve para formar su propio ordenamiento.

**b.- Topologías no homogéneas.-** el rango de transmisión es variable.

## **2. Manejo de la energía identificando procesos no útiles localmente**

Un nodo que forma parte de una red tiene dos funciones básicas: puede actuar como un punto que origina la data o como un puente que retransmite información proveniente de un nodo vecino dirigida a otro sector de la red, generalmente hacia un colector.

En base al análisis de estos procesos se puede identificar algunas acciones innecesarias como: Intentos de comunicación por parte de un nodo a otros que ya no están disponibles, también el cambio de un estado pasivo a activo por algún nodo sin data trascendental a transmitir. Para que el nodo pueda cumplir con sus dos tareas fundamentales realiza tres acciones, que son: capturar data, procesarla y transmitirla.

La primera tarea es capturar la variable a monitorear, el consumo de energía en esta acción depende del escenario y del tipo de sensor usado, si la red es para una aplicación con mucho ruido requerirá mayor consumo de energía para tener una muestra limpia, seguido del sensor esta el conversor análogo digital (ADC), previo paso por un filtro pasa-bajo y un amplificador de señal. La energía asociada al ADC depende directamente de la resolución y la frecuencia, a mayor resolución y frecuencia mayor será el costo de energía y viceversa.

Según la aplicación y variable a monitorear se hace la elección de la resolución y frecuencia. Para algunos casos se puede implementar estados de reposo ante pequeños cambios de la variable monitoreada.

La segunda tarea es el procesamiento de data que es hecha generalmente por un microprocesador basado en tecnología CMOS, esta tecnología tiene como característica la pérdida de potencia cada vez que es requerida, dicha pérdida es proporcional a la frecuencia del CLOCK, a la capacitancia (área del chip) y al cuadrado del voltaje. Otra pérdida de energía se produce cuando hay corriente de fuga entre el voltaje de alimentación y tierra, por lo tanto la pérdida total es la suma de estos dos factores.

$$E_p = N \cdot C \cdot V_{dd}^2 + V_{dd} \left( I_0 \cdot e^{V_{dd}/n \cdot V_T} \right) (N/f) \quad (2.11)$$

Donde:

N: Numero de ciclos por tarea

$V_{dd}$ : Voltaje de alimentación

C: Capacitancia total

$I_0$ : corriente de fuga

n: constante que depende del hardware

$V_T$ : voltaje umbral

F: frecuencia del reloj.

Según la arquitectura del microprocesador se puede añadir otros factores a la expresión (2.11). En general se puede regular los factores que afectan la pérdida de

energía, la fórmula anterior indica que una reducción del voltaje de alimentación  $V_{dd}$  reduce también la pérdida de energía. Por otro lado el retardo de compuerta es:

$$T_g = \frac{V_{dd}}{K(V_{dd}-V_{th})^a} \quad (2.12)$$

Donde:

$K$  y  $a$  son variables que dependen del procesador.

$T_g$  es el retardo de compuerta

$V_{dd}$  es el voltaje de alimentación y

$V_{th}$  es el voltaje umbral.

Para  $a \approx 2$ ,  $T_g$  está en relación inversa con el voltaje  $V_{dd}$ , una reducción en el voltaje  $V_{dd}$  incrementa el retardo de compuerta, el cual puede ser controlado para reducir el tiempo de no actividad del procesador, la idea es que el  $T_g$  tiempo de retardo de la compuerta sea mucho menor a la frecuencia del procesador, esto se logra reduciendo el voltaje de alimentación, entonces para cada frecuencia existe un voltaje de alimentación mínima según la expresión:

$$f \leq \frac{k(V_{dd}-V_{th})^a}{V_{dd}} \quad (2.13)$$

Donde:

$K$  y  $a$  son variables que dependen del procesador.

$V_{dd}$  es el voltaje de alimentación y  $f$  es la frecuencia de operación.

La correspondencia entre frecuencia y voltaje de alimentación es referida como escala dinámica de voltaje (DVS), donde para cada frecuencia existe un voltaje que optimiza el funcionamiento del procesador. Según los periodos de mucha o poca actividad el procesador puede manejar dinámicamente la alimentación y frecuencia de funcionamiento, cuidando siempre no afectar en lo posible su rendimiento.

La tercera tarea es la comunicación que involucra tanto la transmisión como la recepción, el gasto de energía en ambos procesos prácticamente es el mismo, con la diferencia que en la transmisión, si es para distancias grandes usa un amplificador adicional. Los componentes del modulo de comunicación son: el mezclador, el sintetizador de frecuencia, un oscilador de voltaje controlado (VCO), phase-locked loop (PLL), el demodulador y un amplificador de potencia.

Para la transmisión se usa el sintetizador de frecuencia, el oscilador de voltaje controlado (VCO), el modulador y el amplificador de potencia. Usualmente el gasto de los tres primeros componentes es constante porque la velocidad y tamaño del paquete son valores fijados por los protocolos seleccionados, pero el amplificador de potencia si puede variar dependiendo de la distancia o rango a cubrir.

En el modo recepción el nodo usa el sintetizador, el VCO, el amplificador de bajo ruido, el mezclador, el amplificador de frecuencias intermedias y el demodulador. Este consumo es por lo general constante.

Ambos procesos consumen gran parte de la energía disponible, lo ideal es que los circuitos encargados pasen a un estado inactivo cuando no exista data a transmitir o recibir, sin embargo el paso de un estado pasivo a un estado activo también representa un consumo importante de energía conocida como "startup energy consumption" que podría ser el factor dominante si se pasa de un estado a otro con mucha frecuencia, resultando una solución ineficiente.

Un ciclo completo de un nodo se puede resumir en la siguiente fórmula:

$$P_C = +P_{rx} + P_{tx} + P_0 \quad (2.14)$$

Donde:

$P_C$ : potencia en un ciclo completo.

$P_0$ : potencia de salida.

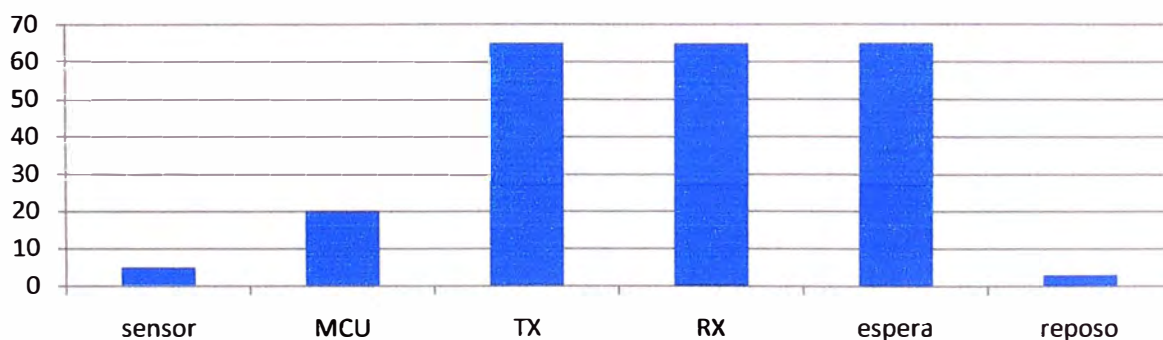
$P_{tx}$ : potencia de la circuitería para la transmisión.

$P_{rx}$ : potencia de la circuitería para la recepción.

De acuerdo a la expresión anterior podemos concluir que el consumo de energía depende de tres factores principales: el consumo propio de la circuitería del modulo de comunicación, el segundo componente depende de la velocidad de transmisión y tamaño del paquete y el tercer componente depende de la distancia de los nodos vecinos y es controlada por los protocolos MAC y routing.

El procesamiento al igual que la etapa de captura de data consumen muy poca energía en comparación con la tercera tarea que es la transmisión, como se indica en la figura 2.5 para un nodo de modelo micaZ, por eso economizar en lo posible la necesidad de transmisión y recepción usando protocolos adecuados junto con técnicas de pre-procesamiento en las etapas anteriores es necesario [7], [8].

**Potencia consumida por el nodo mica Z en sus diferentes estados (mW)**



**Figura 2.5** – Potencia consumida por los componentes de un nodo.



Como factor adicional al modelo de consumo de energía, es necesario tomar en cuenta la frecuencia de uso del modulo, en una aplicación de monitoreo donde la entrega de información periódica es necesaria se gastara mayor energía que si el monitoreo está basado en la captura de eventos. Igualmente depende del protocolo escogido porque algunos requieren un mayor intercambio de paquetes que otros entre los nodos y con mayor o menor frecuencia.

#### 2.2.4. Industria de las WSN

El mayor fabricante es Crossbow Technologies y desde el año 2008 la demanda ha ido en constante crecimiento llegando a las 140 millones de unidades en el 2010, así también las aplicaciones usan cada vez mas cantidades de nodos (en el orden de los cientos), sin embargo existen todavía barreras a superar como: la estandarización, facilidad de uso, consumo de energía, tamaño del nodo, fiabilidad y ciclo de desarrollo [5], [8]. Los principales fabricantes que poseen tecnología de redes de sensores inalámbricos son:

- Crossbow, entre sus modulos esta Mica, Mica2, Micaz, telos, telosb, entre otros.
- Sentilla, también llamada Tmote ha desarrollado la plataforma TmoteSky y Tmote Invent.
- Shockfish SA, está orientada a aplicaciones industriales, ha desarrollado la plataforma Tinynode
- Ember, es uno de los promotores de la alianza Zigbee, por lo tanto sus soluciones son las más escalables.
- Sun, desarrollo el SunSpot bajo la norma IEEE 802.15.14, a diferencia de las otros fabricantes, este nodo esta desarrollado en base a la maquina virutal Java.
- Nanor RK, enfocado a nodos de bajo consumo para aplicaciones en tiempo real.

La tabla 2.3 muestra las características de algunos de los modelos más usados

**Tabla 2.3 – Secuencia de transmisión.**

	MICAZ	SunSPOT	TelosB	IMote2
Procesador	ATmega128L	ARM9	MSP430F1611	PXA271ARM
Procesador (Mhz)	16	180	8	13-416
Memoria Ram (Kb)	4	512	10	256
Memoria Flash (kb)	512	4096	1024	3200
Corriente activa (ma)	48	70	25	>44
Corriente en espera (ma)	8	24	2	>31
Corriente en reposo (µa)	15	36	6	387
Radio Frecuencia (Hz)	2.4GHz	2.4GHz	2.4GHz	2.4GHz
Casa	Crossbow	Sun	Crossbow	Crossbow



### **2.2.5. FPGAs en las WSN**

El desempeño de las funciones que debe cumplir un nodo de una red de sensores inalámbricos da cuenta de la necesidad de contar con procesadores embebidos de capacidad suficiente pero de un tamaño adecuado y que no requiera un consumo de energía elevado. Los tipos de procesadores mayormente usados, como muestra el apartado anterior, debido a su flexibilidad y precio son los microcontroladores, DSP, por eficiencia y consumo de energía se prefiere los FPGAs y ASICs, aunque recientes microcontroladores muestran mejores características, como por ejemplo: TI MSP430F2618 y MSP430F5437 con una memoria RAM de 8 KB y 16 KB memoria flash de 116KB y 256KB respectivamente.

Ante los retos a superar y las potenciales aplicaciones futuras (monitoreo de la salud personal, domotica, automotion, etc) para este tipo de tecnología se toma como alternativa reciente a los FGPA's debido a su capacidad de realizar proceso en paralelo, alta velocidad y bajo consumo de potencia, capacidades ideales para las redes de sensores inalámbricos donde las tareas de captación, procesamiento y comunicación de data se realizan al mismo tiempo. También es importante mencionar la facilidad que tiene de ser reprogramada para una aplicación específica, por ejemplo no cuenta con un conversor análogo digital predefinido, lo cual aumenta la flexibilidad, la programación se realiza usando lenguajes de descripción de hardware como VHDL o Verilog lo cual le da adaptabilidad a la metodología de diseño asíncrono. Entre los puntos en contra se tiene el alto costo que se atenúa por su capacidad de reprogramación y que tiene un consumo de corriente mayor que un ASIC, pero este último no se puede reprogramar [9], [10].

### **2.3. Diseño Digital Asíncrono**

Los problemas y limitaciones que tienen las WSN, sobre todo acerca del manejo de energía de manera individual, pueden ser tratados usando la lógica digital asíncrona.

El desarrollo de circuitos digitales bajo el modelo síncrono actualmente domina la industria del diseño digital y está basado en dos conceptos fundamentales: primero, se trabaja con señales binarias y segundo, los componentes que forman parte del sistema tienen una noción de tiempo común y discreta, donde las señales cambian solo después de los flancos de esta señal común, esto alivia significativamente los procesos de diseño y validación. Sin embargo, como alternativa válida están los circuitos asíncronos que también asumen que las señales son binarias, pero no hay un tiempo común discreto para todos los componentes. En vez de esto usan diferentes modelos de sincronización para la comunicación y secuencia de las operaciones [11].

#### **2.3.1. Entorno de diseño digital asíncrono**

La más significativa diferencia entre circuitos síncronos y asíncronos es la relación que existe entre la data a procesar y las señales de control. En los circuitos síncronos cada

operación realizada por la unidad lógica tiene que ser finalizada dentro de un periodo dado por una señal de reloj general “master clock”. En los circuitos asíncronos el cambio de las señales de control es causado por la misma data, la finalización de una operación lógica es indicada después de producir la salida y antes de realizar la acción siguiente.

### 2.3.2. Protocolos de Comunicación

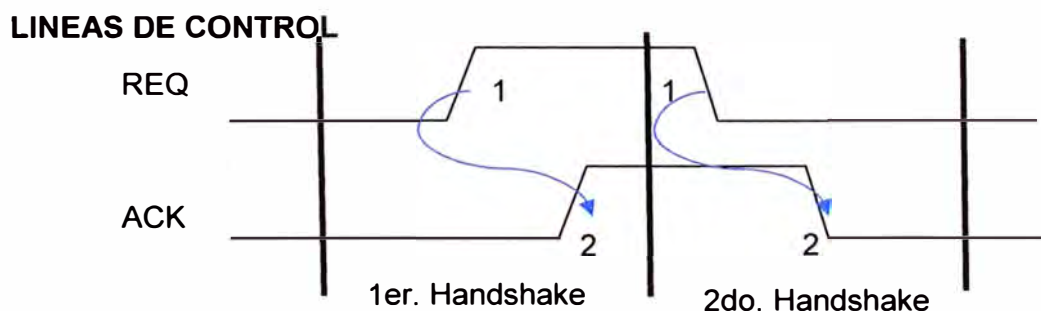
La comunicación entre módulos o subsistemas en el ambiente asíncrono está basada en el uso de “handshakes”, entre transmisor y receptor, esto con el fin de que el receptor conozca el instante donde la data es válida y el transmisor conozca el momento en que puede retirar la data. La base para los protocolos más simples involucra el uso de dos líneas conectadas entre módulos adyacentes: una línea REQUEST (REQ) y una línea ACKNOWLEDGE (ACK). La señal REQ es producida por la parte del sistema que inicia la comunicación (transmisor) y la señal ACK es producida por la parte del sistema que acepta los datos (receptor).

Dependiendo de si el transmisor o receptor inician el proceso de comunicación, los canales de datos pueden ser “push channel”, cuando el transmisor es la parte activa y “pull channel” cuando el receptor inicia el proceso de comunicación.

Existen dos protocolos de comunicación tipo handshaking, que son: señalización a dos fases (sin retorno a cero) y señalización a cuatro-fases (con retorno a cero).

#### 1. Señalización a dos fases.

Conocido también como señalización no retorno a cero (NRZ), este protocolo usa señales de transición para indicar los mensajes de requerimiento (REQ) y aceptación (ACK). Cualquier transición entre dos niveles lógicos, ALTO a BAJO o de BAJO a ALTO, otorga el mismo significado, estas transiciones son referidas también como eventos. La señalización a dos fases opera entre dos módulos de la siguiente manera:



**Figura 2.6** – Handshake de dos fases

De la figura 2.6, para un tipo de circuito push channel, el transmisor genera un evento (en este caso un cambio de BAJO a ALTO) sobre la línea REQUEST. Un tiempo después, el receptor detecta la transición REQ y lo indica la generando una transición sobre la línea ACKNOWLEDGE (transición de BAJO a ALTO). Una vez que el transmisor observa el cambio en la línea ACKNOWLEDGE está habilitado para otro ciclo de comunicación. El funcionamiento descrito, implica que no se puede producir un evento en

la línea ACK sin que antes se hay recibido un evento en la línea REQ y viceversa, mientras esta regla sea respetada se garantiza el funcionamiento correcto del circuito.

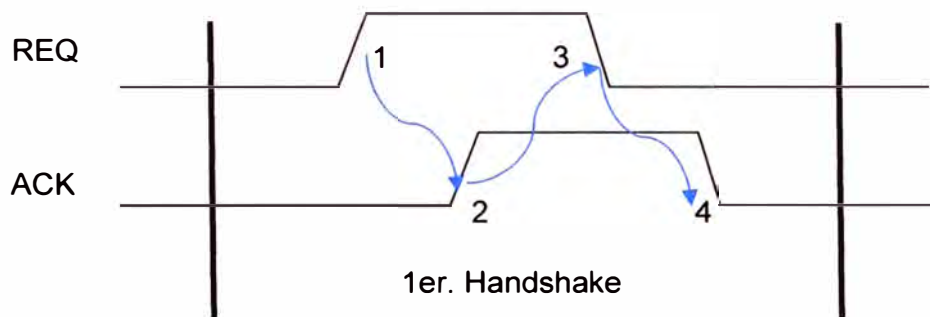
Esta señalización aumenta el desempeño del circuito debido a que ambos flancos de las señales de control tiene significado, es decir, aprovecha los dos flancos de la señal.

## 2. Señalización a cuatro fases.

Llamado también señalización con retorno a cero (RZ), aunque la secuencia de requerimiento y aceptación es la misma que la anterior, la diferencia es que en este caso solo se usa el flanco positivo de las señales de control, entonces después de cada comunicación, las señales de control tienen que retornar al nivel bajo antes de ser usadas otra vez.

No se debe confundir este tipo de señalización con una comunicación a pulsos de las líneas de control. La señalización a cuatro fases opera entre dos módulos de la siguiente manera:

### LINEAS DE CONTROL



**Figura 2.7 – Handshake de cuatro fases**

Una correcta operación, de acuerdo a la figura 2.7, involucra sostener el cambio de BAJO a ALTO de la señal REQ hasta que sea aceptada por el receptor, el cual indicara la recepción cambiando la señal ACK de BAJO a ALTO, solo así la fuente podrá volver a cambiar de ALTO a BAJO a la señal REQ para posteriormente el receptor haga lo mismo con la línea ACK y poder iniciar otro ciclo de comunicación.

El receptor no puede poner en ALTO la señal ACK sin antes estar seguro de que la línea REQ está en ALTO primero y tampoco puede poner a CERO la señal ACK si antes la señal REQ no lo ha hecho. Para la fuente, podemos establecer que no puede retirar la señal REQ si no ha habido respuesta en la línea ACK y tampoco puede generar otro REQ en ALTO hasta que el ACK del receptor no haya retornado a cero. La señalización de cuatro-fases es más fácil de implementar debido a la disponibilidad estándar de herramientas de diseño, los cuales han sido desarrollados para manejar niveles lógicos de flancos positivos y por lo tanto es más familiar para la metodología de diseño digital actual. Teóricamente el protocolo a 2 fases nos puede llevar a circuitos más rápidos que el protocolo a 4 fases, pero frecuentemente la implementación de circuitos que responden

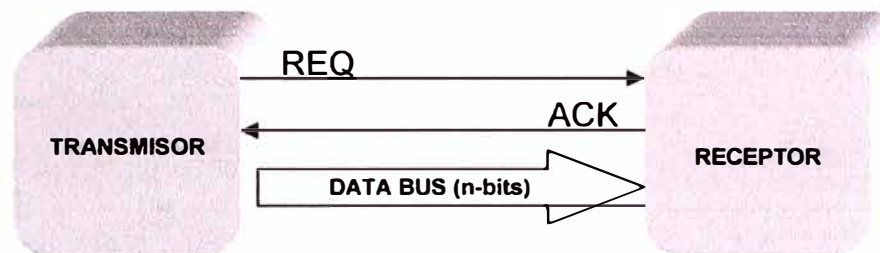
a eventos (como el protocolo a 2 fases) es más compleja, porque consumen muchos recursos en tiempo de diseño, potencia consumida y área ocupada [11], [12].

### 2.3.3 Codificación de data

Debido a que las señales de control no llevan data, una vez definidos los protocolos de comunicación, es necesario definir un mecanismo que especifique como se va a transferir la data entre emisor y receptor, las dos interfaces más usadas son bundled data y dual rail, que se diferencian básicamente en dos puntos: la robustez y el número de líneas usadas. Estos mecanismos pueden ser usados tanto con señalización a dos fases como a cuatro fases.

#### 1. BUNDLED DATA

El termino bundled data también conocido como “single rail” hace referencia a la transmisión de datos usando niveles booleanos normales para codificar la información sobre las líneas de comunicación, es decir, si la línea va a transmitir un ‘0’ lógico, se pone a un nivel BAJO y si va a transmitir un ‘1’ lógico, la línea se pone a un nivel ALTO. Se puede entender también como una transferencia de data serial. Las líneas de control REQ y ACK van junto con el bus de transmisión como muestra la figura 2.8.

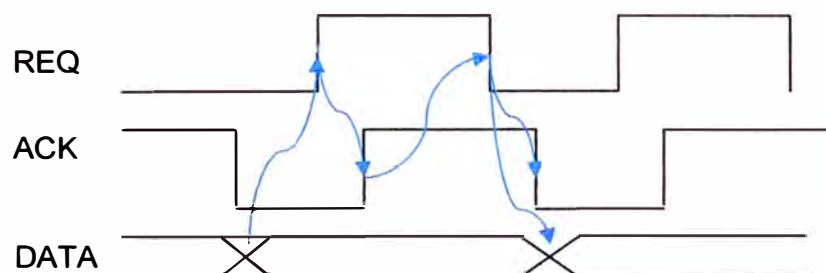


**Figura 2.8** – Protocolo de comunicación Bundled Data o single rail.

Si usamos el protocolo de 4 fases con el bundled con un “push channel”, la secuencia de comunicación sería, conforme la figura 2.9:

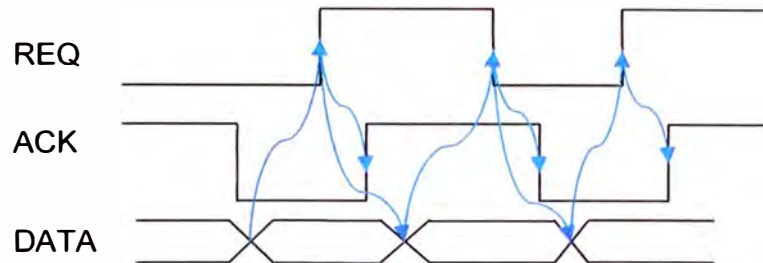
- (1) El transmisor emite data y fija el REQ en alto.
- (2) El receptor absorbe la data y fija el ACK alto.
- (3) El transmisor responde poniendo al REQ en bajo (punto en el cual la data no se garantiza a ser válida).
- (4) El receptor acepta este cambio poniendo el ACK en bajo.

En este punto el transmisor puede iniciar el siguiente ciclo de comunicación.



**Figura 2.9** – Protocolo de 4 fases con transmisión bundled data.

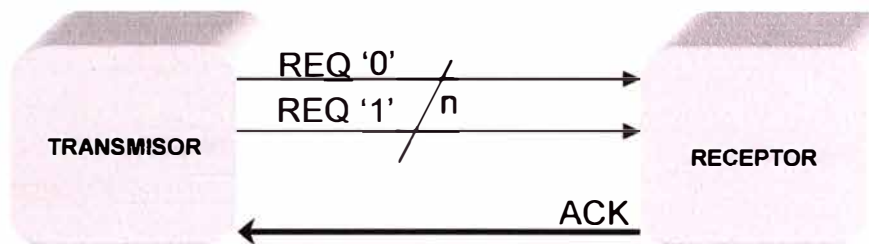
El protocolo bundled data usando la señalización de 2 fases. Aprovecha ambos flancos de la señal, como muestra la figura 2.10, un evento en REQ asegura data valida, un evento en ACK indica la aceptación de la data, en ese punto ya no se garantiza la validez del dato, hasta que otro evento en REQ vuelve a garantizar una nueva data, iniciando otro ciclo de comunicación.



**Figura 2.10** – Protocolo de 2 fases con transmisión bundled data.

## 2. DUAL RAIL

En este tipo de mecanismo se usa dos líneas por bit de información, como muestra la figura 2.11, una por cada posible estado de los datos a transmitir ('0' o '1'), es decir, por cada bit de información se deberá añadir dos líneas más.



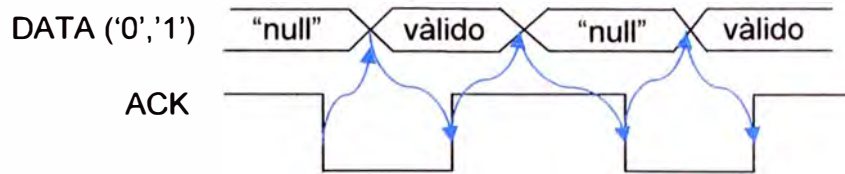
**Figura 2.11** – Protocolo de comunicación double rail.

Estableciendo en ALTO a una de las líneas se transmite un bit de información, cuyo valor dependerá de cuál de las líneas ha sido activada, si REQ0 está en ALTO se transmite un '0' lógico y si REQ1 está en ALTO se transmite un '1' lógico, si las dos líneas están en ALTO es un valor lógico no permitido y si las dos líneas están en un estado BAJO se considera una valor "null", esto se resume en la tabla 2.4.

**Tabla 2.4** – Secuencia de transmisión DUAL RAIL.

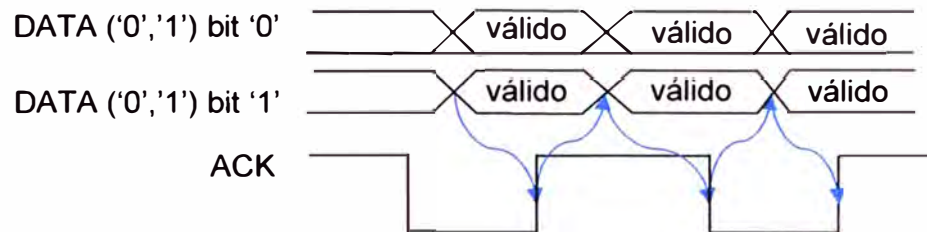
	REQ '0'	REQ '1'
"Null"	0	0
'0' válido	1	0
'1' válido	0	1
No usado	1	1

Cuando se combina este mecanismo de transmisión de data con la señalización a 4 fases, luego de la transmisión del dato y la aceptación del mismo por parte del receptor, ambas líneas REQ deben ser desactivadas, representando un espacio o valor "null" en la transmisión que para un correcto funcionamiento siempre deberá estar presente entre los valores permitidos como muestra la figura 2.12.



