Instituto Politécnico Nacional
Escuela Superior de Cómputo
Sección de Estudios de Posgrado e Investigación

Tesis

Arquitectura de muestreo compreso con matrices estructuradas

Que para obtener el grado de

Maestro en Ciencias en Sistemas Computacionales Móviles

Presenta

Ing. Saúl Daniel Ángeles Vázquez

Directores

Dr. David Ernesto Troncoso Romero
Dr. Manuel Salazar Ramírez

ESCOM

Ciudad de México a 11 de diciembre del 2018

SIP-14-BIS

INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 11:00 horas del día 4 del mes de diciembre del 2018 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de SEPI - ESCOM para examinar la tesis titulada:

"Arquitectura de muestreo compreso con matrices estructuradas"

Presentada por el alumno:

Ángeles	**Vázquez**	**Saúl Daniel**
Apellido paterno	Apellido materno	Nombre(s)

Con registro: A 1 7 0 4 1 5

aspirante de:

Maestría en Ciencias en Sistemas Computacionales Móviles

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de tesis

Dr. Manuel Salazar Ramírez
Vocal 1

Dr. David Ernesto Troncoso Romero
Vocal 2

M. en C. Chadwick Carreto Arellano
Presidente

M. en C. Rodolfo Romero Herrera
Secretario

M. en C. Jesús Alfredo Martínez Nuño
Vocal 3

PRESIDENTE DEL COLEGIO DE PROFESORES

Lic. Andrés Ortigoza Campos

S.E.P.
INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE COMPUTO

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESION DE DERECHOS

En la **Ciudad de México, CDMX.**, el día **11** del mes de **diciembre** del año **2018**, el que suscribe **Ing. Saúl Daniel Ángeles Vázquez** alumno del Programa de **Maestría en Ciencias en Sistemas Computacionales Móviles** con número de registro **A170415**, adscrito a la **Escuela Superior de Cómputo**, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del **Dr. David Ernesto Troncoso Romero** y **Dr. Manuel Salazar Ramírez** y cede los derechos del trabajo intitulado **Arquitectura de muestreo compreso con matrices estructuradas**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciónes sangeless0000@alumno.ipn.mx, detroncosoro@conacyt.mx y msalazarra@ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Ing. Saúl Daniel Ángeles Vázquez

Resumen

Este trabajo presenta un par de arquitecturas digitales basadas en autómatas celulares caóticos (*Cellular Automata Chaos, CAC*) para muestreo compreso (*Compressed Sensing, CS*) de señales electrocardiográficas. La primera arquitectura ha sido desarrollada tomando como punto de partida el reciente algoritmo de muestreo compreso propuesto en la literatura, llamado Autómata Celular Caótico con Umbral de la Señal Original (*Cellular Automata Chaos with Original Signal Thresholding, CAC-OST*). Para su construcción, hemos dividido dicho algoritmo en cinco sencillos módulos digitales. La segunda arquitectura, que hemos llamado Autómata Celular Caótico para Señales Ininterrumpidas (*Cellular Automata Chaos for Streaming Signals*), ha sido diseñada de manera adecuada para procesar señales que fluyen de forma ininterrumpida. Estas arquitecturas digitales son implementadas en un FPGA Cyclone IV de Altera® y comparadas contra el estado del arte de sistemas de muestreo compreso para señales electrocardiográficas. Para cada arquitectura digital reportamos el número de Elementos Lógicos (*Logic Elements, LE*), registros, bits de memoria y frecuencia máxima de operación. La señal comprimida es recuperada en Matlab® y comparada con la señal original, la tasa de error y la relación de señal a ruido entre la señal recuperada y la señal original es calculada vía Error Cuadrático Medio (*Mean Squared Error, MSE*) y Razón Señal a Ruido (*Signal-to-Noise Ratio*). Nuestros resultados experimentales muestran que CAC-OST tiene mejores características de compresión, pero a expensas de una mayor utilización de *hardware*, mientras que CAC-SS es la más simple entre estas arquitecturas.

Abstract

This paper presents a couple of digital architectures based on chaotic cellular automata for Compressed Sensing (CS) of electrocardiographic (ECG) signals. The first architecture has been designed taking as starting point the recent CS algorithm proposed in literature, called Cellular Automata Chaos with Original Signal Thresholding (CAC-OST). For its construction, we have split that algorithm into five simple digital modules. The second architecture, which we call Cellular Automata Chaos for Streaming Signals (CAC-SS), has been designed in such a way that can be adequate for streaming processing. These digital architectures are implemented in FPGA Cyclone IV from Altera® and compared against the state-of-the-art ECG CS system. For each digital architecture we report the number of Logic Elements (LE), registers, memory bits and maximum frequency of operation. The compressed signal is recovered in Matlab® and compared with the original signal, the error rate and the level of signal to noise between recovered signal and original is calculated via Mean Sqared Error (MSE) and Signal-to-Noise Ratio (SNR). Our experimental results show that CAC-OST has better compression characteristics but at expenses of a greater hardware utilization, whereas CAC-SS is the simplest one among these architectures.

Agradecimientos

Gracias Dios mío. Eres al primero a quien debo agradecer. Sólo tú sabes lo que he tenido que pasar para poder llegar hasta aquí. Gracias por permitirme no desfallecer. Gracias por darme el coraje necesario en todo momento para superar las adversidades.

Gracias al Consejo Nacional de Ciencia y Tecnología (CONACYT) por brindarme el apoyo económico mediante la beca No. 726363. Agradezco particularmente al Instituto Politécnico Nacional (IPN), mi querida alma mater, por el apoyo económico brindado. También le agradezco a la Escuela Superior de Cómputo (ESCOM) por brindarme la infraestructura adecuada para la realización de esta investigación.

Mi profundo y sincero reconocimiento al Dr. David Ernesto Troncoso Romero, quien depositó en mí su confianza para realizar esta investigación. Gracias por guiarme con sabiduría, pericia y sobre todo por compartir tu vasto conocimiento Doc. Mi agradecimiento al Dr. Manuel Salazar Ramírez por aceptar integrarse al equipo y co-supervisar la investigación. Gracias por el tiempo dedicado y especialmente por la asesoría en los trámites administrativos. Agradezco a mis sinodales por las observaciones y los comentarios brindados en cada una de las evaluaciones realizadas. También agradezco a los Investigadores y Catedráticos de ESCOM y del Centro de Investigación en Computación (CIC) por los cursos impartidos durante el desarrollo de la investigación.

Gracias a mi familia, especialmente a mis padres y hermano. Les agradezco el inmenso apoyo que me han brindado para culminar esta importante etapa en mi vida. Gracias a Karene, mi novia, por el apoyo y la paciente espera que ha brindado. Dios nos recompensará el tiempo que no hemos podido compartir juntos.

Me encuentro muy agradecido con las personas que conocí en el transcurso de esta investigación. Especialmente con quienes empezaron siendo compañeros y ahora los considero amigos. Gracias a Kev, Gonza, Izya y Abbey por hacer que estos dos años hayan sido muy amenos y apoyarme en temas escolares y personales. Charlie, eres un *crack*, gracias por cada duda resuelta. Gus, Tocayo, Juan José, Ángel, Camacho, Janeth, Capi, Ana, Isra, Juan Jesús y Nes, gracias por el tiempo compartido.

Finalmente, pero no menos importante, le agradezco a los administrativos de ESCOM. Gracias Dra. Blanca, por guiarme en cada trámite durante tu gestión. Gracias a los maestros Araujo y Nuño, así como a Quique y Gaby, por apoyarnos con cada solicitud y realizar el seguimiento adecuado.

Índice

Resumen ..... **IV**
Abstract ..... **V**
Agradecimientos ..... **VI**
Índice ..... **VII**
Lista de tablas ..... **IX**
Lista de figuras ..... **XI**
Capítulo 1: Introducción **1**
1.1 Teorema de Muestreo 2
1.2 Muestreo compreso 3
1.3 Motivación 8
1.4 Problemática a resolver 8
1.5 Pregunta de investigación 9
1.6 Objetivos 9
1.7 Justificación 10
1.8 Clasificación de la investigación 10
Capítulo 2: Arquitecturas de muestreo compreso: estado del arte **12**
2.1 Diseño y análisis de arquitectura de muestreo compreso para compresión de datos en sensores inalámbricos 12
2.2 Consideraciones para muestreo compreso en sensores inalámbricos de señales ECG y EMG 13
2.3 Muestreo compreso para monitoreo remoto de señales EEG 14
2.4 Implementación FPGA de muestreo compreso para compresión de señales ECG 15
2.5 Generación de matriz de muestreo con autómata celular caótico 16
2.6 Marco de referencia 17
2.6.1 Muestreo compreso por *software* 18
2.6.2 Arquitectura digital [13] 20
Capítulo 3: Algoritmo y arquitecturas digitales: métodos propuestos **32**
3.1 Arquitectura digital CAC-OST 32
3.1.1 Memoria 32
3.1.2 Umbral 34
3.1.3 Semilla inicial 35
3.1.4 Generador de la matriz de muestreo **Φ** 37
3.1.5 Compresor 38
3.2 Algoritmo CAC-SS 40
3.3 Arquitectura digital CAC-SS 41
3.3.1 Generador de la matriz de muestreo **Φ** 41
3.3.2 Combinador 45
3.3.3 Acumulador 49
3.4 Ventajas de CAC-SS sobre CAC-OST 49

Capítulo 4: Pruebas y resultados **52**
4.1 Arquitectura digital CAC-OST 52
4.2 Arquitectura digital CAC-SS 55
4.3 Resultados 62
4.3.1 Marco de referencia 62
4.3.2 Arquitecturas digitales CAC-OST y CAC-SS 64
4.3.3 Arquitecturas digitales CAC-OST y CAC-SS *vs* marco de referencia 68
4.4 Pruebas con distintas señales de entrada 71
4.4.1 Señal ECG real 71
4.4.2 Señal EMG real 83
4.4.3 Audio 85
4.4.4 Imagen 87
4.5 Análisis de resultados 89
Capítulo 5: Conclusiones **104**
5.1 Conclusiones 104
5.2 Trabajo a futuro 105
Productos derivados de la investigación **107**
Referencias **108**

Lista de tablas

1.1 Parámetros de M y $\mathbf{\Phi}$ para distintos CR. 7
1.2 Tasa de error para distintos CR. 8

2.1 Reporte de síntesis de la arquitectura digital [13] para CR_1. 24
2.2 Reporte de síntesis de la arquitectura digital [13] para CR_2. 27
2.3 Reporte de síntesis de la arquitectura digital [13] para CR_3. 30

3.1 Reporte de síntesis de la memoria. 34
3.2 Valores esperados y obtenidos de la memoria. 34
3.3 Reporte de síntesis del umbral. 36
3.4 Valores esperados y obtenidos del umbral. 36
3.5 Reporte de síntesis de la semilla inicial. 37
3.6 Valores esperados y obtenidos de la semilla inicial. 37
3.7 Reporte de síntesis del generador de la matriz de muestreo $\mathbf{\Phi}$. 39
3.8 Reporte de síntesis del compresor. 40
3.9 Reporte de síntesis del generador de la matriz de muestreo $\mathbf{\Phi}$ para CR_1. 44
3.10 Reporte de síntesis del generador de la matriz de muestreo $\mathbf{\Phi}$ para CR_2. 45
3.11 Reporte de síntesis del generador de la matriz de muestreo $\mathbf{\Phi}$ para CR_3. 47
3.12 Reporte de síntesis del combinador. 48
3.13 Valores esperados y obtenidos del combinador. 50
3.14 Reporte de síntesis del acumulador. 50

4.1 Reporte de síntesis de la arquitectura digital CAC-OST para CR_1. 53
4.2 Reporte de síntesis de la arquitectura digital CAC-OST para CR_2. 56
4.3 Reporte de síntesis de la arquitectura digital CAC-OST para CR_3. 58
4.4 Reporte de síntesis de la arquitectura digital CAC-SS para CR_1. 60
4.5 Reporte de síntesis de la arquitectura digital CAC-SS para CR_2. 62
4.6 Reporte de síntesis de la arquitectura digital CAC-SS para CR_3. 65
4.7 Comparación de tasa de error en muestreo compreso por *software*. 67
4.8 Comparación de área y velocidad de arquitectura digital [13]. 67
4.9 Comparación de tasa de error de arquitectura digital [13]. 68
4.10 Comparación de área y velocidad de arquitectura digital CAC-OST. 68
4.11 Comparación de tasa de error de arquitectura digital CAC-OST. 69
4.12 Comparación de área y velocidad de arquitectura digital CAC-SS. 70
4.13 Comparación de tasa de error de arquitectura digital CAC-SS. 71
4.14 Comparación de área y velocidad de las arquitecturas digitales CAC-SS, CAC-OST y [13]. 71
4.15 Comparación de tasa de error en la recuperación de la señal para CR_1. 75
4.16 Comparación de tasa de error en recuperación de la señal para CR_2. 75
4.17 Comparación de tasa de error en recuperación de la señal para CR_3. 75

4.18 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_1$ para CR_1.....91
4.19 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_1$ para CR_2. 91
4.20 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_1$ para CR_3. 91
4.21 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_2$ para CR_1. 91
4.22 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_2$ para CR_2. 92
4.23 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_2$ para CR_3. 92
4.24 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_3$ para CR_1. 96
4.25 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_3$ para CR_2. 96
4.26 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_3$ para CR_3. 96
4.27 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_4$ para CR_1. 96
4.28 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_4$ para CR_2. 97
4.29 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_4$ para CR_3. 97
4.30 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_5$ para CR_1. 100
4.31 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_5$ para CR_2. 100
4.32 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_5$ para CR_3. 100
4.33 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_6$ para CR_1. 101
4.34 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_6$ para CR_2. 101
4.35 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_6$ para CR_3. 101
4.36 Comparación de tasa de error, proporción máxima de señal a ruido y relación de señal a ruido en la recuperación de la señal $\mathbf{f}_7$ para CR_3. 103

Lista de figuras

1.1 (a) Señal analógica $f(t)$, (b) 960 muestras de $f(t)$, (c) 120 muestras de $f(t)$ y (d) 45 muestras de $f(t)$, efecto *sampling*. 4
1.2 (a) Señal **f** y (b) Proceso de muestreo compreso con $\mathbf{\Phi}$. 6
1.3 (a) Señal analógica (f) y (b) Magnitud del espectro de frecuencias $|F(\omega)|$. 6
1.4 (a) Señal original $f(t)$, (b) Muestreo compreso de la señal, (c) Recuperación de la señal y (d) Señal original *vs* señal recuperada. 6
1.5 (a) Señal ECG original [12], (b) Muestreo compreso de la señal con CR_1, (c) Recuperación de la señal y (d) Señal original *vs* señal recuperada. 7
1.6 (a) Señal ECG original [12], (b) Muestreo compreso de la señal con CR_2, (c) Recuperación de la señal y (d) Señal original *vs* señal recuperada. 7
1.7 Muestreo compreso en: (a) Dominio digital y (b) Dominio analógico. 8

2.1 Arquitectura digital [13]. 13
2.2 Generador de la matriz de muestreo $\mathbf{\Phi}$. 13
2.3 (a) Arquitectura para un canal [20] y (b) Generador de la matriz de muestreo $\mathbf{\Phi}$. 16
2.4 Matriz de muestreo $\mathbf{\Phi}$ generada por CAC-OST con $N = 32$ y $M = 16$. 18
2.5 *App* para muestreo compreso por *software*. 19
2.6 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada. 19
2.7 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada. 20
2.8 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada. 20
2.9 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada. 21
2.10 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada. 21
2.11 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada. 22
2.12 *App* para muestreo compreso por arquitectura digital [13]. 22
2.13 Diagrama RTL para CR_1 de: (a) Generador de la matriz de muestreo $\mathbf{\Phi}_2$, (b) *Mixer*, (c) *Integrator* y (d) Multiplexor. 23
2.14 Diagrama RTL de la arquitectura digital [13] para CR_1. 23
2.15 Señal ECG cuantificada. 24
2.16 Resultados de compresión con $\mathbf{\Phi}_2$ y CR_1. 24
2.17 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada. 25
2.18 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada. 25

2.19 Diagrama RTL para CR_2 de: (a) Generador de la matriz de muestreo $\mathbf{\Phi}_2$, (b) *Mixer*, (c) *Integrator* y (d) Multiplexor. 26
2.20 Diagrama RTL de la arquitectura digital [13] para CR_2. 26
2.21 Resultados de compresión con $\mathbf{\Phi}_2$ y CR_2 27
2.22 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada. 28
2.23 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada. 28
2.24 Diagrama RTL para CR_3 de: (a) Generador de la matriz de muestreo $\mathbf{\Phi}_2$, (b) *Mixer*, (c) *Integrator* y (d) Multiplexor. 29
2.25 Diagrama RTL de la arquitectura digital [13] para CR_3. 29
2.26 Resultados de compresión con $\mathbf{\Phi}_2$ y CR_3. 30
2.27 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal. Original *vs* señal recuperada. 30
2.28 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal. Original *vs* señal recuperada. 31

3.1 Arquitectura digital CAC.OST. 33
3.2 Memoria. 33
3.3 Diagrama RTL de la memoria. 34
3.4 Resultados de simulación de la memoria. 34
3.5 Diagrama RTL del umbral. 36
3.6 Resultados de simulación del umbral. 36
3.7 Diagrama RTL de la semilla inicial. 36
3.8 Resultados de simulación de la semilla inicial. 37
3.9 Generador CAC-OST de la matriz de muestreo $\mathbf{\Phi}$. 38
3.10 Diagrama RTL del generador de la matriz de muestreo $\mathbf{\Phi}$ 39
3.11 Resultados de simulación del generador de la matriz de muestreo $\mathbf{\Phi}$. 39
3.12 Diagrama RTL del compresor. 40
3.13 Resultados de simulación del compresor. 41
3.14 Matriz de muestreo $\mathbf{\Phi}$ generada por CAC-SS con $N = 32$ y $M = 16$. 41
3.15 Arquitectura digital CAC-SS. 42
3.16 Generador CAC-SS de la matriz de muestreo $\mathbf{\Phi}$. 43
3.17 Diagrama RTL del generador de la matriz de muestreo $\mathbf{\Phi}$ para CR_1. 43
3.18 Resultados de simulación del generador de la matriz de muestreo $\mathbf{\Phi}$ para CR_1 44
3.19 Diagrama RTL del generador de la matriz de muestreo $\mathbf{\Phi}$ para CR_2. 45
3.20 Resultados de simulación del generador de la matriz de muestreo $\mathbf{\Phi}$ para CR_2 46
3.21 Diagrama RTL del generador de la matriz de muestreo $\mathbf{\Phi}$ para CR_3. 46
3.22 Resultados de simulación del generador de la matriz de muestreo $\mathbf{\Phi}$ para CR_3 47
3.23 Diagrama RTL del combinador. 48
3.24 Resultados de simulación del combinador. 50
3.25 Diagrama RTL del acumulador. 50
3.26 Resultados de simulación del acumulador. 51

4.1 Diagrama RTL de la arquitectura digital CAC-OST para CR_1 53
4.2 Señal ECG cuantificada. 53

4.3 Resultados de compresión cuando $\boldsymbol{\Phi}_3$ es CAC-OST y CR_1. 54
4.4 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original *vs* señal recuperada. 54
4.5 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 55
4.6 Diagrama RTL de la arquitectura digital CAC-OST para CR_2 55
4.7 Resultados de compresión con $\boldsymbol{\Phi}_3$ y CR_2. 56
4.8 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_2, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original *vs* señal recuperada. 57
4.9 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_2, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 57
4.10 Diagrama RTL de la arquitectura digital CAC-OST para CR_3 57
4.11 Resultados de compresión con $\boldsymbol{\Phi}_3$ y CR_3. 58
4.12 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_3, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original *vs* señal recuperada. 58
4.13 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_3, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 59
4.14 Diagrama RTL de la arquitectura digital CAC-SS para CR_1. 59
4.15 Señal ECG cuantificada. 60
4.16 Resultados de compresión cuando $\boldsymbol{\Phi}_4$ es CAC-SS y CR_1. 60
4.17 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original *vs* señal recuperada. 61
4.18 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 61
4.19 Diagrama RTL de la arquitectura digital CAC-SS para CR_2 62
4.20 Resultados de compresión con $\boldsymbol{\Phi}_4$ y CR_2. 63
4.21 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_2, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original vs señal recuperada. 63
4.22 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_2, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 64
4.23 Diagrama RTL de la arquitectura digital CAC-SS para CR_3 64
4.24 Resultados de compresión con $\boldsymbol{\Phi}_4$ y CR_3. 65
4.25 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_3, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original vs señal recuperada. 65
4.26 (a) Señal ECG original [12], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_3, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 66
4.27 Señal ECG original [12] *vs* señal recuperada con CR_1 y: (a) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_2$. 66
4.28 Señal ECG original [12] *vs* señal recuperada con CR_2 y: (a) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_2$. 66
4.29 Señal ECG original [12] *vs* señal recuperada con CR_3 y: (a) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_2$. 67
4.30 Señal ECG original [12] *vs* señal recuperada con CR_1 y: (a) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_2$. 67
4.31 Señal ECG original [12] *vs* señal recuperada con CR_2 y: (a) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_2$. 68
4.32 Señal ECG original [12] *vs* señal recuperada con CR_3 y: (a) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_2$. 68
4.33 Señal ECG original [12] *vs* señal recuperada con CR_1 y: (a) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_2$. 69
4.34 Señal ECG original [12] *vs* señal recuperada con CR_2 y: (a) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_2$. 69
4.35 Señal ECG original [12] *vs* señal recuperada con CR_3 y: (a) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_2$. 69
4.36 Señal ECG original [12] *vs* señal recuperada con CR_1 y: (a) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_2$. 70

4.37 Señal ECG original [12] *vs* señal recuperada con CR_2 y: (a) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_2$. 70
4.38 Señal ECG original [12] *vs* señal recuperada con CR_3 y: (a) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_1$ y (b) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_2$. 70
4.39 Señal ECG original [12] *vs* señal recuperada con CR_1 y: (a) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_1$, (b) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_2$, (c) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_1$, (d) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_2$, (e) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_1$, (f) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_2$, (g) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_1$ y (h) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_2$. 72
4.40 Señal ECG original [12] *vs* señal recuperada con CR_2 y: (a) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_1$, (b) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_2$, (c) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_1$, (d) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_2$, (e) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_1$, (f) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_2$, (g) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_1$ y (h) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_2$. 73
4.41 Señal ECG original [12] *vs* señal recuperada con CR_3 y: (a) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_1$, (b) $\boldsymbol{\Phi}_1$ y $\boldsymbol{\Psi}_2$, (c) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_1$, (d) $\boldsymbol{\Phi}_2$ y $\boldsymbol{\Psi}_2$, (e) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_1$, (f) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_2$, (g) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_1$ y (h) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_2$. 74
4.42 Señal $\mathbf{f}_1$ cuantificada. 75
4.43 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original *vs* señal recuperada. 76
4.44 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 76
4.45 Señal $\mathbf{f}_2$ cuantificada. 76
4.46 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original vs señal recuperada. 77
4.47 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 77
4.48 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_2, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original vs señal recuperada. 78
4.49 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_2, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 78
4.50 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_2, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original vs señal recuperada. 79
4.51 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_2, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 79
4.52 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_3, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original vs señal recuperada. 80
4.53 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_3, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 80
4.54 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_3, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original vs señal recuperada. 81
4.55 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_3$ y CR_3, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 81
4.56 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original vs señal recuperada. 82
4.57 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 82
4.58 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original vs señal recuperada. 83
4.59 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_1, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 83
4.60 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_2, (c) Recuperación con $\boldsymbol{\Psi}_1$ y (d) Señal original vs señal recuperada. 84
4.61 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\boldsymbol{\Phi}_4$ y CR_2, (c) Recuperación con $\boldsymbol{\Psi}_2$ y (d) Señal original vs señal recuperada. 84

4.62 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original vs señal recuperada. 85
4.63 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original vs señal recuperada. 85
4.64 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original vs señal recuperada. 86
4.65 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original vs señal recuperada. 86
4.66 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original vs señal recuperada. 87
4.67 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original vs señal recuperada. 87
4.68 Señal ECG real $\mathbf{f}_1$ [24] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 88
4.69 Señal ECG real $\mathbf{f}_1$ [24] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 88
4.70 Señal ECG real $\mathbf{f}_1$ [24] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$ 89
4.71 Señal ECG real $\mathbf{f}_2$ [24] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 89
4.72 Señal ECG real $\mathbf{f}_2$ [24] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 90
4.73 Señal ECG real $\mathbf{f}_2$ [24] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$ 90
4.74 Señal **f**3 cuantificada. 92
4.75 Señal **f**4 cuantificada. 92
4.76 Señal EMG real $\mathbf{f}_3$ [24] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 93
4.77 Señal EMG real $\mathbf{f}_3$ [24] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 93
4.78 Señal EMG real $\mathbf{f}_3$ [24] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 94
4.79 Señal EMG real $\mathbf{f}_4$ [24] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 94
4.80 Señal EMG real $\mathbf{f}_4$ [24] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 95
4.81 Señal EMG real $\mathbf{f}_4$ [24] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$ 95
4.82 Señal de audio $\mathbf{f}_5$ [25] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 97
4.83 Señal de audio $\mathbf{f}_5$ [25] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 98
4.84 Señal de audio $\mathbf{f}_5$ [25] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 98
4.85 Señal de audio $\mathbf{f}_6$ [25] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$. 99

4.86 Señal de audio $\mathbf{f}_6$ [25] *vs* señal recuperada con CR_2 y: (a) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_1$, (b) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_2$, (c) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_1$ y (d) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_2$. 99
4.87 Señal de audio $\mathbf{f}_6$ [25] *vs* señal recuperada con CR_3 y: (a) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_1$, (b) $\boldsymbol{\Phi}_3$ y $\boldsymbol{\Psi}_2$, (c) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_1$ y (d) $\boldsymbol{\Phi}_4$ y $\boldsymbol{\Psi}_2$. 100
4.88 Lena.tiff 102
4.89 Señal de entrada $\mathbf{f}_7$ 102
4.90 Señal f_7 recuperada con CR_3 y: (a) Φ_3 y Ψ_1 y (b) Φ_4 y Ψ_1 103

1 Introducción

Las señales[1] encontradas en casos prácticos (audio, imagen, video, señales biomédicas, etc.), en su mayoría, se encuentran en el dominio analógico (señales analógicas) y el tratamiento de estas conviene realizarse en el dominio digital (señales digitales) para aprovechar los avances que los sistemas digitales han tenido en los últimos años. La conversión de señales analógicas a señales digitales se realiza mediante el siguiente proceso [1]:

1. La señal analógica se muestrea.
2. Las muestras de la señal analógica se cuantifican.

Este proceso conlleva una pérdida natural en la información que imposibilita la reconstrucción exacta de la señal original mediante las muestras cuantificadas de la misma. Es imposible recuperar la señal original (analógica) sin pérdida de información una vez que esta ha cambiado de dominio, de analógico a digital.

La pérdida de información al realizarse el cambio de dominio puede disminuirse si se cumplen algunos requisitos que fueron propuestos, primero por Harry Nyquist [2]-[3], después por Claude Shannon [4]. Las teorías propuestas por Harry Nyquist y Claude Shannon sentaron las bases para el Teorema de Shannon-Nyquist, también conocido como Teorema de Muestreo (*Sampling Theorem*).

El teorema de muestreo, del que se darán a conocer mayores detalles en la sección 1.1, permite digitalizar una señal analógica a partir de cierta información conocida de la señal, particularmente las frecuencias presentes en ella. Si bien este teorema ha resultado preciso para realizar la conversión de dominio, de analógico a digital, se han buscado opciones alternas para disminuir el número de muestras necesarias para cambio de dominio y compresión de las muestras ya digitalizadas. La finalidad en la reducción del número de muestras es almacenar o enviar menor cantidad de información y poder reconstruir la señal.

La reducción en el número de muestras y la compresión de estas puede realizarse si se cumplen algunos requisitos que fueron propuestos por David Donoho [5]. Los requisitos propuestos por David Donoho mencionaban una nueva teoría conocida como Muestreo Compreso (*Compressed Sensing*, CS), siendo la primera vez que se mencionaba dicha teoría.

El muestreo compreso, del que se darán a conocer mayores detalles en la sección 1.2, permite reconstruir una señal con menos muestras en comparación con el teorema de muestreo. En la última década se han desarrollado investigaciones referentes a muestreo compreso centradas en aplicaciones donde los recursos de *hardware* (almacenamiento, potencia, etc.) son casi ilimitados

[1]Una señal está definida por una magnitud física que varía en el tiempo, espacio o cualquier otra(s) variable(s) independiente(s) [1].

y las señales están previamente almacenadas.

La presente investigación permitirá implementar muestreo compreso en un dispositivo móvil y trabajar con señales tipo *streaming* (ininterrumpidas). El aporte científico de la investigación se centrará en la matriz de muestre, donde se pretende que sea una matriz estructurada.

Este capítulo presenta las bases del teorema de muestreo y muestreo compreso. El capítulo incluye ejemplos de teorema de muestreo y muestreo compreso, así como motivación, problemática a resolver, pregunta de investigación, objetivos, justificación y clasificación de la investigación.

1.1 Teorema de Muestreo

El teorema de muestreo surge a partir de los principios propuestos por Harry Nyquist y Claude Shannon, los cuales son:

- Harry Nyquist [2]–[3] determinó que el rango mínimo de frecuencias necesarias para transmitir en un canal de telégrafo, dada una velocidad de señalización, se encontraba limitado por el doble del ancho de banda del canal. La cantidad de pulsos que se pueden enviar por un canal de telégrafo tiene la limitante del ancho de banda del canal.
- Claude Shannon [4] demostró que la información es medible y que los canales de comunicación tienen una capacidad de transmisión que es determinada por la velocidad máxima de transferencia. La transmisión de información sobre un canal tiene la limitante de que esta no debe exceder la capacidad de transmisión del canal.

Los principios descritos por Harry Nyquist y Claude Shannon se pueden resumir en dos y son la base del teorema de muestreo [6]:

1. La señal de interés debe ser de banda limitada.
2. La frecuencia de muestreo (F_s), también conocida como frecuencia de Nyquist o tasa de Nyquist, debe ser mayor al doble de la frecuencia máxima (F_{max}) presente en la señal de interés. En caso de no respetar este principio, aparece un efecto llamado *aliasing* que son componentes de frecuencia no deseados, o bien, que no pertenecen a la señal [1].

$$F_s > 2F_{max}. \tag{1.1}$$

Ejemplo 1.1: Considerando la siguiente señal analógica $f(t)$:

$$f(t) = Asin(\omega_1 * t + \varphi) + Asin(\omega_2 * t + \varphi) + Asin(\omega_3 * t + \varphi) + Asin(\omega_4 * t + \varphi),$$

en donde,

$$A = 1,$$
$$t = 312.5\mu s,$$
$$\varphi = 0,$$

$$\omega_1 = 2 * \pi * F_1, \qquad F_1 = 10Hz,$$
$$\omega_2 = 2 * \pi * F_2, \qquad F_2 = 40Hz,$$
$$\omega_3 = 2 * \pi * F_3, \qquad F_3 = 70Hz,$$
$$\omega_4 = 2 * \pi * F_4, \qquad F_4 = 100Hz,$$

la frecuencia máxima presente en la señal $f(t)$ es:

$$F_{max} = 100Hz,$$

por tanto, aplicando (1.1), la frecuencia de muestreo es:

$$F_s > 2 \times 100Hz,$$
$$F_s > 200Hz.$$

La Figura 1.1 representa la señal analógica $f(t)$ y las muestras de $f(t)$ necesarias a distintas frecuencias de muestreo para reconstruir la señal.

El teorema de muestreo permite convertir una señal analógica en una señal digital. La señal digitalizada aprovecha la evolución en los sistemas digitales para ser procesada con la finalidad de extraer características que difícilmente se podrían observar si se procesara de forma analógica. La señal digitalizada y procesada se puede almacenar, enviar o convertir a su dominio natural, analógico.

1.2 Muestreo compreso

El muestreo compreso surge de la necesidad de poder reconstruir una señal a partir de menos muestras. Las primeras investigaciones para reducir el número de muestras y comprimirlas, se basaron en una característica propia de la señal de interés, la dispersión[2] (*sparsity*) [7]– [9]. La señal de interés puede expresarse como un vector **f** de longitud $N \times 1$ representado por:

$$\mathbf{f} = \mathbf{\Psi x}, \tag{1.2}$$

en donde, $\mathbf{\Psi}$ es la matriz de dominio en la cual la señal **f** es dispersa y **x** es la dispersión de la señal. Las dimensiones de $\mathbf{\Psi}$ y **x** son $N \times N$ y $N \times 1$, respectivamente.

