UNIVERSIDAD RICARDO PALMA
FACULTAD DE INGENIERIA
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRONICA

ESTIMACIÓN DE LA RESPUESTA DE UN SENSOR DE TEMPERATURA A USARSE EN LA CARGA UTIL DEL COHETE SONDA PERUANO MEDIANTE EL FILTRO DE KALMAN USANDO LOGICA RECONFIGURABLE

TESIS PARA OPTAR EL TÍTULO DE INGENIERO ELECTRONICA

AUTOR
Fernando Rodolfo, Raymundo Luyo

LIMA – PERÚ 2008

A mis Padres Rodolfo y Margarita, por haberme apoyado desde el comienzo para salir adelante, por haberme educado para tener logros en la vida, además por haberme dado todo de si para mi educación y gracias a ellos voy a lograr un gran objetivo para nuestra familia.

AGRADECIMIENTO

A Dios por haberme guiado, ordenado y dado muchas alegrías en mi vida, gracias por haberme dado la oportunidad de conocerte.

A mis Padres Rodolfo y Margarita, por haberme apoyado desde el comienzo para salir adelante, por haberme educado para tener logros en la vida, además por haberme dado todo de si para mi educación y gracias a ellos voy a lograr un gran objetivo para nuestra familia.

A Carlos, pues siempre me admiro de lo cuan aplicado y responsable es en sus cosas.

A Arturo Puchuri, que gracias a su perseverancia me enseño mucho sobre Dios y su esposa Mariela Benites, que desde muy joven me ha hecho conocer a Dios, muchas gracias.

A Brenda, ella sin ninguna duda es una persona que para mi vale mucho y que me sabe comprender, gracias por ser alguien especial en mi vida, realmente el aliento que me das me ayuda a superarme.

A Mis tías Maria, Karina, Vilma, Isabel, Martha, Delia y demás tíos y tías por haberme ayudado en mi crecimiento y por enseñarme cosas de la vida.

A mi tío Luis Sanchez, por su ayuda y paciencia en los trabajos de mecánica.

A mi tío Pablo, te voy a cumplir la promesa que te hice.

A mis abuelos Paulina y Simón, por su ayuda y su abnegación con todos los suyos.

A mis primos y primas, gracias por dejarse enseñar, mas adelante van a ver los frutos de sus esfuerzos.

A todos mis compañeros de CONIDA, a Roberto Campos por su paciencia y su ayuda, a Percy Paz, por sus consejos y explicaciones divertidas de física, a Juan por su puntualidad y dedicación a sus trabajos. A Diana Vasquez y Luz Cori, por enseñarme que las chicas son muy habilidosas en instrumentación.

A los astrónomos, Walter Guevara, Mara Pelayo, Gabriel Ferrero, Vanesa y Hernán, pues hacían siempre un trabajo excepcional para la investigación, mucho aprendí de ustedes.

A mis compañeros de Cidib, a Benjamin Gaspar por su amistad y ayuda en mi crecimiento profesional, y también por la colaboración en el desarrollo del oxímetro de pulso, a Henry Dávila por apoyarme en otras ideas para las soluciones de los problemas de electrónica, y su forma madura de ver las cosas, a Percy Benavente, por ayudarnos a ver las cosas del ámbito emprendedor.

Al Ing. Gustavo Roselló, por haberme dado invitado al camino de la investigación y al desarrollo tecnológico, desde el circuito de lógica combinacional para display de 7 segmentos hasta el filtro de kalman.

Al Ing. Manuel Márquez, sin duda, los consejos y la parte teórica que nos inculco me ayudo mucho en mi parte técnica de desarrollo de hardware analógico.

A los profesores de mi universidad, Institutos, colegios, pues cada uno de ellos me ha enseñado algo más de los libros.

A Ana Llosa por darme confianza y por orientarme en la presentación de mi dossier.

A Maribel Saavedra, es una profesora que realmente le importa el avance de sus alumnos, gracias por tu amistad, siempre te voy a recordar por tu interés y amistad.

Al profesor Michel Doisy, por darme la oportunidad de ir a Francia y estudiar en una de las mejores escuelas de ingenieros y por apoyarme desde el comienzo para ir a estudiar allí.

Al profesor Oscar Penny por haberme inculcado en el diseño en control automático, en procesamiento de señales y dado una orientación y visión, también por haberme ayudado en la comprensión del filtro de kalman, debido a su gran experiencia en el tema.

Al Profesor, Guillermo Kemper, por su apoyo en la investigación del filtro Herrera Bendezu y su interés en la investigación.

A David y Mary Tay, gracias por su amistad, de verdad los estimo mucho, gracias por ser tan sencillos y serviciales.

A mi “abuelita” Julia y don Isidro Obando, por su acogedor recibimiento en su hogar, nunca me voy a olvidar de ello, se lo aseguro.

INTRODUCCIÓN

El área de instrumentación científica en C.O.N.I.D.A. (Agencia Espacial del Perú), es la dependencia la cual es encargada de implementar las cargas útiles de los cohetes sonda. Una de las actividades de esta dependencia es la caracterización de la señal de los sensores para los cohetes sonda; en esta actividad se encontró el problemas de ruido, y una de ellas era causado por falta de puesta a tierra de los equipos (fuente de alimentación, osciloscopios, etc.).

En el transcurso del desarrollo de la primera carga útil, se trato de eliminar el ruido con un filtro de fase lineal y centrado en 60 Hz hecho en labview, pero el filtro no proveía del todo una señal deseable para la caracterización.

El uso de los sensores se ha hecho muy frecuente en las aplicaciones de las investigaciones, pero en muchos casos no se caracterizan los sensores que se usan con lo cual se tienen malos resultados en las mediciones.

Se sabe que las hojas de datos de los sensores, nos muestran graficas de la respuesta de estos con condiciones óptimas para su caracterización (como una excelente puesta a tierra) ya que tienen patrones estandarizados y sin problemas de circuitos mal dimensionados (adaptación de impedancias). Cuando se compara la respuesta real de los sensores en los laboratorios, no se acercan a las curvas características que se proponen en las hojas de datos ya sea por ruido o por circuitos mal dimensionados.

En la caracterización de los sensores de la carga útil del cohete sonda, se tuvo problemas de ruido, como por ejemplo, ruido de línea, EMI, acoplamientos capacitivos e inductivos, corrientes y voltajes parásitos entre otros, por lo que este trabajo va a poder describir estos ruidos para poder comprenderlos, analizarlos y modelarlos.

Sabiendo que hay diferentes formas de eliminación del ruido, como por ejemplo el de eliminación del ruido de línea por medio de circuitos con transformadores isoladores, o

filtros de línea (filtros notch) en hardware, que no son reconfigurables, y habiendo obtenido una respuesta no optima en las caracterizaciones de los sensores con el filtro de fase lineal, y a su vez de la influencia de otros ruidos, este trabajo propone implementar el Filtro de Kalman, el cual usa conceptos de procesos aleatorios, en el FPGA V2P30 de lógica reconfigurable, para estimar la señal del sensor de temperatura dentro de la señal con ruido.

Este trabajo se divide en las siguientes partes:

En el primer capitulo, tenemos los fundamentos generales de los cohetes sonda, cargas útiles y la descripción de los ruidos que han podido afectar a la caracterización.

En el segundo capitulo, se explicará el análisis del ruido por medio de procesos estocásticos, se dirá que tipo de ruido se esta eliminando, y también se dará un modelo para analizarlo y por consiguiente, se van a nombrar los ruidos que se pueden encontrar en los sistemas aerospaciales.

En el tercer capitulo, se analiza el sensor mas apropiado para este trabajo, como también se muestra la forma de caracterización del mismo, se describirá los problemas que pasan por la caracterización y por ultimo se mostrara el diseño de la interfase implementada del sensor de temperatura.

En el cuarto capitulo, se explicara los fundamentos teóricos del filtro de kalman, y el desarrollo del mismo.

En el quinto capitulo, se explicara las características de los dispositivos de lógica reconfiguradle, como también el proceso de diseño con los FPGAs, y por último se describirá los tipos de FPGAs para los sistemas aerospaciales.

En el sexto capitulo, se describirá la implementación del filtro de kalman en el FPGA, empezando con el diseño de la arquitectura, después el diseño usando system generador y se presentara la co-simulación con la tarjeta XUP2VP30.

En el séptimo capitulo se darán a conocer los resultados de las simulaciones y las co-simulaciones que se hicieron con el hardware.

En el octavo capitulo se darán las conclusiones del presente trabajo.

Para poder complementar algunos temas teóricos y prácticos de la tesis se presentan los siguientes apéndices.

En el apéndice A, se darán a conocer más propiedades de la media y la varianza.

En el apéndice B, se darán a conocer más propiedades de la covarianza.

En el apéndice C, se mostraran los códigos Utilizados en Matlab.

En el apéndice D, se describirá los significados del Histograma y PMF.

En el apéndice E, se explicara por medio de un ejemplo los significados de exactitud y precisión.

En el apéndice F, Se describirán los conceptos básicos de la termocupla.

En el apéndice G, se describirá un ejemplo Implementado con un filtro de kalman.

En el apéndice H, Se mostrara como se obtiene la ganancia del filtro discreto de kalman, K.

En el apéndice I, se dará una breve introducción a los números flotantes.

En el apéndice J, se dará a conocer el Translation report y Pads.

En el apéndice K, se dará una breve introducción de importación de Cores en Simulink.

Y por ultimo se van a colocar algunos anexos del dato técnico del AD595 y un trabajo relacionado.

CAPITULO 1. COHETE SONDA

1.1 Cohete Sonda.

El cohete sonda, es una nave que transporta instrumental de investigación astrofísica en una trayectoria parabólica, también puede efectuar experimentos a determinadas altura (entre 40 km y 200km de altura). Se desplaza por la atmósfera debido a la fuerza de reacción que produce su motor de propulsión, que es producto de la combustión de su combustible sólido, la cual libera gases a velocidades muy altas, por la tobera.

En la figura 1, se muestra un cohete sonda.

Figura 1. Cohete sonda peruano.

Las aplicaciones están comprendidos en:

- Exploraciones de alta atmósfera.

La exploración o sondaje de la alta atmósfera se realiza en donde no pueden llegar los balones (llegan solo hasta los 40 Km) ni los satélites que orbitan mas allá de 200km.

Los primeros conocimientos sobre el ambiente terrestre (ionosfera, magnetosfera, etc.) fueron adquiridos por los Estados Unidos y la Unión soviética, en el siglo 20, en la utilización de las versiones modificadas del misil balística V2 alemán.

- Investigación en microgravedad.

Los experimentos proveen mucha información, que incluye procesos físicos que toman lugar en la atmósfera, como la radiación natural que se encuentra alrededor de la tierra, proveniente del sol, de las estrellas, y las galaxias.

Es una aplicación muy reciente en la cual permite a los científicos, experimentos que conciernen sobretodo en la ciencia de los materiales (experiencias de fusión y de solidificación) que fueron de interés a los Estados Unidos, Japón y Europa, que usó en 1982, un programa de ese tipo con los cohetes sonda Texus, y después con los Maxus.

También se tomaron pruebas biológicas con animales como perros, ratas gatos, etc.

Más aún, los cohetes sonda proveen un costo económico en relación al test de instrumentos de ingeniería y componentes usados en los satélites y otras naves espaciales.

Los cohetes sonda, tienen como aporte tecnológico el desarrollo de nuevas tecnologías y materiales, lo cual es apto para la formación, investigación y entrenamiento de profesionales y estudiantes en muchas ramas como la física, química, ingeniería electrónica, ingeniería aeronáutica, ingeniería mecánica, ingeniería química entre otras. Esto contribuye a la generación de recursos humanos para las exigencias del desarrollo aerospacial.

Las partes del cohete sonda se detallan a continuación y se muestran en la figura 2.

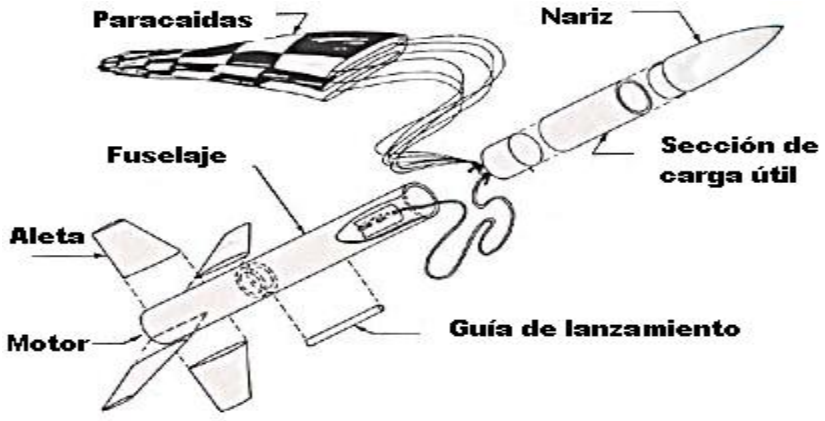


Figura 2. Partes del Cohete Sonda.

- **Nariz,** Puede Contener al paracaídas y el sistema de control de altitud.
- **Carga útil,** contiene la electrónica necesaria para adquirir datos, transmitirlos y guardarlos.
- **Sistema de control de Altitud**, Controla la orientación de la carga útil usando un microcomputador.
- **Cuerda de choque,** es usado para sostener a la carga útil.
- **Fuselaje,** es usado como revestimiento principal del cohete sonda y también para llevar dentro el motor del cohete sonda.
- **Aleta**, son los elementos aerodinámicos necesarios para la estabilización estática y automática del vuelo de un cohete.
- **Motor**, su principio de funcionamiento se basa en la tercera ley de newton, la cual las partículas expulsadas por la tobera hace que haya una reacción que da lugar al movimiento por la dirección opuesta de la nave.
- **Guía de lanzamiento,** se engancha sobre la rampa de lanzamiento.
- **Telemetría**, transmite todos los datos a la estación terrena, incluyendo los datos de investigación, y otras señales, incluso puede recibir comando para conducir al cohete.
- **Tobera:** La tobera es la encargada de adaptar las presiones internas de la cámara de combustión impartiendo velocidad a los gases eyectados, convirtiendo la presión de gases calientes generados en la cámara de combustión de un motor cohete en energía cinética que será útil para su propulsión, empleando finalmente la Tercera Ley de Newton. En un motor cohete, la tobera está formada por dos conos unidos por sus vértices, uno de ellos convergente (su diámetro mayor está orientado hacia la cámara de combustión y su vértice apunta hacia la salida del motor) y el otro divergente (mayor en su salida que a la entrada); la unión de ambos conos es la sección de menor superficie de la tobera y se la llama “garganta”. Se considera que el flujo que recorre a una tobera es compresible al moverse a velocidades supersónicas, por lo que las diferentes secciones transversales de la tobera producen, durante el avance de los gases, variaciones en la densidad y en la velocidad del fluido.

La trayectoria del cohete sonda se muestra en la figura 3.

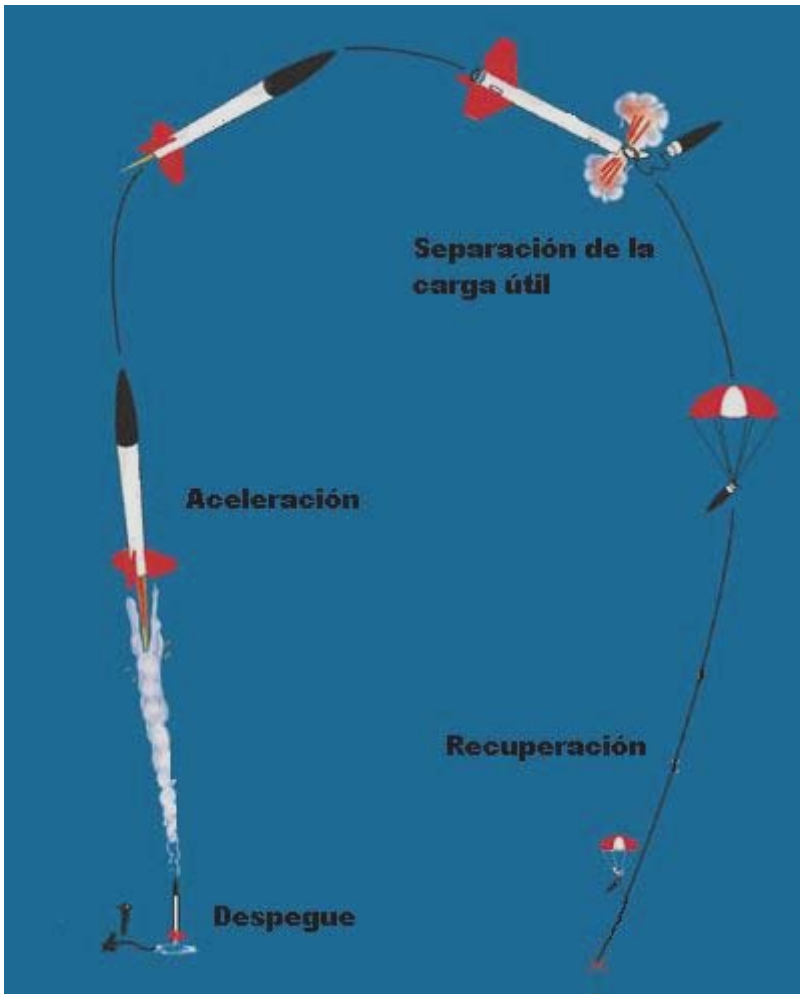


Figura 3. Trayectoria Parabólica.

1.2 Carga Útil.

La carga útil, es la parte electrónica del cohete sonda; ella tiene por misión, la de procesar los datos que vienen de los sensores de temperatura, presión, corriente, velocidad. También su otra función es la de almacenar los datos recibidos y poder transmitirlos vía RF a una estación terrena.

Muchos cohetes sonda contienen sistemas electrónicos para el momento de eyección de la carga útil antes de que el vehículo espacial caiga a tierra.

Dependiendo de la misión se crea una carga útil, sin embargo hay algunas cosas que se tienen en común en las cargas útiles de los cohetes.

En las siguientes figuras se muestra la carga útil del cohete sonda peruano Roberto 1, que son adaptadas de [16].

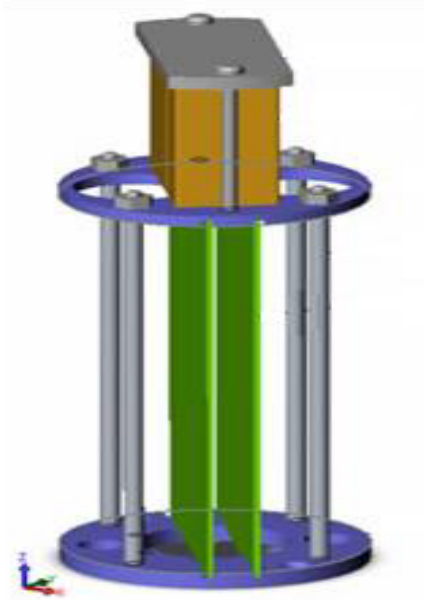


Figura 4. Esquema de la carga útil Roberto 1.

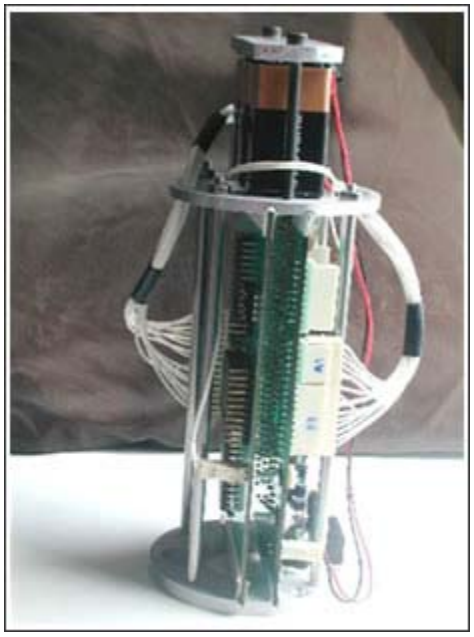


Figura 5. Carga Útil implementada Roberto 1.

Figura 6. Otra vista de la carga Útil implementada Roberto 1.

En la figura 7 se muestra la carga útil prototipo del cohete sonda Paulet I.

Figura 7. Carga Útil implementada para el cohete sonda Paulet I.

En los elementos de la carga útil se puede mencionar los siguientes:

- Sistema de adquisición de datos.
- Sistema de Transmisión de datos.
- Sistema de eyección.

1.2.1 Sistema de adquisición de datos.

En este sistema se tiene los circuitos de acondicionamiento de señal para obtener los valores de las mediciones de los sensores que nos dan una relación muy estrecha con un voltaje. Estas pueden ser almacenadas en bancos de memoria para después ser analizadas en tierra, o también pueden estar en conexión con el sistema de transmisión de datos para el envío en tiempo de real de los mismos.

1.2.2 Sistema de Transmisión de Datos.

Generalmente en este sistema cuenta con un receptor y transmisor de datos, con la cual se comunica con una estación terrena para las operaciones de monitoreo remoto, envió de datos de la carga útil.

1.2.3 Sistema de eyección.

Consiste en poder eyectar a la carga útil cuando se logra el apogeo del cohete, en función de parámetros de presión y temperatura.

1.3 Problemas de Ruido.

En la parte de instrumentación es necesario tener en cuenta las fuentes de ruido para poder realizar adaptaciones en los circuitos de acondicionamiento de señal. Estos ruidos pueden afectar de una manera muy significativa en la caracterización del circuito de acondicionamiento de señal de los sensores, y algunos como el acoplamiento conductivo se pueden evitar.

Para el presente trabajo, durante la caracterización del sensor de temperatura, se tuvo el problema de ruido por la falta de puesta a tierra la que causa inducción de corriente que se añade un voltaje a las mediciones de los sensores.

En los siguientes párrafos se va mostrar los tipos de ruido que existen, y a detallar su origen.

En la figura 8, la cual es una figura adaptada de [15], nos muestra la forma más común de medir voltaje en los sistemas electrónicos, pero en esta forma de medir existe una deficiencia, pues se crea un voltaje inducido al utilizar la referencia a la tierra del sistema para medir la señal con referencia a tierra. En este caso, la medición de tensión, Vm, es la suma de la señal de voltaje, Vs, y la diferencia de potencial, ΔVg , que existe entre la fuente que esta a referencia a tierra y el sistema referenciado a la tierra del sistema. Esta diferencia de potencial no es en general una señal DC, por lo que el resultado es un ruido de línea (60 Hz) en sus componentes de frecuencia en las lecturas. El bucle a tierra, presenta ruido y puede tener dos componentes AC y DC, además de introducir errores de offset en las mediciones. La diferencia de potencial entre las 2 tierras causa una corriente en las interconexiones. Esta corriente se llama corriente de bucle de tierra.

Las tierras de las señales con esta misma referencia no están necesariamente al mismo potencial, la diferencia que puede existir dentro de un mismo edificio puede estar comprendido entre 10mv a 200mv, dependiendo si hay conexiones conectadas correctamente.

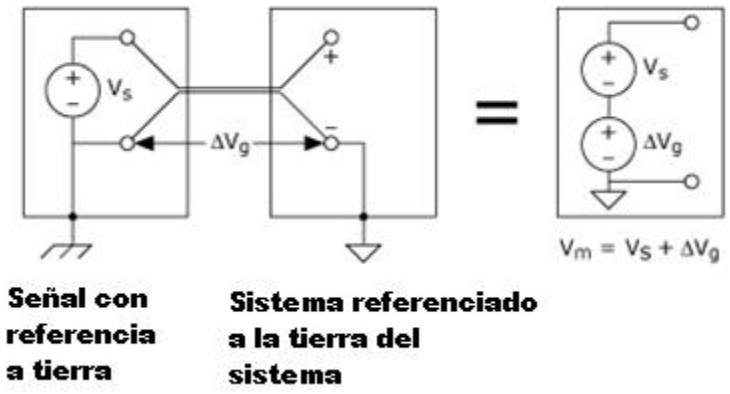


Figura 8. Típico caso de medición de voltaje.

En la figura 9, es una figura adaptada de [15], nos muestra los tipos de ruido.

Figura 9. Tipos de ruido.

Los ruidos pueden ser:

- Ruido Acoplado conductivamente.
- Acoplamiento capacitivo.
- Ruido acoplado inductivamente o magnéticamente.
- EMI.
- Bucle a tierra.
- Ruido del inter sistema.

1.3.1 Ruido Acoplado conductivamente.

El ruido acoplado conductivamente existe porque el cableado tiene una impedancia finita y resulta de compartir las corrientes de diferentes circuitos en impedancias comunes. Cuando se implementa un circuito electrónico, se debe tener en cuenta la impedancia del cable. En la figura 10, es una figura adaptada de [15], nos muestra este acoplamiento.

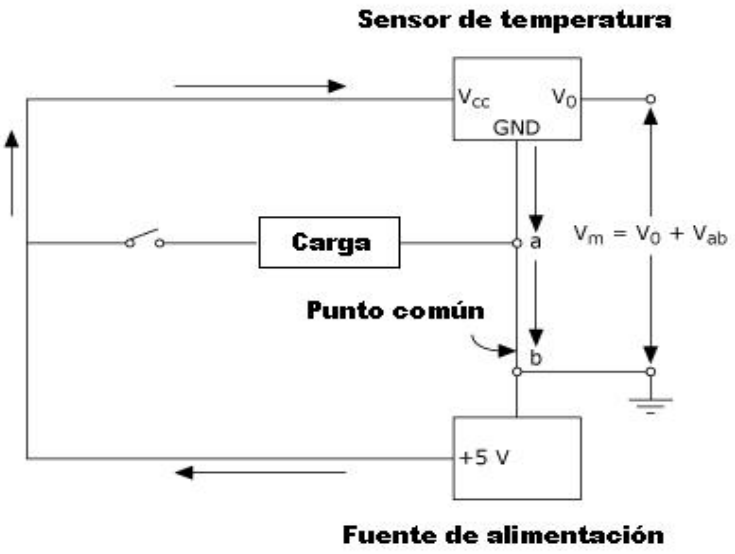


Figura 10. Ruido acoplado conductivamente.

En la figura 11 vemos como podemos evitar ese acoplamiento.

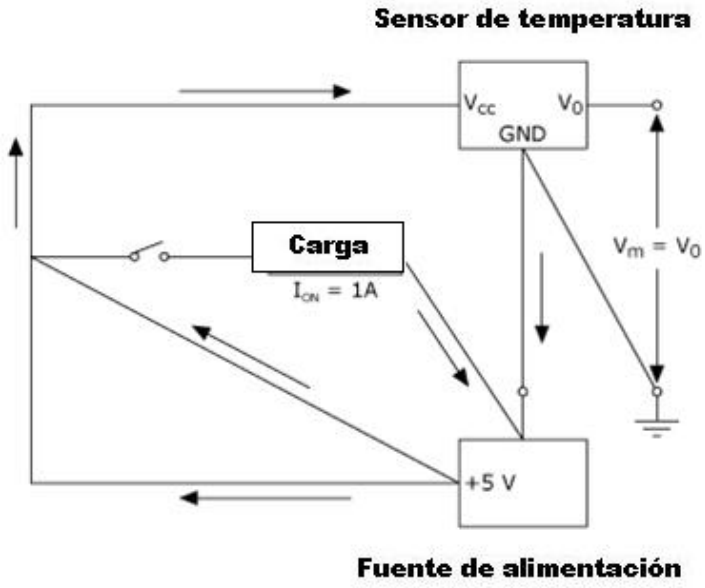


Figura 11. Circuito sin acoplamiento conductivo.

1.3.2 Acoplamiento capacitivo.

El acople capacitivo resulta del campo eléctrico variable que se encuentra en la vecindad del circuito de la señal de fuente o del sensor. En la figura 12 se muestra la representación

física y en la figura 12, la cual es una figura adaptada de [15], nos muestra el circuito equivalente.

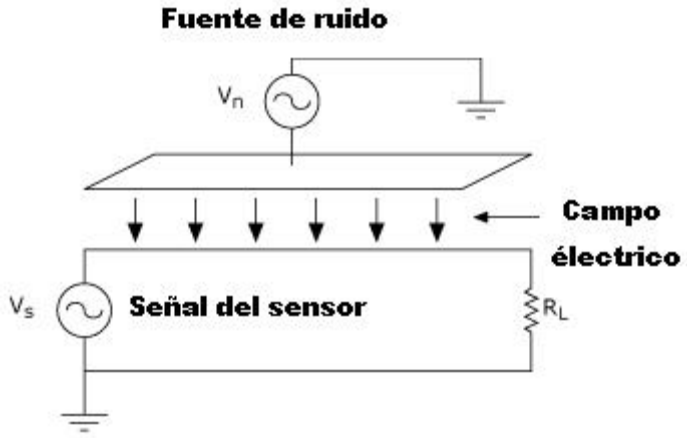


Figura 12. Representación Física del acople capacitivo.

En la figura 13, es una figura adaptada de [15], nos muestra el circuito equivalente del acoplamiento.

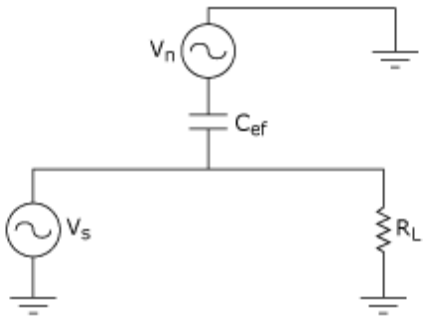


Figura 13. Circuito equivalente del acople capacitivo.

1.3.3 Ruido acoplado inductivamente o magnéticamente

Este ruido resulta del campo variable en el tiempo en el área muy cercana a la señal del circuito. Esto se puede ver en la figura 14, es una figura adaptada de [15].

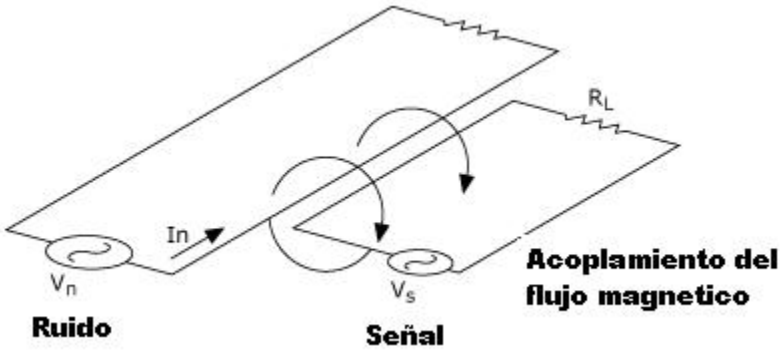


Figura 14. Representación física del acople inductivo.

En la figura 15, es una figura adaptada de [15], que nos muestra el circuito equivalente del acoplamiento.

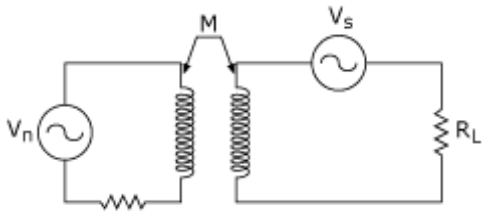


Figura 15. Circuito equivalente del acople inductivo.

El voltaje inducido puede obtenerse de la ecuación 1:

$$Vn = 2\pi fMI_n \text{ ..... (1.1)}$$

Esta ecuación esta relacionada con la inductancia mutua M. Donde I_n es la corriente ruidosa RMS y f es la frecuencia. También M, es directamente proporcional al área del circuito e inversamente proporcional a la distancia entre la fuente de ruido y la señal del circuito.

Para determinar la corriente de bucle de lazo, tenemos los siguientes valores.

I_n = 2 A; f = 60 Hz; and M = 1 µH/m para un cable de 10m.

$$Vn = 2\times\pi\times 60\times\left(1\times 10^{-6}\times 10\right)\times 2 = 7.5mV.$$

1.3.4 Acoplamiento radiactivo o EMI.

Estas son disturbios que afectan un circuito eléctrico debido a la radiación electromagnética emitido de una fuente externa. Este disturbio puede interrumpir, obstruir o degradar la performance o rendimiento del circuito. La fuente es generalmente un objeto natural o artificial que induce rápidamente corrientes eléctricas cambiantes.

Un ejemplo se muestra en la figura 16, como se contamina el acondicionamiento de señal del sensor de temperatura.

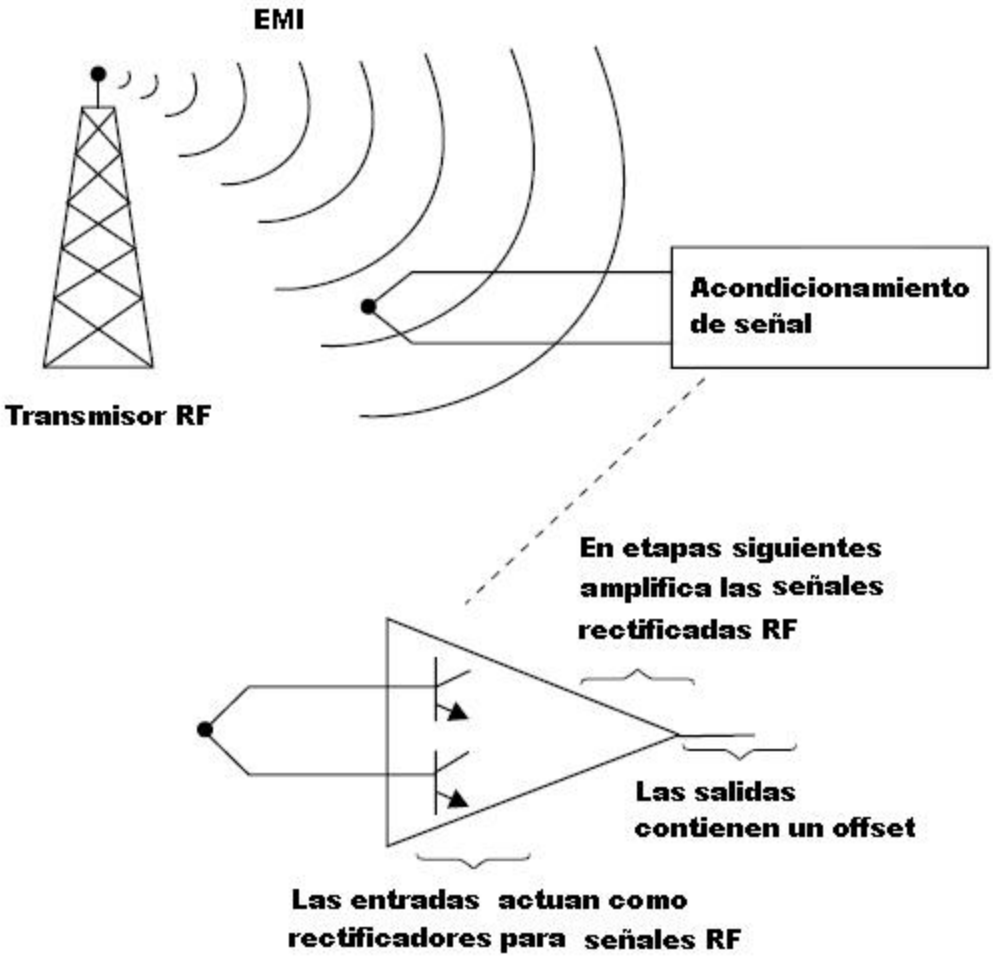


Figura 16. EMI.

La primera categoría comprende radiadores intencionales como los transmisores de radio y televisión, radioaficionados, teléfonos celulares, sistemas de navegación electrónicos, etc., siempre que sea RF las señales que están deliberadamente emitidas.

La segunda categoría está compuesta por radiadores no intencionales tales como computadoras, la televisión, equipo de música, en la parte del equipo de oficina como las impresoras, copiadoras, máquinas de fax, luces fluorescentes, herramientas eléctricas, y las líneas de energía. Cualquier equipo que puede ser apagado y prendido rápidamente es una fuente de ruido potencial.

Esta categoría causa la mayoría de los problemas de los sistemas, dispositivos.

También tenemos que nombrar las fuentes naturales de EMI, como el relámpago, la radiación cósmica, la radiación solar, y la radiación nuclear.

Los circuitos de alta impedancia son más susceptibles de acoplamiento capacitivo cerca de los circuitos rápidos y con grandes oscilaciones de tensión, y de acoplamiento inductivo cerca de los circuitos con rápidos cambios en grandes corrientes. Estos son transitorios momentáneos en el voltaje y la corriente en un tiempo muy corto. Se pueden medir desde milisegundos a nanosegundos. A menudo estos transitorios es llamado voltaje o corriente de pico.

La mayoría de los equipos electrónicos tiene un filtro EMI en la entrada de la fuente de alimentación. La FCC (comisión federal de comunicaciones de USA) recomienda estos filtros para detener la mayoría de ruido llevado a cabo desde las líneas de suministro hasta la fuente de alimentación. Lamentablemente, el ruido puede encontrar otros caminos en las tarjetas electrónicas. La EMI puede ser radiada y puede acoplarse dentro del sistema a través de los conectores metálicos o a través de las líneas de datos. Los cables UTP son un probable candidato para la recolección de ruido. Esto es especialmente probable si no hay suficiente puesta a tierra, o el cable pasa junto a una fuente de ruido.

El ruido inducido puede entrar en el sistema a través del suelo. Por el cable de tierra de las señales eléctricas, pues estos no desaparecen misteriosamente, sino viajan por el camino de menor resistencia y, a veces, regresan a su punto de origen como un dispositivo, la tierra e incluso al suelo.

Muchos problemas en las computadoras se originan de causas eléctricas o magnéticas. Los problemas del monitor por ejemplo a menudo son causados por campos magnéticos, armónicos del cable neutral, o ruido eléctricos conducido o radiado.

Generalmente se contaminan en los picos positivos y negativos de tensión, como se muestra en la figura 17.

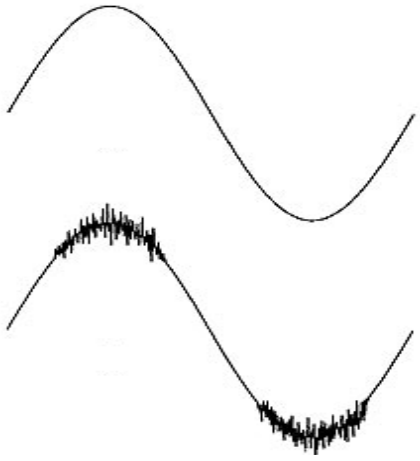


Figura 17. Contaminación con ruido.

1.3.5 Bucles de Tierra.

Uno de los problemas más difíciles de entender, diagnosticar y resolver es el de bucle de tierra. Todos los equipos son susceptibles a este tipo de problema, ya sea médico, industrial, o el de procesamiento de datos. Los bucles de tierra puede causar errores de datos, fallas en los componentes, y en el peor de los casos, incluso causar riesgos para la integridad de las personas.

Muchos equipos, tales como lámparas, tienen solamente un tipo de enchufe de dos hilos, lo que es necesario para entregarle la corriente alterna. Para el equipamiento de instrumentación electrónica, se equipa con un conector de AC de 3 cables. . El tercer alambre es un alambre de tierra de seguridad, que está conectado con todas las piezas exteriores expuestas al metal en los equipos del laboratorio.

Aterrar se utiliza principalmente para asegurar la seguridad y los riesgos de incendio y peligros. Un aspecto importante de esta protección es la confianza en múltiples o

redundantes puesta a tierra. De esta forma, si una puesta a tierra es Extraído o desconectado accidentalmente hay una línea de seguridad. Esta redundancia tiene un importante efecto secundario, que puede crear bucles de tierra.

Aterrar es utilizado también para apoyar al revestimiento en las líneas de transmisión pues proveen blindaje para evitar que las emisiones radiadas entren o salgan.

Cuando se forman bucles de tierra, la corriente que fluye en el sistema de tierra es muy Imprevisible. Esta corriente de tierra puede ser causada por diferencias de tensión, inducción de otros cables o dispositivos, fallas de cableado, fallas de tierra. Las corrientes pueden ser DC, 60 Hz, o de muy alta frecuencia.

Los bucles de tierra pueden causar problemas de los equipos específicos de tres maneras:

1. Corrientes de Baja energía en las tierras generan tensiones que pueden causar errores de datos. Estos pueden ser de baja frecuencia, como zumbidos de 60 Hz o de alta frecuencia clasificado como ruido eléctrico.

2. Transitorios de alta energía escogen tierras de circuitos de datos en lugar de tierras de alimentación. Estos transitorios pueden ser causadas internamente de las corrientes de conmutación o irrupción.

Por ejemplo, la carga inicial de la entrada de los condensadores en una fuente de alimentación, el prendido del motor inductivo, y, por supuesto, el relámpago. Estos transitorios pueden causar daños en el equipo en los puertos, microprocesadores y casi cualquier componente eléctrico si el aumento es lo suficientemente alto.

3. Bucles de tierra son una de las causas de ruido de modo común entre fases, neutral y Tierra, en un sistema de distribución de energía. Este ruido se inyecta en el suministro de energía eléctrica, que a su vez pasa a los componentes electrónicos.

1.3.6 Ruido de tierra del ínter sistema.

Las diferencias del voltaje de tierra se refieren como ruido de tierra del ínter sistema.

El ruido de tierra del ínter sistema no es igual que el ruido común, este es uno de los malentendidos más extensos del campo de la protección de la energía de computadora.

El ruido de modo común se define como ruido que existe entre los conductores de la energía, activo y neutral, con respecto al conductor de tierra, debido a las sobretensiones. El ruido de tierra del ínter sistema existe entre los alambres de tierra entre las computadoras interconectadas. Las computadoras libres que no están conectadas sin ninguna línea de datos no pueden experimentar el ruido de tierra del ínter sistema.

Una razón dominante por la que hay tanta confusión en esta área proviene el hecho de que el equipo de protección de la energía puede reducir ruido de modo común, pero no puede reducir el ruido de tierra del ínter sistema correcto.

El diagrama simplificado de ruido de tierra del ínter sistema de un sistema interconectado ideal se demuestran en la figura 18, la cual es una figura adapta de [14], Los alambres que ponen a tierra a los equipos interconectados vienen de la misma fuente, asegurándose de que tienen el mismo voltaje.

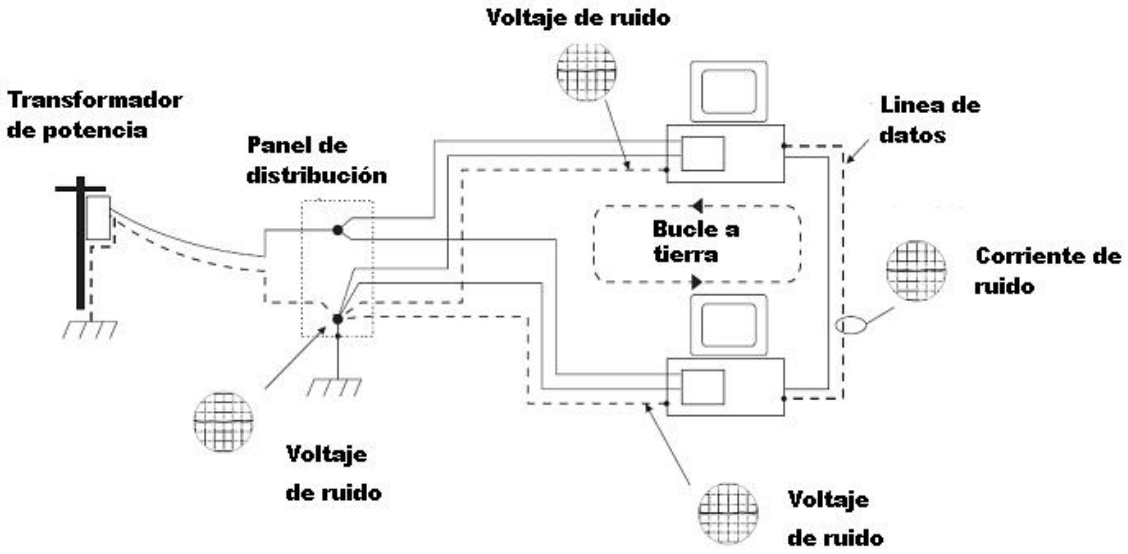


Figura 18. Sistema con puesta a tierra.

Idealmente, ninguna corriente de ninguna clase fluye en alambres de tierra, y no están sujetos a ningún campo magnético. Esto se asegura de que no pueda haber voltaje a lo largo

de los alambres de tierra. El resultado es que todos los puntos a lo largo de todos los alambres de tierra están en el mismo voltaje y que no hay ruido de tierra del ínter sistema entre los dispositivos unidos en los varios puntos a lo largo del sistema que pone a tierra. Los osciloscopios en la figura no demuestran ningún ruido actual. Desafortunadamente, los efectos significativos previenen la realización del panorama ideal precedente. En algunos casos, la desviación del resultado ideal de la poder en la corrupción de los datos e incluso daños del hardware.

Un sistema interconectado que experimenta ruido de tierra del ínter sistema se muestra en el figura 19, la cual es una figura adapta de [14].

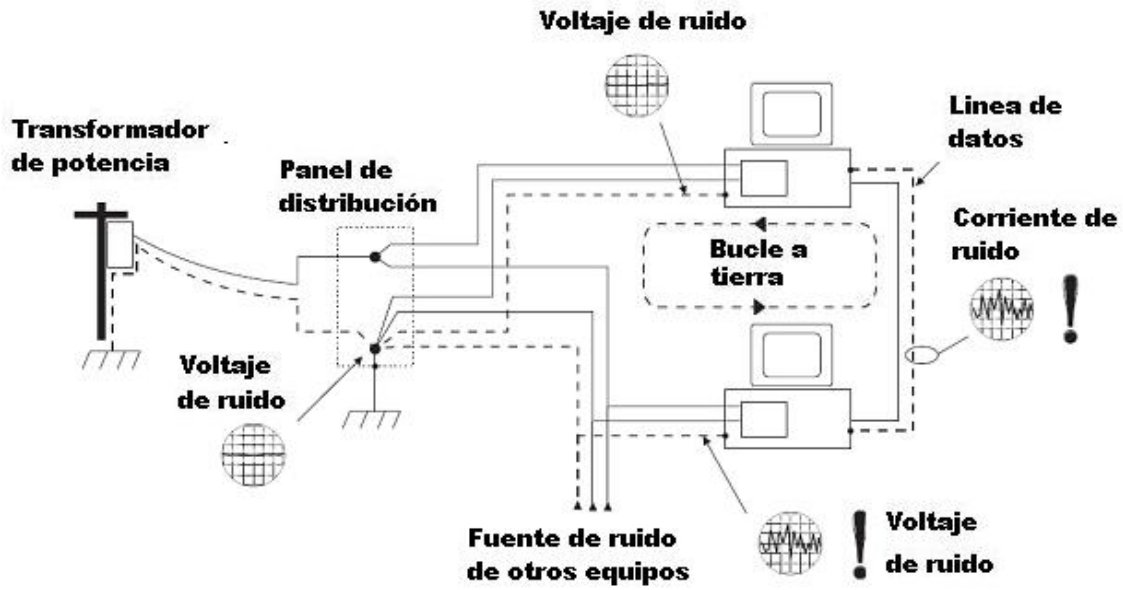


Figura 19. Ruido de tierra del ínter sistema.