**Figura 2.12** – Protocolo de 4 fases con transmisión dual rail.

Si en cambio se combina el dual rail con la señalización a 2 fases, no existe el valor "null" de espacio, luego de que un valor válido es aceptado (mediante una transición en la línea ACK), la fuente coloca otro valor válido en las líneas de comunicación (mediante una transición en cada una de las líneas REQ) que será aceptada por el destinatario. La figura 2.13 muestra la transmisión de data para dos bits, como se observa no existe valor intermedio.



**Figura 2.13** – Protocolo de 2 fases con transmisión dual rail de dos bits.

#### 2.3.4. Estilos de diseño asíncrono

El estilo de diseño asíncrono se establece por los modelos usados en base a las suposiciones del retardo asumidos en las compuertas y en las líneas de comunicación, para las líneas de control y líneas de datos, los modelos son:

##### 1. Bounded Delay

En este modelo se establece que los retardos tanto en los elementos de procesamiento como en las líneas de conexión son conocidos o acotados, los sistemas tienen que estar estables antes de realizar la siguiente acción. Es lo más parecido a los circuitos síncronos. Para su implementación se usa muchos elementos de retardo estandarizados, para evitar inconvenientes al cambiar de plataformas de desarrollo.

##### 2. Delay Insensitive

En este modelo se establece que los retardos no son conocidos o acotados ni en compuertas ni en las líneas de conexión, los sistemas tienen que estar estables antes de realizar la siguiente acción. En este modelo es necesario que el receptor indique que ha recibido la data para que el transmisor pueda iniciar otro ciclo de comunicación, es en este modelo donde se puede usar los "handshakes" de 2 o de 4 fases.

##### 3. Quasy delay insensitive

Se asumen retardos arbitrarios en líneas y compuertas, sin embargo se establece el concepto de "isochronic fork", que son las divisiones de líneas de comunicación donde la diferencia de llegada de la data transmitida se asume despreciable.



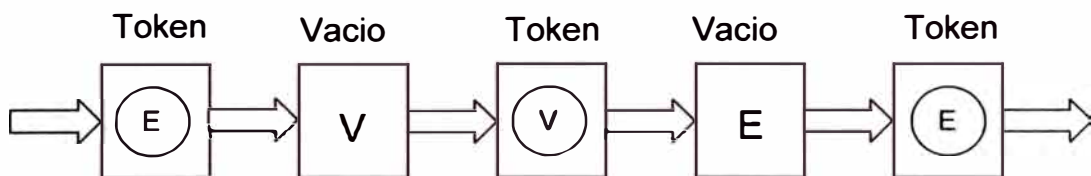
#### 4. Speed Independent

Los retardos en este modelo son en los boques funcionales no acotados, pero en las líneas de comunicación son despreciables o al menos menor que en el retardo mínimo en las compuertas.

##### 2.3.5 Token

Los circuitos de control (REQ y ACK) son incapaces de pasar data, una de las forma de intercambiar mensaje entre módulos asíncronos es usando el concepto de TOKEN. El token es representado por un circulo lleno o vacio según el token es válido o no, dentro del elemento de memoria donde esta almacenada, los tokens pueden pasar de un latch al siguiente y pueden ser divididos, eliminados y reunidos con otros tokens en escenarios lógicos.

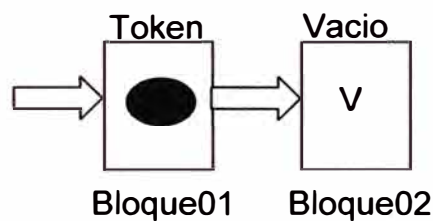
La regla general para almacenar y transmitir tokens es: Un latch (i) puede aceptar un token (vacío 'E' o válido 'V') de su predecesor (i-1) si y solo si su sucesor (i+1) ha aceptado y almacenado el token que previamente el latch (i) tenia [12]. La secuencia de estados es como muestra la figura 2.14:



**Figura 2.14** – Secuencia de estados para la transmisión del token.

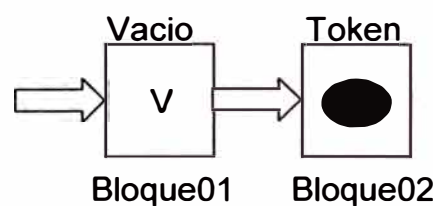
La regla anterior se puede sub dividir en dos sub reglas:

- Regla token: Un latch puede recibir y almacenar un nuevo token (válido o inválido) de su predecesor si y solo si esta "bubble" (vacía), como muestra la figura 2.15.



**Figura 2.15** – Regla token.

- Regla Bubble o Burbuja: Un latch pasa a estado vacío (bubble) si y solo si su sucesor ha recibido y almacenado el token que estaba almacenada antes en su memoria, como muestra la figura 2.16.



**Figura 2.16** – Regla Bubble.

El token puede ser transmitido en cada proceso de “handshaking”, solo el transmisor y receptor conocen el significado de recibir o transmitir uno o varios tokens según los objetivos del circuito en particular. Los latches son los encargados de almacenar los tokens, cualquier otro elemento, como puede ser un circuito combinacional, deberá ser totalmente transparente al control de flujo de data.

Para que un token se pueda propagar deberá tener por delante un elemento vacío y cada sistema asíncrono deberá tener latches en estado vacío así como al menos un token si no estará en estado pasivo sin posibilidad de iniciar su funcionamiento.

La metodología de diseño digital asíncrono, cuenta con compuertas básicas orientadas a la respuesta ante eventos en sus entradas, entre las principales, podemos mencionar a:

### XOR

Este modulo es también conocido como un unión o merge. La tabla lógica, de acuerdo a la figura 2.17, muestra que un evento o transición en la entrada causa una transición en la salida. Una entrada reset podría ser adicionando para forzar a la salida a un estado definido inicial

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

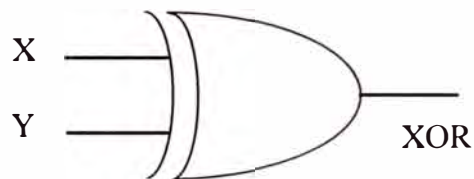


Figura 2.17 – Elemento XOR.

### ELEMENTO C

En este elemento una transición ocurre sobre la salida solo después que una misma transición ha ocurrido sobre cada una de sus entradas. En términos de nivel lógico, cuando los niveles de entrada se emparejan, la salida asume el mismo nivel lógico de las entradas, en otros casos la salida retiene su nivel previo.

X	Y	C[n]*
0	0	0
0	1	C[n-1]
1	0	C[n-1]
1	1	1

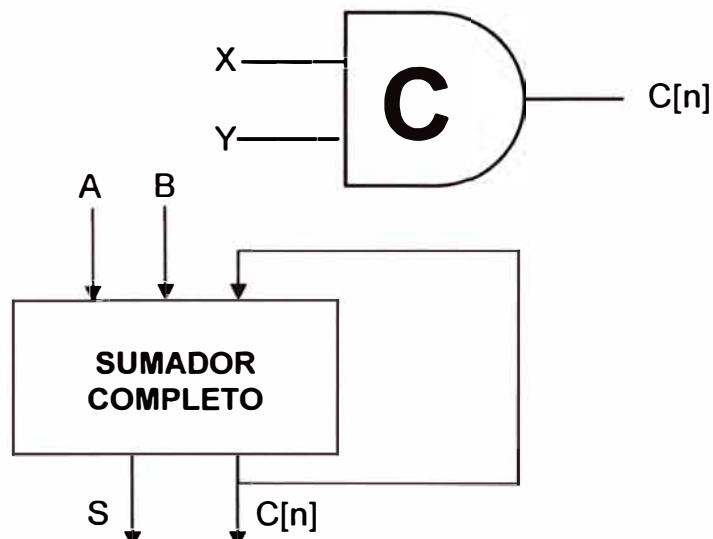
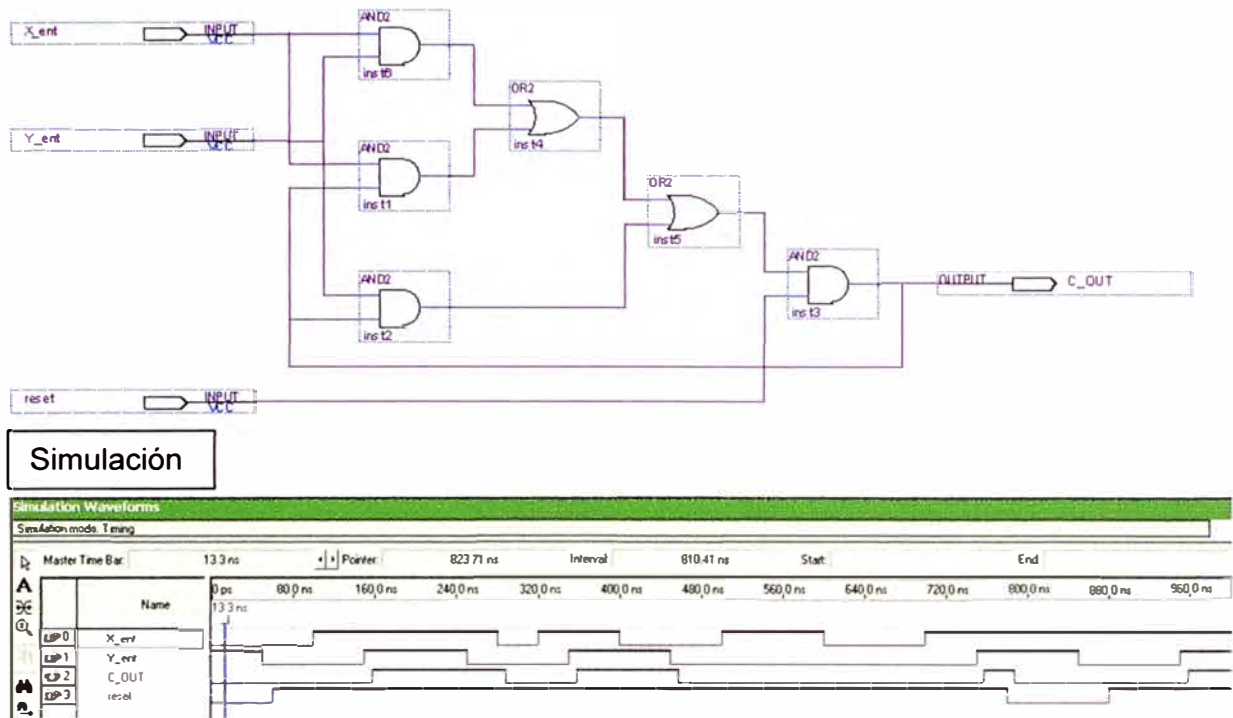


Figura 2.18 – Elemento C.



Una entrada reset podría ser adicionando para forzar a la salida a un estado definido inicial. La tabla lógica y el símbolo de este elemento son mostrados en la figura 2.18. La implementación de este bloque es similar a un sumador completo de un bit con realimentación, a partir de este hecho se diseña el elemento C usando compuertas básicas lo cual contribuye al uso de menor área y disminuye el retraso.

Usando el simulador QUARTUS II, tenemos el siguiente resultado.

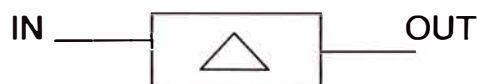


**Figura 2.19 – Elemento C.**

Conforme a las figuras 2.19, el elemento cumple con la descripción y solo usa 1 ALUT del FPGA, cero elementos de memoria y 4 pines, tres de entradas y dos de salida. El retardo es de 7.9nseg

### ELEMENTO DE RETARDO (DELAY)

Este elemento es usado para proveer un retardo puro sobre las líneas de señales de control. Dependiendo de la plataforma de implementación variara el valor de retardo, asegurándose solo el retardo mínimo, el símbolo más conocido entre los diseñadores es mostrado en la figura 2.20.



**Figura 2.20 – Elemento DELAY.**

El diseño de este elemento es una cadena de inversores parametrizable, con un retardo mínimo de 6.5nseg, las sentencias usadas evitan que el compilador elimine los inversores obteniéndose para cada inversor un ALUT en el FPGA, la simulación de la figura 2.21 es para n=25, obteniéndose un retardo de 15.6 nseg y 25 ALUTs del FPGA.