Muestreo compreso permite muestrear y comprimir la señal de interés a través de una matriz de muestreo $\mathbf{\Phi}$. La dimensión de $\mathbf{\Phi}$ es de $M \times N$, en donde $M < N$ [10]. La señal **f** es multiplicada por cada fila de la matriz de muestreo $\mathbf{\Phi}$ para obtener el vector **y** de longitud $M \times 1$, es decir:

$$\mathbf{y} = \mathbf{\Phi f}. \tag{1.3}$$

La razón de compresión (*Compression Ratio*, CR) en muestreo compreso es:

$$CR = \frac{N}{M}. \tag{1.4}$$

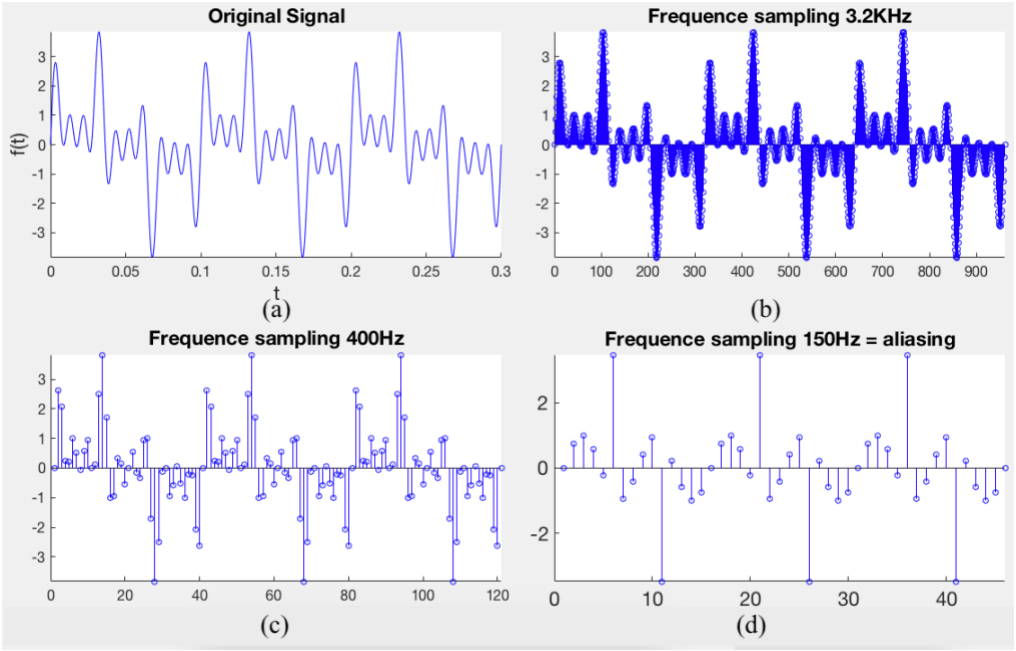


Figura 1.1 (a) Señal analógica $f(t)$, (b) 960 muestras de $f(t)$, (c) 120 muestras de $f(t)$ y (d) 45 muestras de $f(t)$, efecto *sampling*.

Muestreo compreso está basado en dos principios básicos:

1. Dispersión.

   La dispersión de la señal indica que la mayoría de las muestras de **x** son valores cercanos a cero, es decir, contienen información poco relevante que puede ser descartada sin afectar la reconstrucción de la señal. Sólo algunas muestras, K, contienen información relevante de la señal. Dichas muestras son conocidas como K-sparse, en donde $K \ll N$. La Figura 1.2 (a) representa (1.2) cuando la matriz de dominio $\boldsymbol{\Psi}$ es Discrete Cosine Transform (DCT) y vector **x** disperso con $K = 8$. Se ha demostrado que M está relacionado con N y K de la siguiente manera:

$$M \geq c(K) log_2 \left(\frac{N}{K}\right), \tag{1.5}$$

   en donde c es una constante pequeña [7]. La mayoría de las señales encontradas en casos prácticos tienen representación dispersa en algún dominio (Fourier, Wavelet, Gabor, Walsh-Hadamard, etc.) [11].

2. Incoherencia.

   La incoherencia significa que cualquier renglón de la matriz de muestreo $\boldsymbol{\Phi}$ debe tener la menor correlación con cualquier columna de la matriz de dominio $\boldsymbol{\Psi}$. La forma de medir la coherencia entre $\boldsymbol{\Phi}$ y $\boldsymbol{\Psi}$ es [11]:

$$\mu(\boldsymbol{\Phi}, \boldsymbol{\Psi}) = \sqrt{n} \cdot \max_{1 \leq i,j \leq n} |\langle \boldsymbol{\varphi}_i, \boldsymbol{\psi}_i \rangle|, \tag{1.6}$$

en donde, $\boldsymbol{\varphi}_i$ es la i-ésima fila de $\boldsymbol{\Phi}$, $\boldsymbol{\psi}_j$ es la j-ésima columna de $\boldsymbol{\Psi}$ y $\langle \boldsymbol{\varphi}_i, \boldsymbol{\psi}_j \rangle$ significa el producto punto entre $\boldsymbol{\varphi}_i$ y $\boldsymbol{\psi}_j$. A menor correlación entre $\boldsymbol{\Phi}$ y $\boldsymbol{\Psi}$, una mejor reconstrucción de la señal. La mayoría de las matrices aleatorias tienen una baja correlación con la mayoría de las matrices de dominio, por ello, las matrices Gaussianas y Bernoulli son las más utilizadas en muestreo compreso. La Figura 1.2 (b) muestra el proceso de muestreo compreso cuando la matriz de muestreo $\boldsymbol{\Phi}$ es Bernoulli.

Ejemplo 1.2: Considerando la señal analógica $f(t)$ del Ejemplo 1.1, se observa que si representamos la señal en el dominio de Fourier encontramos la representación dispersa adecuada. La Figura 1.3 muestra la señal analógica $f(t)$ y la magnitud del espectro de frecuencias $|F(\omega)|$.

Ejemplo 1.3: Codificando la señal analógica $f(t)$ del Ejemplo 1.1 mediante muestreo compreso con las siguientes características:

$$N = 960,$$
$\boldsymbol{\Psi}$ es DCT,
$$M = 480,$$
$\boldsymbol{\Phi}$ es Gaussian random,

por tanto, aplicando (1.4) el CR es:

$$CR = \frac{960}{480} = 2.$$

La Figura 1.4 representa el proceso completo de muestreo compreso (codificación y recuperación de la señal), en (a) se grafica la señal analógica $f(t)$, en (b) se muestra el vector **y**, en (c) se grafica la recuperación de la señal y en (d) se comparan señal original y señal recuperada.

Ejemplo 1.4: Generando una señal ECG (*Electrocardiographic*) utilizando el simulador de Khartik [12] con las siguientes características:

$$Heart\ Beat\ (HB) = 72,$$
$$F_s = 100Hz,$$
$$N = 512,$$
$\boldsymbol{\Psi}$ es DCT,

se codifica la señal ECG, mediante muestreo compreso, con distintos CR indicados en la Tabla 1.1. Aplicando (1.4) para calcular los distintos CR:

$$CR_1 = \frac{512}{128} = 4,$$
$$CR_2 = \frac{512}{256} = 2.$$

La Figura 1.5 y Figura 1.6 representan el proceso completo de muestreo compreso (codificación y recuperación de la señal) para CR_1 y CR_2 respectivamente. La Tabla 1.2 muestra el error cuadrático medio (*Mean Squared Error*, MSE) que resulta de la comparación entre la señal

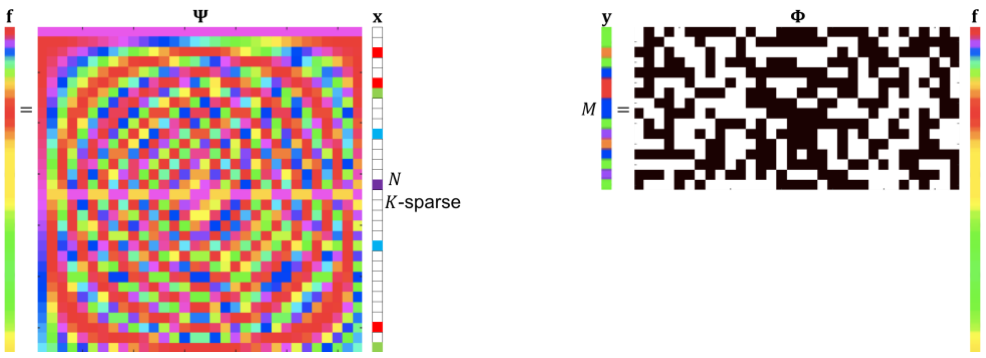


Figura 1.2 (a) Señal **f** y (b) Proceso de muestreo compreso con **Φ**.

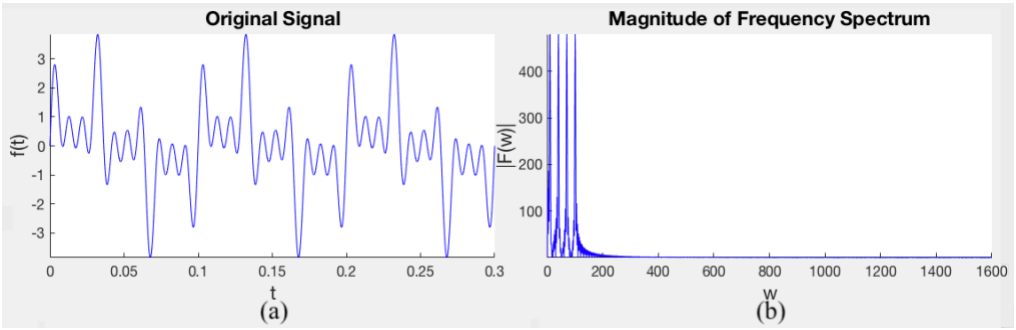


Figura 1.3 (a) Señal analógica (f) y (b) Magnitud del espectro de frecuencias $|F(\omega)|$.

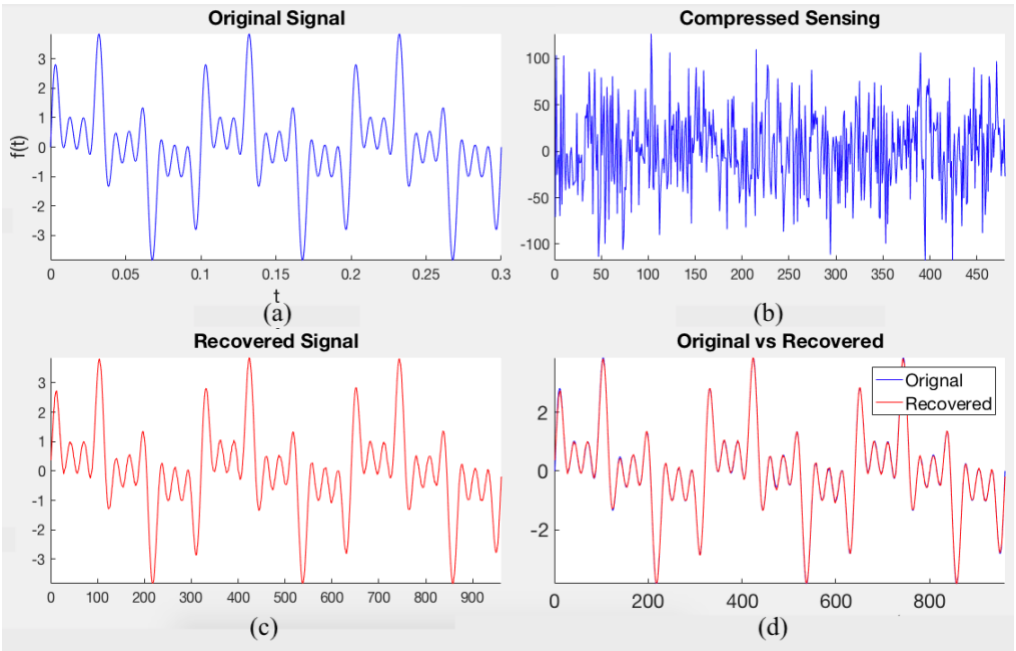


Figura 1.4 (a) Señal original $f(t)$, (b) Muestreo compreso de la señal, (c) Recuperación de la señal y (d) Señal original *vs* señal recuperada.

original y la señal recuperada. La Figura 1.7 indica que muestreo compreso puede ser aplicado en los dominios analógico y digital, en (a) la señal analógica $f(t)$ se amplifica, digitaliza y comprime, y en (b) la señal analógica $f(t)$ se amplifica, comprime y digitaliza. En términos de consumo de

Tabla 1.1 Parámetros de M y $\boldsymbol{\Phi}$ para distintos CR.

CR	M	$\boldsymbol{\Phi}$
CR_1	128	Gaussian random
CR_2	256	

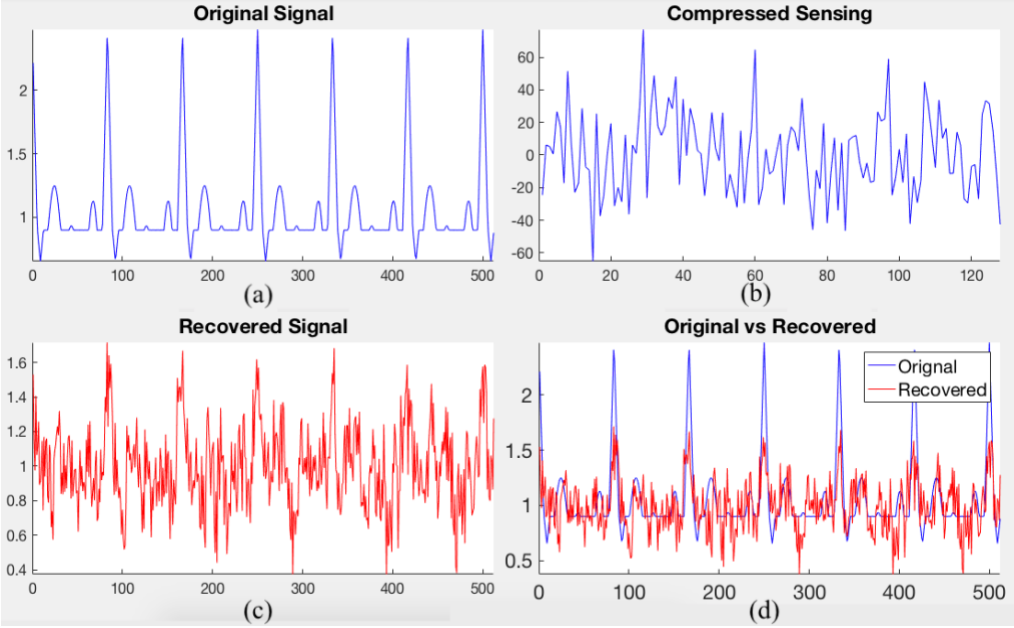


Figura 1.5 (a) Señal ECG original [12], (b) Muestreo compreso de la señal con CR_1, (c) Recuperación de la señal y (d) Señal original *vs* señal recuperada.

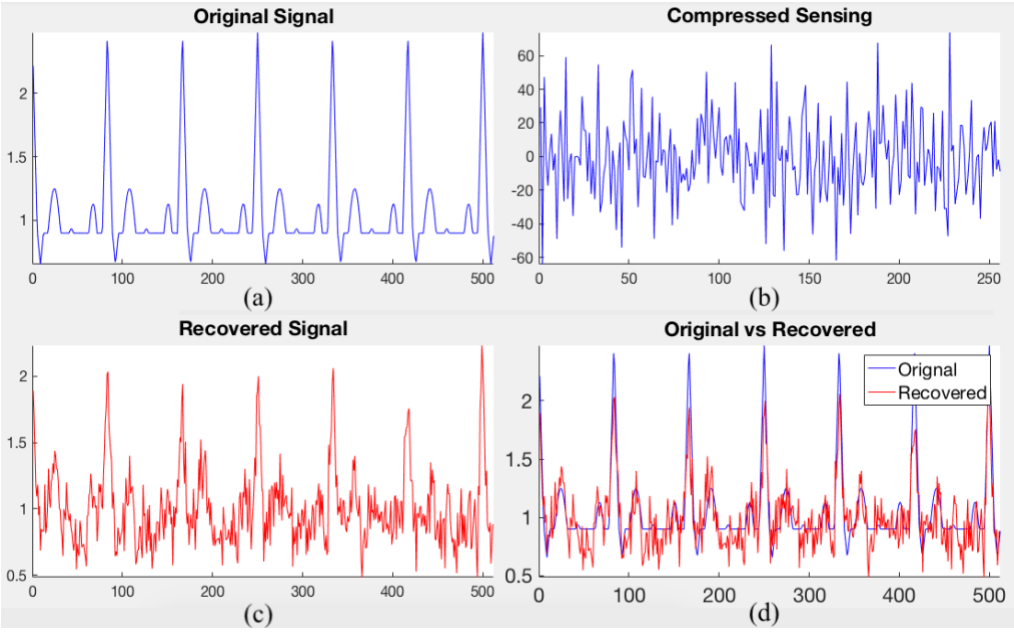


Figura 1.6 (a) Señal ECG original [12], (b) Muestreo compreso de la señal con CR_2, (c) Recuperación de la señal y (d) Señal original *vs* señal recuperada.

potencia, cuando la señal es de baja frecuencia, menos de 1GHz, el ahorro es mayor cuando muestreo compreso se aplica en el dominio digital [13].

Tabla 1.2 Tasa de error para distintos CR.

N	**Ψ**	M	**Φ**	CR	MSE
512	DCT	128	Gaussian random	4	0.0729
		256		2	0.0122

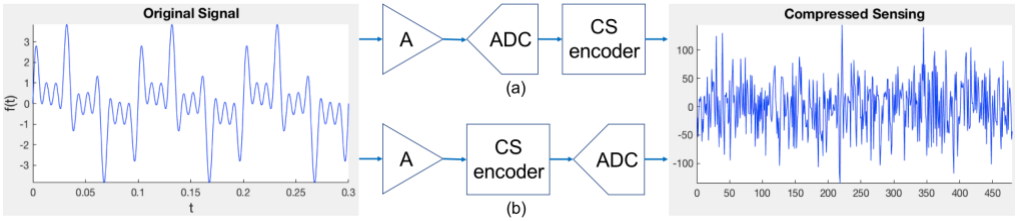


Figura 1.7 Muestreo compreso en: (a) Dominio digital y (b) Dominio analógico.

1.3 Motivación

El aumento en el uso de dispositivos móviles (*smartphones*, *tablets*, sensores biomédicos, etc.) demanda que cada vez más procesos puedan realizarse directamente en dichos dispositivos. La demanda en el uso de los dispositivos móviles enfrente las siguientes limitantes:

- Batería.
- Almacenamiento.
- Recursos de *hardware*.

Muestreo compreso reduce el número de muestras necesarias para reconstruir la señal. Los procesos requeridos en un dispositivo móvil podrían realizarse con muestreo compreso sin que los recursos de batería, almacenamiento o *hardware* sean una limitante.

La investigación permitirá desarrollar una nueva forma de muestrear y comprimir la información. El desarrollo de esa nueva forma de muestrear y comprimir la información permitirá procesar menos información, lo que conlleva a un menor consumo de batería, menor espacio de almacenamiento y menor cantidad de recursos de *hardware*.

1.4 Problemática a resolver

Las señales muestreadas en función de la frecuencia consumen potencia que puede ser reducida si se muestreara en función de la información contenida en la señal. Una cantidad de las muestras obtenidas con el teorema de muestreo contiene información poco relevante que puede ser descartada. El descartar muestras con información poco relevante lograría un ahorro en potencia y almacenamiento al enviar y almacenar menor cantidad de información.

Las matrices de muestreo consumirían una gran cantidad de recursos de *hardware* si se deseara implementar en un dispositivo móvil para señales que fluyen de forma ininterrumpida (*streaming*).

1.5 Pregunta de investigación

¿Cómo diseñar una arquitectura de muestreo compreso que pueda procesar una señal tipo *streaming* con una implementación eficiente y que permita reducir el número de muestras para reconstruir la señal?

La implementación eficiente en un sistema móvil tiene los siguientes requisitos:

- Bajo consumo de potencia: Los sistemas móviles están limitados en batería, un menor consumo de potencia implica una duración mayor de batería y la posibilidad de realizar mayor cantidad de operaciones por ciclo de carga.
- Recursos de *hardware*: Los sistemas móviles también están limitados en recursos de *hardware* debido al tamaño físico de estos. Una implementación eficiente utiliza la menor cantidad de recursos de *hardware* posible, lo cual impacta en el consumo de potencia. La relación entre cantidad de recursos de *hardware* utilizados y potencia consumida es directa, a menor cantidad de recursos de *hardware* utilizados, menor cantidad de potencia consumida.
- Velocidad: Un factor esencial en un sistema móvil es la velocidad. Una implementación eficiente realizar una gran cantidad de operaciones en el menor tiempo posible.

1.6 Objetivos

El objetivo general es:

- Diseñar una arquitectura de bajo consumo de potencia, baja utilización de recursos de *hardware* y alta velocidad de operación para muestreo compreso con base en matrices estructuradas.

Los objetivos específicos son:

- Construir una arquitectura regular: El diseño digital de una arquitectura requiere que su construcción sea tan simple como sea posible. Una arquitectura regular utiliza la menor cantidad posible de recursos de *hardware* para realizar las operaciones necesarias. La arquitectura debe ser síncrona, para ello un solo reloj es utilizado en toda la arquitectura. El reloj que utiliza la arquitectura se encuentra delimitado por la frecuencia de muestreo de la señal.
- Realizar una implementación computacionalmente eficiente: La arquitectura diseñada se implementa en un chip FPGA y debe cumplir con la eficiencia computacional. La eficiencia computacional implica que la complejidad en el diseño de la arquitectura sea la menor. Existen técnicas como *multirate* [14] en donde la frecuencia de muestre se modifica para incremental la eficiencia de varias operaciones en el procesamiento digital de señales.
- Simplificar la unidad de control (*Control Unit*, CU): La CU controla todos los procesos que realiza la arquitectura. El diseño de la arquitectura debe ser eficiente para que en todo momento la CU tenga conocimiento de en qué paso se encuentra y cuál es el paso siguiente.

El tener una CU simplificada reducirá el consumo de potencia e incrementará la velocidad de operación.

1.7 Justificación

El teorema de muestreo opera en función de la frecuencia máxima presente en la señal, mientras que muestreo compreso trabaja en función de la información de la señal permitiendo muestrear y comprimir simultáneamente. El cambio en la manera de muestrear la señal permite la compresión e impacta en la reducción del número de muestras necesarias para reconstruir la señal.

Los dispositivos móviles cuentan con determinadas características de recursos de *hardware*, batería y almacenamiento. Los dispositivos móviles al estar basados en el teorema de muestreo para digitalizar y procesar la señal tienen la capacidad de realizar un determinado número de operaciones. El número de operaciones que un dispositivo móvil puede realizar se vería incrementado si se basara en muestreo compreso, algunos ejemplos:

- Un sensor biomédico registra las señales ECG, *Holter*, tiene la capacidad de almacenar la actividad de una persona por un periodo entre 24 y 48 horas. El periodo de almacenamiento con muestreo compreso podría aumentar entre 96 y 192 horas si se considera una razón de compresión de cuatro.
- Un *smartphone* con capacidad de almacenamiento de 64Gb y una cámara de 8MP tiene la capacidad de almacenar 2,800 fotografías (aproximadamente). La cantidad de fotografías almacenadas con muestreo compreso podría aumentar a 5,600 (aproximadamente) si se considera una razón de compresión de la imagen de dos.

Muestreo compreso utiliza una matriz de muestreo $\mathbf{\Phi}$ para muestrear y comprimir la señal. La matriz de muestreo $\mathbf{\Phi}$ requiere ser densa para poder cumplir con dicho objetivo. Implementar una matriz densa para permitir recibir señales de tipo *streaming* requiere:

- Utilizar grandes recursos de *hardware*.
- Alto consumo de potencia.
- Alta capacidad de almacenamiento.

debido a los requisitos mencionados sería irrealizable implementar ese tipo de matriz en un sistema móvil.

La matriz de muestreo $\mathbf{\Phi}$ que utiliza muestreo compreso puede ser sustituida por una matriz estructurada para permitir recibir señales tipo *streaming* y que pueda ser implementado en un sistema móvil sin comprometer sus recursos (*hardware*, batería, almacenamiento, etc.)

1.8 Clasificación de la investigación

La presente investigación se encuentra clasificada en:

- *Institute of Electrical and Electronics Engineers* (IEEE).
  - *Design and Implementation of Signal Processing Systems* (DIS).
    - *System-on-chip Architectures for Signal Processing* (SOCP).
    - *Algorithm and Architecture Co-optimization* (AAOP).

- *Association for Computing Machinery* (ACM).
  - *Hardware.*
    - *Signal Processing Systems.*
    - *Digital Signal Processing.*

- Sección de Estudios de Posgrado – Escuela Superior de Cómputo (SEPI – ESCOM).
  - Sistemas Digitales para el Cómputo Móvil.

2 Arquitecturas de muestreo compreso: estado del arte

Este capítulo presenta las arquitecturas, consideraciones y métodos de muestreo compreso propuestos en la literatura. Inicia con la explicación del estado del arte. Las secciones 2.1 y 2.4 presentan arquitecturas digitales, se explica el funcionamiento de cada una y se indican las ventajas y desventajas que ofrecen. La sección 2.2 presenta consideraciones para señales ECG y EMG (*Electromiography*). Las secciones 2.3 y 2.5 presentan nuevos métodos de compresión y se indican las ventajas y desventajas que cada uno ofrece. El capítulo finaliza con un marco de referencia en donde la arquitectura de la sección 2.1 es reproducida en un FPGA y se reportan los resultados.

2.1 Diseño y análisis de arquitectura de muestreo compreso para compresión de datos en sensores inalámbricos

Chen *et al.* [13] introducen algoritmos de muestreo compreso para compresión de datos en sensores inalámbricos con la finalidad de abordar las restricciones de energía y telemetría comunes en dichos sensores. Las implementaciones de muestreo compreso que proponen son en ambos dominios, digital y analógico, para comparar los costos de potencia y rendimiento. Los resultados de sus análisis muestran que una implementación en el dominio digital es más eficiente a nivel energético para sensores inalámbricos en donde las señales a medir son biomédicas. La Figura 2.1 muestra la arquitectura digital [13] y consta de tres bloques:

1. Generador de la matriz de muestreo $\mathbf{\Phi}$: La Figura 2.2 muestra la generación de la matriz de muestreo $\mathbf{\Phi}$ en donde las entradas de la matriz aleatoria son generadas por dos generadores *Pseudo-Random Binary Sequence* (PRBS). La salida de cada registro del primer PRBS es conectado a una XOR con la salida del segundo PRBS. La matriz generada es una matriz aleatoria Bernoulli con coeficientes 0 y 1.
2. *Mixer*: Cada muestra de la señal se conecta a M XOR paralelos en cascada con M acumuladores paralelos. Cada XOR tiene una entrada correspondiente con el coeficiente del generador de la matriz de muestreo $\mathbf{\Phi}$, que a su vez es conectado al acarreo de entrada del correspondiente acumulador. Con ello se logra la función de multiplicación entre la señal y la matriz de muestreo $\mathbf{\Phi}$, dando valores ± 1 a las entradas de $\mathbf{\Phi}$ debido a que cuando la salida del XOR es alta el coeficiente es -1 y cuando la salida del XOR es baja el coeficiente es $+1$. Si el coeficiente de la matriz de muestreo $\mathbf{\Phi}$ es 1, el valor de la muestra de la señal se convierte en negativo, de lo contrario mantiene su valor.
3. *Integrator*: Los M acumuladores paralelos permiten sumar o restar los valores de las muestras hasta N muestras acumuladas para mostrar el vector $\mathbf{y}$ en la salida.

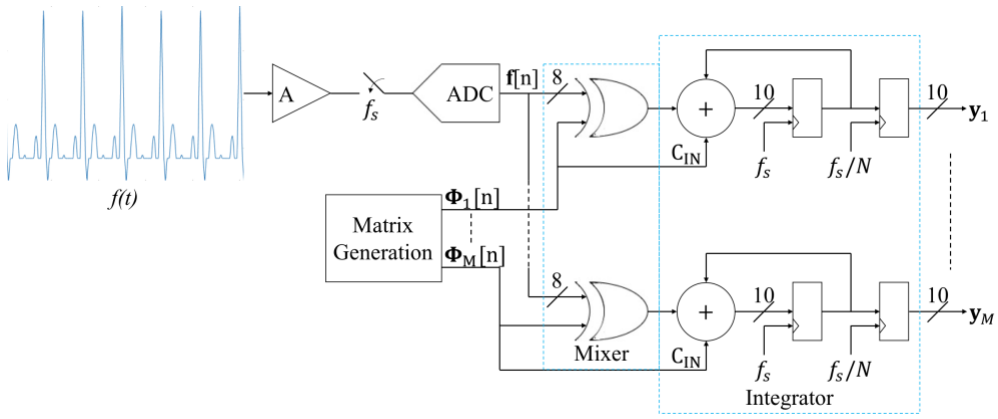


Figura 2.1 Arquitectura digital [13].

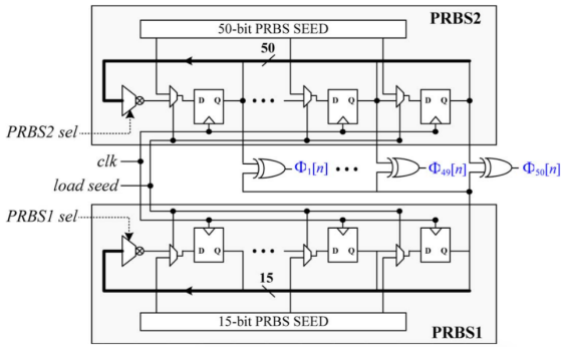


Figura 2.2 Generador de la matriz de muestreo **Φ**.

Las ventajas de la arquitectura digital [13] son:

- Proponen una razón de compresión de 10 para señales EEG (*Electroencephalography*).
- Los autores implementaron la arquitectura digital en un chip de aplicación específica, usando un proceso CMOS de 90nm y reportaron un consumo de $1.9\mu W$ a $0.6V$.

Las desventajas de la arquitectura digital [13] son:

- Realiza el muestreo compreso por bloques y pierde una muestra de la señal cada N muestras.
- La razón de compresión recomendable para señales EEG es de 4 y los autores consideran 10.

2.2 Consideraciones para muestreo compreso en sensores inalámbricos de señales ECG y EMG

Dixon *et al.* [15] comparan muestreo compreso con el convencional muestreo adaptativo para la adquisición de señales ECG y EMG. Los autores presentan consideraciones de diseño para sistemas de adquisición de dichas señales y realizan una comparación entre muestreo compreso y muestreo adaptativo para adquisición de señales ECG y EMG [15]:

- Los sistemas de monitoreo convencionales para señales ECG utilizan una frecuencia de muestreo de 256Hz. Se captura la información de alta frecuencia, pero se realiza un sobre-muestreo innecesario durante los periodos de baja actividad.
- Muestreo adaptativo ajusta la frecuencia de muestreo en función del nivel de actividad [16]-[17]. Requiere de un algoritmo y un circuito para detectar la región de las altas frecuencias de la señal y ajustar la frecuencia de muestreo adecuadamente [17].
- La tasa de digitalización en muestreo adaptativo se ajusta en función del nivel de actividad de la señal, en muestreo compreso es fija.
- Muestreo adaptativo requiere conocer la señal para establecer sus límites, en muestreo compreso no se requiere.
- Muestreo compreso puede ser aplicado a señales ECG, EMG, EEG, etc., sin la necesidad de realizarle ajustes en función de la señal.
- Muestreo adaptativo requiere de un circuito de detección de actividad [17] y un microcontrolador. Muestreo compreso sólo requiere de un circuito para compensar el umbral de dispersión (nivel de detalle) y compresión (ahorro energético).

Las consideraciones propuestas son [15]:

- Control en la dispersión de la señal: La razón de compresión en muestreo compreso puede ser ajustado en función del nivel de detalle requerido a través de un umbral dinámico. Se procesan sólo las muestras que superan dicho umbral.
- Matriz de muestreo **Φ**: La matriz con distribución Bernoulli de 1 bit reduce la complejidad del *hardware* y presenta un ahorro en el consumo de potencia.

Las simulaciones realizadas por los autores muestran una razón de compresión de 16 para señales ECG y EMG.

2.3 Muestreo compreso para monitoreo remoto de señales EEG

Zhang *et al.* [18] proponen un nuevo método para compresión y recuperación de señales EEG. El método propuesto es *Block Sparse Bayesian Learning* (BSBL) y funciona de la siguiente manera:

- Asume que la señal EEG no tiene representación dispersa en tiempo, ni en algún otro dominio. Al no presentar dispersión, no cumple con uno de los requerimientos de muestreo compreso.
- La señal es dividida en bloques concatenados. Los bloques concatenados no se superponen. Algunos bloques presentarán dispersión y a través del algoritmo *bound-optimization* [19] utilizan BSBL (BSBL-BO) para compresión y recuperación de la señal.

Las ventajas del método son [18]:

- Los resultados experimentales de los autores indica que la calidad en la recuperación mediante BSL es mayor en comparación con los algoritmos de recuperación propuestos

para muestreo compreso.
- BSBL puede ser utilizado para otro tipo de señales biomédicas que no cuentan con representación dispersa.

Las desventajas del método son [18]:

- BSBL realiza una doble recuperación, primero recupera los bloques en los que fue dividida la señal y después recupera la señal original.
- Las operaciones en la recuperación de la señal se incrementan al doble (aproximadamente).
- El costo en el consumo de potencia y recursos de *hardware* para una doble recuperación convierten al método [18] es irrealizable para un sistema móvil.

La compresión para señales EEG es compleja por la estructura de la señal. No obstante, muestreo compreso presenta interesantes ventajas sobre otros métodos de compresión como Wavelet. La ventaja de muestreo compreso sobre Wavelet se da en el consumo energético, siendo muestreo compreso el que consume menor cantidad de energía.