En la figura 19, nos muestra que los sistemas interconectados están conforme a diversos voltajes de ruido a tierra, dando lugar a la corriente de tierra en la línea de datos que conecta los dos sistemas. El ruido de tierra en este caso resulta de la corriente del ruido inyectada en el sistema de tierra de una de las computadoras por “otros equipos.” El ruido de tierra del ínter sistema puede resultar de una variedad de diversos problemas, como los siguientes:

- **Inyección de ruido de tierra,** Aunque el alambre de tierra de seguridad es para poner a tierra el chasis del equipo, todo el material informático utiliza este alambre para otro

propósito: para proporcionar un punto de referencia para filtrar hacia fuera las emisiones indeseadas de interferencia de la radiofrecuencia de los equipos informáticos. El ruido eléctrico de modo común y normal en la línea de energía es inyectado en la línea de tierra por el filtro presente en la fuente de alimentación de cada computadora y Workstation. Los supresores incorrectamente diseñados pueden contribuir a este problema. EFECTO: Esta corriente inyectada de ruido da lugar a un voltaje del ruido entre las referencias comunes del equipo interconectado. Porque los voltajes del ruido llegan a ser generalmente más grandes cuando la distancia entre el equipo aumenta, las transmisiones de datos pueden comprometerse en algunas situaciones.

- **Fallas de tierra,** Si los dispositivos interconectados están en diversos circuitos eléctricos en el edificio y uno de estos circuitos es compartido por otro equipo, que tiene una falta de aislamiento, después de las conmutación del interruptor, una corriente de avería muy grande será inyectada en el alambre de tierra de seguridad. Esto dará lugar a una oleada momentánea del voltaje en el material informático que es proveído por el interruptor. El tamaño de esta oleada de voltaje puede ser de algunos voltios sobre los picos del voltaje de la energía (120VAC o 230VAC), dependiendo de la calidad del sistema que pone a tierra. EFECTO: Esto dará lugar a una diferencia del voltaje entre los puntos de referencia comunes del equipo interconectado que puede exceder fácilmente el grado del voltaje de la seguridad de la señal de los datos. La destrucción de los conductores de la entrada-salida y de las placas base de la CPU.

- **Corrientes de tierra,** Éste es el problema identificado más común. Si los equipos interconectados son provistos por diversos paneles eléctricos (cajas del interruptor) en el mismo o diversos edificios, después muchos problemas pueden causar que el voltaje de tierra medido en los dispositivos interconectados sean de diferentes valores. Dependiendo de códigos eléctricos locales o nacionales, muchos cableados diversos de los sistemas son separados entre los sub-paneles. La mayor parte de estos sistemas del cableado de los paneles no garantizan que los voltajes de tierra provistos por los sub-paneles sean iguales y es por ello que se crean una diferencia de potencial la cual da origen a una corriente.

Con la descripción de los ruidos descritos, estos se suman o acoplan al sistema de medición y generan malas mediciones no precisas y sin exactitud.

En los siguientes párrafos se va a modelar el ruido como el total de los ruidos que pudieron haber afectado al sistema de medición.

CAPITULO 2. ANALISIS DEL RUIDO

En esta parte, se va a tratar sobre el modelamiento matemático del ruido total, los ruidos que se describieron en la parte anterior e influyente en la señal acondicionada del sensor de temperatura.

Es necesario saber que una señal siempre es afectada por ruido más o menos importante. Esas fluctuaciones pueden ser diversas y son llamadas ruido eléctrico o simplemente ruido. Algunas fluctuaciones son invisibles a los osciloscopios por el ancho de banda predefinido en el momento de la visualización en el osciloscopio.

Una causa externa es producida cuando el ruido es afectado ya en la entrada de la señal de un amplificador, la cual aumenta también el ruido en su salida, por ejemplo una radio capta no solo las frecuencia de emisión de una estación radial sino también de los ruidos industriales, el ruido cósmico y todas las otras emisiones de ondas electrónicas.

2.1. Análisis del ruido por medio de Procesos Estocásticos

La probabilidad y la estadística se utilizan en procesamiento digital de señales para clasificar y medir las características inusuales las señales y los procesos. Por ejemplo, un uso primario de PDS (procesamiento digital de señales) es reducir la interferencia, y otros componentes indeseables en datos adquiridos. Estas interferencias pueden dar a lugar a las imperfecciones en el sistema de adquisición de datos.

En los próximos párrafos se va describir el modelamiento con procesos randomicos con su respectiva parte de estadística para el modelamiento del ruido.

Para poder avanzar con el trabajo de investigación, tenemos que saber con qué clase de ruido estamos trabajando, por ello tenemos que saber si la señal ruidosa tiene alguna característica para poder describir una función matemática. En la figura 20 se muestra la señal con ruido.

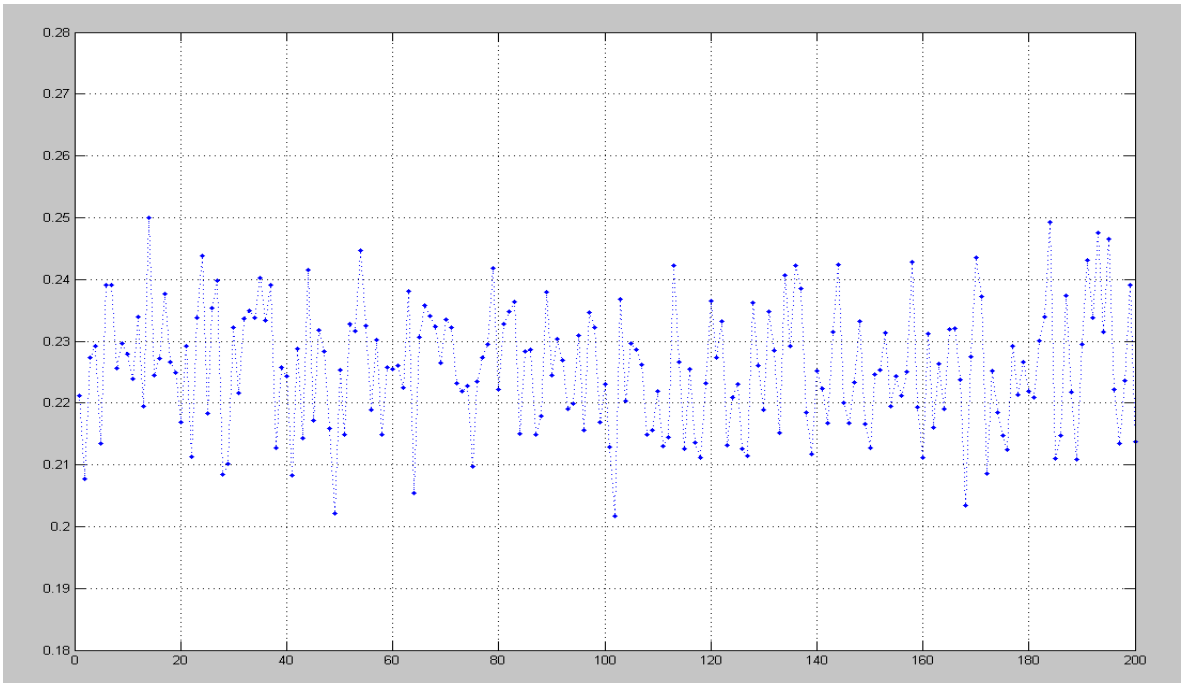


Figura 20. Señal con ruido.

Si notamos en la figura 20, no podemos describir una ecuación matemática fácilmente, es por ello que debemos hacer un modelamiento de su comportamiento, no por una ecuación.

Para ello tenemos que conocer más de este tipo de señales para poder hacer una caracterización correcta. En el siguiente párrafo vamos a describir a las variables aleatorias para poder entender mejor a este tipo de señales.

2.1.1 Variables aleatorias discretas

Una variable aleatoria es una función definida en el espacio muestral. En otras palabras, cada una de los posibles resultados del experimento determina un posible valor de la variable aleatoria. Por lo tanto, el valor tomado por la variable aleatoria no se conocerá hasta que el experimento se ha llevado a cabo.

Para nuestro caso, este ruido total es asumido como ruido aleatorio primeramente. Hay que mencionar que este ruido no esta en correlación con el tiempo.

Vamos a aplicar las propiedades de las variables aleatorias para probar si el ruido es aleatorio.

Las propiedades de una variable aleatoria son las siguientes:

- Debe ser una función.
- Si X es una variable aleatoria y x un número, entonces $\{X \leq x\}$ es un evento, y la probabilidad de este evento $P\{X \leq x\}$, es la suma de las probabilidades de los elementos correspondientes al intervalo $\{X \leq x\}$.
- $P\{X = -\infty\} = 0$ y $P\{X = \infty\} = 0$

Para nuestro caso del ruido, veamos si cumple esas propiedades:

- La señal con ruido de la figura 20, es una función, pues a cada elemento solo le corresponde un valor de su rango.
- Hay elementos que pueden estar comprendido dentro un rango $\{X \leq x\}$, por ejemplo: si $\{x = 0.23\}$ entonces podemos definir lo siguiente: $\{X \leq 0.23\}$, lo cual es verdadero y nos muestra que existe una cantidad de números que están dentro del rango de esta condición.
- Para afirmar esta otra opción, $P\{X = -\infty\} = 0$ y $P\{X = \infty\} = 0$ debemos darnos cuenta de los valores que esta tomando la variable aleatoria, y el conjunto que esta tomando la señal del ruido es aproximadamente: $0.20 \leq X \leq 0.25$, lo cual indica que la variable no puede tomar los valores en el infinito.

Con esto ya probamos que este ruido es una variable aleatoria.

2.1.1.1 Características de las variables aleatorias discretas.

Sólo pueden tener un determinado número de resultados o de valores, y tienen una probabilidad positiva; típicamente son valores enteros obtenidos por conteo. Por ejemplo, la cantidad de resistencias con 10% de tolerancia.

2.1.2 Función de distribución de variables aleatorias discretas

Para entender más el comportamiento del ruido, vamos a explicar la función de distribución de la variable aleatoria.

Es también llamado la *función de probabilidad*, y describe la probabilidad que la variable X toma un valor menor o igual al número x.

$$F_x(x) = P\{X \leq x\} \text{ …….. (2.1)}$$

Las siguientes propiedades son características de las funciones de probabilidad:

- $0 \leq F_x(x) \leq 1$
- $F_x(-\infty) = 0$
- $F_x(\infty) = 1$
- $F_x(x_1) \leq F_x(x2)$ si $x_1 \leq x2$.

La función de probabilidad es también llamada distribución de probabilidad, y es una lista de probabilidades asociadas a sus posibles valores.

También se puede escribir de la siguiente manera:

$$F_x(x) = \sum_{i=1}^{N} P(x_i)u(x - x_i) \text{ ……(2.2)}$$

En donde $u(.)$ es la función de paso unitario.

2.1.3 El teorema del límite central.

La base matemática del análisis del ruido total que afecta a la señal se describe con el teorema del límite central, para poder comprender y poder caracterizar el ruido total.

En su forma más simple, el teorema del límite central establece que una suma de números aleatorios pasan a ser una *distribución normal o gaussiana* a la medida que más y más números aleatorios son añadidos. El teorema del Límite central no requiere que los números aleatorios sean de alguna distribución particular, o incluso que los números aleatorios *sean de la misma distribución*.

El teorema del límite central proporciona la razón por la cual las señales distribuidas normalmente son vistas ampliamente en la naturaleza.

Definiendo:

Sean x1, x2,....., xn, n variables aleatorias independientes idénticamente distribuidas con media μ y varianza σ^2 ambas finitas. La suma de esas variables Sn = x1+x2+ ...+ xn es una variable aleatoria con media nμ y varianza nσ^2, entonces

$$Z = \frac{S_n - n\mu}{\sqrt{n\sigma^2}} \quad \text{.....(2.3)}$$

Se distribuye como una distribución normal , en otras palabras, el teorema expresa que cuando n crece sin límite, la variable z tiende a distribuirse normalmente. Si las variables no son idénticamente distribuidas, se podría demostrar igualmente que:

$$Z = \frac{\sum x_i - \sum \mu_i}{\sqrt{\sum \sigma_i^2}} \quad \text{.....(2.4)}$$

Se distribuye como una distribución normal, es decir que la suma de variables independientes tiende a ser una distribución una normal que esta formada la media con una suma de medias y una varianza con suma de varianzas.

Para nuestro caso podemos aproximarlo de esta manera:

$$Z \cong X \quad \text{.....(2.5)}$$

Con ello analizamos de una manera general el ruido.

2.1.4 La media y desviación estándar de la señal con ruido.

En esta parte, vamos a describir al ruido con sus respectivos valores estadísticos, primero vamos a describir la media y después la desviación estándar del ruido total.

2.1.4.1 La media.

Es el valor estadístico que se obtiene de añadir todos las muestras juntas de la señal aleatoria, y dividir por N (el numero de muestras). Se escribe de la siguiente forma:

$$u = \frac{1}{N}\sum_{i=o}^{N-1} x_i. \quad \text{…….. (2.6)}$$

En otras palabras, es la suma de los valores o muestras de la señal, x , con el índice i de 0 a N - 1. Luego terminar el cálculo dividiendo la suma por N. Este es Idéntica a la ecuación:

$$u = \frac{1}{N}\left(x_0 + x_1 + x_2 + \ldots + x_{N-1}\right). \quad \text{…… (2.7)}$$

Si la señal es una simple forma de onda repetitivas, como una condición seno o de onda cuadrada, sus excursiones pueden ser descritos por el valor pico a pico de amplitud.

2.1.4.2 Desviación estándar.

Lamentablemente, la mayoría de las señales adquiridas no tienen bien definido el valor pico a pico, pero tienen un carácter aleatorio, como las señales en la figura 21, la cual es adaptada de [7]. Para este tipo de señales se utiliza la desviación estándar, denotado por σ .

La desviación estándar es una medida de hasta qué punto la señal oscila de la media.

Como punto de partida, la expresión, $|x_i - u|$, describe hasta qué punto la muestra i se diferencia de la media. Tomamos el valor absoluto de cada desviación.

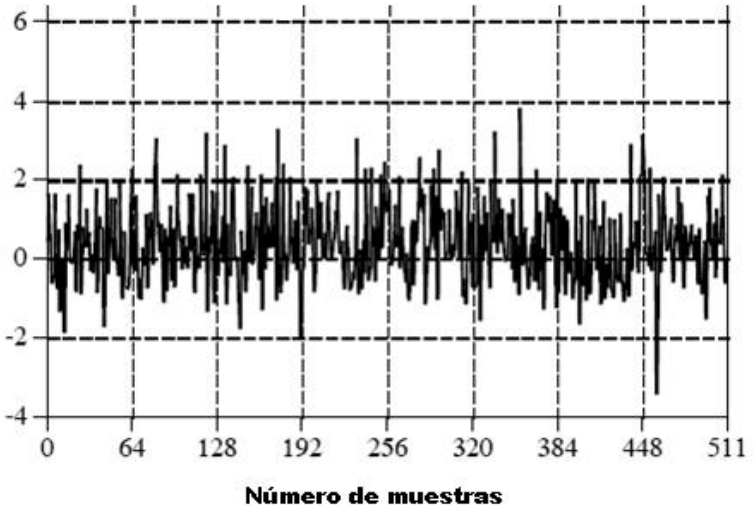


Figura 21. Señal aleatoria. $u = 0.5,\ \sigma = 1.$

La desviación estándar se escribir de esta manera:

$$\sigma = \sqrt{(x_0 - u)^2 + (x_1 - u)^2 + ... + (x_{N-1} - u)^2 / (N-1)} \quad \text{..... (2.8)}$$

2.1.4.3 La varianza.

La varianza de una distribución es el promedio de los cuadrados de las distancias de los valores extraídos a la media de la distribución.

La varianza se calcula:

$$\sigma^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - u)^2 \quad \text{..... (2.9)}$$

Note que el promedio se realiza dividiendo por N - 1 en vez de N. El término, σ^2, ocurre con frecuencia en las estadísticas y se le dio el nombre de varianza. Por definición, la desviación estándar sólo mide la porción AC de una señal, mientras que el valor de RMS mide tanto la AC y las componentes DC. Si una señal no tiene componente DC, su valor es de RMS Idéntica a su desviación estándar.

Para muchos casos, colocar la media y la varianza puede ocupar mucho espacio y memoria en el hardware, la ecuación que nos ayuda a reducir hardware es:

$$\sigma^2 = \frac{1}{N-1}\left[\sum_{i=0}^{N-1} x^2{}_i - \frac{1}{N}\left(\sum_{i=0}^{N-1} x_i\right)^2\right] \quad \text{……(2.10)}$$

Es conveniente calcular la media y desviación estándar cuando hay nuevas muestras que se adquieren

2.1.5 Caracterizando la señal.

Para nuestra señal los valores calculados son los siguientes, utilizando 11000 muestras de la señal.

El cálculo de la media de esta señal por medio de la ecuación 2.6 es:

$$u = \frac{1}{11000}\sum_{i=o}^{11000-1} x_i = 0.2261.$$

$$u = 0.2261.$$

El cálculo de la varianza de esta señal por medio de la ecuación 2.9 es:

$$\sigma^2 = \frac{1}{11000 - 1}\sum_{i=0}^{11000-1}(x_i - u)^2 = 1.2130\text{e} \quad \text{- 004.}$$

$$\sigma^2 = 1.2130\text{e} - 004.$$

El cálculo de la desviación estándar por medio de la ecuación 2.8 es:

$$\sigma = \sqrt{(x_0 - u)^2 + (x_1 - u)^2 + ... + (x_{N-1} - u)^2 /(11000 - 1)}$$

$$\sigma = 0.0110.$$

2.1.6 Métodos de análisis de las características del ruido.

Para poder entender mejor el ruido podemos analizarlo y caracterizarlo con sus métodos estadísticos, que son la función de la densidad de la probabilidad del ruido, función de la densidad de masa, el histograma y la función de autocorrelación.

2.1.6.1 La función de la densidad de la probabilidad (PDF)

Se denota como $f_x(x)$, y es definido por la derivada de la función de distribución:

$$f_x(x) = \frac{dF_x(x)}{dx} \quad \text{……. (2.11)}$$

Para el caso de las variables aleatorias discretas, su derivada es la función del impulso unitario $\delta(x)$, para describir las derivadas de las funciones que tienen la función de paso unitario.

Algunas propiedades También:

- $0 \leq f_x(x)$.
- $\int_{-\infty}^{\infty} f_x(x)dx = 1$ y para el caso discreto en una intervalo [a, b]:

$$\sum_{a}^{b} f_x(x) = 1.$$

- Y la probabilidad en un intervalo [a, b] es definido como:

$$P_x[a,b] = \int_{a}^{b} f_x(x)dx.$$

También llamada la función de distribución de la probabilidad, es a las señales continuas como lo es la función de masa de la probabilidad a las señales discretas. Por ejemplo, imagine una señal análoga al pasar a través de un convertidor de analógico a digital, dando por resultado la señal convertida a digital de la figura 22. Para la simplicidad, asumiremos

que los voltajes entre 0 y 255 milivoltios se convierten a digital en números digitales entre 0 y 255.

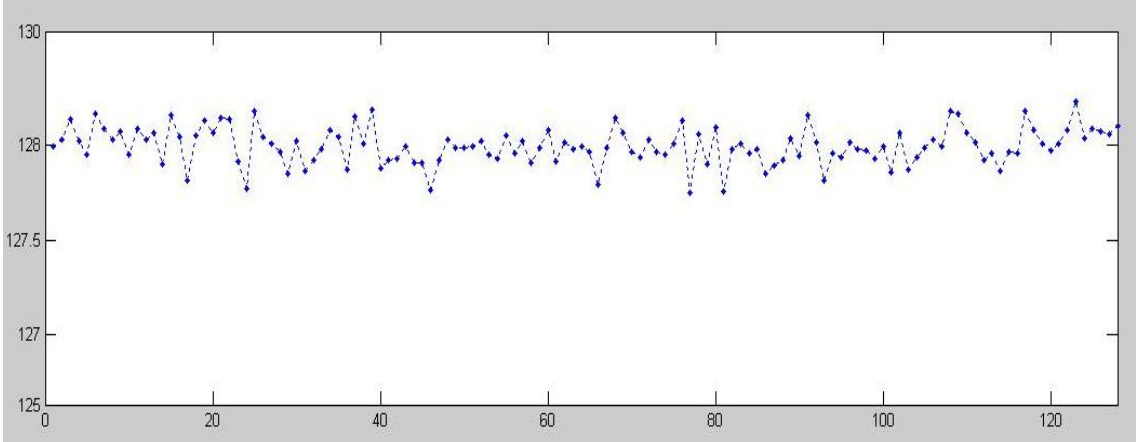


Figura 22. Muestra de una señal aleatoria.

Del mismo modo, la PDF de la señal analógica es mostrada por la línea continua como en la figura 23, la cual es adaptada de [7], nos va indicando que la señal puede tomar sobre un rango continuo de valores, como la tensión en un circuito electrónico.

El eje vertical del PDF es en unidades de densidad de probabilidad, en lugar de sólo probabilidad. Por ejemplo, un PDF de 0,03 a 120,5, no significa que la Tensión de 120,5 millivoltios ocurrirá en el 3% del tiempo.

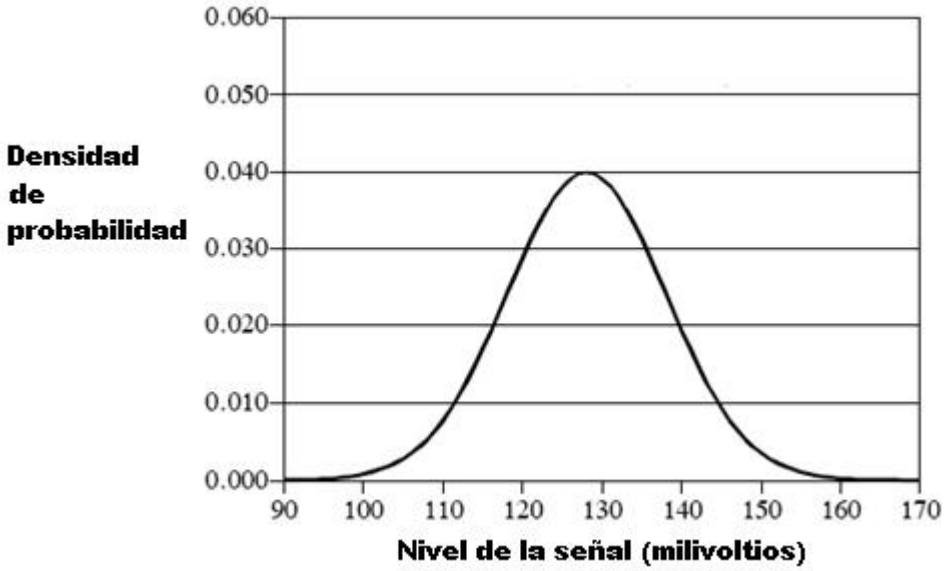


Figura 23. PDF de una señal aleatoria.

Para calcular una probabilidad, la densidad de probabilidad se multiplica por una serie de Valores. Por ejemplo, la probabilidad de que la señal, en un momento determinado, Estar entre los valores de 120 y 121 es:

$$(120-121)\times 0.03 = 0.03$$

La probabilidad de que la señal será de entre 120,4 y 120,5 es: $(120.5-120.4)\times 0.03 = 0.003$. Si el PDF no es constante a lo largo del rango de interés, la multiplicación se convierte en el integral del PDF sobre el rango. En Otras palabras, el área bajo la PDF delimitada por los valores especificados. Desde de que siempre el valor de la señal debe ser siempre un valor, el área total bajo del PDF, de la integral de $-\infty$ al $+\infty$ siempre será 1. Esto es análogo a la suma de todos los valores del PMF van a ser igual a 1, y la suma de los valores del histograma va a ser igual a N.

Para calcular las PDFs, solo basta expandir los niveles de voltaje o del eje y hacia otro sistema de referencia, y representar la cantidad de incidencias que ocurren en un determinado rango de valores, la cual nos ayuda para poder graficar el PDF, y para hallar el otro eje de coordenadas colocar la densidad de probabilidad que está en función a las coincidencias.

En la figura 24.a y 24.b, las cuales están adaptadas de [7], se muestran 2 ejemplos de cómo se puede calcular las PDFs.

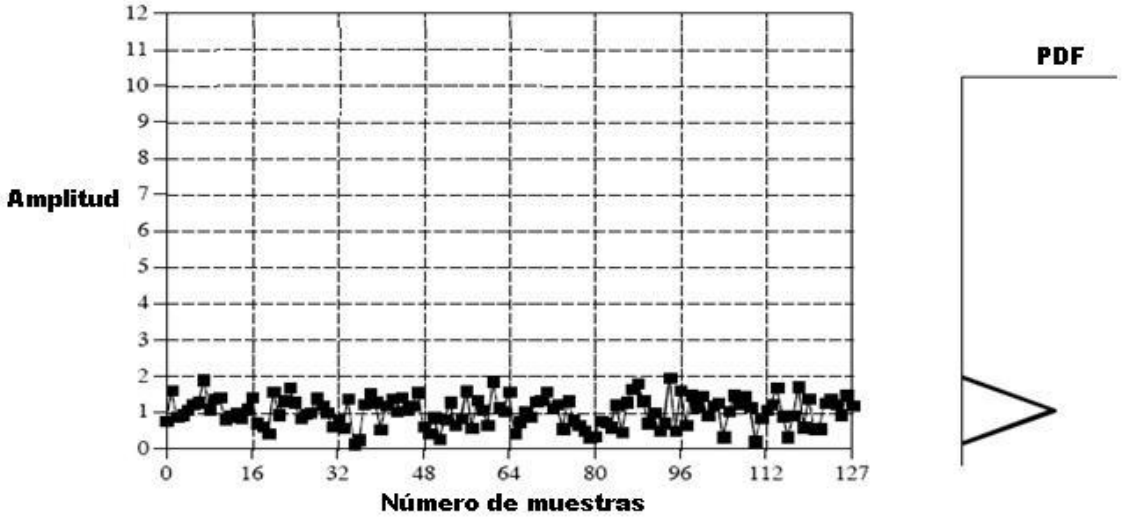


Figura 24.a PDF para una distribución de la forma triangular. $u = 1,\ \sigma = 1/\sqrt{6}.$

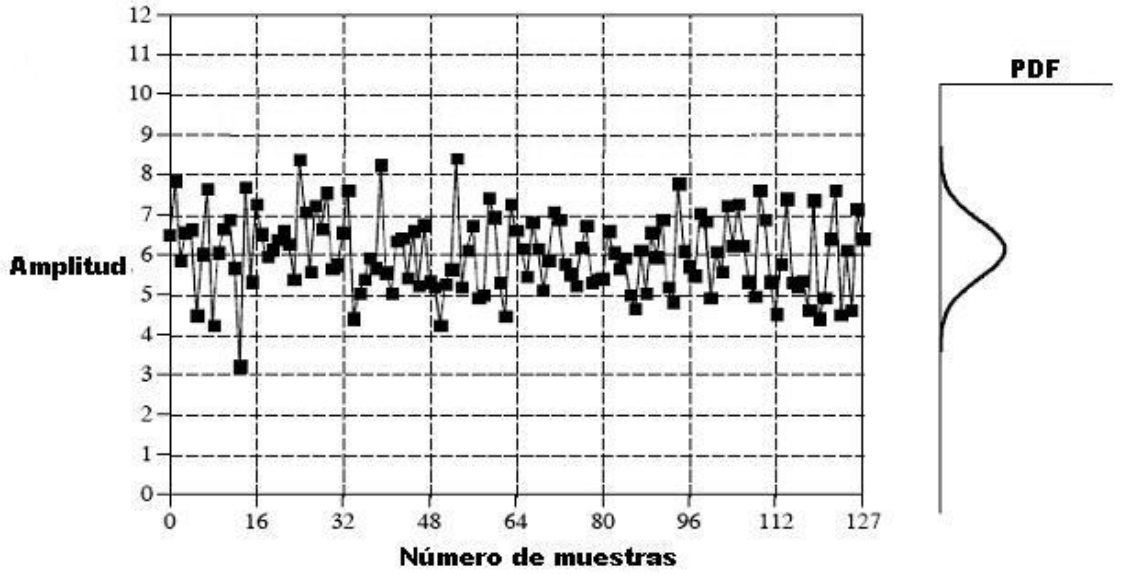


Figura 24.b. PDF para una distribución de la forma Gaussiana. $u = 6,\ \sigma = 1$

2.1.6.2 La distribución normal de procesos aleatorios.

Las señales formadas de un proceso aleatorio usualmente tienen una pdf acampanada. Esto es llamado una distribución normal, o distribución de gauss, o una gaussiana. La forma básica de la curva se genera de un exponente negativo cuadrado:

$$y(x) = e^{-X^2} \quad \text{…….. (2.12)}$$

Esta curva se puede convertir en una campana Gaussiana a agregando una media ajustable u, y una desviación estándar σ. Además, la ecuación se debe normalizar de modo que el área total debajo de la curva sea igual a uno, un requisito de todas las funciones de distribución de probabilidad.

Esto da lugar a la forma general de la distribución normal, a una de las más importantes relaciones en la estadística y la probabilidad:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(x-u)^2 / 2\sigma^2} \quad \text{…… (2.13)}$$

En la figura 24.b muestra un ejemplo de curva gaussiana con una media y desviación estándar. La media centra la curva sobre un valor particular, mientras que la desviación estándar controla la anchura de la forma de la campana.

Una característica interesante de ser Gaussianas es que los extremos caen hacia cero muy rápidamente, mucho más rápido que con otras funciones comunes tales como exponenciales o 1/x.

2.1.6.3 Proceso Gaussiano.

La forma del histograma más estrecha se acerca a la forma de PDF cuando más muestras son utilizadas. Del mismo modo, en el desarrollo de modelos de los sensores con limitados datos, podemos obtener una distribución aproximada de ellos, no exactamente el PDF. Sin embargo, esta medición visual es todavía útil, ya que muestra propiedades importantes de los datos y es fácil de usar sin la complejidad de los cálculos.

En realidad, una exacta solución de este tipo de problema no existe porque, por un modelo probabilístico, limitado de muestras de ensayo nunca podría cubrir todo el espacio de probabilidad.

Nuestro objetivo de modelar un sistema de sensores es construir un modelo probabilístico a través de variables aleatorias o procesos randomicos comúnmente utilizado.

El *Ruido Gaussiano blanco aditivo* (AWGN) es uno de los procesos más típicos aleatorios en estos procesos de medición.

Sabiendo que hay técnicas relacionadas al AWGN que están bien desarrolladas, el cómputo de procesamiento de estas señales sería más simplificado si los sensores de ruido de interés sean AWGN o podría ser una aproximación como AWGN.

Por otra parte, si las características exhibidas por los datos son demasiado complejos para ser modelada de manera fácil y el modelo de precisión no es de interés crucial, podemos implementar la mayoría de los procesos randomicos o combinaciones de las mismas para cercanas aproximaciones. Se puede visualizar una densidad gaussiana en la figura 24.c, la cual es adaptada de [7].

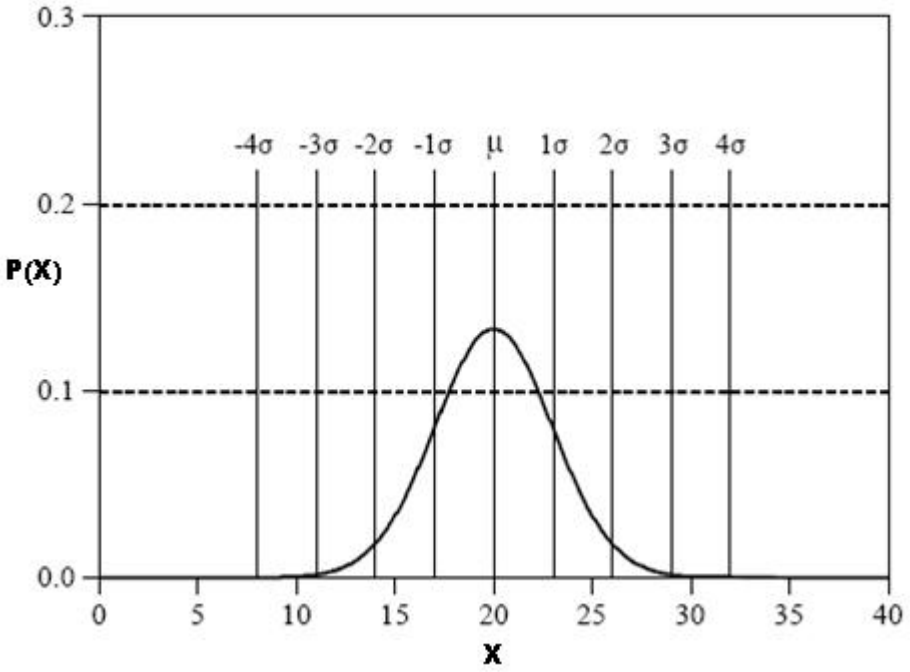


Figura 24.c Figura normalizada con $u = 20$ y $\sigma = 3$.

En nuestro caso, definimos que el ruido total es un ruido gaussiano blanco.

Y la otra forma de caracterizar el ruido es por medio de la autocorrelación, en los siguientes párrafos vamos a definir esta característica.

2.1.6.4 Autocorrelación.

Tanto en los análisis de los sistemas y de las señales, el concepto de autocorrelación desempeña un papel importante, puesto que la función de autocorrelación de una señal aleatoria describe la dependencia general de los valores de las muestras de un tiempo con respecto a valores de muestras de otro tiempo.

Considere el proceso aleatorio $x(t)$, su función de autocorrelación es:

$$R_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)x(t+\tau)dt \quad \text{…….. (2.14)}$$

Donde T es el período de muestreo.

$R_{xx}(\tau)$, es siempre un valor real y una función con un valor máximo en 0.

Para una señal muestreada, la autocorrelación se define como sigue:

$$R_{XX}(m) = \frac{1}{N} \sum_{n=1}^{N-m+1} x(n)x(n+m-1) \quad \text{…… (2.15)}$$

PARA $m = 1,2,...M+1$

2.1.6.5 Covarianza de una señal aleatoria

Cuando tomamos el valor esperado, o media, de un proceso aleatorio, medimos varias características importantes de cómo se comporta el proceso en general. Sin embargo, supongamos que tenemos varios procesos de medición aleatorios de un sistema. La covarianza y la correlación son dos instrumentos importantes para encontrar estas relaciones.

A continuación vamos a entrar en más detalles sobre lo que significan estas palabras y cómo estas herramientas son útiles.

Cuando se trata de más de un proceso aleatorio, la covarianza nos da un valor que nos muestra una idea de cuan similares son los procesos comparados.

Definición:
Una medida de hasta qué punto las desviaciones de dos o más variables o procesos se parecen, relacionan o son se asemejan.

Para 2 procesos, X e Y, si no están estrechamente relacionados con la covarianza entonces esta será pequeña, y si son similares entonces la covarianza será grande. Vamos a aclarar esta declaración de la descripción de lo que entendemos por "relacionados" y "similares".

Dos procesos están "estrechamente relacionadas" si sus distribuciones son casi iguales y están en torno a la misma media, o muy ligeramente en ella.

Matemáticamente, la covarianza es escrita como σ_{xy} y es definido como:

$$\mathrm{cov}(X,Y) = \sigma_{xy} \quad \text{..... (2.16)}$$

$$\mathrm{cov}(X,Y) = E\left[\left(X - \overline{X}\right)\left(Y - \overline{Y}\right)\right] \quad \text{..... (2.17)}$$

Esto puede ser reducido y reescrito de las siguientes 2 formas:

$$\sigma_{xy} = \overline{xy} - (\overline{x})(\overline{y}) \quad \text{..... (2.18)}$$

$$\sigma_{xy} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left(X - \overline{X}\right)\left(Y - \overline{Y}\right)f(x,y)\,dx\,dy \quad \text{..... (2.19)}$$

La propiedad para nuestro caso es la siguiente:

- Si X e Y son independientes y sin correlación o uno de ellos tiene un valor 0 de promedio entonces:

$$\sigma_{xy} = 0.$$

- Covarianza de variables iguales:

$$\mathrm{cov}(X, X) = Var(X)$$

Ahora que hemos analizado el ruido, concluimos con los siguientes parámetros:

La media de esta señal es: $u =$ 0.2261.

La varianza de esta señal es: $\sigma^2 =$ 1.2130e-004.

La desviación estándar es: $\sigma =$ 0.0110.

El ruido total analizado, a partir de ahora se va a caracterizar por ser ruido blanco gaussiano, por tener una distribución normal y ser aleatorio.

2.1.7 Procesos de la señal aleatoria.

Estadística es la ciencia de la interpretación de datos numéricos, como los de las señales adquiridas. En comparación, la probabilidad se usa en PDS para entender el Proceso que generan las señales.

Por ejemplo, la creación de unas muestras de 1000 puntos de una señal tirando una moneda 1000 veces. Si la moneda cae en cara, la muestra correspondiente es hecha al valor de 1. En el caso de sellos se fija en cero. El proceso que ha creado esta señal tiene una media de 0,5 exactamente, determinado por la probabilidad relativa de los posibles resultados: 50% de caras, el 50% de sellos. Sin embargo, es poco probable que el la señal de 1000 puntos tenga una media de 0,5 exactamente. Pues en la conformación de los unos y ceros habrá una leve diferencia cada vez que se genera la señal. Las probabilidades del proceso son constantes, pero la estadística de la señal adquirida cambia cada vez que se repite el experimento.

Esta irregularidad al azar encontrada en datos reales es llamada por los nombres tales como: variación estadística, fluctuación estadística, y ruido estadístico.

En las Figuras 25.a y 25.b, las cuales son adaptadas de [7], nos muestran las variaciones en estas señales son un resultado del proceso aleatorio de las estadísticas, pues es fácil darnos cuenta que estos cambios no son demasiado grandes por el azar, y debe ser relacionado con el proceso. Los procesos que cambian sus características de este modo se llaman no estacionarios. En la figura 25.a, el cambio está en la desviación estándar y la media. En la figura 25.b, la desviación estándar sigue siendo un valor constante, mientras que la media cambia de un valor de cero a dos

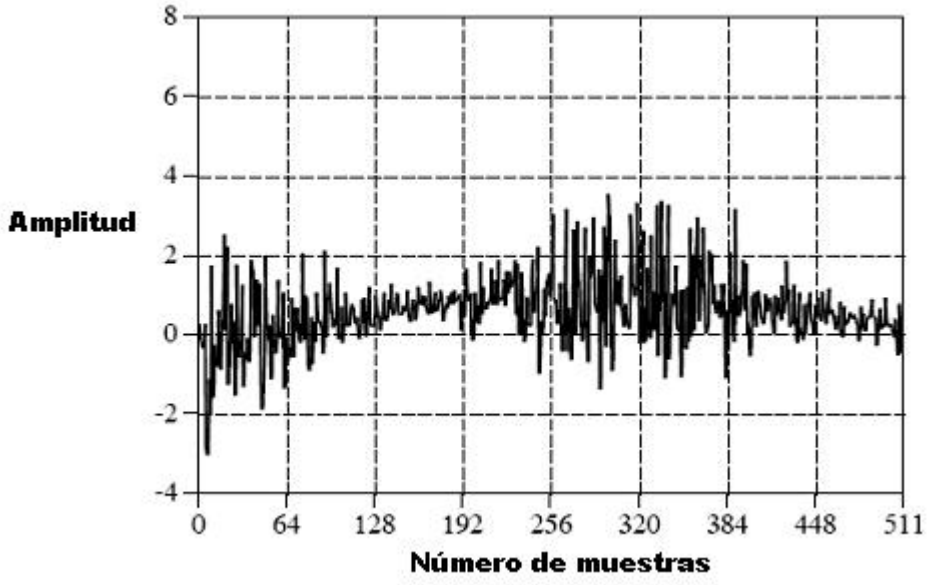


Figura 25.a Cambio de la media y la desviación estándar.

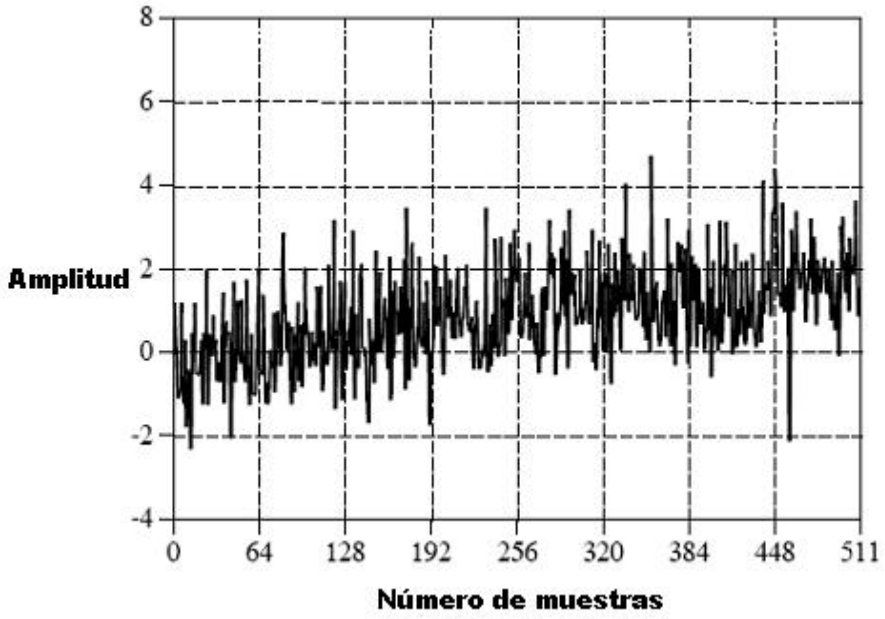


Figura 25.b Cambio de la media y la desviación estándar constante.

En la comparación, las señales que se presentaron previamente en la figura 20, fueron generadas de un proceso inmóvil, y las variaciones resultan totalmente de ruido estadístico. Un problema común se encuentra en la figura 25.b con las señales no estacionarias: el cambio lento de la media interfiere con el cálculo de la desviación estándar. En este ejemplo, la desviación estándar de la señal, sobre un intervalo corto, es 1. Sin embargo, la desviación estándar de la señal entera es 1.16. Este error puede ser eliminado partiendo la señal en secciones cortas, y calculando la estadística para cada sección individualmente. Si es necesario, las desviaciones estándar para cada uno de las secciones se pueden hacer un promedio para producir un solo valor.

2.2. Tipos de Ruidos en las en los sistemas Aeroespaciales.

El entorno espacial no es inofensivo, en sus efectos sobre los sistemas electrónicos aeroespaciales. La comprensión del medio ambiente espacial y sus interacciones es el primer paso para enfrentar estos efectos. Para típicas misiones de la tierra, los siguientes temas deben ser evaluados:

- Campos magnéticos y eléctricos: responsable de torques magnéticos y campos eléctricos inducidos.

- Radiación UV: responsable de fotoelectrones y los cambios a largo plazo en las propiedades de los materiales;
- Radiación IR: conductor de los efectos térmicos.
- Cinturones de Radiación;
- Entorno de partículas (los desechos y micro meteoritos).

La principal preocupación de las interacciones es entonces:

- Los efectos de la radiación acumulativa.
- Un eventos UPSET.
- Latch up.
- Cargas de Superficie.
- Cargas al Interior.
- Pérdida de energía.
- Superficie de degradación y erosión.
- Los desechos espaciales e impactos de los micrometeoritos.
- Efectos térmicos, como la disipación de energía que aumenta.
- El cambio de los voltajes offset en los OPAMS.

CAPITULO 3. SENSOR DE TEMPERATURA

Este sensor deberá detectar los cambios de temperatura como función de la altitud, y fricción con el aire. Al igual que en los acelerómetros, poco se dispone de datos de la temperatura que la carga útil que experimentara. Esta información será ideal para los futuros vuelos, a fin de diseñar futuras cargas útiles que pueden funcionar perfectamente en las temperaturas de vuelo.

3.1 Análisis del Sensor.

En el año 1956, con 2 lanzamientos de cohetes de la NACA, nombre antecesor de la NASA, se hizo en mediciones de temperatura en la superficie de la nariz del cohete, en la cual se obtuvieron un pico de temperatura de 1200 F (648.88 C) a una altura de 18.288 Km. Como el cohete viaja rápidamente, hay consideraciones que se deben tener en cuenta, por ejemplo, a altas velocidades, el calentamiento aerodinámico puede derretir la estructura del cohete. Es debido a que se incrementa la fricción experimentada por el cohete a altas velocidades.

Para este trabajo se decidió utilizar la termocoupla tipo K como sensor de temperatura, pues tiene un rango de trabajo de – 200 °C hasta 1300 °C, la termocupla que se utiliza en este trabajo se muestra en la figura 26.

Figura 26. Termocupla tipo K.

3.2 Caracterización del sensor.

El sensor de temperatura debe operar en los rangos de trabajo que se van a requerir, esto conlleva a tener un óptimo circuito de acondicionamiento de señal para la termocupla de de chromel – Ahlumel tipo k, por lo que se debe pensar en resolver problemas de acondicionamiento para los buenos resultados de las mediciones.

Para poder realizar la caracterización el sensor de temperatura, es necesario hacer una breve descripción del circuito de acondicionamiento de señal, el AD595.

3.2.1 Funcionamiento del Circuito

3.2.1.1 Circuito Integrado AD595

La figura 27, la cual es adaptada de [10], nos muestra el diagrama de bloques del integrado AD595, el acondicionador de señal. Una termocupla tipo K se conecta a los pines 1 y 14, las entradas a un amplificador diferencial. Este es un amplificador que usa la temperatura del chip como su referencia.

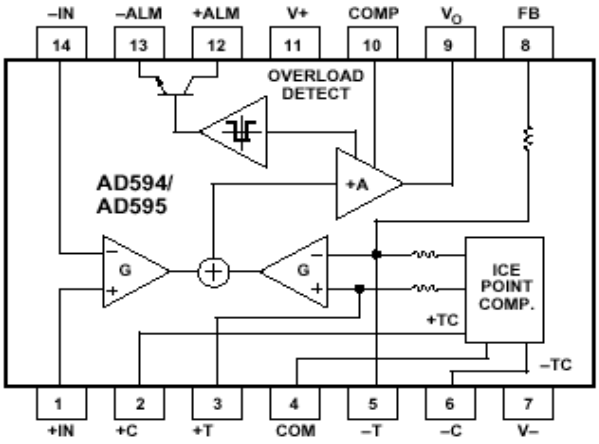


Figura 27. AD595 Diagrama de bloques

El AD595, es un amplificador de instrumentación y compensador de juntura fría en un chip (se explica en el apéndice F), combina su punto de referencia con un amplificador precalibrado para producir un voltaje proporcional a la temperatura de la termocupla de 10mV/°C. También incluye una alarma de falla de la termocupla, cuando alguna de los

conectores de la termocupla se abre, tiene una salida de alarma, que se puede manejar con voltaje TTL.

En operación normal, la salida del amplificador principal es conectado a la red de retroalimentación, Las señales de la termocupla están aplicadas a la etapa de entrada del AD595, de un amplificador de ganancia G, y después amplificados por el amplificador principal de ganancia A. La salida del amplificador principal es retroalimentado a una segunda etapa diferencial en la conexión invertida del amplificador. La señal de retroalimentación es amplificada por esta etapa y es también aplicada a la entrada del amplificador principal a través de un circuito de suma. Esta inversión hace que el amplificador cause la retroalimentación para reducir la señal de diferencia a un pequeño valor.

Los amplificadores diferenciales son hechos para acoplar y que tengan las mismas ganancias G. Como resultado, la señal de retroalimentación que deben ser aplicados al amplificador diferencial de la derecha va a acoplar precisamente a la entrada de la señal de la termocupla cuando la señal de la diferencia es reducida a cero.

