

ADS-B Receiver implementation in Verilog

Implementación de un receptor ADS-B en Verilog

Justo Garay de la Rosa ^{*1}, Víktor Iván Rodríguez Abdlá ¹, Salvador Ibarra Delgado ¹, Remberto Sandoval Aréchiga ¹, José Ricardo Gómez Rodríguez ¹, and Óscar Osvaldo Ordaz García ¹

¹Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica,
Posgrado en Ingeniería para la Innovación Tecnológica,
Siglo XXI, La Escondida, Zacatecas, Zacatecas, México, 98160.
{j.garay, abdala, sibarra, rsandoval, jrgridri, oscarordaz} @uaz.edu.mx

Abstract

In this work, an ADS-B receiver system was implemented on an FPGA using the Verilog hardware description language. The work is divided into two phases: the first involves the implementation of the ADS-B receiver using software tools, which are necessary to translate the functional blocks of the system; the second involves transferring these blocks to hardware, utilizing the Zybo Z7 board and the Verilog development platform.

Keywords— ADS-B, FPGA, Verilog

Resumen

En el presente artículo se describe la implementación de un sistema receptor ADS-B en un FPGA utilizando el lenguaje de descripción de hardware Verilog. El desarrollo se divide en dos fases: la primera consiste en la implementación del receptor ADS-B mediante software, con la finalidad de analizar los bloques del sistema; y la segunda en implementar dichos bloques a hardware, utilizando la tarjeta Zybo Z7 y la plataforma de desarrollo Verilog.

Palabras clave— ADS-B, FPGA, Verilog

I. Introducción

El control aéreo ha evolucionado considerablemente desde sus inicios en la década de 1920, cuando se introdujeron las primeras estaciones de radio para comunicarse con las aeronaves. A lo largo de los años, el radar se convirtió en la piedra angular de la vigilancia aérea, permitiendo a los controladores aéreos rastrear a las aeronaves. Sin embargo, este tiene limitaciones en cuanto a cobertura, precisión y capacidad, lo que motivó la búsqueda de tecnologías más avanzadas.

El concepto de ADS-B (siglas en inglés de Sistema de Vigilancia Dependiente Automática) surgió en la década de 1990, este permite obtener, entre otros parámetros, la posición, velocidad y altitud, de la aeronave. Su adopción

fue impulsada por la Administración Federal de Aviación de los Estados Unidos en 2010.

Existen varios tipos de receptores ADS-B, cada uno con funciones específicas: a) ADS-B In, estos proveen al piloto de la aeronave información de posicionamiento, b) ADS-B Out, estos transmiten la posición de la aeronave a estaciones terrestres y otras aeronaves, c) receptores terrestres, estos recogen los datos para los controladores de tráfico aéreo, d) receptores de mano, estos son portátiles y utilizados por aficionados y pilotos privados, e) receptores integrados, estos mejoran la seguridad en sistemas aéreos, f) receptores basados en satélites, estos proporcionan cobertura global, y g) receptores SDR (siglas en inglés de Radio Definido por Software) que ofrecen flexibilidad y capacidad de actualización.

Este trabajo se centra en la implementación de un receptor ADS-B en un FPGA (siglas en inglés de Arreglos de

* Autor de correspondencia

II.3.2. Diagrama de bloques

El receptor ADS-B se observa en la Fig. 3.

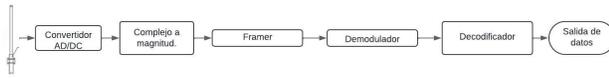


Figura 3: Diagrama de bloques de receptor ADS-B

Esta se compone de las siguientes etapas:

1. Antena

Se encarga de la captura de la señal de 1090 MHz.

2. Convertidor AD/DC

La señal capturada se convierte de analógica a una señal digital compuesta por datos I/Q de 8 bits.

3. Complejo a magnitud

En este bloque se calcula la magnitud de la señal recibida utilizando la Ec. (1).

$$\|\vec{r}\| = \sqrt{r_x^2 + r_y^2} \quad (1)$$

4. Framer

Este bloque consiste de:

- **Umbral:** Las muestras de la señal se comparan con un umbral previamente definido, se determina que el dato de entrada sea igual o mayor al umbral. Si el dato cumple con la condición del umbral propuesto entonces el valor del dato entrará como tal, de lo contrario a este se le asigna un valor igual a cero.
- **Detección de 1 y 0:** Las muestras pasan por un detector de amplitud con el cual toma la decisión si una muestra es 1 o 0.
- **Detección de preámbulo:** Las muestras obtenidas por el detector de 1 y 0 se introducen a un correlador para detectar el preámbulo 1010000101000000.
- **Detección y almacenamiento:** Al detectar un preámbulo, se dejarán pasar 224 muestras correspondientes a los 112 bits del mensaje ADS-B.