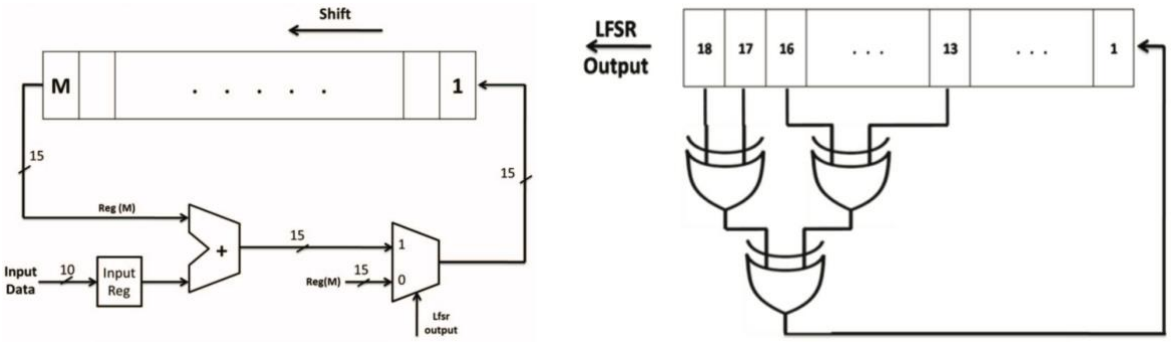
2.4 Implementación FPGA de muestreo compreso para compresión de señales ECG

Attarmoghaddam *et al.* [20] presentan un diseño de *hardware* de baja potencia para la compresión de señales ECG y proponen implementaciones para un canal y milticanal. La arquitectura multicanal que proponen utiliza la técnica de *power gating* y consiguen un ahorro en potencia. La técnica de *power gating* implica que los bloques de la arquitectura que nos utilizados se apagan y sólo son encendidos cuando son utilizados. Los autores realizaron las pruebas de sus arquitecturas implementándolos en FPGA de baja potencia de la familia IGLOO. La Figura 2.3 (a) muestra la arquitectura para un canal [20], la cual consta de tres módulos:

1. Generador de la matriz de muestreo **Φ**: La Figura 2.3 (b) muestra la generación de la matriz de muestreo **Φ**, esta se realiza a través de un LFSR para generar los coeficientes aleatorios utilizando 18 registros y 4 *taps*.
2. *Shift register*: Almacena temporalmente los datos.
3. Multiplexor: Implementa la función de multiplicación entra la muestra de la señal y la matriz de muestreo **Φ**.

El funcionamiento de la arquitectura digital para un canal [20] es el siguiente:

- La primera muestra de la señal es multiplicada por la primera columna de la matriz de muestreo **Φ** y se almacena en el *shift register*. La segunda muestra de la señal es multiplicada por la segunda columna de la matriz de muestreo **Φ**, se suma con el resultado del producto de la muestra anterior y se almacena en el *shift register*. El proceso continúa hasta la última columna de la matriz de muestreo **Φ**.
- Después de *M* ciclos de reloj, los datos se almacenan en su correspondiente registro del



(a) (b)

Figura 2.3 (a) Arquitectura para un canal [20] y (b) Generador de la matriz de muestreo **Φ**.

shift register.

- Los datos de salida estarán disponibles en el *shift register* después de *N* muestras.

La ventaja de esta arquitectura es que es escalable y puede ser implementada para multicanal. La implementación para multicanal agrega otro arreglo de *shift registers* por cada canal extra y un multiplexor para elegir en cuál canal se desea escribir. La investigación reporta una reducción en consumo de potencia de 20% y 40% a diferentes frecuencias de operación utilizando *power gating*. El consumo de potencia para un sistema de 4 y 8 canales se incrementa solamente en 7.1% y 11.2% respectivamente, mientras que el factor de incremento en área es de 2.3 y 5.4 respectivamente.

La desventaja de las arquitecturas propuestas es que utilizan la técnica de *power gating*, la cual no es recomendable en diseño de sistemas síncronos debido al uso de una compuerta AND en la señal de reloj para habilitar o deshabilitar bloques o registros. Los problemas de dicha técnica son:

- El ancho del ciclo del reloj puede ser estrecho.
- Puede permitir *glitches* y habilitar bloques o registros en momentos no deseados.
- Imposible diseñar la red de distribución del reloj.

2.5 Generación de matriz de muestreo con autómata celular caótico

Liu *et al.* [21] proponen el algoritmo *Cellular Automata Chaos with Original Signal Thresholding* (CAC-OST) para generar los coeficientes de la matriz de muestreo **Φ** basado en el generador de secuencias aleatorias *Cellular Automata* (CA) [22]. El algoritmo funciona de la siguiente manera:

- El valor absoluto de las *N* muestras se promedia para establecer el umbral (*threshold*).
- Cada muestra se compara contra el *threshold* para generar la semilla inicial, la cuál se convierte en la primera fila de la matriz de muestreo **Φ**. Los coeficientes de la primera fila son $\boldsymbol{\varphi}_{11}, \boldsymbol{\varphi}_{12}, \ldots, \boldsymbol{\varphi}_{1(N-1)}, \boldsymbol{\varphi}_{1N}$. Si el valor absoluto de la muestra *N* es mayor que el *threshold*, el coeficiente $\boldsymbol{\varphi}_{1N}$ se le asigna un 1, de lo contrario se le asigna un 0.
- Los otros coeficientes de la matriz de muestreo **Φ** son generadas utilizando la regla

propuesta por Stephen Wolfram [22]:

$$\boldsymbol{\varphi}_{(i+1)(j)} = \boldsymbol{\varphi}_{(i)(j-1)} \oplus (\boldsymbol{\varphi}_{(i)(j)} \vee \boldsymbol{\varphi}_{(i)(j+1)}), \tag{2.1}$$

en dónde, $\boldsymbol{\varphi}_{(i)(j)}$ representa el coeficiente de la i-ésima fila y j-ésima columna, mientras que $\oplus$ y $\vee$ son, respectivamente, operaciones XOR y OR. La Figura 2.4 muestra la matriz de muestreo $\boldsymbol{\Phi}$ generada por CAC-OST.

La ventaja del algoritmo [21] es que puede reconstruir una señal con menos muestras que utilizando matrices de muestreo $\boldsymbol{\Phi}$ con distribución Gaussiana o Bernoulli. Las pruebas que realizan los autores indican que pueden reconstruir la señal con $M = K$.

La desventaja del algoritmo [21] es que no puede ser implementado en señales tipo *streaming* debido a que requiere que la señal sea finita para establecer OST. El OST se establece con base en el valor de todas las muestras de la señal. Un OST parcial o segmentado imposibilita reconstruir con $M = K$.

2.6 Marco de referencia

La elaboración del marco de referencia se realiza para poder comparar arquitecturas para señales ECG propuestas en literatura contra las arquitecturas que se proponen en esta investigación. Las arquitecturas deben ser digitales y síncronas. Inicialmente, se decidió elegir las arquitecturas 2.1 y 2.4 para ser reproducidas en un FPGA, sin embargo, la arquitectura 2.4 fue descartada debido a que se trata de un caso específico de la arquitectura 2.1 y que no es síncrona. Finalmente, el marco de referencia quedó integrado por:

1. Muestreo compreso por *software*.
2. Arquitectura digital [13].

Cada arquitectura implementada en FPGA reporta la cantidad de elementos lógicos, registros, bits de memoria y máxima frecuencia de operación, que servirán de parámetros para poder comparar el rendimiento de las arquitecturas en la compresión. Adicionalmente, la señal compresa es recuperada a través de la optimización l_1 utilizando el código l_1-magic en Matlab® [23] y comparada con la señal original para obtener la tasa de error MSE. Cada método de muestreo compreso (marco de referencia y métodos propuestos) cuenta con una *app* desarrollada en Matlab® que permite mostrar la compresión y recuperación de la señal. Las herramientas utilizadas para la implementación de las arquitecturas digitales y la recuperación de la señal son:

- Kit de desarrollo DE2-115 (incluye un chip FPGA Cyclone IV EP4CE115F29C7).
- Quartus Prime 17.0 de Altera® como herramienta de síntesis y VHDL como lenguaje de descripción de *hardware*.
- ModelSim de Mentor Graphics® como herramienta de simulación.

Figura 2.4 Matriz de muestreo **Φ** generada por CAC-OST con $N = 32$ y $M = 16$.

Los valores considerados son:

f es una señal ECG simulada [12],
$N = 512$,
$M_1 = 64$, $CR_1 = 8$,
$M_2 = 128$, $CR_2 = 4$,
$M_3 = 256$, $CR_3 = 2$,
$\mathbf{\Phi}_1$ es Gaussian,
$\mathbf{\Phi}_2$ es Bernoulli,
$\mathbf{\Psi}_1$ es DCT,
$\mathbf{\Psi}_2$ es Walsh-Hadamard Transfrom (WHT),
Recuperación vía optimización l_1 [23],
Tasa de error vía MSE.

2.6.1 Muestreo compreso por *software*

El muestreo compreso por *software* se realizó con Matlab®. La Figura 2.5 muestra la *app* desarrollada para este método. El funcionamiento de la *app* se explica a continuación:

- *Load ECG Signal*: Grafica en plot 1 la señal ECG simulada [12].
- *Heart Beat*: Despliega la frecuencia cardiaca de la señal ECG.
- *Freq Samp*: Despliega la frecuencia de muestreo de la señal ECG.
- *Reload ECG Signal*: Permite modificar los parámetros de *Heart Beat* o *Freq Samp* y grafica en plot 1 la señal ECG con los nuevos parámetros.
- *Compression Ratio*: Permite elegir la razón de compresión.
- *Compressed Sensing*: Genera la matriz de muestreo **Φ**, multiplica la matriz por la señal ECG y grafica en plot 2 el vector **y**.
- *Recovered Signal*: Aplica optimización l_1 [23] para encontrar el vector disperso, lo multiplica por la matriz de dominio **Ψ** para recuperar la señal y lo grafica en plot 3.
- *Original vs Recovered*: Grafica en plot 4 la comparación entre la señal original *vs* la señal recuperada.
- *MSE*: Muestra el MSE de la comparación entre la señal original *vs* la señal recuperada.
- *Reset*: Limpia los plots e inicializa todas las variables en cero.

Los seis escenarios en los que se realizó este método son:

1. Comprime la señal en ocho y asume que la dispersión de la señal ECG es DCT. La Figura 2.6 muestra los resultados obtenidos y se observa el $MSE = 0.19539000$.
2. Comprime la señal en ocho y asume que la dispersión de la señal ECG es WHT. La

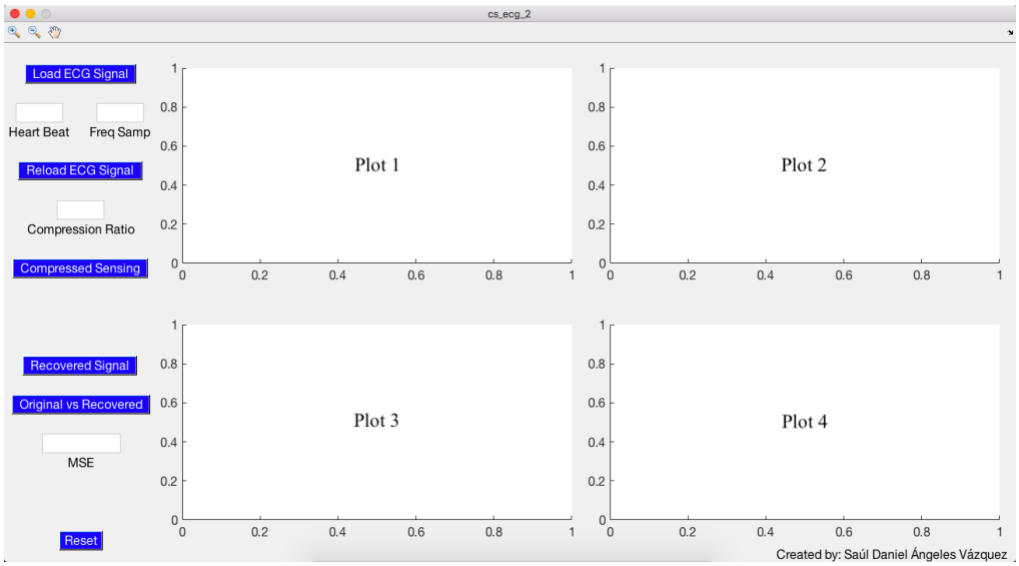


Figura 2.5 *App* para muestreo compreso por *software*.

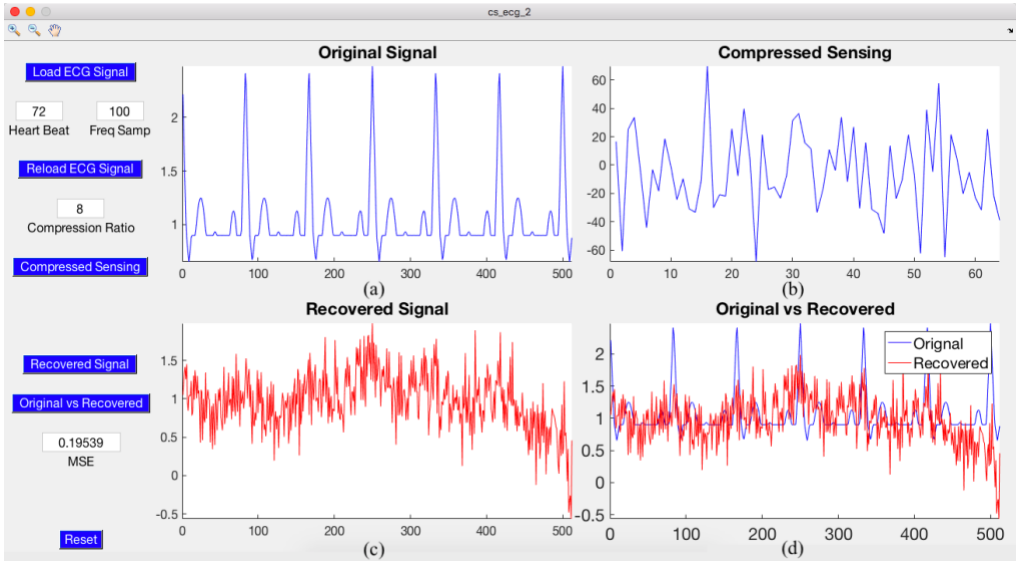


Figura 2.6 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

Figura 2.7 muestra los resultados obtenidos y se observa el $MSE = 0.12744800$.

3. Comprime la señal en cuatro y asume que la dispersión de la señal ECG es DCT. La Figura 2.8 muestra los resultados obtenidos y se observa el $MSE = 0.07467780$.
4. Comprime la señal en cuatro y asume que la dispersión de la señal ECG es WHT. La Figura 2.9 muestra los resultados obtenidos y se observa el $MSE = 0.09413290$.
5. Comprime la señal en dos y asume que la dispersión de la señal ECG es DCT. La Figura 2.10 muestra los resultados obtenidos y se observa el $MSE = 0.01529000$.
6. Comprime la señal en dos y asume que la dispersión de la señal ECG es WHT. La Figura 2.11 muestra los resultados obtenidos y se observa el $MSE = 0.03111390$.

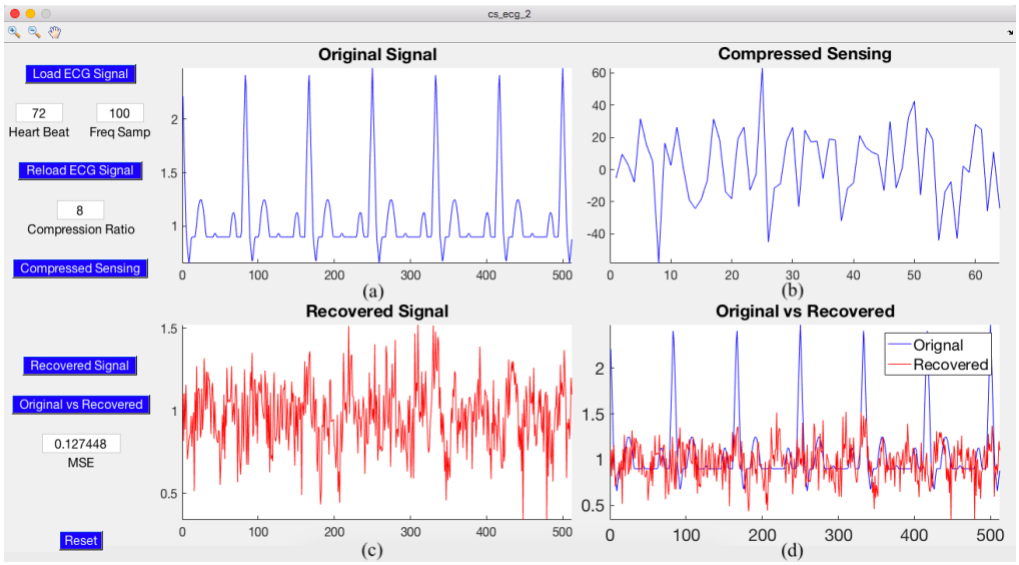


Figura 2.7 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

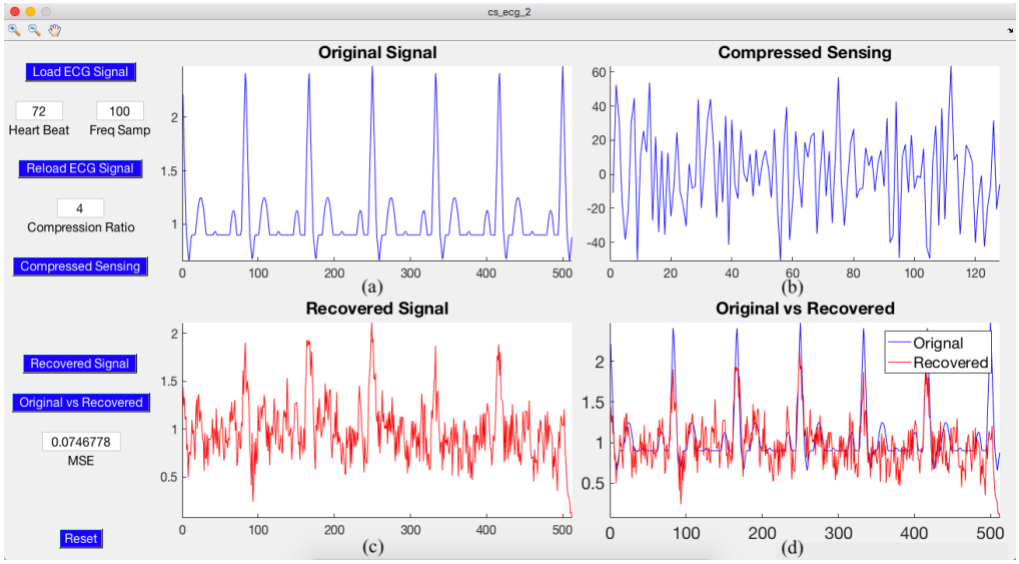


Figura 2.8 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

2.6.2 Arquitectura digital [13]

La arquitectura digital [13] fue reproducida en un kit de desarrollo DE2-115. El kit incluye un chip FPGA *Cyclone* IV EP4CE115F29C7. La herramienta de síntesis fue Quartus Prime 17.0 de Altera® y VHDL como lenguaje de descripción de *hardware*. La herramienta de simulación fue ModelSim de Mentor Graphics®. La Figura 2.12 muestra la *app* desarrollada para este método. El funcionamiento de la *app* se explica a continuación:

- *Load ECG Signal*: Grafica en plot 1 la señal ECG simulada [12].
- *Heart Beat*: Despliega la frecuencia cardiaca de la señal ECG.

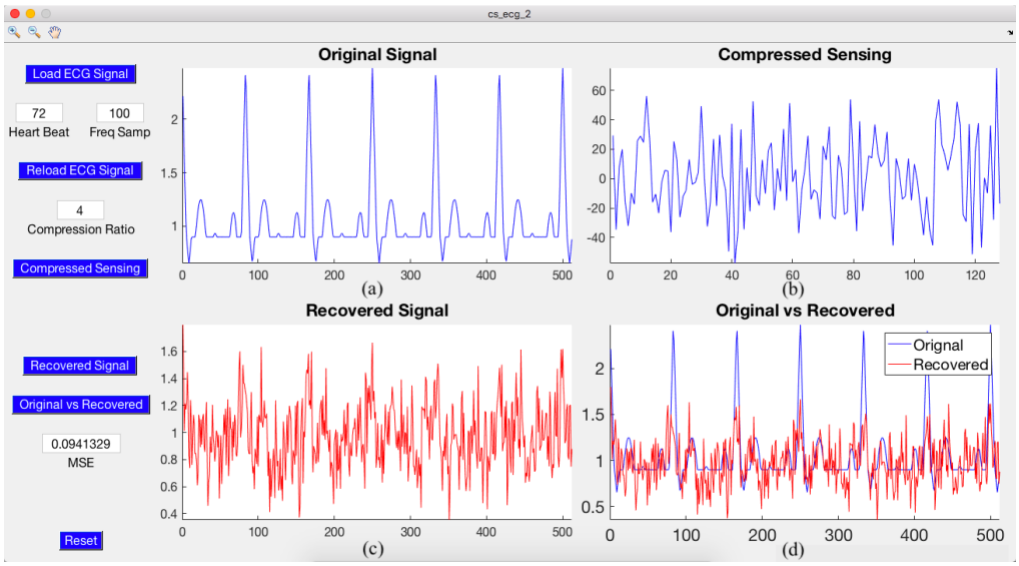


Figura 2.9 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

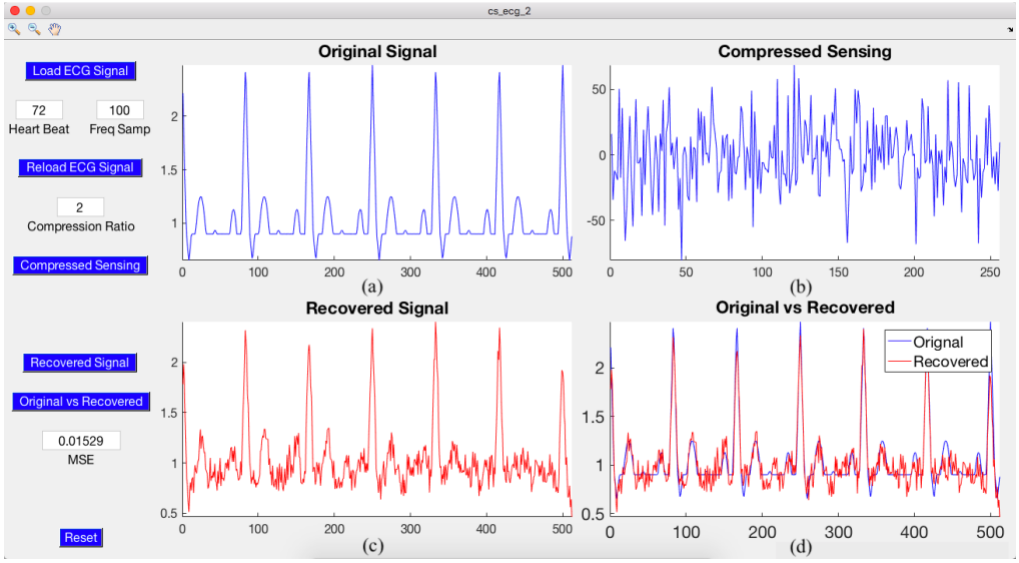


Figura 2.10 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

- *Freq Samp*: Despliega la frecuencia de muestreo de la señal ECG.
- *Quantized ECG Signal*: Grafica en plot 2 la señal ECG cuantificada.
- *Compressed Sensing*: Genera la matriz de muestreo $\mathbf{\Phi}$, multiplica la matriz por la señal ECG y grafica en plot 2 el vector $\mathbf{y}$.
- *Recovered Signal*: Aplica optimización l_1 [23] para encontrar el vector disperso, lo multiplica por la matriz de dominio $\mathbf{\Psi}$ para recuperar la señal y lo grafica en plot 3.
- *Original vs Recovered*: Grafica en plot 4 la comparación entre la señal original *vs* la señal recuperada.
- *MSE*: Muestra el MSE de la comparación entre la señal original *vs* la señal recuperada.
- *Reset*: Limpia los plots e inicializa todas las variables en cero.

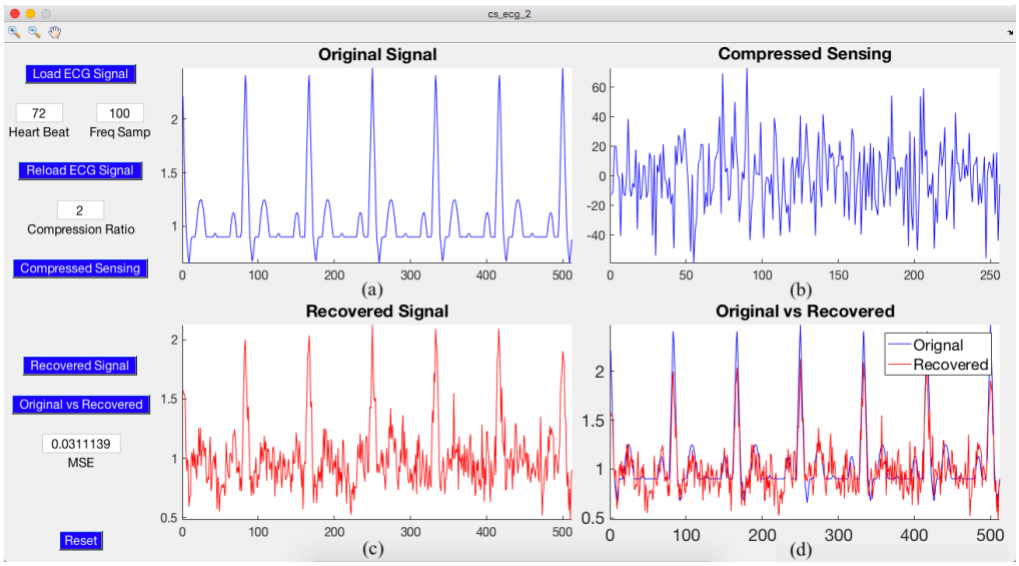


Figura 2.11 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_1$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

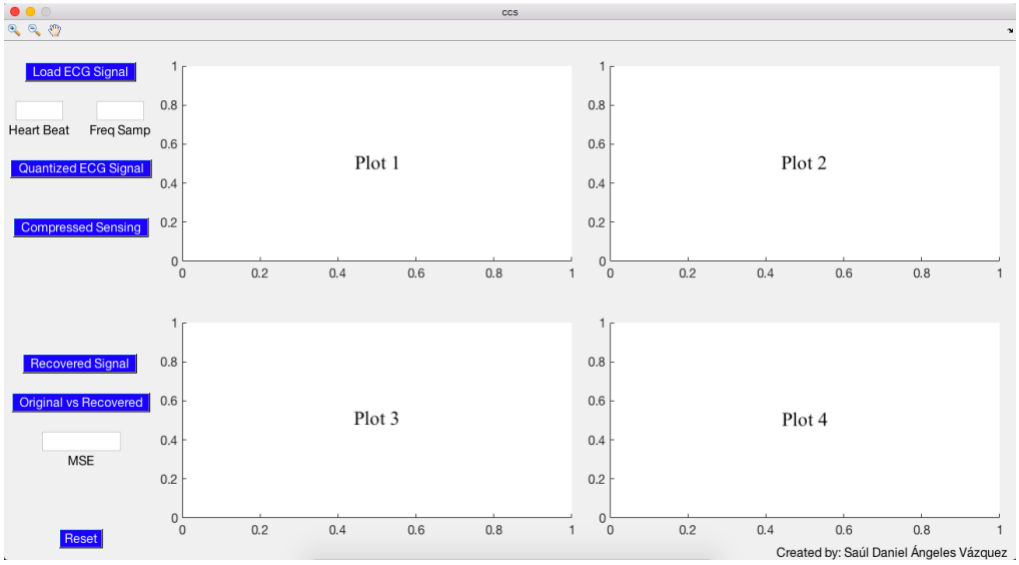


Figura 2.12 *App* para muestreo compreso por arquitectura digital [13].

Los tres escenarios en los que se realizó este método son:

1. Arquitectura digital [13] para CR_1: Comprime la señal en ocho. La Figura 2.13 muestra los diagramas RTL de los cuatro módulos de la arquitectura digital [13] y la Figura 2.14 muestra la interconexión de dichos módulos. La información de síntesis de la Tabla 2.1 indica que la arquitectura digital [13] para CR_1 utiliza 1,750 elementos lógicos, 974 registros y la frecuencia máxima de operación es 166.5MHz. La Figura 2.15 muestra la señal ECG cuantificada que se va a comprimir. La Figura 2.16 muestra los resultados de la compresión. Los resultados en la recuperación de la señal son:

   a) La Figura 2.17 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.22561500$.
   b) La Figura 2.18 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.14120200$.

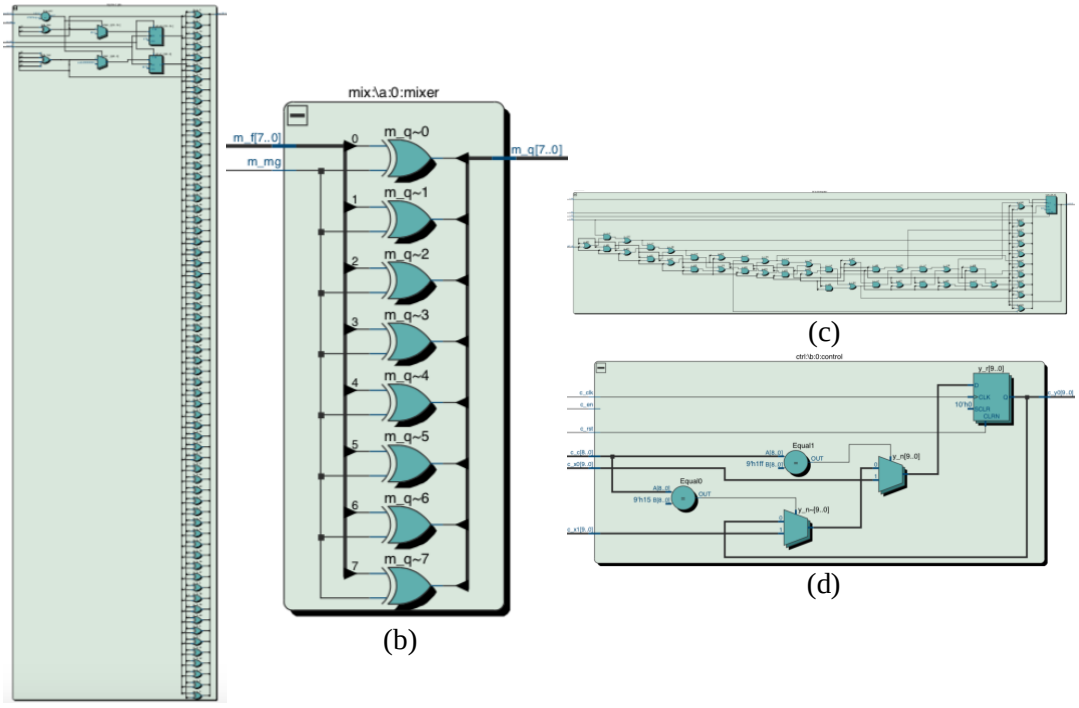


Figura 2.13 Diagrama RTL para CR_1 de: (a) Generador de la matriz de muestreo $\mathbf{\Phi}_2$, (b) *Mixer*, (c) *Integrator* y (d) Multiplexor.

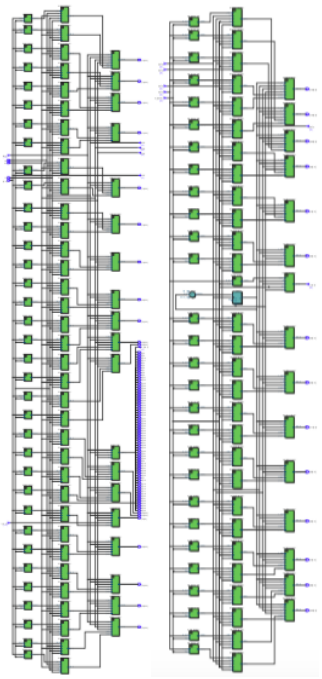


Figura 2.14 Diagrama RTL de la arquitectura digital [13] para CR_1.

2. Arquitectura digital [13] para CR_2: Comprime la señal en cuatro. La Figura 2.19 muestra los diagramas RTL de los cuatro módulos de la arquitectura digital [13] y la Figura 2.20 muestra la interconexión de dichos módulos. La información de síntesis de la Tabla 2.2

Tabla 2.1 Reporte de síntesis de la arquitectura digital [13] para CR_1.

Fitter Status	Successful - Sun May 27 19:55:21 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	ccs
Top-level Entity Name	ccs
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	1,750 / 114,480 (2 %)
Total combinational functions	1,717 / 114,480 (1 %)
Dedicated logic registers	974 / 114,480 (< 1 %)
Total registers	974
Total pins	395 / 529 (75 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	166.5MHz

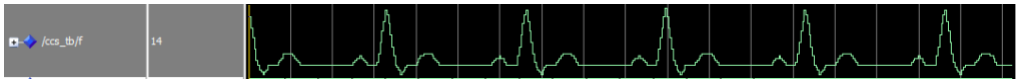


Figura 2.15 Señal ECG cuantificada.