El valor de la temperatura que obtiene el AD595 es proporcional a 10 mV/°C, para una termocupla tipo k.

3.3 Posibles problemas de ruido al sensor.

- Para la caracterización del sensor de temperatura con este circuito se necesita de un ambiente que mantenga una temperatura constante, la cual evite las corrientes de aire que puedan alterar la temperatura.
- Se necesita que la punta sea lo mas sensible posible para respuestas mas rápidas. Para esto se relacionan el área del la punta del sensor con el tiempo de respuesta. Es decir a mayor área menor sensibilidad y a menor área mayor sensibilidad y por lo tanto mayor velocidad de respuesta.

3.4 Diseño de la Interfase para el sensor de temperatura.

Para esta parte, se necesitaba conocer las conexiones y dimensiones del integrado, por lo cual se recurrió a su datasheet, la cual nos muestra todo lo requerido para poder realizar el circuito impreso para el circuito de acondicionamiento de señal y la termocupla.

En la figura 28, la cual es adaptada de [10], se muestra una posibilidad de hacer el circuito impreso, con una sola fuente de operación.

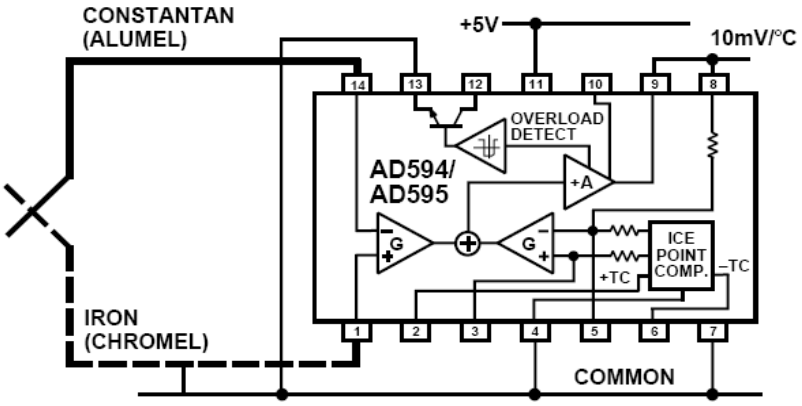


Figura 28. Conexión básica, con una sola fuente de operación.

Con el diagrama mostrado en la figura anterior, se hizo el circuito esquemático en el programa ORCAD 9.1, con el ORCAD Capture, para el circuito de acondicionamiento de señal AD595 y del sensor de temperatura.

El circuito esquemático realizado en ORCAD 9.1 se muestra en la figura 29.

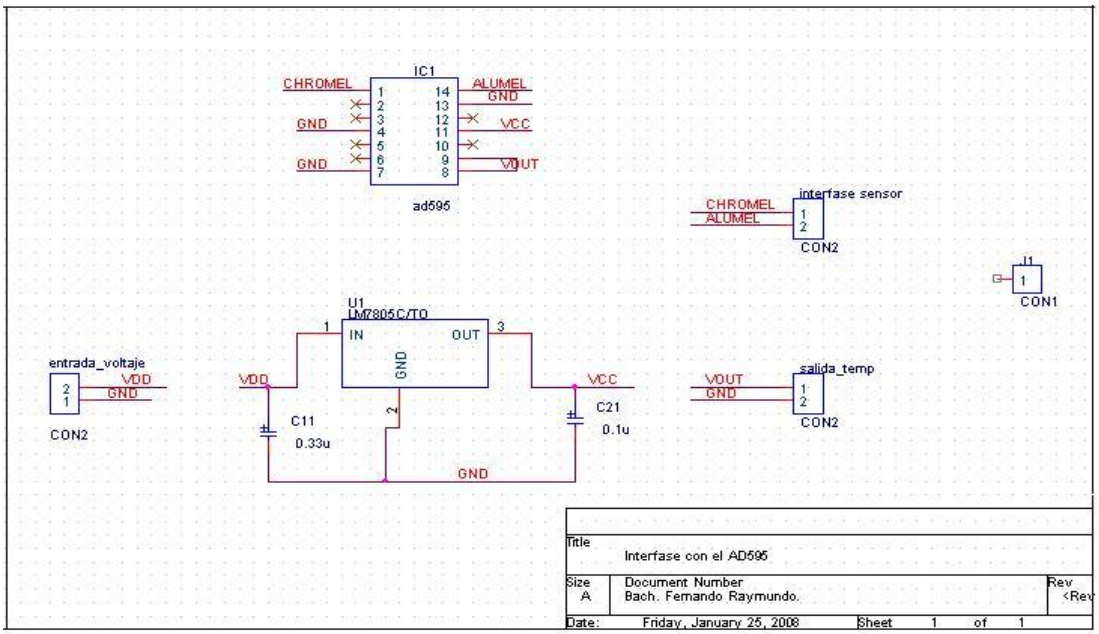


Figura 29. Circuito esquemático del AD595 y del sensor de temperatura.

De allí se paso a importar el archivo creado en ORCAD Capture a ORCAD Layout. Y se procedió a realizar el esquemático utilizando footprints de las librerías de ORCAD Layout.

El resultado del diseño del circuito se muestra en la figura 30.

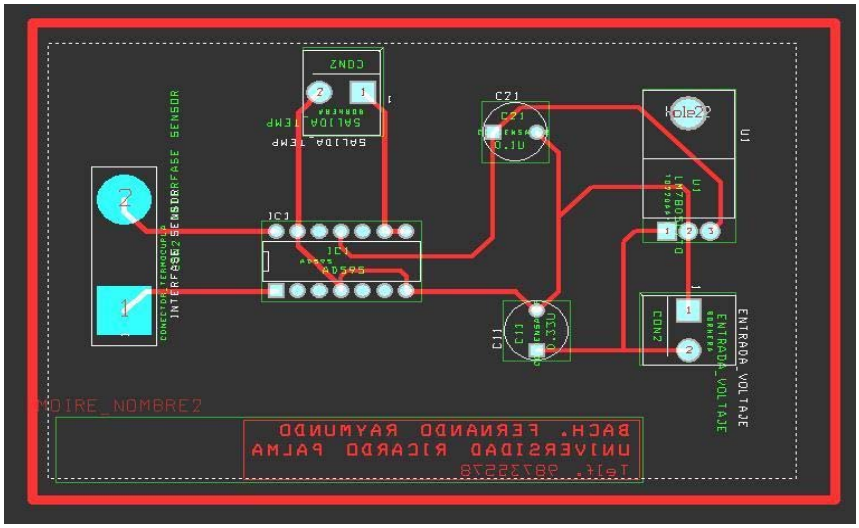


Figura 30. Circuito implementado en ORCAD Layout.

En la figura 31, se muestra el circuito impreso terminado e implementado con el AD595 para el acondicionador de temperatura.

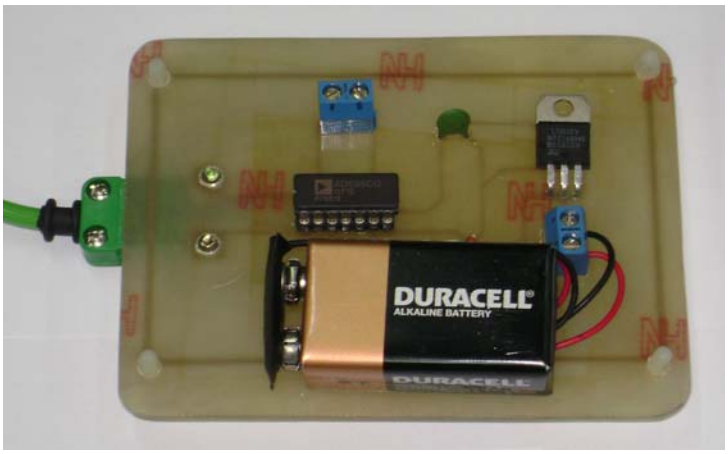


Figura 31. Circuito impreso de acondicionamiento de señal para la termocupla tipo K.

3.5 Procedimiento de la caracterización del sensor de temperatura.

En esta parte ya tenemos construido el circuito impreso del acondicionador de señal, también se dispone de una termocupla tipo K y de un patrón de temperatura, en ese caso un termómetro de mercurio, para realizar la caracterización. Todas estas herramientas son necesarias para poder hacer la caracterización del sensor de temperatura.

En la figura 32, se muestra el sensor de temperatura conectado a la tarjeta acondicionadora para el sensor.

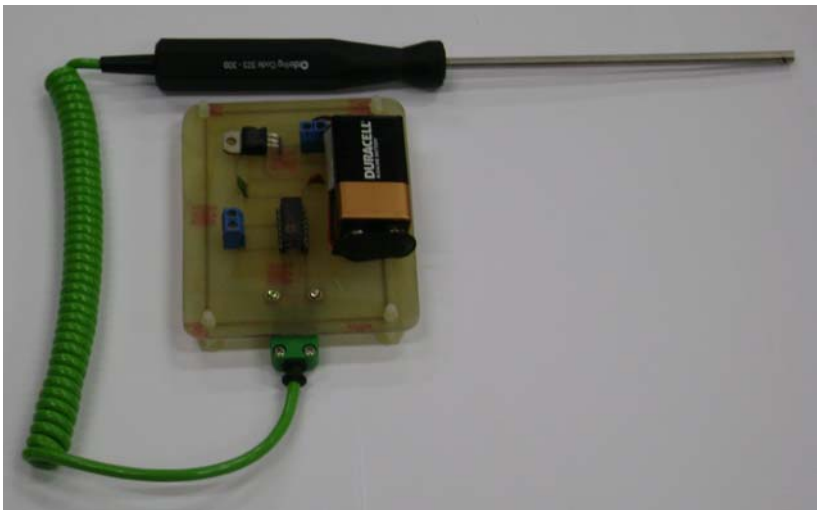


Figura 32. Circuito del acondicionador mas la termocupla.

Ahora tenemos que armar una estructura para poder hacer la caracterización del sistema.

Para ello usamos una estructura facilitada por el laboratorio de física, con ello podemos sostener primero el termómetro de mercurio, como se muestra en la figura 33.

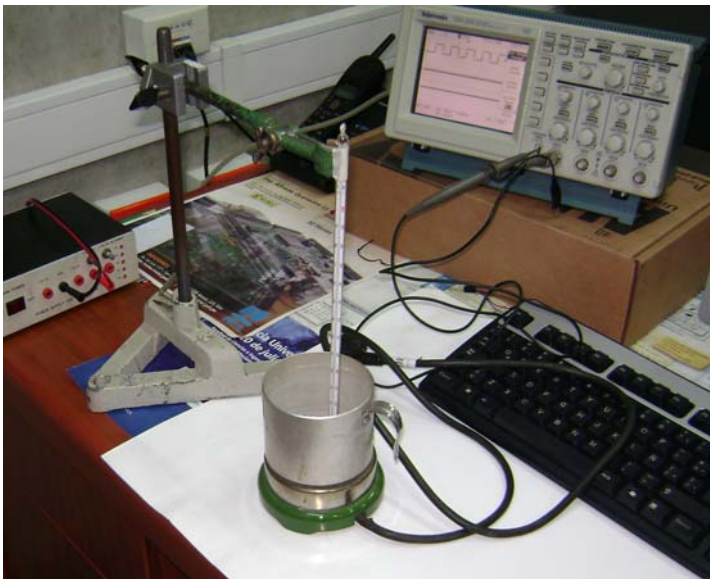


Figura 33. Termómetro de mercurio con la estructura.

Después se paso a colocar el sensor de temperatura con el acondicionador de señal, para poder empezar la caracterización del sensor, esto se muestra en la figura 34.

El multímetro que se muestra nos muestra un valor de 218mV, la cual es la respuesta del sensor de temperatura ya acondicionada del AD595, la cual es de 10mv/°C, y según el termómetro media muy cercano a 21°C, la cual esta en el rango de respuesta de la termocupla. Según su respuesta a 21°C nos mostraba 218mV, la cual nos daba una certidumbre del valor real.

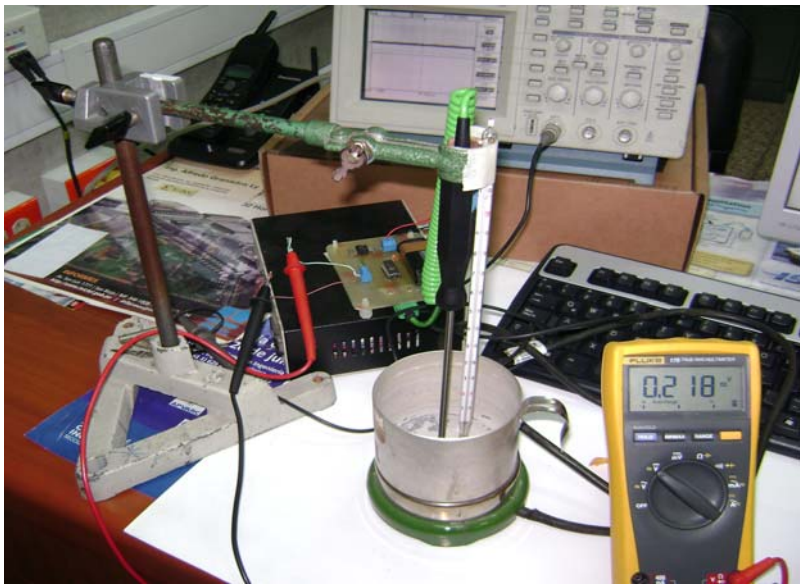


Figura 34. Termocupla y Termómetro.

En esta parte se conecta las salidas del circuito de acondicionamiento de señal de la termocupla, al modulo DAQ USB-6008, la cual cuenta con estas características: 8 entradas analógicas (12 bit, de hasta 10 KS/s), 2 salidas analógicas (12bit, 150 S/s), 12 entradas y salidas digital y un contador de 32 bit. La conexión se muestra en la figura 35.

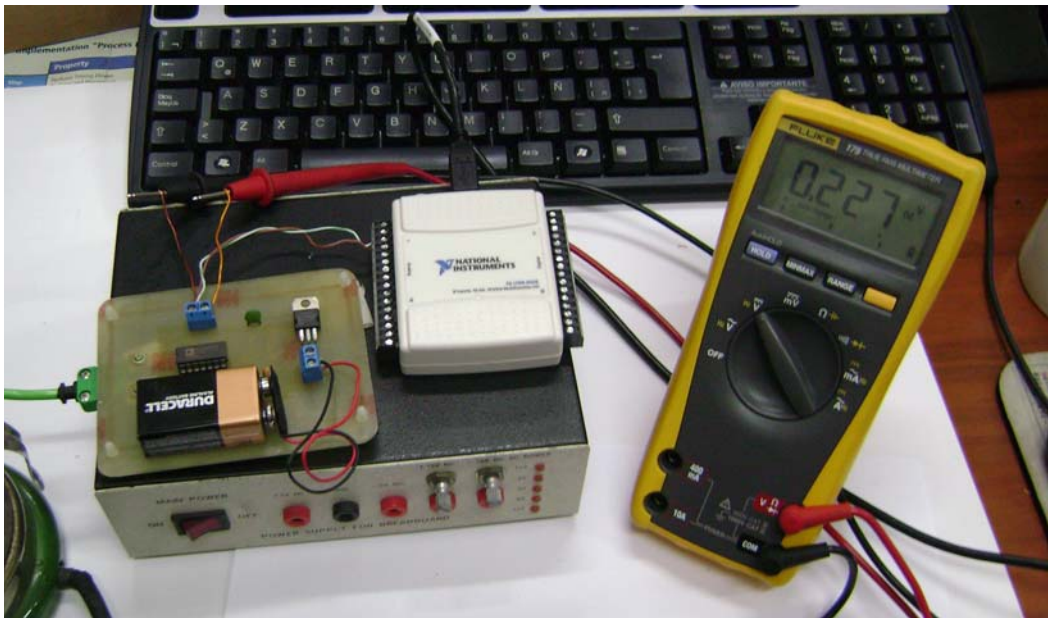


Figura 35. Conexión entre DAQ y Acondicionador de señal.

Se uso el software de national instruments, labview, para poder visualizar los datos adquiridos, los valores adquiridos, eran contaminados por ruido, esto se muestra en la pantalla principal que se muestra en la figura 36.

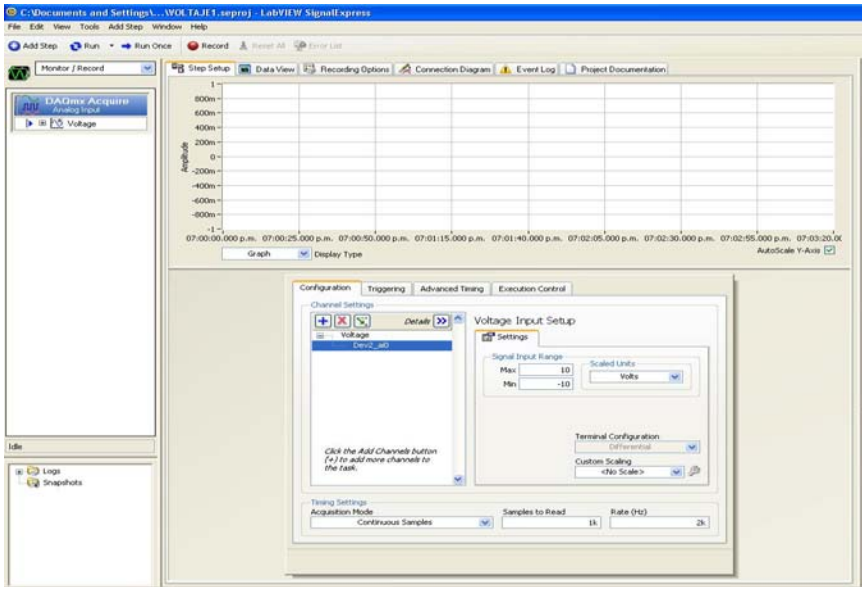


Figura 36. Pantalla principal del software de adquisición de datos.

Haciendo la estas operaciones, hemos podido caracterizar a una temperatura, para probar el sistema funcionamiento y para poder usar el filtro de kalman.

CAPITULO 4. FILTRO DE KALMAN

Debido a las caracterizaciones que se realizaron para la carga útil, se vio como se infiltraba el ruido en las mediciones, entonces como un medio de solución para poder estimar el valor real del sensor de temperatura, se va a usar el estimador óptimo de kalman para poder estimar la señal del sensor de temperatura de la señal con ruido.

4.1. Introducción y Características

Al momento de realizar un análisis o el diseño de un controlador, el ingeniero tiene a su disposición conocimientos previos derivados de los SISTEMAS DETERMINISTICOS, y de TEORIAS DE CONTROL. Con ello no bastaría para poder hacer un modelo del sistema, por eso es necesario ver el problema de otra manera, a través de procesos estocásticos.

Se van a mostrar las falencias de los sistemas deterministicos, por ejemplo, dado un sistema físico, puede ser un avión, un proceso químico o la economía nacional, un ingeniero primero desarrolla un modelo matemático que representa adecuadamente muchos aspectos del comportamiento del sistema. A través de la física, con las leyes fundamentales, y el testeo empírico, trata de establecer las interrelaciones entre variables de interés, entradas a el sistema y salidas del sistema.

Con ese modelo matemático, y las herramientas provistas por el sistema y teorías de control, el ingeniero es capaz de investigar la estructura del sistema y los modos de respuesta, si es deseado, puede diseñar compensadores que alteren las características y controladores que proveen apropiadas entradas para generar respuestas deseadas al sistema.

Para observar el comportamiento del sistema actual, los sensores son construidos para que den señales de salidas de datos proporcionales a algunas variables de interés. Estas salidas de señal y las señales de entradas conocidas al sistema son la única información que es

directamente discernible acerca del comportamiento del sistema. Más aún si un controlador realimentado es diseñado, la salida del sensor es la única entrada al controlador.

A continuación se presentan los 3 casos en donde los sistemas deterministicos y las teorías de control no proveen un significado suficiente de realización del diseño.

Primer caso, *Ningún modelo matemático es perfecto,* cualquier modelo representa, solo esas características del interés para el propósito del ingeniero. El objetivo del modelo es de representar el modo crítico o dominante de la respuesta del sistema, entonces muchos efectos son a sabiendas dejados de modelar. De hecho, los modelos usados para generar procesadores de datos online o controladores deben ser reducidos a solo a modelos básicos en orden de generar un algoritmo computacional factible.

Incluso en los efectos los cuales son modelados son necesariamente *aproximados* por un modelo matemático. Las leyes físicas Newtonianas, son aproximaciones adecuadas debido a que nosotros no estamos acostumbrados a interactuar a la velocidad de la luz. Es a menudo el caso que esas leyes proveen estructuras adecuadas, pero varios parámetros dentro de la estructura que nos son determinados absolutamente. Además, hay muchas fuentes de incertidumbre en cualquier modelo de un sistema.

Segundo caso, *son los disturbios*, puesto que los sistemas dinámicos son *manejados no solo por nuestras entradas de control, sino por disturbancias que no se pueden controlar o modelar deterministicamente*. Si un piloto trata de comandar una orientación angular en su avión, la respuesta actual va a diferir de la expectación debido al golpe del viento, la imprecisión de control del actuador de superficie y aun su inestabilidad para generar exactamente la respuesta deseable de sus propios brazos y manos en la palanca de control.

Ultimo caso, *los sensores de los modelos deterministicos no proveen datos perfectos y completos acerca de un sistema.* Primero, ellos generalmente no proveen toda la información que quisiéramos saber, algunos pueden ser limitados y tener zonas muertas de sensado.

En otras situaciones, un número de sensores diferentes dan señales funcionalmente relacionadas, y se puede después ayudar a generar una mejor estimación de las variables de

interés basadas en particularmente en datos redundantes. Los sensores no proveen exactamente lecturas de cantidades deseables, pero si introducen sus propias distorsiones y dinamismos de sistemas. Más aún, esos componentes siempre se corrompen con el ruido.

Como se pudo analizar de los párrafos anteriores, es inadecuado asumir el conocimiento perfecto de todas las cantidades necesarias para describir un completo sistema y asumir el perfecto control sobre el sistema.

4.2 Desarrollo del filtro Kalman.

En los siguientes párrafos se va describir el estimador lineal óptimo de Kalman, llamado también el filtro de kalman.

4.2.1 Filtro de Kalman.

El filtro de kalman es un *algoritmo de procesamiento de datos recursivo óptimo.* Para entender esto, vamos a definir óptimo, dependiendo sobre el criterio escogido para evaluar el funcionamiento del algoritmo.

Un aspecto de ser *ÓPTIMO* es que el filtro de kalman incorpora toda la información que se le pueda ser proveer. Procesa todas las medidas disponibles, sin mirar su precisión, para estimar el valor actual de las variables de interés, con el uso del conocimiento del sistema y las dinámicas de los sensores, la descripción estadística de los ruidos del sistema, errores de medida, los modelos dinámicos inciertos, y cualquier información disponible acerca de las condiciones de las variables de interés. Por ejemplo, para determinar la velocidad de un avión, se puede usar un radar doppler, o las indicaciones de un sistema de navegación inercial, o el pitot y la presión estática y la información del viento en los sistemas de datos.

Más bien que ignorar cualquiera de esas salidas, el filtro de kalman puede ser construido para combinar todo los datos y el conocimiento de las dinámicas de los varios sistemas para generar una buena estimación general de la velocidad.

La palabra *RECURSIVO* significa, a diferencia de ciertas conceptos de procesamiento de datos, que el filtro de kalman no requiere toda los datos previos requeridos se mantengan

almacenados y reprocesados cada vez que una nueva medida sea tomada. Este concepto es de vital importancia para la practicidad de la implementación del filtro.

El filtro es actualmente un *ALGORITMO DE PROCESAMIENTO DE DATOS*, a pesar de la típica connotación del filtro como caja negra conteniendo redes eléctricas, el hecho es que muchas aplicaciones, el filtro es solo un programa de computadora en un procesador central. Como eso, esto inherentemente incorpora medidas discretas en el tiempo.

En la figura 37, la cual es adaptada de [3], nos muestra una situación en la cual el filtro de kalman puede ser usado de una manera muy útil. Un sistema cualquiera es manejado por algunos controles conocidos, y componentes de medida que proveen el valor de algunas cantidades pertinentes. El conocimiento de las entradas y salidas del sistema es todo lo que es explicito disponible de el sistema físico para los propósitos de estimación.

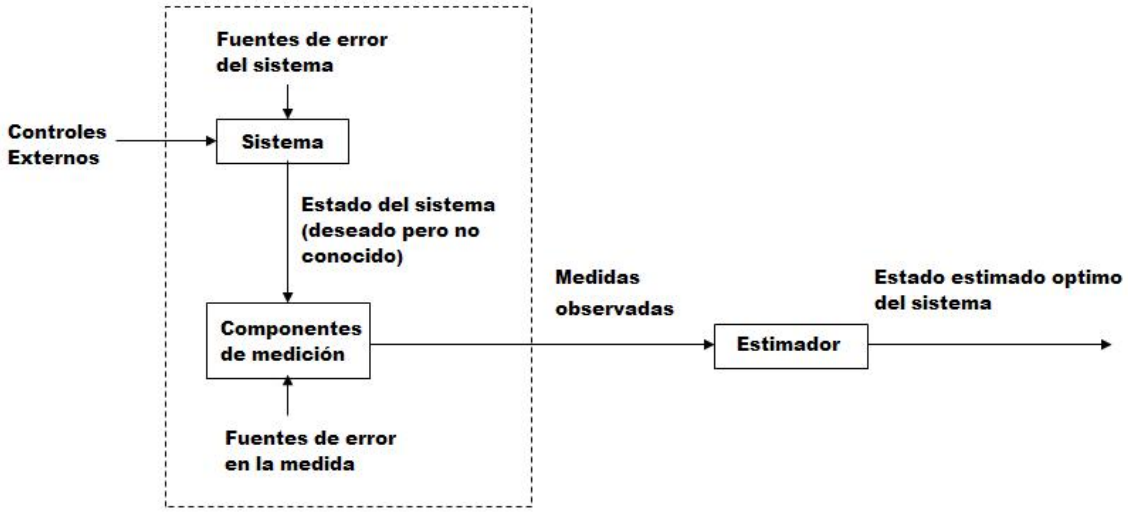


Figura 37. Aplicación típica del filtro de Kalman.

A menudo las variables de interés de alguna cantidad finita describen el estado del sistema, algunas no pueden ser medidas directamente, y otras que se pueden inferir de los datos disponibles.

Más aun, cualquier medida va a ser corrompida de alguna manera por el ruido, como las inexactitudes del componente, por lo que la extracción de la información valuable de una señal ruidosa es muy necesaria.

Hay muchos sensores, que tienen sus propias dinámicas particulares y errores característicos, que proveen alguna información acerca una variable particular, y podría ser deseable de combinar esas salidas en una manera optima y sistemática.

El filtro de kalman combina todas las medidas de los datos, además con un previo conocimiento acerca de los componentes del sistema, para producir un estimado de las variables deseables en una manera en la cual el error es minimizado estadísticamente.

Conceptualmente, cualquier tipo de filtro trata de obtener una estimación óptima de las cantidades deseadas de un ambiente ruidoso, *OPTIMO*, significa que minimiza el error en algún aspecto. Hay muchas maneras para llegar a este objetivo.

Después, nosotros queremos propagar la *DENSIDAD DE PROBABILIDAD*, de las cantidades deseadas, condicionado en el conocimiento del dato actual proveniente de los componentes de medida.

Para entender este concepto, veamos la figura 38, la cual es adaptada de [3], nos muestra una representación de una densidad de probabilidad condicional del valor de una cantidad escalar x en el instante i, x(i), condicionado en el conocimiento del vector z(1)en el instante 1 tomado en el valor z1 y similarmente para los instantes 2 hasta i, graficado como función de los posibles valores x(i).

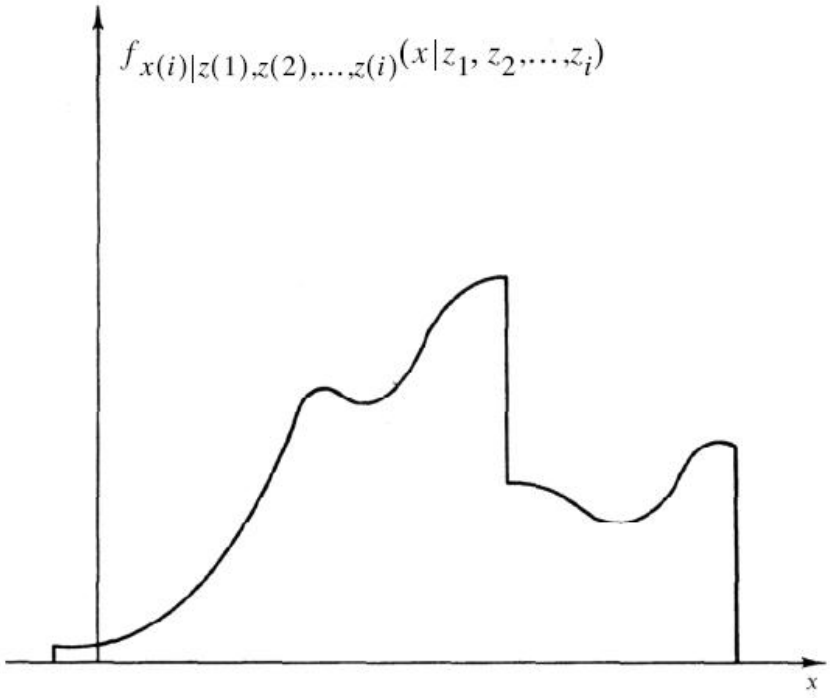


Figura 38. Densidad de probabilidad.

Por ejemplo, sea x(i) la posición de un vehículo unidimensional , y z(i) sea 2 vectores bidimensionales describiendo las medidas de la posición en el tiempo j por 2 radares diferentes, Esa densidad de probabilidad contiene toda la información acerca de x(i), e indica, que por los valores dados de las medidas tomadas hasta el instante de tiempo i, la probabilidad que podría tener x(i) asumiendo cualquier valor particular o rango de valores.

Es llamado densidad de probabilidad condicional porque su forma y locación en el eje x es dependiente de los valores tomados. Su forma transmite la cantidad de certeza que se tiene en el conocimiento del valor de x. Si la densidad dibuja un pico estrecho, después el mayor peso de la probabilidad está concentrado en el pico estrecho de los valores de x. Por otro lado, si el dibujo tiene una figura gradual, el peso de la probabilidad es esparcida sobre un ensanchado rango de x, indicando que es menos seguro de ese valor.

Una vez que la densidad condicional de probabilidad se propaga, es definido el estado óptimo y las opciones pueden incluir los siguientes parámetros:

1. La media o el centro de masa de la probabilidad.

2. La moda, el valor de que tenga la mayor probabilidad, ubicándose en el pico de la densidad.
3. La mediana, el valor de x que tiene la mitad de peso de la probabilidad ligado a la derecha e izquierda de la densidad condicional de probabilidad.

El filtro de kalman efectúa la propagación de la densidad de probabilidad condicional para problemas en la cual, el sistema puede ser descrito a través de un modelo lineal y en el cual el sistema y los ruidos de medida son gausianos y blancos. Bajos estas condicionales, la media, la moda, la mediana son virtualmente una opción razonable para un estimado optimo del valor de x.

4.2.2 Presunciones básicas.

En este punto es útil mirar a las 3 presunciones básicas en la formulación del filtro de kalman. En una primera inspección, ellos aparecen ser restrictivos e irreales. Para despejar cualquier malentendido, esta sección va discutir brevemente las implicaciones físicas de esas presunciones.

4.2.2.1 Sistema Lineal.

Un modelo de sistema lineal es justificable por muchas razones:

A menudo un modelo lineal es adecuado para propósitos específicos, y cuando las no linealidades existen, la típica ingeniería tiende a linealizar en un punto nominal o trayectoria, obteniendo un modelo de perturbación o un modelo de error.

Los sistemas lineales son deseables porque son más fáciles de manipular con herramientas de ingeniería, y la teoría de sistemas lineales (ecuaciones diferenciales) es mucha más completa y practica que la no lineal. El hecho es que hay muchas aplicaciones que usan el filtro de kalman extendido a algunas aplicaciones no lineales en el desarrollo de los filtros no lineales, pero esos son considerados solo si los modelos lineales son probados inadecuados.

4.2.2.2 Blancura.

Implica que el valor del ruido no esta correlacionado en el tiempo, es decir no se sabe que valor va a tener, este desconocimiento hace que no sea bueno en la predicción del valor en cualquier tiempo.

La blancura, también implica que el ruido tenga igual potencia en todas las frecuencias. Desde que esto resulta en una potencia infinita de ruido, el ruido blanco obviamente no puede existir. Entonces se puede considerar ese concepto de una manera singular.

Primero, cualquier sistema de interés tiene una banda de paso, un rango de frecuencias en la cual puede responder, y encima de estas frecuencias, las entradas no tienen efecto en las salidas, o el sistema las atenúa severamente.

En la figura 39, la cual es adaptada de [3], se muestra una gráfica típica del sistema de paso de banda con los ejes coordenados, de esta manera: en las ordenadas se encuentra la "Densidad de potencia espectral" (interpretada como la cantidad de potencia en una cierta frecuencia) y en las abscisas se encuentra la frecuencia.

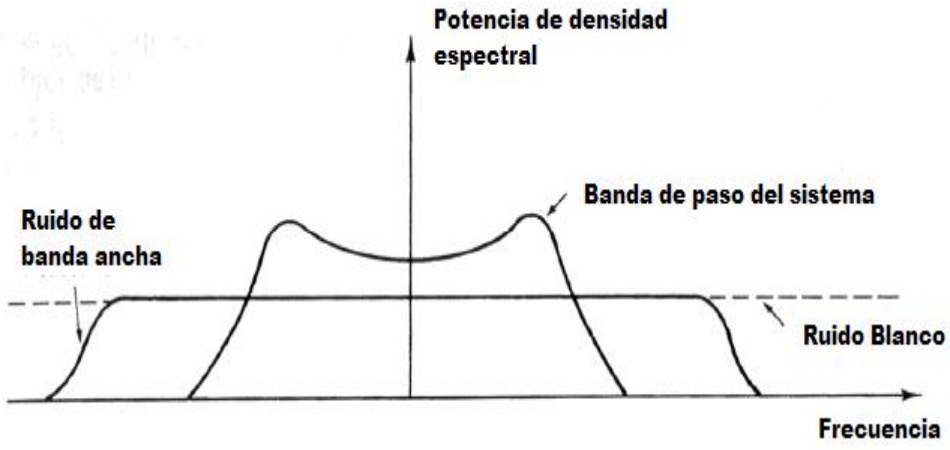


Figura 39. Ancho de banda de las densidades espectrales.

Típicamente un sistema va a ser manejado por el ancho de banda del ruido, teniendo en cuenta la potencia en las frecuencias encima del paso de banda de sistema, como se muestra en la figura 39.

En el mismo dibujo, un ruido blanco puede extender su potencia constante en todas las frecuencias. Ahora dentro de la banda de paso del sistema de interés, el ruido blanco ficticio se coloca y tiene mayor ancho de banda que del sistema, entonces el modelo del ruido blanco es “adecuado” y usado. Esto nos lleva a que las matemáticas involucradas en el filtro son simplificadas (hecho que nos lo hace mas tratable) reemplazando el real ancho de banda del ruido con un ruido blanco ficticio, el cual desde el punto de vista del sistema, es idéntico.

Donde la *blancura se refiere al tiempo o la frecuencia* de un ruido, *la guasianidad tiene que ver con su amplitud*. Además, en cualquier punto en el tiempo, la densidad de probabilidad de una amplitud del ruido gaussiano tiene una forma de campana. Esto puede justificarse físicamente por el hecho que el sistema o ruido medido es típicamente causado por un número de pequeñas fuentes. Esto se definió anteriormente en el teorema del límite central.

4.2.2.3 Densidad Gaussiana.

Hay también una práctica justificación para las densidades gaussianas, similar a la blancura, esto hace que las matemáticas sean más tratables, pero más que eso, típicamente un ingeniero va a saber en mayor grado, las estadísticas de primer y segundo orden (media y la varianza o desviación estándar) del ruido del proceso. En la ausencia de cualquiera de las estadísticas de mayor orden, es mejor asumir la densidad gaussiana. Las estadísticas del primer y segundo orden determinan completamente la densidad gaussiana, no como otras densidades que requieren estadísticas de mayor orden para especificar su forma enteramente.

Las presunciones particulares, son hechas por los objetivos que se quieren alcanzar, y por ese motivo esencial, los modelos son desarrollados. Si nuestro objetivo es simplemente construir modelos descriptivos, no podemos confinar nuestra atención a sistemas lineales manejados por ruido blanco. Más bien, tendríamos que buscar el modelo, de alguna manera, que mejor encaje a los datos generados por el mundo real. Por ello se construyen estimadores y controladores basados en los modelos de sistemas que manejan esas asumpciones.

Afortunadamente, la clase de modelos que llevan a matemáticas tratables también proveen representación adecuada para muchas aplicaciones de interés. Después, la estructura del modelo se va extender a algún lugar para alargar el rango de la aplicabilidad, pero el requerimiento de la utilidad del modelo para el diseño de un estimador o de un controlador va a ser una influencia dominante en la manera en la cual las extensiones son hechas.

4.3 Ecuaciones del proceso.

El filtro de Kalman tiene como prioridad estimar el estado x_k de un proceso en tiempo discreto, el cual es dominado por una ecuación de diferencias Lineal estocástica de la siguiente forma:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \text{ . .....(4.1)}$$

con una medida Z, que es

$$Z_k = Hx_k + v_k \text{ . ..... (4.2)}$$

Las variables aleatorias w_k y v_k representan el error del proceso y de la medida respectivamente. Se asume que no están correlacionadas con el tiempo y son independientes entre ellas, que tienen una distribución de probabilidad normal:

$$p(w) \cong N(0, Q) \text{ ..... (4.3)}$$

$$p(v) \cong N(0, R) \text{ ..... (4.4)}$$

Esto quiere decir que tienen una distribución normal, con media 0 y con una desviación estándar de Q y R, respectivamente.

En la práctica las matrices de covarianza de la perturbación del proceso, Q ,y de la perturbación de la medida, R, podrían cambiar en el tiempo, por simplicidad en general se asumen que son constantes.

La matriz A tiene una dimensión $n \times n$ y relaciona el estado previo k-1 con el estado actual k. La matriz B, se relaciona con la entrada de control y la matriz H de dimensión $m \times n$ relaciona el estado con la medición Z_k.

Estas matrices pueden cambiar en el tiempo, pero en general se asumen como constantes.

4.3.1 Origen computacional del filtro.

Se define el estado estimado, *a priori* en K, con una raya encima, el conocimiento del proceso antes de k, y definimos nuestro estado *a posteriori* en k dada la medición. Entonces podemos definir los errores de estimación a priori y a posteriori.

$$e_k^- \equiv x_k - \hat{x}_k^- \quad \text{.....(4.5)}$$

$$e_k \equiv x_k - \hat{x}_k \quad \text{..... (4.6)}$$

La covarianza del error del estado estimado a priori, es entonces:

$$P_k^- = E[e_k^- e_k^{-T}] \quad \text{..... (4.7)}$$

Y La covarianza del error del estado estimado *a posteriori,* es entonces:

$$P_k = E[e_k e_k^T] \quad \text{..... (4.8)}$$

Por medio de las ecuaciones del filtro Kalman, empezamos con el objetivo de encontrar una ecuación que calcula el *estado estimado a posteriori* como una combinación lineal de una *estimación a priori* y de manera ponderada de diferencias entre la medición actual y la predicción de la medición.

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad \text{..... (4.9)}$$

La diferencia en la ecuación (4.9) es llamada innovación en la medición, o el residuo, que refleja la discrepancia entre la medición y la medición actual. Un resido de cero significa que los dos están completamente cercanos.

La matriz K en la ecuación (4.9) es elegida para ser la ganancia o la mezcla que minimiza el error *a posteriori* de la covarianza (4.8). Esta minimización se puede lograr incorporando en la ecuación (4.9) en e_k, de la ecuación (4.6) y después lo que se incorporo en la definición anterior, se incluye en la ecuación (4.8), para después hacer un procedimiento que se va a describir en el apéndice H, para entonces resolviendo para K, se tiene la siguiente expresión:

$$K_k = P^-_k H^T (HP^-_k H^T + R)^{-1}$$
$$= \frac{P^-_k H^T}{(HP^-_k H^T + R)} \qquad \text{….. (4.10)}$$

En cuanto a la ecuación (4.10) vemos que a medida que la covarianza del error de medición se aproxima a cero, R, la ganancia K tiende a la inversa de H.

$$\lim_{R_k^- \to 0} K_k = H^{-1}$$

Por otra parte, como la estimación de la covarianza del error *a priori* , P_k^-, se aproxima a cero, la ganancia K tiende a cero.

$$\lim_{P_k^- \to 0} K_k = 0$$

Un significado de K, es entender que la covarianza del error de medición se aproxima a cero, entonces la medición real Z_k es de más "confianza", mientras que la medida predecida $H\hat{x}_k^-$ es menos confiable. Por otra parte, como la estimación de la covarianza

del error *a priori,* P_k^- , se aproxima a cero la medición real Z_k es cada vez menos confiable, mientras que la medición predecida $H\hat{x}_k^-$ es mas confiable.

4.3.2 Los orígenes probabilísticos del filtro.

La justificación de la ecuación (4.9) se basa en la probabilidad de la estimación *a priori* condicionado en todas las mediciones anteriores.

$$E[x_k] = \hat{x}_k \quad \text{….. (4.11)}$$

$$E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] = P_k \quad \text{……. (4.12)}$$

El estado estimado *a posteriori* en la ecuación (4.9) corresponde a la media (el primer momento) y es distribuido normalmente si las condiciones en las ecuaciones (4.3) y (4.4) se cumplen. La estimación de la covarianza del error a posteriori (4.8) refleja la varianza del estado de distribución (el segundo no momento central). En otras palabras:

$$\begin{aligned} p(x_k \mid z_k) &\approx N(E[x_k], E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]) \\ &= N(\hat{x}_k, P_k) \end{aligned} \quad \text{…. (4.13)}$$

Vemos que al final se convierte en una distribución normal con media $\hat{x}_k$ y desviación estándar P_k .

4.4 El algoritmo discreto del filtro de kalman.

El filtro de Kalman estima un proceso utilizando una forma de control de retroalimentación, en otras palabras, estima el estado del proceso en algún momento en el tiempo y entonces obtiene la retroalimentación por medio de los datos observados.

Las ecuaciones que se utilizan para derivar el filtro de Kalman se pueden dividir en dos grupos: las ecuaciones que actualizan el tiempo (actualizan el estado) y las que actualizan los datos observados.

Las *ecuaciones de actualización de tiempo* son responsables de la estimación del estado actual y del error de covarianza para obtener la estimación para el próximo estado.

Las *ecuaciones de actualización de medida* son responsables de la retroalimentación, es decir, incorporan nueva medida dentro de la estimación *a priori* para obtener una estimación mejorada del estado.

Las ecuaciones de actualización del tiempo pueden también ser pensadas como ecuaciones predictoras, mientras que las ecuaciones de actualización de medidas pueden considerarse como ecuaciones de corrección.

Efectivamente, el algoritmo de estimación final puede definirse como un algoritmo de predicción-corrección para resolver numerosos problemas.

El estimador óptimo o filtro de Kalman trabaja como proyector y corrector al pronosticar el nuevo estado y su incertidumbre además de corregir la proyección con la nueva medida. Este ciclo se muestra en la figura 40, la cual es adaptada de [6].

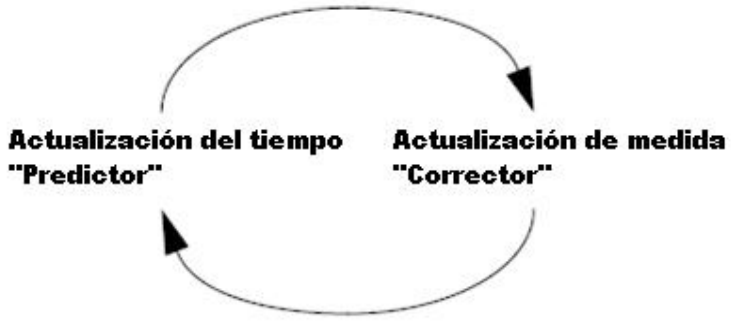


Figura 40. Ciclo de Kalman.

El filtro de kalman, usa la actualización del tiempo como predictor, la cual genera una predicción del estado para el siguiente dato en el tiempo tomando en cuenta la información disponible hasta ese momento, también usa la actualización de medida como corrector, la cual mejora la predicción del estado, de tal forma que el error es minimizado.

A continuación se muestra las ecuaciones que gobiernan en la parte predictora de estado:

$$\hat{x}^{-}_{k} = A\hat{x}_{k-1} + Bu_{k-1} + w_{k-1} \text{ . ..... (4.14)}$$

$$P^{-}_{k} = AP_{k-1}A^{T} + Q \text{ . ..... (4.15)}$$

Por medio de estas ecuaciones podemos proyectar el estado hacia delante de k-1 a k.

$$K_{k} = P^{-}_{k} H^{T} (HP^{-}_{k} H^{T} + R)^{-1} \text{ . .....(4.16)}$$

$$\hat{x}_{k} = \hat{x}^{-}_{k} + K_{k} (z_{k} - H\hat{x}^{-}_{k}) \text{ . ..... (4.17)}$$

$$P_{k} = (I - K_{k}H)P^{-}_{k} \text{ . ..... (4.18)}$$

La primera tarea durante la actualización de la medición es calcular la ganancia de Kalman, K. El siguiente paso es medir el proceso para obtener y después generar una estimación del estado *a posteriori* mediante la incorporación de la medición en la ecuación (4.17). El paso final es obtener la estimación del error de la covarianza a posteriori vía la ecuación (4.18).

Después de cada actualización en la medición y el tiempo, el proceso se repite con las estimaciones anteriores a posteriori utilizadas para proyectar o predecir las nuevas estimaciones *a priori.*

Este carácter recursivo es una de las características muy significativas del filtro Kalman lo cual lo hace practico en las implementaciones que los filtros de Wiener, que está diseñado para operar en todos los datos directamente de cada estimación. El filtro de Kalman usa su recursividad para la estimación actual en base a todas las mediciones anteriores. En la figura 41 se ofrece un panorama completo del funcionamiento del filtro, esta figura es adaptada de [6].

4.4.1 Parámetros del filtro y afinación.

En la aplicación real del filtro, la covarianza del error de medición, R, suele medirse antes de la operación del filtro. La medición de la covarianza del error de medición, R, es en general práctica, porque es posible medir el proceso de todos modos (mientras que el filtro entra en funcionamiento), es por ello recomendable tomar algunas muestras off-line de las mediciones con el fin de determinar la varianza de la medición de ruido.

La determinación de la covarianza del ruido del proceso, Q, es generalmente más difícil de medir ya que normalmente no se tiene la capacidad de observar directamente el proceso que estamos estimando. A veces, con modelos simples del proceso, se puede producir resultados aceptables si se "inyecta" bastante incertidumbre en el proceso a través de la selección de Q.

En cualquier caso, si tenemos o no una base racional para la elección de los parámetros, muchas veces se colocan parámetros superior el rendimiento del filtro (estadísticamente hablando) para después obtener por medio de la afinación de los parámetros del filtro el Q y R. El afinamiento es usualmente en modo off-line, a menudo con la ayuda de otro filtro de Kalman en un proceso conocido como sistema de identificación.