5. Demodulador

Las 224 muestras enviadas por el **Framer** se comparan en amplitud por pares, de acuerdo a la Ec. (2), para obtener 112 bits.

$$Bit[n] = \begin{cases} 1, & \text{si } Muestra1 > Muestra2 \\ 0, & \text{si } Muestra2 > Muestra1 \end{cases} \quad (2)$$

6. Decodificador

El decodificador se divide en:

- **CRC (siglas en inglés de Código de Redundancia Cíclica):** Es un algoritmo basado en aritmética polinomial binaria, se utiliza un divisor predefinido (también conocido como generador) para calcular los bits de paridad mediante la división polinómica binaria [2].
- **Dirección ICAO:** En cada mensaje ADS-B se puede identificar la aeronave emisora a través del código del transceptor Modo S, asignado de acuerdo con la normativa de la ICAO. Este código, también conocido como dirección ICAO, se encuentra entre el bit 9 y 32 en formato binario o en la posición 3 a 8 en formato hexadecimal.
- **Identificador de vuelo:** Si el tipo de mensaje (primeros 5 bits del mensaje) está entre un valor del 1 al 4 significa que el mensaje ADS-B aparte de contener la dirección ICAO tendrá el identificador del vuelo.
- **Posición aérea:** El mensaje de posición aérea puede variar dependiendo de que si la altitud es codificada con GNSS (siglas en inglés de Sistema Global de Navegación por satélites) o altitud por altímetro barométrico

7. Salida de datos

Se presenta el mensaje en un formato legible para el usuario.

III. Desarrollo

Este tiene dos enfoques. El primero consiste en la implementación del receptor ADS-B en software y el segundo es la implementación de los módulos desarrollados en hardware.

III.1. ADS-B en software

El objetivo de la implementación del receptor ADS-B en software es analizar el funcionamiento de diferentes etapas en el receptor, así como comparar los datos de entrada y salida de cada etapa utilizando la misma señal capturada para la implementación, tanto en software como en hardware.

ADS-B con uso de GNU Radio

El receptor ADS-B en GNU Radio almacena la señal demodulada en un archivo para post-procesamiento, se usó `gr_adsb` con [5] y la implementación se ve en la Fig. 4.

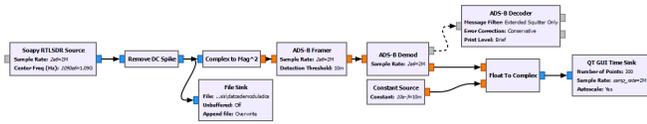


Figura 4: Sistema ADS-B en GNU Radio

ADS-B con uso de software

Para la implementación en software se desarrollaron funciones en base a los bloques de la implementación de GNU Radio. Se tomaron en cuenta los siguientes parámetros:

- Ancho de banda = 1 MHz
- Frecuencia de muestreo = 2 MHz
- Muestras por símbolo = 2
- Umbral = 0.001

A continuación se muestra el diagrama de flujo de la implementación en la Fig.5

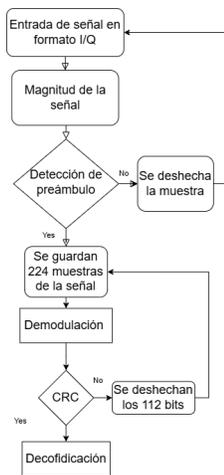


Figura 5: Diagrama de flujo

III.2. ADS-B en hardware

La implementación en hardware se realiza mediante la plataforma Vivado con lenguaje Verilogm [6]. El procedimiento consiste en la traducción de los bloques de la implementación en GNU Radio y funciones en la implementación en software a módulos de hardware implementados en Verilog. El receptor tiene dos entradas de datos de 16 bits pertenecientes a la parte real e imaginaria de la señal capturada, una señal de inicio (start), una señal de reloj (clk), un reinicio (rst) y cuatro salidas de 112 bits pertenecientes al tipo información de la señal ADS-B como se muestra en la Fig. 6.



Figura 6: Diagrama de caja negra receptor ADS-B

El receptor consta de varios módulos como se aprecia en la Fig. 7.