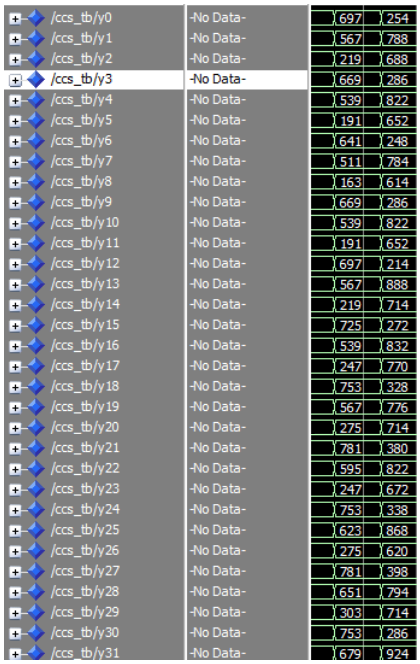


Figura 2.16 Resultados de compresión con $\mathbf{\Phi}_2$ y CR_1.

indica que la arquitectura digital [13] para CR_2 utiliza 3,447 elementos lógicos, 1,615 registros y la frecuencia máxima de operación es de 167.2MHz. La Figura 2.21 muestra los resultados de la compresión. Los resultados en la recuperación de la señal son:

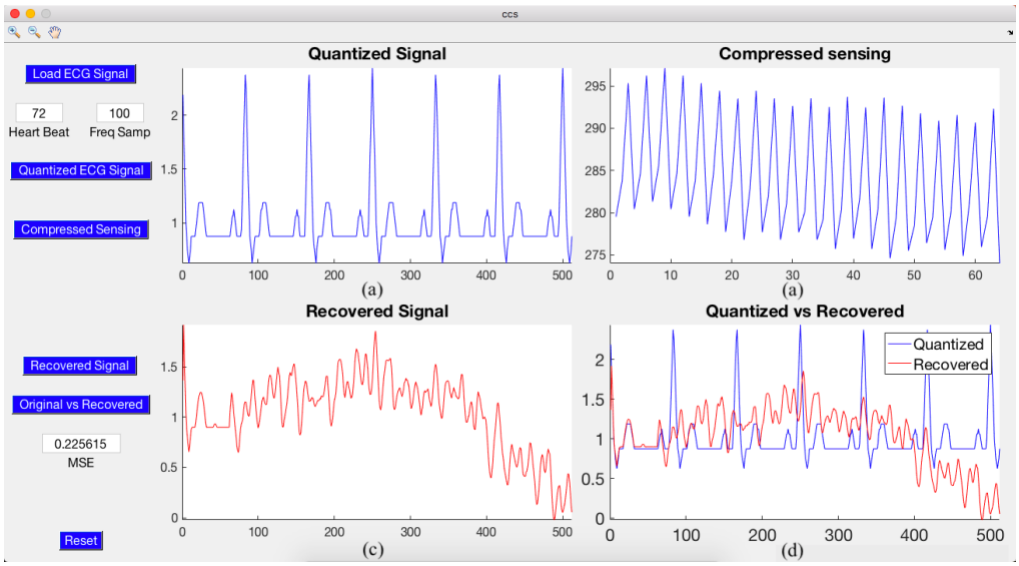


Figura 2.17 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

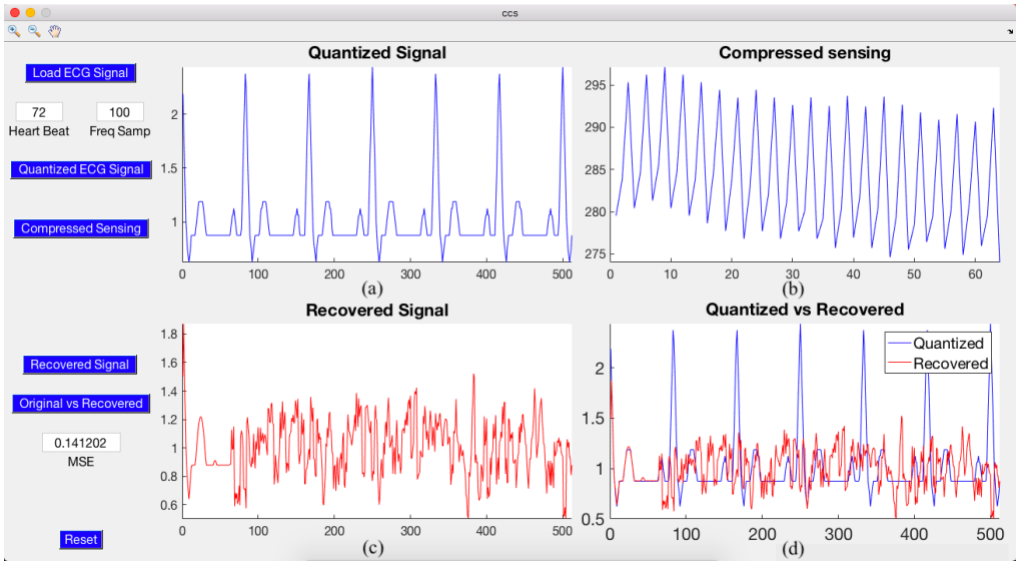


Figura 2.18 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

a) La Figura 2.22 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.05316020$.
b) La Figura 2.23 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.07565700$.

3. Arquitectura digital [13] para CR_3: Comprime la señal en dos. La Figura 2.24 muestra los diagramas RTL de los cuatro módulos de la arquitectura digital [13] y la Figura 2.25 muestra la interconexión de dichos módulos. La información de síntesis de la Tabla 2.3 indica que la arquitectura digital [13] para CR_3 utiliza 7,157 elementos lógicos, 2,762 registros y la frecuencia máxima de operación es de 145.5MHz. La Figura 2.26 muestra los resultados de la compresión. Los resultados en la recuperación de la señal son:

a) La Figura 2.27 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.00743218$.

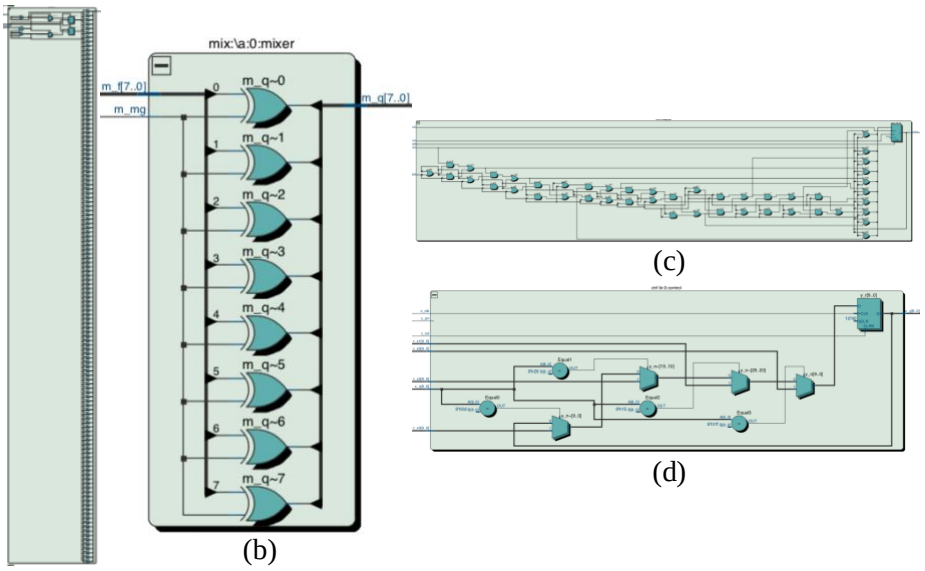


Figura 2.19 Diagrama RTL para CR_2 de: (a) Generador de la matriz de muestreo $\mathbf{\Phi}_2$, (b) *Mixer*, (c) *Integrator* y (d) Multiplexor.

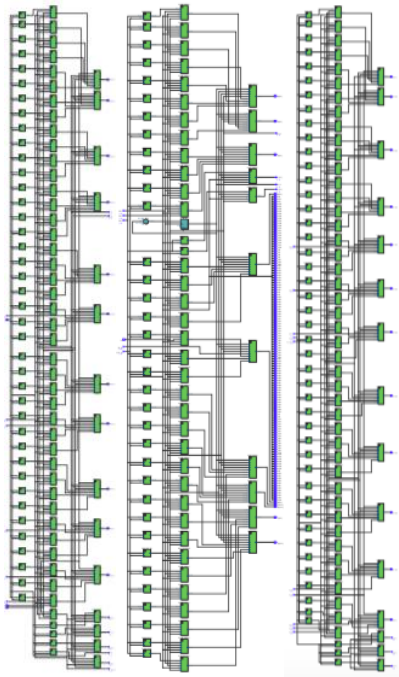


Figura 2.20 Diagrama RTL de la arquitectura digital [13] para CR_2.

b) La Figura 2.28 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.03263760$.

Tabla 2.2 Reporte de síntesis de la arquitectura digital [13] para CR_2.

Fitter Status	Successful - Mon May 28 16:19:17 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	ccs
Top-level Entity Name	ccs
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	3,447 / 114,480 (3 %)
Total combinational functions	3,418 / 114,480 (3 %)
Dedicated logic registers	1,615 / 114,480 (1 %)
Total registers	1615
Total pins	459 / 529 (87 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	167.2MHz

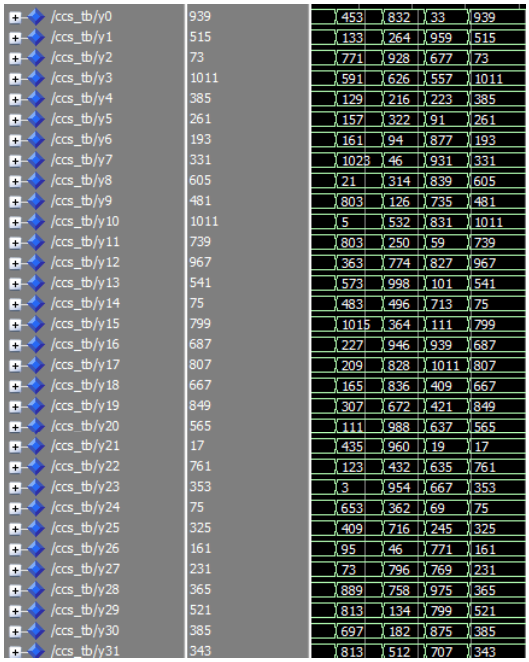


Figura 2.21 Resultados de compresión con $\mathbf{\Phi}_2$ y CR_2.

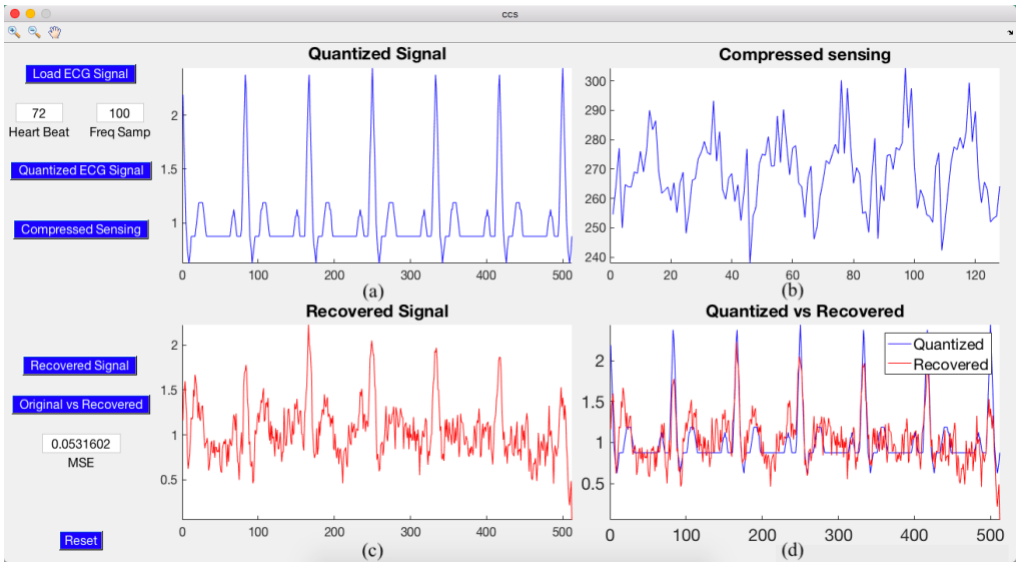


Figura 2.22 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

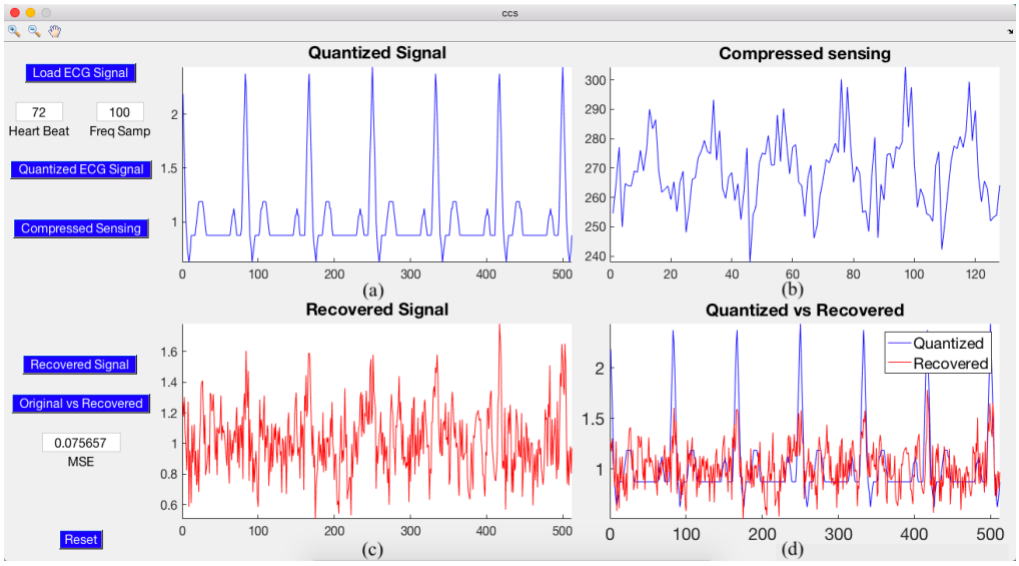


Figura 2.23 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

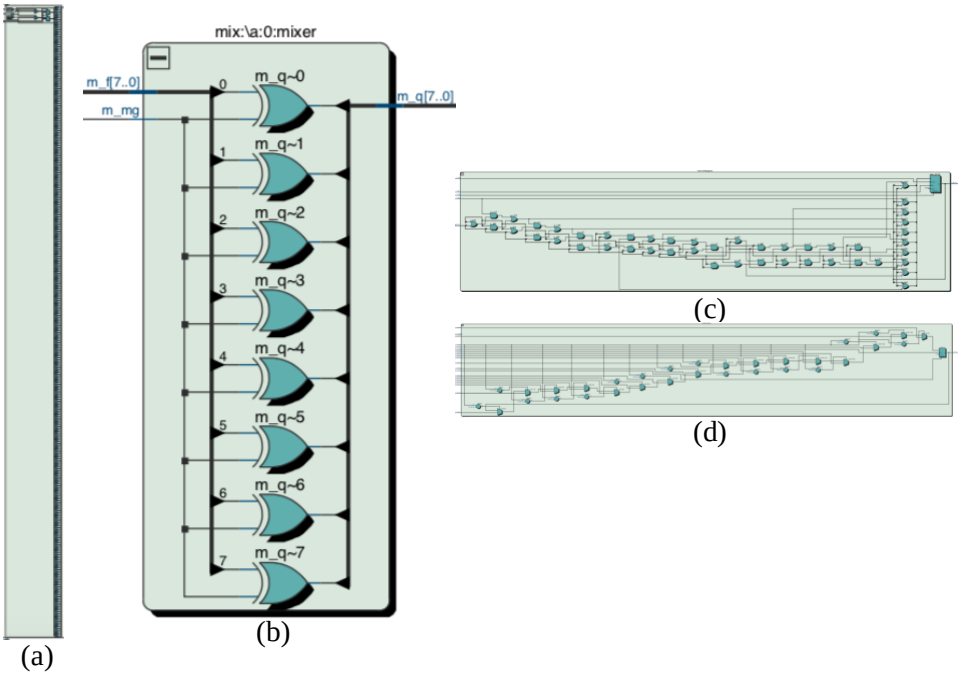


Figura 2.24 Diagrama RTL para CR_3 de: (a) Generador de la matriz de muestreo $\mathbf{\Phi}_2$, (b) *Mixer*, (c) *Integrator* y (d) Multiplexor.

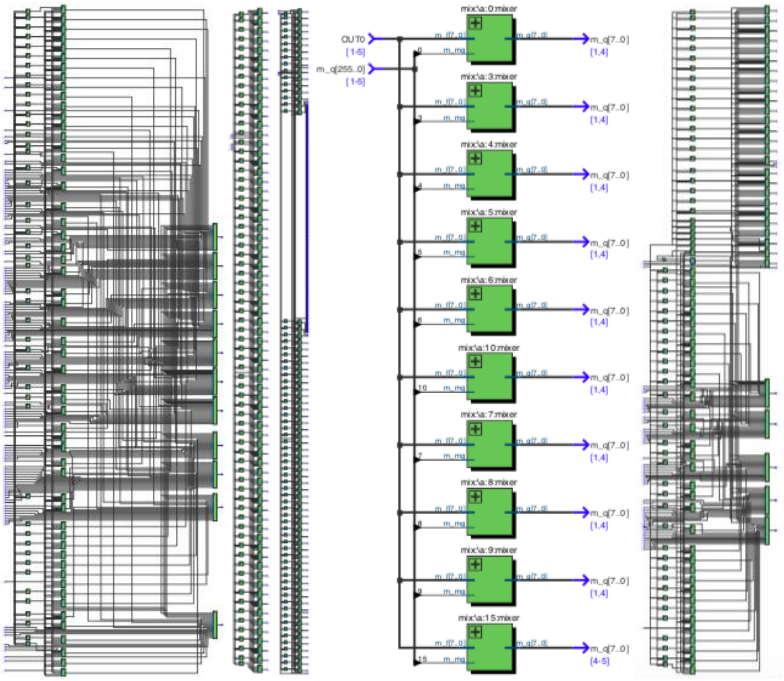


Figura 2.25 Diagrama RTL de la arquitectura digital [13] para CR_3.

Tabla 2.3 Reporte de síntesis de la arquitectura digital [13] para CR_3.

Fitter Status	Successful - Tue May 29 15:07:04 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	ccs
Top-level Entity Name	ccs
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	7,157 / 114,480 (6 %)
Total combinational functions	7,141 / 114,480 (6 %)
Dedicated logic registers	2,762 / 114,480 (2 %)
Total registers	2762
Total pins	427 / 529 (81 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	145.5MHz

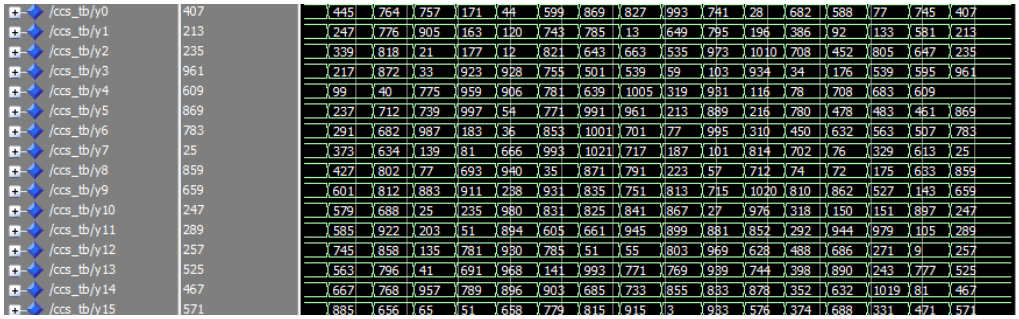


Figura 2.26 Resultados de compresión con $\mathbf{\Phi}_2$ y CR_3.

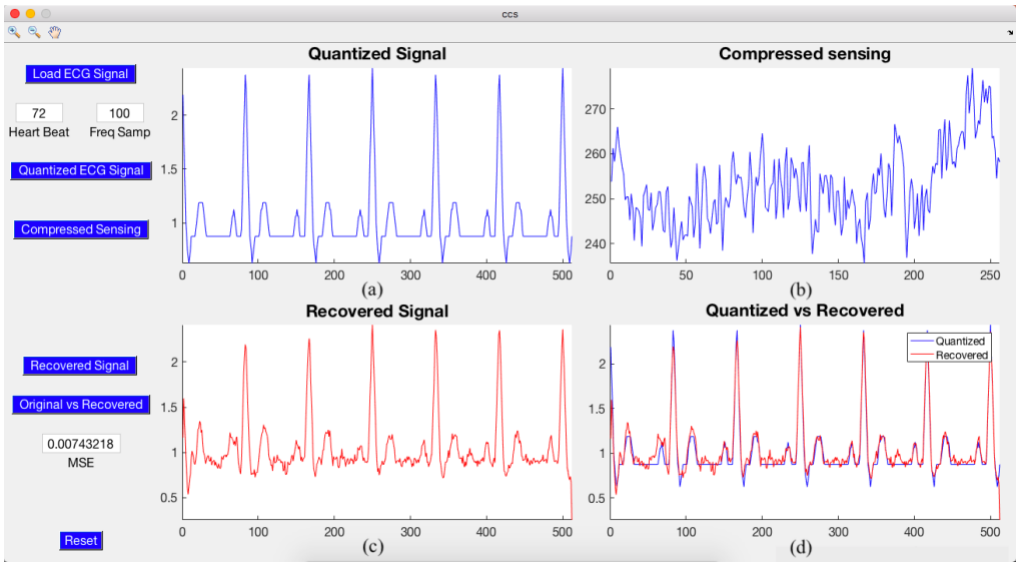


Figura 2.27 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal. Original *vs* señal recuperada.

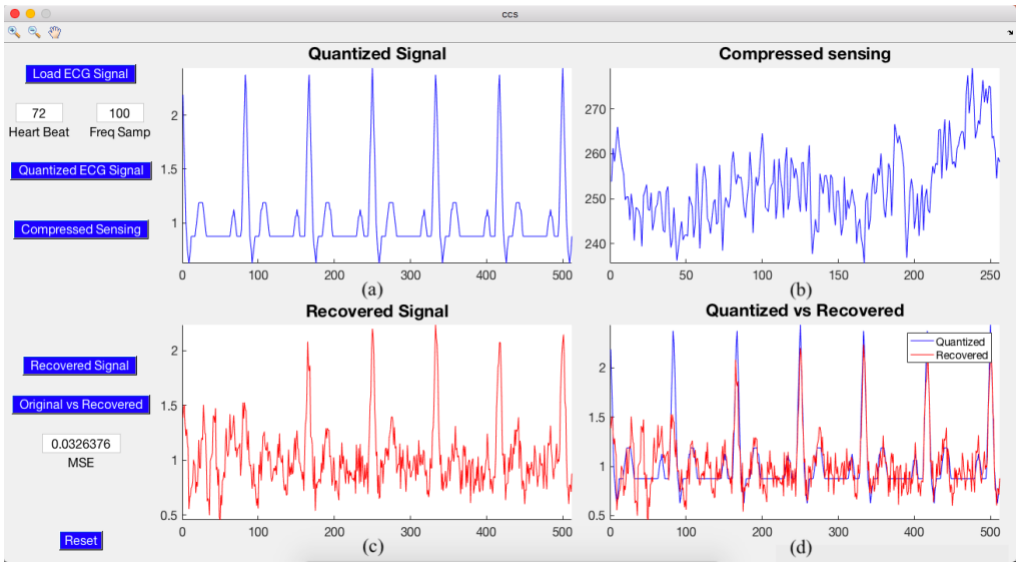


Figura 2.28 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_2$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal. Original *vs* señal recuperada.

3 Algoritmo y arquitecturas digitales: métodos propuestos

Este capítulo presenta las contribuciones de investigación basadas en nuevos métodos de compresión para señales ECG. La sección 3.1 diseña la arquitectura digital basada en el algoritmo CAC-OST [21]. La sección 3.2 presenta un nuevo algoritmo para la generación de la matriz de muestro que hemos llamado *Cellular Automata Chaos for Streaming Signals* (CAC-SS). La sección 3.3 diseña la arquitectura digital basada en el algoritmo CAC-SS. El capítulo finaliza con un comparativo en donde se enlistan las ventajas de CAC-SS sobre CAC-OST.

3.1 Arquitectura digital CAC-OST

El diseño propuesto de la arquitectura digital que corresponde al algoritmo CAC-OST, mostrado en la Figura 3.1, divide el algoritmo en cinco módulos digitales:

1. Memoria.
2. Umbral.
3. Semilla inicial.
4. Generador de la matriz de muestreo **Φ**.
5. Compresor.

La arquitectura digital fue implementada en un kit de desarrollo DE2-115. El kit incluye un chip FPGA *Cyclone* IV EP4CE115F29C7. La herramienta de síntesis fue Quartus Prime 17.0 de Altera® y VHDL como lenguaje de descripción de *hardware*. La herramienta de simulación fue ModelSim de Mentor Graphics®.

3.1.1 Memoria

El algoritmo requiere que la señal sea finita, para ello se utiliza un bloque de memoria en donde la señal es almacenada. La Figura 3.2 muestra la memoria y la ecuación 3.1 calcula el dimensionamiento en bits de dicha memoria:

$$Storage\ (bits) = no.samples \times bits\ per\ sample, \tag{3.1}$$

por tanto, si la señal **f** está compuesta por 512 muestras, y cada muestra está representada por 8 bits, el almacenamiento requerido es de 4,096 bits. El módulo consta de:

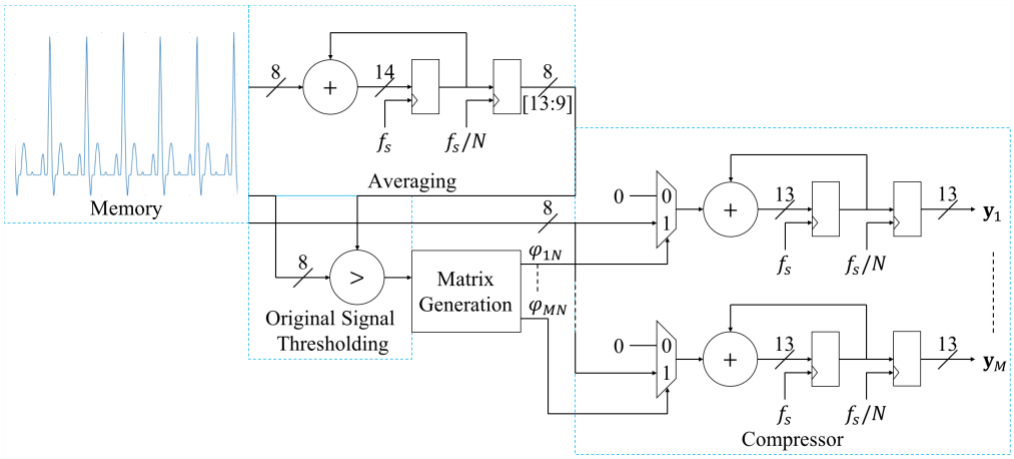


Figura 3.1 Arquitectura digital CAC.OST.

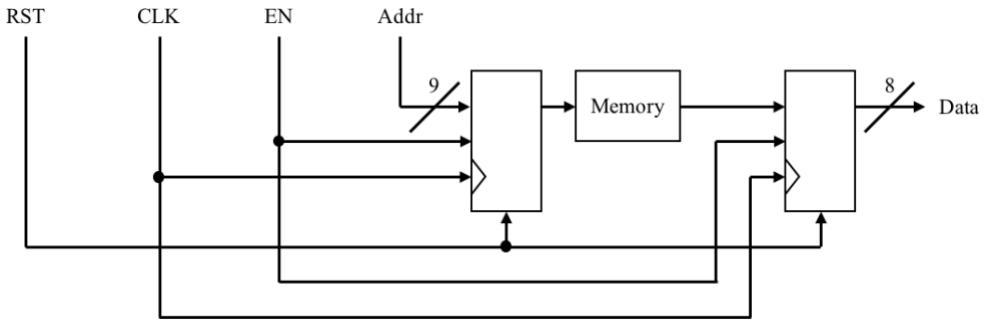


Figura 3.2 Memoria.

- Una señal de entrada de 9 bits que corresponde a la dirección en memoria que se desea leer.
- Una señal de salida de 8 bits que corresponde al dato almacenado en memoria en la dirección indicada.
- Un registro de 9 bits para direcciones.
- Un registro de 8 bits para datos.

a) Diseño.

La Figura 3.3 muestra el diagrama RTL que se obtiene del diseño de la memoria y se observa la conexión entre el registro de direcciones, el de datos y la memoria. La información de síntesis de la Tabla 3.1 indica que la memoria utiliza 18 elementos lógicos, 10 registros, 4,096 bits y la frecuencia máxima de operación es de 1,226.99MHz.

b) Simulación.

El vector de entrada Addr elegido con el propósito de simular y probar el correcto funcionamiento de la memoria diseñada es:

$$Addr = 000000001, 000000100, 000010000, 001000000.$$

La Tabla 3.2 indica los valores esperados $Data_0$ y los valores obtenidos $Data_1$. La Figura 3.4

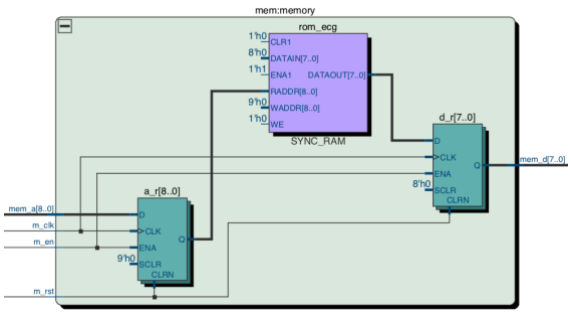


Figura 3.3 Diagrama RTL de la memoria.

Tabla 3.1 Reporte de síntesis de la memoria.

Fitter Status	Successful - Thu Aug 09 22:41:38 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	memory
Top-level Entity Name	memory
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	18 / 114,480 (< 1 %)
Total combinational functions	9 / 114,480 (< 1 %)
Dedicated logic registers	10 / 114,480 (< 1 %)
Total registers	10
Total pins	20 / 529 (4 %)
Total virtual pins	0
Total memory bits	4,096 / 3,981,312 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	1,226.99MHz

Tabla 3.2 Valores esperados y obtenidos de la memoria.

Addr	000000001	000000100	000010000	001000000
$Data_0$	00011110	00010000	00001110	00010000
$Data_1$	00011110	00010000	00001110	00010000

Figura 3.4 Resultados de simulación de la memoria.

muestra los resultados obtenidos en la simulación para la memoria diseñada.

3.1.2 Umbral

Este módulo calcula el límite o umbral leyendo las N muestras de la señal finita almacenada en memoria (lee una muestra por ciclo de reloj) y suma el valor absoluto de cada muestra. Después de haber acumulado las N muestras, el módulo elimina los N bits menos significativos del resultado para implementar la división sobre el valor N, obtener el promedio y asignar dicho valor como límite o umbral. El módulo consta de:

- Una señal de entrada de 8 bits que corresponde al valor de la muestra.
- Una señal de salida de 8 bits que corresponde al valor del límite o umbral.
- Un acumulador que permite ir sumando el valor de las muestras.

a) Diseño.

La Figura 3.5 muestra el diagrama RTL que se obtiene del diseño del umbral y se observa la conexión del acumulador. La información de síntesis de la Tabla 3.3 indica que el umbral utiliza 23 elementos lógicos, 14 registros y la frecuencia máxima de operación es de 427.9MHz.

b) Simulación.

El vector de entrada ECG elegido con el propósito de simular y probar el correcto funcionamiento del umbral diseñado corresponde a un *frame* de 512 muestras de la señal ECG vía simulador [12]. La Tabla 3.4 indica los valores esperados Thr_0 y los valores obtenidos Thr_1. La Figura 3.6 muestra los resultados obtenidos en la simulación para el umbral diseñado.

3.1.3 Semilla inicial

La semilla inicial, que se convierte en la primera fila de la matriz de muestreo **Φ**, la genera este módulo leyendo las *N* muestras de la señal almacenadas en la memoria y comparando cada muestra contra el límite o umbral. Si el valor absoluto de la muestra es mayor que el límite o umbral, al coeficiente correspondiente se le asigna un 1, de lo contrario se le asigna un 0. El módulo consta de:

- Dos señales de entrada de 8 bits que corresponden al valor de la muestra y al valor del límite o umbral.
- Una señal de salida de 1 bit que corresponde al coeficiente de la matriz de muestreo **Φ**.
- Un comparador que permite contrastar el valor de la muestra y el valor del límite o umbral.

a) Diseño.

La Figura 3.7 muestra el diagrama RTL que se obtiene del diseño de la semilla inicial y se observa la conexión del comparador. La información de síntesis de la Tabla 3.5 indica que la semilla inicial utiliza 8 elementos lógicos.

b) Simulación.

El vector de entrada ECG elegido con el propósito de simular y probar el correcto funcionamiento de la semilla inicial diseñada corresponde a un *frame* de 512 muestras de la señal ECG vía simulador [12]. La Tabla 3.6 indica los valores esperados OST_0 y los valores obtenidos OST_1. La Figura 3.8 muestra los resultados obtenidos en la simulación para la semilla inicial diseñada.