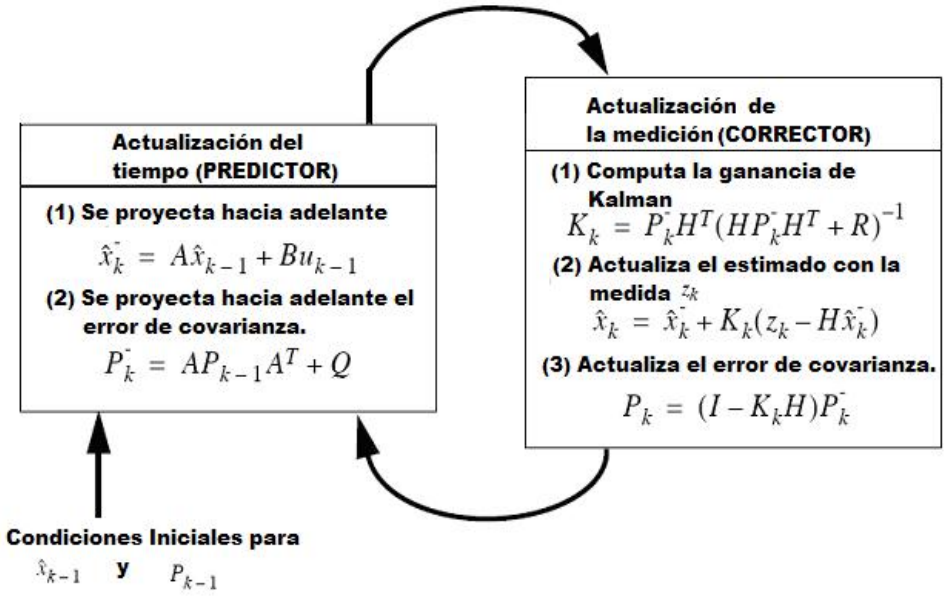


Figura 41. Ecuaciones del filtro de kalman.

Bajo condiciones en donde Q y R son constantes, ambas estimaciones de la covarianza del error de P_k y la ganancia de Kalman K se estabilizarán rápidamente y luego se mantendrán constantes. Si este es el caso, estos parámetros pueden ser pre-calculado corriendo el filtro off-line.

CAPITULO 5. DISPOSITIVOS DE LOGICA RECONFIGURABLE

Los FPGA fueron introducidos por Xilinx en la mitad de 1980. Ellos difieren de los CPLDs en la arquitectura, por su tecnología de almacenamiento, número de características internas, y el costo, y tienen por objeto la aplicación de alto rendimiento, de circuitos de gran tamaño. Estos tienen bloques de lógica interna que se pueden interconectar mediante software de desarrollo.

5.1. Características principales.

Algunas ventajas de los FPGAs son:

- Reprogramables.
- Flexibles.
- Costes de desarrollo más baratos.
- Tiempo de desarrollo es menor.

Es cierto que en comparación con los ASICs, en forma general, consumen mayor potencia y en algunos casos son más lentos.

Los FPGAs son en su mayoría volátiles, ya que hacen uso de la SRAM para almacenar las conexiones, por lo cual una configuración ROM es necesaria para cargar las interconexiones en el encendido. Hay por ejemplo, algunos FPGAs no volátiles que usan antifusibles (anti-fuses).

La arquitectura básica de una FPGA se ilustra en la figura 42, la cual es adaptada de [8], y consiste en matrices de CLBs (Configurable Logic Blocks – Bloques lógicos reconfigurables), interconectadas por un conjunto de matrices switches o interruptores.

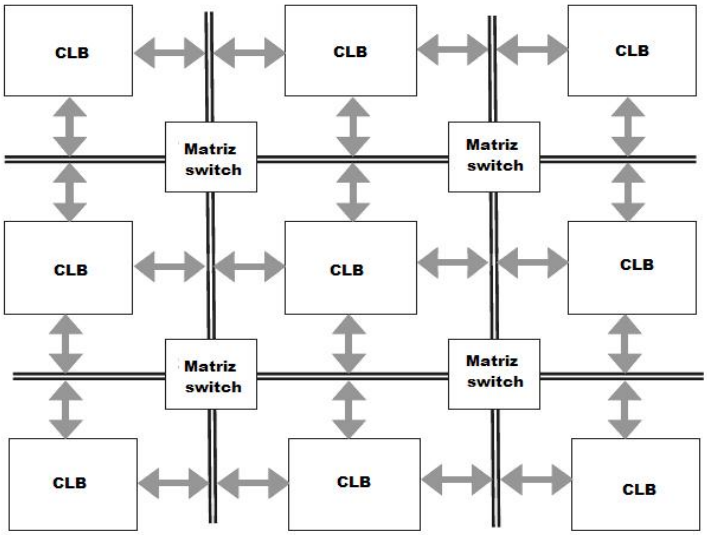


Figura 42. Arquitectura básica de un FPGA.

La arquitectura interna de un CLB es diferente a la de un PLD. En primer lugar, en lugar de implementar aplicar las expresiones de suma de productos con compuertas AND seguidos de las compuertas OR como en las GALs, su implementación es normalmente sobre la base de una LUT (lookup table).

Por otra parte, en una FPGA el número de flip-flops es mucho más abundante que en un CPLD, lo que permite la construcción de circuitos secuenciales más sofisticados.

Además de estar relacionado con un estándar como el JTAG para la interfaz, y otras características adicionales que se incluyen en los chips de FPGA, al igual que la memoria SRAM, multiplicación de reloj (PLL o DLL), interfaz PCI, etc. Algunos FPGAs también incluyen bloques dedicados, como multiplicadores y microprocesadores.

Otra diferencia fundamental entre un FPGA y CPLD es referente al almacenamiento de las interconexiones. Si bien los CPLDs no son volátiles (es decir, hace uso de antifuse, EEPROM, Flash, etc), la mayoría de FPGAs usan SRAM, y, por tanto, es volátil.

Esta característica hace que el FPGA ahorre espacio y reduce el costo del chip, porque los FPGAs tienen un gran número de interconexiones programables, pero requiere una ROM externa.

Hay, sin embargo, FPGAs no volátiles (con antifuse), lo que podría ser ventajoso cuando la reprogramación no es necesaria.

Los FPGAs pueden ser muy sofisticados, hay Chips fabricados con tecnología CMOS de 0,09 mm, con nueve capas de cobre y más de 1000 pines de entrada y salida, los cuales están disponibles actualmente.

Algunos ejemplos de encapsulados FPGA se ilustran en la figura 43, la cual esta adaptada de [8], que muestra uno de los más pequeños encapsulados como el de la izquierda (64 pines), una de tamaño mediano (324 pines), y otra de encapsulado grande (1152 pines).

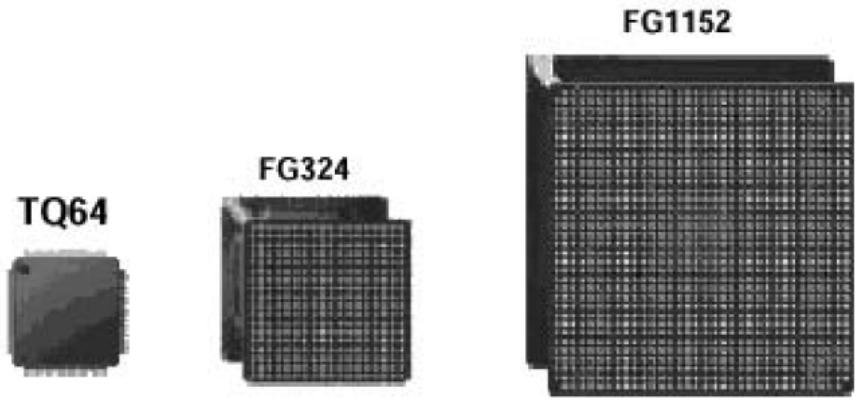


Figura 43. Encapsulados de FPGAs.

Hay muchas empresas de fabricación de FPGAs, como Xilinx, ACTEL, Altera, QuickLogic, Atmel, etc. Un ejemplo de las gamas de FPGA se muestra en la siguiente tabla 1, la cual esta adaptada de [8].

Familia	Virtex II Pro	Virtex II	Virtex E	Virtex	Spartan 3	Spartan IIE	Spartan II
Bloques Lógicos -CLBs	352–11,024	64–11,648	384–16,224	384–6,144	192–8,320	384–3,456	96–1,176
Celdas lógicas	3,168–125,136	576–104,882	1,728–73,008	1,728–27,648	1,728–74,880	1,728–15,552	432–5,292
Compuerta del sistema		40 k–8 M	72 k–4 M	58 k–1.1 M	50 k–5 M	23 k–600 k	15 k–200 k
Pines de entrada y salida	204–1,200	88–1108	176–804	180–512	124–784	182–514	86–284
Flip-flops	2,816–88,192	512–93,184	1,392–64,896	1,392–24,576	1,536–66,560	1,536–13,824	384–4,704
Frecuencia Máxima.	547 MHz	420 MHz	240 MHz	200 MHz	326 MHz	200 MHz	200 MHz
Voltaje de Alimentación	1.5 V	1.5 V	1.8 V	2.5 V	1.2 V	1.8 V	2.5 V
Interconexión	SRAM	SRAM	SRAM	SRAM	SRAM	SRAM	SRAM
Tecnología	0.13 u 9-layer copper CMOS	0.15 u 8-layer metal CMOS	0.18 u 6-layer metal CMOS	0.22 u 5-layer metal CMOS	0.09 u 8-layer metal CMOS		
Bits de SRAM	216 k–8 M	72 k–3 M	64 k–832 k	32 k–128 k	72 k–1.8 M	32 k–288 k	16 k–56 k

Tabla 1. FPGAs de Xilinx.

Tenga en cuenta que todas las FPGAs de Xilinx usan SRAM para almacenar las interconexiones, por lo que son reprogramables, pero volátiles (por lo que se necesita una ROM exterior). Por otro lado, los FPGAs de ACTEL son no volátiles (que utilizan antifuse), pero no son reprogramables (excepto alguno que utiliza memoria Flash). Dado que cada método tiene sus propias ventajas y desventajas, la aplicación efectiva demanda que arquitectura de chip es más conveniente.

Las aplicaciones en FPGA son aparentemente infinitas. Mientras el precio y el tiempo de desarrollo se pueden justificar, un FPGA tiene un lugar en cualquier diseño. Muchos FPGAs se utilizan en aplicaciones complejas en las que el diseño de un ASIC no tiene un sentido financiero.

Como por ejemplo: en equipo de redes, de imágenes médicas, de defensa y aeroespacial, son algunas de las áreas de aplicación que cuentan con una gran cantidad de diseños en FPGA.

En cuanto a las medianas y grandes aplicaciones, los FPGAs son vistos como una forma de prototipo para un determinado circuito y para luego convertirse en ASICs cuando alcanzan un cierto umbral. En otras palabras, una FPGA se puede personalizar con la intención de convertirse en un ASIC. Altera es un fabricante de FPGA que ofrece un plan de trabajo a un ASIC estructurado.

Para muchas aplicaciones se usan también los microcontroladores y microprocesadores, sin embargo, a menudo no cumplen con las necesidades exactas del diseñador. Por el contrario los FPGAs dan paso a los diseñadores la flexibilidad casi ilimitada. Un microcontrolador básico, como por ejemplo un ARM 7 o 8051, puede ser incluido en un diseño y, a continuación, cualquier periférico puede agregarse, siempre y cuando la FPGA sea lo suficientemente grande.

5.2 Proceso de diseño en los FPGAs.

Hay tres enfoques utilizados en diseño de los FPGA. El primero es por medio de circuitos esquemáticos, la cual es la más sencilla y más centrada en diseño de hardware. Se trata de esquemáticos de circuitos que se introducen dentro de un FPGA usando las herramientas de diseño. Aunque es visual, este enfoque es la menos flexible y no permite la fácil migración de una plataforma FPGA a otro.

El segundo es el uso de HDL o lenguaje de descripción de hardware, que son lenguajes de idiomas específicos para el desarrollo de FPGAs y ASICs, y es más abstracta para un diseño de ingeniería. Muchos diseños basados en FPGA son completados con el HDL, pues hay varios vendedores que usan componentes que necesitan un controlador especifico, la salida clásica es un desarrollo de estos controladores por medio de HDL. Una de las principales ventajas de HDL es que el diseño es más portátil y flexible que la del esquemático.

El más bien impopular es la tercera y última opción la cual es funciones de máquina de estados. Esta técnica implica básicos "sí" y "no" en sus estados, en otras palabras, diseño digital fundamental.

En el caso de implementarlo en el código VHDL, el archivo creado se guarda en un archivo con la extensión “.vhd” y con el mismo nombre de la entidad.

El primer paso en el proceso de síntesis es la compilación. Para VHDL, la compilación es la conversión del lenguaje de alto nivel (VHDL), que describe el circuito en el Nivel de Transferencia de Registro (RTL), dentro de un netlist a nivel de compuertas.

El segundo paso es la optimización, que se realiza en el netlist de nivel de compuerta para mejorar en la velocidad o en el área. En esta etapa, el diseño puede ser simulado.

Una vez que el diseño se considera completo, el diseño es simulado. La simulación es una forma la cual el diseñador, pueda entender cómo el circuito va a funcionar en forma temporal para hacer ajustes si son necesarios.

Por último, el place and route (enrutamiento o colocación precisa) generará la disposición física de un PLD / FPGA chip o generará las máscaras de un ASIC. Es decir el netlist sintetizado es mapeado dentro la distribución física del dispositivo y ruteado dentro las capas de interconexión de datos.

Debido a que el “place and route” (enrutamiento), conoce las limitaciones de diseño, es capaz de seleccionar la lógica de bloques e interconectar por medio del software de desarrollo los elementos necesarios para que sean programados en un dispositivo FPGA. Tras la finalización del “place and route”, el diseño puede ser cargado al FPGA y también puede ser físicamente testeado. En caso de surgir algún problema, el diseño puede ser modificado, re-sintetizado, y el proceso se repite.

Una de las principales utilidades, de cualquier forma de diseño, es que permite la síntesis de un sistema o de un circuito a un dispositivo programable (PLD o FPGA) o en un ASIC.

Los pasos para seguidos durante un proyecto de este tipo se resumen en la figura 44, la cual esta adaptada de [8].

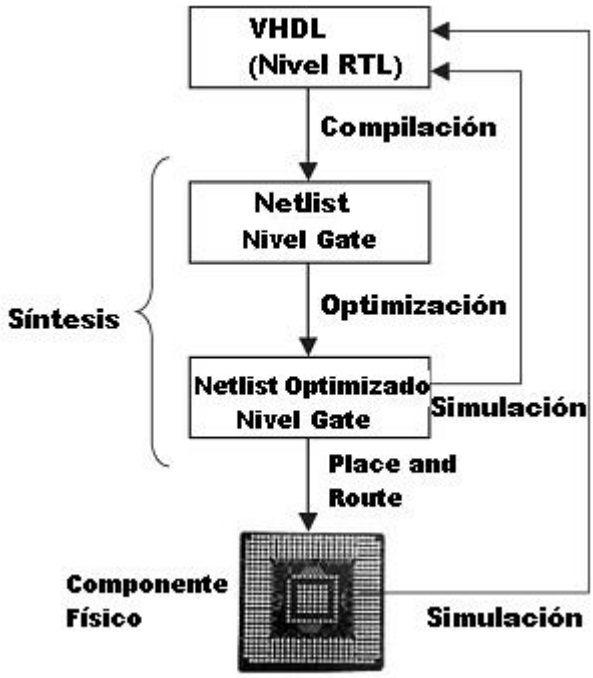


Figura 44. Proceso del diseño.

Los softwares de diseño se conocen como EDA (Electronic Design Automation) las cuales tienen herramientas disponibles para la síntesis, implementación y simulación usando VHDL. Algunas herramientas (place and route, por ejemplo) se añaden como parte de software de los proveedores de FPGAs, como por ejemplo, Quartus II de Altera, que permite la síntesis de código de VHDL en chips CPLD y FPGAs. Por otro lado, Xilinx tiene el ISE, para los chips CPLDs y FPGAs de Xilinx.

Otras herramientas (sintetizadores por ejemplo), además de ser parte de las gamas de diseño, también puede ser proporcionada por las empresas especializadas en EDA como por ejemplo Mentor Graphics, Sinopsis, Synplicity, etc. Ejemplos de este último grupo son Leonardo Spectrum (un sintetizador de Mentor Graphics), Synplify (un sintetizador de Synplicity), y ModelSim (un simulador de Model Technology, una empresa de Mentor Graphics).

5.3. Tipos de FPGAs para los sistemas Aeroespaciales.

A diferencia de los FPGAs no volátiles basados en ”antifuse” o antefusible, Los FPGAs basados en tecnología SRAM son reprogramables, los diseñadores pueden tener la

posibilidad de reconfigurar un circuito de un satélite en el espacio exterior (no es tan imposible si se encuentra cerca un transbordador espacial, pero es muy improbable) . Además, los vendedores de FPGAs basadas en SRAM, suelen ofrecer densidades mayores que las soluciones de antifuse basadas con casi el mismo rendimiento. Como resultado, los diseñadores a menudo consideran los FPGAs basadas en SRAM en la carga útil de satélites.

Sin embargo, esta flexibilidad tiene un precio. La gran mayoría de los FPGAs basados en SRAMs son vulnerables a los efectos de los altos niveles de la radiación cósmica. Los iones pesados de los rayos cósmicos pueden depositar fácilmente suficiente carga cerca de una celda SRAM para causar un solo bit de error, o SEU. Por otra parte, los FPGAs basados en SRAM almacenan su configuración lógica en switches SRAM, son susceptibles a la configuración UPSET, lo que significa que el enrutamiento y la funcionalidad del circuito puede ser alterado o corrompido. Este tipo de errores son muy difíciles de detectar y corregir, y son casi imposibles de prevenir, puesto que la configuración de switches representa más del 90 por ciento del total de SRAM bits en una FPGA. Los UPSETs inducidas por la radiación pueden dar lugar a fallos en el sistema.

Característica	**SRAM**	**FLASH**	**ANTIFUSE**
Reprogramable	Si	Si	No
Volátil	Si	No	No
Necesidad de Memoria externa	Si	No	No
Consumo de potencia	Alto	Medio	Bajo
Inmunidad a la radicación	No	No	Si

Tabla 2. Comparación entre las tecnologías de FPGAs.

Los diseñadores de sistemas espaciales se enfrentan a retos únicos. Se va a explicar sobre la versión de Xilinx para desarrollo aeroespacial. Para hacer frente a esos desafíos Xilinx tiene FPGAs para aplicaciones espaciales, que pueden mejorar el rendimiento del sistema, aumentar la flexibilidad y reducir los ciclos de desarrollo. Sus chips tolerantes a la

radiación están basados en FPGAs Virtex, los cuales tienen encapsulados especiales, y reúnen las condiciones para los requerimientos del espacio.

La gama de los FPGAs Virtex-4QV, se muestran en la figura 45, la cual esta adaptada de [12], y la inmunidad ante TID (total ionizing dose) y SEL (single-event latch-up) están garantizadas. Xilinx fue pionera en la aplicación de SRAM basadas en FPGAs en alta radiación, que caracteriza y reporta resultados con SEU (single-event upset).

		Virtex-4QV Optimizado para:			
		Lógica	PDS	Procesamiento embebido	
Componente		XQR4VLX200	XQR4VSX55	XQR4VFX60	XQR4VFX140
Voltaje		1.2V	1.2V	1.2V	1.2V
Slices		89,088	24,576	25,280	63,168
Celdas lógicas		200,448	55,296	56,880	142,128
CLBs		178,176	49,152	50,560	126,336
Max. RAM		1,392	384	395	987
Manejador de CLk digital		12	8	12	20
Max. I/O de forma simple		960	640	576	896
Max. I/O de forma diferencial		480	320	224	448
Impedancia controlada Digital		Yes	Yes	Yes	Yes
Slices para PDS		96	512	128	192
Procesador PowerPC		—	—	2	2
Dosis de Radiación	(krad)	300	300	300	300
Inmunidad SEL	(MeV-cm²/mg)	>125	>125	>125	>125
Empaquetadura	Area	I/Os disponibles para el usuario			
CGA Packages (CG): ceramic column grid array (1.27 mm ball spacing)					
CF1140	35 x 35 mm		640		
CF1144	35 x 35 mm			576	

Figura 45. Virtex-4QV para diferentes aplicaciones aeroespaciales.

CAPITULO 6. IMPLEMENTACION DEL FILTRO DE KALMAN EN EL FPGA

Para la implementación del filtro, es necesario conocer el proceso de diseño con FPGAs, utilizando la herramienta system generator de xilinx, la cual nos dará un mejor panorama del diseño final.

6.1 SYSTEM GENERATOR.

Los FPGAs son ampliamente utilizados para PDS de alto rendimiento. La Plataforma de FPGAs Virtex II de Xilinx, se ha convertido rápidamente en la principal plataforma de alto rendimiento para aplicaciones de PDS.

El System Generator de Xilinx llena el vacío entre la versión abstracta de un diseño de PDS y su aplicación efectiva en un FPGA de Xilinx. El System Generator de PDS desarrollado en colaboración con MathWorks, permite a los diseñadores implementar sistemas de alto rendimiento de PDS en FPGAs de Xilinx utilizando el Simulink de MATLAB. Las capacidades incluyen la Co-Simulación, Hardware in the loop (HIL), y la capacidad para estimar los recursos del FPGA dentro de Simulink. En la figura 46, se muestra la pantalla principal de Matlab.

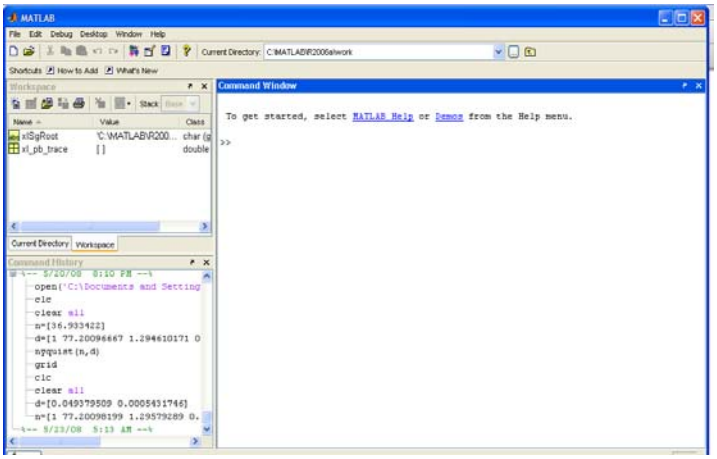


Figura 46. Pantalla principal de Matlab.

Al utilizar Simulink de Matlab, es necesario estar familiarizado con los Toolbox de esta herramienta, puesto que System generator se va colocar como una de ellas. Esta herramienta simplificadora permite a los diseñadores utilizar el software de desarrollo ISE, que sirve para la síntesis de VHDL, la cual es llamada desde Matlab por System Generator para la compilación, ruteo y mapeo en el FPGA, la cual puede ser implementada directamente en el FPGA desde Matlab.

En la figura 47, se muestra el toolbox de simulink en donde aparece el blockset de xilinx.

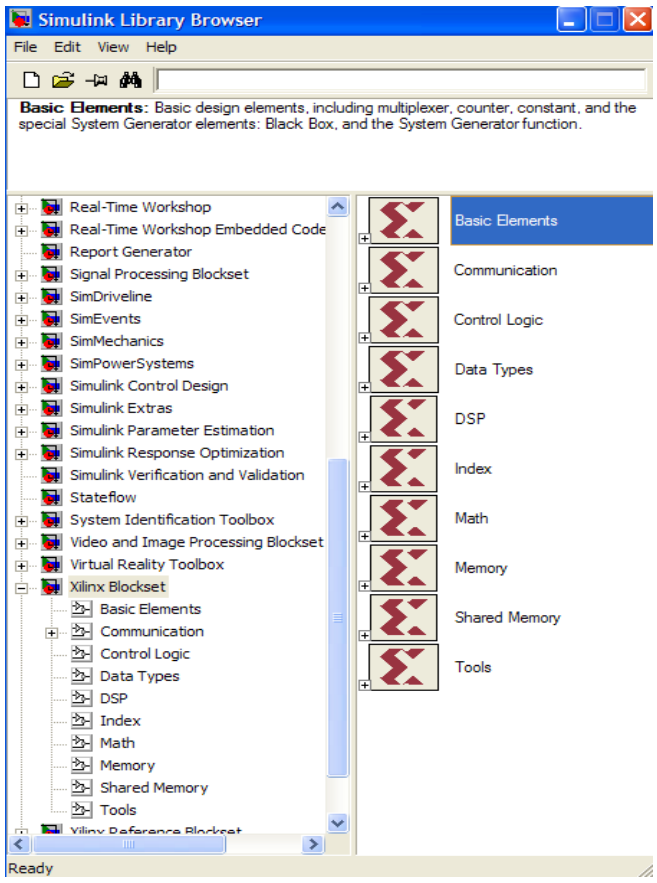


Figura 47. Blockset de Xilinx.

El System Generator provee una simulación que se proporciona a través de hardware co-simulation. La herramienta crea automáticamente una plataforma de simulación de hardware para un diseño hecho con el blockset de Xilinx, la que se puede ejecutar en más de 20 hardwares.

Esta herramienta también puede ser co-simulado con el resto del sistema de Simulink.

Un ejemplo de desarrollo en simulink y system generator se muestra en la figura 48.

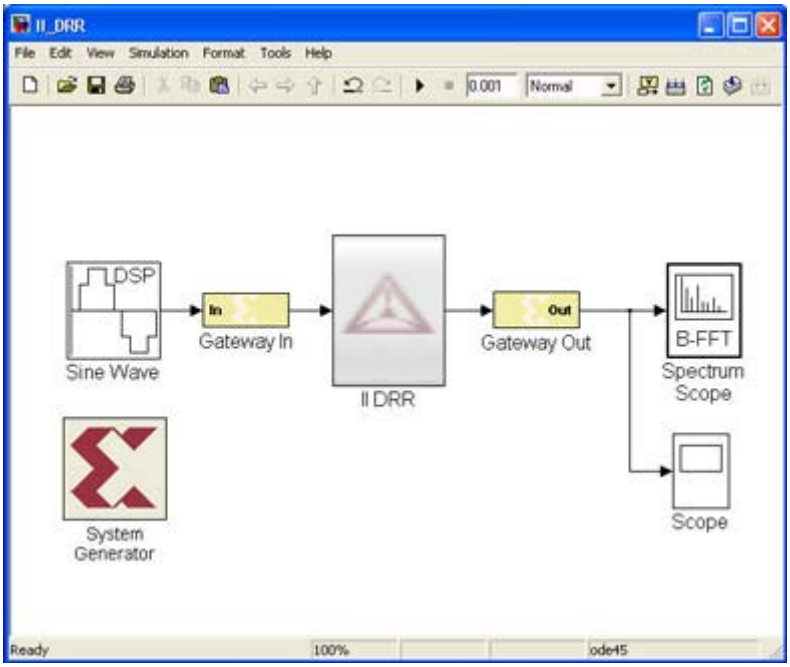


Figura 48. Desarrollo en simulink con system generator.

6.1.1 El diseño con system generator.

System Generator trabaja dentro de Simulink, que es una herramienta de matlab basado en diseño de sistemas. A menudo, los modelos los sistemas se realizan en simulink mediante los bloques de Simulink, los cuales usan punto flotante de precisión numérica, conforme al estándar IEEE 754, y sin ninguna característica de hardware.

Una vez que la funcionalidad y flujo de datos básicos de las cuestiones se han definido, System Generator puede usarse para especificar los detalles de la implementación de hardware para los dispositivos de Xilinx.

Ya creado el sistema con system generator, se puede dar inicio al Core Generador Xilinx para generar los netlists altamente optimizados para la construcción de bloques de PDS.

Desde system generador, llama a ISE para ejecutar todas las operaciones necesarias para crear el bitstream para la programación del FPGA. Se pueden crear Testbenchs opcionales a través de Simulink para ser usados con ModelSim o el Simulador de ISE de Xilinx.

El flujo de diseño se muestra en la figura 49, la cual es adaptada de [12].

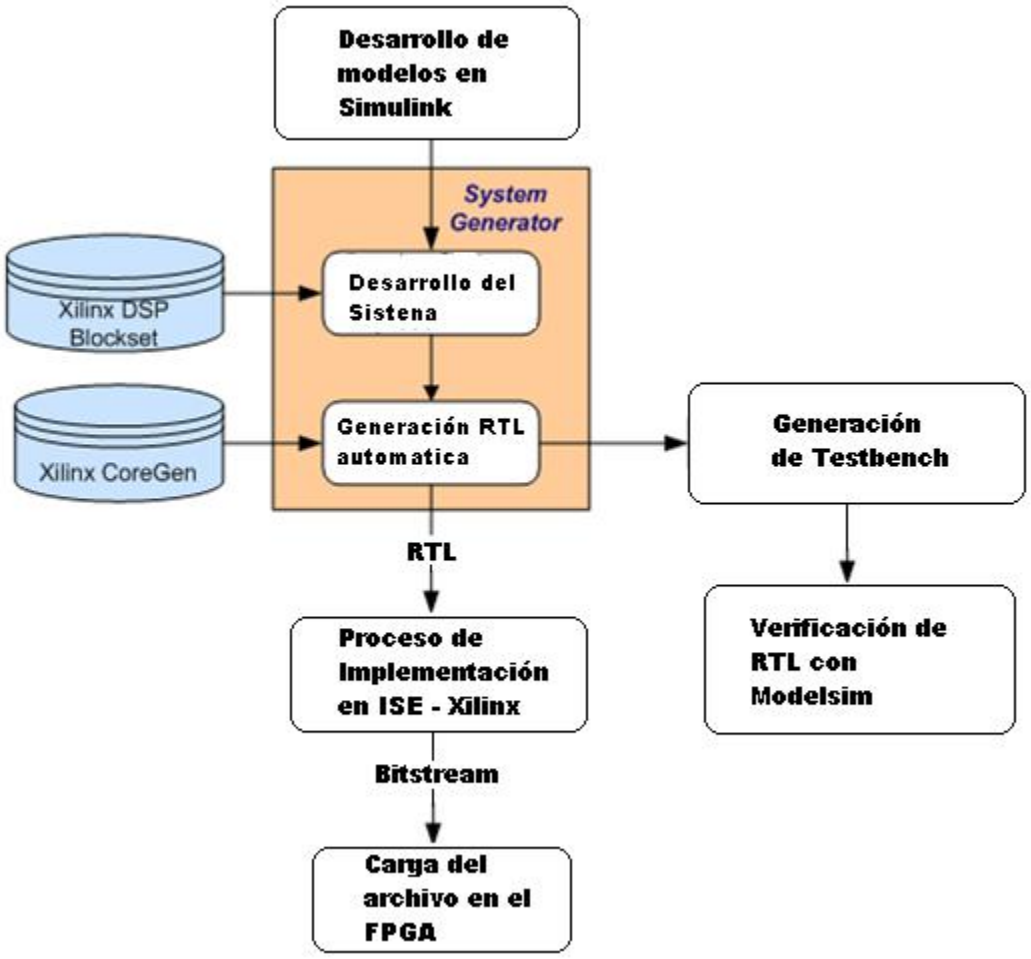


Figura 49. Flujo de desarrollo con System generator.

Un diagrama del sistema de desarrollo se muestra en la figura 50, la cual es adaptada de [12].

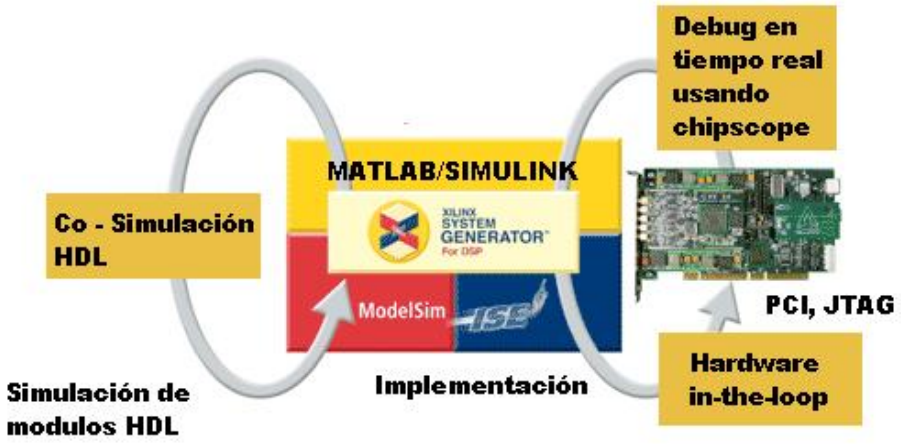


Figura 50. Diagrama del Sistema de desarrollo.

Con estas herramientas, se pueden hacer las comprobaciones HARDWARE-SOFTWARE, para probar de una manera más realista los filtros desarrollados en Matlab, que después se implementan en hardware por medio de system generator.

6.2 Desarrollo del Hardware para el Filtro de Kalman.

En esta parte del trabajo de investigación, primero se va a analizar el sistema con ecuaciones de espacio de estado, para poder implementarlas después en hardware. Segundo se va a mostrar cómo se empezó a implementar el filtro de kalman, comenzando con un diagrama de bloques hasta el uso de system generator.

Para este trabajo usamos el FPGA Virtex 2 pro de xilinx, el cual tiene muchos recursos para PDS, tal como se describió anteriormente, el cual su layout del FPGA se muestra en la figura 51, que es adaptada de [12].

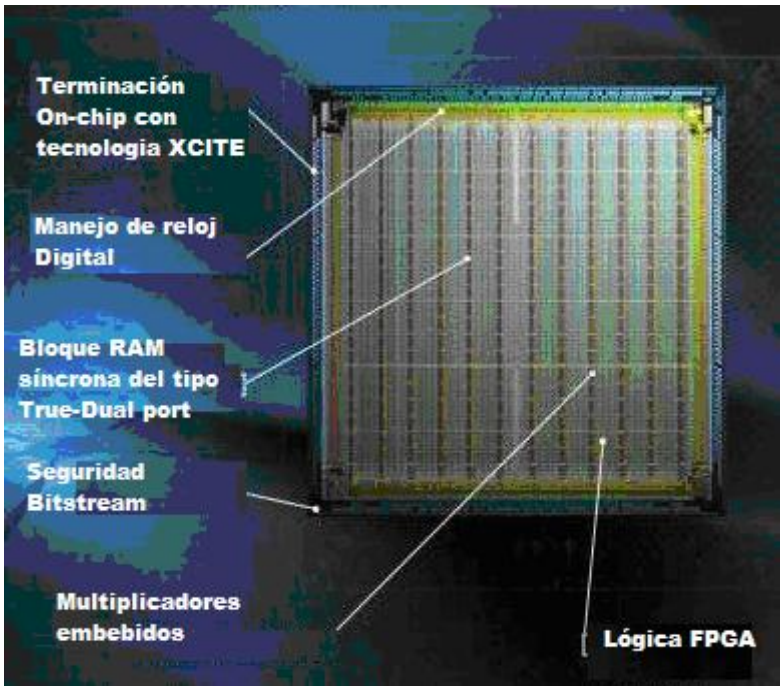


Figura 51. Layout real del FPGA virtex 2 pro.

Este FPGA ya se mencionó anteriormente, y sus características están en la tabla 1, entonces procedamos a implementar el filtro de kalman.

Ahora, tenemos que ver que ecuaciones se van a implementar, es por ello que tenemos que establecer las ecuaciones para este proceso.

6.2.1 El modelo del proceso.

Para poder implementar las ecuaciones del filtro de kalman es muy necesario conocer el proceso, por ello empezamos definiendo el trabajo, el cual trata de estimar la señal del sensor de temperatura, que es la salida de voltaje del circuito de acondicionamiento de señal que está en función de la temperatura.

Con el análisis en los anteriores capítulos, hemos comprobado que es un ruido blanco gaussiano, la cual contamina a la señal con un voltaje aproximado de 0.01v. Esto se mostró en la figura 20, con ello vemos también la no correlación del ruido con el tiempo.

El sistema se puede escribir de esta manera, de la ecuación 4.1 podemos simplificar a la siguiente:

$$x_k = Ax_{k-1} + w_k \text{ ……. (6.1)}$$

Y para el caso de la medición se escribe de esta manera de la ecuación 4.2:

$$z_k = x_k + v_k \text{ ……. (6.2)}$$

Nuestro objetivo es el de estimar el estado, y viendo que no hay entrada de control podemos colocar ($A = 1$). Nuestra medición de ruido es de manera directa, entonces $H = 1$

6.2.2 Las ecuaciones y los parámetros del filtro.

Nuestras ecuaciones de tiempo de actualización son

$$\hat{x}^{-}_k = \hat{x}_{k-1} \text{ …… (6.3)}$$

$$P^{-}_k = P_{k-1} + Q \text{ . ……(6.4)}$$

Y nuestras ecuaciones de medida de actualización son:

$$K_k = \frac{P^{-}_k}{(P^{-}_k + R)} \text{ . ……. (6.5)}$$

$$\hat{x}_k = \hat{x}^{-}_k + K_k(z_k - \hat{x}^{-}_k) \text{ . …….. (6.6)}$$

$$P_k = (1 - K_k) P^{-}_k \text{ . ……(6.7)}$$

Analizando el proceso, la varianza es muy pequeña, (pero para afinar el filtro de kalman ciertamente se podría dejar el valor de $Q = 0$, pero podemos colocar un valor más pequeño con $Q = 0.00001$ para obtener un mejor resultado, pues nos da mayor flexibilidad.

Se sabe que el verdadero valor randomica tiene una distribución estándar de probabilidad normal, por lo que podemos comenzar con un valor de inicio para x , haciéndolo de esta manera: $\hat{x}_{k-1} = 0$.

Del mismo modo tenemos que elegir un valor inicial para P. Si estuviéramos absolutamente seguros de que nuestra estimación inicial era correcta entonces $\hat{x} = 0$. Sin embargo, dada la incertidumbre en nuestra estimación inicial, la elección podría causar que el filtro se demore en el procesamiento, aunque la alternativa no es crítica, podríamos elegir un $P = 0$, y sabiendo que el valor P del filtro a la larga convergen.

6.2.3 Implementación del filtro de kalman en el FPGA.

Como ya tenemos el proceso modelado con las respectivas ecuaciones, ahora con ello podemos realizar una arquitectura característica que envuelva rapidez y área para poder implementar el filtro discreto de kalman.

Primero, las ecuaciones deben usar un tipo de formato, lo cual se escogió un formato de punto flotante de 64 bits, por tener muchas ventajas con respecto a los punto fijo, también es necesario tener un conocimiento previo de los cálculos con punto flotante en hardware, como también tener un conocimiento previo de usar el punto flotante en Matlab.

Entonces empezamos haciendo una arquitectura que nos describa las ecuaciones del filtro discreto de kalman, que trabajen en forma paralela como se muestra en la figura 52.

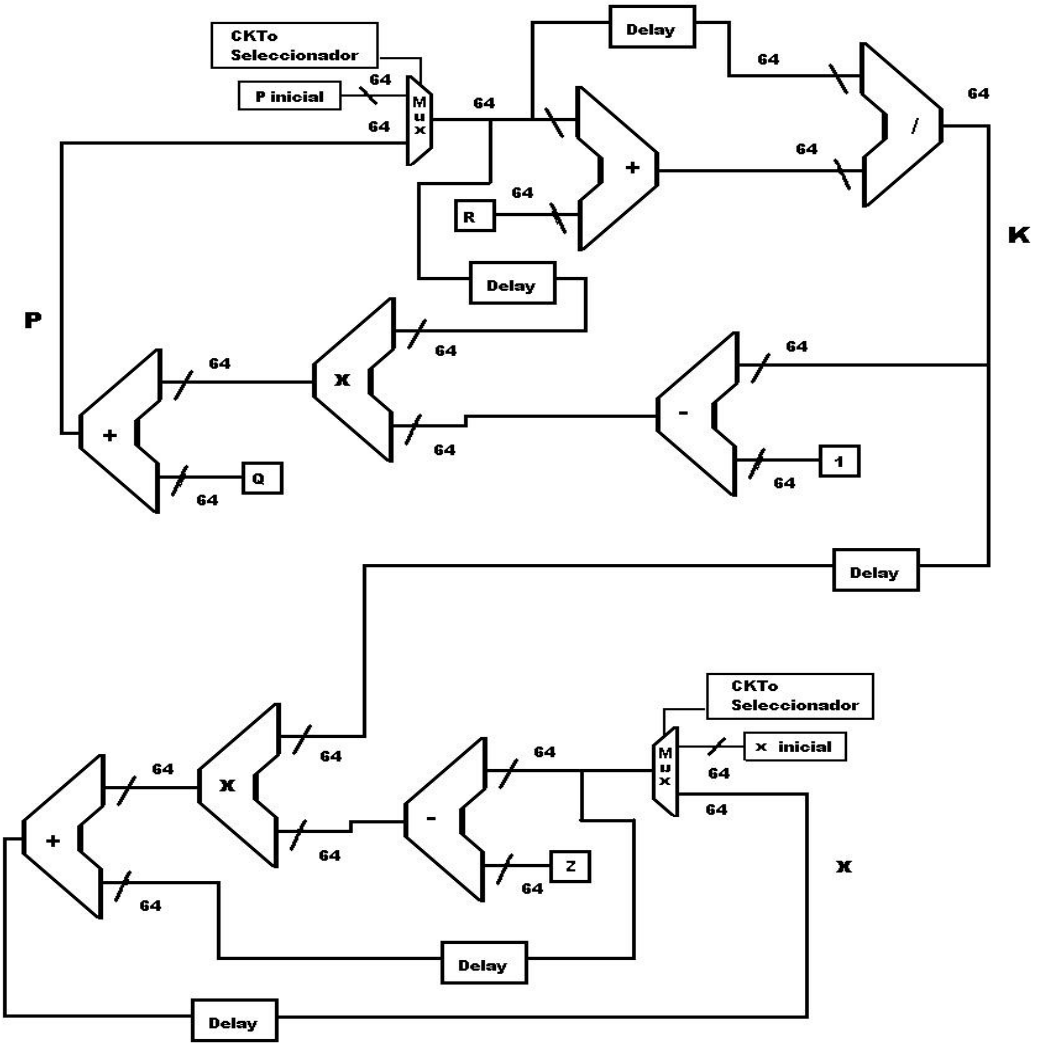


Figura 52. Diagrama de bloques del filtro discreto de kalman.

En este caso tenemos que tener en cuenta los siguientes puntos:

- **Bloques de aritmética en punto flotante**, el cual nos ayuda a trabajar con números de 64 bits, se deben primero ser implementados usando el core generator de xilinx, los cuales son exportados por Matlab, por medio de simulink con los bloques de black box de system generator.
- **Datos para el procesamiento,** para este caso, es muy necesario tener en cuenta los datos en Matlab, para poder exportarlos a simulink, y asi poder hacer el procesamiento con el filtro de kalman.

- **Sincronización de los circuitos,** en este caso cuando tengamos el sistema funcionando, tenemos que tener en cuenta cuanto tiempo consume cada bloque, con ello se coloca los delays necesarios para poder hacer que el bloque total funcione correctamente y se estén haciendo las operaciones en su respectivo orden. También hay que enfatizar que los circuitos funcionan desde el primer ciclo de reloj, lo cual nos muestra que empiezan todos en paralelo.

Por ejemplo se muestra la cantidad de pulsos de clock que se demoran en hacer las respectivas operaciones en la tabla 3, que es adaptado de [11].

Parametros		Latencia (Ciclos)					
Tamaño Fracción	Tamaño Exponente	Suma	Multiplicación				División y Raiz Cuadrada
			Lógica sola	Mult18x18	Dsp 48	Dsp 48 + lógica	
8	4 & 6	10	6	5	6	6	11
10	4 & 6	10	6	5	6	6	12
12	4 & 6	10	7	5	6	6	15
14	6 & 8	10	7	5	6	6	17
17	6 & 8	11	7	5	6	6	20
20	6, 8 & 10	11	7	6	9	11	23
22	6, 8 & 10	11	7	6	9	11	25
24	6, 8, & 10	11	8	6	9	11	27
34	8	12	8	6	9	9	37
53	11	12	9	10	21	17	56

Tabla 3. Retardos de procesamiento de datos, acondicionado del datasheet del core generator.

Al poder entrar a simulink, tenemos que usar los bloques black box, el cual tenemos que modificar el archivo .m creado por matlab para poder importarlos desde simulink.

Cada circuito que se crea, se debe tener en cuenta que cuando se corre en el simulink, todos estos empiezan a correr al mismo tiempo, pues son paralelos.

Cuando se termina de importar, en la misma pagina de simulink, se deben estructurar las ecuaciones como la arquitectura ya mostrada en la figura 52. Cuando terminemos de armar

las ecuaciones del filtro de kalman dentro de simulink, tendremos un diagrama como en la siguiente figura 53.

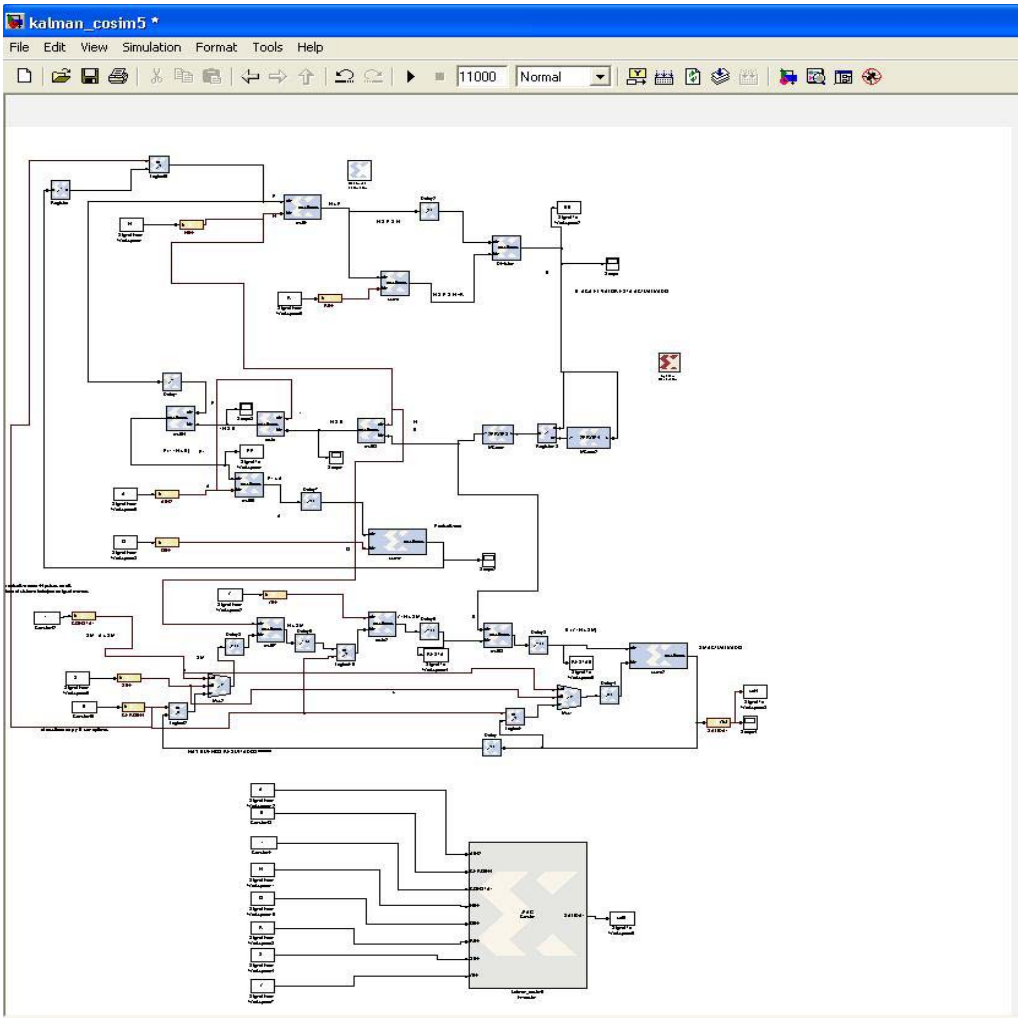


Figura 53. Diagrama de hardware en simulink.