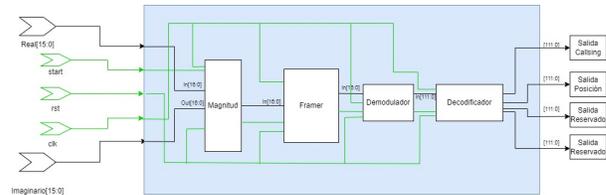


Figura 7: Módulo receptor ADS-B

1. **Módulo magnitud:** Módulo encargado de procesar la magnitud de la señal, con entradas de 16 bits para las componentes real e imaginaria, señales de control (start, rst y clk), una salida de estado (done) y una salida de 16 bits (Magnitud) como se aprecia en la Fig. 8.



Figura 8: Diagrama de caja negra Magnitud

La implementación se lleva a cabo a través de diferentes sub-módulos como se muestra en la Fig. 9, estos son encargados del procesamiento de cada muestra.

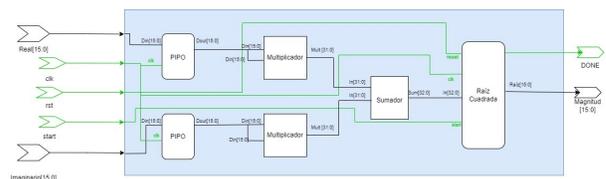


Figura 9: Módulo Magnitud

2. **Módulo Framer:** El módulo Framer tiene una entrada de 16 bits la cual es la magnitud de cada muestra, tres señales de control (start, rst y clk), y dos salidas: una de 16 bits (muestras pertenecientes de

mensajes AS-B) y una señal de estado (Done), como se muestra en la Fig. 10.



Figura 10: Módulo Framer

El módulo está constituido por diferentes sub-módulos como se muestra en la Fig. 11.

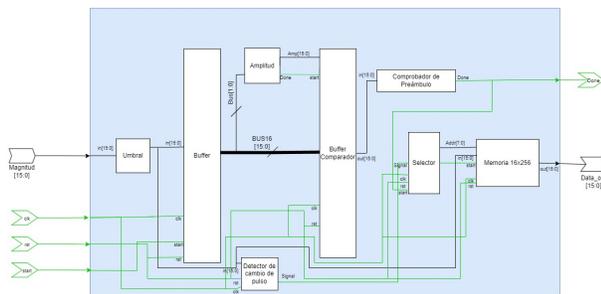


Figura 11: Módulo Framer

- Módulo Demodulador:** Las 224 muestras de 16 bits guardados en la memoria RAM del *Framer* se reciben en este módulo y son demoduladas mediante una toma de decisión de muestras utilizando la Ec. (2), teniendo como resultado una salida de 112 bits. El módulo se muestra en la Fig. 12.

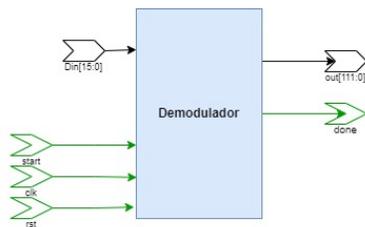


Figura 12: Módulo Demodulador

- Módulo Decodificador:** El módulo Decodificador cuenta con una entrada de 112 bits proporcionada por el *Demodulador*, tres señales de control (*start*, *rst* y *clk*), y cuatro salidas de 112 bits, cada una correspondiente a un tipo de mensaje, como se muestra en la Fig. 13. Las funciones principales de este módulo son la detección de errores mediante CRC, la identificación



Figura 13: Diagrama de caja negra Decodificador.

del tipo de mensaje y su decodificación. El módulo se divide en seis sub-módulos como se muestra en la Fig. 14.

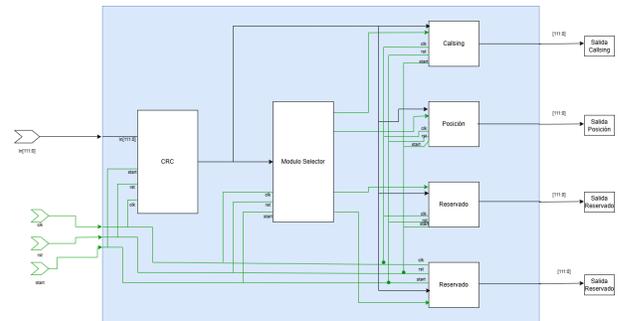


Figura 14: Módulo Decodificador

IV. Resultados

Se capturó una señal ADS-B con datos de dos aeronaves: una de la que solo se obtuvieron mensajes de tipo posición y otra con mensajes de tipo reservado. En la parte de hardware solo se decodificaron los datos de la primera aeronave, esto para optimizar el tamaño de las memorias ROM.