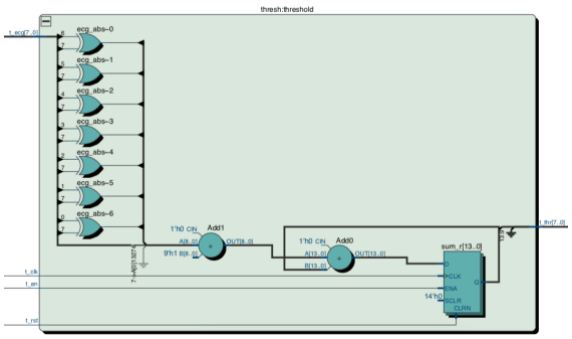


Figura 3.5 Diagrama RTL del umbral.

Tabla 3.3 Reporte de síntesis del umbral.

Fitter Status	Successful - Fri Aug 10 12:01:28 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	thresh
Top-level Entity Name	thresh
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	23 / 114,480 (< 1 %)
Total combinational functions	23 / 114,480 (< 1 %)
Dedicated logic registers	14 / 114,480 (< 1 %)
Total registers	14
Total pins	19 / 529 (4 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	427.9MHz

Tabla 3.4 Valores esperados y obtenidos del umbral.

CLK	1	2	3	4	5	6	7	8	…	505	506	507	508	509	510	511	512	1
ECG	0	35		30	26	21	16	14		26	21	16	14	12	11	10	11	12
Thr_0	0									0								16
Thr_1	0									0								16

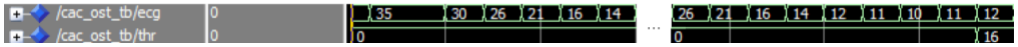


Figura 3.6 Resultados de simulación del umbral.

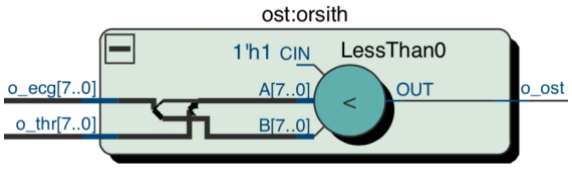


Figura 3.7 Diagrama RTL de la semilla inicial.

Tabla 3.5 Reporte de síntesis de la semilla inicial.

Fitter Status	Successful - Sun Aug 12 15:44:12 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	ost
Top-level Entity Name	ost
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	8 / 114,480 (< 1 %)
Total combinational functions	8 / 114,480 (< 1 %)
Dedicated logic registers	0 / 114,480 (0 %)
Total registers	0
Total pins	17 / 529 (3 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

Tabla 3.6 Valores esperados y obtenidos de la semilla inicial.

CLK	1	2	3	4	5	6	7	…	505	506	507	508	509	510	511	512	1
ECG	12	14	35	30	26	21	16		26	21	16	14	12	11	10	11	12
OST_0	0		1						1			0					
OST_1	0		1						1			0					

Figura 3.8 Resultados de simulación de la semilla inicial.

3.1.4 Generador de la matriz de muestreo Φ

Las otras filas de la matriz de muestreo **Φ** son generadas utilizando la ecuación (2.1):

- El coeficiente $\boldsymbol{\varphi}_{(i)(1)}$ se genera conectando el coeficiente $\boldsymbol{\varphi}_{(i-1)(1)}$ y el coeficiente $\boldsymbol{\varphi}_{(i-1)(2)}$ a una compuerta OR.
- Los coeficientes $\boldsymbol{\varphi}_{(i)(j)}$ al $\boldsymbol{\varphi}_{(i)(N-1)}$ se generan conectando el coeficiente $\boldsymbol{\varphi}_{(i-1)(j)}$ y el coeficiente $\boldsymbol{\varphi}_{(i-1)(j+1)}$ a una compuerta OR y el resultado se conecta con el coeficiente $\boldsymbol{\varphi}_{(i-1)(j-1)}$ a una compuerta XOR.
- El coeficiente $\boldsymbol{\varphi}_{(i)(N)}$ se genera conectando el coeficiente $\boldsymbol{\varphi}_{(i-1)(N-1)}$ y el coeficiente $\boldsymbol{\varphi}_{(i-1)(N)}$ a una compuerta XOR.

La Figura 3.9 muestra el generador de la matriz de muestreo **Φ**. El módulo consta de:

- Una señal de entrada de 1 bit.
- Una señal de salida de 1 bit.
- Dos registros de 1 bit.

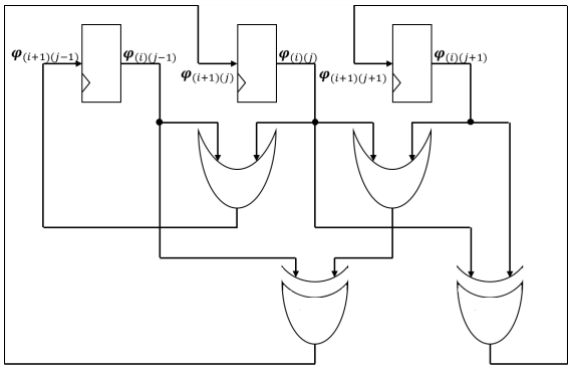


Figura 3.9 Generador CAC-OST de la matriz de muestreo **Φ**.

a) Diseño.

La Figura 3.10 muestra el diagrama RTL que se obtiene del diseño del generador de la matriz de muestreo **Φ** y se observa la conexión de los registros y las compuertas OR y XOR. La información de síntesis de la Tabla 3.7 indica que el generador de la matriz de muestreo **Φ** utiliza 2 elementos lógicos, 2 registros y la frecuencia máxima de operación es de 1,084.6MHz.

b) Simulación.

El vector de entrada con el propósito de simular y probar el correcto funcionamiento del generador de la matriz de muestreo **Φ** corresponde a la semilla inicial generada con el módulo 3.1.3. La Figura 3.11 muestra los resultados obtenidos en la simulación para el generador de la matriz de muestreo **Φ** diseñado.

3.1.5 Compresor

Este módulo multiplica la señal por la matriz de muestreo **Φ** sin la necesidad de utilizar multiplicadores, debido a que la función de multiplicación se implementa con un multiplexor de la siguiente manera:

- Si el coeficiente correspondiente de la matriz de muestreo **Φ** es 0, al valor de la muestra se le asigna un 0 y se acumula.
- Si el coeficiente correspondiente de la matriz de muestreo **Φ** es 1, el valor de la muestra se acumula.

El vector **y** es mostrado una vez que la multiplicación ha sido realizada. El módulo consta de:

- Una señal de entrada de 1 bit que corresponde al coeficiente de la matriz de muestreo **Φ**.
- Una señal de entrada de 8 bits que corresponde al valor de la muestra de la señal.
- Una señal de salida de 13 bits que corresponde al vector **y**.
- Un multiplexor que implementa la función de multiplicación.
- Un acumulador que permite sumar el resultado de la multiplicación.

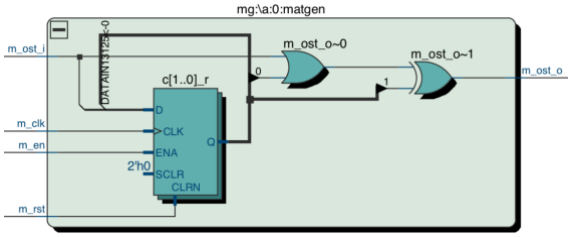


Figura 3.10 Diagrama RTL del generador de la matriz de muestreo **Φ**.

Tabla 3.7 Reporte de síntesis del generador de la matriz de muestreo **Φ**.

Fitter Status	Successful - Sun Aug 12 21:21:12 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	mg
Top-level Entity Name	mg
Family	Cyclone IV E
Device	EP4CE6E22C6
Timing Models	Final
Total logic elements	2 / 114,480 (< 1 %)
Total combinational functions	1 / 114,480 (< 1 %)
Dedicated logic registers	2 / 114,480 (< 1 %)
Total registers	2
Total pins	5 / 529 (< 1 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	1,084.6MHz

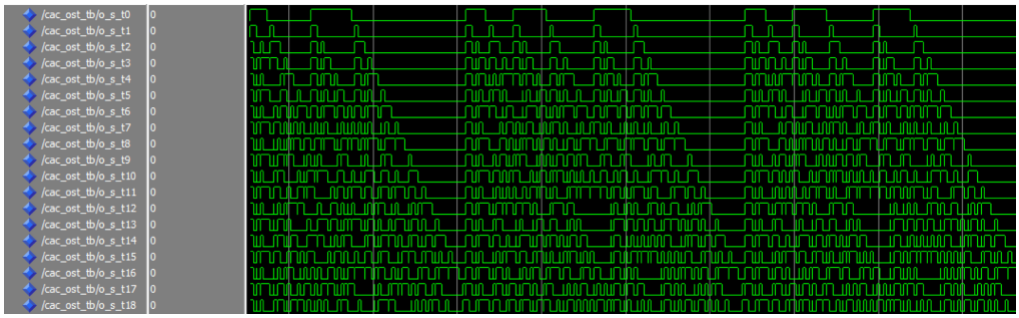


Figura 3.11 Resultados de simulación del generador de la matriz de muestreo **Φ**.

a) Diseño.

La Figura 3.12 muestra el diagrama RTL que se obtiene del diseño del compresor y se observa la conexión del multiplexor y acumulador. La información de síntesis de la Tabla 3.8 indica que el compresor utiliza 24 elementos lógicos, 13 registros y la frecuencia máxima de operación es de 473.04MHz.

b) Simulación.

Los vectores de entrada OST y ECG elegidos con el propósito de simular y probar el correcto

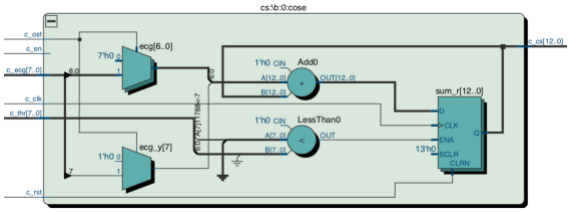


Figura 3.12 Diagrama RTL del compresor.

Tabla 3.8 Reporte de síntesis del compresor.

Fitter Status	Successful - Mon Aug 13 09:02:06 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	cs
Top-level Entity Name	cs
Family	Cyclone IV E
Device	EP4CE6E22C6
Timing Models	Final
Total logic elements	24 / 114,480 (< 1 %)
Total combinational functions	24 / 114,480 (< 1 %)
Dedicated logic registers	13 / 114,480 (< 1 %)
Total registers	13
Total pins	33 / 529 (6 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	473.04MHz

funcionamiento del compresor corresponden a los coeficientes generados con el módulo 3.1.4 y un *frame* de 512 muestras de la señal ECG vía simulador [12] respectivamente. La Figura 3.13 muestra los resultados obtenidos en la simulación para el compresor diseñado.

3.2 Algoritmo CAC-SS

El algoritmo propuesto CAC-SS genera los coeficientes de la matriz de muestreo **Φ** en forma conveniente para señales que fluyen tipo *streaming*. El algoritmo funciona de la siguiente manera:

1. La semilla inicial se elige de forma arbitraria y genera la primera columna de la matriz de muestreo **Φ**.
2. Las otras columnas de la matriz de muestreo **Φ** son generadas utilizando la regla [22]:

$$\boldsymbol{\varphi}_{(i)(j+1)} = \boldsymbol{\varphi}_{(i-1)(j)} \oplus \left(\boldsymbol{\varphi}_{(i)(j)} \vee \boldsymbol{\varphi}_{(i+1)(j)} \right) \tag{3.2}$$

La Figura 3.14 muestra la matriz de muestreo **Φ** generada por CAC-SS.

El algoritmo es adecuado para señales que fluyen tipo *streaming* debido a que no requiere que la señal sea finita para generar la matriz de muestreo **Φ**.

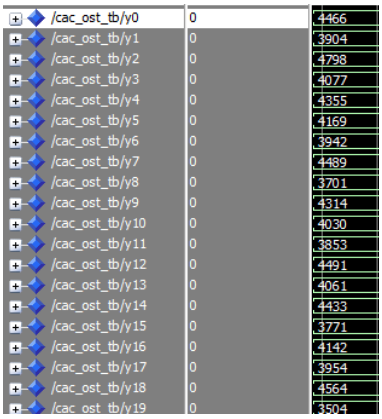


Figura 3.13 Resultados de simulación del compresor.

Figura 3.14 Matriz de muestreo **Φ** generada por CAC-SS con $N = 32$ y $M = 16$.

3.3 Arquitectura digital CAC-SS

El diseño propuesto de la arquitectura digital que corresponde al algoritmo CAC-SS, mostrado en la Figura 3.15, divide el algoritmo en tres módulos digitales:

1. Generador de la matriz de muestreo **Φ**.
2. Combinador.
3. Acumulador.

La arquitectura digital fue implementada en un kit de desarrollo DE2-115. El kit incluye un chip FPGA *Cyclone* IV EP4CE115F29C7. La herramienta de síntesis fue Quartus Prime 17.0 de Altera® y VHDL como lenguaje de descripción de *hardware*. La herramienta de simulación fue ModelSim de Mentor Graphics®.

3.3.1 Generador de la matriz de muestreo Φ

Una vez que la semilla inicial ha sido seleccionada para generar la primera columna, este módulo genera las otras columnas utilizando la ecuación (3.2):

- El coeficiente $\boldsymbol{\varphi}_{(1)(j)}$ se genera conectando el coeficiente $\boldsymbol{\varphi}_{(1)(j-1)}$ y el coeficiente $\boldsymbol{\varphi}_{(2)(j-1)}$ a una compuerta OR.
- Los coeficientes $\boldsymbol{\varphi}_{(i)(j)}$ al $\boldsymbol{\varphi}_{(M-1)(j)}$ se generan conectando el coeficiente $\boldsymbol{\varphi}_{(i)(j-1)}$ y el coeficiente $\boldsymbol{\varphi}_{(i+1)(j-1)}$ a una compuerta OR y el resultado se conecta con el coeficiente $\boldsymbol{\varphi}_{(i-1)(j-1)}$ a una compuerta XOR.

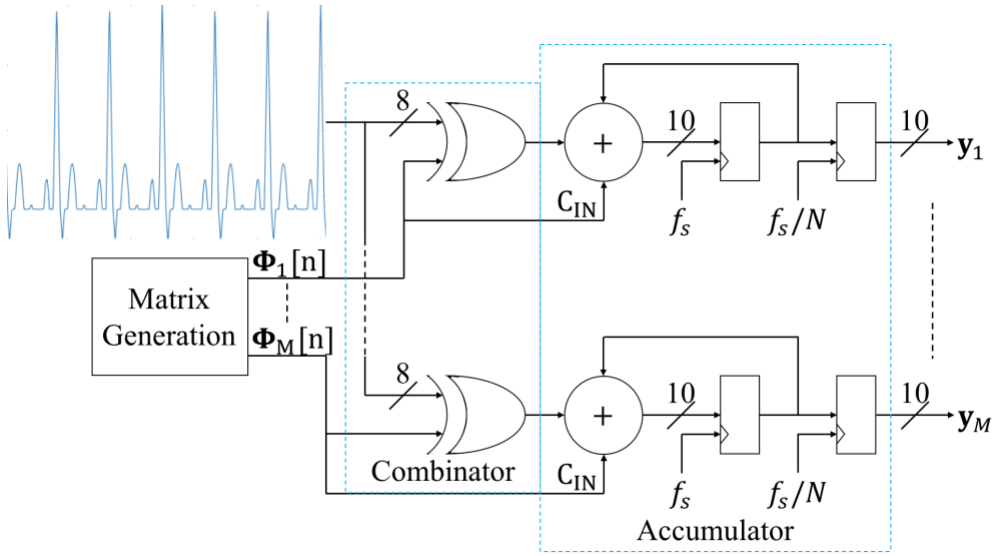


Figura 3.15 Arquitectura digital CAC-SS.

- El coeficiente $\boldsymbol{\varphi}_{(M)(j)}$ se genera conectando el coeficiente $\boldsymbol{\varphi}_{(M-1)(j-1)}$ y el coeficiente $\boldsymbol{\varphi}_{(M)(j-1)}$ a una compuerta XOR.

La Figura 3.16 muestra el generador de la matriz de muestreo **Φ**. El módulo consta de:

- Una señal de salida de M bits que corresponde a la columna generada.
- Un registro de M bits.

El diseño y simulación del módulo se realizó considerando los mismos valores de **f** y CR considerados en el marco de referencia.

1.a Diseño para CR_1.

La Figura 3.17 muestra el diagrama RTL que se obtiene del diseño del generador de la matriz de muestreo **Φ** y se observa la conexión del registro con las compuertas OR y XOR. La información de síntesis de la Tabla 3.9 indica que el generador de la matriz de muestreo **Φ** utiliza 64 elementos lógicos, 64 registros y la frecuencia máxima de operación es de 529.94MHz.

1.b Simulación para CR_1.

El vector de entrada utilizado con el propósito de simular y probar el correcto funcionamiento del generador de la matriz de muestreo **Φ** corresponde a una semilla inicial arbitraria. La Figura 3.18 muestra los resultados obtenidos en la simulación para el generador de la matriz de muestreo **Φ** diseñado.

2.a Diseño para CR_2.

La Figura 3.19 muestra el diagrama RTL que se obtiene del diseño y se observa la conexión

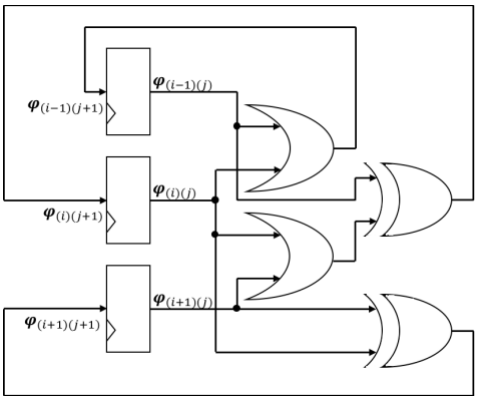


Figura 3.16 Generador CAC-SS de la matriz de muestreo **Φ**.

Figura 3.17 Diagrama RTL del generador de la matriz de muestreo **Φ** para CR_1.

del registro con las compuertas OR y XOR. La información de síntesis de la Tabla 3.10 indica que el generador utiliza 128 elementos lógicos, 128 registros y la frecuencia máxima de operación es de 512.3MHz.

2.b Simulación para CR_2.

El vector de entrada utilizado con el propósito de simular y probar el correcto funcionamiento del generador corresponde a una semilla inicial arbitraria. La Figura 3.20 muestra los resultados obtenidos en la simulación para el generador.

3.a Diseño para CR_3.

La Figura 3.21 muestra el diagrama RTL que se obtiene del diseño y se observa la conexión del registro con las compuertas OR y XOR. La información de síntesis de la Tabla 3.11 indica que

Tabla 3.9 Reporte de síntesis del generador de la matriz de muestreo **Φ** para CR_1.

Fitter Status	Successful - Tue Jun 05 22:20:16 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	mg
Top-level Entity Name	mg
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	64 / 114,480 (< 1 %)
Total combinational functions	64 / 114,480 (< 1 %)
Dedicated logic registers	64 / 114,480 (< 1 %)
Total registers	64
Total pins	67 / 529 (13 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	529.94MHz

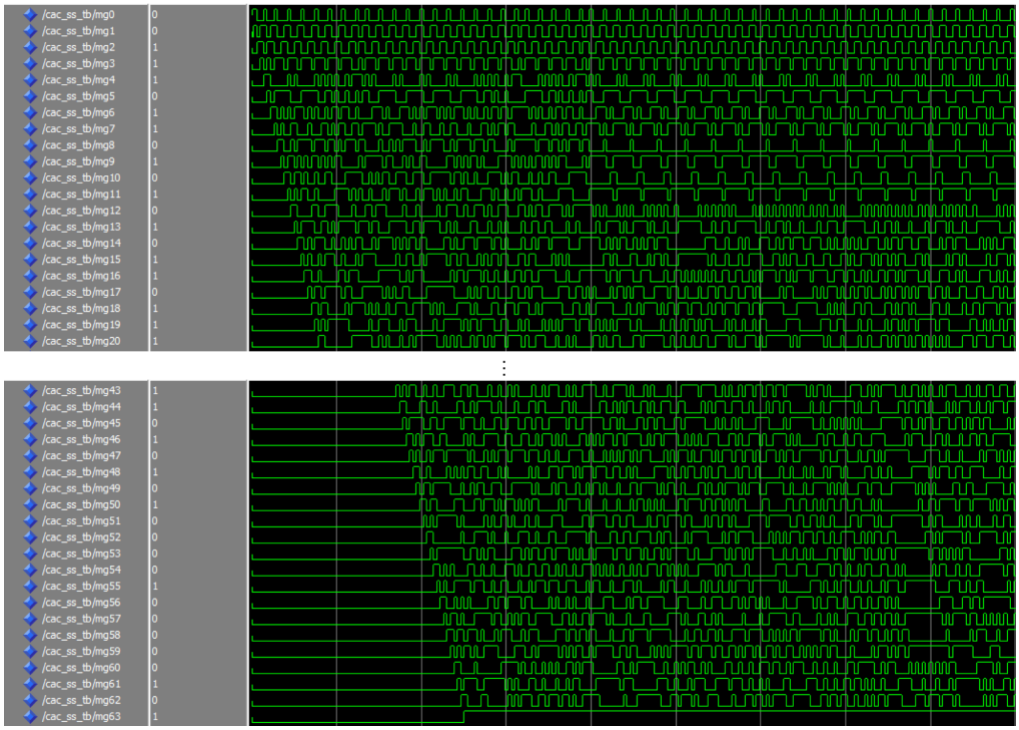


Figura 3.18 Resultados de simulación del generador de la matriz de muestreo **Φ** para CR_1.

el generador utiliza 256 elementos lógicos, 256 registros y la frecuencia máxima es de 380.95MHz.

3.b Simulación para CR_3.

El vector de entrada utilizado con el propósito de simular y probar el correcto funcionamiento del generado corresponde a una semilla inicial arbitraria. La Figura 3.22 muestra los resultados obtenido en la simulación para el generador.

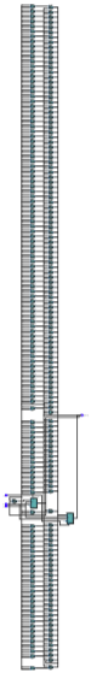


Figura 3.19 Diagrama RTL del generador de la matriz de muestreo **Φ** para CR_2.

Tabla 3.10 Reporte de síntesis del generador de la matriz de muestreo **Φ** para CR_2.

Fitter Status	Successful - Wed Jun 06 21:17:47 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	mg
Top-level Entity Name	mg
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	128 / 114,480 (< 1 %)
Total combinational functions	128 / 114,480 (< 1 %)
Dedicated logic registers	128 / 114,480 (< 1 %)
Total registers	128
Total pins	131 / 529 (25 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	512.3MHz

3.3.2 Combinador

La función de multiplicar la señal por la matriz de muestreo **Φ** se implementa conectando la muestra de la señal (una muestra por ciclo de reloj) a *M* XOR paralelas con su correspondiente coeficiente de la matriz de muestreo **Φ**, el cual también es conectado al acarreo de entrada del acumulador. El módulo consta de:

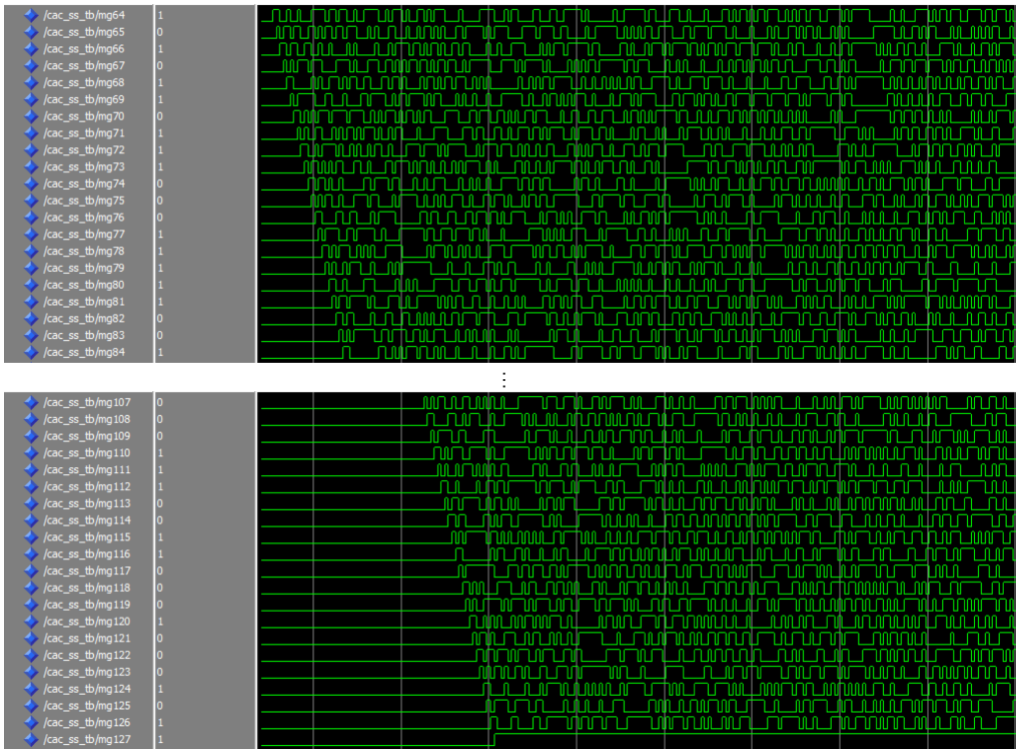


Figura 3.20 Resultados de simulación del generador de la matriz de muestreo **Φ** para CR_2.

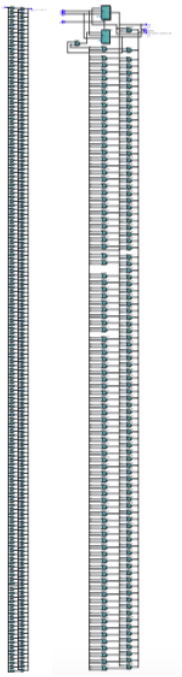


Figura 3.21 Diagrama RTL del generador de la matriz de muestreo **Φ** para CR_3.

- Una señal de entrada ECG de 8 bits que corresponde al valor de la muestra de la señal.
- Una señal de entrada MG de 1 bit que corresponde al coeficiente de la matriz de muestreo **Φ**.
- Una señal de salida Q de 8 bits que corresponde al resultado de la multiplicación.

Tabla 3.11 Reporte de síntesis del generador de la matriz de muestreo **Φ** para CR_3.

Fitter Status	Successful - Wed Jun 06 22:14:46 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	mg
Top-level Entity Name	mg
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	256 / 114,480 (< 1 %)
Total combinational functions	256 / 114,480 (< 1 %)
Dedicated logic registers	256 / 114,480 (< 1 %)
Total registers	256
Total pins	259 / 529 (49 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	380.95MHz

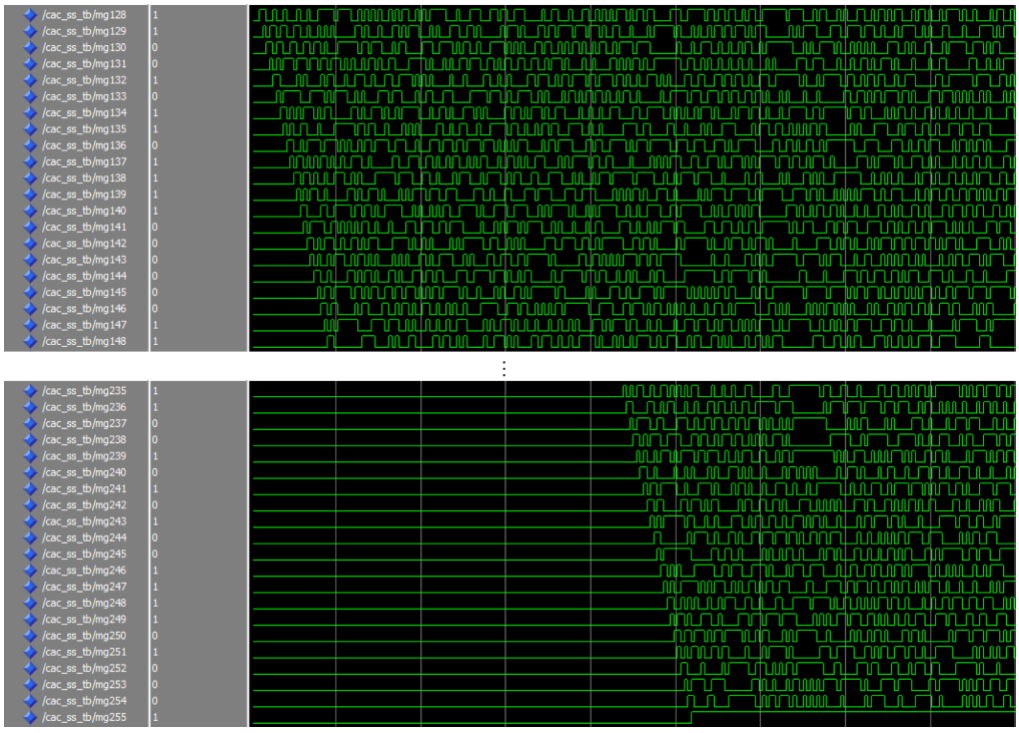


Figura 3.22 Resultados de simulación del generador de la matriz de muestreo **Φ** para CR_3.

a) Diseño.

La Figura 3.23 muestra el diagrama RTL que se obtiene del diseño del combinador y se observa la implementación de la multiplicación a través de la compuerta XOR. La información de síntesis de la Tabla 3.12 indica que el combinador utiliza 8 elementos lógicos.

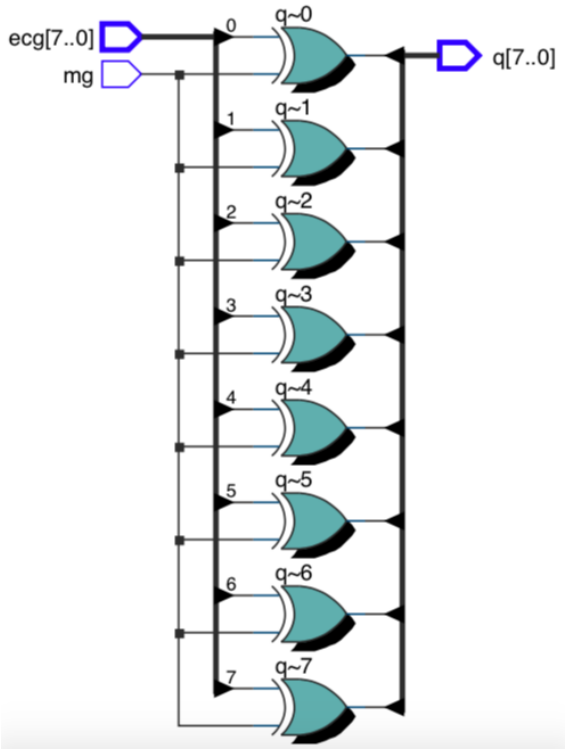


Figura 3.23 Diagrama RTL del combinador.

Tabla 3.12 Reporte de síntesis del combinador.

Fitter Status	Successful - Thu Jun 07 00:09:15 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	Comb
Top-level Entity Name	Comb
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	8 / 114,480 (< 1 %)
Total combinational functions	8 / 114,480 (< 1 %)
Dedicated logic registers	0 / 114,480 (0 %)
Total registers	0
Total pins	17 / 529 (3 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

b) Simulación.

Los vectores de entrada ECG y MG elegidos con el propósito de simular y probar el correcto funcionamiento del combinador diseñado son:

$$ECG = 01010101,$$
$$MG = 0,1.$$

La Tabla 3.13 indica los valores esperados Q_0 y los valores esperados Q_1. La Figura 3.24 muestra los resultados obtenidos en la simulación para el combinador diseñado.

3.3.3 Acumulador

La salida del XOR del combinador se conecta a un acumulador y el vector **y** se obtiene cada N muestras. El módulo consta de:

- Una señal de entrada de 1 bit que corresponde al coeficiente de la matriz de muestreo **Φ**.
- Una señal de entrada de 8 bits que corresponde a la salida del combinador.
- Una señal de salida de 10 bits que corresponde al vector **y**.
- Un sumador con acarreo de entrada
- Un registro de 10 bits.

a) Diseño.

La Figura 3.25 muestra el diagrama RTL que se obtiene del diseño del acumulador, se observa que el coeficiente de la matriz de muestreo **Φ** se conecta al acarreo de entrada del sumador y la conexión del sumador y el registro. La información de síntesis de la Tabla 3.14 indica que el acumulador utiliza 23 elementos lógicos, 10 registros y la frecuencia máxima de operación es de 277.7MHz.

b) Simulación.

Los vectores de entrada MG y ECG elegidos con el propósito de simular y probar el correcto funcionamiento del acumulador corresponden a los coeficientes generados con el módulo 3.3.1 y un *frame* de 512 muestras de la señal ECG vía simulador [12] respectivamente. La Figura 3.26 muestra los resultados obtenidos en la simulación para el acumulador diseñado.