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
LIBRARY altera;
USE altera.maxplus2.ALL;
ENTITY DELAY_ELEMENT IS
GENERIC (
N : INTEGER := 25); -- N, numero de inversores en la cadena
PORT (
X : IN STD_LOGIC; -- X, entrada de la cadena inversora
F : OUT STD_LOGIC); -- F, salida de la cadena inversora
SIGNAL T : STD_LOGIC_VECTOR (N downto 0); -- T, la interconexion entre los inversores
BEGIN
END DELAY_ELEMENT;
ARCHITECTURE DELAY_ELEM OF DELAY_ELEMENT IS
BEGIN
T(0) <= X;
F <= T(N);
g0 : for i in (N-1) downto 0 generate
BUFn : LCELL PORT MAP (T(i), T(i+1));
end generate;
END DELAY_ELEM;
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
LIBRARY altera;
USE altera.maxplus2.ALL;
PACKAGE DELAY_ELEMENT_PACKAGE IS
COMPONENT DELAY_ELEMENT
GENERIC (
N : INTEGER := 20); -- N, numero de inversores
PORT (
X : IN STD_LOGIC; -- X, entrada de la cadena
F : OUT STD_LOGIC); -- F, salida de la cadena
END COMPONENT;
END DELAY_ELEMENT_PACKAGE;

```

### Simulación

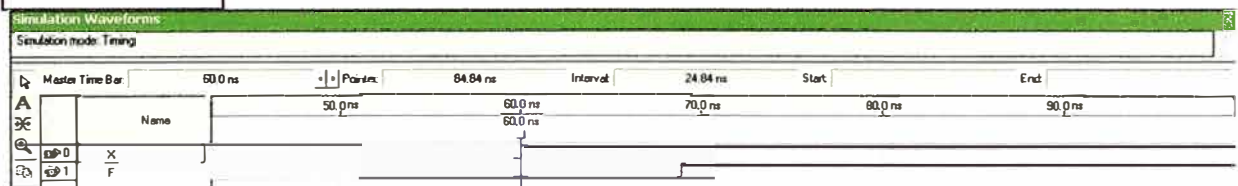


Figura 2.21 – Código VHDL del elemento DELAY y simulación.

### SELECT KELLER (Sel-K)

Este elemento permite recuperar estados lógicos. La figura 2.22 muestra el símbolo para este elemento.

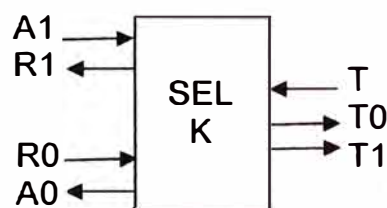
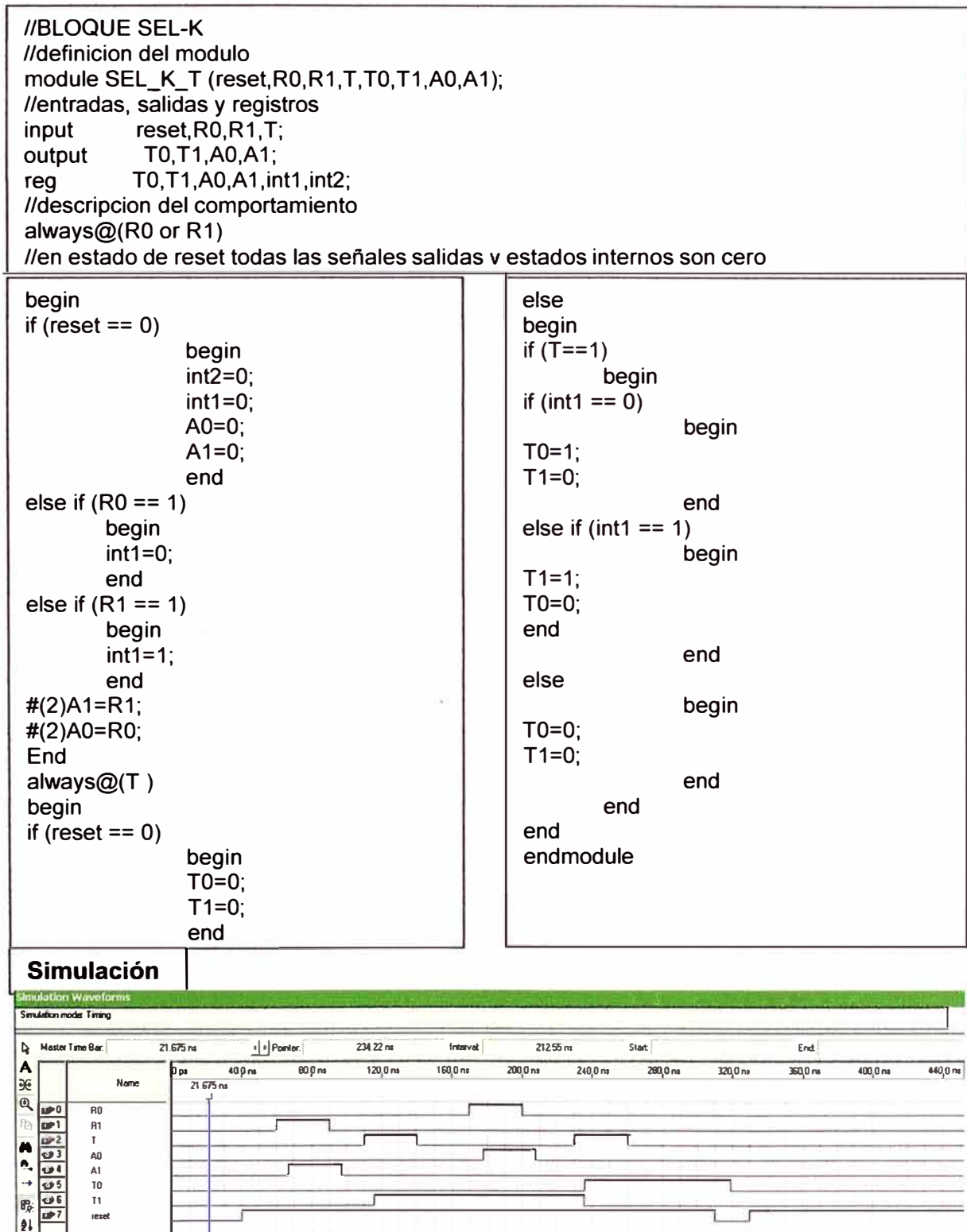


Figura 2.22 – Elemento K.

El estado interno es puesto a '1' mediante un evento en la entrada R1 y otro evento A1 confirma el cambio. De igual forma el estado interno cambia a '0' con un evento en la

entrada R0 y confirmada posteriormente en A0. La consulta se hace produciendo una transición en la entrada T, el estado interno estará en '0' si se obtiene un evento en la salida T0 y en '1' si se tiene un evento en T1.

La simulación y código en Verilog se muestran en la figura 2.23, el elemento SEL\_K utiliza 4 ALUTs y 8 pines de los cuales tres son de entrada y cinco de salida. El retardo de la salida es 7.9 nseg.



**Figura 2.23** – Código VHDL del elemento SEL K\_ y simulación.

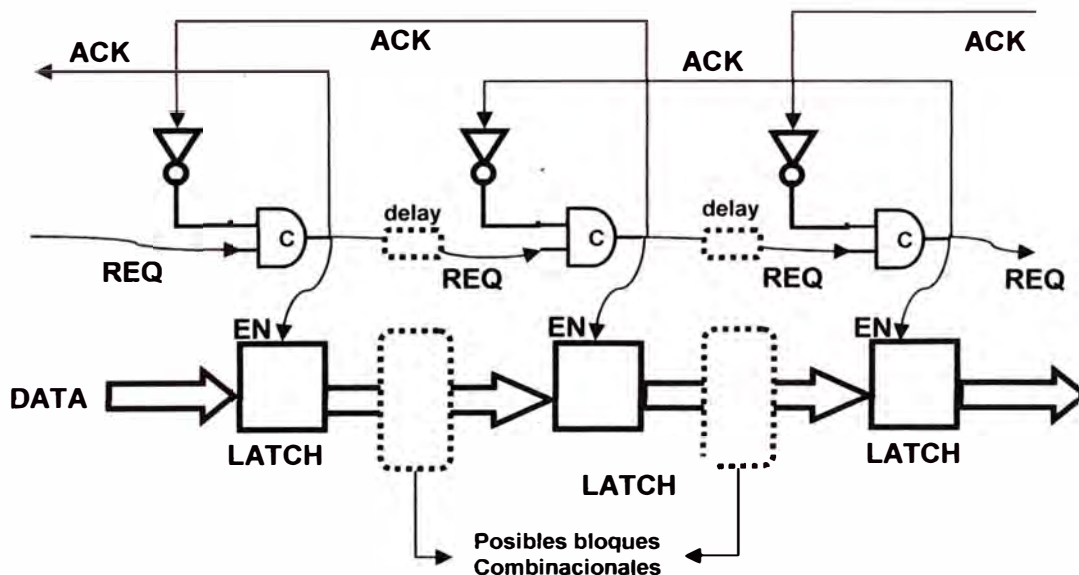
## MICROPIPELINE

Dentro del ámbito de desarrollo de circuitos digitales asíncronos, un elemento crucial es la micropipeline propuesta por Sutherland [13]. El diseño adopta la teoría asíncrona de la señalización por eventos (el principio general no distingue entre flancos de subida o bajada) y descarta el marco teórico síncrono de la señal de referencia global.

El mecanismo permite manejar las señales REQ Y ACK. Elementos de almacenamiento (latch) y bloques combinatoriales. La configuración y lógica de funcionamiento requerido solo se realiza usando una cadena de elementos 'C's y elementos inversores para formar lazos alrededor del flujo de eventos, la dinámica se resume en la siguiente regla: Si dos elementos tiene el mismo estado, este se mantiene, pero si dos elementos tiene diferentes estados, se copia el estado del elemento (i) en el elemento (i+1).

Para el protocolo de 04 fases, como se observa en la figura 2.24, se usa un Latch que tiene una entrada 'EN' que habilita el paso de la data al siguiente modulo, en caso se use un bloque combinatorial entre módulos, se debe incluir un elemento de retardo en la línea REQ y para una transmisión de data DUAL RAIL se agrupan las señales REQ a través de una compuerta OR.

Dentro de las propiedades más importantes esta que puede funcionar tanto para un tipo push como pull channel, son totalmente transparentes al bloque combinatorial y se emplean para protocolo de 02 fases como de 04 fases.



**Figura 2.24** – Pipeline 04 fases – bundled data

Para el protocolo de 02 fases, como se observa en la figura 2.25, se usa un Latch que tiene dos entradas 'C' y 'P', porque se usa los dos flancos de las señales de señalización. Con la entrada 'C' primera se captura la data entrante y con 'P' se permite el paso al siguiente modulo, en caso se use un bloque combinatorial entre módulos, se debe incluir

un elemento de retardo en la línea REQ, igual que el caso anterior y para una transmisión DUAL RAIL se agrupan las señales REQ a través de una compuerta OR.

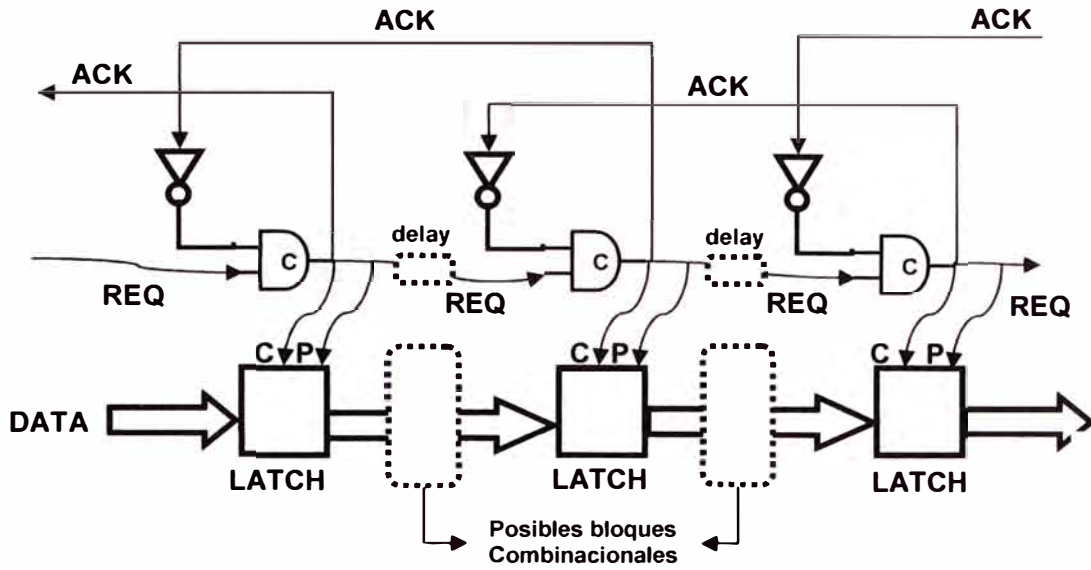


Figura 2.25 – Pipeline 02 fases – bundled data

La figura 2.26 muestra un escenario micropipeline de 03 elementos, la simulación usa 03 ALUTs y tiene un retardo de 7.5nseg.

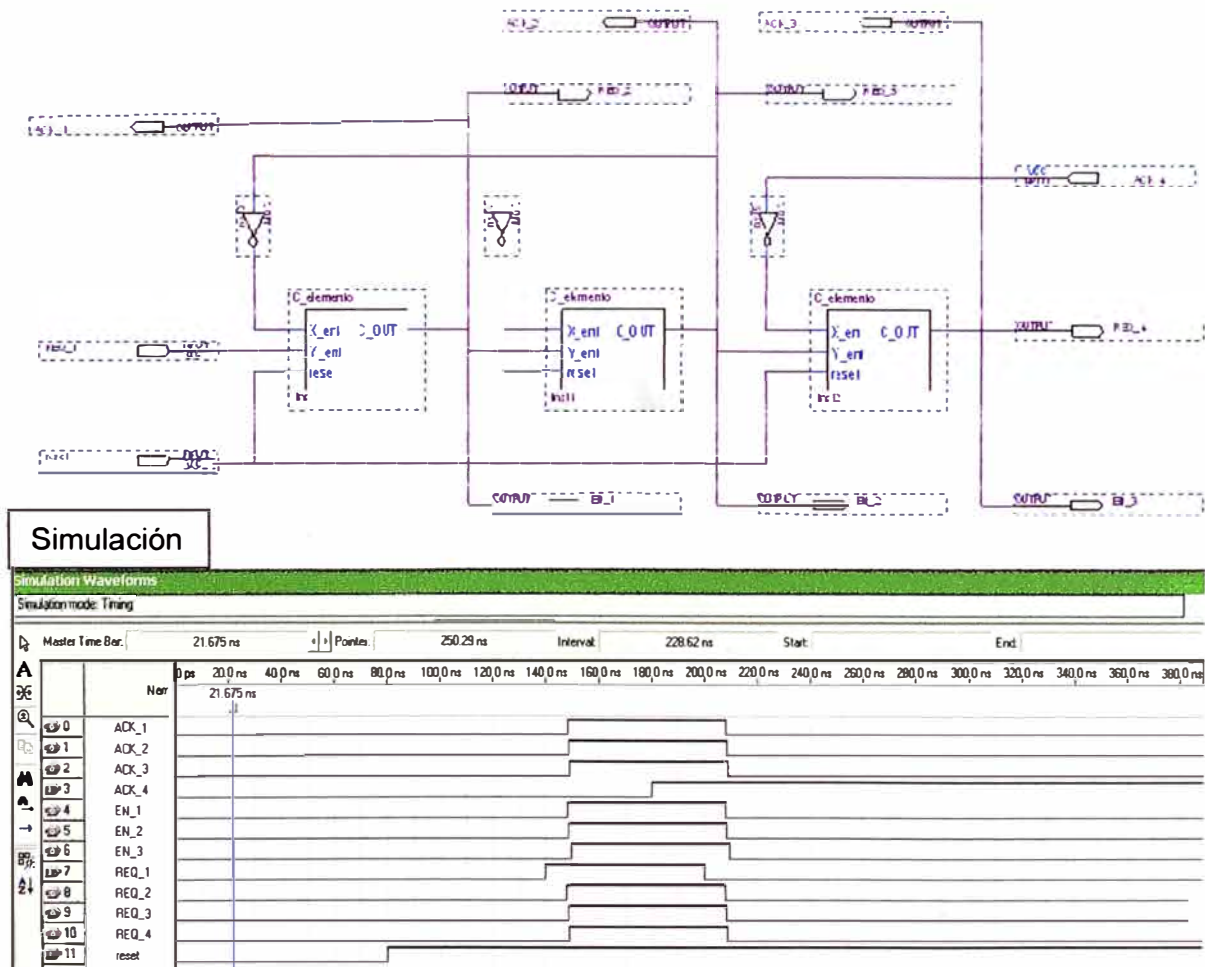


Figura 2.26 – Simulación de un escenario micropipeline de 03 elementos

## **CAPITULO III**

### **CONVERTIDOR ANALOGO DIGITAL ASINCRONO Y MUESTREO NO UNIFORME**

#### **3.1. Esquema de Análisis**

En este capítulo se describe el conversor análogo digital asíncrono, el cual incluye componentes análogos y digitales, así como algunos conceptos relacionados al módulo.

El objetivo del diseño del ADC es minimizar la potencia de consumo, manteniendo una razonable frecuencia de operación.

#### **3.2. Muestreo no uniforme**

Tradicionalmente las señales análogas son convertidas a señales discretas, usando un proceso de muestreo sobre puntos uniformemente espaciados en el tiempo, la progresión autónoma del tiempo dispara la ejecución de toma de cada muestra. Existe una teoría conocida acerca de la frecuencia de muestreo, la cual establece que la ubicación de los puntos en el dominio del tiempo, como mínimo debe cumplir con el siguiente teorema para su posterior recuperación: "Una señal puede ser reconstruida exactamente, si las muestras que se toman de ellas son espaciadas al menos a la mitad del periodo de la frecuencia más alta", esta sentencia es más conocida como el teorema de Nyquist.

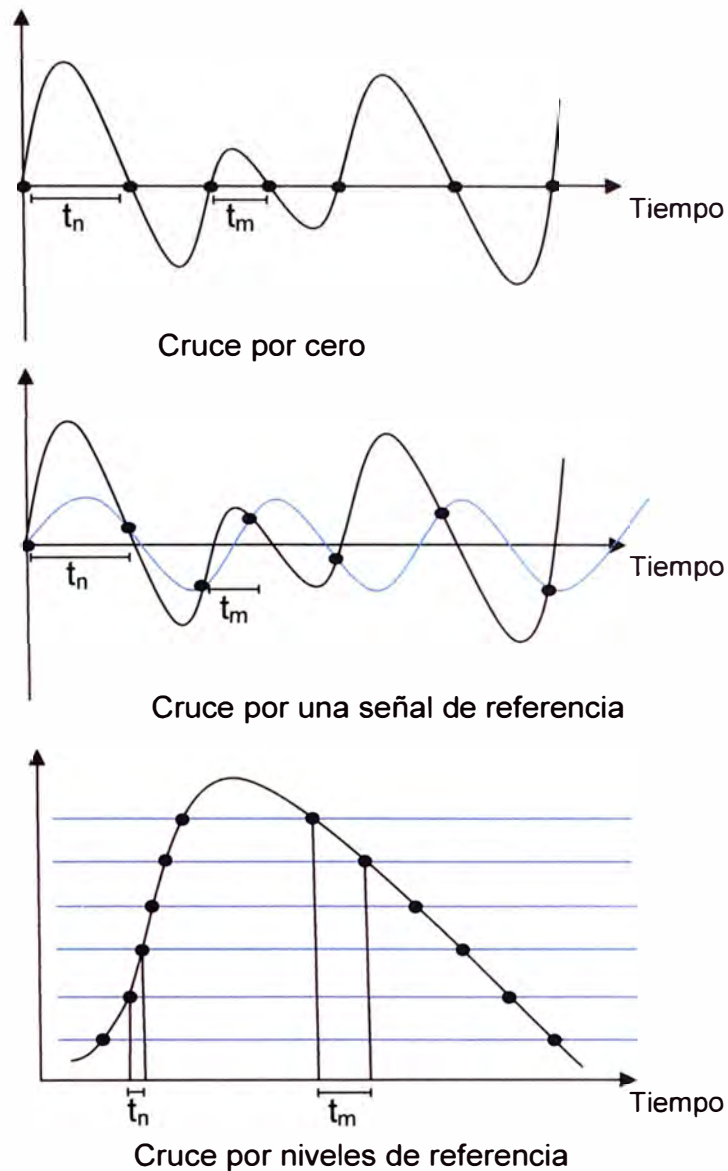
Sin embargo existen métodos alternativos de representación de una señal motivado por la naturaleza de los sistemas sensoriales, como son: temperatura, presión, sensores médicos, etc; caracterizados por ser señales con variaciones significativas solo en tiempo cortos. La idea es que el proceso de muestreo dependa de las propiedades locales de la señal, la toma de las muestras ya no se hace a partir de un periodo fijo si no del comportamiento de la señal frente a un tipo de referencia en el dominio de la amplitud, así las muestra son tomadas cada vez que la señal cruza la referencia escogida [14]. Los tres tipos de muestreo dependiente de la señal, según una referencia en el dominio de amplitud, como se grafica en la figura 3.1, son:

- a) Cruce por cero
- b) Cruce a una señal de referencia y
- c) Cruce por nivel

Los tres tipos tienen en común algunas características importantes como:

- Las muestras de la señal no están espaciadas uniformemente.
- Es imposible determinar o predecir el instante de tiempo de la muestra siguiente.
- La densidad de muestreo local depende de las características locales de la señal.





**Figura 3.1** – Diferentes tipos de muestreo no uniforme, donde  $t_n \neq t_m$

Aunque el muestreo periódico es una aproximación crucial en la mayoría de aplicaciones de procesamiento de señal, si el ambiente a ser monitoreado incluye señales "lentas", definido como aquellas señales que por naturaleza no cambian en una gran parte del tiempo, entonces enviar data periódicamente cuando nada significativo ha ocurrido se puede tomar como un desperdicio de recursos de procesamiento. Se considera así esta condición como un sobre muestreo, porque la variable a monitorear usualmente esta en equilibrio o con cambios poco considerables.

Uno de los tipos de muestreo no uniforme está basado en un detector de cruce de nivel y en la modulación sigma-delta asíncrona, la información acerca de la señal está contenida en los intervalos de tiempo entre cada muestra y en las direcciones de cada cruce, bajo este esquema se reduce la tasa de actividad, complejidad, envío de información, espacio de almacenamiento y consumo de potencia. El muestreo de cruce de nivel es una extensión del cruce por cero, que lleva información acerca de la

frecuencia fundamental de la señal. En el proceso digital de señales la reconstrucción de una señal es importante, para la recuperación de la señal en el muestreo de cruce por nivel se han desarrollado diferentes métodos que involucran funciones de reconstrucción asimétricas e iterativas, dependientes de la cantidad de niveles de referencia 'n' y del intervalo de tiempo ( $\Delta t = t_m - t_n$ ) entre cada muestra puesto que son las variables que llevan toda la información de la señal de entrada, entre las teorías desarrolladas una de ellas establece que la condición para que la reconstrucción sea posible es:

$$[f_{\text{muestreo}}]_{\text{prom}} \geq 2 \cdot f_{\text{max}} \quad (3.1)$$

Expresión conocida como la generalización del teorema de Shannon.

Donde:

$[f_{\text{muestreo}}]_{\text{prom}}$ : Frecuencia promedio de muestras tomadas.

$f_{\text{max}}$ : Frecuencia máxima de la señal de entrada

El proceso de muestreo implica modificaciones en la señal original y pérdida de información, para un muestreo por cruce de nivel de M bits se requiere  $2^M - 1$  niveles de referencia. Para un rango dinámico E, el paso entre cada nivel de referencia es:

$$q = \frac{E}{2^M - 1} \quad (3.2)$$

Donde:

q : Paso entre cada nivel de referencia.

E : Rango dinámico.

M: Número de bits.

La condición de seguimiento indica que, la variación de la señal debe ser menor o igual al paso de la señal 'q' entre el retardo del procesamiento 'δ'.

$$\left| \frac{dx(t)}{dt} \right| \leq \frac{q}{\delta} \quad (3.3)$$

El retardo de procesamiento ayuda al cálculo de la frecuencia máxima de la señal de entrada, de la expresión anterior, para una señal de entrada limitada en banda  $V_{in}$   $[0, f_{\text{max}}]$ , el teorema de Bernstein (4) indica:

$$\left| \frac{dV_{in}}{dt} \right| \leq 2\pi \cdot f_{\text{MAX}} \cdot \Delta V_{in} \quad (3.4)$$

Donde:

$\Delta V_{in}$ : rango dinámico

De (3.3) y (3.4) el retardo de procesamiento máximo es:

$$\delta_{\text{MAX}} = \frac{1}{2\pi \cdot f_{\text{MAX}} (2^M - 1)} \quad (3.5)$$

Entonces:

$$f_{\text{MAX}} = \frac{1}{2\pi \cdot \delta_{\text{MAX}} (2^M - 1)} \quad (3.6)$$



Donde:

Fmax: frecuencia máxima de entrada

δMAX: retardo máximo de procesamiento

M: numero de bits

Los intervalos de tiempo en entre cada cruce pueden ser contabilizados en base a un timer  $T_c$ , de esta forma la relación señal ruido (SNR) depende del periodo del timer y no del numero de bits, el SNR puede ser mejorado reduciendo el periodo  $T_c$  del timer. La salida del conversor en caso sea necesario una reconstrucción son dos datos  $(ax_n, dtx_n)$ , donde:  $dtx_n = kT_c$ ,  $k \in \mathbb{N}$ , la relación SNR está dada por:

$$SNR_{dB} = 10 \cdot \log \left( \frac{3 \cdot P(Vin)}{P\left(\frac{dvin}{dt}\right)} \right) + 20 \cdot \log \left( \frac{1}{T_c} \right) \quad (3.7)$$

Donde:

P(Vin): Potencia de la señal de entrada,

$T_c$ : Periodo del timer.

La primera parte de la ecuación (3.7) es constante, por lo tanto el SNR solo depende del periodo del timer. Dependiendo de si es necesaria o no la reconstrucción posterior de la señal, se usa ambos datos o solo el valor binario. El resultado de este tipo de conversor puede ser usado por otros módulos asíncronos o se pueden hacer adaptaciones para sistemas síncronos usando convertidores de tiempo digital TDC donde se cuantifica los intervalos de tiempo entre cada muestra mas no la amplitud de las mismas. En la tabla 3.1 se resume las características de los convertidores análogo digital síncronos y sus contrapartes asíncronos [15].

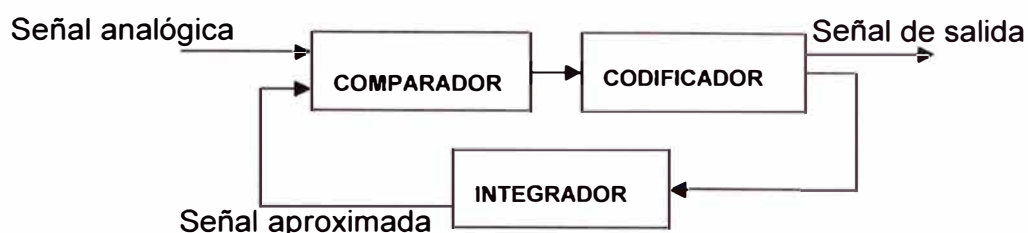
**Tabla 3.1** – Resumen de características de convertidores síncronos y asíncronos

	A/D sincrono	A/D asincrono
Captura de la muestra	clock	Cruce de nivel
Amplitud	Aproximado	Exacto
Tiempo	Exacto	Aproximado
SNR	Numero de bits	Resolución del timer
Salida del conversor	Vout	Vout y $\Delta t$

### 3.3. Modulación Sigma – Delta Síncrona

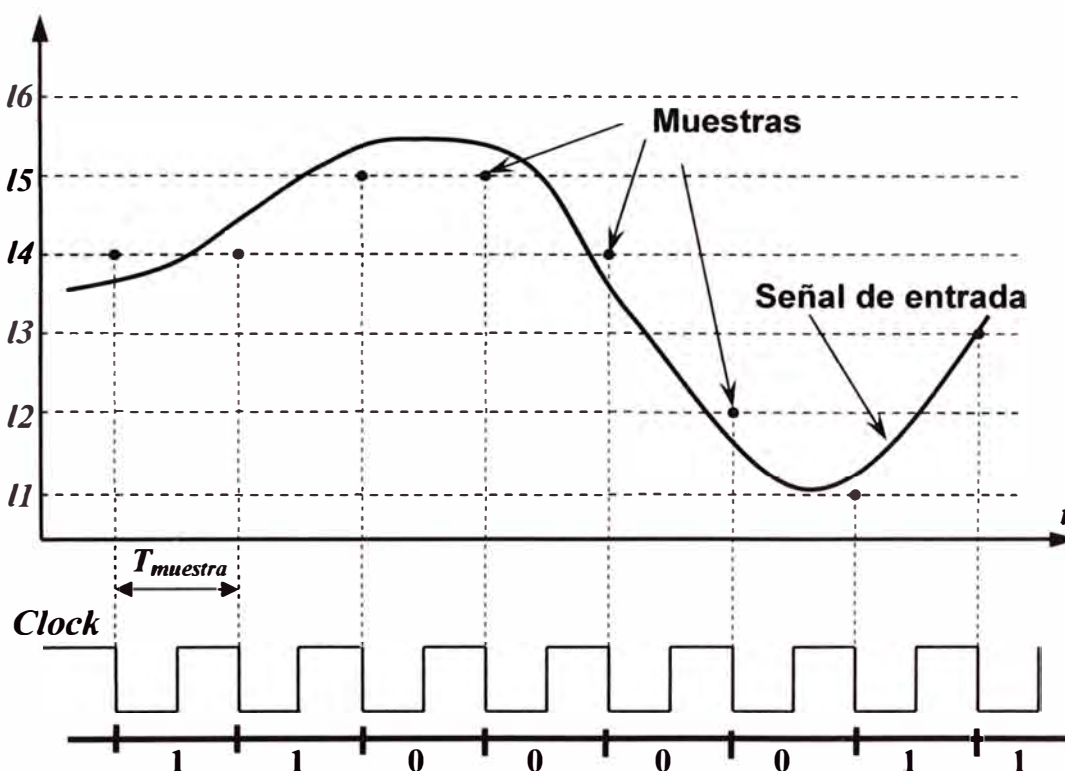
La modulación sigma-delta o convertidores de 1-bit son los sistemas que mejor se pueden adaptar a un muestreo de cruce por nivel. La modulación delta que es la base para la modulación sigma-delta, incluye básicamente, un comparador, un cuantificador de un bit y un acumulador, como indica la figura 3.2, aproxima la señal análoga con una serie de segmentos, luego cada uno de ellos comparado con la señal original para determinar si crece o decrece en amplitud; la diferencia es cuantificada en dos niveles

$\pm\Delta$ , dependiendo del signo de la diferencia, el valor de los bits está determinado por esta comparación y solo los cambios de información son transmitidos, es decir, se transmite el valor, '1' si la señal es de pendiente positiva y '0' si es negativa.