Cuando esta correcto, sin ningún error y ya depurado por medio de system genrator , la herramienta crea una nueva librería para la co-simulacion, la cual se debe interfasear con el hadrware del FPGA, en este caso el XUPV2P30, esta librería se muestra en la figura 54.

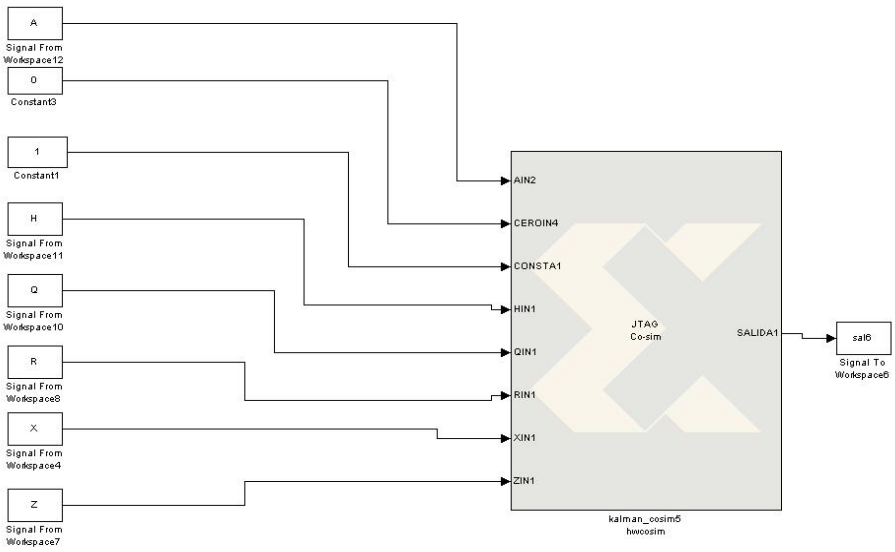


Figura 54. Circuito generado para la co-simulación.

Tenemos que tener en cuenta que en simulink, estamos viendo el hardware como todo un sistema, como si estuviéramos usando solo componentes como en VHDL, que se unen por medio de señales.

Usando el *Block Parameters Dialog Box* de system generator, como se muestra en la figura 55, podemos configurar los parámetros para la simulación, también qué tipo de FPGA se va a usar y por último el de generar la compilación para crear el hardware que se desarrollo en simulink.

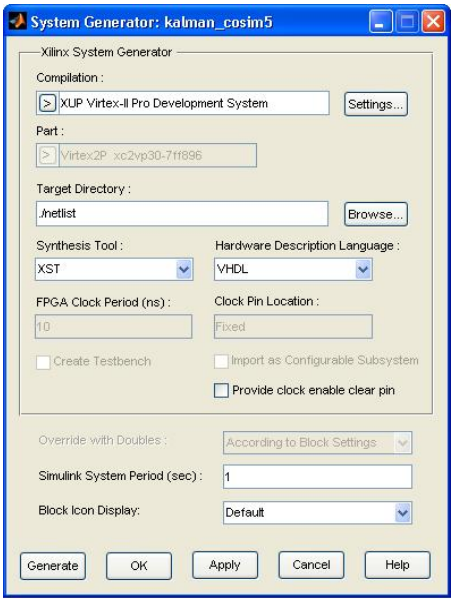


Figura 55. Parámetros del *Block Parameters Dialog Box* de system generator.

Si hay errores en el hardware que se crea por medio de ISE de Xilinx, se pueden corregir también en simulink. Cuando el hardware esta creado, nos aparece una librería nueva para poder hacer la co-simulación con la tarjeta XUP2VP30 de Xilinx, la que se muestra en la figura 56.

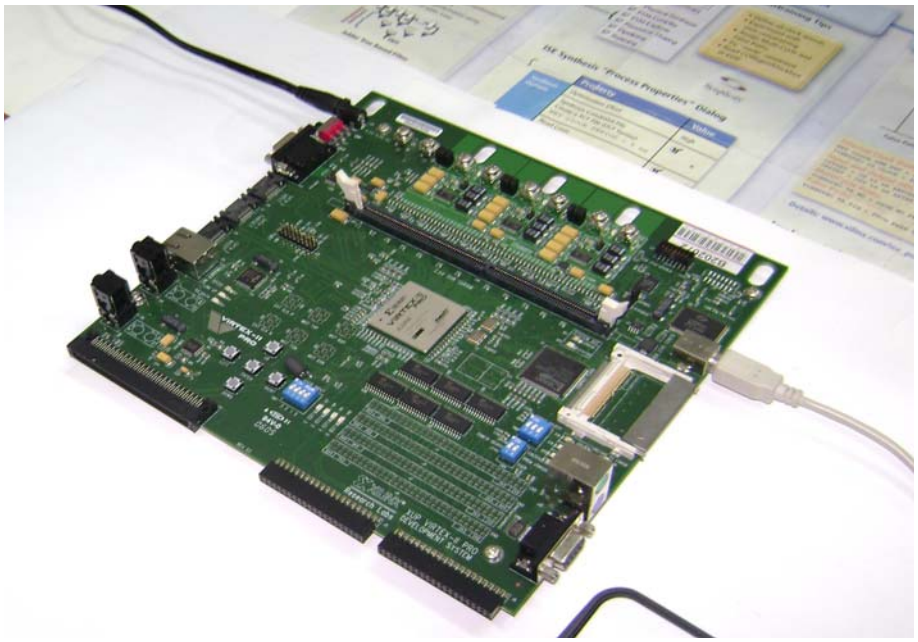


Figura 56. Tarjeta de desarrollo XUP2VP30.

Para hacer la co-simulación tenemos que cambiar de parámetros para poder llevar a cabo el trabajo, tal como se muestra en la figura 57.

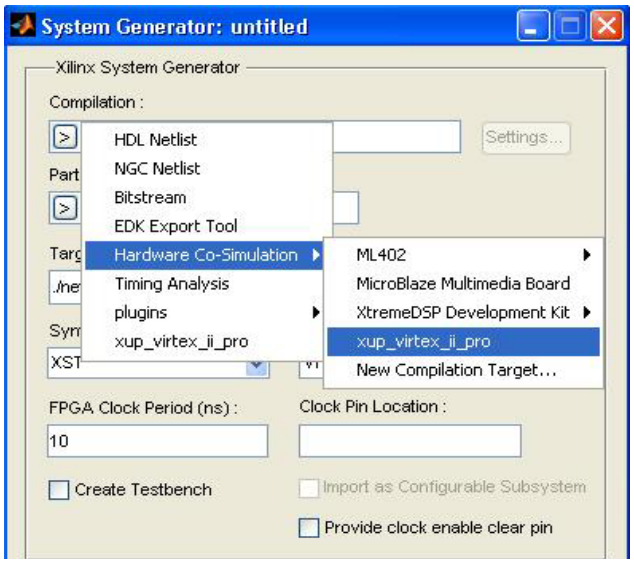


Figura 57. Parámetros para la co-simulación.

Al poder generar el hardware requerido podemos ir a ISE para ver los archivos generados por la herramienta, como se muestra en la figura 58.

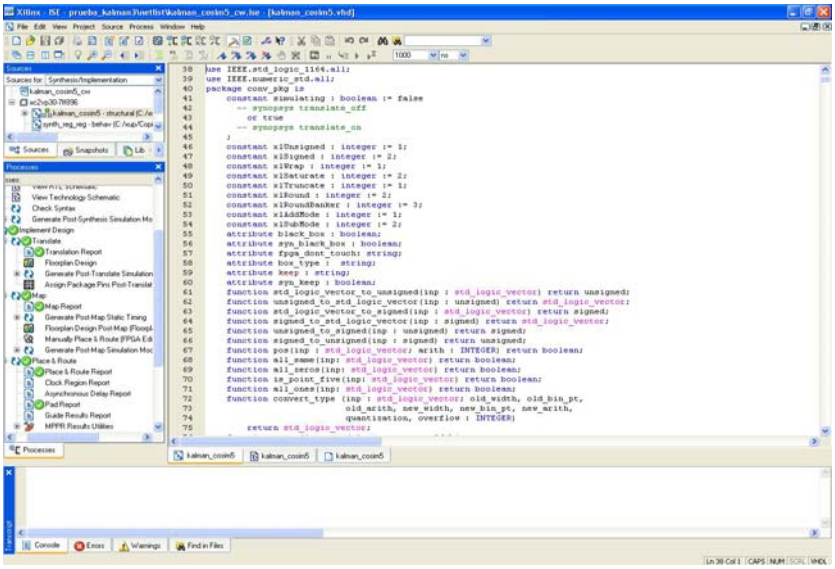


Figura 58. Pantalla principal de ISE de Xilinx.

También podemos visualizar el consumo de los recursos para nuestro filtro, puesto que en esta versión ya existe un generador de resumen de consumo de recursos de los FPGAs y CPLDs soportados por esta herramienta de xilinx, como se muestra en la figura 59.

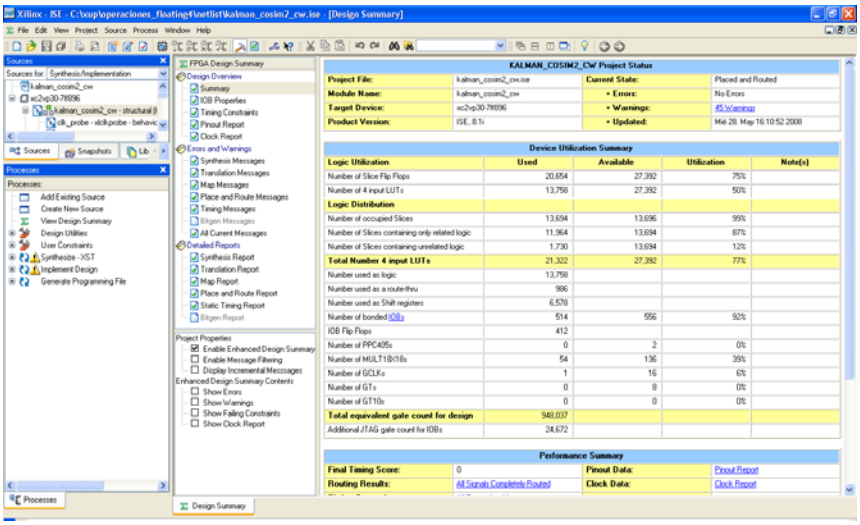


Figura 59. Resumen de los recursos utilizados para el filtro.

En la tabla 4 se muestra los recursos utilizados por el filtro discreto de kalman.

Utilización Lógica	**Usado**	**Disponible**	**Utilizado**
Número de flip-flops	22,152	27,392	80%
Número de LUTs de 4 entradas.	14,289	27,392	52%
Distribución Lógica	**Usado**	**Disponible**	**Utilizado**
Números de Slices ocupados	13,684	13,696	99%
Número de LUTs de 4 entradas	22,319	27,392	81%
Numero de lógica usada.	14,289		
Numero de route-thru	1,069		
Numero de Shift registers	6,961		

Tabla 4. Recursos utilizados por el filtro de kalman.

El equivalente de la cantidad de compuertas usadas es de: 1,116.401.

En ISE de Xilinx, podemos ver el diagrama RTL que se compiló para el filtro discreto de kalman, la cual se muestra en la figura 60.

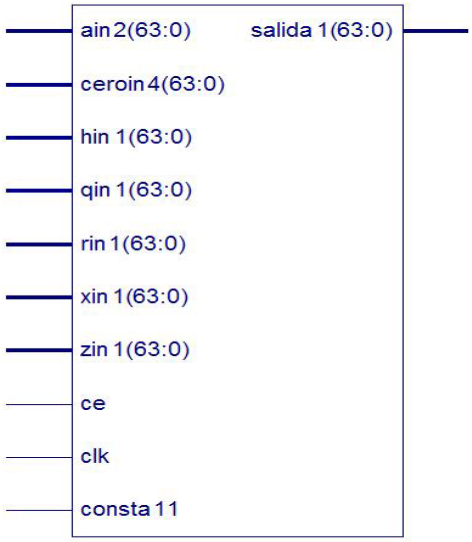


Figura 60. Filtro de Kalman empaquetado.

Se puede expandir el RTL, para obtener el circuito de la siguiente figura 61.

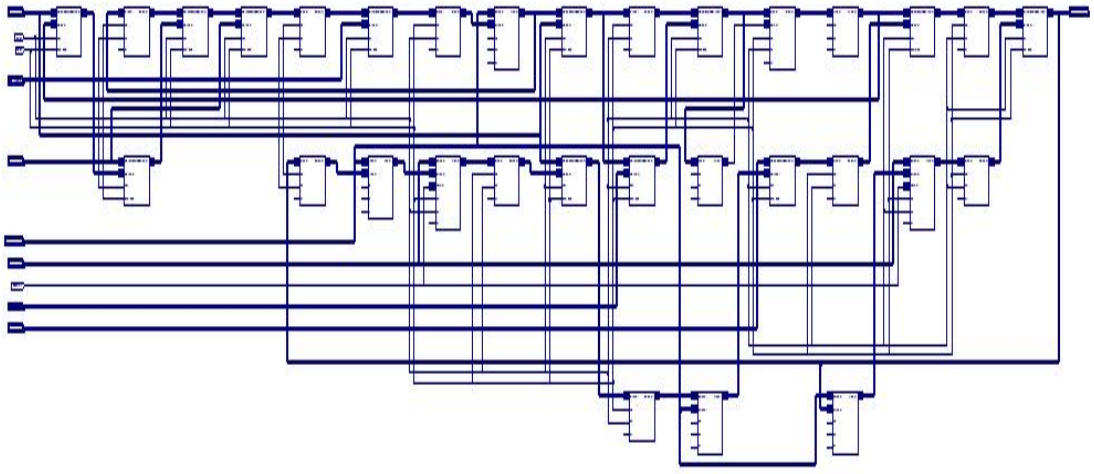


Figura 61. RTL del filtro de Kalman.

Usando las herramientas de ISE, entramos al *FPGA editor*, en donde podemos visualizar los recursos requerido por el filtro para su desarrollo por la herramienta, esta se muestra en la figura 62.

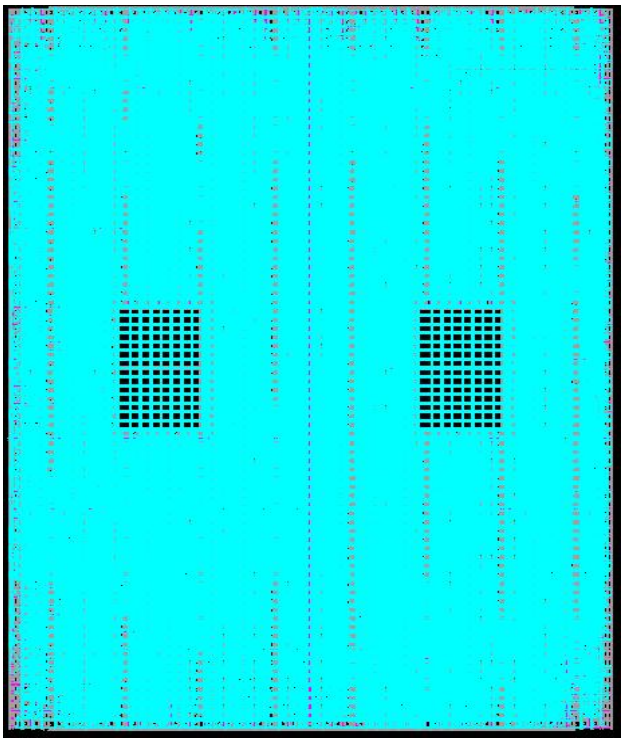


Figura 62. Layout total del filtro de Kalman.

Expandiendo dentro del layout, podemos ver lo siguiente, como se muestra en la figura 63.

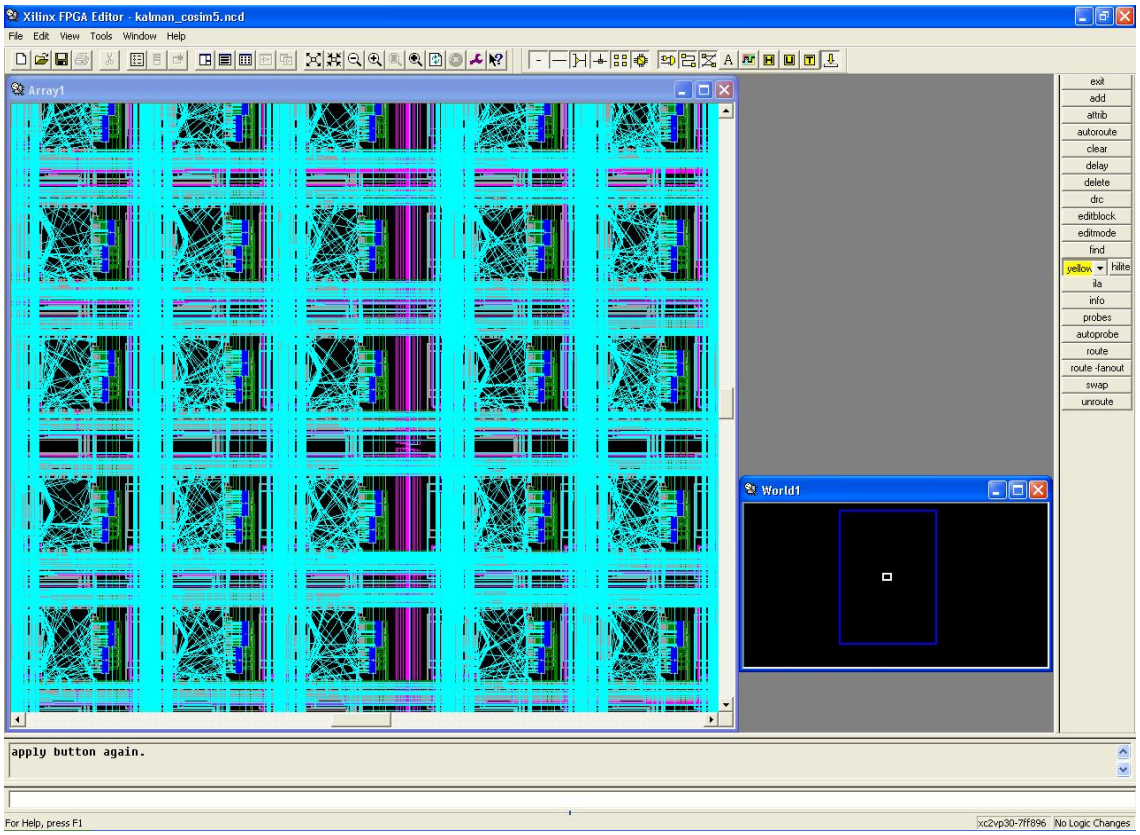


Figura 63. Layout Expandido del filtro de Kalman.

6.2.4 Prueba de linealidad del filtro.

En esta parte del trabajo se le hace una prueba al filtro discreto de kalman con una señal constante creada en matlab, en este caso se le coloca el valor de 0.226 voltios constante, para ver la respuesta del filtro.

En la figura 64, se puede visualizar la señal constante, la señal estimada de co-simulación y la resta entre la señal constante y la señal estimada por el filtro discreto de kalman.

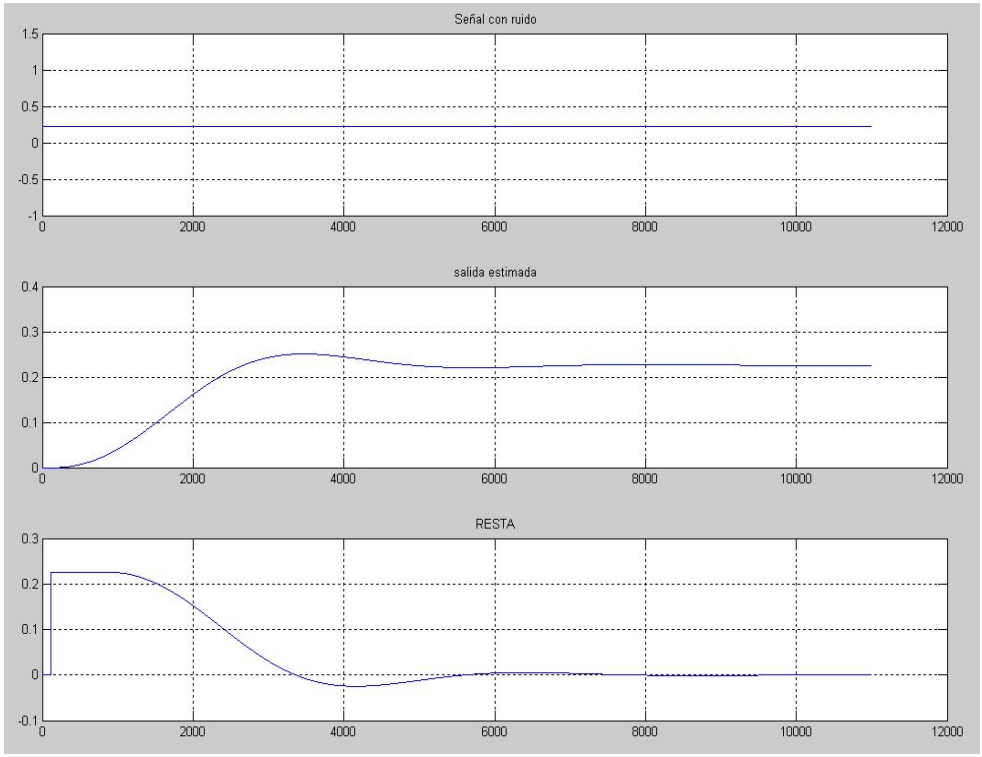


Figura 64. Señal constante, señal estimada y la resta.

En la figura 65, se puede visualizar la conexión de la tarjeta XUP2VP30 de xilinx con la PC, para realizar la co-simulación, para ello se debe definir que el puerto de comunicación es el USB.

Figura 65. Sistema de co-simulación entre la XUP2VP30 y la PC.

Al termino de la co-simulación podemos observar la respuesta mediante matlab, la señal co-simulada y la señal constante en la misma gráfica para su comparación en el tiempo, tal como se muestra en la figura 66.

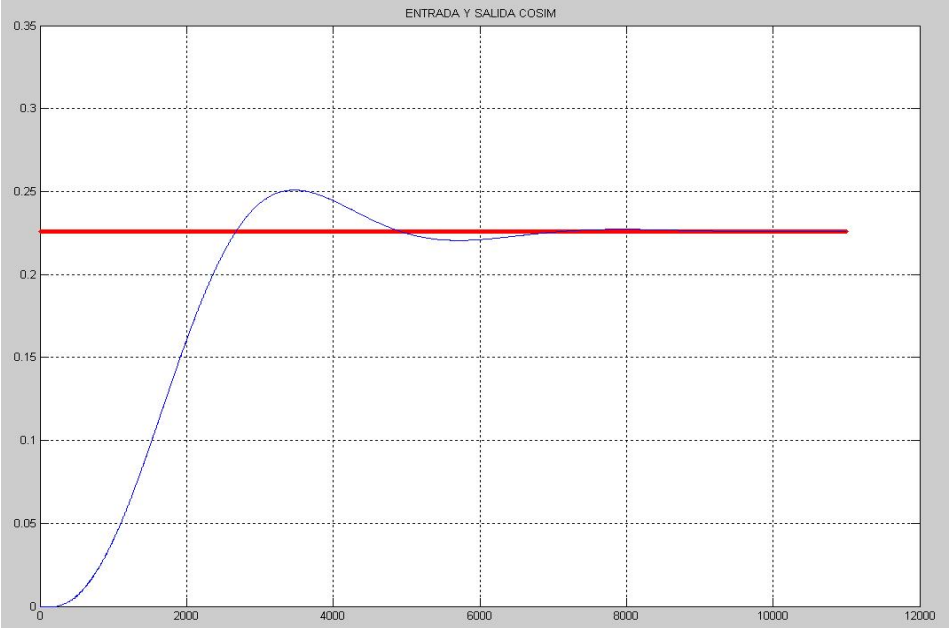


Figura 66. Entrada constante y señal co-simulada.

Al analizar la gráfica anterior, vemos que la señal estimada sigue a la señal constante por lo cual se concluye que el filtro de kalman es lineal.

En la siguiente figura, podemos visualizar una versión mas ampliada de la grafica anterior como se muestra en la figura 67.

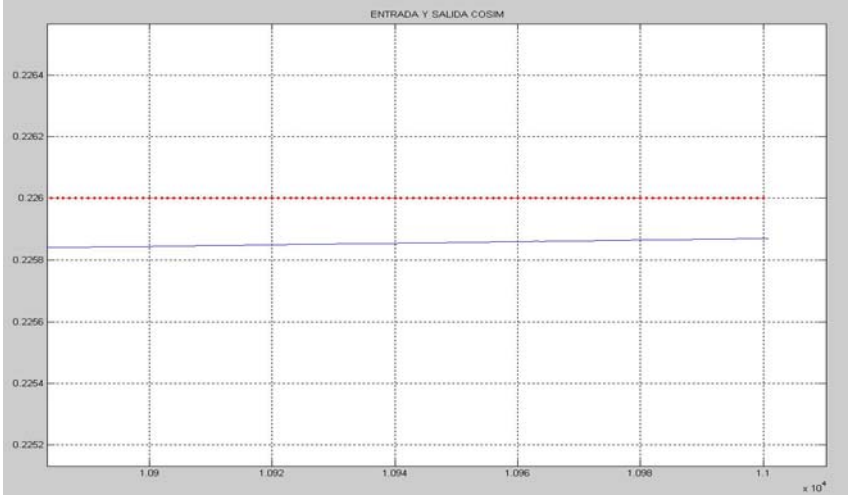


Figura 67. Versión ampliada de las señales.

En la figura 68, podemos visualizar como los parámetros del filtro de kalman evolucionan con respecto al tiempo.

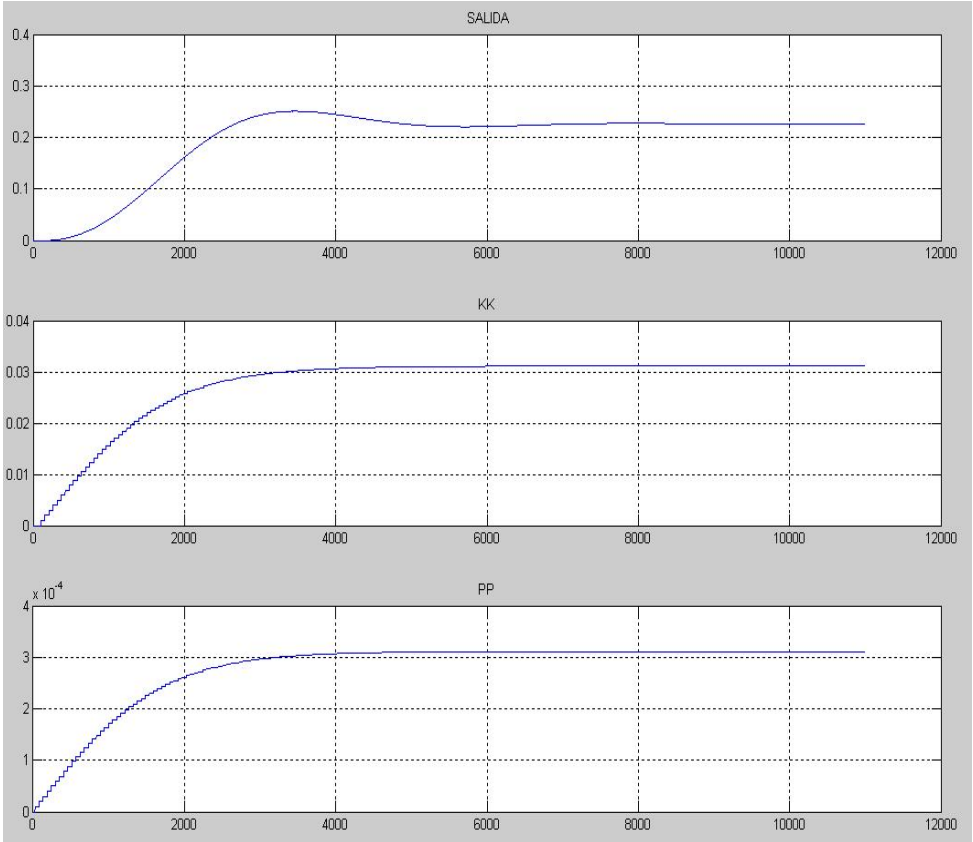


Figura 68. Evolución de los parámetros del filtro.

Al inicio, inicializamos los filtros con valores fijos: K=0 y P=0, Lo cual evolucionan en el tiempo, llegando a un valor constante al momento de que la señal está estimada de una manera óptima.

Si vemos más en detalle en la figura, podemos apreciar como las señales están en forma de escalera, esto se debe a que hay registros o delays que mantienen los datos hasta que se pueda obtener la respuesta de un bloque para poder hacer la operación.

6.3 Estimación de la respuesta del sensor de temperatura.

En esta parte del trabajo vamos a visualizar el funcionamiento implementado del filtro discreto de kalman, usando las medidas que se tienen en matlab, con ello vamos a ver el funcionamiento real tanto en software como en hardware del filtro de kalman.

Para esa temperatura medida por medio de la termocupla y comparada con la del patrón y a través del circuito de acondicionamiento de señal, podemos ver como esta contaminada la

señal de la termocupla con un ruido aleatorio, así mismo podemos visualizar la dispersión de las medidas, como se muestra en la figura 69.

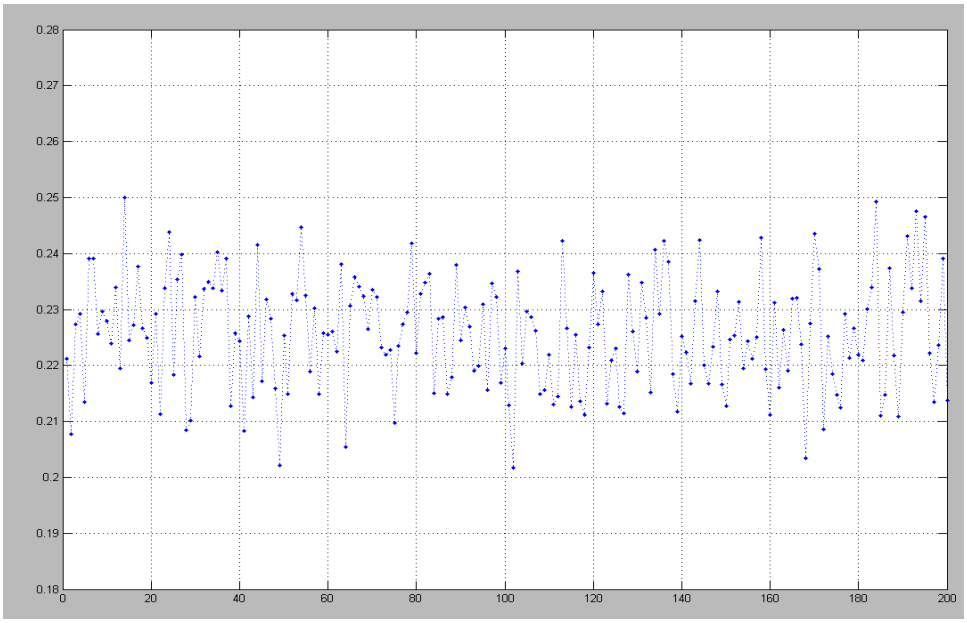


Figura 69. Señal contaminada con ruido blanco gaussiano.

Así mismo, como ya hemos definido al ruido en los capítulos anteriores, vamos a ver histograma de la señal con ruido contaminada, se nota que tiene una distribución gaussiana, esto se muestra en la figura 70.

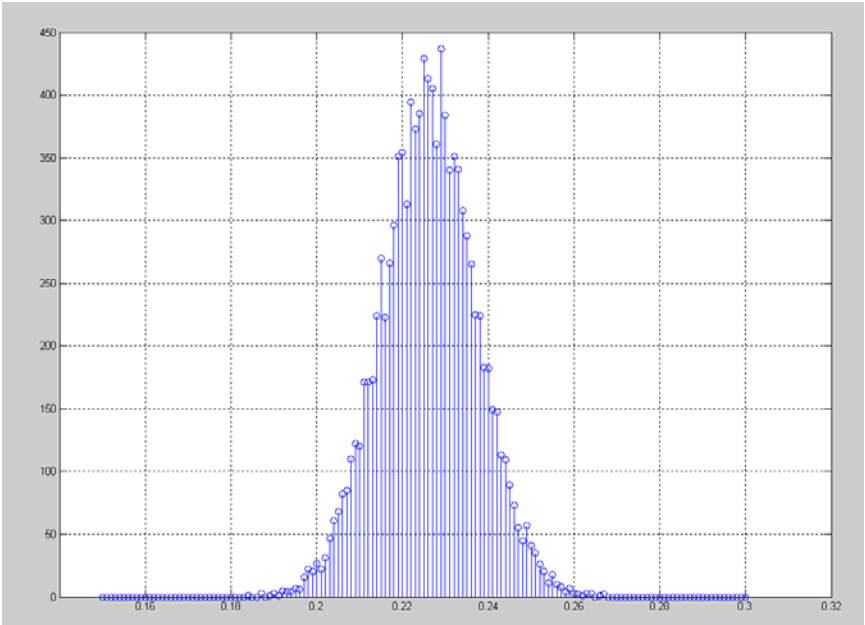


Figura 70. Histograma de la señal con ruido.

Los parámetros calculados son los siguientes:

Media = 0.2261.

Varianza = 1.2130e-004.

Desviación estándar = 0.0110.

Se han utilizado 11000 muestras de la variable con ruido para el procesamiento con el filtro de kalman, también se muestra la respuesta estimada en hardware que hizo el calculo del filtro de kalman, y se muestra la resta de la señal con ruido y la señal estimada, todo lo que se menciono se muestra en al figura 71.

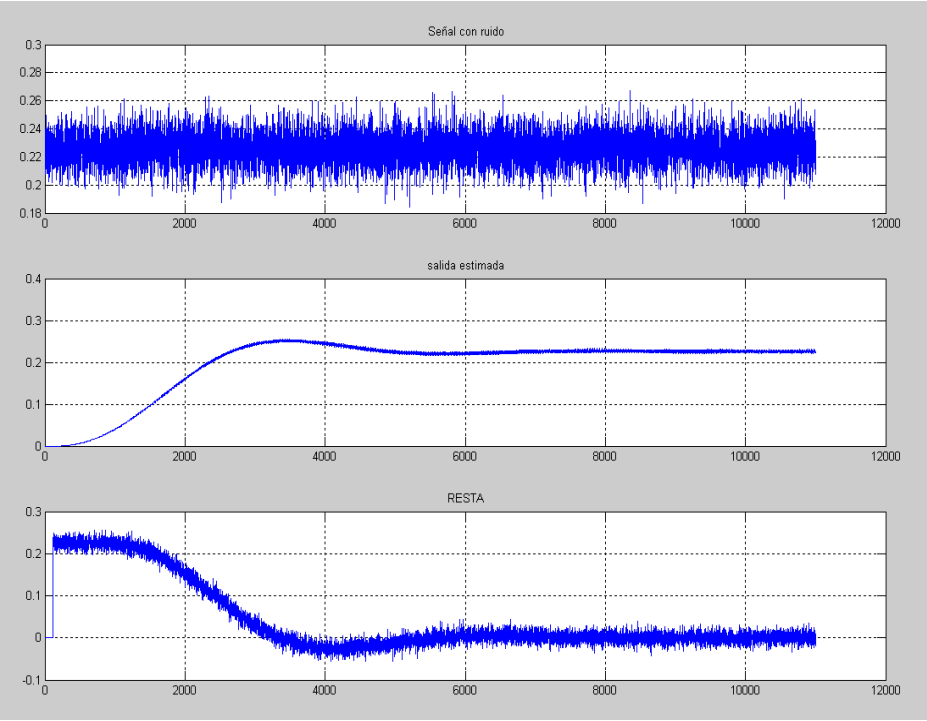


Figura 71. Graficas de la Señal ruidosa, señal estimada y resta de la señal ruidosa y estimada.

También hacemos referencia a las señales estimadas por el filtro para este caso como se muestra en la figura 72.

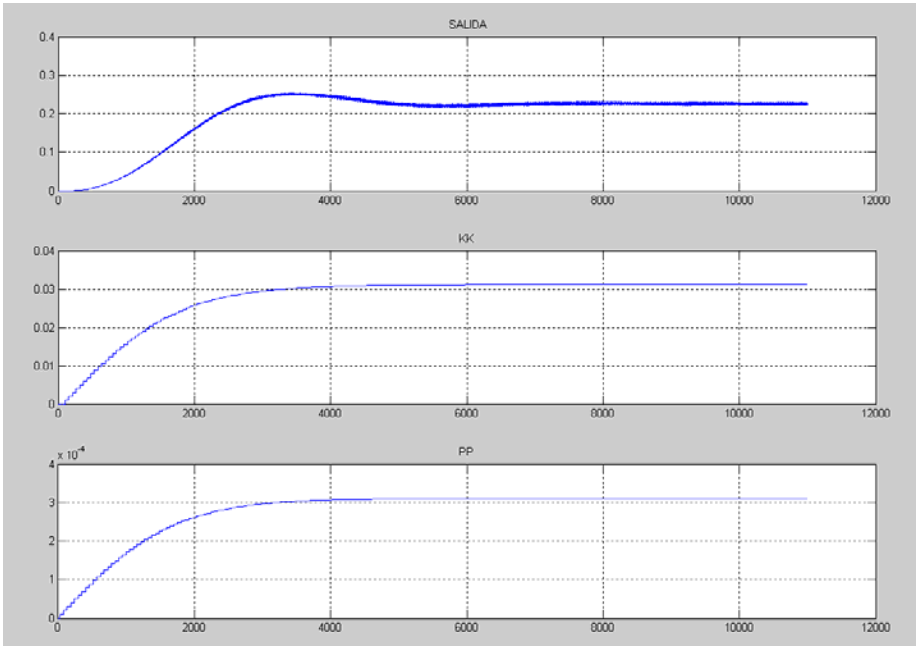


Figura. 72. Evolución de los parámetros del filtro de kalman para estimar la señal del sensor de temperatura.

En la siguiente figura 73, podemos visualizar la respuesta de la señal estimada por el filtro de kalman y la señal de entrada ruidosa de temperatura.

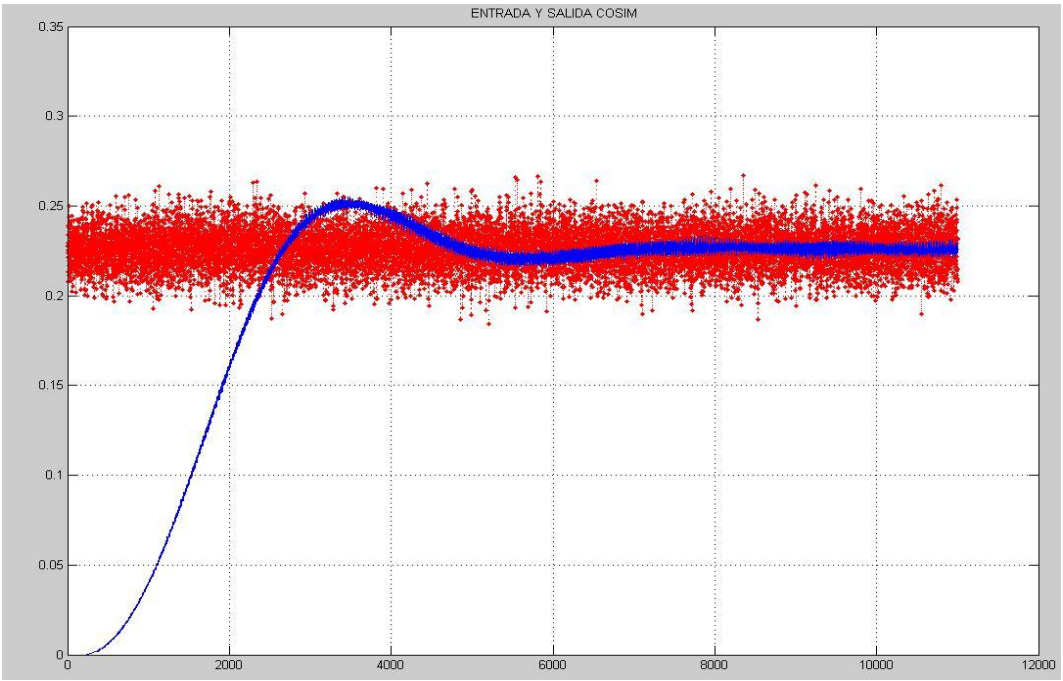


Figura 73. Señal ruidosa y señal estimada.

Para hacer una comparación entre la salida co-simulada o que resulta del filtraje, se toma la porción de la muestra 8000 a la muestra 11000, tal como se muestra en la figura 74.

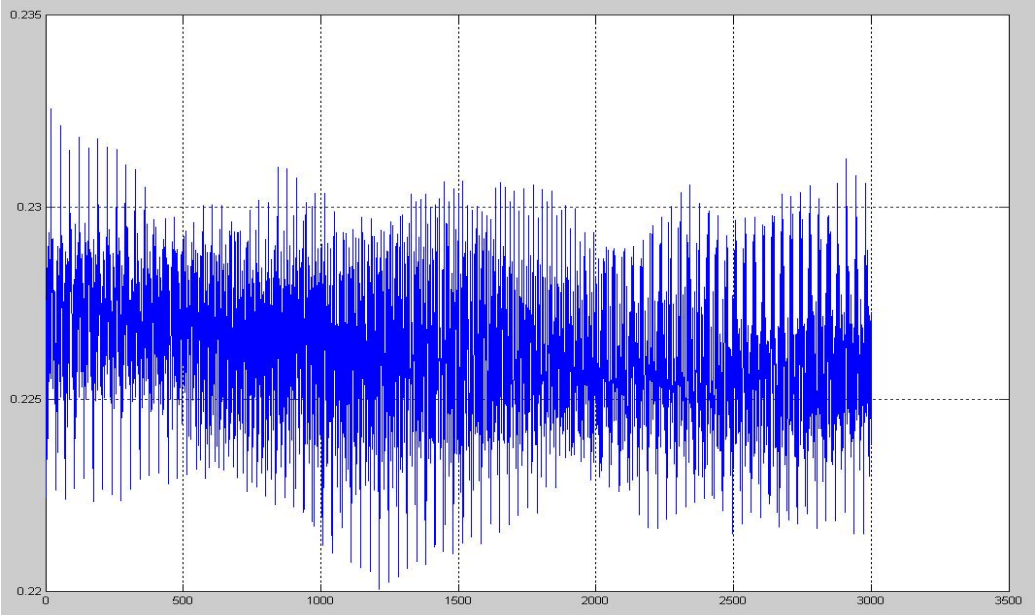


Figura 74. Muestras de la señal estimada.

En la figura 75, se muestra el histograma de la señal estimada, y se nota un grafico mas estrecho que el histograma de la señal contaminada con ruido.

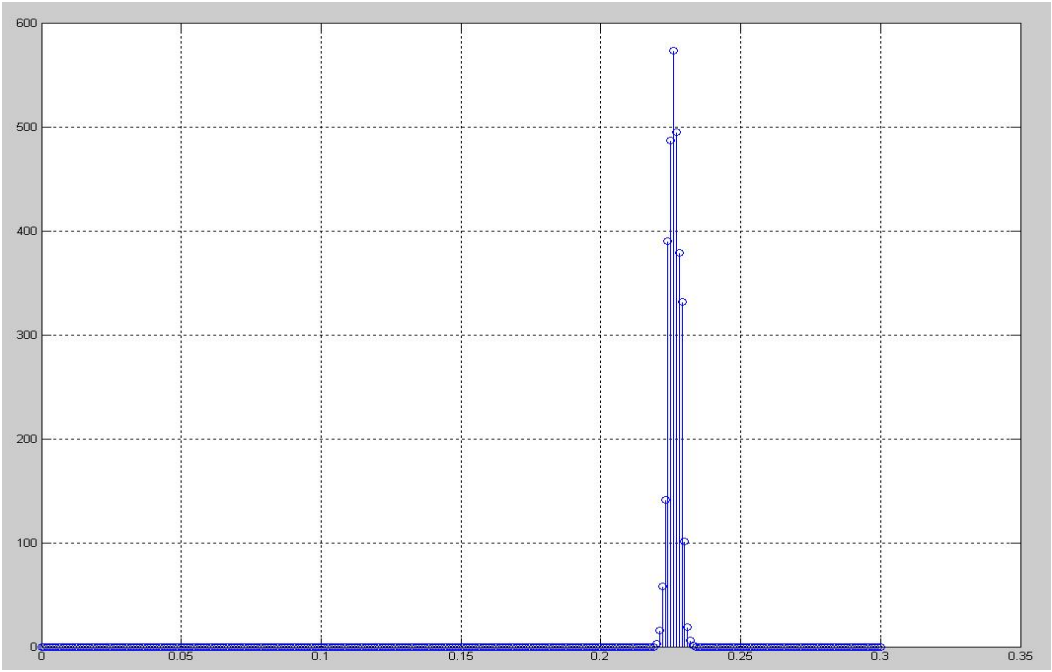


Figura 75. Histograma de la señal estimada.

También se procedió a calcular la media, varianza y desviación estándar de la señal estimada y estos son los siguientes:

Media = 0.2263.

Varianza = 3.9019e-006

Desviación estándar = 0.0020

En la figura 76, se muestra cuando el filtro no tiene un buen tunning con sus parámetros iniciales de ruido, como la covarianza de ruido de medición, lo cual da a lugar una distorsión en su salida.

Esto se debe que el filtro debe actualizar sus datos del ruido cada tramo de datos, para poder adaptarse mejor al sistema, con ello se garantiza una mejor performance del filtro discreto de kalman.

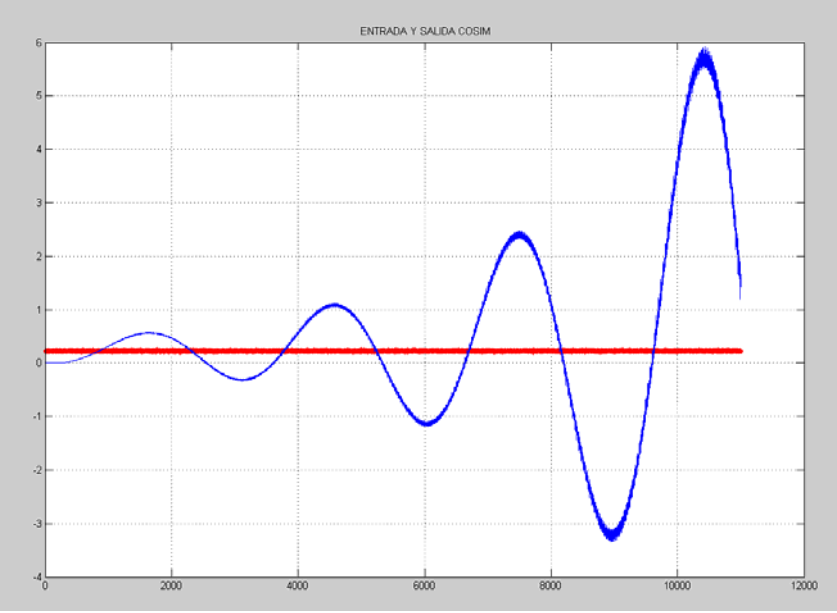


Figura 76. Salida no lineal debido al parámetro de covarianza del ruido de medición.