3.4 Ventajas de CAC-SS sobre CAC-OST

Las ventajas de CAC-SS sobre CAC-OST son:

- CAC-SS permite recibir señales que fluyen tipo *streaming*.
- CAC-SS no requiere que la señal sea finita.
- CAC-SS no requiere *hardware* extra para calcular el límite o umbral y la semilla inicial.
- CAC-SS no requiere bloques de memoria para almacenar la señal.
- CAC-SS no requiere leer la señal entera para generar la matriz de muestreo **Φ**.

Con base en las ventajas indicadas, CAC-SS:

- Requiere menor *hardware* que CAC-OST.
- Es más rápido que CAC-OST.

Tabla 3.13 Valores esperados y obtenidos del combinador.

ECG	01010101	
MG	0	1
Q_0	01010101	10101010
Q_1	01010101	10101010

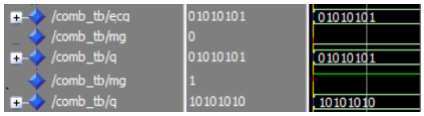


Figura 3.24 Resultados de simulación del combinador.

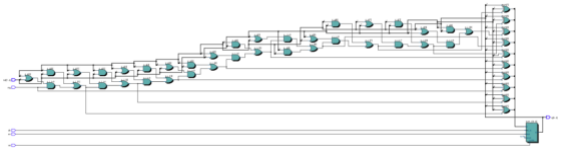


Figura 3.25 Diagrama RTL del acumulador.

Tabla 3.14 Reporte de síntesis del acumulador.

Fitter Status	Successful - Thu Jun 07 18:21:18 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	acum
Top-level Entity Name	acum
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	23 / 114,480 (< 1 %)
Total combinational functions	23 / 114,480 (< 1 %)
Dedicated logic registers	10 / 114,480 (< 1 %)
Total registers	10
Total pins	22 / 529 (4 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	277.7MHz

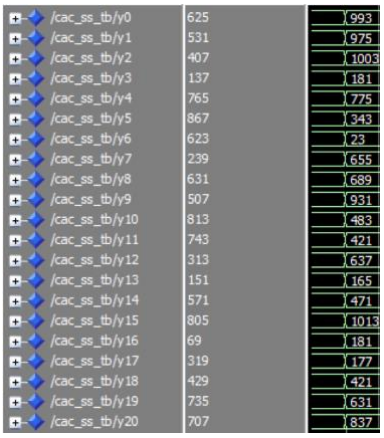


Figura 3.26 Resultados de simulación del acumulador.

4 Pruebas y resultados

Este capítulo presenta las pruebas y resultados que arrojan las arquitecturas digitales CAC-OST y CAC-SS implementadas en un FPGA. Los valores utilizados en las pruebas son los mismos que los utilizados en el marco de referencia. La sección 4.1 presenta las pruebas realizadas a la arquitectura digital CAC-OST. La sección 4.2 presenta las pruebas realizadas a la arquitectura digital CAC-SS. La sección 4.3 compara los resultados de las arquitecturas digital CAC-OST y CAC-SS *vs* los resultados del marco de referencia. El capítulo finaliza con la sección 4.4 en donde se demuestra la universalidad de las arquitecturas digitales CAC-OST y CAC-SS realizando pruebas con distintas señales de entrada.

4.1 Arquitectura digital CAC-OST

La arquitectura digital CAC-OST fue reproducida en un kit de desarrollo DE2-115. El kit incluye un chip FPGA *Cyclone* IV EP4CE115F29C7. La herramienta de síntesis fue Quartus Prime 17.0 de Altera® y VHDL como lenguaje de descripicón de *hardware*. La herramienta de simulación fue ModelSim de Mentor Graphics®.

Los tres escenarios en los que se realizó este método son:

1. Arquitectura digital CAC-OST para CR_1: Comprime la señal en ocho. La Figura 4.1 muestra la interconexión de los cinco módulos. La información de síntesis de la Tabla 4.1 indica que la arquitectura digital CAC-OST para CR_1 utiliza 1,965 elementos lógicos, 983 registros, 4,096 bits de memoria y la frecuencia máxima de operación es de 35.38MHz. La Figura 4.2 muestra la señal ECG cuantificada que se va a comprimir. La Figura 4.3 muestra los resultados de la compresión. Los resultados en la recuperación de la señal son:

   a) La Figura 4.4 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.129172$.
   b) La Figura 4.5 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.0955934$.

2. Arquitectura digital CAC-OST para CR_2: Comprime la señal en cuatro. La Figura 4.6 muestra la interconexión de los cinco módulos. La información de síntesis de la Tabla 4.2 indica que la arquitectura digital CAC-OST para CR_2 utiliza 4,148 elementos lógicos, 1,956 registros, 4,096 bits de memoria y la frecuencia máxima de operación es de 20.54MHz. La Figura 4.7 muestra los resultados de la compresión. Los resultados en la recuperación de la señal son:

   a) La Figura 4.8 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.0475137$.

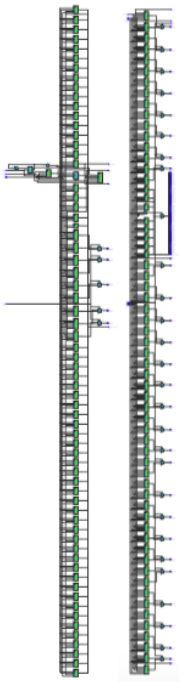


Figura 4.1 Diagrama RTL de la arquitectura digital CAC-OST para CR_1.

Tabla 4.1 Reporte de síntesis de la arquitectura digital CAC-OST para CR_1.

Fitter Status	Successful - Tue May 29 20:28:26 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	cac_ost
Top-level Entity Name	cac_ost
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	1,965 / 114,480 (2 %)
Total combinational functions	1,943 / 114,480 (2 %)
Dedicated logic registers	983 / 114,480 (< 1 %)
Total registers	983
Total pins	500 / 529 (95 %)
Total virtual pins	0
Total memory bits	4,096 / 3,981,312 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	35.38MHz

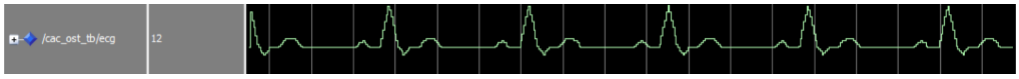


Figura 4.2 Señal ECG cuantificada.

b) La Figura 4.9 muestra los resultados con $\boldsymbol{\Psi}_2$ y se observa el $MSE = 0.0468365$.

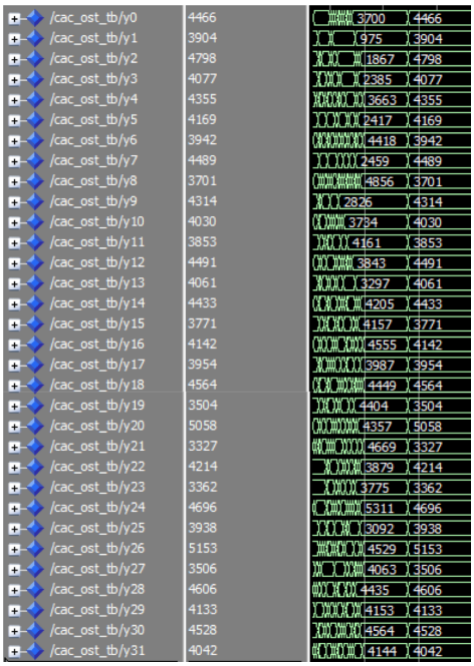


Figura 4.3 Resultados de compresión cuando $\mathbf{\Phi}_3$ es CAC-OST y CR_1.

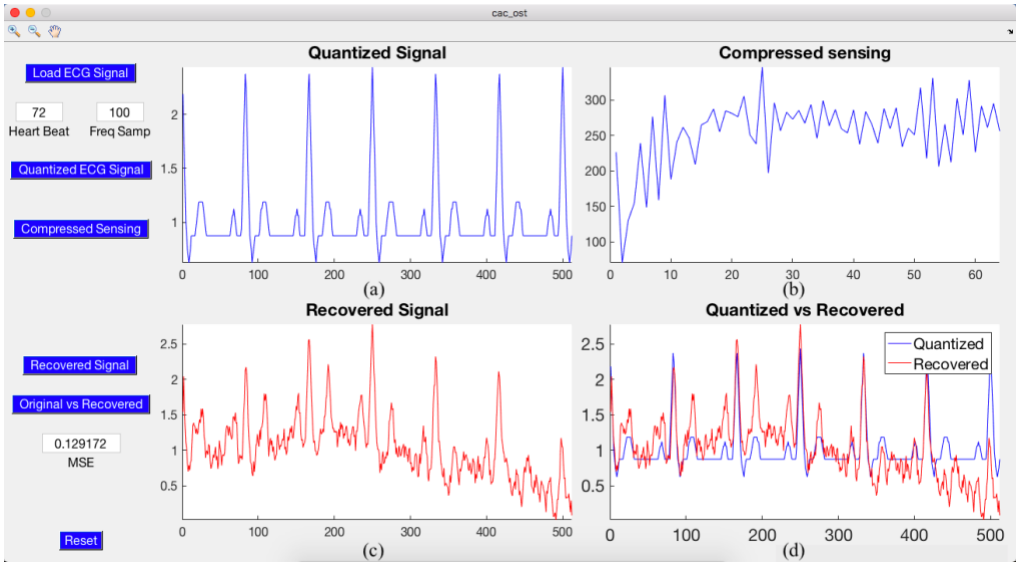


Figura 4.4 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

3. Arquitectura digital CAC-OST para CR_3: Comprime la señal en dos. La Figura 4.10 muestra la interconexión de los cinco módulos. La información de síntesis de la Tabla 4.3 indica que la arquitectura digital CAC-OST para CR_3 utiliza 8,174 elementos lógicos, 3,850 registros, 4,096 bits de memoria y la frecuencia máxima de operación es de 11.15MHz. La Figura 4.11 muestra los resultados de la compresión. Los resultados en la recuperación de la señal son:

   a) La Figura 4.12 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.0093602$.
   b) La Figura 4.13 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.0216762$.

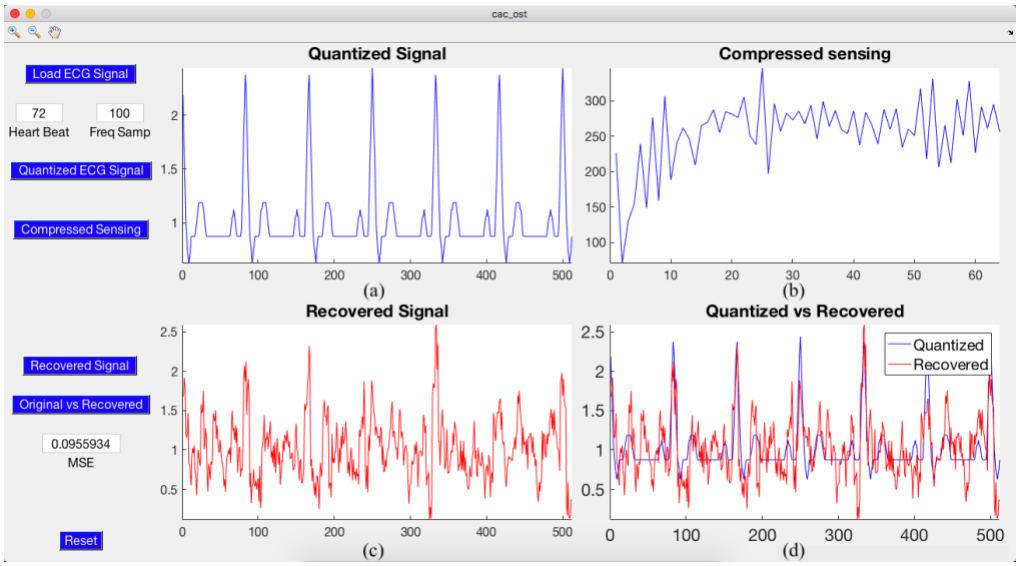


Figura 4.5 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

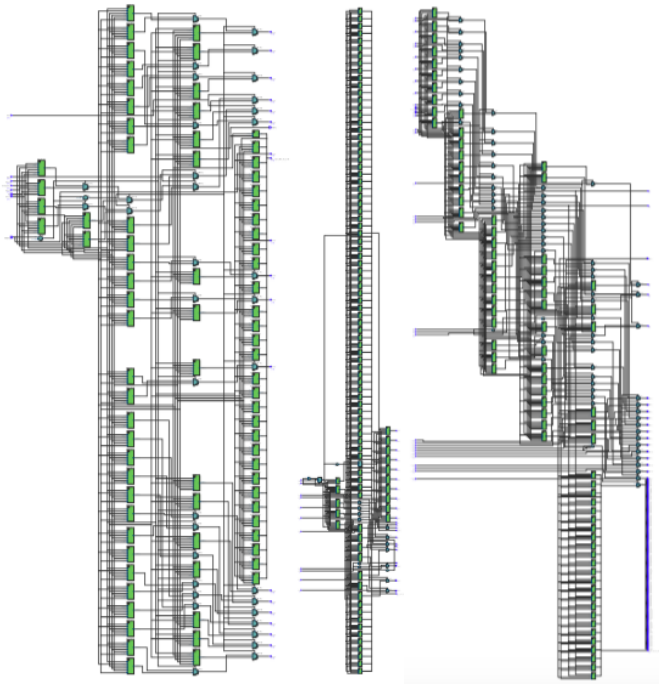


Figura 4.6 Diagrama RTL de la arquitectura digital CAC-OST para CR_2.

4.2 Arquitectura digital CAC-SS

La arquitectura digital CAC-SS fue reproducida en un kit de desarrollo DE2-115. El kit incluye un chip FPGA *Cyclone* IV EP4CE115F29C7. La herramienta de síntesis fue Quartus Prime 17.0 de Altera® y VHDL como lenguaje de descripción de *hardware*. La herramienta de simulación fue ModelSim de Mentor Graphics®.

Tabla 4.2 Reporte de síntesis de la arquitectura digital CAC-OST para CR_2.

Fitter Status	Successful - Tue May 29 23:37:57 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	cac_ost
Top-level Entity Name	cac_ost
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	4,148 / 114,480 (4 %)
Total combinational functions	4,086 / 114,480 (4 %)
Dedicated logic registers	1,956 / 114,480 (2 %)
Total registers	1956
Total pins	358 / 529 (68 %)
Total virtual pins	0
Total memory bits	4,096 / 3,981,312 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	20.54MHz

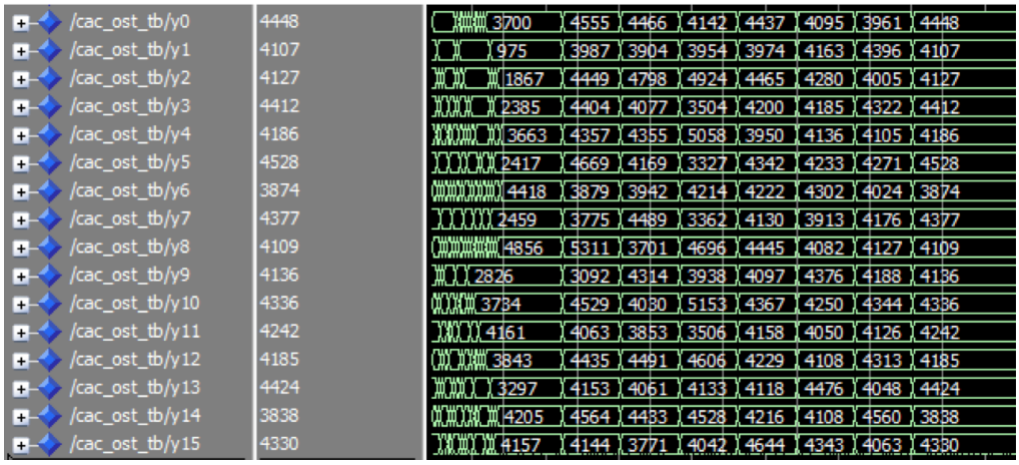


Figura 4.7 Resultados de compresión con $\mathbf{\Phi}_3$ y CR_2.

Los tres escenarios en los que se realizó este método son:

1. Arquitectura digital CAC-SS para CR_1: Comprime la señal en ocho. La Figura 4.14 muestra la interconexión de los cinco módulos. La información de síntesis de la Tabla 4.4 indica que la arquitectura digital CAC-SS para CR_1 utiliza 1,682 elementos lógicos, 970 registros y la frecuencia máxima de operación es de 171.06MHz. La Figura 4.15 muestra la señal ECG cuantificada que se va a comprimir. La Figura 4.16 muestra los resultados de la compresión. Los resultados en la recuperación de la señal son:

   a) La Figura 4.17 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.20799200$.
   b) La Figura 4.18 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.13206900$.

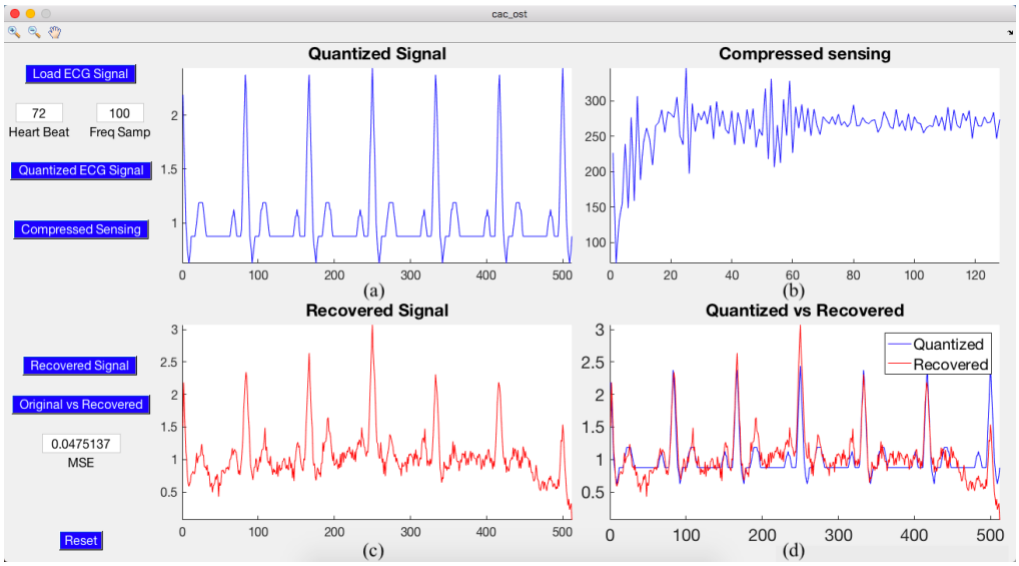


Figura 4.8 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

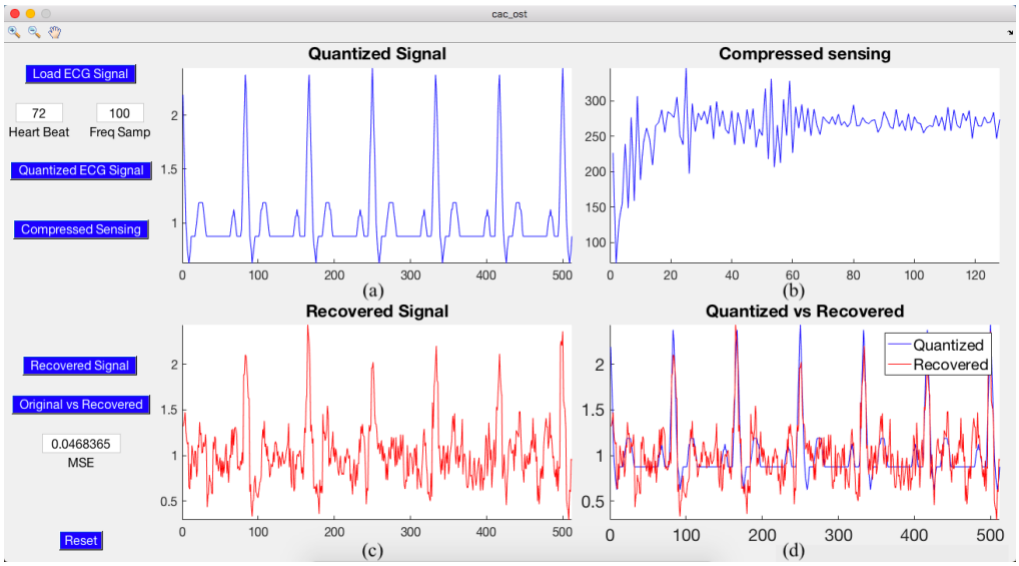


Figura 4.9 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

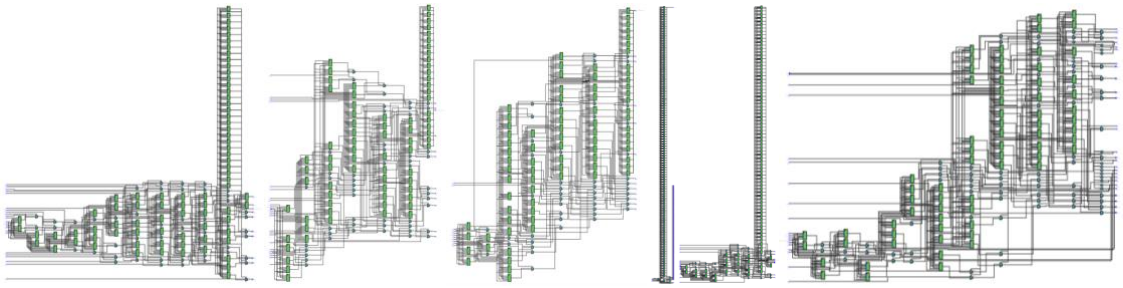


Figura 4.10 Diagrama RTL de la arquitectura digital CAC-OST para CR_3.

2. Arquitectura digital CAC-SS para CR_2: Comprime la señal en cuatro. La Figura 4.19

Tabla 4.3 Reporte de síntesis de la arquitectura digital CAC-OST para CR_3.

Fitter Status	Successful - Wed May 30 11:57:30 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	cac_ost
Top-level Entity Name	cac_ost
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	8,174 / 114,480 (7 %)
Total combinational functions	8,064 / 114,480 (7 %)
Dedicated logic registers	3,850 / 114,480 (3 %)
Total registers	3850
Total pins	487 / 529 (92 %)
Total virtual pins	0
Total memory bits	4,096 / 3,981,312 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	11.15MHz

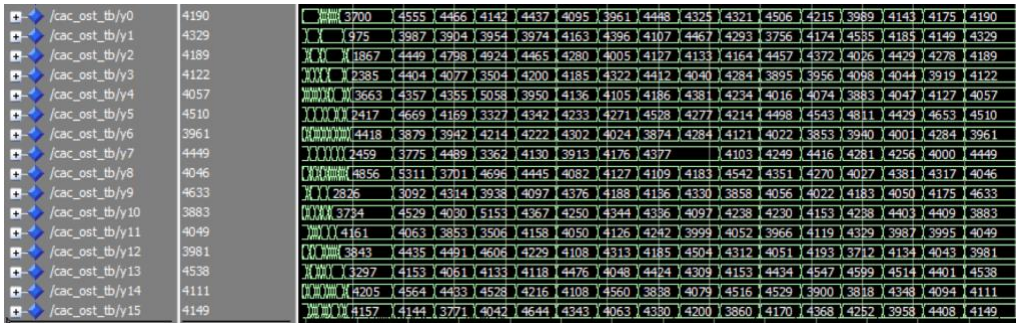


Figura 4.11 Resultados de compresión con $\mathbf{\Phi}_3$ y CR_3.

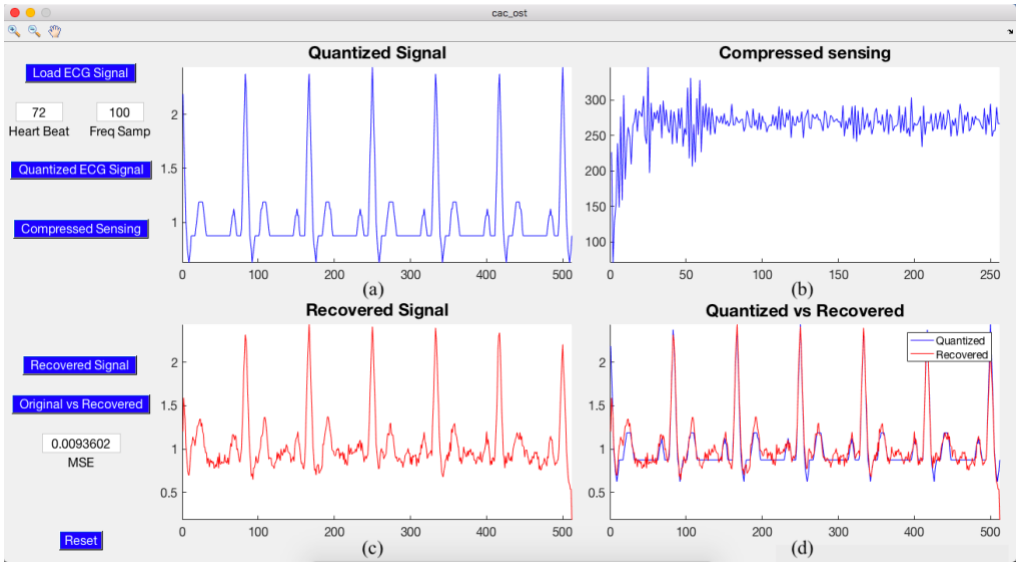


Figura 4.12 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

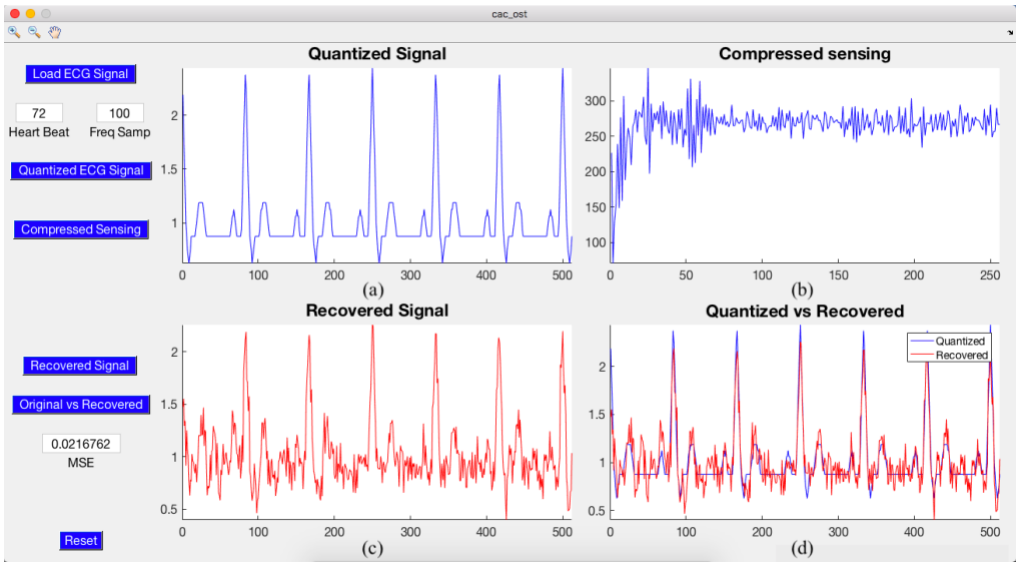


Figura 4.13 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

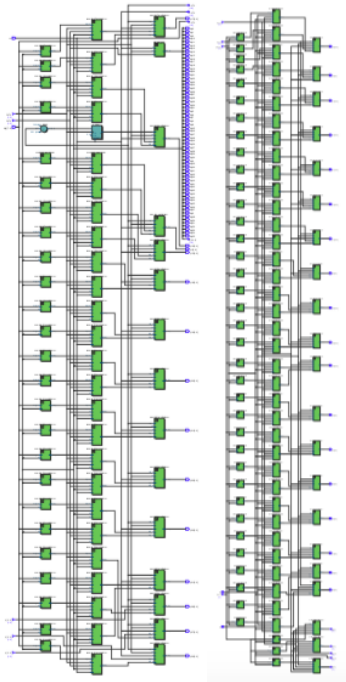


Figura 4.14 Diagrama RTL de la arquitectura digital CAC-SS para CR_1.

muestra la interconexión de los cinco módulos. La información de síntesis de la Tabla 4.5 indica que la arquitectura digital CAC-SS para CR_2 utiliza 3,324 elementos lógicos, 1,610 registros y la frecuencia máxima de operación es de 197.12MHz. La Figura 4.20 muestra los resultados de la compresión. Los resultados en la recuperación de la señal son:

a) La Figura 4.21 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.07669780$.
b) La Figura 4.22 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.07739020$.

Tabla 4.4 Reporte de síntesis de la arquitectura digital CAC-SS para CR_1.

Fitter Status	Successful - Tue Jun 12 17:58:24 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	cac_ss
Top-level Entity Name	cac_ss
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	1,682 / 114,480 (1 %)
Total combinational functions	1,648 / 114,480 (1 %)
Dedicated logic registers	970 / 114,480 (< 1 %)
Total registers	970
Total pins	395 / 529 (75 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	171.06MHz

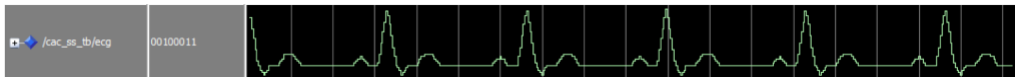


Figura 4.15 Señal ECG cuantificada.

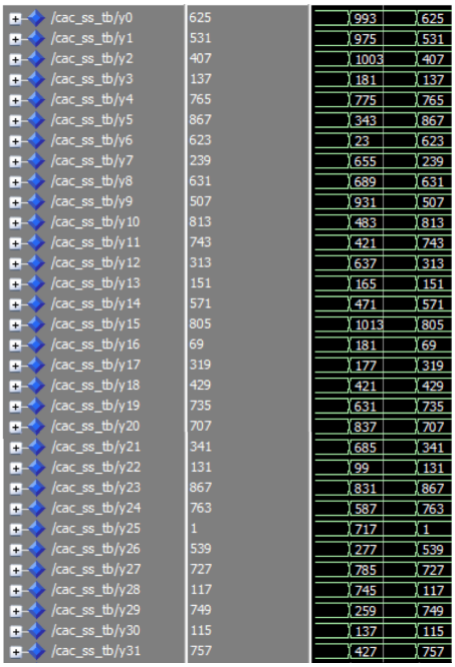


Figura 4.16 Resultados de compresión cuando $\mathbf{\Phi}_4$ es CAC-SS y CR_1.

3. Arquitectura digital CAC-SS para CR_3: Comprime la señal en dos. La Figura 4.23 muestra la interconexión de los cinco módulos. La información de síntesis de la Tabla 4.6 indica que la arquitectura digital CAC-SS para CR_3 utiliza 6,899 elementos lógicos, 2,730

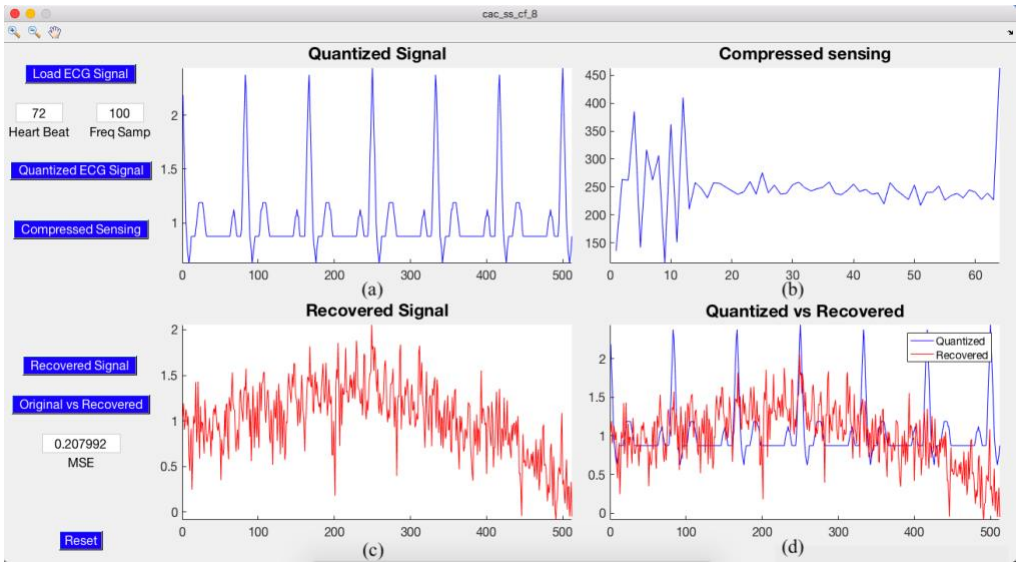


Figura 4.17 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

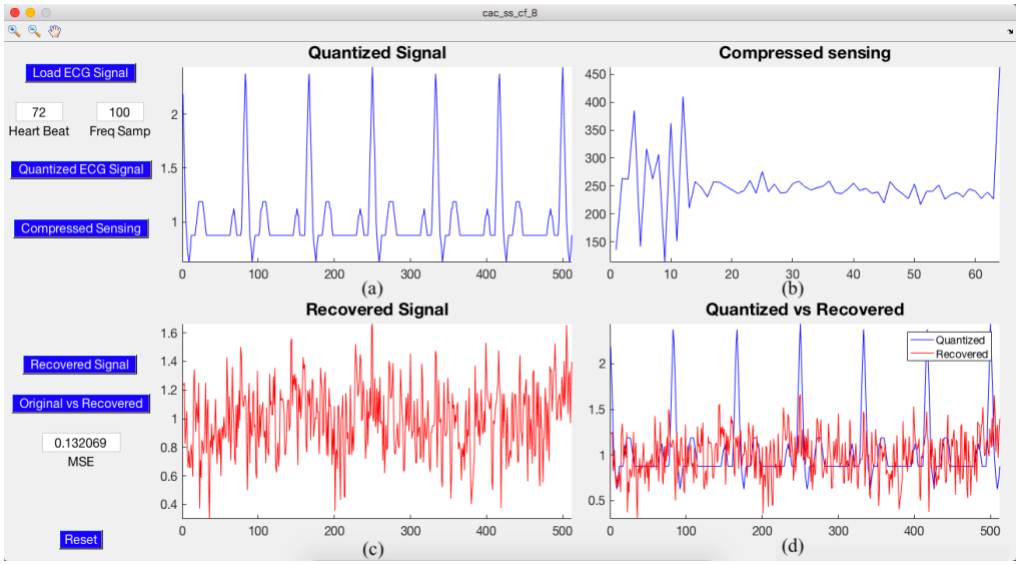


Figura 4.18 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

registros y la frecuencia máxima de operación es de 184.4MHz. La Figura 4.24 muestra los resultados de la compresión. Los resultados en la recuperación de la señal son:

a) La Figura 4.25 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.00575565$.
b) La Figura 4.26 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.02539030$.