IV.1. GNU Radio

El receptor ADS-B de GNU Radio procesó la señal en tiempo real para la visualización de las aeronaves, obteniendo los resultados presentados en las figuras 15 y 16.

El primer mensaje ADS-B, muestra una aeronave con etiqueta ICAO *a6b253*, un tipo de mensaje posición (11), una latitud de 22.64° , una longitud de -102.69° y una altitud de 36050 pies , el segundo mensaje es un mensaje tipo reservado (29) y una etiqueta ICAO *aaf007*.

```
[INFO] -----
[INFO] Datetime: 2024-10-24 23:37:08.079519 UTC
[INFO] SNR: 32.54 dB
[INFO] Downlink Format (DF): 17 Extended Squitter
[INFO] Capability (CA): 5 Level 2 or Above Transponder, Can Set CA 7, In Air
[INFO] Address Announced (AA): a6b253
[INFO] Callsign: None
[INFO] Type Code (TC): 11 Airborne Position
[INFO] Surveillance Status (SS): 0 No Condition Information
[INFO] Time: 0 Not Synced to 0.2s UTC Epoch
[INFO] Latitude: 22.640533447265625 N
[INFO] Longitude: -102.69640003551137 E
[INFO] Altitude: 36050 ft
[INFO] -----
```

Figura 15: Aeronave mensaje tipo Posición

```
[INFO] -----
[INFO] Datetime: 2024-10-24 23:36:58.211691 UTC
[INFO] SNR: 22.40 dB
[INFO] Downlink Format (DF): 17 Extended Squitter
[INFO] CRC: Passed
[INFO] -----
[INFO] Datetime: 2024-10-24 23:36:58.211691 UTC
[INFO] SNR: 22.40 dB
[INFO] Downlink Format (DF): 17 Extended Squitter
[INFO] Capability (CA): 5 Level 2 or Above Transponder, Can Set CA 7, In Air
[INFO] Address Announced (AA): aaF007
[INFO] Callsign: None
[INFO] Type Code (TC): 29 Reserved
[DEBUG] TC: 29 To be implemented
[DEBUG] DF: 12 Unknown DF
[INFO] -----
```

Figura 16: Aeronave mensaje tipo Reservado

IV.2. Receptor de software

La señal ADS-B se procesó con software, la visualización de los datos se muestran en la Fig. 17.

```
ICAO extraídos son: AAF007
Tipo de mensaje: 29
ICAO extraídos son: A6B253
Latitud: 22.6410
Altitud 36050
Longitud: -102.6913
Tipo de mensaje: 11
```

Figura 17: Resultados en software

Los resultados muestran dos aeronaves, una con etiqueta ICAO AAF007, un tipo de mensaje reservado (29) y la otra aeronave con una etiqueta ICAO A6B253, un mensaje de tipo posición (11), una latitud de 22.64°, longitud de -102.69° y una altitud de 36050 *pies*.

IV.3. Hardware

A continuación, se presentan los resultados del receptor ADS-B.

1. **ICAO:** Etiqueta ICAO perteneciente a la aeronave en este caso *a6b253*, el resultado se muestra en la Fig. 18.

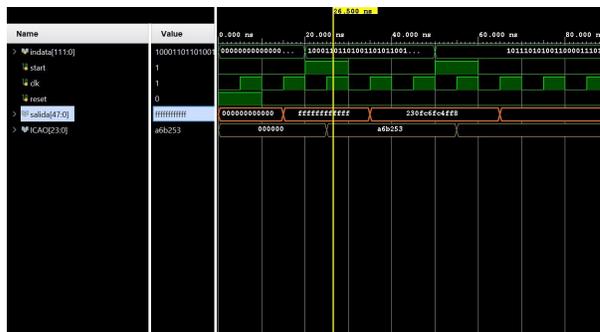


Figura 18: Resultado Demodulador ADS-B

2. **Posición:** La aeronave detectada tiene una latitud de 22.64°, longitud de -102.69° y altitud de 36050 *pies*, el resultado se muestra en la Fig. 19.



Figura 19: Resultado Posición ADS-B

Se observan 2 mensajes de cuatro salidas de 32 bits cada una para estimar la posición de la aeronave, los resultados de estos cálculos se muestran en la Fig. 20.

```
*****POSICION DE AERNAVE*****
Longitud de la aeronave: -102.691307

Latitud de la aeronave: 22.637833

Altitud de la aeronave: 36050.000000 pies

*****
Process exited after 0.04681 seconds with return value 0
Presione una tecla para continuar . . .
```

Figura 20: Resultado Posición ADS-B

3. **CALLSING:** El número de vuelo de la aeronave (mensaje previamente conocido) es [KLM1023], en la Fig. 21 se muestra el resultado obtenido.