**Figura 3.2** – Diagrama de bloque para la modulación delta.

La señal obtenida es una sucesión de dígitos binarios los cuales sólo indican el signo de la diferencia entre la señal análoga y su aproximación, como muestra la figura 3.3.

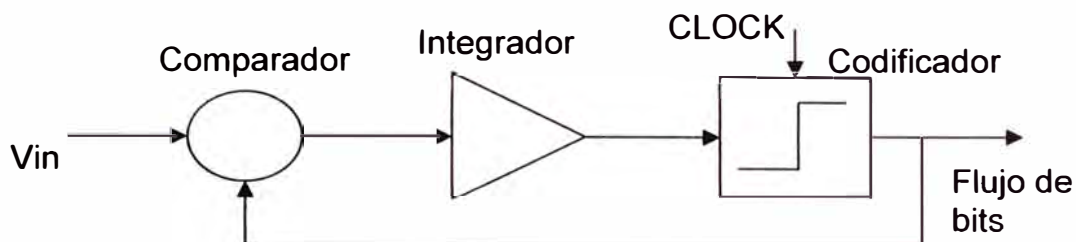


**Figura 3.3** – Modulación delta síncrona.

Los moduladores delta tiene como desventaja el “ruido granular”, que se presenta cuando la señal permanece estable con ligeras variaciones, lo que provoca una salida intercalada de ‘0’s y ‘1’s, otra desventaja es el “slope overload” (pendiente sobrecargada) para subidas rápidas de señales de entrada, que produce un alejamiento en la aproximación en la salida. El desempeño del modulador es por lo tanto dependiente de la frecuencia de la señal de entrada [16], [17].

Existe una solución para el problema de la pendiente sobrecargada, que se conoce como el modulador sigma-delta, como indica la figura 3.4, que consiste en integrar la señal de error para suavizarla y así controlar su pendiente, también este arreglo puede

muestrear a una frecuencia mayor que la mínima requerida, esto permite que la diferencia entre muestras sucesivas o adyacentes sea menor.

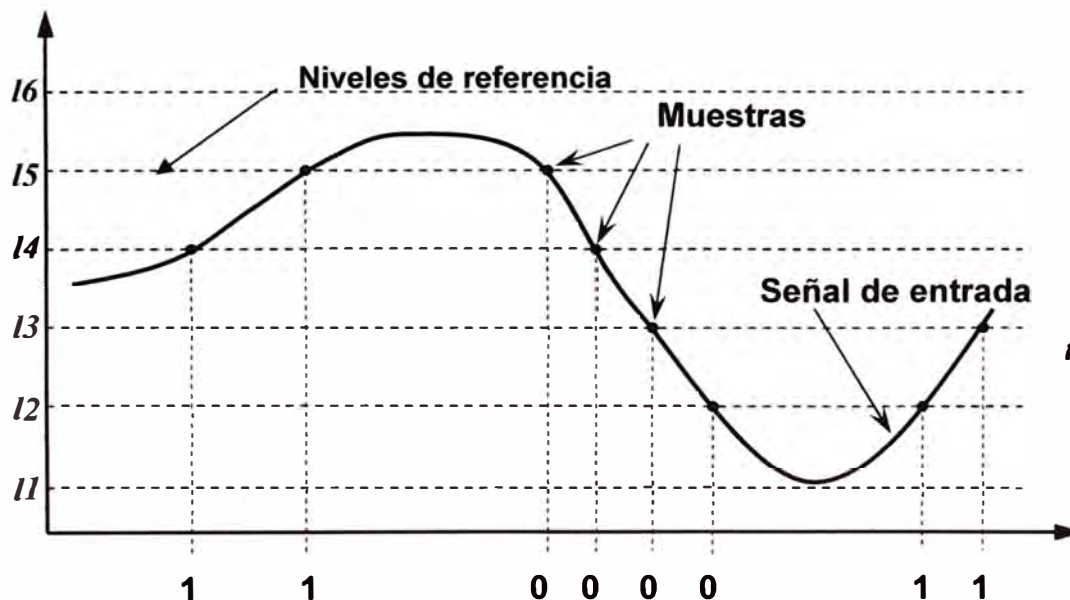


**Figura 3.4** – Modulación sigma - delta.

### 3.4. Modulación Sigma – Delta Asíncrona

La modulación sigma-delta asíncrona es un poco diferente a lo anterior, los pulsos en la salida se mantienen indicando las subidas o descensos de la señal, pero ya no usa el esquema de la figura 3.4, ni el teorema de Nyquist, si no usa el concepto de muestreo por cruce de nivel enviando un pulso cada vez que la señal cruza algún nivel de referencia hacia arriba o hacia abajo, como se detalla en la figura 3.5.

La salida es '1' cuando la pendiente de la señal es positiva y es '0' cuando la pendiente es negativa, esto puede ser enviado usando bundled data o dual rail. Esta forma de interpretación es más cercana a la naturaleza de la señal

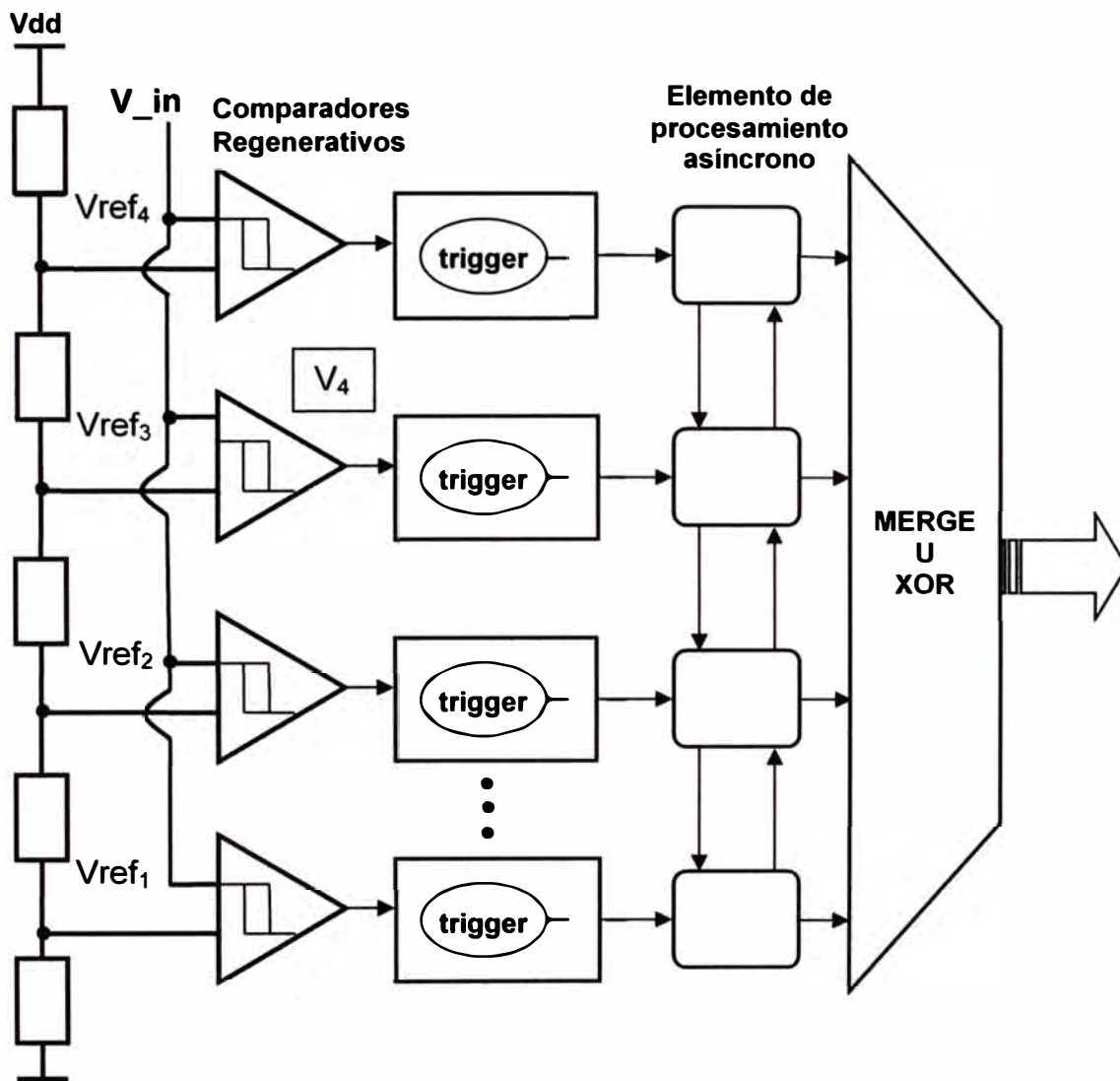


**Figura 3.5** – Modulación delta asíncrona.

De esta forma ante cambios rápidos y significantes de la señal, se envía una cadena de pulsos y si la señal está en un estado cuasi-estacionario, se envía una cantidad reducida de data. Los pulsos enviados contienen suficiente información para un procesamiento posterior, aunque la forma de la data no es convencional con los sistemas actuales, sin embargo, existen interfaces o se puede incluir un timer como forma de cuantificar el tiempo entre cada muestra [18], [19].

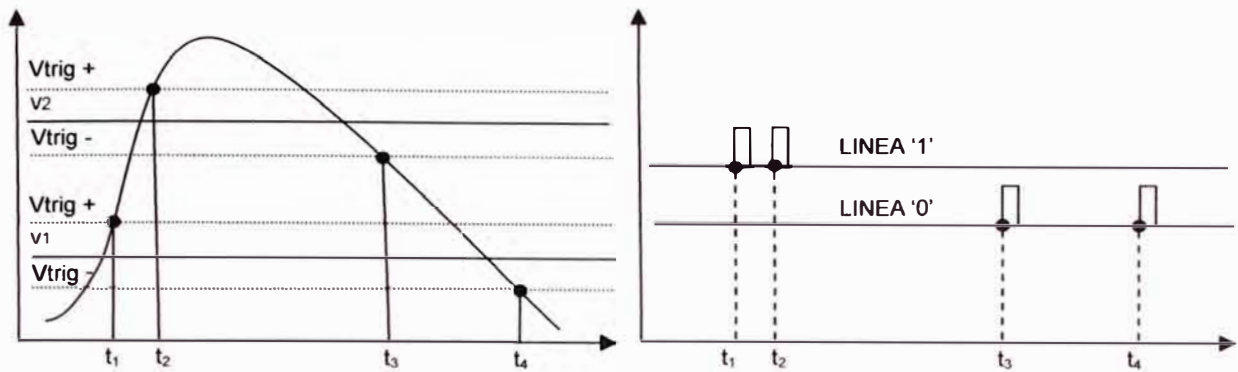
### 3.5. Conversor análogo digital asíncrono

El circuito a desarrollar tiene las siguientes partes: Un divisor de voltaje que origina los voltajes de referencia, luego por cada nivel de referencia existe; un comparador análogo, un disparador digital (trigger) y un bloque de procesamiento digital asíncrono, que incluye el elemento de transmisión y almacenamiento del token. Finalmente el esquema de la figura 3.6 incluye un bloque "merge" que produce el flujo de salida (presenta los pulsos '1' y '0' como una modulación delta asíncrona vía dual rail) [22]. Adicionalmente se puede incluir un contador ascendente y descendente para producir los códigos binarios.



**Figura 3.6** – Esquema general del convertidor análogo digital asíncrono.

El comparador análogo es la primera etapa del circuito, recibe la señal análoga y la compara con los niveles de referencia prefijados. La salida de este circuito controla al disparador digital (trigger) que envía un requerimiento al modulo asíncrono para procesar la data muestreada, cada bit en la salida representa un cruce por nivel. Si el nivel fue cruzado por la señal de entrada desde abajo hacia arriba, el circuito envía un "1" sobre el canal de salida dual-rail y si la señal de entrada cruza el nivel desde arriba hacia abajo, el circuito produce un "0" sobre el mismo canal dual-rail como muestra la figura 3.7.

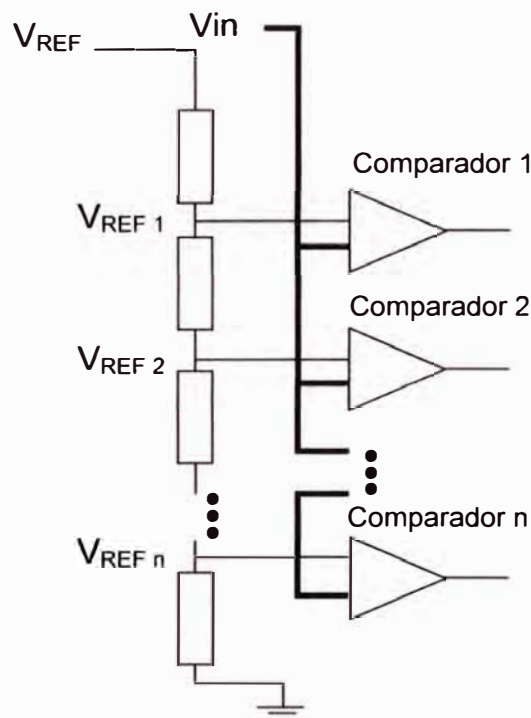


**Figura 3.7** – Muestreo no uniforme con rango de histéresis y datos DUAL RAIL de salida del convertidor.

A continuación se desarrolla cada parte del circuito en detalle:

### 3.5.1. Parte Análoga: Comparador Regenerativo

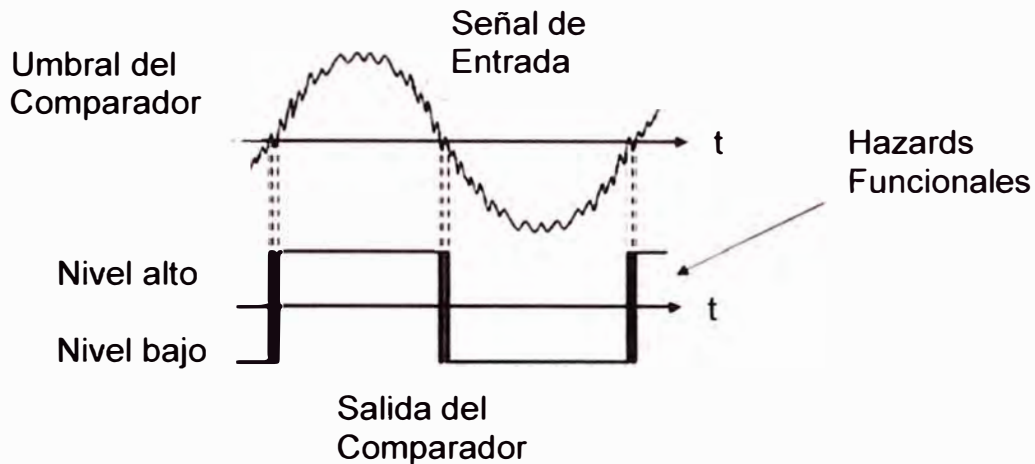
La parte análoga del convertidor es muy similar al de un convesor A/D tipo flash, es decir, consta de un divisor de voltaje para obtener los niveles de referencia y una cadena de comparadores, como indica la figura 3.8.



**Figura 3.8** – Uso del comparador análogo dentro del modulo del convesor.

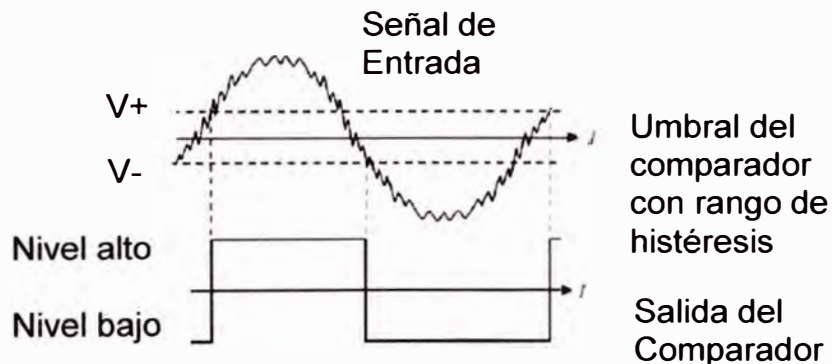
El uso de los comparadores impone un límite en la resolución y potencia, puesto que, a mayor número de niveles de referencia aumenta el número de comparadores y resistencias para divisor de voltaje lo cual eleva la potencia del convesor, sin embargo la diferencia para equilibrar este aspecto está en la parte digital. La función del comparador análogo es convertir cambios de la señal de entrada, llámese cruces de la señal a los niveles de referencia, a un nivel de valor lógico con flancos finos para alimentar a la parte digital, conocido también como código termométrico.

El principal requisito para el comparador es que la respuesta en el tiempo sea menor a la variación de la señal de entrada alrededor del nivel umbral, para evitar que la salida fluctúe entre valores lógicos 'alto' y 'bajo'. Una señal fluctuante se puede presentar si el ambiente es ruidoso o si hay interferencia alta desde el resto del circuito, teniendo como consecuencia la presencia de valores lógicos no estables, a este tipo de fenómeno se le conoce como "hazards", más precisamente son hazards funcionales que afectan el procesamiento posterior, la figura 3.9 ejemplifica este concepto.



**Figura 3.9** – Comparador sin histéresis

Estas dificultades pueden ser eliminadas empleando realimentación positiva que fuerza al comparador a comportarse como un latch regenerativo, es decir, el circuito tendrá un rango de histéresis, el comportamiento de un comparador con rango de histéresis es mostrado en la figura 3.10.

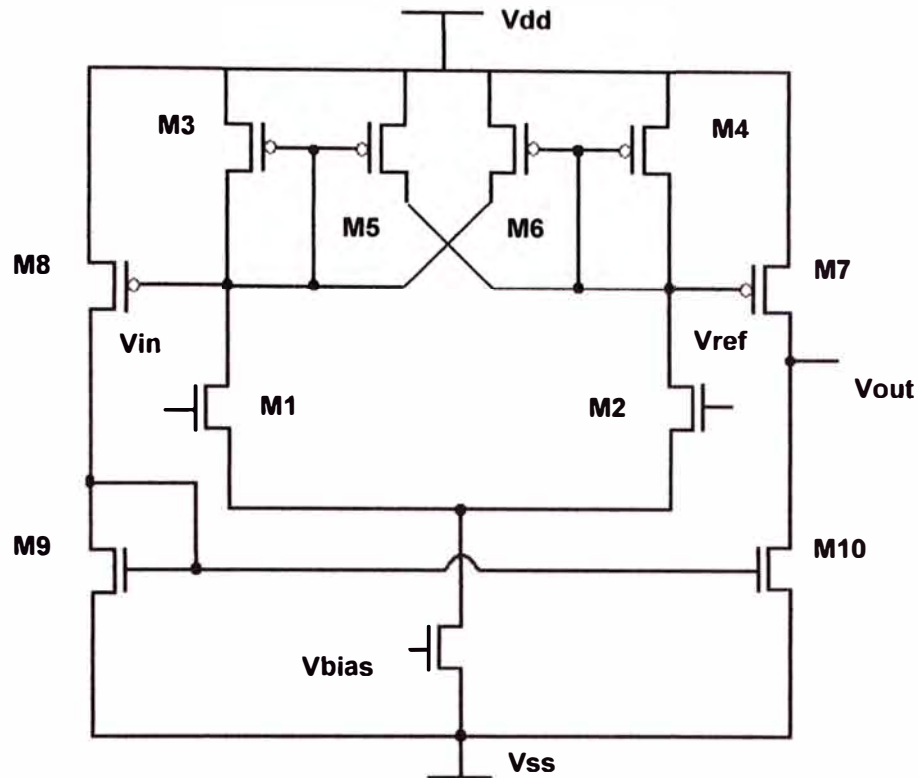


**Figura 3.10** – Comparador con histéresis.

### Topología del comparador análogo

La parte análoga que produce los niveles de referencia, usa un divisor de voltaje en base a resistencias o capacitancias y los comparadores por cada nivel de referencia se encargan de producir el código termométrico. Entre los diferentes diseños para un comparador regenerativo se presenta una topología en la figura 3.11, donde solo se usa once transistores y no hay resistencias.





**Figura 3.11** – Topología del comparador análogo con histéresis.

Las características de ganancia y rango de histéresis dependen exclusivamente de las propiedades de dimensionamiento del CMOS y voltaje de alimentación. Si se desea que la salida sea un nivel alto '1' cuando la señal análoga sobrepasa el nivel de referencia, este deberá tener como entrada al circuito el elemento M1 y la señal análoga al elemento M2, puesto que es la diferencia positiva  $V_{M2} - V_{M1}$  es la que produce valores altos a la salida. En caso contrario, es decir, el nivel de referencia por M2 y la señal análoga por M1, el comparador producirá salida baja '0' cuando la señal análoga sobrepasa el nivel de referencia.

Para el conversor análogo digital se usa la configuración de la figura 3.11 para obtener valores altos '1' cuando la señal cruza el nivel de referencia y salidas bajas '0' cuando no lo ha hecho. El diseño está basado en el par diferencial M1 y M2, el transistor bias M0, los dispositivos de carga M3 y M4 y la pareja de cruce M5 y M6. El único propósito del espejo de corriente y de los transistores M7, M8, M9, M10 es proveer ganancia adicional para flancos de salida más finos.

### **Funciones de los transistores en el comparador**

El transistor bias M0 fija la corriente que fluye a través del circuito. A corriente más alta, mayor potencia es consumida, la impedancia de salida disminuye y un ancho de banda mayor puede ser alcanzado. M3 y M4, junto con M1 y M2, fijan la ganancia de la primera parte: a más grandes los dispositivos de carga, más grande es la ganancia y



mayor capacitancia parasita es introducida. M5 y M6 proveen realimentación positiva y determinan la cantidad de histéresis para el comparador.

Alta precisión de operación puede ser alcanzada solo si los transistores son apropiadamente encajados o ajustados. De otro modo, los puntos que fijan el intervalo de histéresis del circuito, así como la ganancia y el ancho de banda pueden ser no muy convenientes dependiendo de la aplicación.

Un buen encaje es alcanzado por la propiedad de dimensionamiento del transistor y técnicas de layout. El ancho de banda del comparador puede ser mejorado incrementando el ancho de los transistores M1 y M2 a la vez que los dispositivos de carga son encogidos (para mantener la misma ganancia del comparador), también se incrementa el voltaje de bias en la compuerta M0, a costas de elevar la potencia de consumo. Los límites que fijan la propiedad de histéresis del circuito son determinados principalmente por la proporción de M5 a M3 (M6 a M4 tienen que ser parejos). La ganancia diferencial es grande debido a la relativa alta resistencia de salida del circuito.

Las fórmulas que determinan los puntos de conmutación del comparador y especifican la cantidad de histéresis presente en el circuito son:

$$V_{trig+} = \sqrt{\frac{i_0}{K'(W/L)_1}} \times \frac{\sqrt{\alpha}-1}{\sqrt{1+\alpha}} \quad (3.8)$$

$$V_{trig-} = \sqrt{\frac{i_0}{K'(W/L)_1}} \times \frac{1-\sqrt{\alpha}}{\sqrt{1+\alpha}} \quad (3.9)$$

$$\alpha = \frac{(W/L)_5}{(W/L)_3} = \frac{(W/L)_6}{(W/L)_4} \quad (3.10)$$

$$K' = \frac{1}{2} \cdot \mu \cdot C_{ox} \quad (3.11)$$

Donde:

W/L: proporción del ancho al largo del transistor.

C<sub>ox</sub>: capacitancia oxido.

μ: movilidad de huecos.

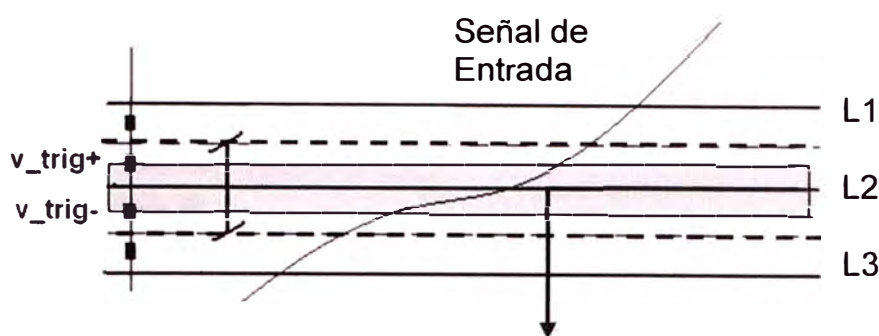
α: factor de realimentación positiva.

Las expresiones (3.8), (3.9), (3.10) y (3.11) indican que el comportamiento del comparador está principalmente determinado por las dimensiones de M1 y su simétrico M2, M5, M3, M6, M4 y el voltaje bias que determina  $i_0$  que es la suma de las corrientes  $i_1$  e  $i_2$  a través de M1 y M2 respectivamente.

El rango de histéresis, está en función de la proporción de los tamaños de la pareja de cruce a los transistores de carga y de los transistores de entrada.

Para obtener comportamiento regenerativo el valor de α tiene que ser mayor que 1 y dependerá de la aplicación (según el nivel de ruido del ambiente). Pero existe un límite

superior, bajo operación normal la cantidad de histéresis sobre uno de los lados del nivel de referencia no deberá exceder la mitad del voltaje diferencial entre dos niveles adyacentes, como muestra la figura 3.12.



Máximo alcance del intervalo de histéresis

**Figura 3.12**– Rango máximo de histéresis

Los transistores en el comparador deben ser cuidadosamente dimensionados, transistores muy pequeños son susceptibles a un pobre emparejamiento, lo cual puede resultar en una no linealidad en la salida, transistores muy grandes presentan capacidad parasita significativa lo cual puede limitar la velocidad del circuito. Hay un natural equilibrio entre el ancho de banda del comparador y el transistor dimensionado.

La corriente a través del comparador puede variar externamente si se desea, de ahí que se podría enviar el nodo de voltaje de bias a la salida del chip para controlar el ancho de banda del comparador [20].

### 3.5.2. Parte Digital: Lógica Digital Asíncrona

La parte digital del conversor es implementada usando la lógica asíncrona, el principal propósito es reconocer los cruces de la señal analógica a los niveles prefijados y producir una adecuada data de salida. Se usa una canal "dual rail", de esta manera la salida representa el cambio de la señal de entrada con respecto a la muestra previa. A continuación se presenta y se describe los diferentes componentes de la parte digital del conversor.

#### Bloque disparador digital (TRIGGER)

El elemento TRIGGER (disparador digital) es el nexo entre la parte analógica y la digital, controla la comunicación entre el comparador y el elemento asíncrono, como se ve en la figura 3.13, verificando el cumplimiento o no de las condiciones de cruce para dar paso luego a la siguiente etapa activando o "despertando" al bloque de procesamiento asíncrono y espera la respuesta de este último para responder a otro cambio.