Debido a la respuesta mostrada, se pudo ver en la figura 73 que el ruido se reduce estadísticamente para la estimación de la señal de temperatura.

En la tabla 5, se muestran los resultados mas resaltantes del filtro discreto de kalman.

	Señal con ruido	**Señal estimada**	**Reducción %**
Media (u)	0.2261.	0.2263.	**0.08**
Varianza (σ^2)	1.2130e-004	3.9019e-006	**96.78**
Desviación estándar (σ)	0.0110.	0.0020	**81.81**

Tabla 5. Resultados de comparación.

CAPITULO 7. RESULTADOS Y CONCLUSIONES

En los siguientes párrafos se van a detallar los resultados que se tuvo en la implementación del filtro en el FPGA.

Se pudo describir el ruido total como una suma de ruidos que afectan a la medición de temperatura, con lo cual se pudo modelar el ruido total como un ruido blanco gaussiano, por medio del teorema del límite central.

Se implementó una tarjeta impresa para el circuito de acondicionamiento del sensor de temperatura, con la cual se caracterizó el sensor de temperatura en base al algoritmo de kalman.

Se diseñó una arquitectura que trabaje en forma paralela, para que sea mas rápida.

Se implementó el filtro de kalman en el FPGA virtex 2 pro de xilinx, teniendo en cuenta una frecuencia de 100Mhz.

El filtro de kalman ocupa un poco mas del 80% de lo recursos del FPGA virtex 2 pro.

Se probó que estima la señal con menor desviación estándar y varianza, con lo cual la señal se acerca mas a la señal deseada, pues se redujo la desviación estándar en mas del 80%.

CAPITULO 8. CONCLUSIONES

Las siguientes conclusiones se derivaron del trabajo, y estas son:

Se debe tener presente de contar con un laboratorio con un suministro principal, y que cuente con pocos equipos posibles para hacer las caracterizaciones de los sensores.

Se debe hacer un estudio previo de la señal que se va a evaluar, para poder sacar sus parámetros estadísticos.

El modelamiento matemático del ruido y el teorema del límite central nos sirve para tener en cuenta más ruidos infiltrados que puedan añadirse al ruido total.

Para poder mejorar en performance del filtro se debe tener en cuenta un porcentaje de área de mapeo de 80%.

Se debe tener en cuenta, los tiempos de procesamiento de las bloques del filtro, para utilizar tener en cuenta un delay necesario, con lo cual se va a poder hacer los cálculos con los verdaderos valores.

Se podría optimizar el filtro, usando parámetros de enrutamiento previos en el momento del mapeo del filtro en el FPGA, pero seria necesario tener en cuenta un chip con mayor capacidad de integración.

Para sistemas no lineales se puede pensar en el filtro de kalman extendido o los filtros alfa beta.

Se puede implementar hardware por medio de simulink en FPGA, con lo cual se puede probar los sistemas creados en matlab, para el complemento del aprendizaje del investigador.

ANEXOS

APENDICE A. La media y la varianza.

Media:

$$u(t) = \overline{X}$$
$$= E(X)$$
$$= \int_{-\infty}^{+\infty} xf(x)dx$$

Varianza :

$$\sigma^2 = Var(X)$$
$$= E[(X - E[X])^2]$$
$$= \int_{-\infty}^{+\infty} (x - \overline{X})^2 f(x)dx$$

APENDICE B. Propiedades de la covarianza.

Propiedades de la covarianza:

- Si X e Y son independientes y sin correlación o uno de ellos tiene un valor 0 de promedio entonces:

$$\sigma_{xy} = 0.$$

- Si x e y son ortogonales, entonces:

$$\sigma_{xy} = -(E[X]E[Y])$$

.

- Si la covarianza es simetrica entonces:

$$\text{cov}(X,Y) = \text{cov}(Y,X)$$

.

La identidad de la covarianza básica:

$$\text{cov}(X+Y,Z) = \text{cov}(X,Z) + \text{cov}(Y,Z)$$

- Covarianza de variables iguales:

$$\text{cov}(X,X) = Var(X)$$

APENDICE C. Códigos Utilizados en Matlab.

CODIGO GENERADOR DE RUIDO.

```
clear all
close all
clc
P=ones(8000);
tam=max(size(P));  %% me da el tamaño del vector
largo=0:tam-1;
ZETA=2*ones(size(largo-1));
n=0.1*randn(1,tam);
ZT=ZETA+n;
x = 0:0.01:3;
HISTO=HIST(ZT,x);
MEANN=MEAN(ZT)
VARR=VAR(ZT)
STDD=STD(ZT)
cov1=cov(ZT,largo,1)
cov2=cov(ZT,ZT,1)
cov3=cov(ZT,largo,0)
cov4=cov(ZT,ZT,0)
cov5=cov(ZT,n,0)
cov6=cov(n,n,0)
```

```
figure(1)
plot(ZT)
GRID
figure(2)
plot(x,HISTO)
GRID
figure(3)
stem(x,HISTO)
GRID
```

CODIGO GENERADOR DE SEÑAL RUIDOSA.

```
clear all
close all
clc
 P=ones(11000);
tam=max(size(P));  %% me da el tamaño del vector
largo=0:tam-1;
Q10=0.00001*ones(size(largo-1));
H10=1*ones(size(largo-1));
A10=1*ones(size(largo-1));
R10=0.01*ones(size(largo-1));
X10=0*ones(size(largo-1));
xm=0;

Q1=num2hex(Q10);
Q=hex2dec(Q1);

H1=num2hex(H10);
H=hex2dec(H1);

A1=num2hex(A10);
A=hex2dec(A1);

R1=num2hex(R10);
R=hex2dec(R1);
```

```
X1=num2hex(X10);
X=hex2dec(X1);

ZETA=0.226*ones(size(largo-1));

n=0.011*randn(1,tam);

ZT=ZETA+n;

NME=num2hex(ZT);
Z=hex2dec(NME);
```

APENDICE D. Histograma y PMF.

Para comprender más sobre el histograma, Imaginemos que usamos un convertidor de analógico a digital de 8 bits para ingresar datos a la computadora, y que adquirimos 256.000 muestras de una cierta señal. Como por ejemplo, en la siguiente figura D1, la cual muestra 128 muestras que pudieron ser una parte de una señal. El valor de cada muestra será una de 256 posibilidades, 0 a 255. El histograma exhibe el número de las muestras que están presentes en la señal que se agrupan por cantidad en los valores posibles

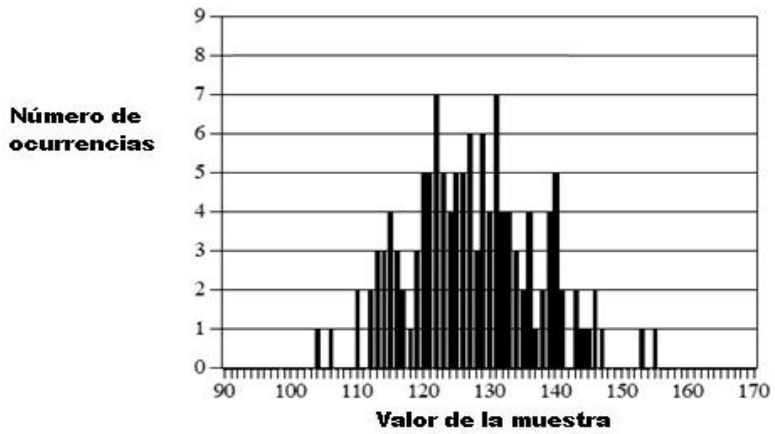


Figura D1. Señal con solo 128 muestras.

En esta figura D2, que es adaptada de [7], se muestra el histograma para las 256 mil muestras.

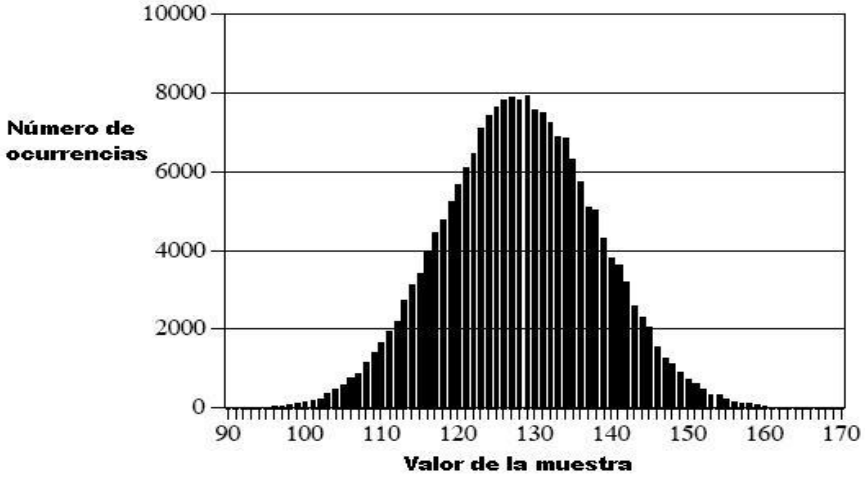


Figura D2. Señal con solo 256,000 muestras.

En la figura D1 hay 2 muestras que tienen un valor de 110, 7 muestras que tengan un valor de 131, etc. Representaremos el histograma por Hi, donde está un índice i que funciona a partir de la 0 a M-1, y M es el número de los valores posibles que cada muestra puede tomar. Por ejemplo, H50 es el número de las muestras que tienen un valor de 50.

De la forma que se define, la suma de todos los valores en el histograma debe ser igual al número de puntos de la señal:

$$N = \sum_{i=0}^{M-1} H_i$$

El histograma puede ser utilizado para calcular eficientemente la desviación estándar y la media de muchos conjuntos de datos grandes. Esto es especialmente importante para las imágenes, que pueden contener millones de muestras. El histograma agrupa a los que tienen el mismo valor.

Esto permite que la estadística sea calculada trabajando con algunos grupos, más bien que una gran cantidad de muestras de la señal. Usando este acercamiento, la desviación estándar es calculada del histograma por las ecuaciones:

$$u = \frac{1}{N} \sum_{i=0}^{M-1} iH_i$$

$$\sigma^2 = \frac{1}{N-1} \sum_{i=0}^{M-1} (i-u)^2 H_i$$

El cálculo del histograma es muy rápido, puesto que requiere solamente la indexación de direcciones y el incremento.

En la comparación, del cálculo de la desviación estándar y la media requiere las operaciones desperdiciadoras de tiempo de la adición y de la multiplicación.

La estrategia de este algoritmo es utilizar estas operaciones lentas solamente en los pocos números en el histograma, no las muchas muestras en la señal. Esto hace el algoritmo mucho más rápido que los métodos previamente descritos. Como por ejemplo tomar un número cada diez para las señales muy largas con los cálculos que son realizados en una computadora de fines generales. El histograma es lo que esta formado de una señal adquirida.

Por medio de Matlab, podemos tener el histograma de la señal aleatoria, como se muestra en la figura D3, ya estamos viendo que se trata de una señal del tipo acampanada, o guasiana.

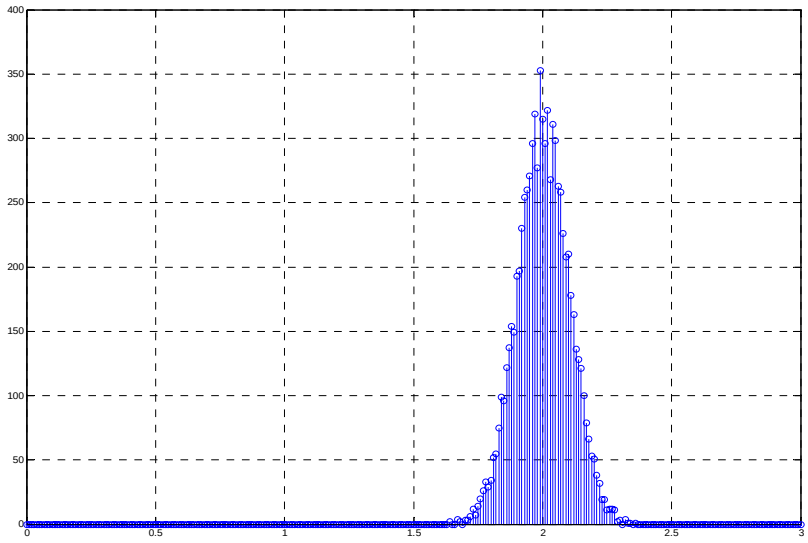


Figura D3. Histograma de una señal ruidosa.

Y Los parámetros de esta señal son los siguientes:

La media de esta señal es: 2.0025.

La varianza de esta señal es: 0.0101.

La desviación estándar es: 0.1004.

Con estos parámetros podemos tener en cuenta cuan desviado esta la señal contaminada de la media y podemos colocar estos parámetros en el filtro de kalman.

La función de la masa de la probabilidad (PMF).

El pmf es importante porque describe la probabilidad de cierto valor generada y se puede estimar del histograma.

En la figura D4, se muestra un pmf de ejemplo, y en la figura D5 uno de los histogramas posibles que se podrían asociar a él, estas 2 figuras son adaptadas de [7]. La llave a entender estos conceptos se reclina en las unidades del eje vertical. Según lo descrito previamente, el eje vertical del histograma es el número de veces que un valor particular ocurre o aparece en la señal. El eje vertical del pmf contiene la información similar, excepto expresado sobre una base fraccionaria. Es decir cada valor en el histograma es dividido por el número total de muestras para aproximar el pmf. Esto significa que cada valor en el pmf debe estar entre cero y uno, y que la suma de todos los valores en el pmf será igual a una.

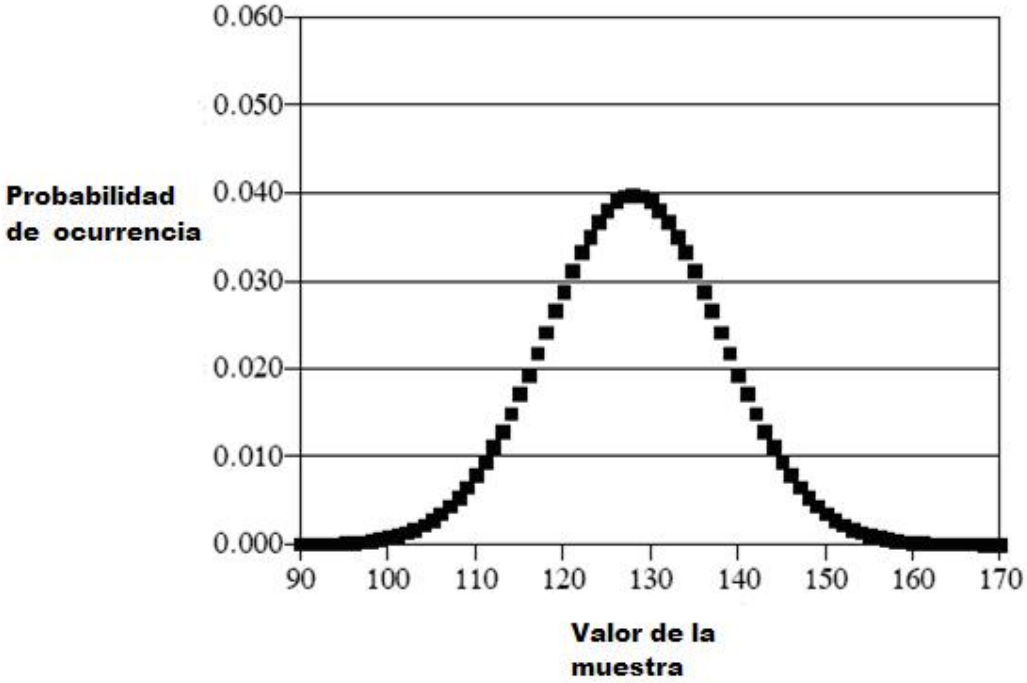


Figura D4. PMF de una señal aleatoria.

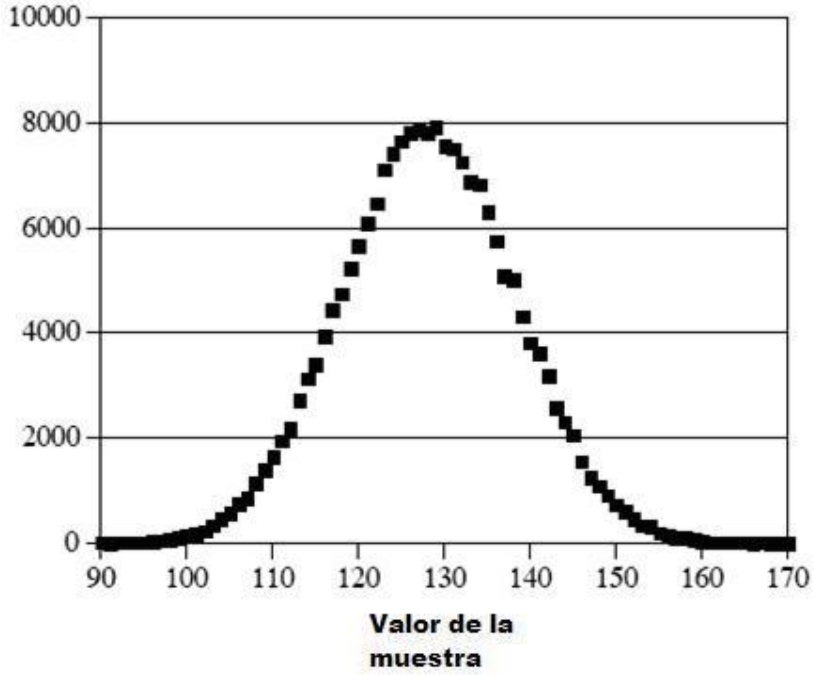


Figura D5. Histograma de una señal aleatoria.

El histograma y el pmf se pueden utilizar solamente con datos discretos, tales como una señal convertida a digital que reside en una computadora.

APENDICE E. Exactitud y precisión.

La precisión y la exactitud son los términos utilizados para describir los sistemas y métodos que miden, calculan, o predicen En todos estos casos, hay algunos parámetros que desean conocer el valor real. Esto se le llama el valor verdadero.

El método proporciona un valor medido, que quiere estar tan cerca del valor verdadero como sea posible. La precisión y la exactitud son maneras de describir el error que puede existir entre estos dos valores.

Como ejemplo, considere un oceanógrafo midiendo la profundidad del agua usando un sistema de sonar. Ráfagas cortas de sonido se transmiten de la nave, que se refleja de los fondos oceánicos, y recibida en la superficie como un eco.

Las ondas de sonido viajan a una velocidad relativamente constante en el agua, lo que permite que sea encontrada la profundidad del tiempo transcurrido entre la transmisión y recepción de pulsos. Al igual que con todas las mediciones empíricas, una cierta cantidad de error existe entre el valor medido y el valor verdadero.

Esta medida particular, puede verse afectada por muchos factores: ruido aleatorio en la electrónica, olas en la superficie del océano, las plantas que crecen en el fondo del océano, variaciones en la temperatura del agua, causando el cambio de la velocidad del sonido, etc.

Para investigar estos efectos, el oceanógrafo tiene muchas lecturas sucesivas en un lugar que se sabe que son exactamente 1000 metros de profundidad (el valor real). Estas mediciones son organizadas como el histograma se muestra en la siguiente figura, que es adaptada de [7]

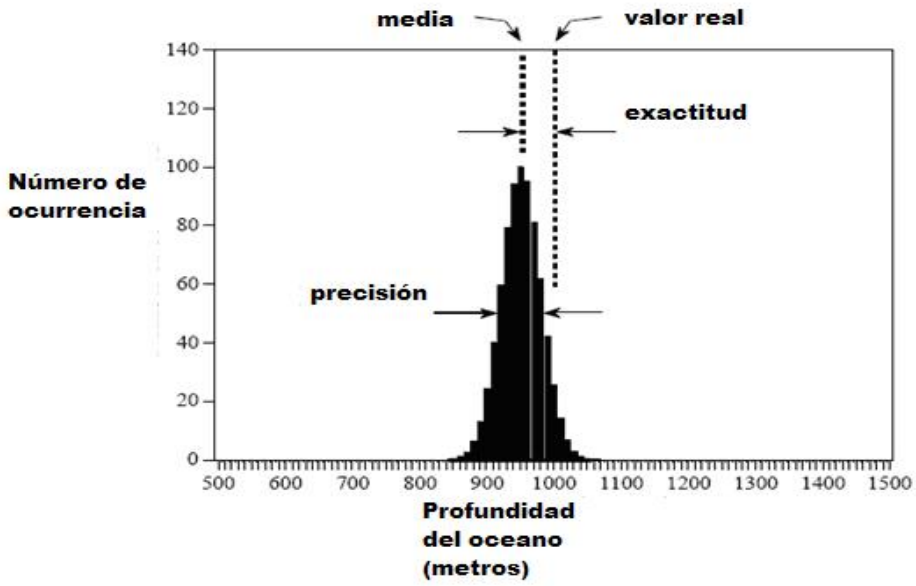


Figura E1. Histograma de la medicion.

Como pudiera esperarse de del teorema del límite Central, los datos adquiridos son normalmente distribuidos. La media se produce en el centro de la distribución, y representa la mejor estimación de la profundidad sobre la base de la totalidad de los datos medidos.

La desviación estándar se define como el ancho de la distribución, que describe que mucha variación se produce entre las sucesivas mediciones. Esta situación da lugar a dos tipos generales de error que el sistema pueda experimentar. En primer lugar, la media puede ser desplazada del valor verdadero.

La *cantidad o valor de este desplazamiento* se llama ***la exactitud*** de la medición.

En segundo lugar, las distintas mediciones pueden no estar de acuerdo unos con otros, tal como indica el *ancho de la distribución,* esto se llama la ***precisión*** de la medida, y es expresado citando a la desviación estándar, o la SNR.

Considere una medida que tenga buena exactitud, pero pobre precisión, el histograma está centrado sobre el verdadero valor, pero la distribución es muy amplia.

Aunque las mediciones son correctas, como un grupo, cada lectura individual es una pobre medida del valor real. Esta situación se dice que tiene pobre repetitividad; Las mediciones tomadas en sucesión no satisfacen. La pobre precisión resulta de los errores

aleatorios. Este es el nombre dado a los errores que cambian en el tiempo La medida cuando esta es repetida.

Promediar varias mediciones siempre va a mejorar la precisión. En resumen, la precisión es una medida del ruido aleatorio.

Ahora, imagine que es una medida muy precisa, pero tiene una pobre precisión. Este Hace que el histograma muy delgadas, pero no centrado sobre el verdadero valor.

Las sucesivas lecturas están cerca al valor, pero todos ellos tienen un gran error. La pobre exactitud resulta de los errores sistemáticos. Estos son los errores que vienen a ser repetidos en exactamente la misma manera, cada vez que se realiza la medición.

La precisión suele depender de la forma de cómo se calibra el sistema.

Por ejemplo, En la medición de la profundidad del océano, el parámetro que se mide directamente es el tiempo transcurrido. Esto se convierte en profundidad por un procedimiento de calibración que relaciona los milisegundos a metros.

Esto puede ser tan simple como multiplicar por una fija velocidad, o tan complicado como decenas de correcciones de segundo orden.

Promediar las mediciones individuales no hace nada para mejorar la precisión. En resumen, la exactitud es una medida de calibración.

En la práctica hay muchas formas de que la precisión y la exactitud pueden ser Entrelazados. Por ejemplo, imaginar la construcción de un amplificador electrónico con resistencias del 1%. Esta tolerancia indica que el valor de cada resistencia será dentro de 1% del valor declarado en una amplia gama de condiciones, como la temperatura, Humedad, la edad, etc. Este error en la resistencia producirá un correspondiente error en la ganancia del amplificador. ¿Este error es un problema de exactitud o Precisión?

La respuesta depende de la forma de cómo se toma las mediciones. Por ejemplo, Supongamos que queremos construir un amplificador y probar varias veces durante unos minutos.

El error de la ganancia permanece constante con cada prueba, y se puede concluir que el problema es la exactitud.

En comparación, supongamos que deseamos construir mil amplificadores. La ganancia del dispositivo a dispositivo va a fluctuar aleatoriamente, y el problema parece ser de precisión. Del mismo modo, cualquiera de estos amplificadores mostrará obtener fluctuaciones de ganancia en respuesta a la temperatura y cambios ambientales. Una vez más, el problema se llama precisión.

En el momento de decidir qué nombre usar para llamar al problema, hágase dos preguntas. La primera: ¿promediando las sucesivas lecturas van a proporcionar una mejor medición? Si es Sí, es el error de precisión, si es no, lo llaman la exactitud.

Segundo: ¿la calibración corrige el error? Si la respuesta es sí, lo llama exactitud, y si es no, lo llaman precisión. Esto tiene relación con la forma de cómo será calibrado el dispositivo, y con qué frecuencia se hará.

APENDICE F. Conceptos básicos de la termocupla.

Las termocuplas son baratas y resistentes, tienen razonablemente buena estabilidad a largo plazo. Debido a su pequeño tamaño, responde con rapidez y son buenas opciones donde una respuesta rápida es importante.

Funcionan en temperaturas criogénicas hasta temperaturas de Jet-engine (tobera de escape del motor) y dan una razonable linealidad y precisión.

Debido a que el número de electrones libres en un trozo de metal depende en tanto la temperatura y la composición del metal, dos piezas de distintos metales en contacto isotérmico y exhibirá una diferencia de potencial que esta en función de la temperatura, como se muestra en la Figura F1, la cual es adaptada de [10]. El resultado depende de la tensión las temperaturas, T1 y T2.

Desde que la termocupla es básicamente un componente de medida diferencial en lugar de un dispositivo de medición absoluta, una temperatura de referencia es necesaria para saber el voltaje de uno de los metales, entonces la temperatura del otro metal es inferida de la tensión de salida.

Las termocuplas son hechas de materiales especialmente seleccionados que han sido exhaustivamente caracterizados en términos de tensión frente a la temperatura.

En particular el punto de hielo o juntura fría es de 0 ° C y dando a lugar a un voltaje de 0V, y es utilizado para realizar las tablas estándar de las termocuplas.

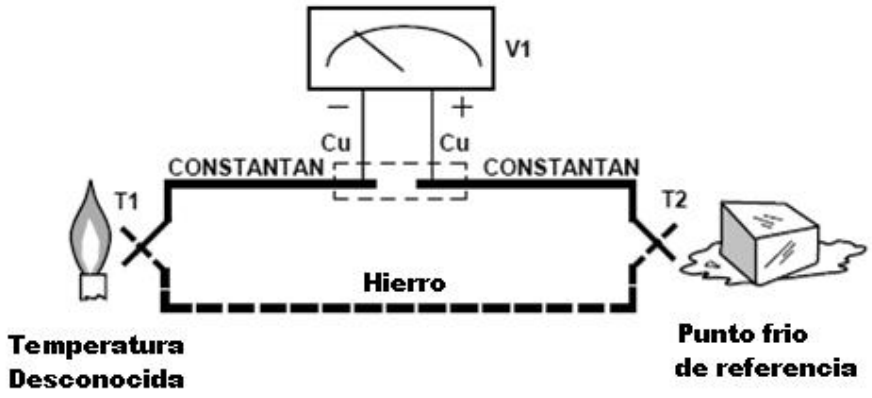


Figura F1. Voltaje de la termocupla con $0^{\circ}C$ de referencia.

Una alternativa técnica de medición, se ilustra en la figura F2, que es adaptado de [10] y se utiliza en la mayoría de las aplicaciones prácticas donde los requisitos de precisión no justifican el mantenimiento de normas primarias.

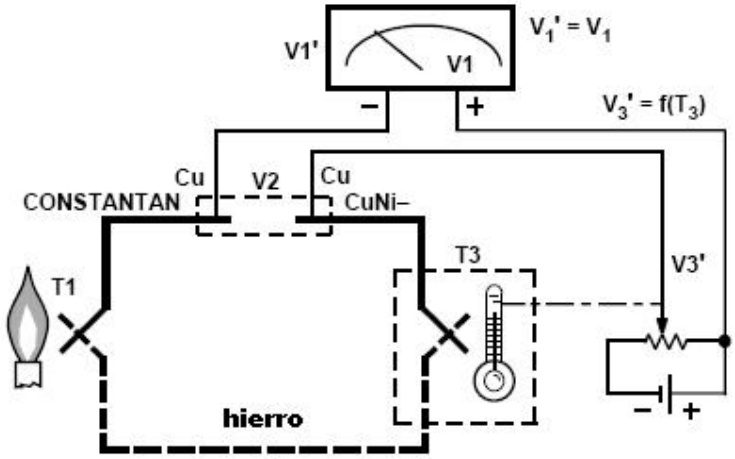


Figura F2. Alternativa técnica de medición.

A la referencia de temperatura se le permite cambiar con el medio ambiente del sistema de medición, pero es cuidadosamente medido por algún tipo de termómetro absoluto. Una medición del voltaje de la termocupla combinado con un conocimiento de la referencia de temperatura puede ser usada para calcular la medición de temperatura en la juntura.

Es una práctica habitual, sin embargo, es para utilizar un método termoeléctrico para medir la temperatura de referencia y para arreglar su voltaje de salida a fin de que corresponda a una termocupla referenciada a 0°C. Esta tensión es simplemente añadida al voltaje de la termocupla y la

suma corresponde con el voltaje estándar para una termocupla referenciada a un punto de hielo.

Por otro lado, la sensibilidad de temperatura de los circuitos de transistores integrados de silicio son bastantes predecibles y repetibles.

Esta sensibilidad es explotada en el AD594/AD595 para producir un voltaje de temperatura relacionado para compensar la referencia de "hielo" de una termocupla tal como se muestra en la figura F3, que es adaptado de [10].

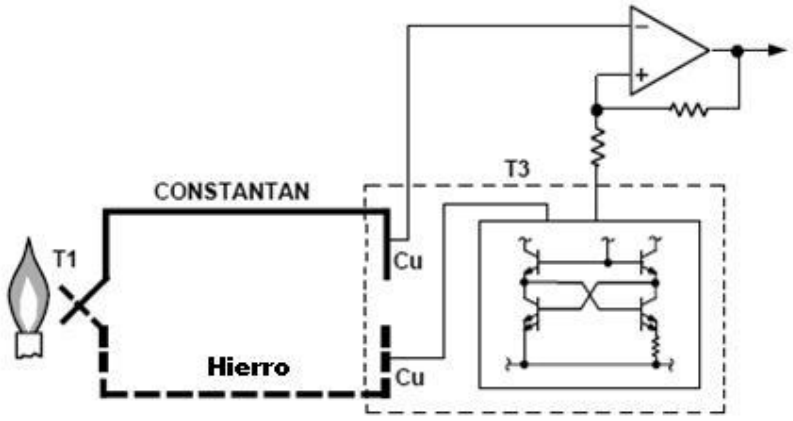


Figura F3. Conectando las junturas isotermales.

Dado que la compensación es en la referencia de juntura, es a menudo conveniente para formar la referencia de "juntura" conectar directamente al cableado del circuito. Por lo tanto, siempre y cuando estas conexiones y la compensación son a la misma temperatura no va a haber error en el resultado.

APENDICE G. Implementación de un filtro de kalman.

Para ver cómo funciona el filtro de kalman, se va a describir un ejemplo, en este caso, un simple sensor provee datos de una variable simple de la posición, pero la determinación de una posición escogida nos conlleva a la probabilidad de una posición

exacta, lo cual es un concepto familiar que rápidamente permite las dinámicas del sistema que pueden ser incorporadas al problema.

Supongamos que está perdido en el mar durante la noche y no tiene idea de su ubicación. Entonces toma una estrella brillante para establecer su posición (por la simplicidad, considerar una locación unidimensional). En algún tiempo t1 se determinas la locación que es z1. Sin embargo, porque el componente de medida presenta inexactitudes, el error humano, y otras cosas más, el resultado de la medida va a ser incierta. Entonces, decide que la precisión es como la desviación estándar σ_{z1} (o equivalentemente, la varianza o la estadística de segundo orden σ^2_{z1}).

Además puede establecer que la probabilidad condicional de x(t1), es decir la posición en el tiempo t1, condicionado en la observación en el valor observado de la medida z1 como se muestra en la figura G1. Es una grafica de $f_{x(t1)|z(t1)}(x \mid z1)$ como función de la ubicación x, la cual nos muestra la probabilidad de estar en cualquier ubicación, basado en la medidas que se tomaron.

Note que σ_{z1} es una medida directa de la incertidumbre: si es grande el σ_{z1}, entonces es más ancho el pico de probabilidad, con lo cual se esparce la probabilidad sobre el rango de valores de x. Para la densidad gaussiana, el 68.3% de la probabilidad está dentro de la banda de σ unidades en cada lado de la media, como se muestra en la figura G1, que es adaptado de [3].

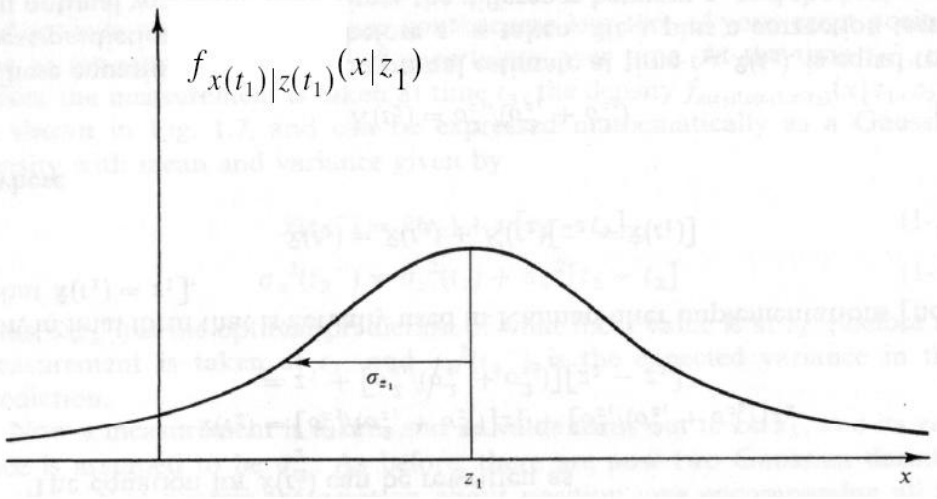


Figura G1. Densidad condicional de la posición basado en el valor medido Z_1.

Basado en la densidad de probabilidad condicional, la mejor estimación es:

$$\hat{x}(t1) = z1$$

Y la varianza del error en el estimado es:

$$\sigma^2{}_x(t1) = \sigma^2{}_{z1}$$

Notar que $\hat{X}$ es ambos la moda (pico) y la mediana (valor con la mitad del peso de la probabilidad en cada lado), como bien la media (centro de masa).

Ahora, un navegador entrenado toma valores independientes después de la medición hecha, en el tiempo $t2 \cong t1$ (entonces la verdadera posición no ha cambiado del todo), y obtiene una media z2 con una varianza de σ_{z2}. Este navegador tiene una habilidad mayor, asume que la varianza en esta medida es de algún modo más pequeña que el del primero. En la figura G2, que es adaptado de [3], se representa la densidad condicional de la posición en el tiempo t2, basado solo en el valor de medida z2. Se nota que el pico es angosto debido a la varianza pequeña, indicando que se esta mas cercano a la posición basado en la medida.

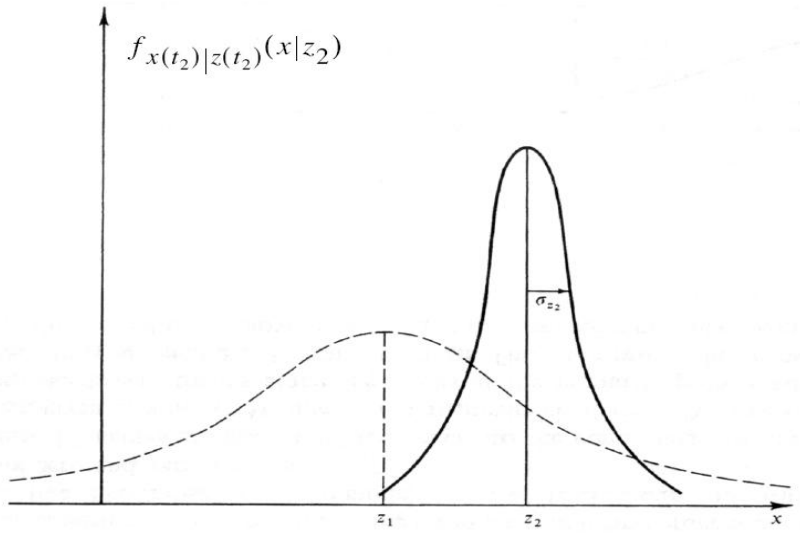


Figura G2. Segunda medición.

En este punto, se tienen 2 medidas disponibles para estimar la posición. El siguiente paso es combinar estos datos, entonces basado en las asumpciones hechas, en donde la densidad condicional de la posición en el tiempo $t2 \cong t1$, x(t2), dados z1 y z2, nos lleva a una densidad gaussiana con media u y una varianza σ como se muestra en la figura G3, adaptado de [3].

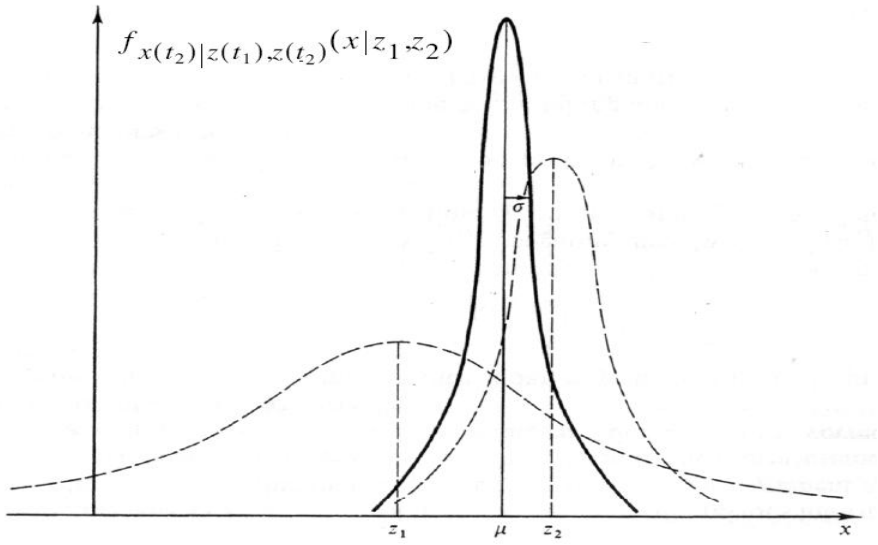


Figura G3. Densidad condicional basada en los datos de Z_1 y Z_2.

Con:

$$u = \left[\sigma_{z2}^2 / (\sigma_{z1}^2 + \sigma_{z2}^2)\right] z1 + \left[\sigma_{z1}^2 / (\sigma_{z1}^2 + \sigma_{z2}^2)\right] z2$$
$$1/\sigma^2 = (1/\sigma_{z1}^2) + (1/\sigma_{z2}^2)$$

Note que en la ecuación anterior, σ es menos que σ_{z1} o σ_{z2}, es decir que la incertidumbre en la posición estimada ha sido decrementada combinando las 2 medidas de información.

Dado esta densidad, el mejor estimado es:

$$\hat{x}(t2) = u$$

Con un error de varianza σ^2. Más aun, es la máxima vecindad estimada, y el estimado más cercano a la realidad.

Si la media u dada en una ecuación anterior se analiza un poco se puede notar lo siguiente, si σ_{z1} fuera igual a σ_{z2}, como si las medidas fueran de igual precisión, la ecuación muestra la posición optima estimada el cual es el simple promedio de las 2 medidas.

Por otro lado, si σ_{z1} fuera más larga que σ_{z2}, entonces la ecuación mostraría la incertidumbre envuelta en la medida z1 que es mayor que z2.

La ecuación para $\hat{x}(t2)$ puede ser escrita así:

$$\begin{aligned}\hat{x}(t2) &= \left[\sigma_{z2}^2/(\sigma_{z1}^2+\sigma_{z2}^2)\right]z1+\left[\sigma_{z1}^2/(\sigma_{z1}^2+\sigma_{z2}^2)\right]z2 \\ &= z1+\left[\sigma_{z1}^2/(\sigma_{z1}^2+\sigma_{z2}^2)\right]z2-z1]\end{aligned}$$

O en la forma final, la cual es actualmente usado en el filtro de kalman (notando que $\hat{x}(t1)=z1$).

$$\hat{x}(t2)=\hat{x}(t1)+K(t2)[z2-\hat{x}(t1)]$$

Donde

$$K(t2)=\left[\sigma_{z1}^2/(\sigma_{z1}^2+\sigma_{z2}^2)\right]$$

Estas ecuaciones nos muestran que la estimación optima en el tiempo $t2$, $\hat{x}(t2)$, es igual a la mejor predicción de su valor $\hat{x}(t1)$ antes que z2 sea tomado, además se añade un termino de corrección en el que esta involucrado la nueva constante de kalman y la diferencia entre z2 y la mejor predicción estimada $\hat{x}(t1)$. Es valioso comprender esta estructura "corrector- predictor" del filtro.

Basado en toda la información, una predicción del valor de las variables deseables y las medidas se van a actualizar en el momento después que se adquiera una medida. Cuando el valor de la medida es tomada, la diferencia entre esta y su predecida es usada para corregir la predicción de las variables deseables.

La ecuación de la varianza puede ser escrita como:

$$\sigma_x^2(t2) = \sigma_x^2(t1) - K(t2)\sigma_x^2(t1)$$

Notar que los valores de $\hat{x}(t2)$ y $\sigma_x^2(t2)$ expresan toda la información en $f_{x(t2)|z(t1),z(t2)}(x \mid z1, z2)$. Indicándolo de otra manera, por la propagación de esas 2 variables, la densidad condicional de la posición en el tiempo t2, dado z1 y z2 es completamente especificado.

APENDICE H: Encontrando la ganancia del filtro de kalman K.

Para empezar, vamos a definir los errores de nuestra estimación. Habrá dos errores, un error a priori, e_j^-, y un error a posteriori, e_j. Cada uno se define como la diferencia entre el valor real de x_j y la estimación (ya sea a priori o a posteriori).

$$e_k^- = x_k - \hat{x}_k^- \quad \text{..... (H.1).}$$

$$e_k = x_k - \hat{x}_k \quad \text{..... (H.2).}$$

Asociado con cada uno de estos errores es un error cuadrático medio, su varianza está dada por:

$$P_k^- = E\{(e_k^-)^2\} \quad \text{.....(H.3).}$$

$$P_k = E\{(e_k)^2\} \quad \text{..... (H.4).}$$

donde el operador $E\{\}$ representa la media, o promedio. Estas definiciones se utilizan para el cálculo de la cantidad k.

El filtro de Kalman minimiza la varianza a posteriori, P_j, eligiendo adecuadamente el valor de k. Empezamos por la sustitución en la ecuación E.2 en la ecuación E.4, se tiene lo siguiente:

$$P_k = E\{(x_k - \hat{x}_k)^2\} \quad \text{..... (H.5)}$$

$$P_k = E\{(x_k - \hat{x}_k^- + K(z_k - H\hat{x}_k^-))^2\} \quad \text{.....(H.6)}$$

Para encontrar el valor de k que minimiza la varianza diferenciamos esta expresión con respecto a k e igualando a cero.

$$\frac{\partial P_k}{\partial K} = \frac{\partial E\{(x_k - \hat{x}_k^- + K(z_k - H\hat{x}_k^-))^2\}}{\partial K} = 0$$

$$\Rightarrow \quad 0 = 2E\{(x_k - \hat{x}_k^- + K(z_k - H\hat{x}_k^-)(z_k - H\hat{x}_k^-)\}$$

$$0 = 2E\{x_k z_k - \hat{x}_k^- z_k + Kz_k^2 - KH\hat{x}_k^- z_k - Hx_k\hat{x}_k^- + H(\hat{x}_k^-)^2 - KHz_k\hat{x}_k^- + KH^2(\hat{x}_k^-)^2\} \quad \text{..... (H.7)}$$

Aprovechamos esta última expresión y la utilizamos para resolver k.

$$K = \frac{E\{x_k z_k - \hat{x}_k^- z_k - Hx_k\hat{x}_k^- + H(\hat{x}_k^-)^2\}}{E\{z_k^2 - 2H\hat{x}_k^- z_k + H^2(\hat{x}_k^-)^2\}} \quad \text{..... (H.8)}$$

Esta expresión es todavía bastante complicado. Para simplificar, vamos a considerar el numerador y el denominador por separado. Comenzamos con el numerador, de esta manera:

$$\begin{aligned} numerador &= E\{x_k z_k - \hat{x}_k^- z_k - Hx_k\hat{x}_k^- + H(\hat{x}_k^-)^2\} \\ &= E\{x_k(Hx_k + v_k) - \hat{x}_k^-(Hx_k + v_k) - Hx_k\hat{x}_k^- + H(\hat{x}_k^-)^2\} \\ &= E(H(x_k)^2 + x_k v_k - Hx_k\hat{x}_k^- - \hat{x}_k^- v_k - Hx_k\hat{x}_k^- + H(\hat{x}_k^-)^2\} \end{aligned}$$

$$E(H(x_k)^2 - 2Hx_k\hat{x}_k^- + H(\hat{x}_k^-)^2 + v_k(x_k - \hat{x}_k^-)\} \quad \text{.... (H.9)}$$

La medición de ruido, v, no esta en correlación ni con la entrada, ni la estimación a priori de x, por lo tanto:

$$E\{v_k(x_k - \hat{x}_k^-)\} = E\{v_k(e_k^-)\} = 0$$

Esto simplifica la expresión para el numerador, entonces:

$$\begin{aligned} numerador &= E\{H(x_k)^2 - 2Hx_k\hat{x}_k^- + H(\hat{x}_k^-)^2\} \\ &= HE\{(x_k)^2 - 2x_k\hat{x}_k^- + (\hat{x}_k^-)^2\} \\ &= HE\{(x_k - \hat{x}_k^-)^2\} = HE\{(e_k^-)^2\} \end{aligned}$$

$$HP_k^- \quad \text{…… (H.10)}$$

Ahora, de la misma manera, considerar el denominador.

$$\begin{aligned} Denom\,inador &= E\{z_k^2 - 2H\hat{x}_k^- z_k + H^2(\hat{x}_k^-)^2\} \\ &= E\{(Hx_k + v_k)^2 - 2H\hat{x}_k^-(Hx_k + v_k) + H^2(\hat{x}_k^-)^2\} \\ &= E\{H^2x_k^2 + 2Hx_kv_k + v_k^2 - 2H^2\hat{x}_k^-x_k - 2H\hat{x}_k^-v_k + H^2(\hat{x}_k^-)^2\} \end{aligned}$$

$$E\{H^2x_k^2 - 2H^2\hat{x}_k^-x_k + H^2(\hat{x}_k^-)^2 + v_k^2 + 2H(x_k - \hat{x}_k^-)v_k\} \quad \text{……. (H.11)}$$

Una vez más, podemos utilizar la condición de ortogonalidad, entonces para establecer el último termino a cero, por lo tanto:

$$\begin{aligned} Denom\,inador &= E\{H^2x_k^2 - 2H^2\hat{x}_k^-x_k + H^2(\hat{x}_k^-)^2 + v_k^2\} \\ &= E\{H^2x_k^2 - 2H^2\hat{x}_k^-x_k + H^2(\hat{x}_k^-)^2\} + E\{v_k^2\} \end{aligned}$$

$$H^2P_k^- + R \quad \text{…… (H.12)}$$

Utilizando la expresión para el numerador y denominador, por último, se consigue una simple expresión de k:

$$K = \frac{HP_k^-}{H^2 P_k^- + R} \quad \text{..... (H.12)}$$

APENDICE I. Números con punto flotante.