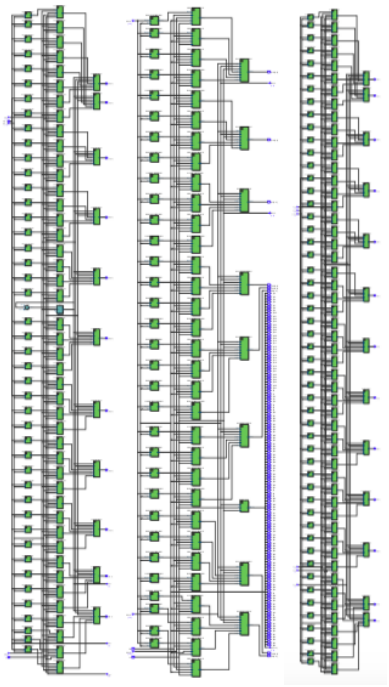


Figura 4.19 Diagrama RTL de la arquitectura digital CAC-SS para CR_2.

Tabla 4.5 Reporte de síntesis de la arquitectura digital CAC-SS para CR_2.

Fitter Status	Successful - Wed Jun 13 22:05:48 2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	cac_ss
Top-level Entity Name	cac_ss
Family	Cyclone IV E
Device	EP4CE30F29C6
Timing Models	Final
Total logic elements	3,324 / 28,848 (12 %)
Total combinational functions	3,282 / 28,848 (11 %)
Dedicated logic registers	1,610 / 28,848 (6 %)
Total registers	1610
Total pins	459 / 533 (86 %)
Total virtual pins	0
Total memory bits	0 / 608,256 (0 %)
Embedded Multiplier 9-bit elements	0 / 132 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	197.12MHz

4.3 Resultados

4.3.1 Marco de referencia

La Figura 4.27, Figura 4.28 y Figura 4.29 comparan las gráficas de la señal ECG original [12] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos respectivamente en los seis

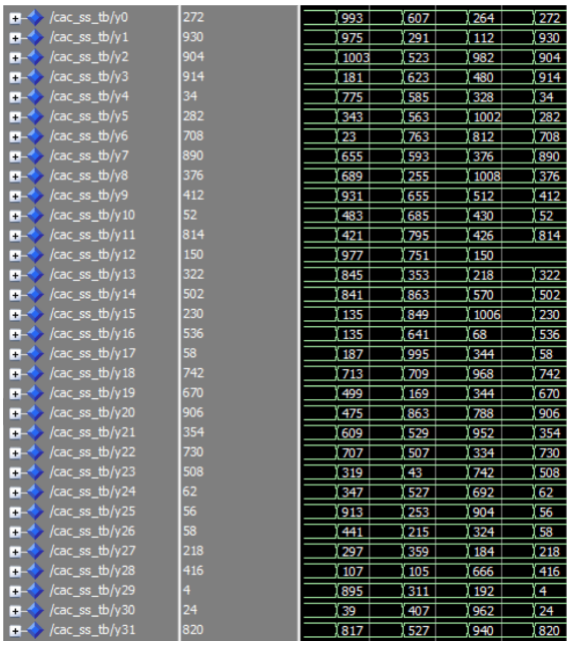


Figura 4.20 Resultados de compresión con $\mathbf{\Phi}_4$ y CR_2.

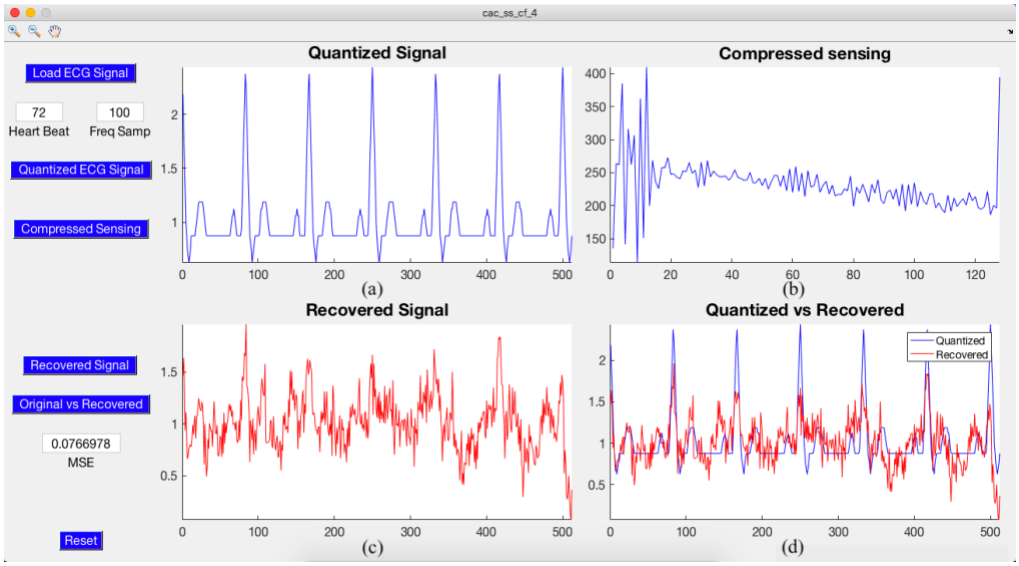


Figura 4.21 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

escenarios en los que se aplicó muestreo compreso por *software* (2.6.1). La dispersión de la señal ECG en DCT recupera con mayor precisión en comparación con WHT para cada razón de compresión. La Tabla 4.7 compara la tasa de error y se observa que la menor tasa de error se presenta cuando la razón de compresión es de dos y la dispersión de la señal ECG es DCT.

La Tabla 4.8 compara el área y velocidad en los tres escenarios en los que se aplicó muestreo compreso por arquitectura digital [13] (2.6.2). El resultado con menor área es cuando la razón de compresión es de ocho. La Figura 4.30, Figura 4.31 y Figura 4.32 comparan las gráficas de la señal ECG original [12] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos

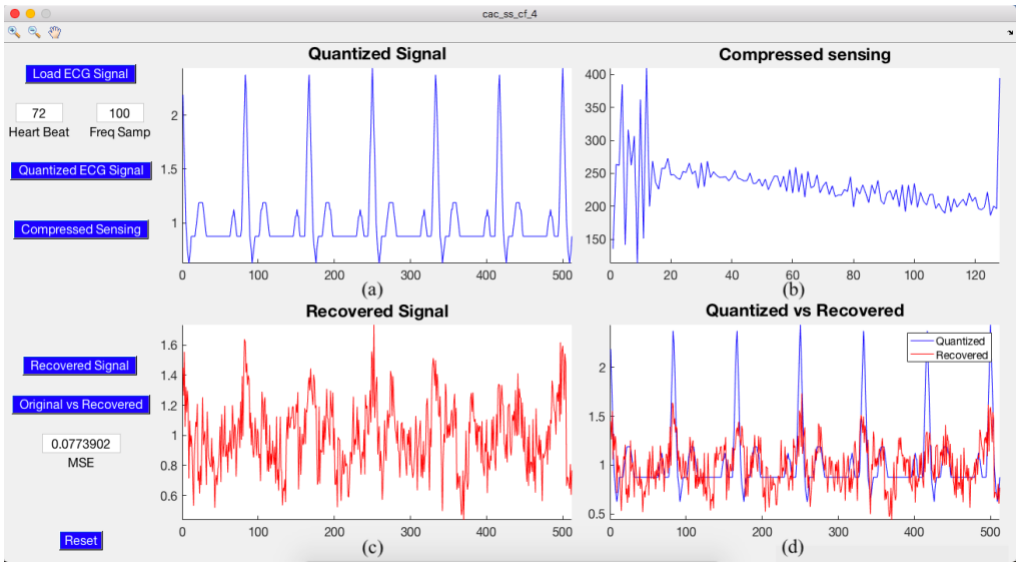


Figura 4.22 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

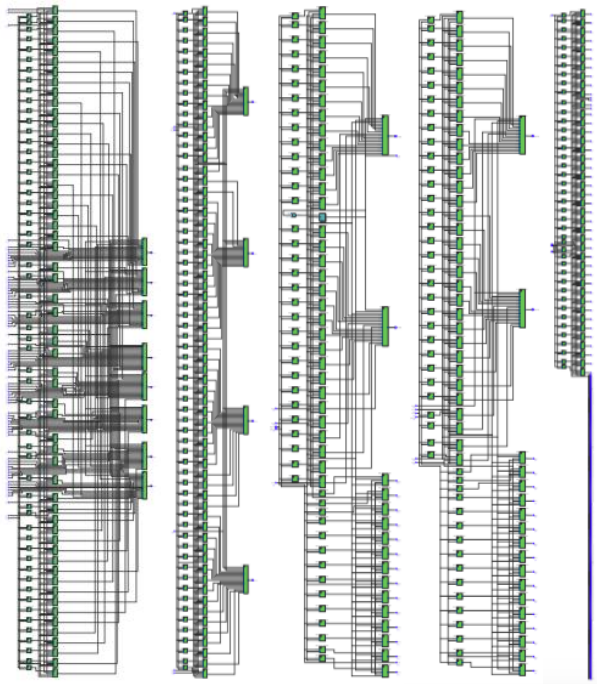


Figura 4.23 Diagrama RTL de la arquitectura digital CAC-SS para CR_3.

respectivamente. La Tabla 4.9 compara la tasa de error y se observa que la menor tasa de error se presenta cuando la razón de compresión es de dos y la dispersión de la señal ECG es DCT.

4.3.2 Arquitecturas digitales CAC-OST y CAC-SS

La Tabla 4.10 compara el área y velocidad en los tres escenarios en los que se aplicó muestreo compreso por arquitectura digital CAC-OST (4.1). El resultado con menor área y mayor velocidad

Tabla 4.6 Reporte de síntesis de la arquitectura digital CAC-SS para CR_3.

Fitter Status	Successful - Thu Jun 14 00:13:16:2018
Quartus Prime Version	17.0.0 Build 595 04/25/2017 SJ Lite Edition
Revision Name	cac_ss
Top-level Entity Name	cac_ss
Family	Cyclone IV E
Device	EP4CE30F29C6
Timing Models	Final
Total logic elements	6,899 / 28,848 (24 %)
Total combinational functions	6,833 / 28,848 (24 %)
Dedicated logic registers	2,730 / 28,848 (9 %)
Total registers	2730
Total pins	427 / 533 (80 %)
Total virtual pins	0
Total memory bits	0 / 608,256 (0 %)
Embedded Multiplier 9-bit elements	0 / 132 (0 %)
Total PLLs	0 / 4 (0 %)
Fmax	184.4MHz

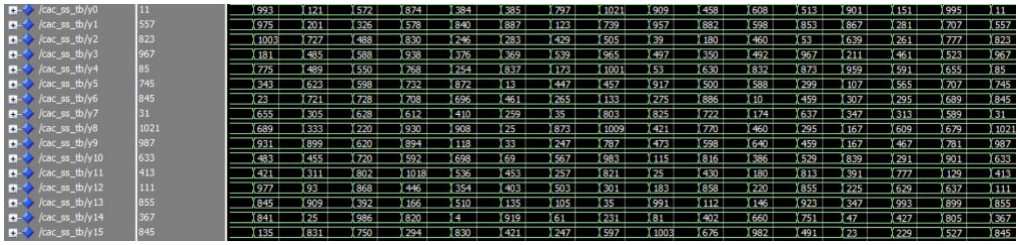


Figura 4.24 Resultados de compresión con $\mathbf{\Phi}_4$ y CR_3.

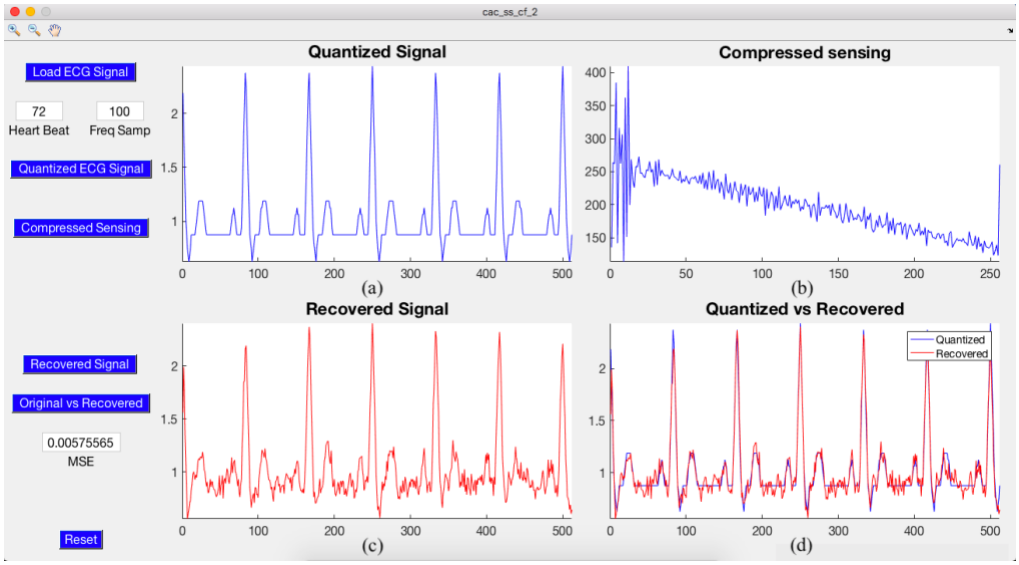


Figura 4.25 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

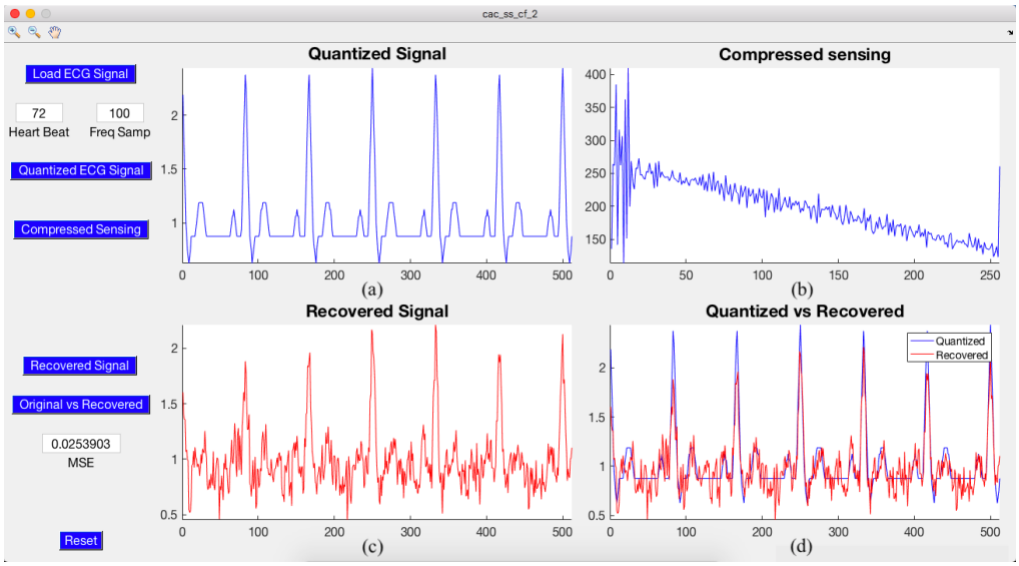


Figura 4.26 (a) Señal ECG original [12], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

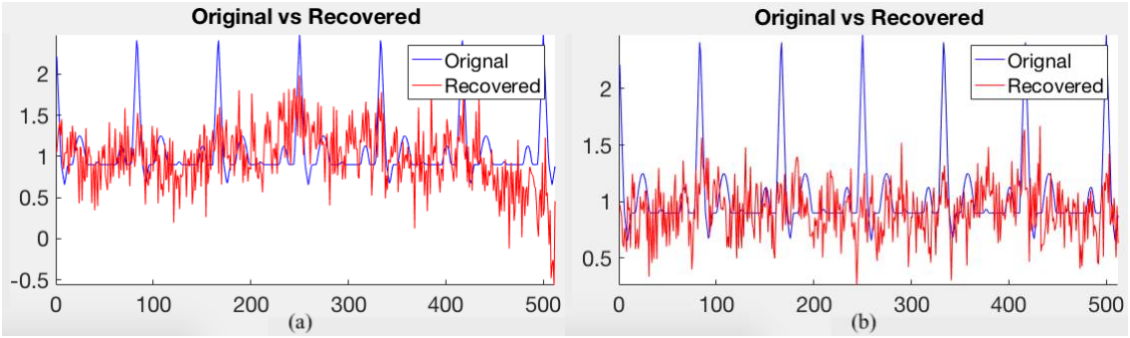


Figura 4.27 Señal ECG original [12] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_2$.

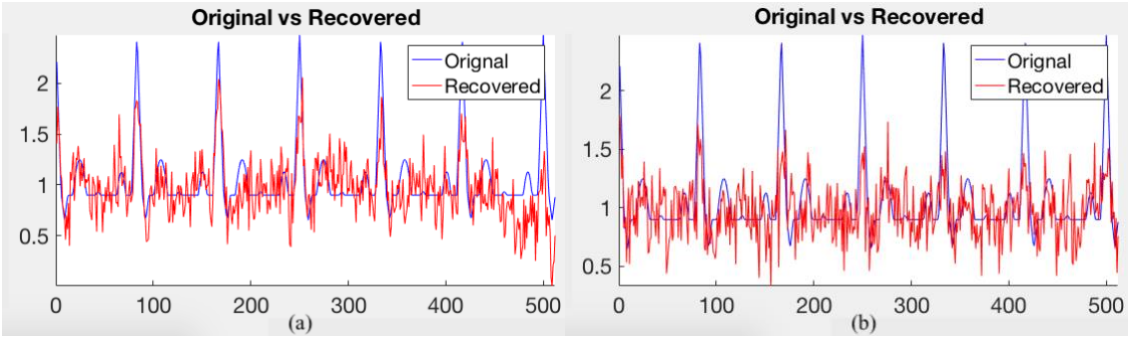


Figura 4.28 Señal ECG original [12] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_2$.

es cuando la razón de compresión es de ocho. La Figura 4.33, Figura 4.34 y Figura 4.35 comparan las gráficas de la señal ECG original [12] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos respectivamente. La Tabla 4.11 compara la tasa de error y se observa que la menor tasa de error se presenta cuando la razón de compresión es de dos y la dispersión de la señal ECG es DCT.

La Tabla 4.12 compara el área y velocidad en los tres escenarios en los que se aplicó muestreo

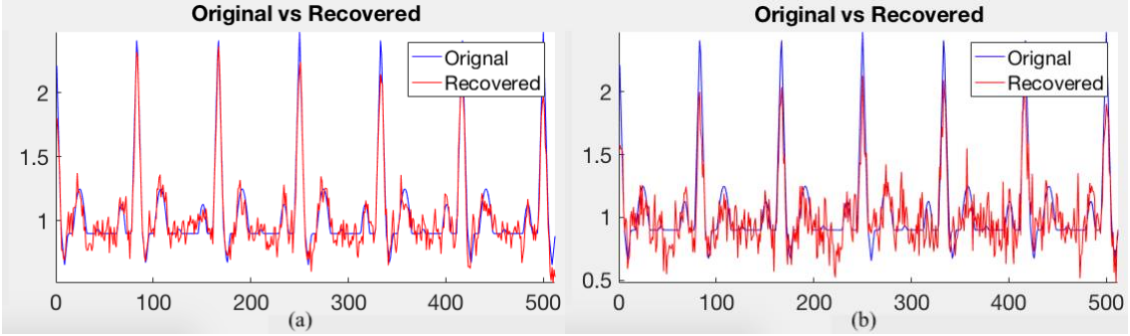


Figura 4.29 Señal ECG original [12] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_2$.

Tabla 4.7 Comparación de tasa de error en muestreo compreso por *software*.

f	Señal ECG simulada [12]					
N	512					
M	64		128		256	
CR	8		4		2	
Φ	Gaussian					
Ψ	DCT	WHT	DCT	WHT	DCT	WHT
Recuperación	Optimización l_1 [23]					
MSE	0.19539000	0.12744800	0.07467780	0.09413290	0.01529000	0.03111390

Tabla 4.8 Comparación de área y velocidad de arquitectura digital [13].

Compression ratio	8	4	2
Logical Elements	1,750	3,447	7,157
Registers	974	1,615	2,762
Fmax (MHz)	166.5	167.2	145.5

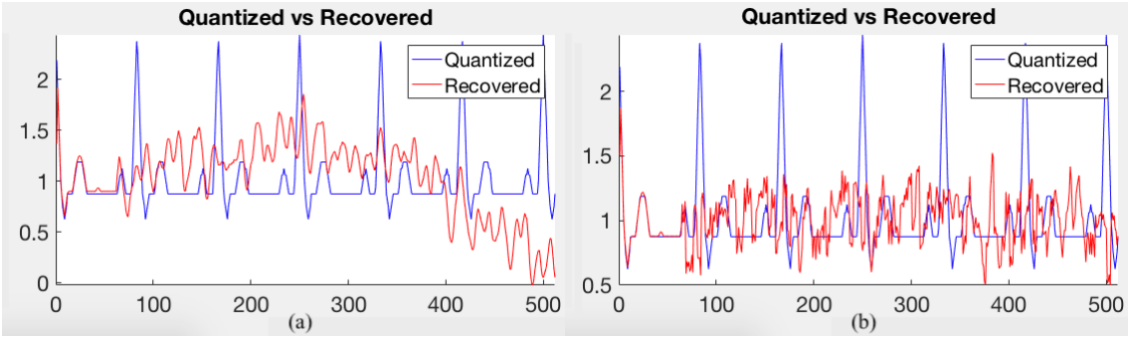


Figura 4.30 Señal ECG original [12] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_2$.

compreso por arquitectura digital CAC-SS (4.2). El resultado con menor área es cuando la razón de compresión es de ocho. La Figura 4.36, Figura 4.37 y Figura 4.38 comparan las gráficas de la señal ECG original [12] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos respectivamente. La Tabla 4.13 compara la tasa de error y se observa que la menor tasa de error se presenta cuando la razón de compresión es de dos y la dispersión de la señal ECG es DCT.

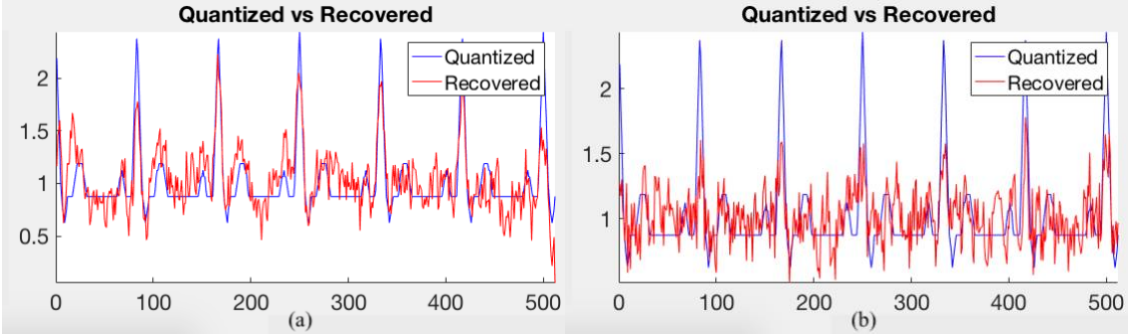


Figura 4.31 Señal ECG original [12] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_2$.

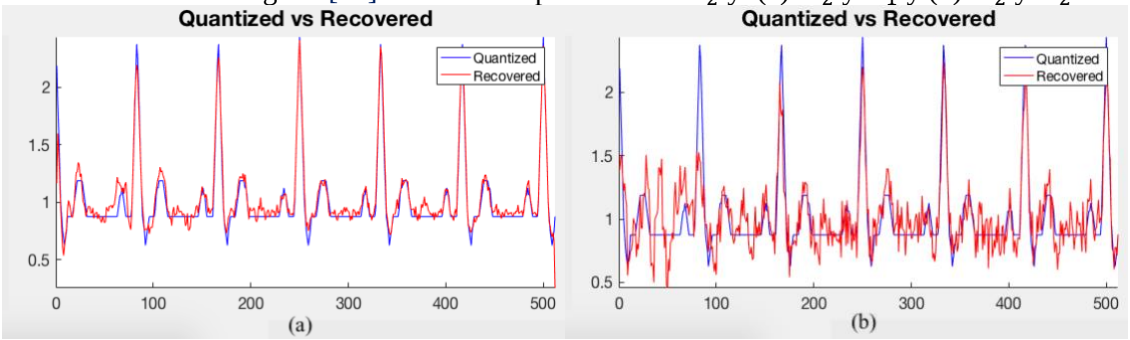


Figura 4.32 Señal ECG original [12] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_2$.

Tabla 4.9 Comparación de tasa de error de arquitectura digital [13].

f	Señal ECG simulada [12]					
N	512					
M	64		128		256	
CR	8		4		2	
Φ	Bernoulli					
Ψ	DCT	WHT	DCT	WHT	DCT	WHT
Recuperación	Optimización l_1 [23]					
MSE	0.22561500	0.14120200	0.05316020	0.07565700	0.00743218	0.03263760

Tabla 4.10 Comparación de área y velocidad de arquitectura digital CAC-OST.

Compression factor	8	4	2
Logical elements	1,965	4,148	8,174
Registers	983	1,956	3,850
Memory bits	4,096	4,096	4,096
Fmax (MHz)	35.38	20.54	11.15

4.3.3 Arquitecturas digitales CAC-OST y CAC-SS *vs* marco de referencia

Los resultados de las arquitecturas digitales CAC-OST y CAC-SS se contrastan con los resultados del marco de referencia. La comparación se realiza en tres ejes. El primer eje compara el área y velocidad de las arquitecturas (se contrastan la cantidad de elementos lógicos, registros, bloques de memoria y frecuencia máxima de operación). El segundo eje compara las gráficas de

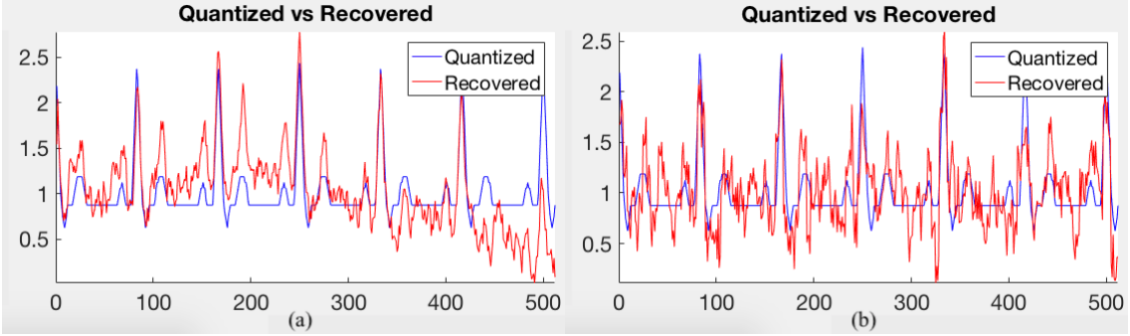


Figura 4.33 Señal ECG original [12] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$.

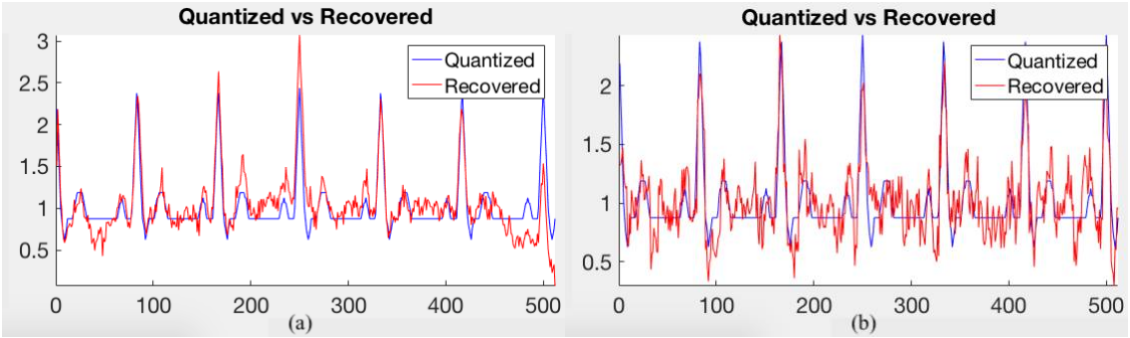


Figura 4.34 Señal ECG original [12] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$.

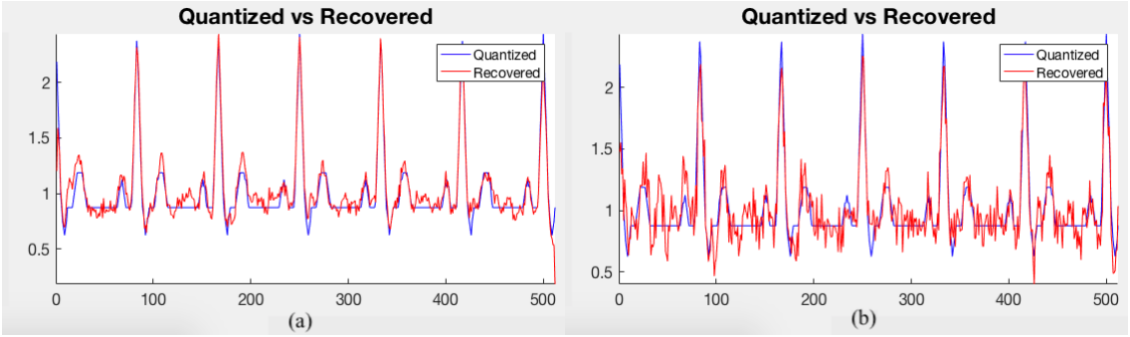


Figura 4.35 Señal ECG original [12] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$.

Tabla 4.11 Comparación de tasa de error de arquitectura digital CAC-OST.

f	Señal ECG simulada [12]					
N	512					
M	64		128		256	
CR	8		4		2	
Φ	CAC – OST					
Ψ	DCT	WHT	DCT	WHT	DCT	WHT
Recuperación	Optimización l_1 [23]					
MSE	0.12917200	0.09559340	0.04751370	0.04683650	0.00936020	0.02167620

la señal ECG original [12] *vs* señal recuperada. El tercer eje compara la tasa de error en la recuperación (se contrasta el MSE resultante).

Tabla 4.12 Comparación de área y velocidad de arquitectura digital CAC-SS.