#### Descripción del bloque

Cuando el disparador o trigger digital identifica el cambio en la salida del comparador, revisa las variables de estado internas del elemento asíncrono que le corresponde, del anterior y del siguiente, si todas las variables indican que el cruce ha ocurrido, envía un

requerimiento mediante un "handshake" a su correspondiente elemento de procesamiento asíncrono el cual luego genera la salida, el TRIGGER sostiene el requerimiento hasta que sea liberado mediante una señal de habilitación para volver a atender otra condición de cruce.

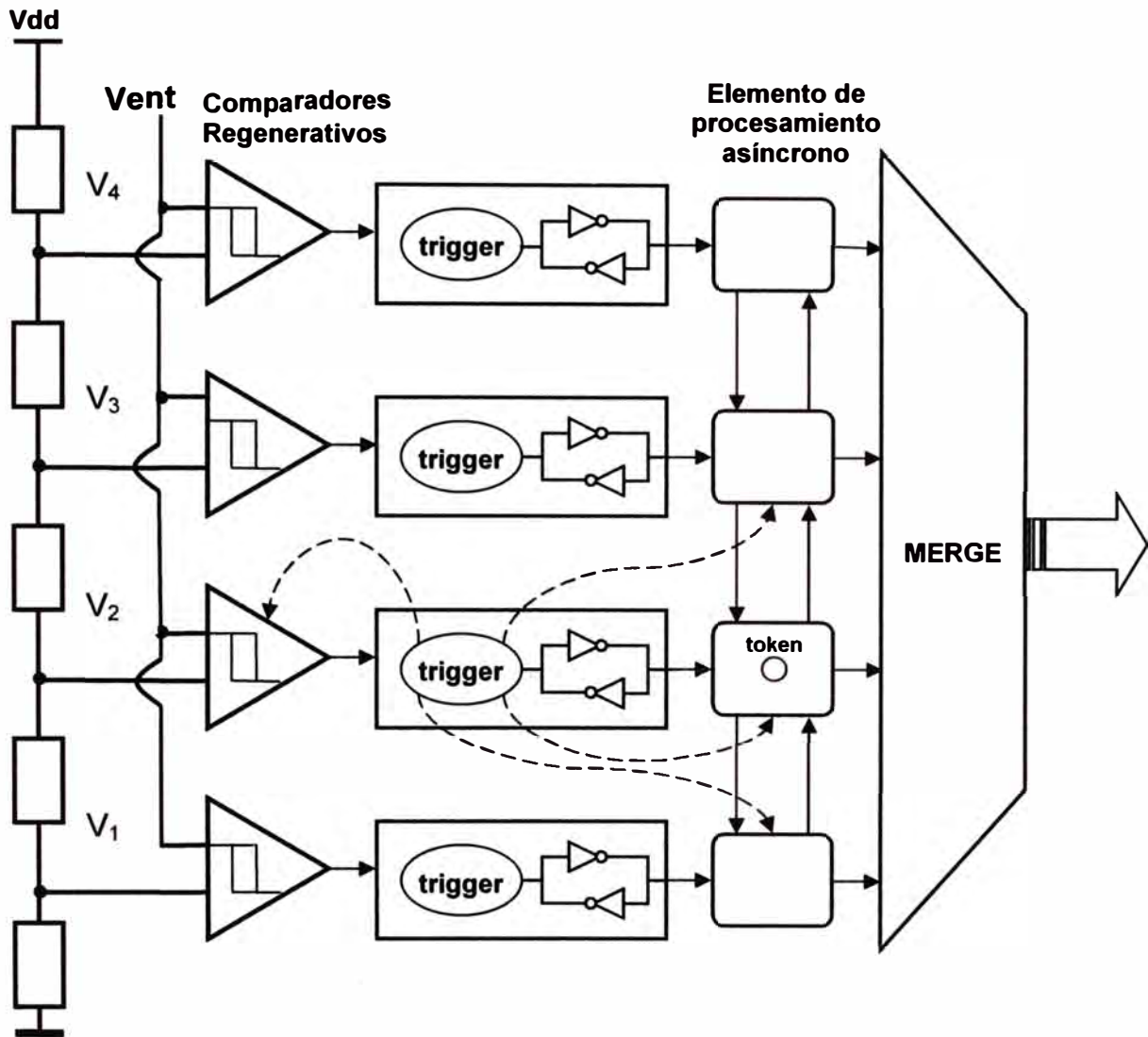


Figura 3.13 – Bloque TRIGGER.

Al momento de encender el dispositivo o después de un "reset" todos los bloques TRIGGER no presentan requerimiento alguno. El elemento TRIGGER emite un requerimiento si los estado internos tanto del bloque que le corresponde como de los adyacentes, cumplen con las siguientes condiciones tanto para un cruce hacia arriba como hacia abajo.

#### 1. Primer Caso.- Cuando la señal cruza hacia arriba un nivel de referencia

- Salida del comparador en alto "1"
- Elemento de procesamiento (i) indica que la señal estaba por debajo antes:  $k=0$ .
- Elemento de procesamiento (i-1) indica que la señal está arriba:  $k=1$ .
- Elemento de procesamiento (i-1) ha completado su tarea:  $ACK(i-1)=1$ .
- Elemento de procesamiento (i) ha completado su tarea:  $ENABLE = 1$ .

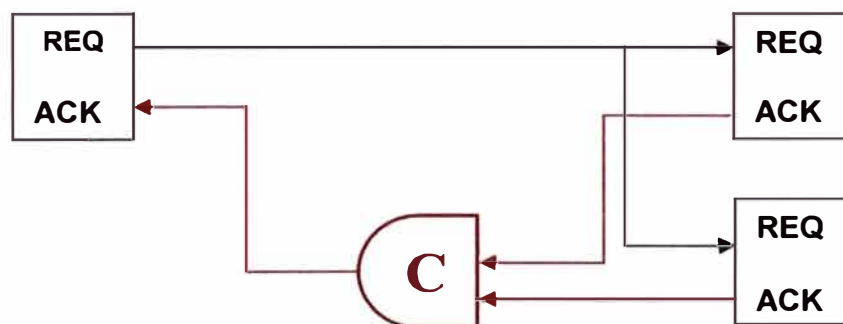
## 2. Segundo caso.- Cuando la señal cruza hacia abajo un nivel de referencia

- Salida del comparador en bajo "0"
- Elemento de procesamiento (i) indica que la señal estaba por encima antes:  $k=1$ .
- Elemento de procesamiento (i+1) indica que la señal esta debajo:  $k=0$ .
- Elemento de procesamiento (i+1) ha completado su tarea:  $done=1$ .
- Elemento de procesamiento (i) ha completado el tarea:  $ENABLE = 1$ .

El requerimiento para la unidad de procesamiento asíncrono debe permanecer estable, aun en el caso de un cambio en la señal análoga de entrada, hasta que la unidad de procesamiento termine de atender el requerimiento. El requerimiento es considerado atendido cuando se produce un dato en la salida de la unidad. El requerimiento es luego despejado automáticamente por la señal ACK del bloque de salida Merge, que se traduce en una señal ENABLE cuando llega al bloque TRIGGER.

El TRIGGER es un elemento que depende de muchas otras condiciones, es decir, reúne el valor de varias variables para producir una salida lógica, esto en el entorno asíncrono se puede implementar usando el elemento 'C' y los conceptos Grouping (agrupar) y Separating (separar). De ambos el primero genera circuitos con menos consumo de potencia y área mientras que el segundo genera circuitos con menos sincronización.

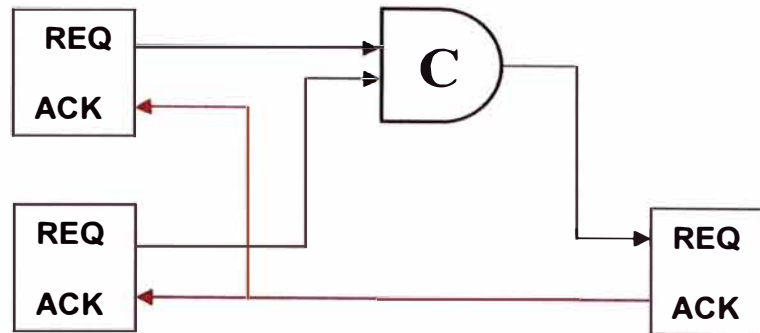
La transmisión de una señal a dos o más escenarios requiere que la fuente de la señal interactúe correctamente con más de un destinatario. Un caso es cuando la señal REQ necesita ser dividida para varios destinatarios, la señal ACK de todos los receptores es agrupada para generar una sola señal ACK que la fuente de la señal espera.



**Figura 3.14** – Elemento C para agrupar señales ACK.

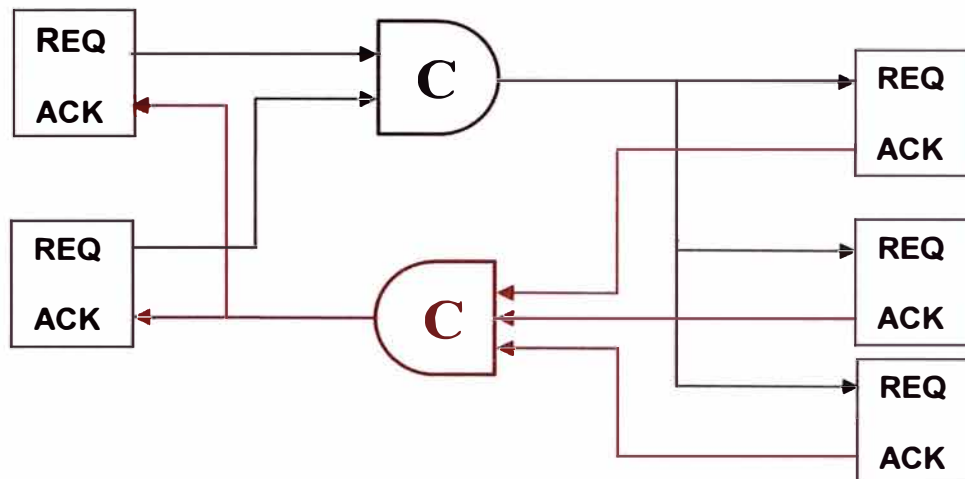
Un elemento C puede ser usado para combinar todas las señales ACK, de manera que se puede asegurar que todos los destinatarios han aceptado la data, como se muestra en la figura 3.14.

Se puede dar también el caso inverso al anterior donde, primero las señales REQ son reunidas (usando el elemento C) antes de ser enviadas al siguiente modulo, así el receptor recibirá un REQ si todas las entradas ya presentaron sus respectivos REQs y el receptor responderá a la señal REQ enviando a cada transmisor una señal ACK, como se aprecia en la figura 3.15.



**Figura 3.15** – Elemento C para agrupar señales REQ.

La combinación de los casos anteriores también es posible, es decir, tanto las señales REQ de varios transmisores necesitan ser agrupadas y las repuestas ACKs de varios transmisores también, igual que en los casos descritos anteriormente el elemento C es muy útil para estos casos, esta combinación se presenta en la figura 3.16.



**Figura 3.16** – Elemento C para agrupación combinada de REQ y ACK.

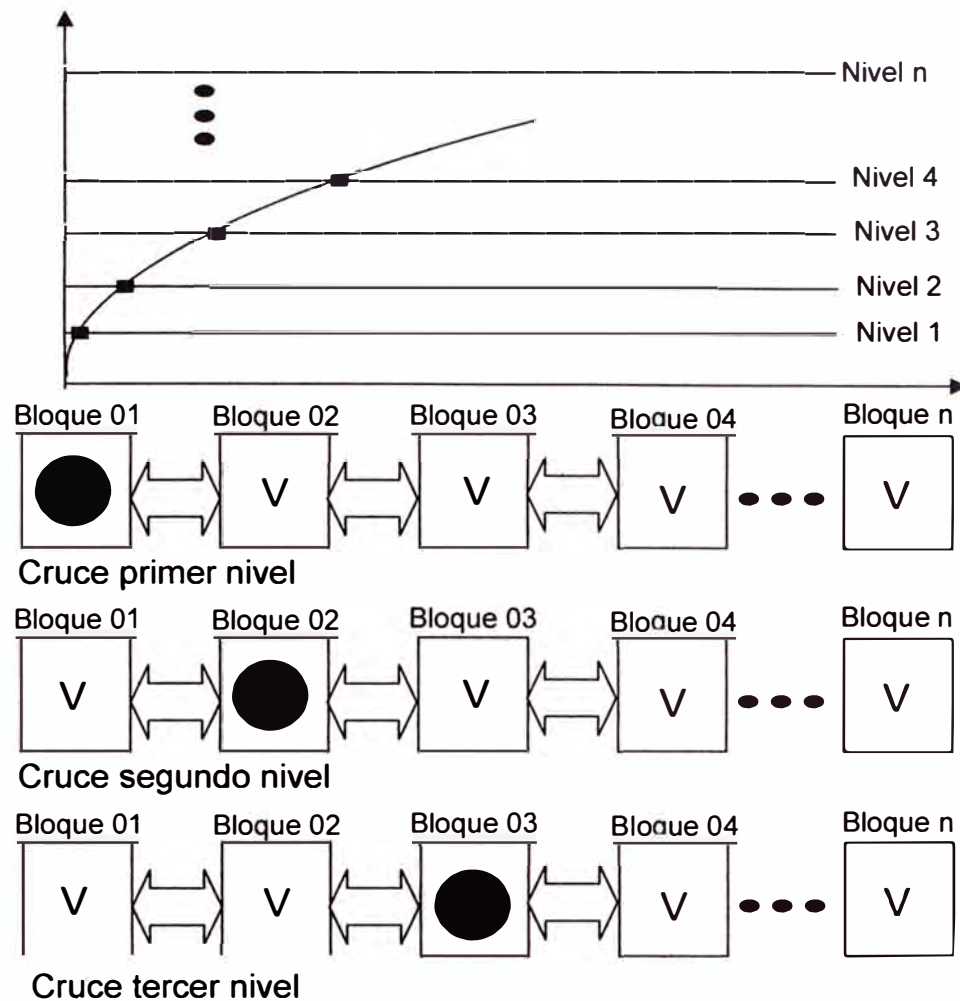
En general se pueden agrupar las señales de control según la dependencia entre módulos, el elemento C permite agrupar las señales de control tanto REQ como ACK.

### **Bloque de procesamiento Asíncrono**

Es el bloque principal de todo el convertidor, existe uno por cada nivel, están intercomunicados entre sí para poder enviar y recibir el token según el movimiento de la señal de entrada como se observa en la figura 3.17. es decir, a cada cruce de un nivel de referencia el token es transmitido a su correspondiente elemento digital asíncrono.

El elemento de procesamiento asíncrono controla las variables internas que indican el sentido de los cruces y producen las salidas correspondientes hacia el elemento Merge, son los encargados también de manejar el único token que existe en todo el circuito, la condición para que cualquier bloque tenga el token tiene que ver con el cruce de la señal analógica de entrada a los niveles de referencia, cada cruce produce un requerimiento de proceso dado primero por el comparador y después por el trigger, el elemento de procesamiento asíncrono para atender el requerimiento y poder producir la salida correspondiente '0' o '1', necesita tener en su elemento de memoria el token, que en

caso no lo tenga deberá pedirlo a sus bloques vecinos. Es la presencia del token la que garantiza la mutua exclusión de los bloques a la hora de atender los requerimientos.



**Figura 3.17 –** Movimiento del token en los elementos de procesamiento asíncrono según cambios en la señal de entrada.

En condiciones iniciales el único token del circuito digital asíncrono se encuentra en el primer nivel inferior.

#### **Descripción del bloque de procesamiento asíncrono**

Cada elemento tiene dos variables locales 't' y 'k', 't' indica la presencia o no del token en el elemento y 'k' indica dos cosas según sea el caso, primero si el bloque tiene el token 'k' indica el sentido del cruce de la señal y el segundo caso es cuando el elemento no tiene el token, ahí k indica la ubicación del token, es decir, si el token está por encima o por debajo. El elemento de procesamiento asíncrono tiene 6 canales de comunicación, como indica la figura 3.18, dos para la entrada y salida de la data y los 04 restantes para la comunicación entre bloques y traspaso del TOKEN de bloque a bloque, sin embargo cada canal de comunicación cuenta a su vez con dos sub canales para cumplir con el protocolo de comunicación asíncrono, como muestra la figura 3.19.



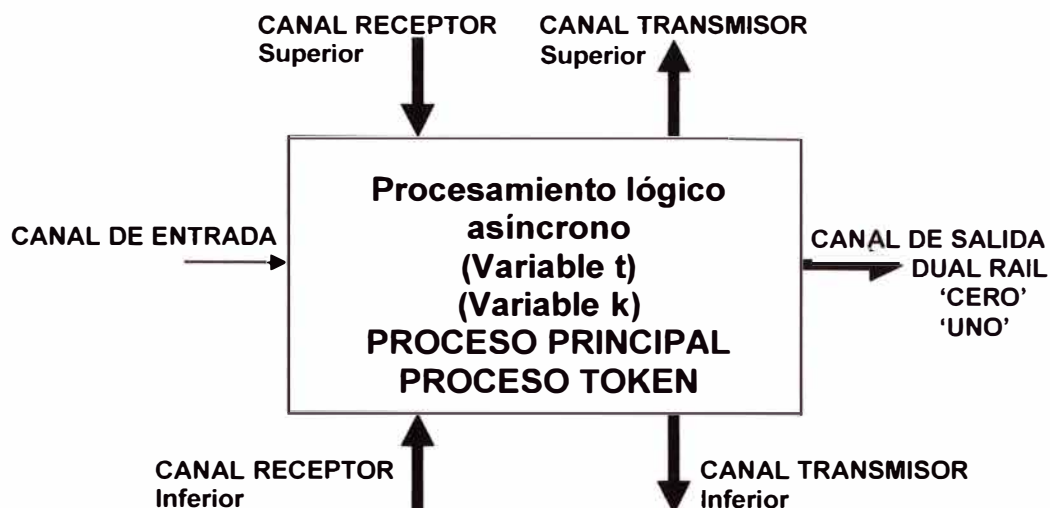


Figura 3.18 – Bloque de procesamiento asíncrono.

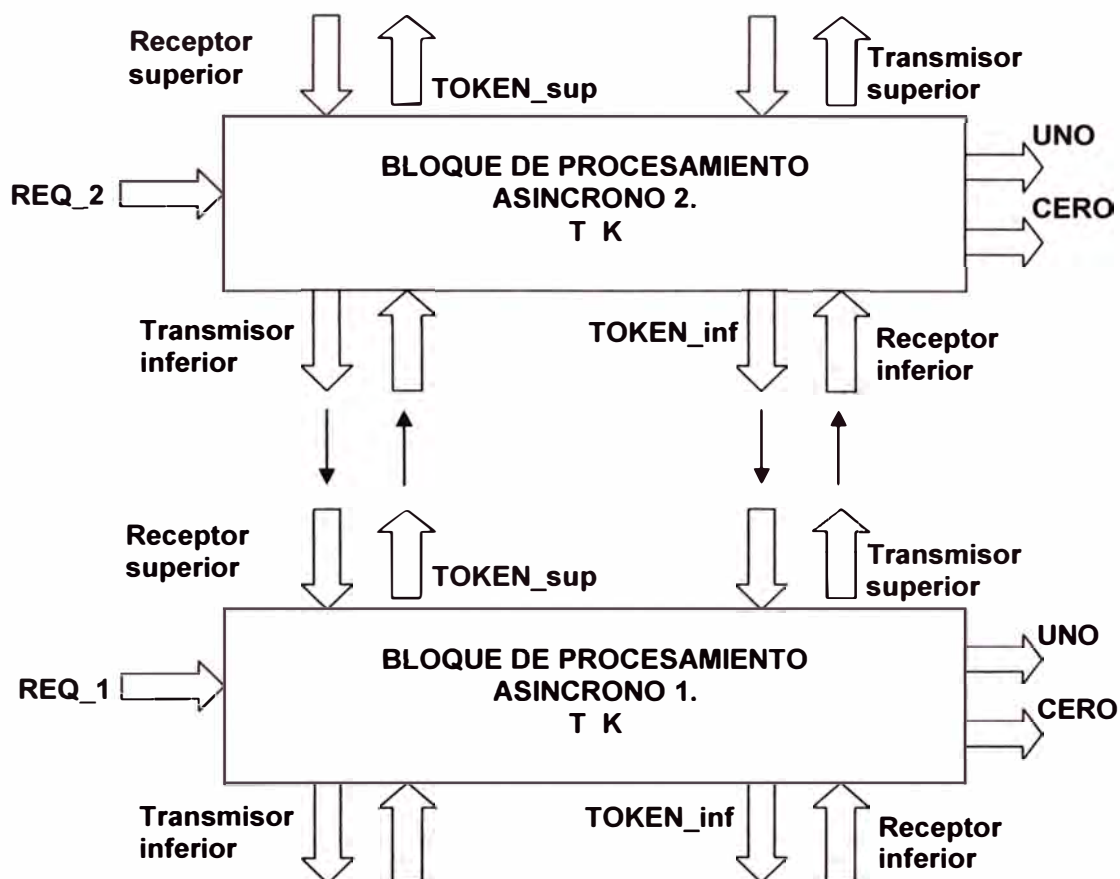


Figura 3.19 – Bloques de procesamiento asíncrono y canales de comunicación.

Los procesos que realiza el elemento de procesamiento asíncrono son:

Toda la comunicación entre bloques asíncronos para el traspaso del token y con el elemento Merge, es realizada de manera asíncrona, usando el protocolo a 04 fases, transmisión de data dual rail y modelo de retardo delay insensitive.

### 1. TOKEN CHECK

El proceso de TOKEN CHECK, revisa si el elemento actual tiene el TOKEN del sistema, si la respuesta es verdadera ( $t = '1'$ ) termina el proceso y pasa al siguiente paso,



pero si el resultado no lo es ( $t=0$ ), es necesario pedir el token a uno de los elementos asíncronos vecinos, para lo cual se basa en el valor de 'k':

- si ( $k=1$ ) pide el token al elemento superior,  $Send\_sup='1'$  y
- si ( $k=0$ ) pide el token al elemento inferior,  $Send\_inf='1'$

Mediante un handshake con uno de sus elementos vecinos, el elemento actual recibe el token y está listo para continuar con el proceso.

## 2. PROCESO PRINCIPAL

Dentro del proceso principal existen dos tipos según cuál de las entradas ha sido activada, si el requerimiento viene del elemento TRIGGER se pasa al proceso TOKEN CHECK y una vez que elemento posee el token, está apto para producir la salida según el valor de 'k':

- si ( $k=1$ ) la dirección de cruce es descendente CERO = '1' y
- si ( $k=0$ ) la dirección de cruce es ascendente UNO = '0'

Los datos son enviado al elemento MERGE por un canal DUAL RAIL y un proceso de handshake, después del cual, el elemento de procesamiento asíncrono cambia el valor de 'k' por su opuesto y envía la señal ACK al elemento TRIGGER, indicando que está listo para atender otro requerimiento.

Si el requerimiento viene de las señales de entrada 'receptor superior' o 'receptor inferior', quiere decir que el elemento superior o inferior están solicitando el token, para lo cual el elemento realiza el proceso de TOKEN CHECK y un vez que tiene el token lo envía mediante un handshake por el canal correspondiente ( $toke\_sup$  o  $token\_inf$ ) y se queda con la variable interna  $t=0$  para finalizar el proceso.

### Bloque Merge

Es la parte final del conversor en su etapa de producir la modulación sigma-delta asíncrona, recoge los datos producidos por los bloques asíncronos y producir los pulsos sobre el canal DUAL RAIL para su transmisión al escenario siguiente, indica también, usando un handshake, el fin del proceso iniciado al producirse un cruce de nivel a su elemento de procesamiento asíncronos correspondiente, a fin de que el sistema este apto a atender otra solicitud de cruce de nivel. Está formando básicamente por elementos XOR separados para datos tipo '1' y '0' que reciben las entradas y las direccionan a un solo canal tipo '1' y '0' general en forma de pulsos, de ancho regulable según un elemento de retardo. Como paso final se implementa, seguido al bloque MERGE, un contador binario ascendente y descendente, como se señala en la figura 3.20, que produce el código binario que puede ir acompañado, según sea el caso, de un timer para una posible reconstrucción de la señal posterior. La transmisión de los datos finales puede ser síncrona a asíncrona dependiendo del escenario siguiente en el proceso digital completo del sistema.

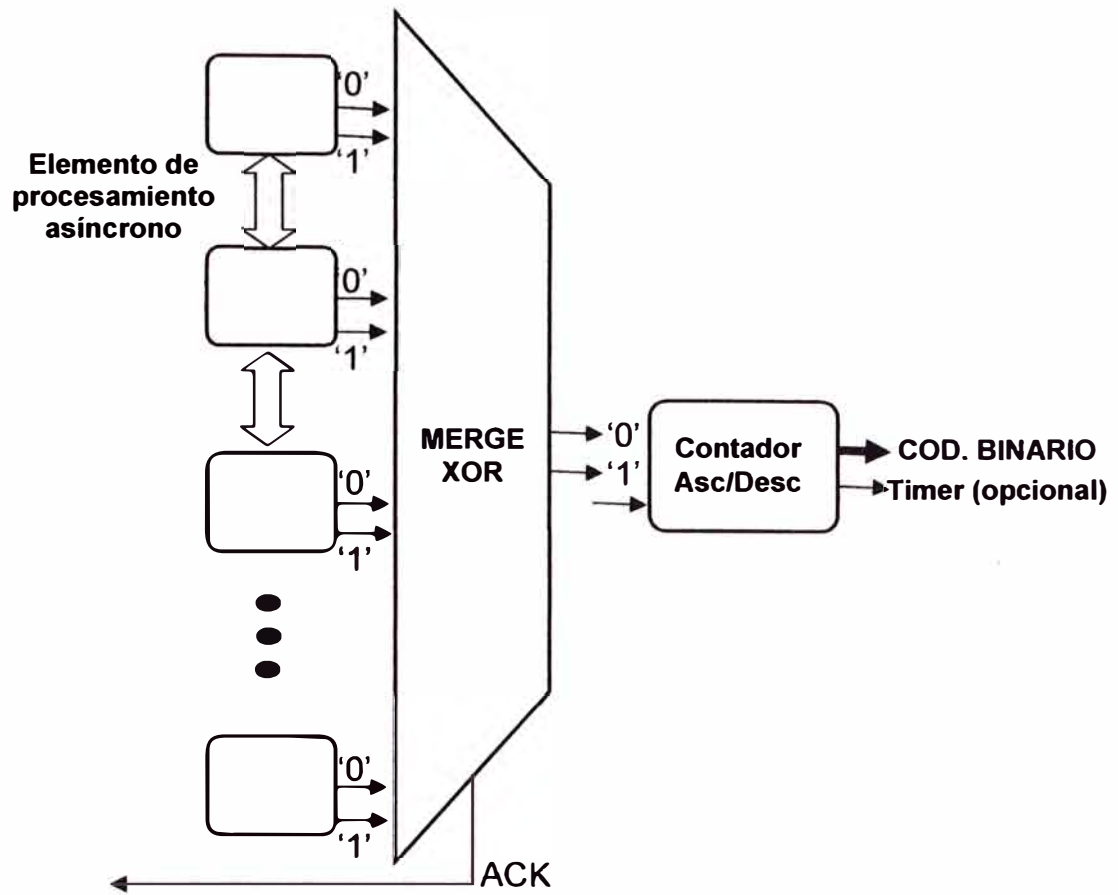


Figura 3.20 – Bloque Merge y contador ascendente y descendente

## CAPITULO IV ANALISIS Y PRESENTACION DE RESULTADOS

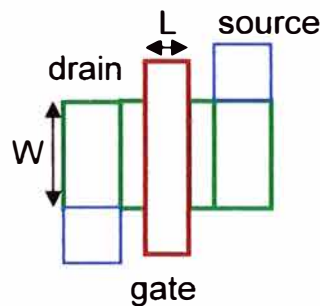
### 4.1. Introducción

En este capítulo se realiza la simulación y análisis de los componentes de manera que puede ser adaptable para cualquier sensor de temperatura ambiental. El convertidor a desarrollar será de 03 bits de resolución. Se usa para las simulaciones del comparador análogo el software Microwind 3.1 y para la parte digital el software Quartus II de Altera.

### 4.2. Simulación de la sección analógica

#### 4.2.1. Comparador regenerativo con histéresis

El software Microwind 3.1, es un programa que permite diseñar circuitos a nivel micro electrónico en diferentes tecnologías, donde los datos de ancho (W) y largo (L) del transistor se indican en la figura 4.1. Como el diseño del comparador está sujeto al tamaño de los transistores, tenemos varios casos, según las formulas (3.8), (3.9), (3.10), (3.11), definidas en el capítulo 3:



**Figura 4.1** – Parámetros W y L para diseño microelectrónico.

A continuación se comprueban los diferentes casos, donde:

Se empieza con la tecnología de 25 $\mu$ m, por la familiaridad de los valores  $V_{dd} = 2.5v$  y  $V_T=0.7$ , al momento de cambiar de referencia de tamaño, el software se encarga de reducir los valores de voltaje y corriente, manteniendo el comportamiento del circuito.

#### 1. Elementos CMOS de igual tamaño.

Al usar elementos CMOS de igual tamaño y reemplazando en la (3.10), obtenemos:

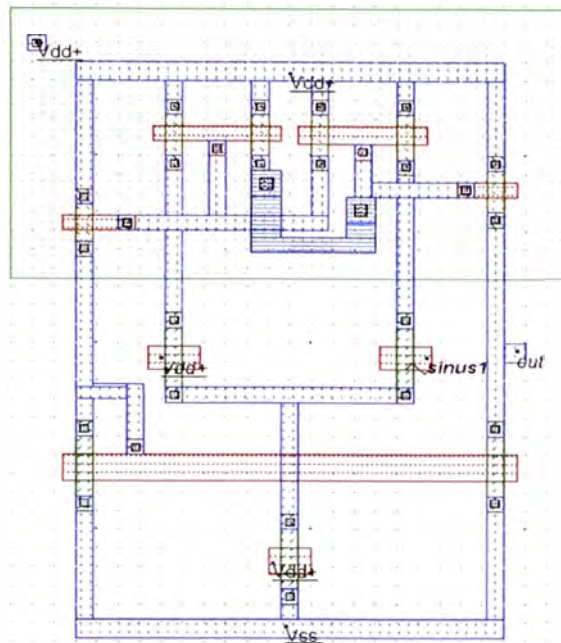
$$\alpha = \frac{(W/L)_5}{(W/L)_3} = \frac{(W/L)_6}{(W/L)_4} = 1$$

Y en las ecuaciones (3.8) y (3.9) nos das  $V_{trig+} = 0$  y  $V_{trig-}=0$ , según los resultados no se consigue el comportamiento con histéresis, existe el riesgo de tener niveles inestables

en la salida, el comportamiento en estas condiciones se muestran en la figura 4.2 el layout y en la figura 4.3 la simulación sin ruido y en la figura 4.4 señal de entrada con ruido.

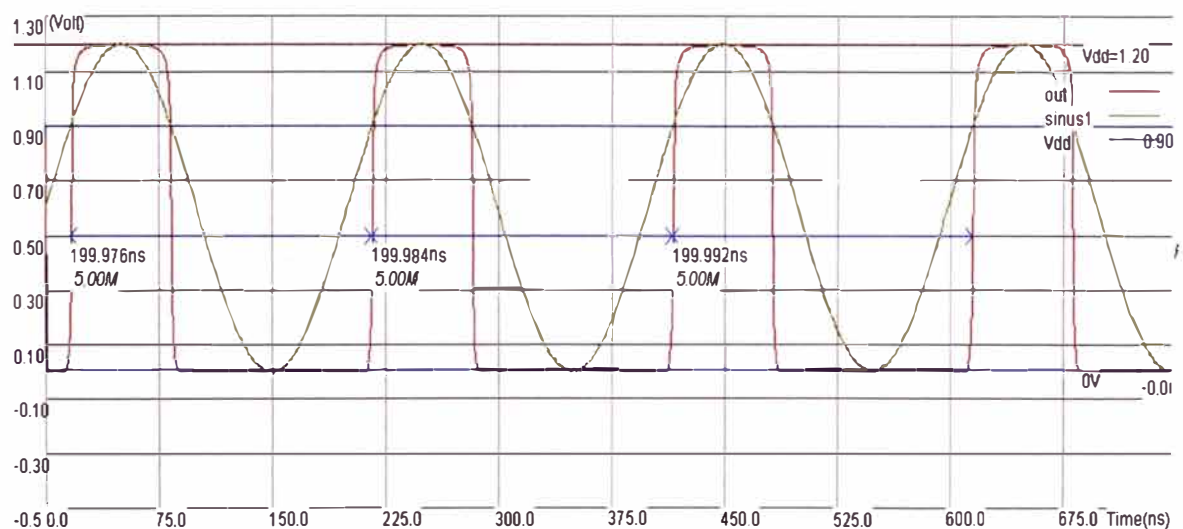
La señal de entrada es una onda senoidal de:

- 5MHz
- Amplitud 0.6v
- Offset fijado en 0.6v
- El voltaje de referencia del comparador es 0.9v



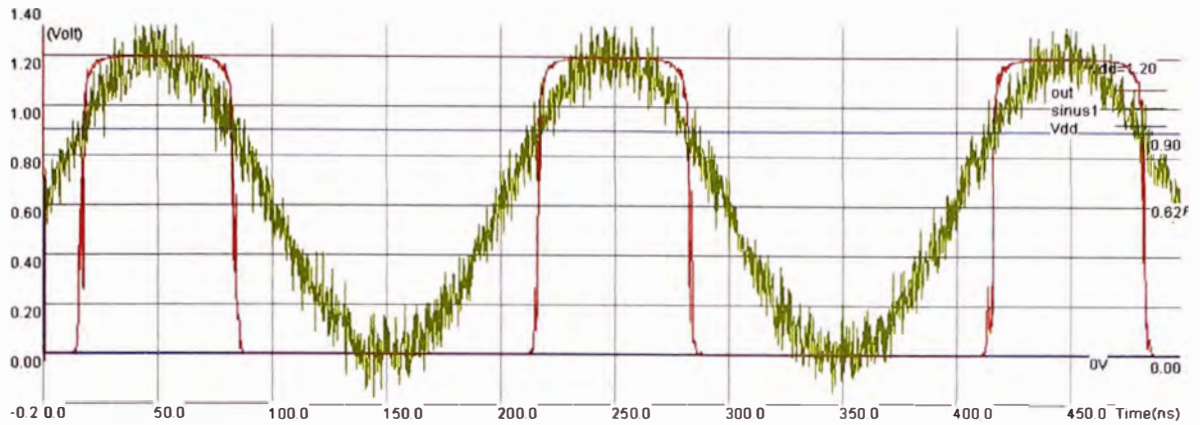