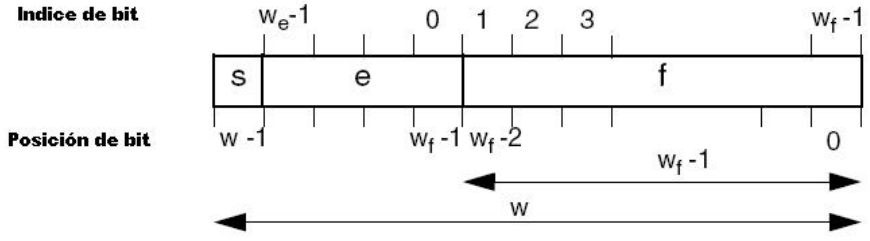
El CORE emplea la representación de punto flotante del estándar IEEE-754 para diversos tamaños de representación de números. Cuando los tamaños estándar son elegidos, el formato y valores especiales empleados son idénticos a la descrita por el estándar IEEE-754.

Dos parámetros han sido adoptado para propósitos de generalizar el formato procesado por el CORE de punto flotante.

Esto especifica el formato del ancho total y el ancho de la parte fraccionaria. Por estándar, el tipo de precisión simple tiene como formato de ancho 32 bits y la fracción es de 24 bits de ancho, el formato se muestra en la figura I1, que es adaptada de [11].

Un número de punto flotante se representa mediante un signo ('s'), un exponente ('E') y una fracción $b_0, b_1, b_2, b_3, ...b_{w_{f-1}}$.

El valor de un número de punto flotante está dado por: $(-1)^s 2^E b_0.b_1.b_2...b_{w_{f-1}}$.



I1. Formato de los números de punto flotante.

Como b_0 es una constante, sólo la parte fraccional se mantiene, es decir: $f = b_1.b_2.b_3....b_{w_{f-1}}$. Para ello se requiere sólo w_{f-1} bits.

De los restantes bits, un bit se emplea para representar el signo, y $w_e = w - w_f$ representan los bits del exponente.

En el campo del exponente, emplea un valor entero sin signo para la representación, cuyo valor está dado por:

$$e = \sum_{i=o}^{w_e - 1} e_i 2^i$$

El valor del exponente, E, se obtiene de esta manera, $E = e - 2^{w_e - 1}$. En realidad, w_f no es la longitud de la palabra de la fracción, sino la fracción con el bit escondido b_0.

Valores especiales

Una serie de valores, se han reservado para la representación de números especiales, tales como: No es un número (NaN), Infinity (∞), cero (0) y los números en general. Estos valores especiales se resumen en la Tabla 1, que es adaptada de [11].

Simbolos para valores especiales	s	e	f
NaN	No importa	$2^{w_e - 1}$ (e = 11...11))	Bit mas significativo de la fracción (f = 10...00)
$\pm\infty$	signo de ∞	$2^{w_e - 1}$ (e = 11...11)	cero
± 0	signo de 0	0	cero

Tabla I.1 Valores especiales.

Tenga en cuenta que en el cuadro 1 el bit de signo no está definido cuando el resultado es un NaN. Por otra parte, cero y el infinito tienen signo.

Siempre que sea posible, el signo se maneja de la misma manera como número finito.

Estándar IEEE-754

El CORE de Xilinx de punto flotante cumple con gran parte del estándar IEEE-754. Pero usa desviaciones para compensar en recursos.

El CORE de Xilinx de punto flotante soporta una gran variedad de longitud de fracción y exponente que los definidos en el estándar IEEE-754.

Formatos estándar comúnmente aplicadas por los procesadores programables:

- Formato Único o simple: utiliza 32 bits, con una fracción de 24 bits y 8 bits exponente.
- Formato doble: utiliza 64 bits y 53 bits de fracción y 11 bits de exponente.

Los formatos estándar menos usados son los siguientes:

- Extendida único o simple: tamaños de 43 bits y superiores.
- Extendida Doble: tamaños de 79 bits y superiores.

Actualmente, sólo el modo de redondeo "**redondea al mas cercano"**, tal como se define en el estándar IEEE-754, es soportado en este trabajo.

La especificación del estándar IEEE-754 requiere de señalización. Sin embargo, el CORE de Xilinx de punto flotante sólo admite NaNs. Cuando un NaN se presenta como uno de los operandos al CORE, el resultado va a ser un NaN.

APENDICE J: Translation report y Pads.

```
Release 8.1i ngdbuild I.24
Copyright (c) 1995-2005 Xilinx, Inc.  All rights reserved.

Command Line: ngdbuild -intstyle ise -dd _ngo -nt timestamp -i -p
xc2vp30-ff896-7 kalman_cosim5.ngc kalman_cosim5.ngd

Reading NGO file 'C:/xup/Copia de prueba_kalman3/netlist/kalman_cosim5.ngc' ...
Loading design module "C:\xup\Copia de
prueba_kalman3\netlist/division_flot64.ngc"...
Loading design module "C:\xup\Copia de
prueba_kalman3\netlist/multiplicacion_point_v1_0.ngc"...
Loading design module "C:\xup\Copia de
prueba_kalman3\netlist/resta_point_v1_0.ngc"...
Loading design module "C:\xup\Copia de
prueba_kalman3\netlist/suma_point_v1_0.ngc"...
```

Checking timing specifications ...
Checking expanded design ...

NGDBUILD Design Results Summary:
  Number of errors: 0
  Number of warnings: 0

Total memory usage is 210556 kilobytes

Writing NGD file "kalman_cosim5.ngd" ...

Writing NGDBUILD log file "kalman_cosim5.bld"...

PADS:

#Release 8.1i - par I.24
#Copyright (c) 1995-2005 Xilinx, Inc. All rights reserved.

#Sat May 24 11:51:16 2008

#
NOTE: This file is designed to be imported into a spreadsheet program
such as Microsoft Excel for viewing, printing and sorting. The |
character is used as the data field separator. This file is also designed
to support parsing.
#
#INPUT FILE: kalman_cosim5_map.ncd
#OUTPUT FILE: kalman_cosim5_pad.csv
#PART TYPE: xc2vp30
#SPEED GRADE: -7
#PACKAGE: ff896
#
Pinout by Pin Number:
#
-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,
Pin Number,Signal Name,Pin Usage,Pin Name,Direction,IO Standard,IO Bank Number,Drive (mA),Slew Rate,Termination,IOB Delay,Voltage,Constraint,DCI Value,IO Register,Signal Integrity,
A2,,,VCCAUX,,,,,,,,2.5,,,,,
A3,qin1<1>,IOB,IO_L01N_2/VRP_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
A4,,GTIPAD,RXNPAD9,UNUSED,,,,,,,,,,,
A5,,GTIPAD,RXPPAD9,UNUSED,,,,,,,,,,,
A6,,GTOPAD,TXPPAD9,UNUSED,,,,,,,,,,,
A7,,GTOPAD,TXNPAD9,UNUSED,,,,,,,,,,,
A8,xin1<29>,IOB,IO_L44N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
A9,,,GND,,,,,,,,,,,,
A10,ceroin4<23>,IOB,IO_L53_1/No_Pair,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
A11,,GTIPAD,RXNPAD7,UNUSED,,,,,,,,,,,
A12,,GTIPAD,RXPPAD7,UNUSED,,,,,,,,,,,
A13,,GTOPAD,TXPPAD7,UNUSED,,,,,,,,,,,
A14,,GTOPAD,TXNPAD7,UNUSED,,,,,,,,,,,
A15,,,VCCAUX,,,,,,,,2.5,,,,,
A16,,,VCCAUX,,,,,,,,2.5,,,,,
A17,,GTIPAD,RXNPAD6,UNUSED,,,,,,,,,,,
A18,,GTIPAD,RXPPAD6,UNUSED,,,,,,,,,,,
A19,,GTOPAD,TXPPAD6,UNUSED,,,,,,,,,,,

A20,,GTOPAD,TXNPAD6,UNUSED,,,,,,,,,,,
A21,salida1<9>,IOB,IO_L53_0/No_Pair,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
A22,,,GND,,,,,,,,,,,,
A23,zin1<62>,IOB,IO_L44P_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
A24,,GTIPAD,RXNPAD4,UNUSED,,,,,,,,,,,
A25,,GTIPAD,RXPPAD4,UNUSED,,,,,,,,,,,
A26,,GTOPAD,TXPPAD4,UNUSED,,,,,,,,,,,
A27,,GTOPAD,TXNPAD4,UNUSED,,,,,,,,,,,
A28,salida1<4>,IOB,IO_L01N_7/VRP_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
A29,,,VCCAUX,,,,,,,,2.5,,,,,
AA1,salida1<44>,IOB,IO_L51N_3/VREF_3,OUTPUT,LVCMOS25,3,12,SLOW,NONE**,,,,,NO,NONE,
AA2,,DIFFM,IO_L54P_3,UNUSED,,3,,,,,,,,,
AA3,rin1<32>,IOB,IO_L48N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AA4,rin1<44>,IOB,IO_L48P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AA5,rin1<45>,IOB,IO_L44N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AA6,xin1<44>,IOB,IO_L44P_3,INPUT,LVCMOS25,3,,,,NONE,,,,NO,NONE,
AA7,rin1<47>,IOB,IO_L38N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AA8,rin1<11>,IOB,IO_L38P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AA9,,,VCCO_3,,,3,,,,,2.50,,,,,
AA10,,,VCCO_4,,,4,,,,,2.50,,,,,
AA11,,,VCCO_4,,,4,,,,,2.50,,,,,
AA12,,,VCCO_4,,,4,,,,,2.50,,,,,
AA13,,,VCCO_4,,,4,,,,,2.50,,,,,
AA14,,,VCCO_4,,,4,,,,,2.50,,,,,
AA15,,,VCCO_4,,,4,,,,,2.50,,,,,
AA16,,,VCCO_5,,,5,,,,,2.50,,,,,
AA17,,,VCCO_5,,,5,,,,,2.50,,,,,
AA18,,,VCCO_5,,,5,,,,,2.50,,,,,
AA19,,,VCCO_5,,,5,,,,,2.50,,,,,
AA20,,,VCCO_5,,,5,,,,,2.50,,,,,
AA21,,,VCCO_5,,,5,,,,,2.50,,,,,
AA22,,,VCCO_6,,,6,,,,,2.50,,,,,
AA23,hin1<32>,IOB,IO_L38P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AA24,hin1<63>,IOB,IO_L38N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AA25,xin1<46>,IOB,IO_L44P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AA26,ceroin4<46>,IOB,IO_L44N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AA27,salida1<57>,IOB,IO_L48P_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
AA28,hin1<44>,IOB,IO_L48N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AA29,hin1<40>,IOB,IO_L54P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AA30,hin1<47>,IOB,IO_L51N_6/VREF_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AB1,rin1<46>,IOB,IO_L51P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AB2,rin1<35>,IOB,IO_L45N_3/VREF_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AB3,rin1<39>,IOB,IO_L46N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AB4,rin1<34>,IOB,IO_L46P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AB5,rin1<16>,IOB,IO_L40N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AB6,rin1<24>,IOB,IO_L40P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AB7,rin1<9>,IOB,IO_L32N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AB8,rin1<1>,IOB,IO_L32P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AB9,,,VCCO_3,,,3,,,,,2.50,,,,,
AB10,,,VCCO_4,,,4,,,,,2.50,,,,,
AB11,,,VCCO_4,,,4,,,,,2.50,,,,,
AB12,,,VCCO_4,,,4,,,,,2.50,,,,,
AB13,,,VCCO_4,,,4,,,,,2.50,,,,,
AB14,ain2<16>,IOB,IO_L49N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AB15,ain2<24>,IOB,IO_L67N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AB16,ain2<41>,IOB,IO_L67P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AB17,ain2<51>,IOB,IO_L49P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AB18,,,VCCO_5,,,5,,,,,2.50,,,,,

AB19,,,VCCO_5,,,5,,,,,2.50,,,,,
AB20,,,VCCO_5,,,5,,,,,2.50,,,,,
AB21,,,VCCO_5,,,5,,,,,2.50,,,,,
AB22,,,VCCO_6,,,6,,,,,2.50,,,,,
AB23,hin1<21>,IOB,IO_L32P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AB24,hin1<22>,IOB,IO_L32N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AB25,ceroin4<45>,IOB,IO_L40P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AB26,xin1<45>,IOB,IO_L40N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AB27,hin1<50>,IOB,IO_L46P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AB28,hin1<41>,IOB,IO_L46N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AB29,xin1<36>,IOB,IO_L45N_6/VREF_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AB30,xin1<55>,IOB,IO_L51P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AC1,,,GND,,,,,,,,,,,,
AC2,rin1<36>,IOB,IO_L45P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AC3,rin1<31>,IOB,IO_L43N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AC4,rin1<33>,IOB,IO_L43P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AC5,rin1<30>,IOB,IO_L35N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AC6,rin1<2>,IOB,IO_L35P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AC7,,,CCLK,,,,,,,,,,,,
AC8,,,DONE,,,,,,,,,,,,
AC9,,DIFFS,IO_L02N_4/D0,UNUSED,,4,,,,,,,,,
AC10,rin1<55>,IOB,IO_L07N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AC11,rin1<62>,IOB,IO_L37N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AC12,rin1<57>,IOB,IO_L43N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AC13,ain2<15>,IOB,IO_L46N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AC14,ain2<8>,IOB,IO_L49P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AC15,ain2<1>,IOB,IO_L67P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AC16,ain2<40>,IOB,IO_L67N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AC17,ain2<50>,IOB,IO_L49N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AC18,zin1<35>,IOB,IO_L46P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AC19,zin1<34>,IOB,IO_L43P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AC20,ain2<53>,IOB,IO_L37P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AC21,ain2<56>,IOB,IO_L07P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AC22,,DIFFM,IO_L02P_5/D7,UNUSED,,5,,,,,,,,,
AC23,,,M2,,,,,,,,,,,,
AC24,,,M1,,,,,,,,,,,,
AC25,hin1<23>,IOB,IO_L35P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AC26,hin1<29>,IOB,IO_L35N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AC27,salida1<34>,IOB,IO_L43P_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
AC28,hin1<43>,IOB,IO_L43N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AC29,ceroin4<36>,IOB,IO_L45P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AC30,,,GND,,,,,,,,,,,,
AD1,rin1<28>,IOB,IO_L42N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AD2,rin1<43>,IOB,IO_L42P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AD3,rin1<48>,IOB,IO_L37N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AD4,rin1<49>,IOB,IO_L37P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AD5,,DIFFS,IO_L05N_3,UNUSED,,3,,,,,,,,,
AD6,rin1<6>,IOB,IO_L05P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AD7,,,PWRDWN_B,,,,,,,,,,,,
AD8,rin1<13>,IOB,IO_L05_4/No_Pair,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AD9,,DIFFM,IO_L02P_4/D1,UNUSED,,4,,,,,,,,,
AD10,rin1<19>,IOB,IO_L07P_4/VREF_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AD11,rin1<58>,IOB,IO_L37P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AD12,rin1<60>,IOB,IO_L43P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AD13,ain2<13>,IOB,IO_L46P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AD14,ain2<30>,IOB,IO_L68N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AD15,ain2<27>,IOB,IO_L73N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AD16,ain2<39>,IOB,IO_L73P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,

AD17,ain2<48>,IOB,IO_L68P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AD18,zin1<37>,IOB,IO_L46N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AD19,zin1<44>,IOB,IO_L43N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AD20,ain2<62>,IOB,IO_L37N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AD21,hin1<54>,IOB,IO_L07N_5/VREF_5,INPUT,LVCMOS25,5,,,,NONE,,,,NO,NONE,
AD22,,DIFFS,IO_L02N_5/D6,UNUSED,,5,,,,,,,,,,
AD23,consta1,IOB,IO_L05_5/No_Pair,INPUT,LVCMOS25,5,,,,NONE,,,,NO,NONE,
AD24,,,M0,,,,,,,,,,,,
AD25,,DIFFM,IO_L05P_6,UNUSED,,6,,,,,,,,,,
AD26,,DIFFS,IO_L05N_6,UNUSED,,6,,,,,,,,,,
AD27,hin1<31>,IOB,IO_L37P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AD28,ceroin4<63>,IOB,IO_L37N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AD29,hin1<37>,IOB,IO_L42P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AD30,hin1<39>,IOB,IO_L42N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AE1,rin1<29>,IOB,IO_L39N_3/VREF_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AE2,rin1<21>,IOB,IO_L39P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AE3,rin1<15>,IOB,IO_L33N_3/VREF_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AE4,rin1<8>,IOB,IO_L33P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AE5,rin1<14>,IOB,IO_L31N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AE6,,,GND,,,,,,,,,,,,
AE7,rin1<17>,IOB,IO_L08N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AE8,rin1<10>,IOB,IO_L08P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AE9,rin1<54>,IOB,IO_L44N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AE10,rin1<61>,IOB,IO_L44P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AE11,ain2<6>,IOB,IO_L47N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AE12,ain2<5>,IOB,IO_L47P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AE13,,,GND,,,,,,,,,,,,
AE14,ain2<19>,IOB,IO_L68P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AE15,ain2<20>,IOB,IO_L73P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AE16,ain2<21>,IOB,IO_L73N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AE17,ain2<23>,IOB,IO_L68N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AE18,,,GND,,,,,,,,,,,,
AE19,zin1<45>,IOB,IO_L47N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AE20,zin1<43>,IOB,IO_L47P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AE21,ain2<61>,IOB,IO_L44N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AE22,ain2<60>,IOB,IO_L44P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AE23,salida1<60>,IOB,IO_L08N_5,OUTPUT,LVCMOS25,5,12,SLOW,NONE**,,,,,NO,NONE,
AE24,hin1<55>,IOB,IO_L08P_5,INPUT,LVCMOS25,5,,,,NONE,,,,NO,NONE,
AE25,,,GND,,,,,,,,,,,,
AE26,hin1<19>,IOB,IO_L31N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AE27,hin1<26>,IOB,IO_L33P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AE28,hin1<25>,IOB,IO_L33N_6/VREF_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AE29,hin1<34>,IOB,IO_L39P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AE30,hin1<35>,IOB,IO_L39N_6/VREF_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AF1,rin1<3>,IOB,IO_L36N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AF2,rin1<40>,IOB,IO_L36P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AF3,rin1<27>,IOB,IO_L34N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AF4,rin1<26>,IOB,IO_L34P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AF5,,,GND,,,,,,,,,,,,
AF6,rin1<0>,IOB,IO_L31P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AF7,,DIFFM,IO_L01P_4/INIT_B,UNUSED,,4,,,,,,,,,,
AF8,rin1<59>,IOB,IO_L38N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AF9,rin1<53>,IOB,IO_L38P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AF10,rin1<52>,IOB,IO_L39N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AF11,ain2<4>,IOB,IO_L50_4/No_Pair,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AF12,ain2<3>,IOB,IO_L56N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AF13,ain2<12>,IOB,IO_L56P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AF14,ain2<29>,IOB,IO_L69N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,

AF15,ain2<47>,IOB,IO_L74N_4/GCLK3S,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AF16,ain2<37>,IOB,IO_L74P_5/GCLK4P,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AF17,ain2<28>,IOB,IO_L69P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AF18,ain2<49>,IOB,IO_L56N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AF19,ain2<32>,IOB,IO_L56P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AF20,ain2<33>,IOB,IO_L50_5/No_Pair,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AF21,ain2<57>,IOB,IO_L39P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AF22,ain2<59>,IOB,IO_L38N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AF23,ain2<54>,IOB,IO_L38P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AF24,,DIFFS,IO_L01N_5/RDWR_B,UNUSED,,5,,,,,,,,,
AF25,hin1<20>,IOB,IO_L31P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AF26,,,GND,,,,,,,,,,,,
AF27,hin1<27>,IOB,IO_L34P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AF28,hin1<24>,IOB,IO_L34N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AF29,xin1<60>,IOB,IO_L36P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AF30,hin1<28>,IOB,IO_L36N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
AG1,rin1<7>,IOB,IO_L06N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AG2,rin1<25>,IOB,IO_L06P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AG3,rin1<20>,IOB,IO_L04N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
AG4,,,GND,,,,,,,,,,,,
AG5,,DIFFS,IO_L02N_3,UNUSED,,3,,,,,,,,,
AG6,,DIFFS,IO_L01N_4/DOUT,UNUSED,,4,,,,,,,,,
AG7,,DIFFS,IO_L03N_4/D2,UNUSED,,4,,,,,,,,,
AG8,rin1<18>,IOB,IO_L06N_4/VRP_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AG9,,,GND,,,,,,,,,,,,
AG10,rin1<56>,IOB,IO_L39P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AG11,ain2<14>,IOB,IO_L53_4/No_Pair,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AG12,,,GND,,,,,,,,,,,,
AG13,ain2<2>,IOB,IO_L57N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AG14,ain2<25>,IOB,IO_L69P_4/VREF_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AG15,ain2<26>,IOB,IO_L74P_4/GCLK2P,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AG16,ain2<45>,IOB,IO_L74N_5/GCLK5S,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AG17,ain2<38>,IOB,IO_L69N_5/VREF_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AG18,ain2<31>,IOB,IO_L57P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AG19,,,GND,,,,,,,,,,,,
AG20,ain2<34>,IOB,IO_L53_5/No_Pair,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AG21,ain2<55>,IOB,IO_L39N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AG22,,,GND,,,,,,,,,,,,
AG23,,DIFFM,IO_L06P_5/VRN_5,UNUSED,,5,,,,,,,,,
AG24,,DIFFM,IO_L03P_5/D5,UNUSED,,5,,,,,,,,,
AG25,,DIFFM,IO_L01P_5/CS_B,UNUSED,,5,,,,,,,,,
AG26,,DIFFS,IO_L02N_6,UNUSED,,6,,,,,,,,,
AG27,,,GND,,,,,,,,,,,,
AG28,,DIFFS,IO_L04N_6,UNUSED,,6,,,,,,,,,
AG29,hin1<57>,IOB,IO_L06P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AG30,hin1<56>,IOB,IO_L06N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AH1,,DIFFS,IO_L03N_3/VREF_3,UNUSED,,3,,,,,,,,,
AH2,,DIFFM,IO_L03P_3,UNUSED,,3,,,,,,,,,
AH3,,,GND,,,,,,,,,,,,
AH4,,DIFFM,IO_L04P_3,UNUSED,,3,,,,,,,,,
AH5,,DIFFM,IO_L02P_3,UNUSED,,3,,,,,,,,,
AH6,,,GNDA16,,,,,,,,,,,,
AH7,,DIFFM,IO_L03P_4/D3,UNUSED,,4,,,,,,,,,
AH8,rin1<23>,IOB,IO_L06P_4/VRN_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AH9,ain2<18>,IOB,IO_L45N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AH10,ain2<10>,IOB,IO_L48N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AH11,ain2<11>,IOB,IO_L48P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AH12,,,GNDA18,,,,,,,,,,,,

AH13,ain2<0>,IOB,IO_L57P_4/VREF_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AH14,,,GND,,,,,,,,,,,,
AH15,ain2<36>,IOB,IO_L75N_4/GCLK1S,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AH16,ain2<44>,IOB,IO_L75P_5/GCLK6P,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AH17,,,GND,,,,,,,,,,,,
AH18,ain2<22>,IOB,IO_L57N_5/VREF_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AH19,,,GNDA19,,,,,,,,,,,,
AH20,ain2<63>,IOB,IO_L48N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AH21,rin1<63>,IOB,IO_L48P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AH22,zin1<33>,IOB,IO_L45P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AH23,,DIFFS,IO_L06N_5/VRP_5,UNUSED,,5,,,,,,,,,
AH24,,DIFFS,IO_L03N_5/D4,UNUSED,,5,,,,,,,,,
AH25,,,GNDA21,,,,,,,,,,,,
AH26,,DIFFM,IO_L02P_6,UNUSED,,6,,,,,,,,,
AH27,,DIFFM,IO_L04P_6,UNUSED,,6,,,,,,,,,
AH28,,,GND,,,,,,,,,,,,
AH29,,DIFFM,IO_L03P_6,UNUSED,,6,,,,,,,,,
AH30,,DIFFS,IO_L03N_6/VREF_6,UNUSED,,6,,,,,,,,,
AJ1,,,VCCAUX,,,,,,,,2.5,,,,,
AJ2,,,GND,,,,,,,,,,,,
AJ3,,DIFFS,IO_L01N_3/VRP_3,UNUSED,,3,,,,,,,,,
AJ4,,,AVCCAUXRX16,,,,,,,,2.5,,,,,
AJ5,,,VTRXPAD16,,,,,,,,,,,,
AJ6,,,AVCCAUXTX16,,,,,,,,2.5,,,,,
AJ7,,,VTTXPAD16,,,,,,,,,,,,
AJ8,rin1<5>,IOB,IO_L09N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AJ9,ain2<17>,IOB,IO_L45P_4/VREF_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AJ10,ain2<7>,IOB,IO_L54N_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AJ11,,,AVCCAUXRX18,,,,,,,,2.5,,,,,
AJ12,,,VTRXPAD18,,,,,,,,,,,,
AJ13,,,AVCCAUXTX18,,,,,,,,2.5,,,,,
AJ14,,,VTTXPAD18,,,,,,,,,,,,
AJ15,ain2<43>,IOB,IO_L75P_4/GCLK0P,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AJ16,ain2<46>,IOB,IO_L75N_5/GCLK7S,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AJ17,,,AVCCAUXRX19,,,,,,,,2.5,,,,,
AJ18,,,VTRXPAD19,,,,,,,,,,,,
AJ19,,,AVCCAUXTX19,,,,,,,,2.5,,,,,
AJ20,,,VTTXPAD19,,,,,,,,,,,,
AJ21,ain2<42>,IOB,IO_L54P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AJ22,zin1<42>,IOB,IO_L45N_5/VREF_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AJ23,ain2<52>,IOB,IO_L09P_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AJ24,,,AVCCAUXRX21,,,,,,,,2.5,,,,,
AJ25,,,VTRXPAD21,,,,,,,,,,,,
AJ26,,,AVCCAUXTX21,,,,,,,,2.5,,,,,
AJ27,,,VTTXPAD21,,,,,,,,,,,,
AJ28,,DIFFS,IO_L01N_6/VRP_6,UNUSED,,6,,,,,,,,,
AJ29,,,GND,,,,,,,,,,,,
AJ30,,,VCCAUX,,,,,,,,2.5,,,,,
AK2,,,VCCAUX,,,,,,,,2.5,,,,,
AK3,,DIFFM,IO_L01P_3/VRN_3,UNUSED,,3,,,,,,,,,
AK4,,GTIPAD,RXNPAD16,UNUSED,,,,,,,,,,,
AK5,,GTIPAD,RXPPAD16,UNUSED,,,,,,,,,,,
AK6,,GTOPAD,TXPPAD16,UNUSED,,,,,,,,,,,
AK7,,GTOPAD,TXNPAD16,UNUSED,,,,,,,,,,,
AK8,rin1<4>,IOB,IO_L09P_4/VREF_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AK9,,,GND,,,,,,,,,,,,
AK10,ain2<9>,IOB,IO_L54P_4,INPUT,LVCMOS25,4,,,,IFD,,,,YES,NONE,
AK11,,GTIPAD,RXNPAD18,UNUSED,,,,,,,,,,,

AK12,,GTIPAD,RXPPAD18,UNUSED,,,,,,,,,,,
AK13,,GTOPAD,TXPPAD18,UNUSED,,,,,,,,,,,
AK14,,GTOPAD,TXNPAD18,UNUSED,,,,,,,,,,,
AK15,,,VCCAUX,,,,,,,,2.5,,,,,
AK16,,,VCCAUX,,,,,,,,2.5,,,,,
AK17,,GTIPAD,RXNPAD19,UNUSED,,,,,,,,,,,
AK18,,GTIPAD,RXPPAD19,UNUSED,,,,,,,,,,,
AK19,,GTOPAD,TXPPAD19,UNUSED,,,,,,,,,,,
AK20,,GTOPAD,TXNPAD19,UNUSED,,,,,,,,,,,
AK21,ain2<35>,IOB,IO_L54N_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AK22,,,GND,,,,,,,,,,,,
AK23,ain2<58>,IOB,IO_L09N_5/VREF_5,INPUT,LVCMOS25,5,,,,IFD,,,,YES,NONE,
AK24,,GTIPAD,RXNPAD21,UNUSED,,,,,,,,,,,
AK25,,GTIPAD,RXPPAD21,UNUSED,,,,,,,,,,,
AK26,,GTOPAD,TXPPAD21,UNUSED,,,,,,,,,,,
AK27,,GTOPAD,TXNPAD21,UNUSED,,,,,,,,,,,
AK28,ce_1,IOB,IO_L01P_6/VRN_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
AK29,,,VCCAUX,,,,,,,,2.5,,,,,
B1,,,VCCAUX,,,,,,,,2.5,,,,,
B2,,,GND,,,,,,,,,,,,
B3,qin1<25>,IOB,IO_L01P_2/VRN_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
B4,,,AVCCAUXRX9,,,,,,,,2.5,,,,,
B5,,,VTRXPAD9,,,,,,,,,,,,
B6,,,AVCCAUXTX9,,,,,,,,2.5,,,,,
B7,,,VTTXPAD9,,,,,,,,,,,,
B8,qin1<54>,IOB,IO_L44P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
B9,ceroin4<11>,IOB,IO_L47N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
B10,ceroin4<8>,IOB,IO_L50_1/No_Pair,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
B11,,,AVCCAUXRX7,,,,,,,,2.5,,,,,
B12,,,VTRXPAD7,,,,,,,,,,,,
B13,,,AVCCAUXTX7,,,,,,,,2.5,,,,,
B14,,,VTTXPAD7,,,,,,,,,,,,
B15,clk_1,IOB,IO_L74N_1/GCLK1P,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
B16,ceroin4<42>,IOB,IO_L74P_0/GCLK6S,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
B17,,,AVCCAUXRX6,,,,,,,,2.5,,,,,
B18,,,VTRXPAD6,,,,,,,,,,,,
B19,,,AVCCAUXTX6,,,,,,,,2.5,,,,,
B20,,,VTTXPAD6,,,,,,,,,,,,
B21,xin1<2>,IOB,IO_L50_0/No_Pair,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
B22,xin1<3>,IOB,IO_L47P_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
B23,zin1<32>,IOB,IO_L44N_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
B24,,,AVCCAUXRX4,,,,,,,,2.5,,,,,
B25,,,VTRXPAD4,,,,,,,,,,,,
B26,,,AVCCAUXTX4,,,,,,,,2.5,,,,,
B27,,,VTTXPAD4,,,,,,,,,,,,
B28,xin1<49>,IOB,IO_L01P_7/VRN_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
B29,,,GND,,,,,,,,,,,,
B30,,,VCCAUX,,,,,,,,2.5,,,,,
C1,qin1<18>,IOB,IO_L04P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
C2,qin1<19>,IOB,IO_L04N_2/VREF_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
C3,,,GND,,,,,,,,,,,,
C4,qin1<51>,IOB,IO_L06N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
C5,qin1<43>,IOB,IO_L03N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
C6,,,GNDA9,,,,,,,,,,,,
C7,xin1<7>,IOB,IO_L08N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
C8,qin1<53>,IOB,IO_L38N_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
C9,qin1<34>,IOB,IO_L47P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
C10,ceroin4<26>,IOB,IO_L56P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,

C11,ceroin4<16>,IOB,IO_L56N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
C12,,,GNDA7,,,,,,,,,,,,
C13,salida1<53>,IOB,IO_L68N_1,OUTPUT,LVCMOS25,1,12,SLOW,NONE**,,,,,NO,NONE,
C14,,,GND,,,,,,,,,,,,
C15,salida1<52>,IOB,IO_L74P_1/GCLK0S,OUTPUT,LVCMOS25,1,12,SLOW,NONE**,,,,,NO,NONE,
C16,ceroin4<39>,IOB,IO_L74N_0/GCLK7P,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
C17,,,GND,,,,,,,,,,,,
C18,xin1<9>,IOB,IO_L68P_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
C19,,,GNDA6,,,,,,,,,,,,
C20,ceroin4<9>,IOB,IO_L56P_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
C21,zin1<36>,IOB,IO_L56N_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
C22,zin1<31>,IOB,IO_L47N_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
C23,ceroin4<31>,IOB,IO_L38P_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
C24,salida1<3>,IOB,IO_L08P_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
C25,,,GNDA4,,,,,,,,,,,,
C26,salida1<49>,IOB,IO_L03N_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
C27,salida1<51>,IOB,IO_L06N_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
C28,,,GND,,,,,,,,,,,,
C29,salida1<23>,IOB,IO_L04N_7/VREF_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
C30,salida1<8>,IOB,IO_L04P_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
D1,qin1<48>,IOB,IO_L31P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
D2,qin1<36>,IOB,IO_L31N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
D3,qin1<35>,IOB,IO_L06P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
D4,,,GND,,,,,,,,,,,,
D5,qin1<5>,IOB,IO_L03P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
D6,,,NC,,,,,,,,,,,,
D7,qin1<24>,IOB,IO_L08P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
D8,qin1<58>,IOB,IO_L38P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
D9,,,GND,,,,,,,,,,,,
D10,qin1<62>,IOB,IO_L37N_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
D11,qin1<57>,IOB,IO_L43N_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
D12,,,GND,,,,,,,,,,,,
D13,xin1<53>,IOB,IO_L68P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
D14,ceroin4<53>,IOB,IO_L67N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
D15,ceroin4<6>,IOB,IO_L73N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
D16,xin1<52>,IOB,IO_L73P_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
D17,zin1<27>,IOB,IO_L67P_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
D18,xin1<11>,IOB,IO_L68N_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
D19,,,GND,,,,,,,,,,,,
D20,zin1<10>,IOB,IO_L43P_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
D21,salida1<39>,IOB,IO_L37P_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
D22,,,GND,,,,,,,,,,,,
D23,salida1<16>,IOB,IO_L38N_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
D24,salida1<31>,IOB,IO_L08N_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
D25,,,DXN,,,,,,,,,,,,
D26,salida1<10>,IOB,IO_L03P_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
D27,,,GND,,,,,,,,,,,,
D28,xin1<51>,IOB,IO_L06P_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
D29,hin1<15>,IOB,IO_L31N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
D30,hin1<16>,IOB,IO_L31P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
E1,qin1<15>,IOB,IO_L34P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
E2,qin1<13>,IOB,IO_L34N_2/VREF_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
E3,qin1<42>,IOB,IO_L33P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
E4,qin1<39>,IOB,IO_L33N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
E5,,,GND,,,,,,,,,,,,
E6,qin1<4>,IOB,IO_L01P_1/VRN_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
E7,qin1<44>,IOB,IO_L01N_1/VRP_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
E8,qin1<50>,IOB,IO_L03P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,

E9,qin1<30>,IOB,IO_L03N_1/VREF_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
E10,qin1<61>,IOB,IO_L37P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
E11,salida1<13>,IOB,IO_L43P_1,OUTPUT,LVCMOS25,1,12,SLOW,NONE**,,,,,NO,NONE,
E12,ceroin4<13>,IOB,IO_L46P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
E13,ceroin4<33>,IOB,IO_L46N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
E14,ceroin4<52>,IOB,IO_L67P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
E15,ceroin4<38>,IOB,IO_L73P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
E16,ceroin4<47>,IOB,IO_L73N_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
E17,salida1<18>,IOB,IO_L67N_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
E18,ceroin4<62>,IOB,IO_L46P_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
E19,zin1<30>,IOB,IO_L46N_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
E20,zin1<8>,IOB,IO_L43N_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
E21,ceroin4<2>,IOB,IO_L37N_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
E22,ceroin4<49>,IOB,IO_L03P_0/VREF_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
E23,salida1<33>,IOB,IO_L03N_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
E24,hin1<53>,IOB,IO_L01P_0/VRN_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
E25,hin1<52>,IOB,IO_L01N_0/VRP_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
E26,,,GND,,,,,,,,,,,,
E27,hin1<14>,IOB,IO_L33N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
E28,hin1<13>,IOB,IO_L33P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
E29,hin1<11>,IOB,IO_L34N_7/VREF_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
E30,hin1<10>,IOB,IO_L34P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
F1,zin1<2>,IOB,IO_L37P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
F2,zin1<5>,IOB,IO_L37N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
F3,qin1<32>,IOB,IO_L36P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
F4,qin1<14>,IOB,IO_L36N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
F5,,,TDO,,,,,,,,,,,,
F6,,,GND,,,,,,,,,,,,
F7,qin1<37>,IOB,IO_L02P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
F8,qin1<0>,IOB,IO_L02N_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
F9,qin1<59>,IOB,IO_L07P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
F10,qin1<60>,IOB,IO_L07N_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
F11,ceroin4<21>,IOB,IO_L48P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
F12,xin1<16>,IOB,IO_L48N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
F13,,,GND,,,,,,,,,,,,
F14,ceroin4<30>,IOB,IO_L49N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
F15,xin1<8>,IOB,IO_L75N_1/GCLK3P,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
F16,xin1<39>,IOB,IO_L75P_0/GCLK4S,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
F17,xin1<32>,IOB,IO_L49P_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
F18,,,GND,,,,,,,,,,,,
F19,ceroin4<3>,IOB,IO_L48P_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
F20,zin1<38>,IOB,IO_L48N_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
F21,salida1<32>,IOB,IO_L07P_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
F22,xin1<5>,IOB,IO_L07N_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
F23,salida1<17>,IOB,IO_L02P_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
F24,xin1<10>,IOB,IO_L02N_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
F25,,,GND,,,,,,,,,,,,
F26,,,TDI,,,,,,,,,,,,
F27,hin1<6>,IOB,IO_L36N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
F28,hin1<8>,IOB,IO_L36P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
F29,hin1<2>,IOB,IO_L37N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
F30,zin1<56>,IOB,IO_L37P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
G1,qin1<12>,IOB,IO_L40P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
G2,ceroin4<14>,IOB,IO_L40N_2/VREF_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
G3,salida1<14>,IOB,IO_L39P_2,OUTPUT,LVCMOS25,2,12,SLOW,NONE**,,,,,NO,NONE,
G4,qin1<2>,IOB,IO_L39N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
G5,qin1<29>,IOB,IO_L02P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
G6,qin1<6>,IOB,IO_L02N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,

G7,,,TCK,,,,,,,,,,,,
G8,ceroin4<24>,IOB,IO_L05_1/No_Pair,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
G9,ceroin4<18>,IOB,IO_L06N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
G10,qin1<55>,IOB,IO_L09N_1/VREF_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
G11,qin1<56>,IOB,IO_L39N_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
G12,ceroin4<48>,IOB,IO_L45N_1/VREF_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
G13,xin1<23>,IOB,IO_L54N_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
G14,ceroin4<20>,IOB,IO_L49P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
G15,zin1<55>,IOB,IO_L75P_1/GCLK2S,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
G16,ceroin4<7>,IOB,IO_L75N_0/GCLK5P,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
G17,zin1<24>,IOB,IO_L49N_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
G18,zin1<25>,IOB,IO_L54P_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
G19,zin1<26>,IOB,IO_L45P_0/VREF_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
G20,salida1<2>,IOB,IO_L39P_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
G21,zin1<48>,IOB,IO_L09P_0/VREF_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
G22,ceroin4<60>,IOB,IO_L06P_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
G23,xin1<31>,IOB,IO_L05_0/No_Pair,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
G24,,,PROG_B,,,,,,,,,,,,
G25,hin1<60>,IOB,IO_L02N_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
G26,hin1<61>,IOB,IO_L02P_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
G27,hin1<4>,IOB,IO_L39N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
G28,hin1<3>,IOB,IO_L39P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
G29,salida1<5>,IOB,IO_L40N_7/VREF_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
G30,xin1<42>,IOB,IO_L40P_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
H1,,,GND,,,,,,,,,,,,
H2,xin1<26>,IOB,IO_L43N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
H3,xin1<38>,IOB,IO_L42P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
H4,qin1<23>,IOB,IO_L42N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
H5,qin1<47>,IOB,IO_L32P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
H6,qin1<38>,IOB,IO_L32N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
H7,,,VBATT,,,,,,,,,,,,
H8,,,TMS,,,,,,,,,,,,
H9,ceroin4<32>,IOB,IO_L06P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
H10,qin1<52>,IOB,IO_L09P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
H11,qin1<63>,IOB,IO_L39P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
H12,qin1<49>,IOB,IO_L45P_1,INPUT,LVCMOS25,1,,,,IFD,,,,YES,NONE,
H13,ceroin4<29>,IOB,IO_L54P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
H14,ceroin4<35>,IOB,IO_L57N_1/VREF_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
H15,xin1<30>,IOB,IO_L69N_1/VREF_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
H16,xin1<33>,IOB,IO_L69P_0/VREF_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
H17,zin1<29>,IOB,IO_L57P_0/VREF_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
H18,zin1<49>,IOB,IO_L54N_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
H19,ceroin4<51>,IOB,IO_L45N_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
H20,salida1<7>,IOB,IO_L39N_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
H21,ceroin4<10>,IOB,IO_L09N_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
H22,xin1<18>,IOB,IO_L06N_0,INPUT,LVCMOS25,0,,,,NONE,,,,NO,NONE,
H23,,,HSWAP_EN,,,,,,,,,,,,
H24,,,DXP,,,,,,,,,,,,
H25,zin1<61>,IOB,IO_L32N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
H26,zin1<60>,IOB,IO_L32P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
H27,salida1<47>,IOB,IO_L42N_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
H28,salida1<35>,IOB,IO_L42P_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
H29,xin1<58>,IOB,IO_L43N_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
H30,,,GND,,,,,,,,,,,,
J1,zin1<7>,IOB,IO_L46N_2/VREF_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
J2,salida1<41>,IOB,IO_L43P_2,OUTPUT,LVCMOS25,2,12,SLOW,NONE**,,,,,NO,NONE,
J3,xin1<17>,IOB,IO_L48P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
J4,salida1<12>,IOB,IO_L48N_2,OUTPUT,LVCMOS25,2,12,SLOW,NONE**,,,,,NO,NONE,

J5,qin1<41>,IOB,IO_L38P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
J6,qin1<40>,IOB,IO_L38N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
J7,qin1<28>,IOB,IO_L05P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
J8,xin1<13>,IOB,IO_L05N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
J9,,,VCCO_2,,,2,,,,,2.50,,,,,
J10,,,VCCO_1,,,1,,,,,2.50,,,,,
J11,,,VCCO_1,,,1,,,,,2.50,,,,,
J12,,,VCCO_1,,,1,,,,,2.50,,,,,
J13,,,VCCO_1,,,1,,,,,2.50,,,,,
J14,xin1<48>,IOB,IO_L57P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
J15,xin1<6>,IOB,IO_L69P_1,INPUT,LVCMOS25,1,,,,NONE,,,,NO,NONE,
J16,salida1<30>,IOB,IO_L69N_0,OUTPUT,LVCMOS25,0,12,SLOW,NONE**,,,,,NO,NONE,
J17,zin1<39>,IOB,IO_L57N_0,INPUT,LVCMOS25,0,,,,IFD,,,,YES,NONE,
J18,,,VCCO_0,,,0,,,,,2.50,,,,,
J19,,,VCCO_0,,,0,,,,,2.50,,,,,
J20,,,VCCO_0,,,0,,,,,2.50,,,,,
J21,,,VCCO_0,,,0,,,,,2.50,,,,,
J22,,,VCCO_7,,,7,,,,,2.50,,,,,
J23,xin1<4>,IOB,IO_L05N_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
J24,salida1<29>,IOB,IO_L05P_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
J25,hin1<5>,IOB,IO_L38N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
J26,hin1<7>,IOB,IO_L38P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
J27,salida1<0>,IOB,IO_L48N_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
J28,zin1<9>,IOB,IO_L48P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
J29,zin1<28>,IOB,IO_L43P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
J30,zin1<57>,IOB,IO_L46N_7/VREF_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
K1,salida1<50>,IOB,IO_L46P_2,OUTPUT,LVCMOS25,2,12,SLOW,NONE**,,,,,NO,NONE,
K2,ceroin4<0>,IOB,IO_L49N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
K3,xin1<20>,IOB,IO_L51P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
K4,xin1<14>,IOB,IO_L51N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
K5,zin1<4>,IOB,IO_L45P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
K6,zin1<3>,IOB,IO_L45N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
K7,qin1<3>,IOB,IO_L35P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
K8,qin1<33>,IOB,IO_L35N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
K9,,,VCCO_2,,,2,,,,,2.50,,,,,
K10,,,VCCO_1,,,1,,,,,2.50,,,,,
K11,,,VCCO_1,,,1,,,,,2.50,,,,,
K12,,,VCCO_1,,,1,,,,,2.50,,,,,
K13,,,VCCO_1,,,1,,,,,2.50,,,,,
K14,,,VCCO_1,,,1,,,,,2.50,,,,,
K15,,,VCCO_1,,,1,,,,,2.50,,,,,
K16,,,VCCO_0,,,0,,,,,2.50,,,,,
K17,,,VCCO_0,,,0,,,,,2.50,,,,,
K18,,,VCCO_0,,,0,,,,,2.50,,,,,
K19,,,VCCO_0,,,0,,,,,2.50,,,,,
K20,,,VCCO_0,,,0,,,,,2.50,,,,,
K21,,,VCCO_0,,,0,,,,,2.50,,,,,
K22,,,VCCO_7,,,7,,,,,2.50,,,,,
K23,hin1<12>,IOB,IO_L35N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
K24,hin1<9>,IOB,IO_L35P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
K25,hin1<0>,IOB,IO_L45N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
K26,zin1<53>,IOB,IO_L45P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
K27,hin1<18>,IOB,IO_L51N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
K28,hin1<17>,IOB,IO_L51P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
K29,zin1<23>,IOB,IO_L49N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
K30,zin1<14>,IOB,IO_L46P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
L1,xin1<35>,IOB,IO_L52N_2/VREF_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
L2,ceroin4<43>,IOB,IO_L49P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,