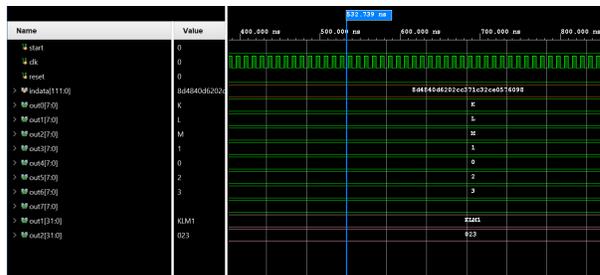


Figura 21: Resultado Calling ADS-B

Se observa que el mensaje conocido `0x8D4840D6202CC371C32CE0576098` es decodificado correctamente obteniendo la llamada *Callsing* y este mensaje es guardado en dos registros de 32 bits.

4. **Latencia:** El sistema receptor ADS-B presenta una latencia de $30.295 \mu s$ desde la detección del preámbulo hasta la decodificación de los mensajes, como se muestra en la Fig. 22.

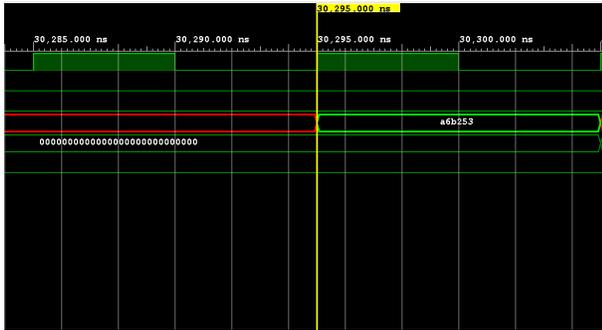


Figura 22: Resultado Latencia

V. Conclusiones

La evaluación de los resultados obtenidos del módulo ADS-B diseñado e implementado en hardware demuestran que es posible capturar una señal ADS-B en 1090 MHz para su demodulación y decodificación en una plataforma hardware, aunque con algunas limitaciones.

El hecho de que la señal ADS-B deba procesarse en punto fijo representa importantes limitaciones debido al hardware, en comparación al uso de punto flotante, esto implica que diversas herramientas aritméticas empleadas en el procesamiento de señales, como radicales, exponenciales, operaciones logarítmicas y funciones trigonométricas, deban implementarse mediante aproximaciones numéricas o en su defecto, no implementarse. Como consecuencia, puede incrementarse el uso de recursos de la FPGA o presentarse variaciones en los resultados. Esto también afecta la decodificación de los mensajes obtenidos de la señal, teniendo así que recurrir al empleo de herramientas de software para completar la decodificación.

El procesamiento de señales ADS-B en hardware presenta ventajas en comparación con el software, debido a su capacidad para realizar operaciones en paralelo y al uso de máquinas de estados. Esto se traduce en una menor latencia para la recepción y decodificación de los mensajes, pero también conlleva un mayor consumo de recursos en la tarjeta de desarrollo. Sin embargo, no fue posible medir de manera precisa las métricas de consumo de recursos, ya que el trabajo se

realizó en un entorno de simulación.

Agradecimientos

Agradecimientos dirigidos a la Universidad Autónoma de Zacatecas que con su contribución en términos de recursos, instalaciones y conocimiento especializado ha sido fundamental para el desarrollo de esta investigación.

Referencias

- [1] ICAO. «Guía de vigilancia global». En: ICAO (2014).
- [2] Junzi Sun. *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*. 2.^a ed. TU Delft OPEN Publishing, 2021. ISBN: 978-94-6366-402-8. DOI: 10.34641/mg.11.
- [3] Alicia Parra Ruiz. «Sistema de multilateración Modo S». Universidad Politécnica de Madrid, sep. de 2017.
- [4] Gianluca Chiesa Pastor. «Desarrollo de un sistema de recepción para señales ADS-B.» Tesis doct. Universidad Nacional de Cuyo, 2017.
- [5] Michael Hostetter y contributors. *gr-adsb: An open-source ADS-B receiver implementation*. Accessed: 2025-01-08. 2025. URL: <https://github.com/mhostetter/gr-adsb>.
- [6] Digilent, Inc. *Zybo Z7 Reference Manual*. Recuperado el 01 de noviembre, 2024. 2023. URL: <https://digilent.com/reference/programmable-logic/zybo-z7/reference-manual>.