**Figura 4.2 – LAYOUT comparador analógico sin histéresis.**

### RESULTADOS (señal de entrada sin ruido)



**Figura 4.3 – Simulación del comparador analógico sin histéresis.**

Se observa en la figura 4.4 inestabilidad prevista en los flancos de subida y bajada en la salida, debido a la presencia de ruido, por eso es necesario el uso del comparador regenerativo con histéresis.



**Figura 4.4** – Simulación del comparador analógico sin histéresis y con presencia de ruido sumado a la señal de entrada.

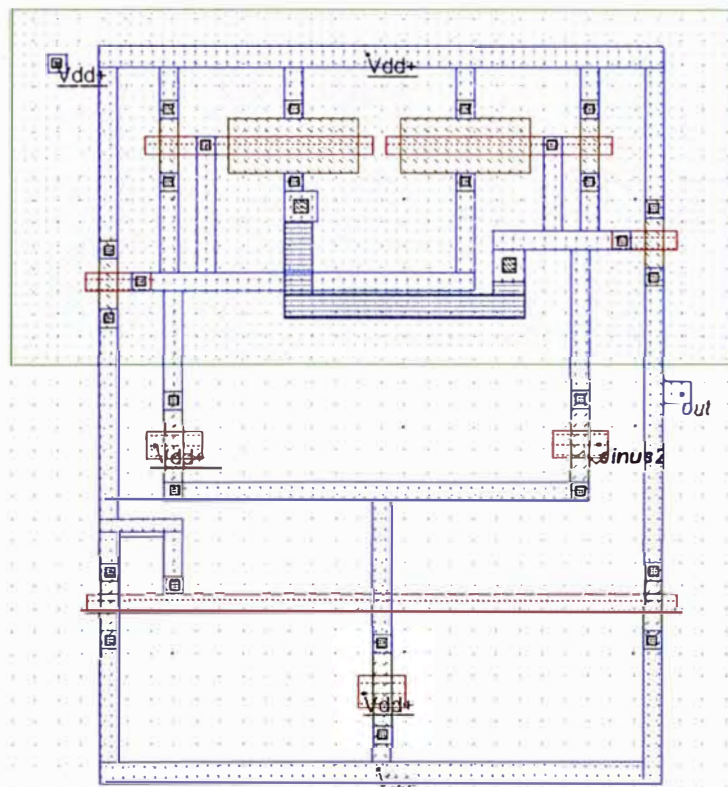
Elementos de realimentación (M5 y M6) de mayor tamaño que los demás para provocar el comportamiento con histéresis. Al dimensionar M5 y M6 de tamaño mayor a los demás CMOS, según la fórmula (3.10) tenemos un valor aproximado de:

$$\alpha = \frac{(W/L)_5}{(W/L)_3} = \frac{(W/L)_6}{(W/L)_4} = \frac{3.5/0.5}{0.5/0.5} \approx 7$$

Lo que da el comportamiento de realimentación positiva, luego para M1 y M2 de igual W y L, en (3.8) y (3.9), tenemos valores de  $V_{trig+}$  y  $V_{trig-}$  diferentes de cero. Al usar la tecnología  $25\mu\text{m}$   $k' = 300\mu\text{A/V}^2$  y  $i_0 = 0.125\text{mA}$ , reemplazando:

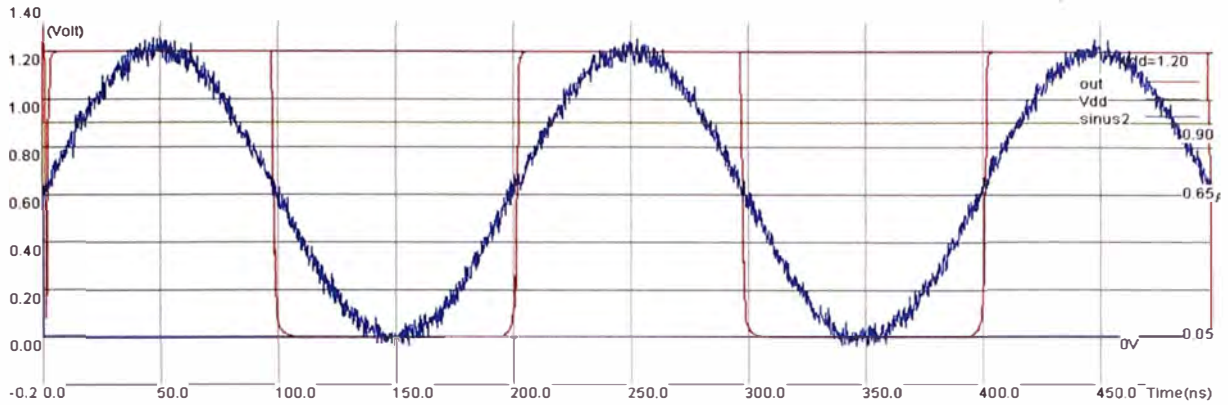
$$V_{trig+} \approx 0.374$$

$$V_{trig-} \approx 0.374$$



**Figura 4.5** – LAYOUT comparador analógico con histéresis I.





**Figura 4.6 – Simulación del comparador analógico con histéresis I.**

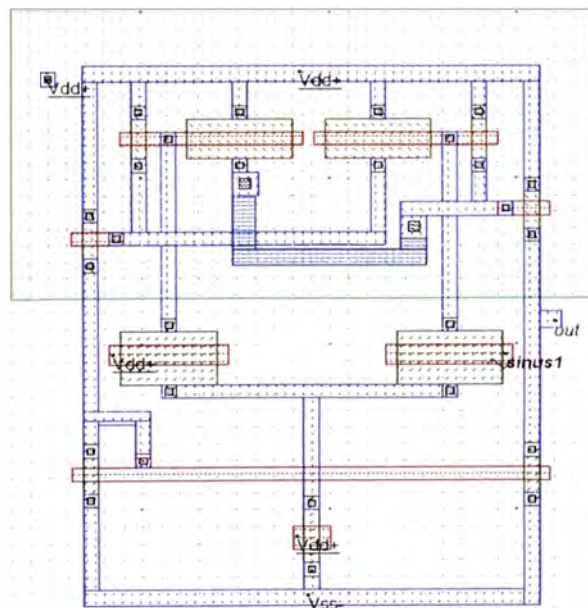
El comportamiento en estas condiciones se muestran en la figura 4.5 el layout y en la figura 4.6 la simulación cuando la señal de entrada tiene ruido.

Según la simulación el intervalo de histéresis se ajusta a lo calculado, sin embargo al tener varios niveles de referencia los intervalos de cada comparador pueden sufrir un cruce, por lo tanto se requiere un intervalo de histéresis menor a  $0.26/2=0.13$ .

Para reducir el intervalo de histéresis tenemos examinando las formulas (3.8), (3.9), (3.1) y (3.11) dos alternativas: Reducir el voltaje bias con lo cual se perdería ancho de banda, aunque se ganaría en menor consumo de potencia o redimensionar en el layout a M1 y por consiguiente a M2 (simétrico de M1), al escoger la segunda alternativa mantenemos el equilibrio entre ancho de banda y potencia consumida.

## **2. Elementos de realimentación (m5 y m6) y elementos de entrada (M1 y M2) de mayor tamaño.**

Los elementos M5 y M6 conservan las dimensiones anteriores, es decir tenemos un valor de  $\alpha=7$ , pero en este caso hacemos los CMOS M1 y M2 de mayor tamaño con el propósito de afinar el intervalo de histéresis como muestra la figura.



**Figura 4.7 – LAYOUT comparador analógico con histéresis II.**

Para este diseño la proporción entre W y L para M1 y M2 es 2, entonces en las formulas (3.8), (3.9) y (3.10):

$$(W/L)_1 = (3.25/0.75)_1 = 4.33$$

$$(W/L)_2 = (3.25/0.75)_2 = 4.33$$

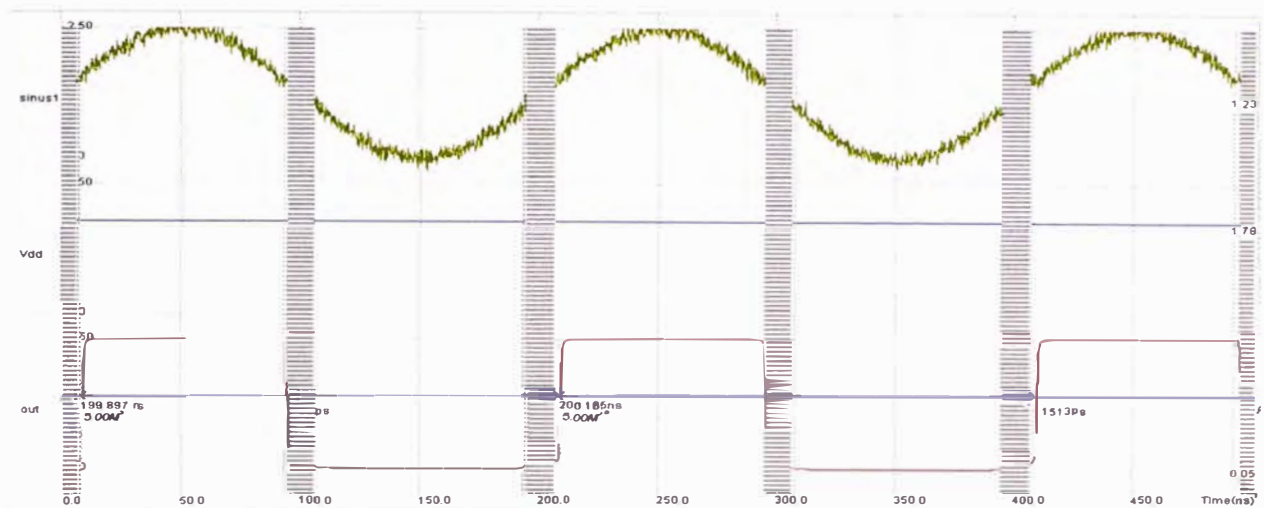
Al usar la tecnología 25 $\mu$ m  $k' = 300\mu\text{A}/\text{V}^2$  y  $i_0 = 0.125\text{mA}$ , se obtiene:

$$V_{\text{trig}+} \approx 0.1$$

$$V_{\text{trig}-} \approx -0.1$$

Así el  $V_{\text{trig}+}$  y  $V_{\text{trig}-}$  es menor que en el caso anterior, El comportamiento en estas condiciones se muestran en la figura 4.7 el layout y en la figura 4.8 la simulación cuando la señal de entrada tiene ruido.

Con esta configuración tenemos un intervalo más conveniente para ser usado en varios niveles de referencia, la potencia que consume el comparador es 0.189mW, para una tecnología de 25 $\mu$ m y  $V_{\text{dd}}=2.5\text{V}$ .



**Figura 4.8**– Simulación del comparador analógico con histéresis II.

El software permite cambiar de tecnología y observar el retardo y potencia, que se consume en cada una de ellas, la tabla 4.1 resume las comprobaciones realizadas

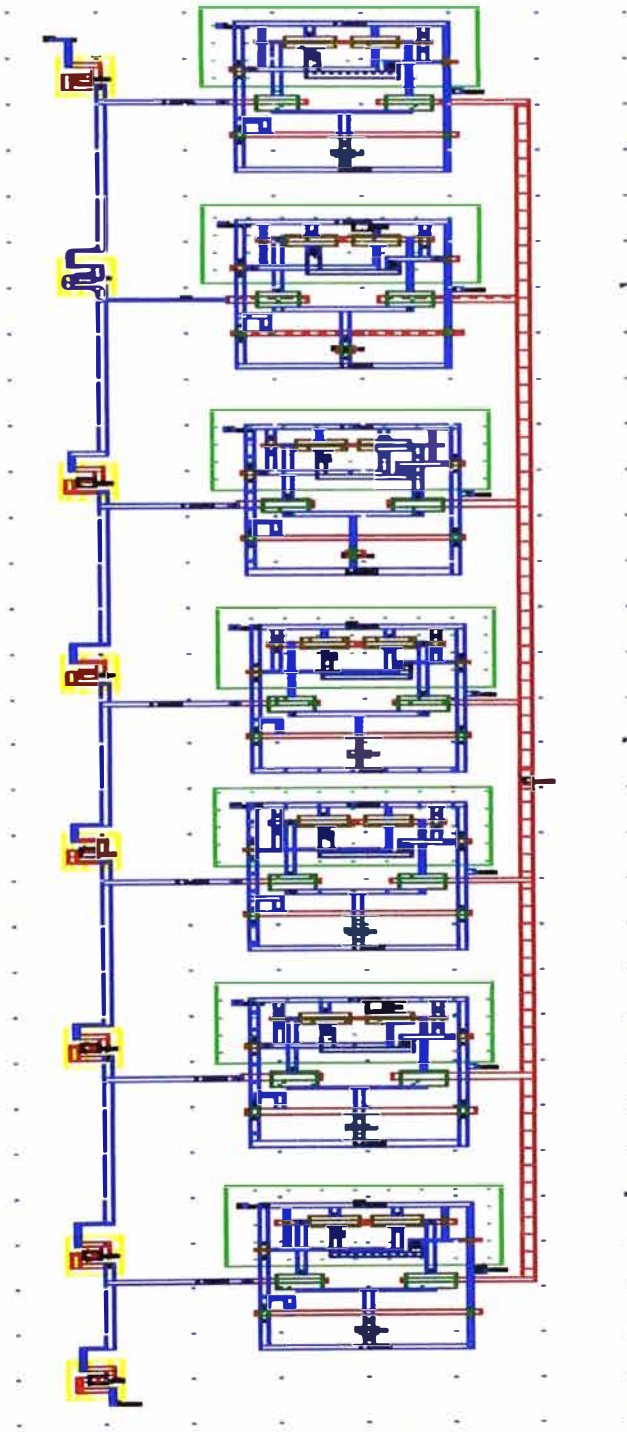
**Tabla 4.1** – Resultados del comparador analógico para diferentes tecnologías.

CMOS 25 $\mu$ m, $V_{\text{dd}} = 2.5\text{v}$	
Tiempo de retardo	0.6nseg
potencia	0.243mW
CMOS 90nm, $V_{\text{dd}} = 1.2\text{v}$	
Tiempo de retardo	0.4nseg
potencia	34.487 $\mu$ W
CMOS 65nm, $V_{\text{dd}}=0.7\text{v}$	
Tiempo de retardo	0.35nseg
potencia	8.654 $\mu$ W



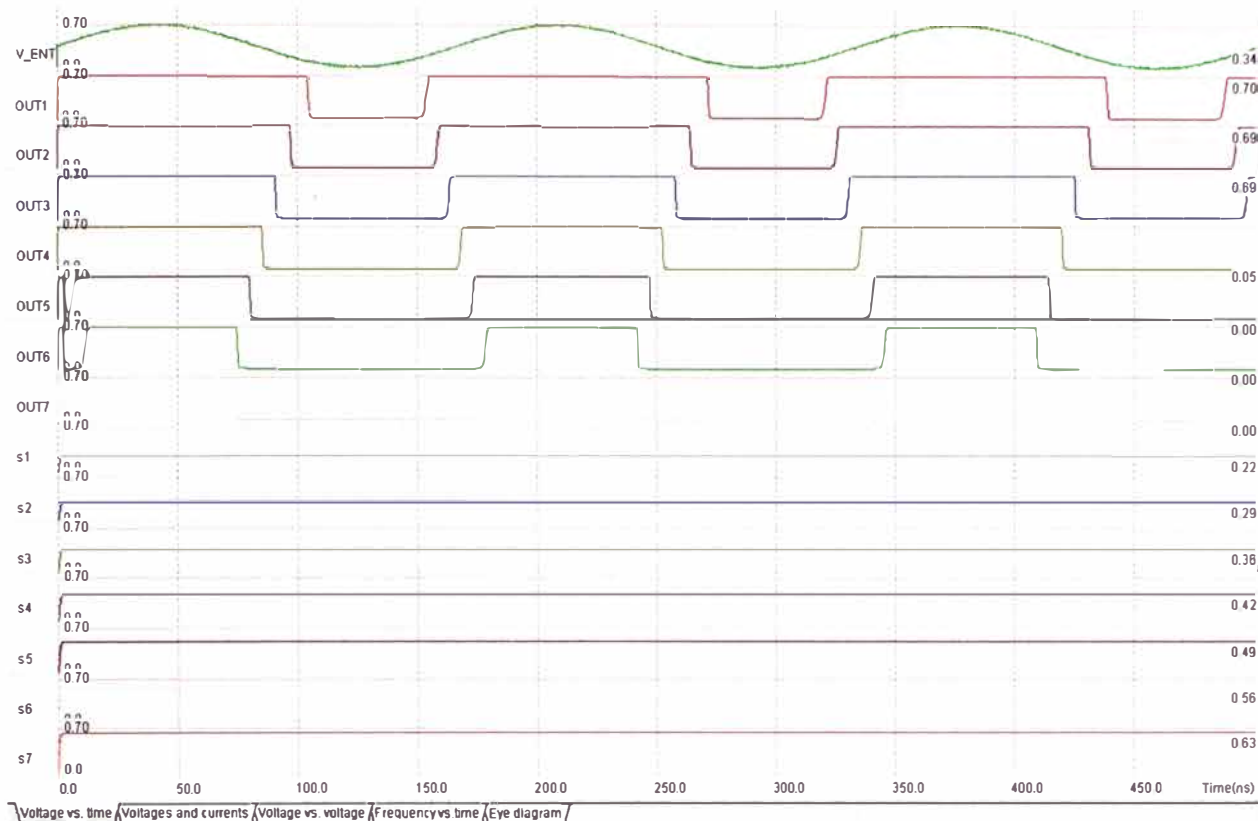
Una mejora en el consumo de potencia se consigue si se cambia a una mejor tecnología como la de 90nm o 65nm.

La parte análoga del convertidor también necesita un divisor de voltaje para generar los niveles de referencia. Un divisor capacitivo o resistivo puede ser usado, con la ventaja de este último de minimizar la potencia de drenaje desde el suministro, pero también se puede usar una resistencia de polysilicon con la ventaja de no incrementar la potencia aunque requiere mayor área. El esquema para 7 niveles de referencia se muestra en la figura 4.9 y la simulación para una señal de tipo senoidal en la figura 4.10:



**Figura 4.9** – Esquema del comparador para ocho niveles.

La simulación de la parte análoga para 7 niveles es:



**Figura 4.10** – Simulación del comparador analógico de 07 niveles.

Al igual que en el caso de un nivel de referencia inicial, también se comprueba para diferentes tecnologías y se resume los resultado en la tabla 4.2

**Tabla 4.2** - Resultados del comparador analógico con 7 niveles de referencia

CMOS 25 $\mu$ m, Vdd = 2.5v	
Tiempo de retardo	0.6nseg
potencia	1.049mW
Frecuencia max	100MHz
CMOS 90nm, Vdd = 1.2v	
Tiempo de retardo	0.5nseg
potencia	0.177mW
Frecuencia max	180MHz
CMOS 65nm, Vdd = 0.7v	
Tiempo de retardo	0.4nseg
potencia	24.827 $\mu$ W
Frecuencia max	70MHz

Una mejora en el consumo de potencia se consigue si se cambia a una mejor tecnología como la de 90 nm o 65nm, sin embargo hay una disminución de la frecuencia máxima, debido al umbral del transistor.

### 4.3. Simulación de la sección digital

#### 4.3.1 Consideraciones particulares para el test de circuitos asíncronos

Para la simulación de circuitos digitales asíncronos se toma en cuenta los siguientes puntos:

- Se pierda la controlabilidad total del proceso debido a la no presencia de un GLOBAL CLOCK.
- Existe un desarrollo pobre de técnicas de test para diseños de circuitos digitales asíncronos, como alternativa se simula cada modulo de manera independiente y luego todo en conjunto, verificando el cumplimiento de las condiciones y especificaciones en cada paso.
- La detección de “hazards” y “races” es un aspecto crítico, debido a que llevan consigo un significado trascendental en el funcionamiento global.
- En el caso de los handshake, se considera que son autotest, puesto que la no presencia de alguna de las señales involucradas (REQ o ACK) causara un estado de espera infinito que es fácilmente observable.
- El diseño es modular porque facilita la definición con precisión da cada modulo y la interconexión posterior entre ellos es sencilla debido al handshake, siguiendo la tendencia actual de “plug & play”. La condición principal de los módulos es que no debe presentar ‘hazards’, es decir, las señales resultantes y de interconexión deben tener un comportamiento monótono.
- En algunos casos cuando el diseño digital asíncrono no es “independent delay” es necesario estandarizar un elemento de retardo independiente del compilador para facilitar la portabilidad, usualmente se usa el elemento C (ocupa mínima área) o el buffer LCELL en el caso de ALTERA.
- Se toma como entorno de desarrollo los FPGAs y el compilador QUARTUS II de altera, tanto para la biblioteca de módulos asíncronos, porque permite crearlos de manera individual usando vhdl, verilog o esquemáticos para luego añadirlos al proyecto final. El compilador junto con los lenguajes (VHDL o Verilog) son herramientas maduras, se dispone de los datos técnicos de los FPGAs que soportan y además es la que mejor se adapta a los conceptos de concurrencia y señalización por eventos, esto aumenta la portabilidad y la posibilidad de tener subsistemas síncronos y asíncronos en un modulo combinado. Familias más recientes de FPGAs como STRATIX IV ofrecen mayor desempeño en potencia y velocidad.
- Dentro de los puntos en contra de la metodología asíncrona se puede mencionar que para su implementación se requiere mayor área, número de líneas de comunicación y pines que su contraparte síncrona.

La parte digital del conversor contiene tres elementos:

## 1. BLOQUE TRIGGER

Este bloque usa elementos C, como se puede apreciar en la figura 4.11, para reunir o agrupar las condiciones que producen un requerimiento de un cruce de nivel, una vez que el requerimiento es atendido el bloque se vuelve a habilitar a través de una señal ENABLE.

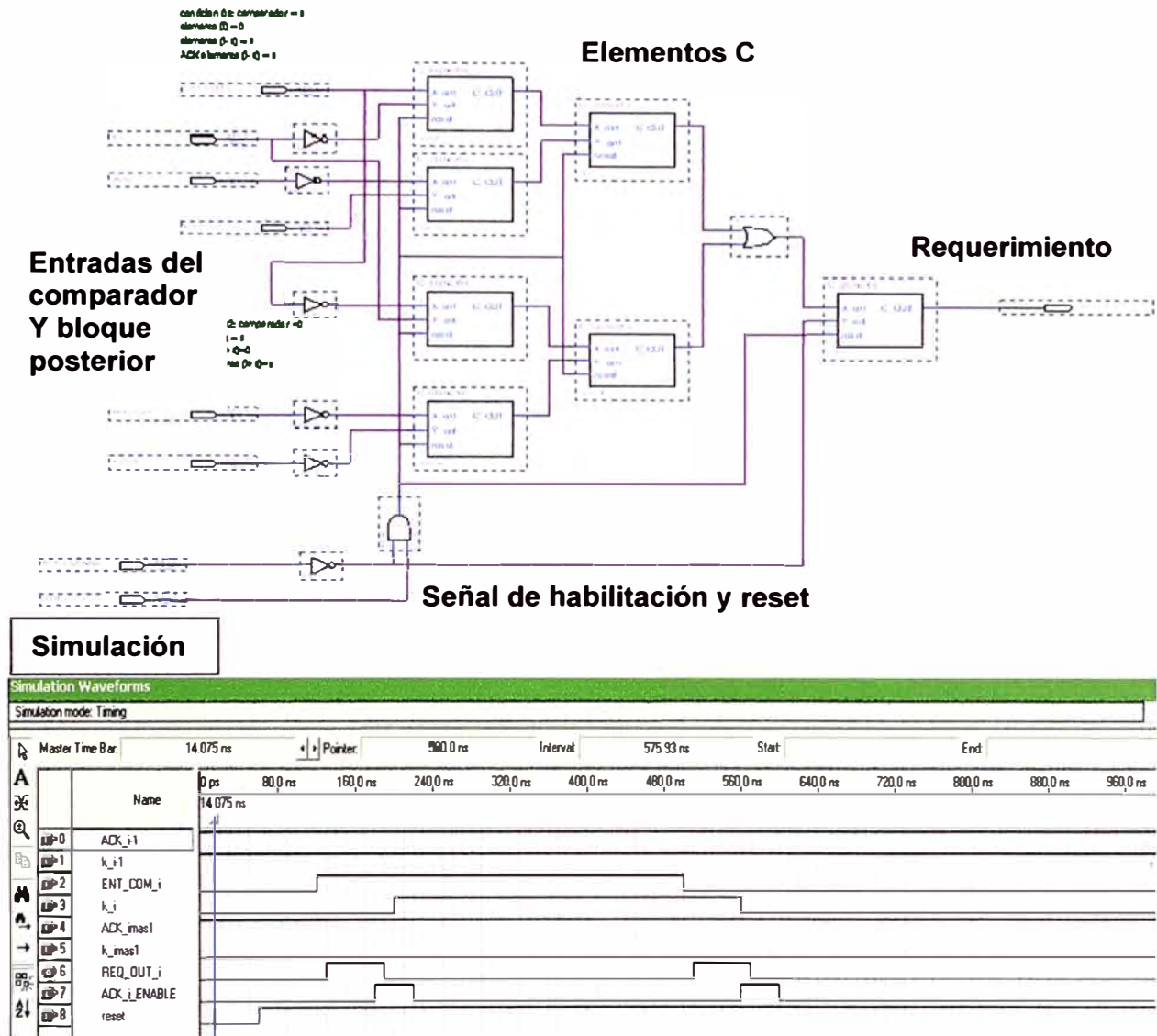
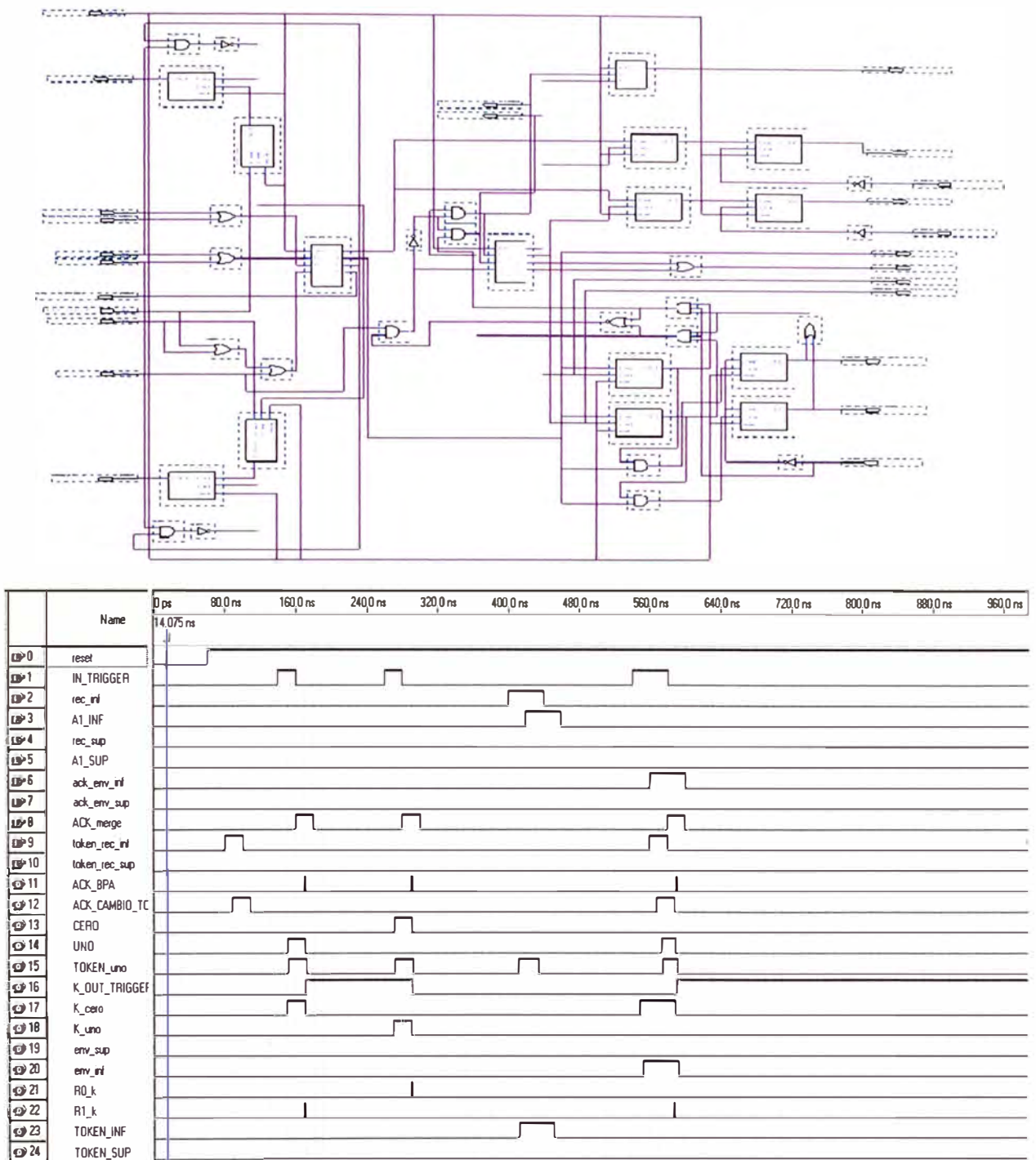


Figura 4.11 – Esquema y simulación del bloque TRIGGER.

## 2. BLOQUE DE PROCESAMIENTO ASINCRONO

En la figura 4.12 se observa el esquema del elemento de procesamiento asíncrono, que toma en cuenta el desempeño de los dos procesos mencionado en el capítulo 3, está formado por elementos C que reúnen la condiciones necesarias para los diferentes procesos de este bloque y dos elementos SEL-K, que sirven como registros para almacenar los valores de 't' y 'k', la implementación de este bloque permite también el sostenimiento del requerimiento hasta que no se produzca la salida correspondiente, así mismo en cada canal de comunicación esta implementada una etapa de handshake tipo micropipeline para la petición y transmisión del token entre bloques y para la salida dual

rail de los datos también hay un proceso de handshake con el bloque Merge. Cada vez que se inicializa el sistema o después de una señal reset los bloques están al a espera de cualquier requerimiento.



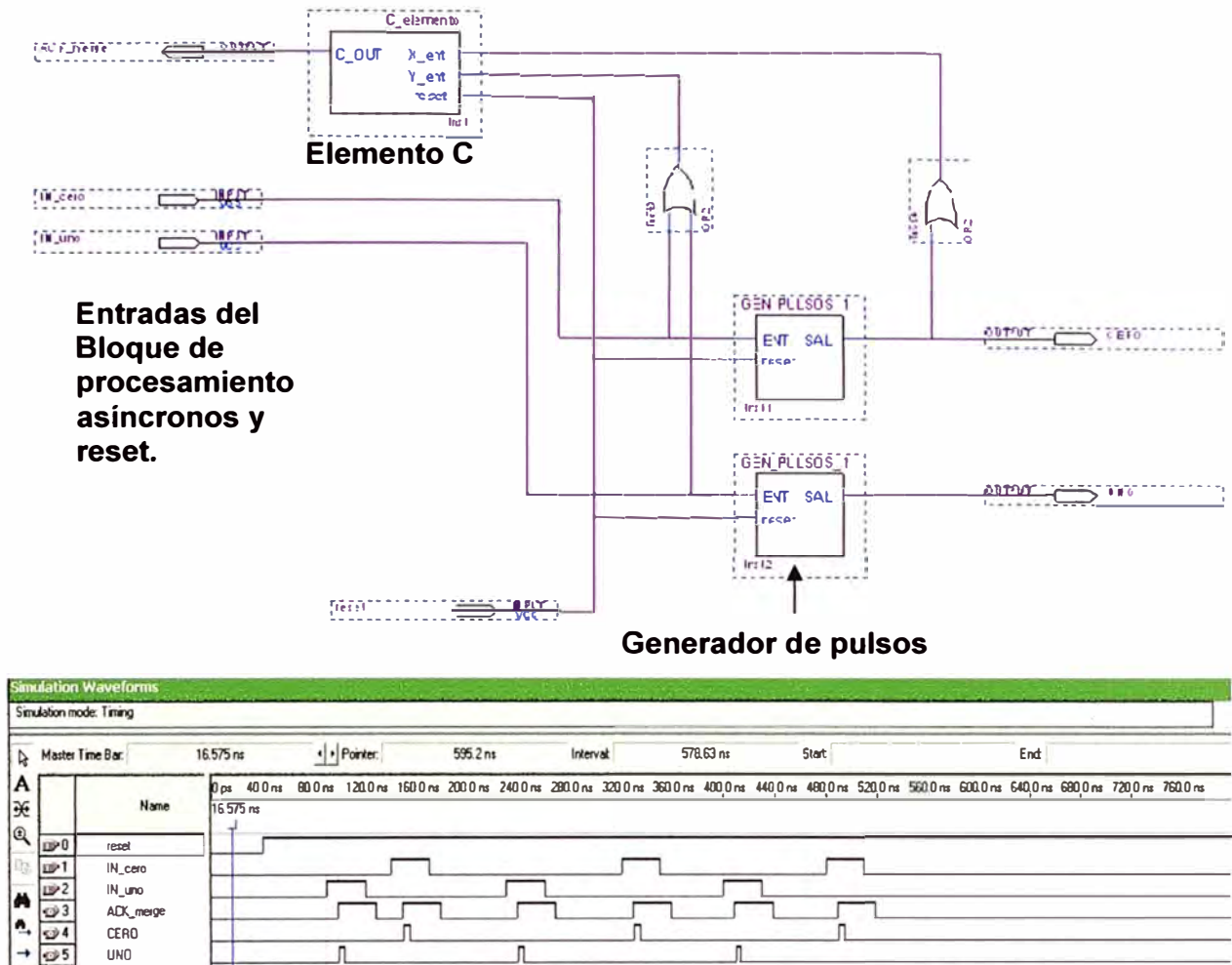
**Figura 4.12–** Esquema y simulación del bloque procesamiento asíncrono.

### 3. BLOQUE MERGE

Es el último bloque de la parte digital, está a su vez sub dividido en dos módulos mas, el primero tiene como función realizar el proceso de handshake con el bloque de procesamiento asíncrono así como producir los pulsos de salida, existe uno por cada nivel y es el que esta dibujado en la figura 4.13.

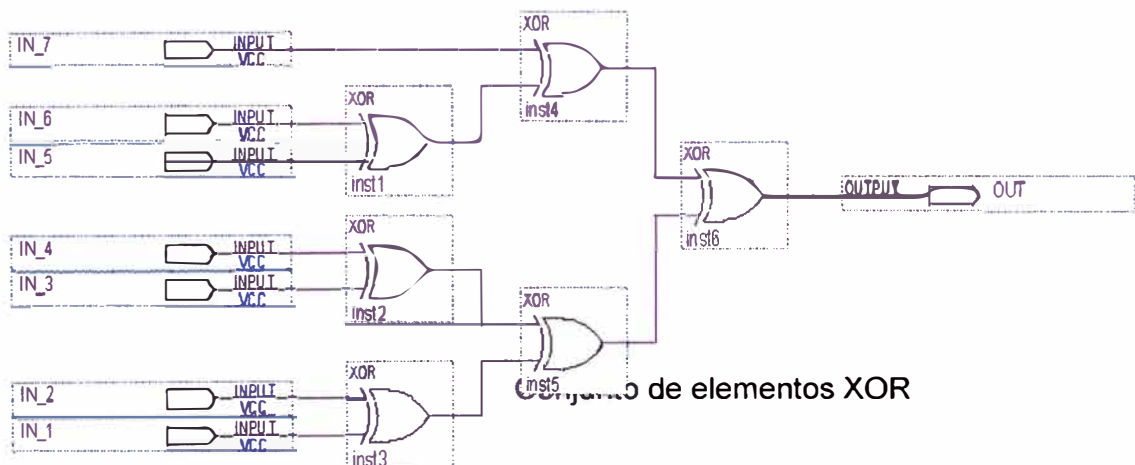


Los pulsos de salida son producidos en base a un generador de pulsos que está elaborado en base a un elemento de retardo estándar que usa inversores y es parametrizable, esto ayuda a la portabilidad y reducción de área de uso en el FPGA.



**Figura 4.13 – Esquema y simulación del MERGE I**

La segunda parte son dos bloques de elementos XOR que reúne los pulsos de salida '1's y '0's por separado para su presentación final en el modo dual rail y es el que esta dibujado en la figura 4.14.



**Figura 4.14 – Esquema general del MERGE II**

Hasta este punto se ha desarrollado la parte análoga y la parte digital obteniendo la modulación delta asíncrona, el escenario siguiente es la producción del código binario.

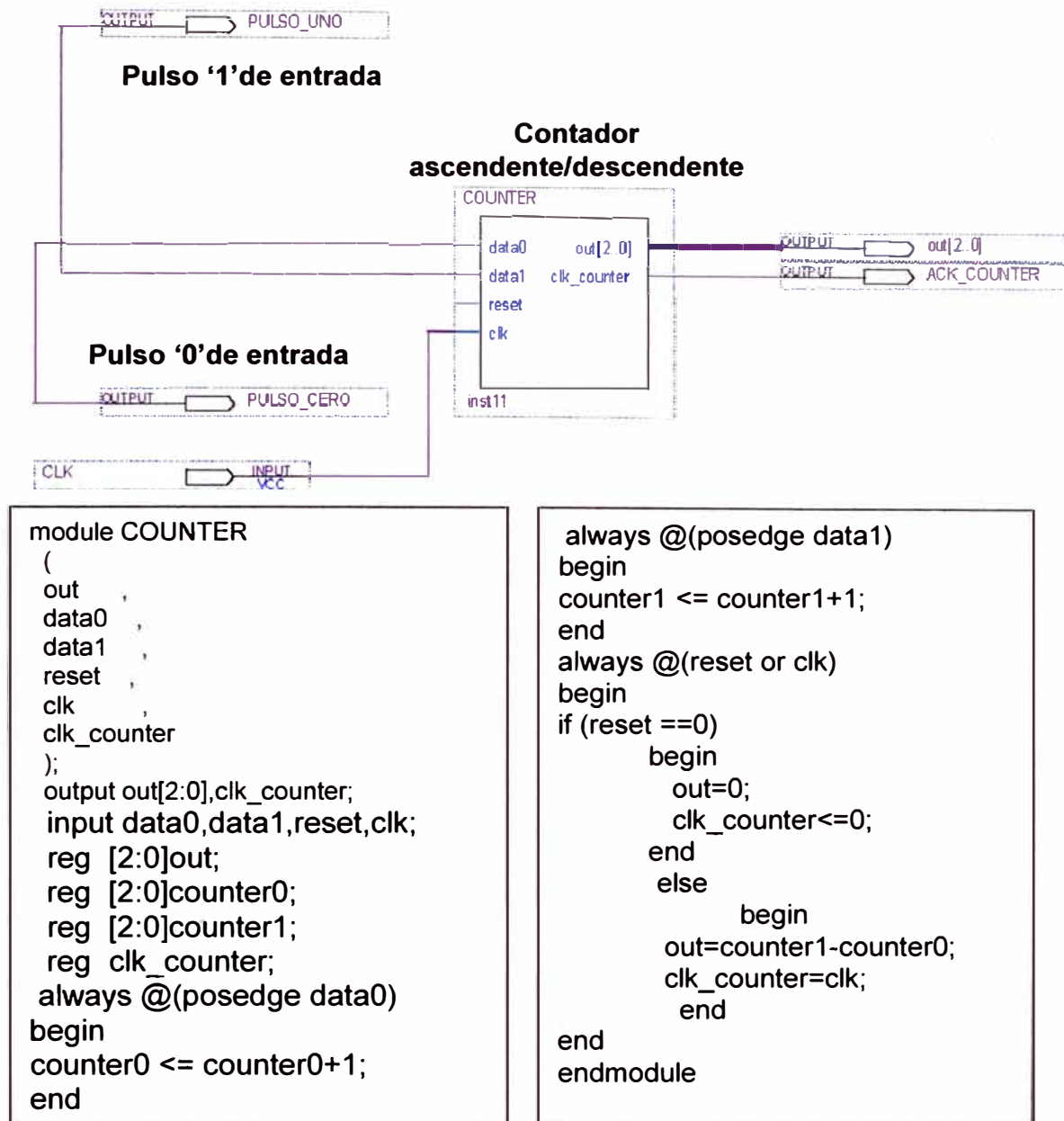
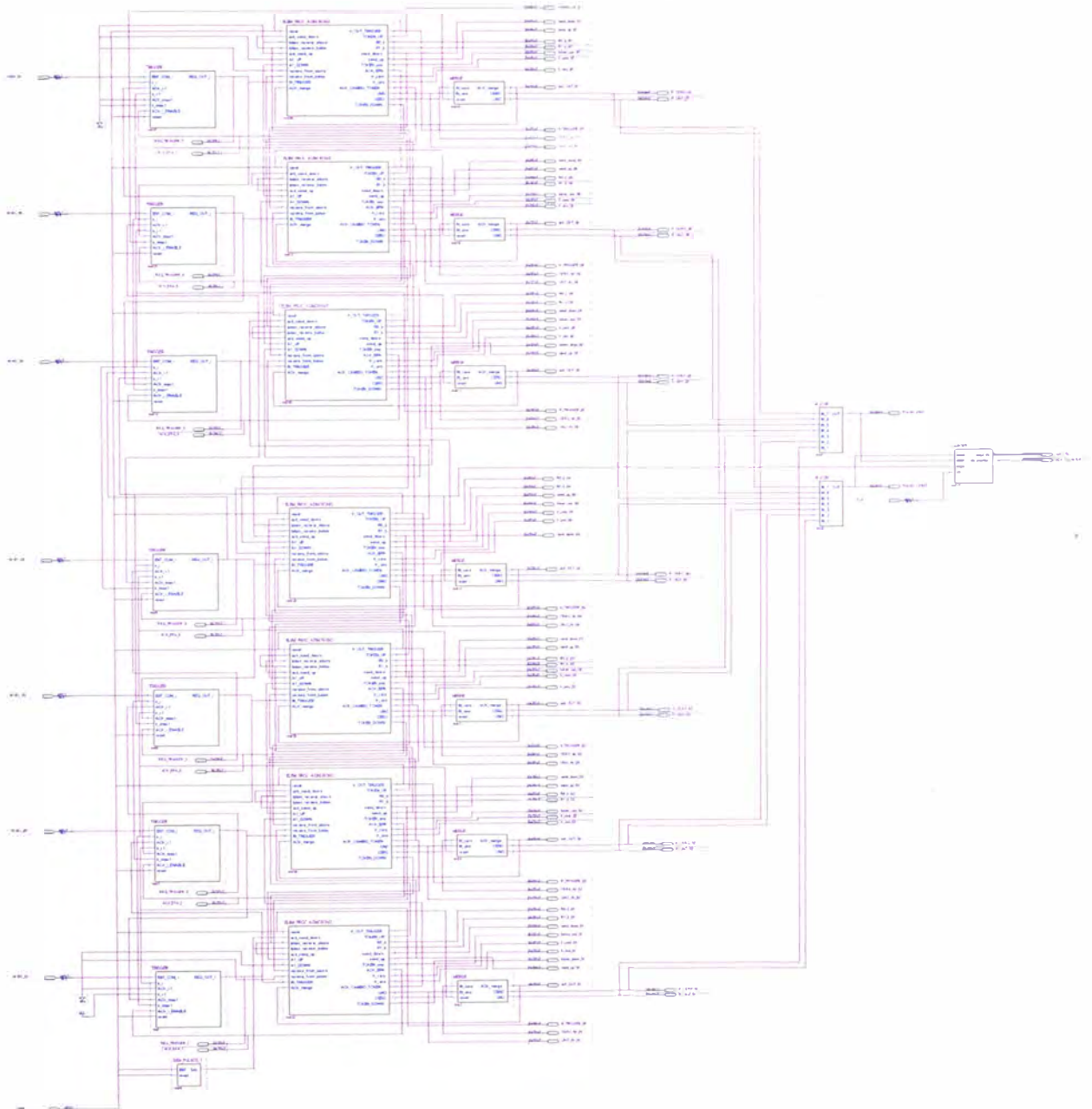


Figura 4.15 – Simulación y código verilog del Contador.

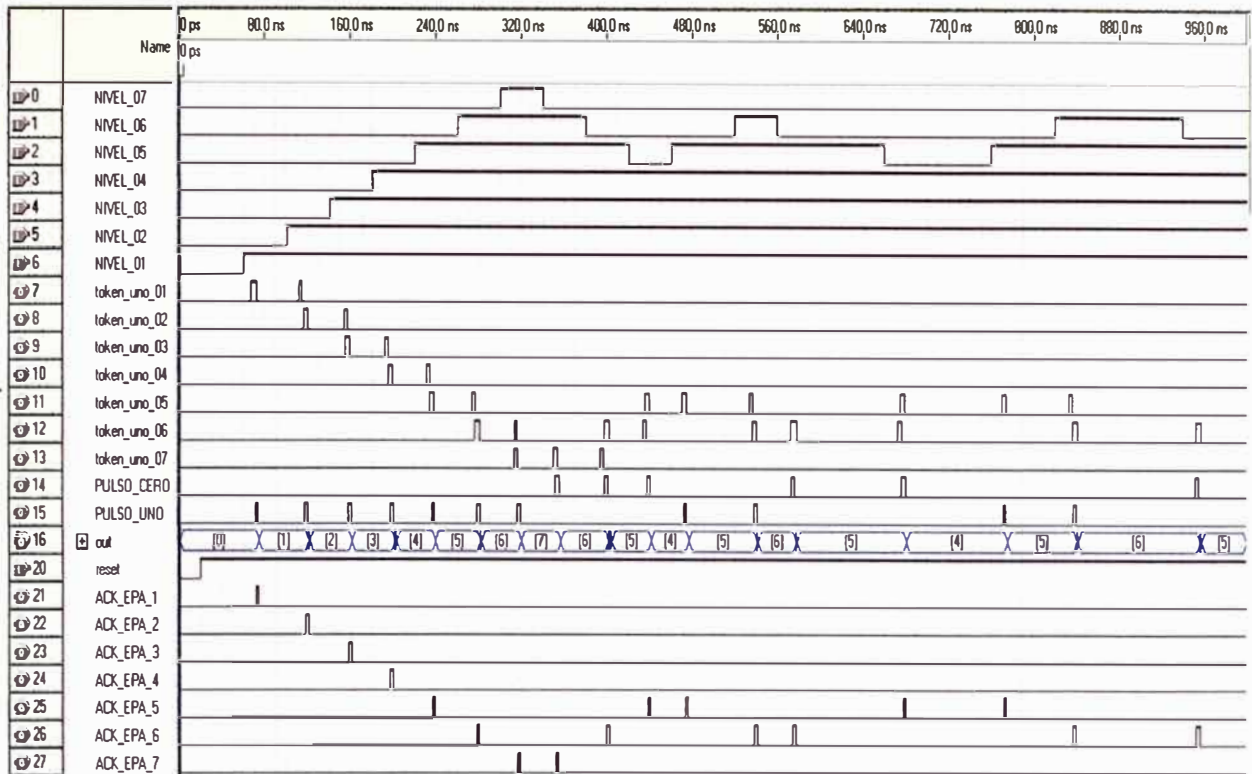


Los pulsos dual rail, son interpretados con un contador ascendente y descendente (en este caso el contador es de tres bits, correspondiente al número de niveles de referencia) que suma cuando el pulso es '1' y resta cuando el pulso es '0', también con el fin de implementar una posible reconstrucción de la señal en un proceso siguiente es factible implementar adicionalmente una entrada y salida de clock opcional, que aporta también para un enlace con un elemento de procesamiento síncrono, como muestra la figura 4.15.

La parte digital completa con las interconexiones entre los módulos, para un caso de análisis de 07 niveles y 03 bits, es la que esta esquematizada en la figura 4.16, este esquema incluye un pequeño generador de pulsos de una vida, como entrada de token del primer bloque inferior, esto con el fin de que después de cada inicio o reset el token esté presente en el circuito. La simulación se muestra en la figura 4.17.



**Figura 4.16** – Esquema del conversor AD asíncrono (parte digital).



**Figura 4.17** – Simulación del conversor AD asíncrono (parte digital).

### 4.3.2 Metodología de las mediciones

El objetivo de esta etapa es comprobar el correcto funcionamiento de los módulos, el concepto de verificación funcional implica demostrar que el circuito final cumple de manera correcta la función para la cual fue creado.

Luego se hacen medidas del throughput, latencia, número de recursos usados y potencia consumida, para establecer las características técnicas y parámetros del módulo.