L3,,,GND,,,,,,,,,,,,
L4,qin1<11>,IOB,IO_L54P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
L5,xin1<12>,IOB,IO_L54N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
L6,,,GND,,,,,,,,,,,,
L7,qin1<26>,IOB,IO_L41P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
L8,ceroin4<15>,IOB,IO_L41N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
L9,,,VCCO_2,,,2,,,,,2.50,,,,,
L10,,,VCCO_2,,,2,,,,,2.50,,,,,
L11,,,GND,,,,,,,,,,,,
L12,,,VCCINT,,,,,,,,1.5,,,,,
L13,,,VCCINT,,,,,,,,1.5,,,,,
L14,,,VCCINT,,,,,,,,1.5,,,,,
L15,,,VCCINT,,,,,,,,1.5,,,,,
L16,,,VCCINT,,,,,,,,1.5,,,,,
L17,,,VCCINT,,,,,,,,1.5,,,,,
L18,,,VCCINT,,,,,,,,1.5,,,,,
L19,,,VCCINT,,,,,,,,1.5,,,,,
L20,,,GND,,,,,,,,,,,,
L21,,,VCCO_7,,,7,,,,,2.50,,,,,
L22,,,VCCO_7,,,7,,,,,2.50,,,,,
L23,salida1<26>,IOB,IO_L41N_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
L24,zin1<52>,IOB,IO_L41P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
L25,,,GND,,,,,,,,,,,,
L26,zin1<58>,IOB,IO_L54N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
L27,zin1<59>,IOB,IO_L54P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
L28,,,GND,,,,,,,,,,,,
L29,salida1<58>,IOB,IO_L49P_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
L30,salida1<19>,IOB,IO_L52N_7/VREF_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
M1,xin1<47>,IOB,IO_L52P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
M2,zin1<15>,IOB,IO_L55N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
M3,zin1<0>,IOB,IO_L57P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
M4,xin1<50>,IOB,IO_L57N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
M5,zin1<54>,IOB,IO_L47P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
M6,salida1<28>,IOB,IO_L47N_2,OUTPUT,LVCMOS25,2,12,SLOW,NONE**,,,,,NO,NONE,
M7,ceroin4<40>,IOB,IO_L44P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
M8,ceroin4<17>,IOB,IO_L44N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
M9,,,VCCO_2,,,2,,,,,2.50,,,,,
M10,,,VCCO_2,,,2,,,,,2.50,,,,,
M11,,,VCCINT,,,,,,,,1.5,,,,,
M12,,,GND,,,,,,,,,,,,
M13,,,GND,,,,,,,,,,,,
M14,,,GND,,,,,,,,,,,,
M15,,,GND,,,,,,,,,,,,
M16,,,GND,,,,,,,,,,,,
M17,,,GND,,,,,,,,,,,,
M18,,,GND,,,,,,,,,,,,
M19,,,GND,,,,,,,,,,,,
M20,,,VCCINT,,,,,,,,1.5,,,,,
M21,,,VCCO_7,,,7,,,,,2.50,,,,,
M22,,,VCCO_7,,,7,,,,,2.50,,,,,
M23,ceroin4<58>,IOB,IO_L44N_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
M24,salida1<61>,IOB,IO_L44P_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
M25,zin1<63>,IOB,IO_L47N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
M26,hin1<1>,IOB,IO_L47P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
M27,salida1<42>,IOB,IO_L57N_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
M28,salida1<21>,IOB,IO_L57P_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
M29,zin1<12>,IOB,IO_L55N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
M30,xin1<27>,IOB,IO_L52P_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,

N1,qin1<46>,IOB,IO_L58N_2/VREF_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
N2,salida1<22>,IOB,IO_L55P_2,OUTPUT,LVCMOS25,2,12,SLOW,NONE**,,,,,NO,NONE,
N3,xin1<24>,IOB,IO_L60P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
N4,xin1<43>,IOB,IO_L60N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
N5,xin1<0>,IOB,IO_L53P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
N6,salida1<25>,IOB,IO_L53N_2,OUTPUT,LVCMOS25,2,12,SLOW,NONE**,,,,,NO,NONE,
N7,salida1<43>,IOB,IO_L50P_2,OUTPUT,LVCMOS25,2,12,SLOW,NONE**,,,,,NO,NONE,
N8,salida1<24>,IOB,IO_L50N_2,OUTPUT,LVCMOS25,2,12,SLOW,NONE**,,,,,NO,NONE,
N9,,,VCCO_2,,,2,,,,,2.50,,,,,
N10,,,VCCO_2,,,2,,,,,2.50,,,,,
N11,,,VCCINT,,,,,,,,1.5,,,,,
N12,,,GND,,,,,,,,,,,,
N13,,,GND,,,,,,,,,,,,
N14,,,GND,,,,,,,,,,,,
N15,,,GND,,,,,,,,,,,,
N16,,,GND,,,,,,,,,,,,
N17,,,GND,,,,,,,,,,,,
N18,,,GND,,,,,,,,,,,,
N19,,,GND,,,,,,,,,,,,
N20,,,VCCINT,,,,,,,,1.5,,,,,
N21,,,VCCO_7,,,7,,,,,2.50,,,,,
N22,,,VCCO_7,,,7,,,,,2.50,,,,,
N23,xin1<61>,IOB,IO_L50N_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
N24,salida1<15>,IOB,IO_L50P_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
N25,salida1<11>,IOB,IO_L53N_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
N26,zin1<13>,IOB,IO_L53P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
N27,ceroin4<61>,IOB,IO_L60N_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
N28,zin1<16>,IOB,IO_L60P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
N29,xin1<15>,IOB,IO_L55P_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
N30,zin1<46>,IOB,IO_L58N_7/VREF_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
P1,xin1<40>,IOB,IO_L58P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
P2,ceroin4<41>,IOB,IO_L85P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
P3,xin1<21>,IOB,IO_L85N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
P4,ceroin4<22>,IOB,IO_L87P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
P5,xin1<28>,IOB,IO_L87N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
P6,,,GND,,,,,,,,,,,,
P7,zin1<6>,IOB,IO_L59P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
P8,qin1<45>,IOB,IO_L59N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
P9,zin1<1>,IOB,IO_L56N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
P10,,,VCCO_2,,,2,,,,,2.50,,,,,
P11,,,VCCINT,,,,,,,,1.5,,,,,
P12,,,GND,,,,,,,,,,,,
P13,,,GND,,,,,,,,,,,,
P14,,,GND,,,,,,,,,,,,
P15,,,GND,,,,,,,,,,,,
P16,,,GND,,,,,,,,,,,,
P17,,,GND,,,,,,,,,,,,
P18,,,GND,,,,,,,,,,,,
P19,,,GND,,,,,,,,,,,,
P20,,,VCCINT,,,,,,,,1.5,,,,,
P21,,,VCCO_7,,,7,,,,,2.50,,,,,
P22,zin1<47>,IOB,IO_L56N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
P23,zin1<22>,IOB,IO_L59N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
P24,salida1<20>,IOB,IO_L59P_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
P25,,,GND,,,,,,,,,,,,
P26,ceroin4<37>,IOB,IO_L87N_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
P27,zin1<18>,IOB,IO_L87P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
P28,ceroin4<5>,IOB,IO_L85N_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,

P29,ceroin4<4>,IOB,IO_L85P_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
P30,zin1<17>,IOB,IO_L58P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
R1,,,VCCAUX,,,,,,,,2.5,,,,,
R2,ceroin4<25>,IOB,IO_L88N_2/VREF_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
R3,qin1<20>,IOB,IO_L90P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
R4,qin1<31>,IOB,IO_L90N_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
R5,xin1<54>,IOB,IO_L89P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
R6,ceroin4<54>,IOB,IO_L89N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
R7,xin1<22>,IOB,IO_L86P_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
R8,xin1<41>,IOB,IO_L86N_2,INPUT,LVCMOS25,2,,,,NONE,,,,NO,NONE,
R9,zin1<11>,IOB,IO_L56P_2,INPUT,LVCMOS25,2,,,,IFD,,,,YES,NONE,
R10,,,VCCO_2,,,2,,,,,2.50,,,,,
R11,,,VCCINT,,,,,,,,1.5,,,,,
R12,,,GND,,,,,,,,,,,,
R13,,,GND,,,,,,,,,,,,
R14,,,GND,,,,,,,,,,,,
R15,,,GND,,,,,,,,,,,,
R16,,,GND,,,,,,,,,,,,
R17,,,GND,,,,,,,,,,,,
R18,,,GND,,,,,,,,,,,,
R19,,,GND,,,,,,,,,,,,
R20,,,VCCINT,,,,,,,,1.5,,,,,
R21,,,VCCO_7,,,7,,,,,2.50,,,,,
R22,zin1<21>,IOB,IO_L56P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
R23,zin1<19>,IOB,IO_L86N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
R24,zin1<20>,IOB,IO_L86P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
R25,zin1<50>,IOB,IO_L89N_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
R26,xin1<37>,IOB,IO_L89P_7,INPUT,LVCMOS25,7,,,,NONE,,,,NO,NONE,
R27,salida1<36>,IOB,IO_L90N_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
R28,salida1<45>,IOB,IO_L90P_7,OUTPUT,LVCMOS25,7,12,SLOW,NONE**,,,,,NO,NONE,
R29,zin1<41>,IOB,IO_L88N_7/VREF_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
R30,,,VCCAUX,,,,,,,,2.5,,,,,
T1,,,VCCAUX,,,,,,,,2.5,,,,,
T2,salida1<54>,IOB,IO_L88P_2,OUTPUT,LVCMOS25,2,12,SLOW,NONE**,,,,,NO,NONE,
T3,ceroin4<50>,IOB,IO_L88N_3,INPUT,LVCMOS25,3,,,,NONE,,,,NO,NONE,
T4,qin1<27>,IOB,IO_L88P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
T5,qin1<16>,IOB,IO_L89N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
T6,qin1<10>,IOB,IO_L89P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
T7,qin1<22>,IOB,IO_L86N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
T8,ceroin4<28>,IOB,IO_L86P_3,INPUT,LVCMOS25,3,,,,NONE,,,,NO,NONE,
T9,,DIFFS,IO_L59N_3,UNUSED,,3,,,,,,,,,
T10,,,VCCO_3,,,3,,,,,2.50,,,,,
T11,,,VCCINT,,,,,,,,1.5,,,,,
T12,,,GND,,,,,,,,,,,,
T13,,,GND,,,,,,,,,,,,
T14,,,GND,,,,,,,,,,,,
T15,,,GND,,,,,,,,,,,,
T16,,,GND,,,,,,,,,,,,
T17,,,GND,,,,,,,,,,,,
T18,,,GND,,,,,,,,,,,,
T19,,,GND,,,,,,,,,,,,
T20,,,VCCINT,,,,,,,,1.5,,,,,
T21,,,VCCO_6,,,6,,,,,2.50,,,,,
T22,hin1<62>,IOB,IO_L59N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
T23,xin1<56>,IOB,IO_L86P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
T24,salida1<38>,IOB,IO_L86N_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
T25,xin1<62>,IOB,IO_L89P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
T26,salida1<48>,IOB,IO_L89N_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,

T27,salida1<6>,IOB,IO_L88P_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
T28,ceroin4<56>,IOB,IO_L88N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
T29,zin1<40>,IOB,IO_L88P_7,INPUT,LVCMOS25,7,,,,IFD,,,,YES,NONE,
T30,,,VCCAUX,,,,,,,,2.5,,,,,
U1,qin1<17>,IOB,IO_L90N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
U2,qin1<8>,IOB,IO_L87N_3/VREF_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
U3,ceroin4<12>,IOB,IO_L87P_3,INPUT,LVCMOS25,3,,,,NONE,,,,NO,NONE,
U4,salida1<40>,IOB,IO_L85N_3,OUTPUT,LVCMOS25,3,12,SLOW,NONE**,,,,,NO,NONE,
U5,qin1<7>,IOB,IO_L85P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
U6,,,GND,,,,,,,,,,,,,
U7,,DIFFS,IO_L56N_3,UNUSED,,3,,,,,,,,,,
U8,,DIFFM,IO_L56P_3,UNUSED,,3,,,,,,,,,,
U9,xin1<25>,IOB,IO_L59P_3,INPUT,LVCMOS25,3,,,,NONE,,,,NO,NONE,
U10,,,VCCO_3,,,3,,,,,2.50,,,,,
U11,,,VCCINT,,,,,,,,1.5,,,,,
U12,,,GND,,,,,,,,,,,,,
U13,,,GND,,,,,,,,,,,,,
U14,,,GND,,,,,,,,,,,,,
U15,,,GND,,,,,,,,,,,,,
U16,,,GND,,,,,,,,,,,,,
U17,,,GND,,,,,,,,,,,,,
U18,,,GND,,,,,,,,,,,,,
U19,,,GND,,,,,,,,,,,,,
U20,,,VCCINT,,,,,,,,1.5,,,,,
U21,,,VCCO_6,,,6,,,,,2.50,,,,,
U22,hin1<59>,IOB,IO_L59P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
U23,hin1<45>,IOB,IO_L56P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
U24,hin1<48>,IOB,IO_L56N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
U25,,,GND,,,,,,,,,,,,,
U26,salida1<27>,IOB,IO_L85P_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
U27,salida1<62>,IOB,IO_L85N_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
U28,ceroin4<27>,IOB,IO_L87P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
U29,salida1<37>,IOB,IO_L87N_6/VREF_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
U30,salida1<46>,IOB,IO_L90N_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
V1,qin1<9>,IOB,IO_L90P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
V2,qin1<21>,IOB,IO_L60N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
V3,,DIFFS,IO_L58N_3,UNUSED,,3,,,,,,,,,,
V4,xin1<19>,IOB,IO_L58P_3,INPUT,LVCMOS25,3,,,,NONE,,,,NO,NONE,
V5,,DIFFS,IO_L55N_3,UNUSED,,3,,,,,,,,,,
V6,hin1<42>,IOB,IO_L55P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
V7,ceroin4<44>,IOB,IO_L53N_3,INPUT,LVCMOS25,3,,,,NONE,,,,NO,NONE,
V8,ceroin4<55>,IOB,IO_L53P_3,INPUT,LVCMOS25,3,,,,NONE,,,,NO,NONE,
V9,,,VCCO_3,,,3,,,,,2.50,,,,,
V10,,,VCCO_3,,,3,,,,,2.50,,,,,
V11,,,VCCINT,,,,,,,,1.5,,,,,
V12,,,GND,,,,,,,,,,,,,
V13,,,GND,,,,,,,,,,,,,
V14,,,GND,,,,,,,,,,,,,
V15,,,GND,,,,,,,,,,,,,
V16,,,GND,,,,,,,,,,,,,
V17,,,GND,,,,,,,,,,,,,
V18,,,GND,,,,,,,,,,,,,
V19,,,GND,,,,,,,,,,,,,
V20,,,VCCINT,,,,,,,,1.5,,,,,
V21,,,VCCO_6,,,6,,,,,2.50,,,,,
V22,,,VCCO_6,,,6,,,,,2.50,,,,,
V23,salida1<59>,IOB,IO_L53P_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
V24,xin1<59>,IOB,IO_L53N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,

V25,hin1<36>,IOB,IO_L55P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
V26,hin1<51>,IOB,IO_L55N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
V27,salida1<63>,IOB,IO_L58P_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
V28,hin1<58>,IOB,IO_L58N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
V29,salida1<1>,IOB,IO_L60N_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
V30,zin1<51>,IOB,IO_L90P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
W1,,DIFFS,IO_L57N_3/VREF_3,UNUSED,,3,,,,,,,,,,
W2,ceroin4<19>,IOB,IO_L60P_3,INPUT,LVCMOS25,3,,,,NONE,,,,NO,NONE,
W3,salida1<55>,IOB,IO_L52N_3,OUTPUT,LVCMOS25,3,12,SLOW,NONE**,,,,,NO,NONE,
W4,,DIFFM,IO_L52P_3,UNUSED,,3,,,,,,,,,,
W5,rin1<50>,IOB,IO_L50N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
W6,rin1<51>,IOB,IO_L50P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
W7,rin1<38>,IOB,IO_L47N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
W8,rin1<37>,IOB,IO_L47P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
W9,,,VCCO_3,,,3,,,,,2.50,,,,,
W10,,,VCCO_3,,,3,,,,,2.50,,,,,
W11,,,VCCINT,,,,,,,,1.5,,,,,
W12,,,GND,,,,,,,,,,,,
W13,,,GND,,,,,,,,,,,,
W14,,,GND,,,,,,,,,,,,
W15,,,GND,,,,,,,,,,,,
W16,,,GND,,,,,,,,,,,,
W17,,,GND,,,,,,,,,,,,
W18,,,GND,,,,,,,,,,,,
W19,,,GND,,,,,,,,,,,,
W20,,,VCCINT,,,,,,,,1.5,,,,,
W21,,,VCCO_6,,,6,,,,,2.50,,,,,
W22,,,VCCO_6,,,6,,,,,2.50,,,,,
W23,hin1<46>,IOB,IO_L47P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
W24,hin1<49>,IOB,IO_L47N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
W25,ceroin4<57>,IOB,IO_L50P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
W26,xin1<57>,IOB,IO_L50N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
W27,xin1<63>,IOB,IO_L52P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
W28,ceroin4<59>,IOB,IO_L52N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
W29,salida1<56>,IOB,IO_L60P_6,OUTPUT,LVCMOS25,6,12,SLOW,NONE**,,,,,NO,NONE,
W30,xin1<1>,IOB,IO_L57N_6/VREF_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
Y1,,DIFFM,IO_L57P_3,UNUSED,,3,,,,,,,,,,
Y2,,DIFFS,IO_L54N_3,UNUSED,,3,,,,,,,,,,
Y3,,,GND,,,,,,,,,,,,
Y4,rin1<12>,IOB,IO_L49N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
Y5,rin1<41>,IOB,IO_L49P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
Y6,,,GND,,,,,,,,,,,,
Y7,rin1<22>,IOB,IO_L41N_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
Y8,rin1<42>,IOB,IO_L41P_3,INPUT,LVCMOS25,3,,,,IFD,,,,YES,NONE,
Y9,,,VCCO_3,,,3,,,,,2.50,,,,,
Y10,,,VCCO_3,,,3,,,,,2.50,,,,,
Y11,,,GND,,,,,,,,,,,,
Y12,,,VCCINT,,,,,,,,1.5,,,,,
Y13,,,VCCINT,,,,,,,,1.5,,,,,
Y14,,,VCCINT,,,,,,,,1.5,,,,,
Y15,,,VCCINT,,,,,,,,1.5,,,,,
Y16,,,VCCINT,,,,,,,,1.5,,,,,
Y17,,,VCCINT,,,,,,,,1.5,,,,,
Y18,,,VCCINT,,,,,,,,1.5,,,,,
Y19,,,VCCINT,,,,,,,,1.5,,,,,
Y20,,,GND,,,,,,,,,,,,
Y21,,,VCCO_6,,,6,,,,,2.50,,,,,
Y22,,,VCCO_6,,,6,,,,,2.50,,,,,

Y23,hin1<30>,IOB,IO_L41P_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
Y24,hin1<33>,IOB,IO_L41N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
Y25,,,GND,,,,,,,,,,,,
Y26,ceroin4<34>,IOB,IO_L49P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
Y27,xin1<34>,IOB,IO_L49N_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,
Y28,,,GND,,,,,,,,,,,,
Y29,hin1<38>,IOB,IO_L54N_6,INPUT,LVCMOS25,6,,,,IFD,,,,YES,NONE,
Y30,ceroin4<1>,IOB,IO_L57P_6,INPUT,LVCMOS25,6,,,,NONE,,,,NO,NONE,

-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,-----,
#
#* Default value.
#** This default Pullup/Pulldown value can be overridden in Bitgen.
#****** Special VCCO requirements may apply. Please consult the device
family datasheet for specific guideline on VCCO requirements.
#
#

APENDICE

Release 8.1i Map I.24
Xilinx Mapping Report File for Design 'kalman_cosim5'

Design Information

Command Line   : C:\Xilinx\bin\nt\map.exe -ise C:/xup/Copia de
prueba_kalman3/netlist/prueba_kalman3/netlist/kalman_cosim5_cw.ise -intstyle ise
-p xc2vp30-ff896-7 -cm area -pr b -k 4 -c 100 -tx off -o kalman_cosim5_map.ncd
kalman_cosim5.ngd kalman_cosim5.pcf
Target Device : xc2vp30
Target Package : ff896
Target Speed   : -7
Mapper Version : virtex2p -- $Revision: 1.34 $
Mapped Date   : Sat May 24 11:35:42 2008

Design Summary

Number of errors: 0
Number of warnings: 2
Logic Utilization:
  Number of Slice Flip Flops: 20,909 out of 27,392   76%
  Number of 4 input LUTs: 13,768 out of 27,392   50%
Logic Distribution:
  Number of occupied Slices: 13,694 out of 13,696   99%
  Number of Slices containing only related logic: 11,602 out of 13,694   84%
  Number of Slices containing unrelated logic: 2,092 out of 13,694   15%
       *See NOTES below for an explanation of the effects of unrelated logic
Total Number 4 input LUTs: 21,715 out of 27,392   79%
  Number used as logic: 13,768
  Number used as a route-thru: 986
  Number used as Shift registers: 6,961

  Number of bonded IOBs: 515 out of   556   92%
    IOB Flip Flops: 412
  Number of PPC405s: 0 out of   2   0%
  Number of MULT18X18s: 54 out of   136   39%
  Number of GCLKs: 1 out of   16   6%
  Number of GTs: 0 out of   8   0%

Number of GT10s: 0 out of 0 0%

Total equivalent gate count for design: 974,649
Additional JTAG gate count for IOBs: 24,720
Peak Memory Usage: 604 MB

NOTES:

Related logic is defined as being logic that shares connectivity - e.g. two LUTs are "related" if they share common inputs. When assembling slices, Map gives priority to combine logic that is related. Doing so results in the best timing performance.

Unrelated logic shares no connectivity. Map will only begin packing unrelated logic into a slice once 99% of the slices are occupied through related logic packing.

Note that once logic distribution reaches the 99% level through related logic packing, this does not mean the device is completely utilized. Unrelated logic packing will then begin, continuing until all usable LUTs and FFs are occupied. Depending on your timing budget, increased levels of unrelated logic packing may adversely affect the overall timing performance of your design.

Table of Contents

Section 1 - Errors

Section 6 - IOB Properties

IOB Name	Type	Direction	IO Standard	Drive Strength	Slew Rate	Reg (s)	Resistor	IOB Delay
ain2<0>	IOB	INPUT	LVCMOS25			INFF1 INFF2		IFD
ain2<1>	IOB	INPUT	LVCMOS25			INFF1 INFF2		IFD
ain2<2>	IOB	INPUT	LVCMOS25			INFF1 INFF2		IFD

ain2<3>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<4>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<5>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<6>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<7>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<8>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<9>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<10>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<11>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<12>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<13>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<14>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<15>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<16>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<17>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<18>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<19>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<20>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<21>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<22>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<23>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<24>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<25>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<26>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<27>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<28>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<29>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<30>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
ain2<31>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		

ain2<32>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<33>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<34>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<35>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<36>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<37>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<38>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<39>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<40>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<41>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<42>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<43>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<44>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<45>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<46>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<47>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<48>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<49>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<50>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<51>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2			
ain2<52>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<53>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<54>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<55>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<56>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<57>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<58>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<59>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<60>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<61>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<62>	IOB	INPUT	LVCMOS25			INFF1		IFD
ain2<63>	IOB	INPUT	LVCMOS25			INFF1		IFD
ce_1	IOB	INPUT	LVCMOS25					
ceroin4<0>	IOB	INPUT	LVCMOS25					
ceroin4<1>	IOB	INPUT	LVCMOS25					
ceroin4<2>	IOB	INPUT	LVCMOS25					
ceroin4<3>	IOB	INPUT	LVCMOS25					
ceroin4<4>	IOB	INPUT	LVCMOS25					

ceroin4<5>	IOB	INPUT	LVCMOS25					
ceroin4<6>	IOB	INPUT	LVCMOS25					
ceroin4<7>	IOB	INPUT	LVCMOS25					
ceroin4<8>	IOB	INPUT	LVCMOS25					
ceroin4<9>	IOB	INPUT	LVCMOS25					
ceroin4<10>	IOB	INPUT	LVCMOS25					
ceroin4<11>	IOB	INPUT	LVCMOS25					
ceroin4<12>	IOB	INPUT	LVCMOS25					
ceroin4<13>	IOB	INPUT	LVCMOS25					
ceroin4<14>	IOB	INPUT	LVCMOS25					
ceroin4<15>	IOB	INPUT	LVCMOS25					
ceroin4<16>	IOB	INPUT	LVCMOS25					
ceroin4<17>	IOB	INPUT	LVCMOS25					
ceroin4<18>	IOB	INPUT	LVCMOS25					
ceroin4<19>	IOB	INPUT	LVCMOS25					
ceroin4<20>	IOB	INPUT	LVCMOS25					
ceroin4<21>	IOB	INPUT	LVCMOS25					
ceroin4<22>	IOB	INPUT	LVCMOS25					
ceroin4<23>	IOB	INPUT	LVCMOS25					
ceroin4<24>	IOB	INPUT	LVCMOS25					
ceroin4<25>	IOB	INPUT	LVCMOS25					
ceroin4<26>	IOB	INPUT	LVCMOS25					
ceroin4<27>	IOB	INPUT	LVCMOS25					
ceroin4<28>	IOB	INPUT	LVCMOS25					
ceroin4<29>	IOB	INPUT	LVCMOS25					
ceroin4<30>	IOB	INPUT	LVCMOS25					
ceroin4<31>	IOB	INPUT	LVCMOS25					
ceroin4<32>	IOB	INPUT	LVCMOS25					
ceroin4<33>	IOB	INPUT	LVCMOS25					
ceroin4<34>	IOB	INPUT	LVCMOS25					
ceroin4<35>	IOB	INPUT	LVCMOS25					
ceroin4<36>	IOB	INPUT	LVCMOS25					
ceroin4<37>	IOB	INPUT	LVCMOS25					
ceroin4<38>	IOB	INPUT	LVCMOS25					
ceroin4<39>	IOB	INPUT	LVCMOS25					
ceroin4<40>	IOB	INPUT	LVCMOS25					
ceroin4<41>	IOB	INPUT	LVCMOS25					
ceroin4<42>	IOB	INPUT	LVCMOS25					
ceroin4<43>	IOB	INPUT	LVCMOS25					
ceroin4<44>	IOB	INPUT	LVCMOS25					
ceroin4<45>	IOB	INPUT	LVCMOS25					
ceroin4<46>	IOB	INPUT	LVCMOS25					
ceroin4<47>	IOB	INPUT	LVCMOS25					
ceroin4<48>	IOB	INPUT	LVCMOS25					
ceroin4<49>	IOB	INPUT	LVCMOS25					
ceroin4<50>	IOB	INPUT	LVCMOS25					
ceroin4<51>	IOB	INPUT	LVCMOS25					
ceroin4<52>	IOB	INPUT	LVCMOS25					
ceroin4<53>	IOB	INPUT	LVCMOS25					
ceroin4<54>	IOB	INPUT	LVCMOS25					
ceroin4<55>	IOB	INPUT	LVCMOS25					
ceroin4<56>	IOB	INPUT	LVCMOS25					
ceroin4<57>	IOB	INPUT	LVCMOS25					
ceroin4<58>	IOB	INPUT	LVCMOS25					
ceroin4<59>	IOB	INPUT	LVCMOS25					
ceroin4<60>	IOB	INPUT	LVCMOS25					
ceroin4<61>	IOB	INPUT	LVCMOS25					
ceroin4<62>	IOB	INPUT	LVCMOS25					

ceroin4<63>	IOB	INPUT	LVCMOS25					
clk_1	IOB	INPUT	LVCMOS25					
consta1	IOB	INPUT	LVCMOS25					
hin1<0>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<1>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<2>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<3>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<4>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<5>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<6>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<7>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<8>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<9>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<10>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<11>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<12>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<13>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<14>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<15>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<16>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<17>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<18>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<19>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<20>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<21>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<22>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<23>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<24>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<25>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<26>	IOB	INPUT	LVCMOS25			INFF1		IFD
						INFF2		
hin1<27>	IOB	INPUT	LVCMOS25			INFF1		IFD

					INFF2					
hin1<28>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<29>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<30>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<31>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<32>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<33>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<34>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<35>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<36>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<37>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<38>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<39>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<40>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<41>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<42>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<43>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<44>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<45>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<46>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<47>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<48>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<49>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<50>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<51>	IOB	INPUT	LVCMOS25			INFF1		IFD		
					INFF2					
hin1<52>	IOB	INPUT	LVCMOS25							
hin1<53>	IOB	INPUT	LVCMOS25							
hin1<54>	IOB	INPUT	LVCMOS25							
hin1<55>	IOB	INPUT	LVCMOS25							
hin1<56>	IOB	INPUT	LVCMOS25							
hin1<57>	IOB	INPUT	LVCMOS25							
hin1<58>	IOB	INPUT	LVCMOS25							
hin1<59>	IOB	INPUT	LVCMOS25							
hin1<60>	IOB	INPUT	LVCMOS25							

hin1<61>	IOB	INPUT	LVCMOS25					
hin1<62>	IOB	INPUT	LVCMOS25					
hin1<63>	IOB	INPUT	LVCMOS25					
qin1<0>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<1>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<2>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<3>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<4>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<5>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<6>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<7>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<8>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<9>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<10>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<11>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<12>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<13>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<14>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<15>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<16>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<17>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<18>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<19>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<20>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<21>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<22>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<23>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<24>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<25>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<26>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<27>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<28>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<29>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<30>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<31>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<32>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<33>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<34>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<35>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<36>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<37>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<38>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<39>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<40>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<41>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<42>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<43>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<44>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<45>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<46>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<47>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<48>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<49>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<50>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<51>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<52>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<53>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<54>	IOB	INPUT	LVCMOS25			INFF1		IFD

qin1<55>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<56>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<57>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<58>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<59>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<60>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<61>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<62>	IOB	INPUT	LVCMOS25			INFF1		IFD
qin1<63>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<0>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<1>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<2>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<3>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<4>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<5>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<6>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<7>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<8>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<9>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<10>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<11>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<12>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<13>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<14>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<15>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<16>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<17>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<18>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<19>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<20>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<21>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<22>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<23>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<24>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<25>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<26>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<27>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<28>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<29>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<30>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<31>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<32>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<33>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<34>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<35>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<36>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<37>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<38>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<39>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<40>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<41>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<42>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<43>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<44>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<45>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<46>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<47>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<48>	IOB	INPUT	LVCMOS25			INFF1		IFD

rin1<49>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<50>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<51>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<52>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<53>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<54>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<55>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<56>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<57>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<58>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<59>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<60>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<61>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<62>	IOB	INPUT	LVCMOS25			INFF1		IFD
rin1<63>	IOB	INPUT	LVCMOS25			INFF1		IFD
salida1<0>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<1>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<2>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<3>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<4>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<5>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<6>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<7>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<8>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<9>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<10>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<11>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<12>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<13>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<14>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<15>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<16>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<17>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<18>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<19>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<20>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<21>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<22>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<23>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<24>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<25>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<26>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<27>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<28>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<29>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<30>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<31>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<32>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<33>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<34>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<35>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<36>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<37>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<38>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<39>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<40>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<41>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<42>	IOB	OUTPUT	LVCMOS25	12	SLOW			

salida1<43>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<44>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<45>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<46>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<47>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<48>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<49>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<50>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<51>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<52>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<53>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<54>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<55>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<56>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<57>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<58>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<59>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<60>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<61>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<62>	IOB	OUTPUT	LVCMOS25	12	SLOW			
salida1<63>	IOB	OUTPUT	LVCMOS25	12	SLOW			
xin1<0>	IOB	INPUT	LVCMOS25					
xin1<1>	IOB	INPUT	LVCMOS25					
xin1<2>	IOB	INPUT	LVCMOS25					
xin1<3>	IOB	INPUT	LVCMOS25					
xin1<4>	IOB	INPUT	LVCMOS25					
xin1<5>	IOB	INPUT	LVCMOS25					
xin1<6>	IOB	INPUT	LVCMOS25					
xin1<7>	IOB	INPUT	LVCMOS25					
xin1<8>	IOB	INPUT	LVCMOS25					
xin1<9>	IOB	INPUT	LVCMOS25					
xin1<10>	IOB	INPUT	LVCMOS25					
xin1<11>	IOB	INPUT	LVCMOS25					
xin1<12>	IOB	INPUT	LVCMOS25					
xin1<13>	IOB	INPUT	LVCMOS25					
xin1<14>	IOB	INPUT	LVCMOS25					
xin1<15>	IOB	INPUT	LVCMOS25					
xin1<16>	IOB	INPUT	LVCMOS25					
xin1<17>	IOB	INPUT	LVCMOS25					
xin1<18>	IOB	INPUT	LVCMOS25					
xin1<19>	IOB	INPUT	LVCMOS25					
xin1<20>	IOB	INPUT	LVCMOS25					
xin1<21>	IOB	INPUT	LVCMOS25					
xin1<22>	IOB	INPUT	LVCMOS25					
xin1<23>	IOB	INPUT	LVCMOS25					
xin1<24>	IOB	INPUT	LVCMOS25					
xin1<25>	IOB	INPUT	LVCMOS25					
xin1<26>	IOB	INPUT	LVCMOS25					
xin1<27>	IOB	INPUT	LVCMOS25					
xin1<28>	IOB	INPUT	LVCMOS25					
xin1<29>	IOB	INPUT	LVCMOS25					
xin1<30>	IOB	INPUT	LVCMOS25					
xin1<31>	IOB	INPUT	LVCMOS25					
xin1<32>	IOB	INPUT	LVCMOS25					
xin1<33>	IOB	INPUT	LVCMOS25					
xin1<34>	IOB	INPUT	LVCMOS25					
xin1<35>	IOB	INPUT	LVCMOS25					
xin1<36>	IOB	INPUT	LVCMOS25					

xin1<37>	IOB	INPUT	LVCMOS25					
xin1<38>	IOB	INPUT	LVCMOS25					
xin1<39>	IOB	INPUT	LVCMOS25					
xin1<40>	IOB	INPUT	LVCMOS25					
xin1<41>	IOB	INPUT	LVCMOS25					
xin1<42>	IOB	INPUT	LVCMOS25					
xin1<43>	IOB	INPUT	LVCMOS25					
xin1<44>	IOB	INPUT	LVCMOS25					
xin1<45>	IOB	INPUT	LVCMOS25					
xin1<46>	IOB	INPUT	LVCMOS25					
xin1<47>	IOB	INPUT	LVCMOS25					
xin1<48>	IOB	INPUT	LVCMOS25					
xin1<49>	IOB	INPUT	LVCMOS25					
xin1<50>	IOB	INPUT	LVCMOS25					
xin1<51>	IOB	INPUT	LVCMOS25					
xin1<52>	IOB	INPUT	LVCMOS25					
xin1<53>	IOB	INPUT	LVCMOS25					
xin1<54>	IOB	INPUT	LVCMOS25					
xin1<55>	IOB	INPUT	LVCMOS25					
xin1<56>	IOB	INPUT	LVCMOS25					
xin1<57>	IOB	INPUT	LVCMOS25					
xin1<58>	IOB	INPUT	LVCMOS25					
xin1<59>	IOB	INPUT	LVCMOS25					
xin1<60>	IOB	INPUT	LVCMOS25					
xin1<61>	IOB	INPUT	LVCMOS25					
xin1<62>	IOB	INPUT	LVCMOS25					
xin1<63>	IOB	INPUT	LVCMOS25					
zin1<0>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<1>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<2>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<3>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<4>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<5>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<6>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<7>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<8>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<9>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<10>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<11>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<12>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<13>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<14>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<15>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<16>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<17>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<18>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<19>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<20>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<21>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<22>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<23>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<24>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<25>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<26>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<27>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<28>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<29>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<30>	IOB	INPUT	LVCMOS25			INFF1		IFD

zin1<31>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<32>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<33>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<34>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<35>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<36>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<37>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<38>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<39>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<40>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<41>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<42>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<43>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<44>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<45>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<46>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<47>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<48>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<49>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<50>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<51>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<52>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<53>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<54>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<55>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<56>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<57>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<58>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<59>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<60>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<61>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<62>	IOB	INPUT	LVCMOS25			INFF1		IFD
zin1<63>	IOB	INPUT	LVCMOS25			INFF1		IFD

Section 7 - RPMs

Section 8 - Guide Report

Guide not run on this design.

Section 9 - Area Group Summary

No area groups were found in this design.

Section 10 - Modular Design Summary

Modular Design not used for this design.

Section 11 - Timing Report

This design was not run using timing mode.

Section 12 - Configuration String Details

Use the "-detail" map option to print out Configuration Strings

APENDICE K: Importación de Cores en Simulink.

Este trabajo uso el CORE de XILINX, para operaciones de punto flotante, como se muestra en la figura K1.

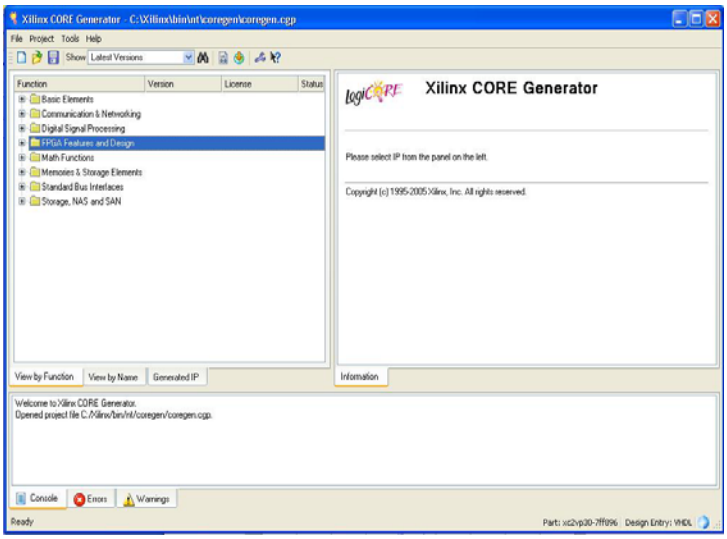


Figura K1. Pantalla principal del CORE generator de Xilinx.

Después de crear nuestra carpeta, en donde vamos a colocar nuestros COREs y donde vamos a llamarlos desde Matlab, empezamos a designar las operaciones y los nombres para nuestros trabajos, tal como se muestra en la figura K2.

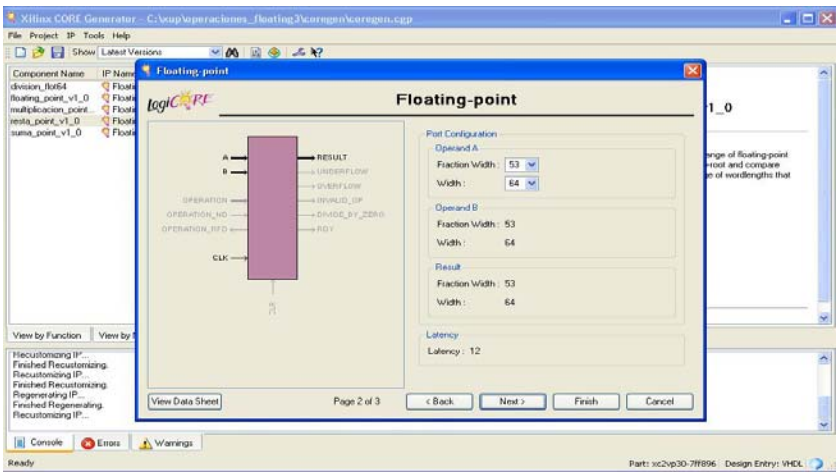


Figura K2. CORE de punto flotante.

En la carpeta que creamos tenemos que crear los archivos .vhd, para poder exportar estos ejemplos al Matlab, a estos core los tenemos que llamar como componentes, como el código siguiente.

```
----------------------------------------------------------------
-----------
-- Company: Universite Ricardo Palma
-- Engineer: Eng. Fernando Raymundo
-- Create Date:    16:39:24 02/21/2008
-- Design Name:
-- Module Name:    division_segundo- Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
-- Dependencies:
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
----------------------------------------------------------------
-----------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if
instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity suma_float is
    Port ( input_ce : in std_logic;
           ain : in  std_logic_VECTOR(63 downto 0);
```

```
           bin : in  std_logic_VECTOR(63 downto 0);
           clck : in  STD_LOGIC;
           resultados  :  out    STD_LOGIC_VECTOR(63  downto
0));
end suma_float;

architecture Behavioral of suma_float is

component suma_point_v1_0
     port (
     a: IN std_logic_VECTOR(63 downto 0);
     b: IN std_logic_VECTOR(63 downto 0);

     clk: IN std_logic;
     result: OUT std_logic_VECTOR(63 downto 0));

end component;

begin

u1 : suma_point_v1_0

          port map (clk=>clck,
                       a=>ain,b=>bin,result=>resultados);

end Behavioral;
```

Para poder importarlos desde simulink, tenemos que añadir a nuestro trabajo los black box, como se muestra en la figura K3.

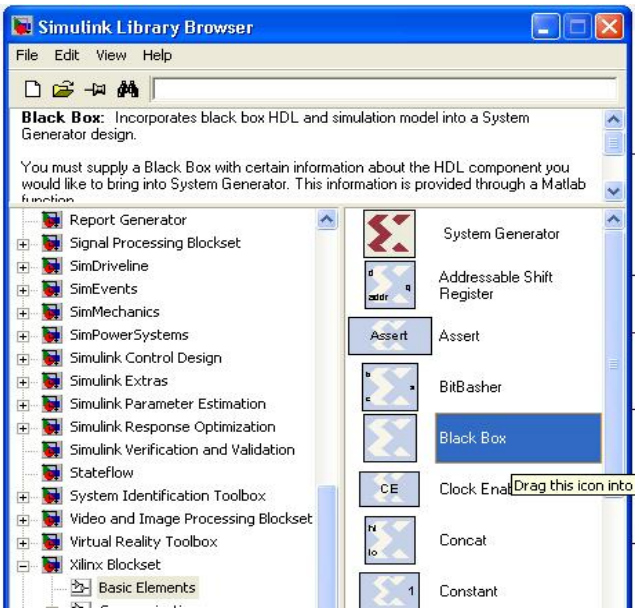


Figura K3. Black Box de system generador.

Cuando se añade el black box, se da doble clic y nos aparece este archivo en .m, como se muestra en la figura

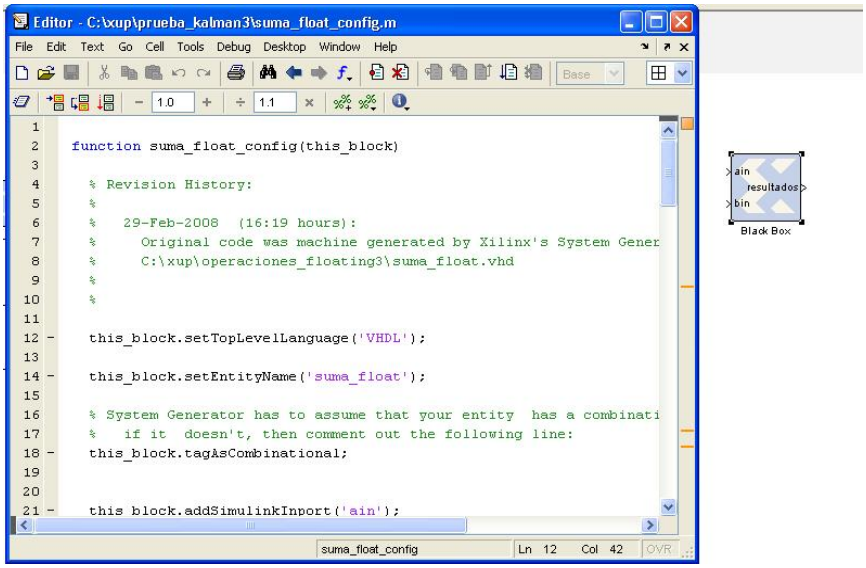


Figura K4. Archivo .m para el black box.

Por ultimo se debe completar el archivo .m para poder hacer funcionar correctamente el hardware importado desde simulink, se debe copiar este modelo para todas las circuitos importados.

```
function suma_float_config(this_block)

  % Revision History:
  %
  %   29-Feb-2008  (16:19 hours):
  %      Original code was machine generated by Xilinx's
System Generator after parsing
  %     C:\xup\operaciones_floating3\suma_float.vhd

  this_block.setTopLevelLanguage('VHDL');
  this_block.setEntityName('suma_float');
  % System Generator has to assume that your entity  has a
combinational feed through;
  %   if it  doesn't, then comment out the following line:
  this_block.tagAsCombinational;

  this_block.addSimulinkInport('ain');
  this_block.addSimulinkInport('bin');
  this_block.addSimulinkOutport('resultados');
  resultados_port = this_block.port('resultados');

  % ------------------------------
  if (this_block.inputTypesKnown)
    % do input type checking, dynamic output type and
generic setup in this code block.

    if (this_block.port('ain').width ~= 64);
      this_block.setError('Input data type for port "ain"
must have width=64.');
    end

    if (this_block.port('bin').width ~= 64);
      this_block.setError('Input data type for port "bin"
must have width=64.');
```

```
    end

  end  % if(inputTypesKnown)
  % -----------------------------
  % -----------------------------
   if (this_block.inputRatesKnown)
     inputRates = this_block.inputRates;
     uniqueInputRates = unique(inputRates);
     outputRate = uniqueInputRates(1);
     for i = 2:length(uniqueInputRates)
       if (uniqueInputRates(i) ~= Inf)
         outputRate = gcd(outputRate,uniqueInputRates(i));
       end
     end  % for(i)
     for i = 1:this_block.numSimulinkOutports
       this_block.outport(i).setRate(outputRate);
     end  % for(i)
   end  % if(inputRatesKnown)
  % -----------------------------

  % Add addtional source files as needed.
  %  |-------------
  %   | Add files in the order in which they should be
compiled.
  %   | If two files "a.vhd" and "b.vhd" contain the
entities
  %  | entity_a and entity_b, and entity_a contains a
  %  | component of type entity_b, the correct sequence of
  %  | addFile() calls would be:
  %  |    this_block.addFile('b.vhd');
  %  |    this_block.addFile('a.vhd');
  %  |-------------

  %    this_block.addFile('');
```

```
  %    this_block.addFile('');
  this_block.addFile('suma_float.vhd');
  this_block.addFile('coregen\suma_point_v1_0.ngc');
  this_block.addFile('coregen\suma_point_v1_0.vhd');

return;
```

Con ello se garantiza el funcionamiento del filtro con todos sus circuitos.

REFERENCIAS BIBLIOGRÁFICAS

Fuentes Bibliográficas

Grewal, Mohinder S., and Angus P. Andrews. Kalman Filtering Theory and Practice. Prentice Hall, 1993.

Kalman, R. E., A New Approach to Linear Filtering and Prediction Problems, Transaction of the ASME—Journal of Basic Engineering, pp. 35-45 (March 1960), 1960.

Maybeck, Peter S., Stochastic Models, Estimation, and Control, Volume 1, Academic Press, Inc. ,1979.

Sorenson, H. W., Least-Squares estimation: from Gauss to Kalman, *IEEE Spectrum*, vol. 7, pp. 63-68, 1970.

Leonard A. McGee, Stanley Schmidt , Discovery of the Kalman Filter as a practical Tool for aerospace and industry". Nasa echnical Memorandum 86847.

Greg Welch, Gary Bishop, An introduction to the kalman Filter, Siggraph, 2001.

Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing,
Hard Cover, 1997.

Volnei A. Pedroni, Circuit Design With VHDL, Editorial MIT Press, 2004.

A. Granados Ly, Diseño Digital con VHDL, Inictel, 2004.

Fuentes de Información Electrónica

E XILINX, http://www.xilinx.com.

THE MATH WORKS, http://www.mathworks.com/

INTER – SYSTEM GROUND
,http://www.apcmedia.com/salestools/FLUU-5T3TLT_R1_EN.pdf/

NATIONAL INSTRUMENTS, http://www.ni.com/

CONIDA, Agencia Espacial del Perú, http://www.conida.gob.pe/