La medición del throughput se realiza calculando la latencia del circuito la cual es la resta entre el REQ inicial y el ACK final, así la inversa de la latencia será el throughput, esta diferencia depende de la tecnología final de implementación, sin embargo, los valores obtenidos de la simulación son bastantes cercanos al desempeño real.

El promedio del consumo de potencia en un sistema digital puede ser dividido en 4 componentes: potencia dinámica, potencia de corto circuito, potencia de fugas y potencia estática. En la tecnología CMOS los componentes de corto circuito y estática son de valores despreciables. Igualmente, la potencia de fugas en las actuales tecnologías comerciales es muy reducida, por lo cual se considera que la potencia dominante es la potencia dinámica. El consumo de potencia en los circuitos CMOS es debido en gran parte a la actividad de conmutación, en cualquier nodo dicha potencia dinámica puede ser calculada por la ecuación:

$$P = f(CV^2)/2 \quad (4.1)$$

Donde:

$C$  es la capacidad del nodo

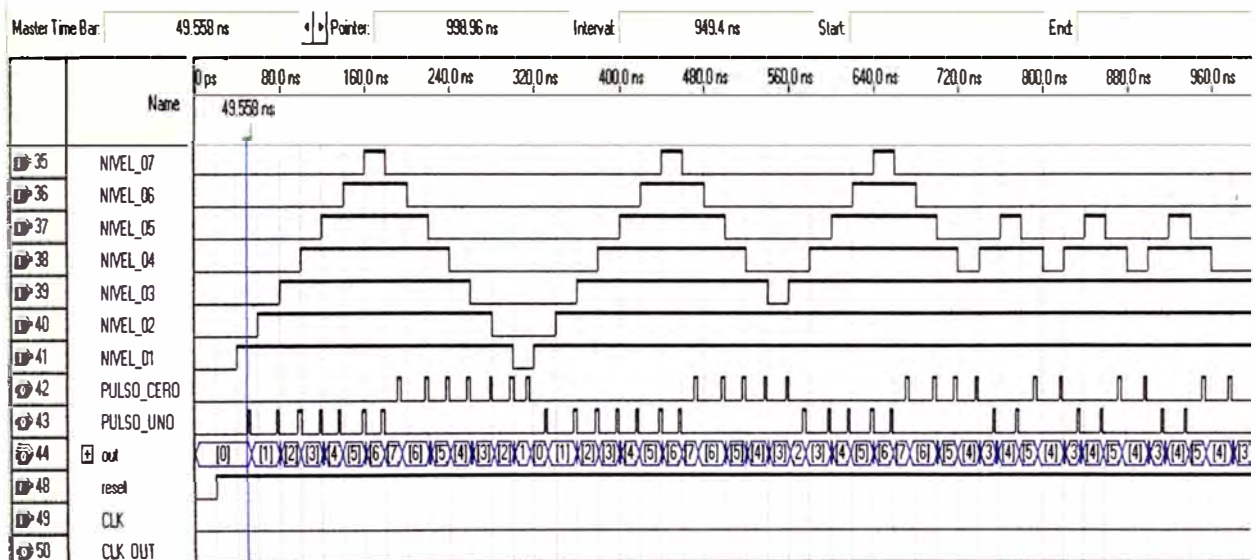
$V$  es el voltaje de alimentación y

$f$  es la frecuencia de conmutación en dicho nodo.

A partir de esta ecuación se pueden plantear alternativas para disminuir la potencia dinámica de los circuitos síncronos, una reducción de la fuente de alimentación tiene un efecto cuadrático, sin embargo, también reduce la velocidad de operación y reduce el margen de ruido. Otra alternativa es la reducción global de la capacidad de conmutación dentro del circuito con unas apropiadas dimensiones de los CMOS, se puede emplear también técnicas de regulación de la señal de reloj para bloques inactivos.

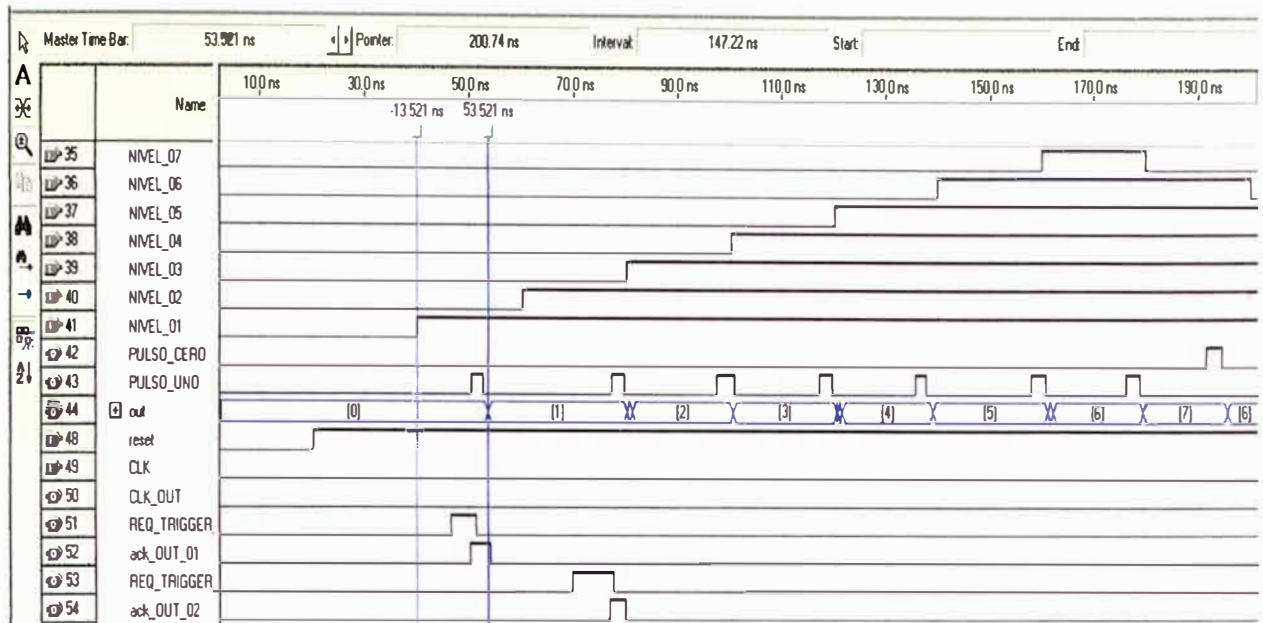
En los circuitos asíncronos no existe una señal de reloj global, por lo cual disminuye la potencia dinámica. La característica de los bloques asíncronos es el rendimiento tipo "average-case", debido a la adaptación a diferentes requerimientos de cambio de la señal de entrada, lo cual permite un bajo consumo de potencia. Entonces para la medición de la potencia se toma el promedio a diferentes entradas.

Para realizar las pruebas se usa una señal onda triangular o senoide con  $2n$  cruces por periodo y se escogió el FPGA de la familia STRATIX III (65nm), modelo EP3SL50F484C2 el diseño ocupa 468 ALUTs de un total de 38 000, representando solo el 1% y 07 registros lógicos de un total de 38 000. La figura 4.18 muestra la comprobación funcional del circuito.



**Figura 4.18** – Simulación parte funcional digital.

De la figura anterior, el software QUARTUS II 9.1 nos muestra la formación de los pulsos y el código binario del contador ascendente y descendente según el sentido del código termométrico de entrada. Para las mediciones se toman dos casos, cuando el bloque posee el token y cuando no lo posee y se consideran los retardos entre las señales REQ y ACK, tomando como referencia la entrada nivel\_01 y nivel\_02, según la metodología de análisis de resultados descrito anteriormente y la figura 4.19.



**Figura 4.19 – Cálculo de los retardos.**

**CASO 1.-** el bloque posee el TOKEN y lo único que tiene que hacer es producir la salida correspondiente.

REFERENCIA: entrada del comparador:

$$t_{\text{nivel1}} - t_{\text{REQ\_TRIGGER\_1}} = 6.602 \text{ nseg}$$

$$t_{\text{nivel1}} - t_{\text{ACK\_OUT\_01}} = 10.112 \text{ nseg}$$

$$t_{\text{REQ\_TRIGGER\_1}} - t_{\text{ACK\_OUT\_01}} = 3.573 \text{ nseg}$$

$$t_{\text{REQ\_TRIGGER\_1}} - t_{\text{COD\_BIN}} = 13.521 \text{ nseg}$$

**CASO 2.-** el bloque no posee el TOKEN, entonces lo pide al bloque vecino y luego produce la salida correspondiente.

REFERENCIA: entrada del comparador:

$$t_{\text{nivel2}} - t_{\text{REQ\_TRIGGER\_2}} = 9.700 \text{ nseg}$$

$$t_{\text{nivel2}} - t_{\text{ACK\_OUT\_02}} = 16.627 \text{ nseg}$$

$$t_{\text{REQ\_TRIGGER\_2}} - t_{\text{ACK\_OUT\_02}} = 7.259 \text{ nseg}$$

$$t_{\text{nivel2}} - t_{\text{COD\_BIN}} = 21.689 \text{ nseg}$$

Consideramos la mayor latencia del sistema:

El factor de seguridad debido a variaciones del proceso es una recarga del 30%

$$t_{\text{LATENCIA\_PROC\_ASINCRONO}} = 16.627(130/100) = 21.615 \text{ nseg}$$

$$t_{\text{LATENCIA\_TOTAL}} = 21.689(130/100) = 28.1957 \text{ nseg}$$

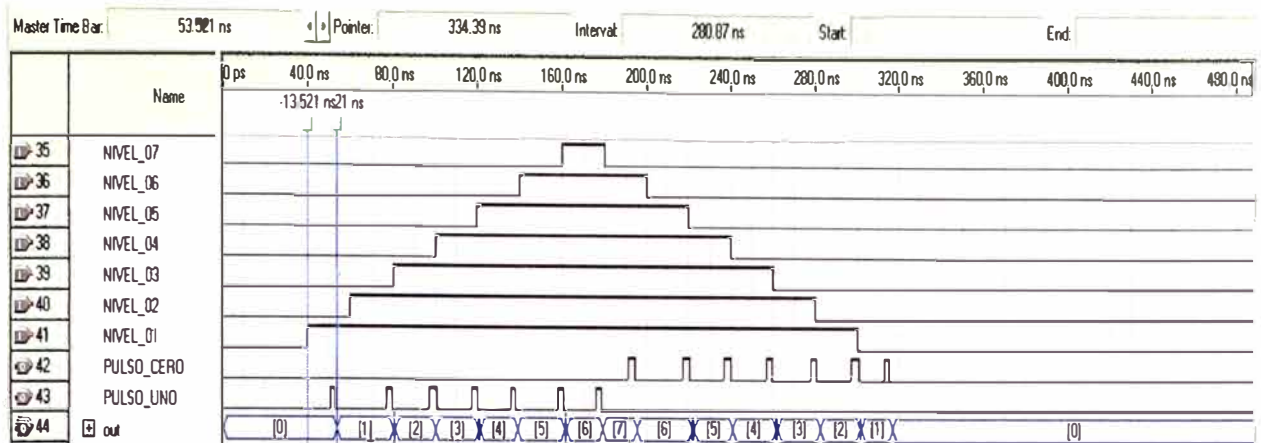
Como el throughput es la inversa de la latencia entonces:

$$\text{Throughput (total)} = 1/t_{\text{LATENCIA}} = 35.4664 \times 10^6 \text{ outputs / seg}$$

$$\text{Throughput (asíncrono)} = 1/t_{\text{LATENCIA}} = 46.2639 \times 10^6 \text{ outputs / seg}$$

Para realizar las pruebas de consumo de potencia se usa una señal onda triangular o senoide con 2n cruces por periodo. Como se muestra en la figura 4.20.





**Figura 4.20** – Código termométrico de entrada para el cálculo de potencia

La potencia dinámica es 1.68mW, para un código termométrico de periodo=280nseg

El estilo de diseño asíncrono empleado da una flexibilidad a la velocidad de procesamiento digital, si hay cambios en el comparador en un tiempo menor al de latencia, la parte digital asíncrona retiene la requerimiento hasta que se termine de procesar el evento anterior, una vez finalizado procede recién a atender el nuevo requerimiento, esta forma de operar da la garantía de que no se va a perder ningún cruce y que el límite de ancho de banda de la señal de entrada depende mayormente del comparador análogo.

Finalmente podemos resumir las características del diseño en la tabla 4.3, para una señal de prueba triangular sinusoidal con  $2n$  cruces, donde  $n=7$  y para aumentos en la resolución las frecuencias máximas de entrada se muestra en la tabla 4.4.

**Tabla 4.3** – Resultados de la simulación total para STRATIX III

EP3SE50F484C2 65nm

	Parte Análoga	Parte Digital	Total
Tecnología	CMOS 65nm	STRATIX III EP3SE50F484C2	
Aplicación	Señal análoga: Temperatura ambiental		
Niveles de referencia	07		
Resolución		3 - bits	3 - bits
Voltaje Vdd	0.7V	1.1V	
Rango dinámico $\Delta V$	0.7v		0.7V
Paso de la señal (q)	0.07V		
Potencia total ( $2n$ cruces, $n=7$ )	0.174mW	1.68mW	1.184mW
Potencia/muestra	8.654 $\mu$ W	0.14mW	0.1486mW
Retardo $\delta_{max}$	0.4nseg	28.1985nseg	28.5985nseg
Frecuencia máxima (condición de rastreo y teorema de Bersntein)	70.00MHz	5.00MHz	5.00MHz

**Tabla 4.4– Variación de la frecuencia y la resolución:**

Resolución (bits)	Paso 'q' para $\Delta V=0.7$	Frecuencia máxima de entrada
4	0.046mV	2.33MHz
5	0.022mV	1.128MHz
6	0.011mV	555.194KHz
7	0.55 $\mu$ V	275.411KHz

Los resultados muestran bajo consumo de potencia en la parte digital y analógica por muestra, también conforme se aumenta los niveles de referencia y por consiguiente la resolución la frecuencia disminuye, puesto que el paso 'q' se hace más pequeño y aumenta el número de cruces en tiempo determinado.



## CONCLUSIONES Y RECOMENDACIONES

El estudio y análisis de las redes de sensores inalámbricos y la metodología de diseño asíncrono, ha servido para comprender los fundamentos y conceptos básicos de estas tecnologías y ver sus potenciales aplicaciones en diversas áreas, revisar la topología de este tipo de redes tanto a nivel local como global aporta a la identificación de procesos que pueden ser objetos de mejora y optimización, para lo cual es necesario un conocimiento especializado de las diversas áreas de ingeniería involucradas. De la misma forma el estudio de la metodología de diseño digital asíncrono ha mostrado que esta nueva forma emergente de diseño digital tiene ventajas que se pueden aprovechar, contando cada vez más con herramientas especializadas, que toman como base la teoría presentada y que puede significar un cambio significativo en algunos sistemas.

1.- La adaptación de las redes de sensores inalámbricas para monitoreo ambiental, es el método menos invasivo entre las alternativas de monitoreo y significa menos daño potencial para las áreas protegidas.

2.- Las redes de sensores inalámbricas brindan datos más exactos sobre las variables a monitorear, porque los nodos pueden ser ubicados muy cerca del área de interés y las topologías son escalables llegando a soportar cientos de nodos.

3.- Teniendo en cuenta que la tendencia de hoy es tener dispositivos de mayor rendimiento, con menor área, mayor velocidad, pero con ahorro en el consumo de energía y alta durabilidad. Los conceptos digitales asíncronos son cada vez más tomados en cuenta en la industria del semiconductor, porque disminuyen el consumo de potencia y alivian muchos temas relacionados con el global clock. Los bloques asíncronos dentro de un circuito integrado cada vez ocupan más espacio porcentual dentro del chip y también se da la variante de combinar ambos modos de implementación digital.

4.- Las propiedades de los sistemas asíncronos son deseables en diferentes aplicaciones: dispositivos portátiles como computadoras personales o celulares pueden tomar ventaja de las propiedades de la metodología asíncrona.

5.- La toma de datos de manera asíncrona es más cercana a la naturaleza de las señales del mundo real, lo cual permite aumentar la eficiencia del circuito digital.

6.- Aunque la industria creciente de redes de sensores inalámbricos usa casi en su totalidad micro controladores, los FPGAs se presentan como alternativa para aplicaciones

más complejas además que son las que mejor asimilan la metodología de diseño asíncrono gracias a su flexibilidad y soporte, lo cual en conjunto ofrece beneficios para el diseño de redes de sensores inalámbricos. Actualmente esta opción está en etapa de investigación académica.

7.- El convertidor en su etapa analógica es similar al convertidor análogo digital tipo FLASH, esto limita la resolución final a 6-7 bits, debido a las complicaciones para manejar mayor cantidad de comparadores por la potencia que consumen, también dificulta la regulación de voltaje CMOS y disminuye la frecuencia de la señal de entrada.

8.- Con el fin de aumentar el ancho de banda y resolución, se podría cambiar la parte análoga y reemplazarla por una adaptación similar a la modulación sigma delta síncrona, esto incluye un comparador análogo y un conversor análogo digital para comparar la señal muestreada con la señal de entrada y solo digitalizar la diferencia cuando la diferencia supera cierto rango determinado.

9.- El conversor análogo digital desarrollado, tiene como base de control el protocolo "handshake" de 04 fases, una mejor implementación desde el punto de vista energético se podría alcanzar si se plantea la solución usando el protocolo "handshake" de 02 fases, que teóricamente no desperdicia recursos, puesto que usa los dos flancos, aunque requiere el uso de una mayor carga lógica para su funcionamiento, sin embargo aun con ese inconveniente podrá conservar sus ventajas.

10.- La ventaja de los circuitos digitales asíncronos es que funcionan a partir de la presencia de eventos, en el modulo desarrollado la única parte que no tiene este concepto es el comparador analógico, por lo tanto, el desarrollo de un elemento adicional entre la parte análoga y digital, que tenga como función cambiar entre un estado "sleep" y "active" a los comparadores análogos según la presencia o no de la señal análoga de entrada, es decir, en un determinado tiempo solo están activos dos comparadores que son aquellos que encierran a la señal análoga.

11.- La Implementación de una red de sensores inalámbricos, según el área geográfica de nuestro país, requerirá en algunos casos módulos adicionales para la protección física del nodo contra acciones propias de la naturaleza (lluvia, niebla, helada, etc) y manipulación por parte de animales propios de la zona.

12.- La puesta en marcha de una red de monitoreo requiere la colaboración de todos los actores involucrados (comunidades cercanas, gobiernos locales y las instituciones estatales pertinentes, como el ministerio del ambiente), desde que se realiza el estudio detallado del escenario, pasando por la distribución e instalación de los nodos y finalmente para el seguimiento e interpretación de los datos.

**ANEXO A**  
**Herramientas de especificación y soporte para diseños digitales asincronos**

El desarrollo de la metodología de diseño digital asíncrono necesita una arquitectura de especificación y herramientas CAD para la implementación, actualmente se cuenta con una variedad de soluciones, cuya elección depende del tipo de proyecto que se quiere realizar, a continuación se enumeran y se describen algunas plataformas de desarrollo [11].

### **Signal Transition Graphs (STGs).**

Especifican los circuitos asíncronos mediante redes de Petri (las redes Petri son esquemas compuestos de arcos direccionados y dos tipos de nodos: transiciones y estados). En STGs las transiciones equivalen a los cambios en las señales y los estados y arcos direccionados capturan las relaciones causales entre las transiciones de la señal. Los STGs son considerados una sub-clase de las redes Petri, los estados pueden ser marcados con tokens y permitir su ejecución solo si el token está presente. Representan una formalización de los diagramas de tiempos para el caso asíncrono.

### **Lenguaje de programación CSP (communicating sequential process)**

Es un lenguaje propuesto por Hoare, encaja de forma natural con las características propias del diseño digital asíncrono, como son: la comunicación usando “handshakes” y los procesos que altamente concurrentes; permite a diferencia de los lenguajes clásicos como VHDL la composición en paralelo de una sentencia en un proceso.

CSP es un lenguaje para procesos concurrentes y es base de lenguajes más específicos para procesos digitales asíncronos como son: CHP, Balsa Y TANGRAM.

### **CHP (communication hardware process)**

CHP usa símbolos especiales con una sintaxis similar a CSP y tiene un conjunto de herramientas que entre otras aplicaciones, permitieron automatizar el proceso de control 04 fases, dual rail.

### **Balsa**

Balsa es un sistema aplicativo para describir y sintetizar circuitos asíncronos. La ventaja principal es que hace una compilación transparente, el mapeo es directo entre el lenguaje de la especificación y el circuito handshake producido. La aplicación es muy fácil de usar, aunque se requiere cierta experiencia del diseñador en circuitos asíncronos y el compilador no garantiza sistemas correctos pero si una correcta construcción, para tener una mayor garantía del funcionamiento final una compilación post layout es requerida.

### **TANGRAM**

Tangram es un conjunto de herramientas con un lenguaje de programación mas tradicional similar al C y Pascal, con sentencia para procesos concurrentes. Un compilador Tangram produce circuitos basados en handshakes usando una librería de mas de 40 componentes handshake. El simulador cuenta con un analizador de potencia y área para optimizar los diseños.

En base a estas herramientas se han desarrollado microprocesadores asíncronos como:

- En la Universidad de Manchester, Inglaterra, se han diseñado e implementado varios microprocesadores de la serie: AMULET2e, Amulet3i
- En Instituto Tecnológico de California, EE.UU, se reporta el microprocesador MIPS R3000 basado en una arquitectura pipeline.
- La empresa Philips Research Laboratories, Holanda, ha desarrollado y utiliza comercialmente el microcontrolador asíncrono 80c51.
- La empresa Theseus Logic, EE.UU, reporta el NCL08, una versión asíncrona compatible del microcontrolador de Motorola HC08.

### **LARD**

Es un lenguaje de desarrollo para describir sistemas asíncronos, verifica el comportamiento y la funcionalidad, Lard interactúa con otros simuladores para permitir la aplicación de los vectores de test. Esta herramienta sirve para modelar controladores, memorias RAM y ROM.

### **PETRIFY**

Permite sintetizar la redes Petri, lee especificaciones en STG, sintetiza ecuaciones booleanas, elementos Cs generalizados y posee una biblioteca. Se usa para la síntesis de microcontroladores.

### **3D**

Denominado también XBM (extend busrt-mode), maneja circuitos secuenciales y maquinas de estado asíncrona de estados finitos. Una de sus principales ventajas es el soporte que brinda a circuitos síncronos, asíncronos y sobre todo a los estados intermedios algo muy valorado por los diseñadores. Es adaptable a la implementación CMOS. Se puede mencionar otras herramientas desarrolladas por centros de investigaciones de universidades alrededor del mundo o por compañías dedicadas a la producción digital, como por ejemplo:

Fulcrum de Microsystem orientado a diseño dual rail, delay insensitive y handshaking de 04 fases.

Theseus NCL se ocupa de los circuitos delay insensitive, multi rail, proporciona so para pasar de VHDL a NCL y permite la verificación usando vectores funcionales, se uso para desarrollar microcontroladores y smartcards.

Pipefitter, herramienta desarrollada por el politécnico de Torino, sintetiza circuitos asíncronos micropipeline a 04 fases, usa como lenguaje de especificación al Verilog, en la simulación analiza retardos y puede generar especificaciones STG.

**ANEXO B**  
**Definición de bloques digitales asíncronos básicos**



En un diseño asíncrono todos los instantes de tiempo son validos, cualquier cambio o transición de alguna señal tiene consecuencias y lleva un significado propio, por lo tanto se deben de evitar los estados inestables de las señales.

Para lograr implementar la señalización REQ y ACK, tanto en 02 y 04 fases, es necesario contar con compuertas que indiquen por medio de sus salidas cambios de eventos en sus entradas [11]. Entre las principales tenemos:

### LATCH DE TRANSICION O DE CAPTURA

El latch de transición es un elemento de almacenamiento. Un cambio en la entrada (D) es retenida en la compuerta cuando que un evento en C ocurre, en este estado el latch mantiene el valor actual del valor de entrada y es insensible a cambios en la entrada de data, a la siguiente transición en P recién el elemento iguala la salida al dato  $Q = D$ . Cuando las entradas Capture y Pass están en el mismo estado (alto o bajo), el latch está en el estado PASS, la salida del latch sigue cualquier cambio en el valor de entrada. El símbolo de este elemento se muestra en la figura B.1.

Para que el El latch de transición opere correctamente, los eventos en C y P deben alternarse, la primera transición debe ocurrir en C y luego seguir con P,

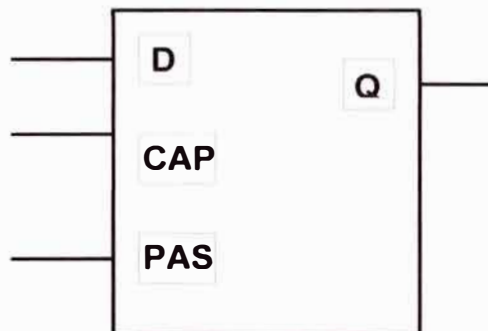


Figura B.1 – Elemento latch de transición.

### TOGGLE

Este elemento cambia de estado a las salidas alternadamente. Las transiciones en la entrada del elemento toggle es dirigida solo a uno de las salidas de manera alternada. El círculo cerca a una de las salidas indica hacia donde se dirige la primera transición, después de un reset. El símbolo de este elemento se muestra en la figura B.2.

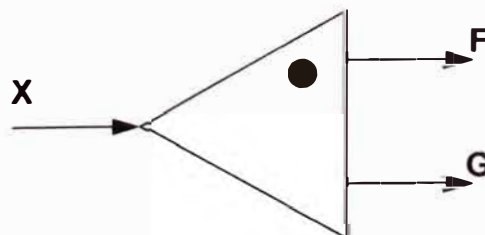
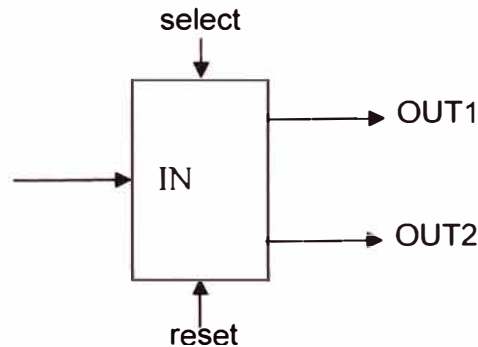


Figura B.2 – Símbolo del elemento Toggle.

Este elemento tiene una restricción que especifica que dos eventos de entrada no pueden ocurrir antes que el primero de ellos no haya sido direccionado a alguna de las salidas.

## SELECT

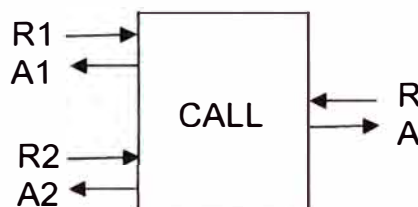
El elemento SELECT direcciona una transición de entrada a una de las dos salidas dependiendo del valor booleano de una segunda entrada, 'selector'. El valor del selector booleano debe ser válido antes que la transición en input ocurra. El símbolo de este elemento se muestra en la figura B.3.



**Figura B.3** – Símbolo del elemento Select.

## CALL

Permite a dos procesos independientes compartir un subproceso común. Al recibir un requerimiento en R1 o R2, el elemento CALL produce otro requerimiento en R para el proceso común. Cuando la subrutina ha sido completada, indicado por la llegada de un evento en el A (subrutina done), el elemento CALL genera un evento output sobre la correspondiente A1 o A2. El símbolo de este elemento se muestra en la figura B.4.

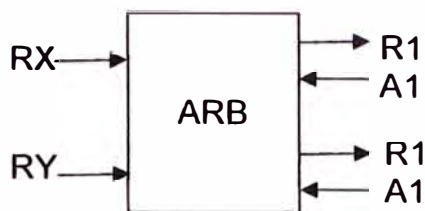


**Figura B.4** – Símbolo del elemento Call.

Para la correcta operación, el ciclo completo request-subrutina request/subrutina do/done, deben completarse antes de que el siguiente request ocurra y más aun los dos input request señal, R1 y R2, deben ser mutuamente exclusivos. Para un circuito topológico donde R1 y R2 no pueden ser garantizados a ser mutuamente exclusivos, el input request puede ser enviado al elemento CALL vía un arbiter.

## ARBITER

Acepta peticiones asíncronas en sus dos entradas y transfiere solo una de ellas, en un instante determinado, hacia el recurso compartido. Cuando concluye dicha petición transfiere la otra petición si aun está pendiente. Usualmente se construye basándose en el denominado elemento MUTEX (mutual exclusión). En el diseño de árbitro debe considerarse el problema de metaestabilidad interna del circuito para garantizar una correcta operación. El símbolo de este elemento se muestra en la figura B.5.



**Figura B.5** – Símbolo del elemento ARBITER.

Cuando el elemento arbiter esta en uso por un RX, el segundo RY es inhabilitado hasta recibir la confirmación correspondiente A1 o A2. Aunque las entradas REQs pueden ocurrir en cualquier tiempo, aun simultáneamente, el elemento deberán garantizar salidas mutuamente exclusivas.

### **ELEMENTO-C GENERALIZADO**

Un elemento-C generalizado es un puerta secuencial en la cual ambos eventos (transiciones de subida y bajada) de salida pueden ser descritas como una conjunción de un sub-grupo de sus entradas. El símbolo para dicho elemento es derivado de elemento C-simétrico. La entrada que contribuye en ambos eventos es conectada a la base del símbolo, la entrada que contribuye en un solo evento es conectada al terminal de extensión, puede ser +/- según la transición del evento.

Elementos C asimétricos tienen entradas las cuales afectan la operación del elemento solo cuando hay un cambio en una de las direcciones. Entradas asimétricas son atadas a menos (-) o mas (+). Cuando hay una transición de 0 a 1 el elemento C tomara en cuenta la entrada común y la entrada asimétrica mas (+). Similarmente cuando hay una transición de 1 a 0 el elemento C tomara en cuenta la entrada común y la entrada asimétrica menos (-)

## BILIOGRAFIA

- [1] José Sanjurjo Vilchez y Risto Kalliola. "Manual para la elaboración de mosaicos de imágenes de satélite Landsat TM para la selva baja peruana" Instituto de Investigación de la Amazonia Peruana documento técnico N° 03 - 2004.
- [2] Joel Young. "Una guía práctica sobre tecnologías de pilas para redes de sensores inalámbricos" DIGI INTERNATIONAL – 2008.
- [3] Mr. Hoang Duc Chinh, Dr. Yen Kheng Tan. "Smart Wireless Sensor Networks" – 2010.
- [4] Ian F. Akyildiz, Mehmet Can Vuran. "Wireless Sensor Networks"- 2010.
- [5] Sebastián Román Maroto Cantillo, "Desarrollo de aplicaciones basadas en WSN", Escuela Técnica Superior de Ingeniería Informática, Universida Politecnica de Valencia – 2010
- [6] Paolo Santi. "Topology Control in Wireless Ad Hoc and Sensor Networks"- 2005.
- [7] Waltenegus Dargie, Christian Poellabauer. "Fundamentals of wireless sensors networks"- 2010.
- [8] Dr. Geoff V. Merret, Dr. Yen Kheng Tan. "Wireless Sensor Networks: Application Centric Design" - 2010.
- [9] Dr. Winston Seah, Dr. Yen Kheng. "Sustainable Wireless Sensor Networks"- 2010.
- [10] Rohit Vaish, "Application of wireless sensor networks for environmental monitoring and development of an energy efficient cluster based routing", Department of Electrical Engineering, National Institute of Technology Rourkela INDIA- 2009
- [11] Jens Sparso, Steve Furber. "PRINCIPLES OF ASYNCHRONOUS CIRCUIT DESIGN – A Systems Perspective"- 2001.
- [12] Ruben V. Alarcon Matutti. "Síntesis de Circuitos Digitales Asíncronos Aplicado a Comunicaciones: una Aproximación Micropipeline para QoS-ATM". Universidad Autónoma de Barcelona – 2003.
- [13] Ivan Sutherland. "Micropipelines communications of the ACM" Vol 32, No6, pp.720-738 IEEE – 1989.
- [14] Jon W. Mark, Terence Todd "A .Non-uniform Sampling Approach to Data Compression". IEEE. 1981
- [15] Marc Renaudin. "Spectral analysis of level-crossing sampling shceme". TIMA Laboratory, Instituto Politecnico de Grenoble – Francia. 2006.

- [16] Sangil Park, Ph. D. "Principles of Sigma-Delta Modulation for Analog-to-Digital Converters"- 2010.
- [17] Leon W. Couch. "Digital and Analog Communication Systems" – 2006
- [18] E. Allier, J. Goulier, G. Sicard, A. Dezzani, E. André, M. Renaudin "A 120nm Low Power Asynchronous ADC". TIMA Laboratory, Instituto Politecnico de Grenoble – Francia 2007.
- [19] Karen Guan, Andrew C. Singer "A Level-Crossing Sampling Scheme for Bursty Signals", Princeton IEEE - 2006
- [20] R. Shavelis. "Signal Reconstruction from Multiple Level Crossings Using Asymmetric Constructing Functions". Instituto de Electronica Letonia IEEE – 2007.
- [21] Dariusz Koscielnik, Marek Miskowicz. "Designing Time-to-Digital Converter for Asynchronous ADCs". Universidad de Ciencia y Tecnologia IEEE – Polonia. 2006.
- [22] Filipp Akopyan, Rajit Manohar, Alyssa B. Apsel. "A Level-Crossing Flash Asynchronous Analog-to-Digital Converter". Cornell University – USA IEEE. 2006
- [23] "Microwind & Dsch User's Manual Version 2" Department of Electrical & Computer Engineering. Tolouse Francia – 2003