Compression factor	8	4	2
Logical elements	1,682	3,324	6,899
Registers	970	1,610	2,730
Fmax (MHz)	171.06	197.12	184.4

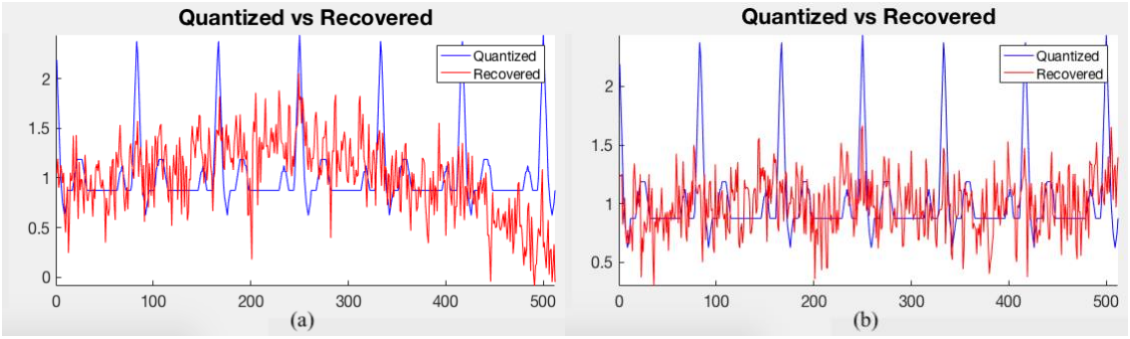


Figura 4.36 Señal ECG original [12] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

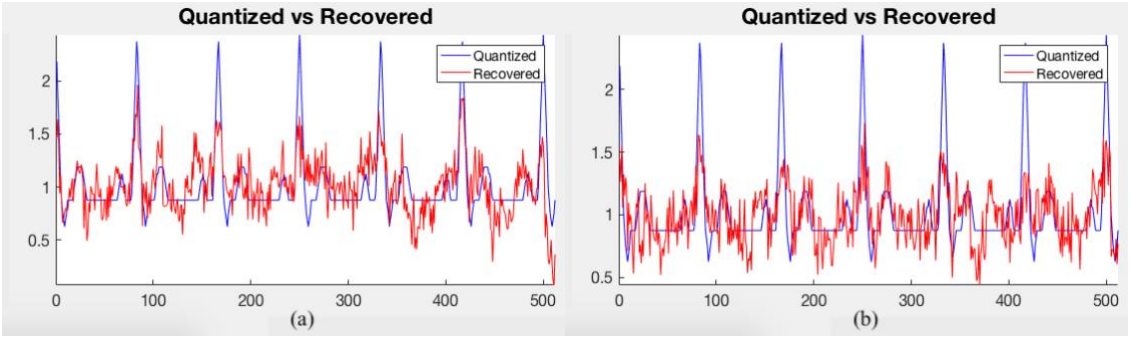


Figura 4.37 Señal ECG original [12] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

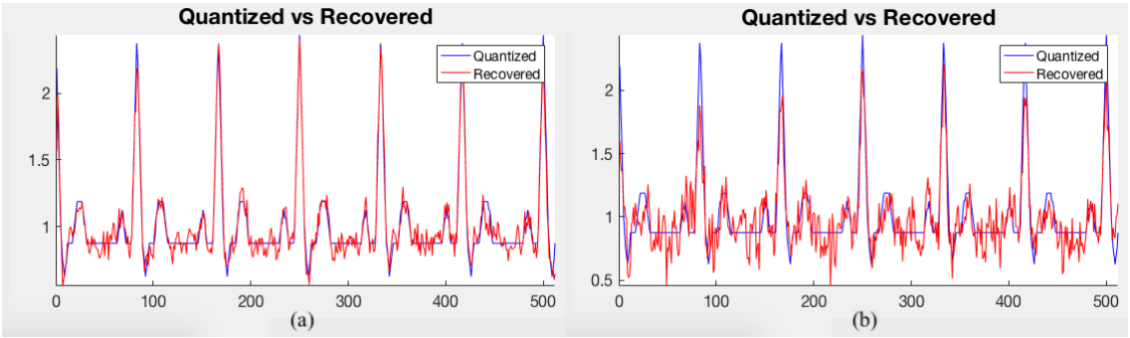


Figura 4.38 Señal ECG original [12] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

1. Comparación de área y velocidad: La arquitectura digital CAC-SS utiliza menor área (menos cantidad de elementos lógicos, registros y no utiliza bloques de memoria) y es más rápida (la frecuencia máxima de operación es mayor), en comparación con las arquitecturas digitales CAC-OST y [13] para cada razón de compresión. La Tabla 4.14 compara los resultados de área y velocidad de las arquitecturas digitales CAC-SS, CAC-OST y [13].
2. Comparación de las gráficas de la señal ECG original [12] *vs* señal recuperada: La dispersión de la señal ECG en DCT recupera con mayor precisión en comparación con

Tabla 4.13 Comparación de tasa de error de arquitectura digital CAC-SS.

f	Señal ECG simulada [12]					
N	512					
M	64		128		256	
CR	8		4		2	
Φ	CAC-SS					
Ψ	DCT	WHT	DCT	WHT	DCT	WHT
Recuperación	Optimización l_1 [23]					
MSE	0.20799200	0.13206900	0.07669780	0.07739020	0.00575565	0.02539030

Tabla 4.14 Comparación de área y velocidad de las arquitecturas digitales CAC-SS, CAC-OST y [13].

	Arquitectura digital CAC-SS			Arquitectura digital CAC-OST			Arquitectura digital [13]		
CR	8	4	2	8	4	2	8	4	2
Logical elements	1,682	3,324	6,899	1,965	4,148	8,174	1,750	3,447	7,157
Registers	970	1,610	2,730	983	1,956	3,850	974	1,615	2,762
Memory bits	0			4,096			0		
Fmax (MHz)	171.06	197.12	184.4	35.38	20.54	11.15	166.5	167.2	145.5

WHT para cada matriz de muestreo **Φ** y cada razón de compresión. La Figura 4.39, Figura 4.40 y Figura 4.41 muestran las comparaciones entre la señal ECG original [12] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos respectivamente.

3. Comparación de la tasa de error en la recuperación: La arquitectura digital CAC-SS presenta la menor tasa de error cuando la razón de compresión es de dos y la dispersión de la señal ECG es DCT. La arquitectura digital CAC-OST presenta la menor tasa de error cuando la razón de compresión de cuatro y ocho y la dispersión de la señal ECG es WHT. La Tabla 4.15, Tabla 4.16 y Tabla 4.17 contrastan la tasa de error en la recuperación de la señal para razón de compresión de ocho, cuatro y dos respectivamente y considera la dispersión de la señal ECG en DCT y WHT.

4.4 Pruebas con distintas señales de entrada

Las arquitecturas digitales propuestas, CAC-OST y CAC-SS, se sometieron a pruebas con distintas señales de entrada. La intención de estas pruebas es demostrar que ambas arquitecturas son universales.

4.4.1 Señal ECG real

Las señales de entrada utilizadas en este escenario fueron tomadas de la base de datos de *PhysioBank* [24]. Las señales consideradas para estas pruebas son:

a) $\mathbf{f}_1$ es MIT-BIH *Arrythmia Database* record 100: Paciente masculino de 69 años con trastorno cardiovascular tipo supraventricular ectópico (latidos prematuros ventriculares aislados) y frecuencia cardiaca de 70 – 89 latidos por minuto.

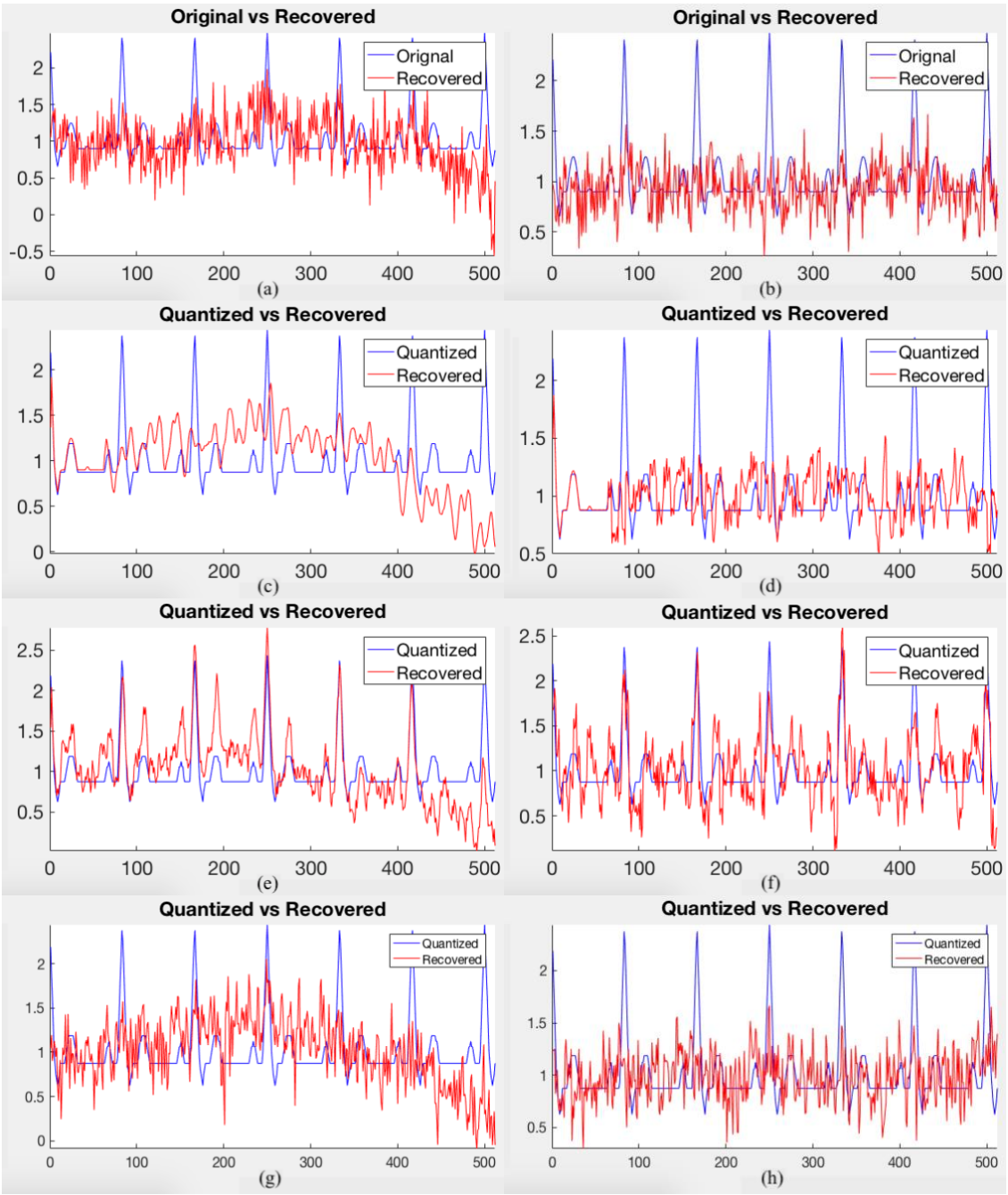


Figura 4.39 Señal ECG original [12] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_1$, (d) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_2$, (e) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (f) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (g) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (h) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

b) $\mathbf{f}_2$ es MIT-BIH *Arrythmia Database* record 102: Paciente femenina de 84 años con trastorno cardiovascular tipo ventricular ectópico (latidos prematuros ventriculares aislados) y frecuencia cardiaca de 72 – 78 latidos por minuto.

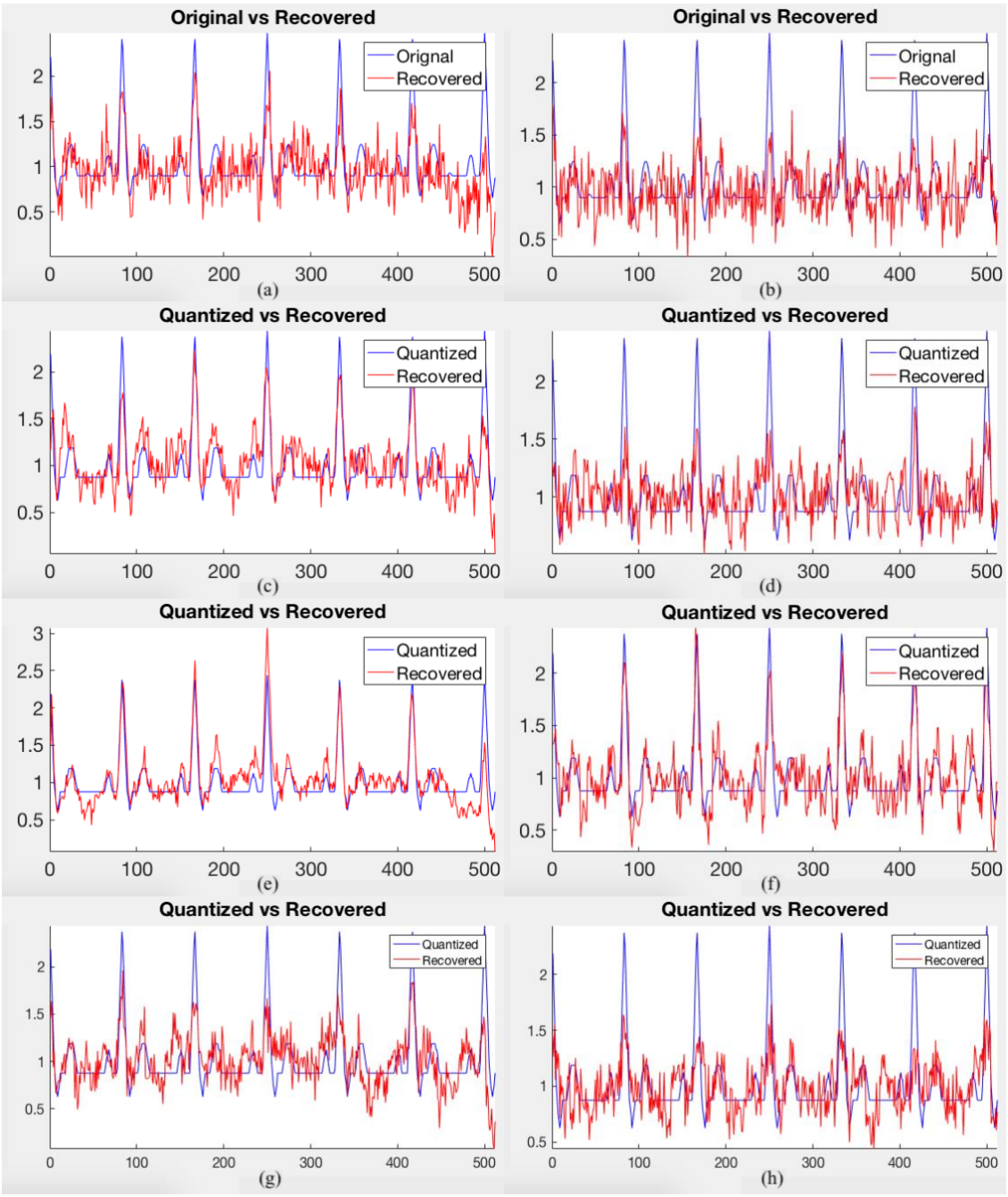


Figura 4.40 Señal ECG original [12] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_1$, (d) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_2$, (e) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (f) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (g) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (h) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

Los seis escenarios en los que se probaron ambas señales de entrada son:

1.a Arquitectura digital CAC-OST para CR_1 y $\mathbf{f}_1$: Comprime la señal en ocho. La Figura 4.42 muestra la señal cuantificada que se va a comprimir. Los resultados en la recuperación de la señal son:

a) La Figura 4.43 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.03501920$.

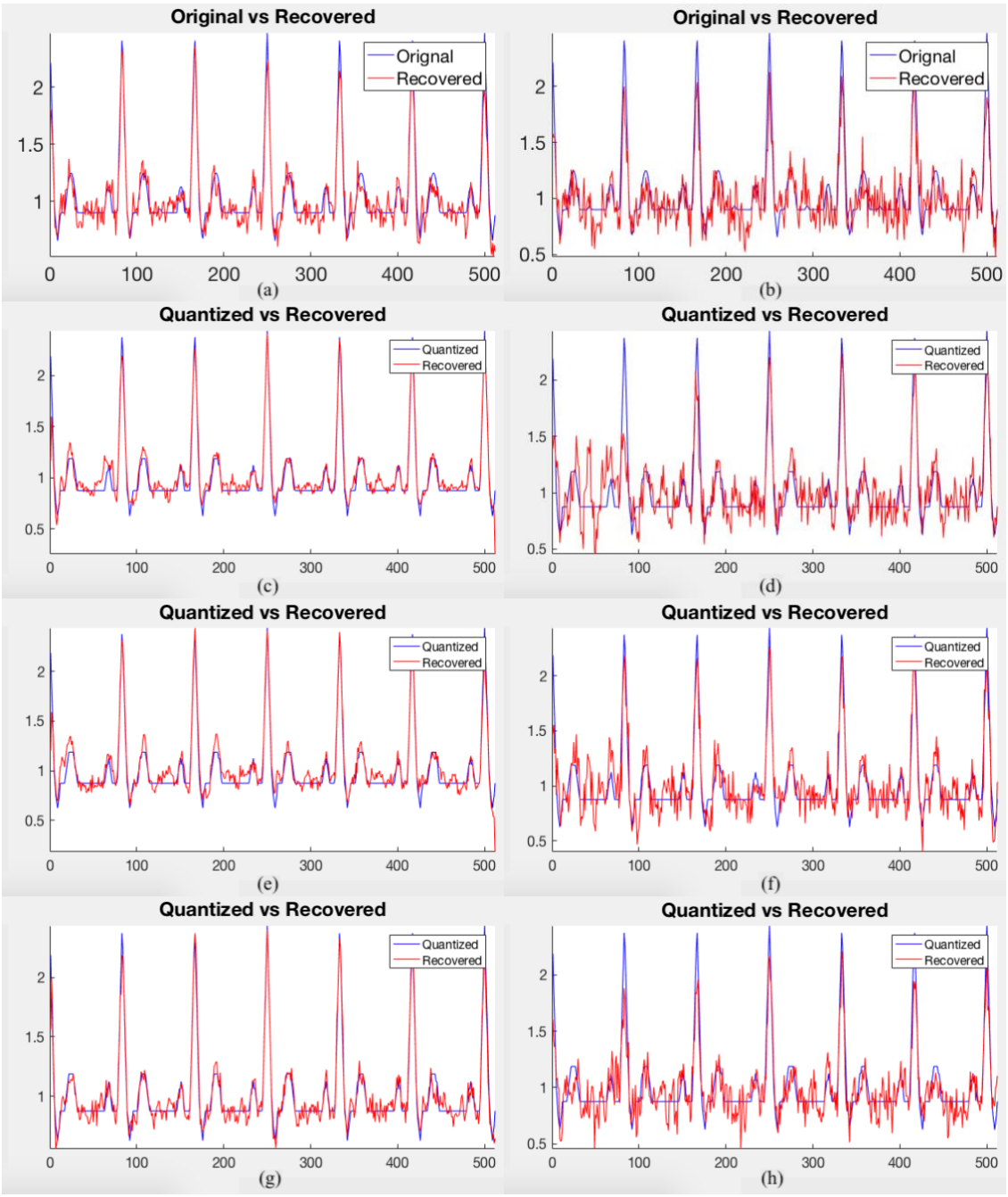


Figura 4.41 Señal ECG original [12] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_1$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_1$, (d) $\mathbf{\Phi}_2$ y $\mathbf{\Psi}_2$, (e) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (f) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (g) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (h) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

b) La Figura 4.44 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.03458600$.

1.b Arquitectura digital CAC-OST para CR_1 y $\mathbf{f}_2$: Comprime la señal en ocho. La Figura 4.45 muestra la señal cuantificada que se va a comprimir. Los resultados en la recuperación de la señal son:

a) La Figura 4.46 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.04096190$.

Tabla 4.15 Comparación de tasa de error en la recuperación de la señal para CR_1.

Muestreo compreso	Arquitectura digital CAC-SS	Arquitectura digital CAC-OST	Arquitectura digital [13]	*Software*
f	Señal ECG simulada [12]			
N	512			
M	64			
CR	8			
Φ	CAC-SS	CAC-OST	Bernoulli	Gaussian
Recuperación	Optimización l_1 [23]			
Ψ	DCT			
MSE	0.20799200	0.12917200	0.22561500	0.19539000
Ψ	WHT			
MSE	0.13206900	0.09559340	0.14120200	0.12744800

Tabla 4.16 Comparación de tasa de error en recuperación de la señal para CR_2.

Muestreo compreso	Arquitectura digital CAC-SS	Arquitectura digital CAC-OST	Arquitectura digital [13]	*Software*
f	Señal ECG simulada [12]			
N	512			
M	128			
CR	4			
Φ	CAC-SS	CAC-OST	Bernoulli	Gaussian
Recuperación	Optimización l_1 [23]			
Ψ	DCT			
MSE	0.07669780	0.04751370	0.05316020	0.07467780
Ψ	WHT			
MSE	0.07739020	0.04683650	0.07565700	0.09413290

Tabla 4.17 Comparación de tasa de error en recuperación de la señal para CR_3.

Muestreo compreso	Arquitectura digital CAC-SS	Arquitectura digital CAC-OST	Arquitectura digital [13]	*Software*
f	Señal ECG simulada [12]			
N	512			
M	256			
CR	2			
Φ	CAC-SS	CAC-OST	Bernoulli	Gaussian
Recuperación	Optimización l_1 [23]			
Ψ	DCT			
MSE	0.00575565	0.00936020	0.00743218	0.01529000
Ψ	WHT			
MSE	0.02539030	0.02167620	0.03263760	0.03111390

Figura 4.42 Señal $\mathbf{f}_1$ cuantificada.

b) La Figura 4.47 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.03448580$.

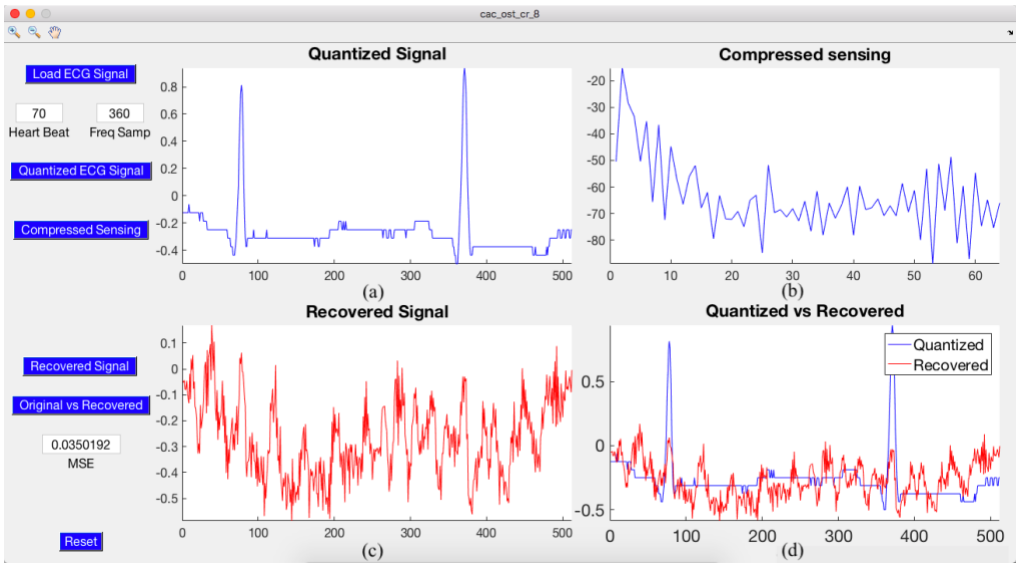


Figura 4.43 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

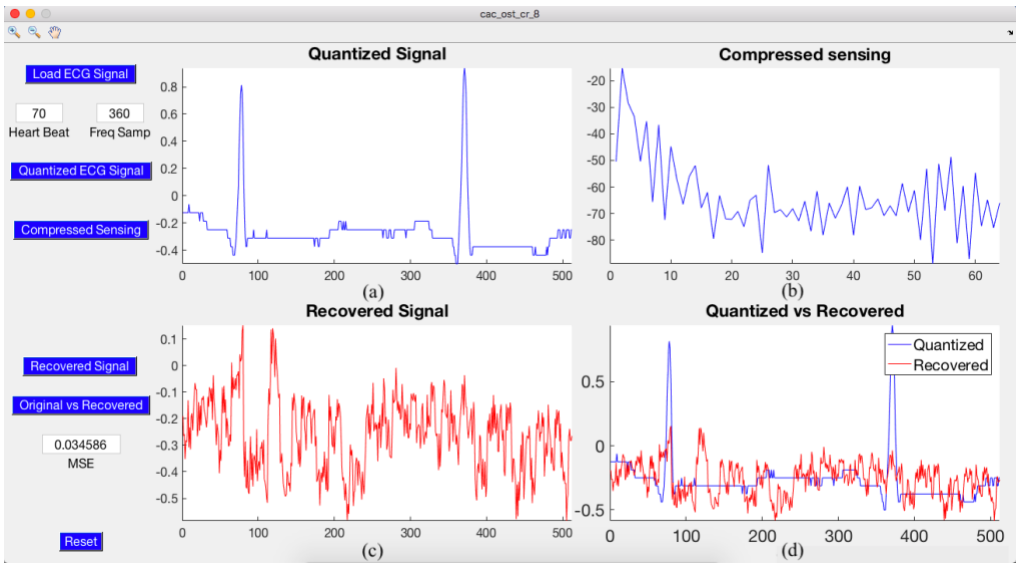


Figura 4.44 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

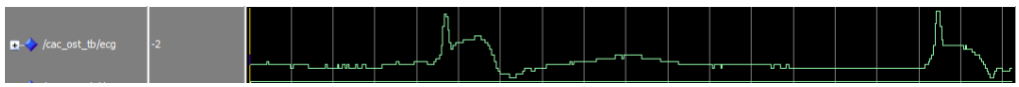


Figura 4.45 Señal $\mathbf{f}_2$ cuantificada.

2.a Arquitectura digital CAC-OST para CR_2 y $\mathbf{f}_1$: Comprime la señal en cuatro. Los resultados en la recuperación de la señal son:

a) La Figura 4.48 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el MSE =0.02056760.
b) La Figura 4.49 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el MSE =0.01857970.

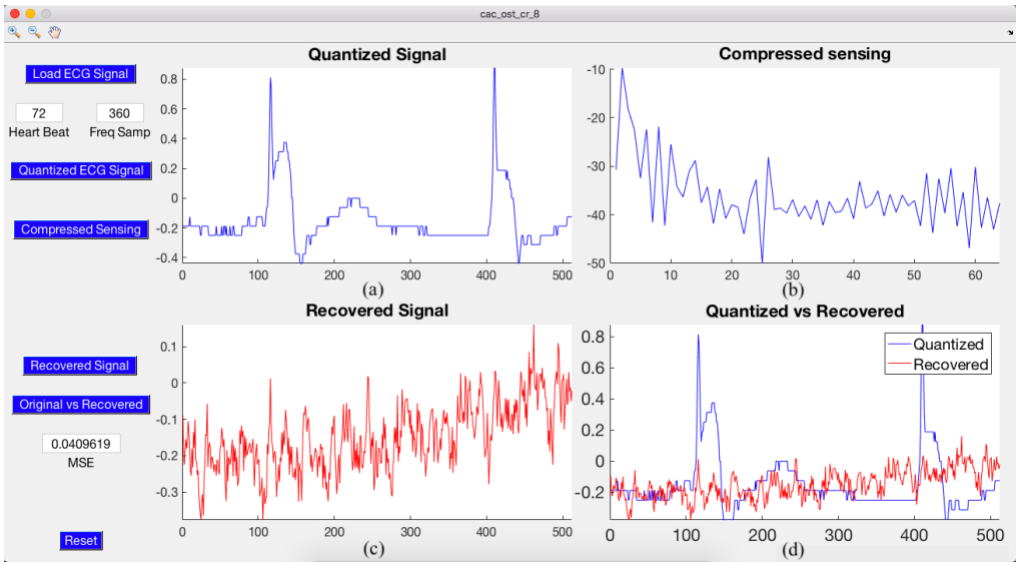


Figura 4.46 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

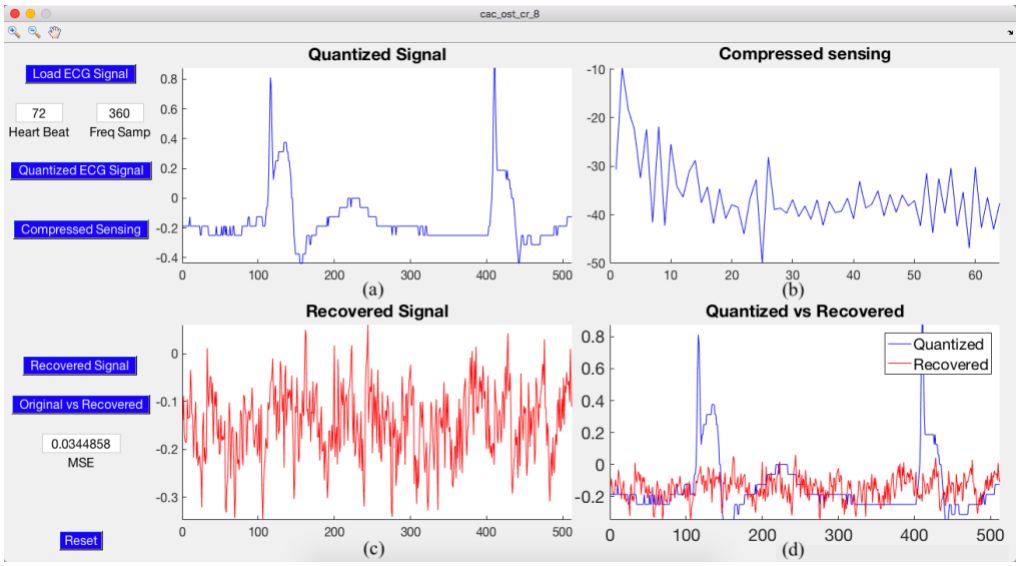


Figura 4.47 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

2.b Arquitectura digital CAC-OST para CR_2 y $\mathbf{f}_2$: Comprime la señal en cuatro. Los resultados en la recuperación de la señal son:

a) La Figura 4.50 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.02019170$.
b) La Figura 4.51 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.02200230$.

3.a Arquitectura digital CAC-OST para CR_3 y $\mathbf{f}_1$: Comprime la señal en dos. Los resultados en la recuperación de la señal son:

a) La Figura 4.52 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.00174662$.
b) La Figura 4.53 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.00425785$.

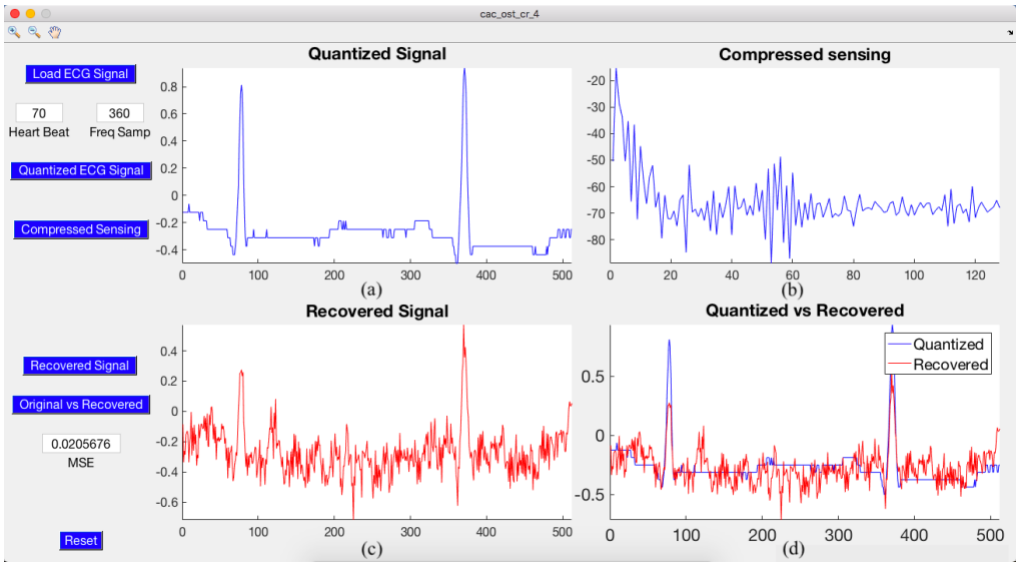


Figura 4.48 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

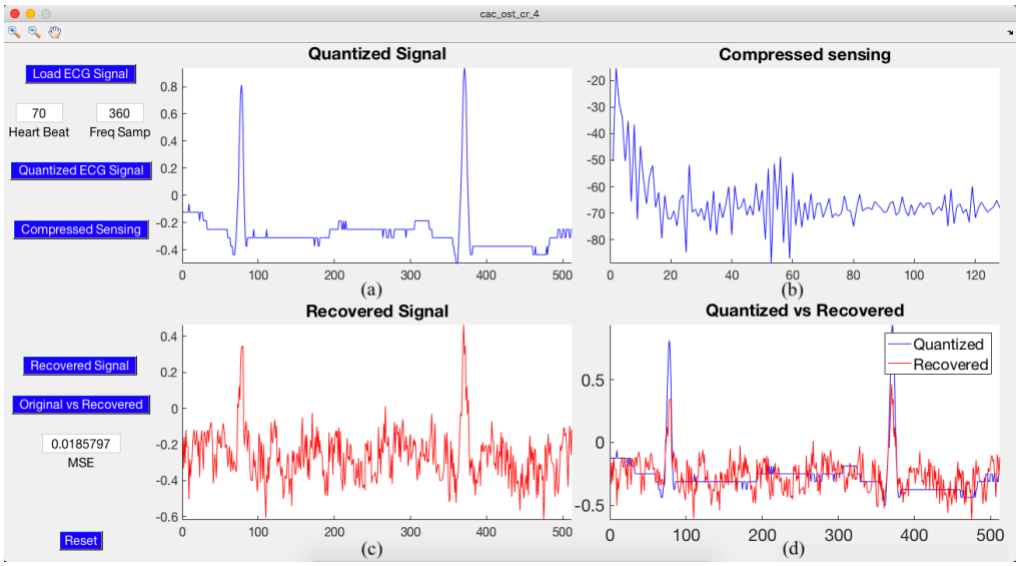


Figura 4.49 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

3.b Arquitectura digital CAC-OST para CR_3 y $\mathbf{f}_2$: Comprime la señal en dos. Los resultados en la recuperación de la señal son:

a) La Figura 4.54 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.00616377$.
b) La Figura 4.55 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.00676680$.

4.a Arquitectura digital CAC-SS para CR_1 y $\mathbf{f}_1$: Comprime la señal en ocho. Los resultados en la recuperación de la señal son:

a) La Figura 4.56 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.04127130$.
b) La Figura 4.57 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.02550090$.

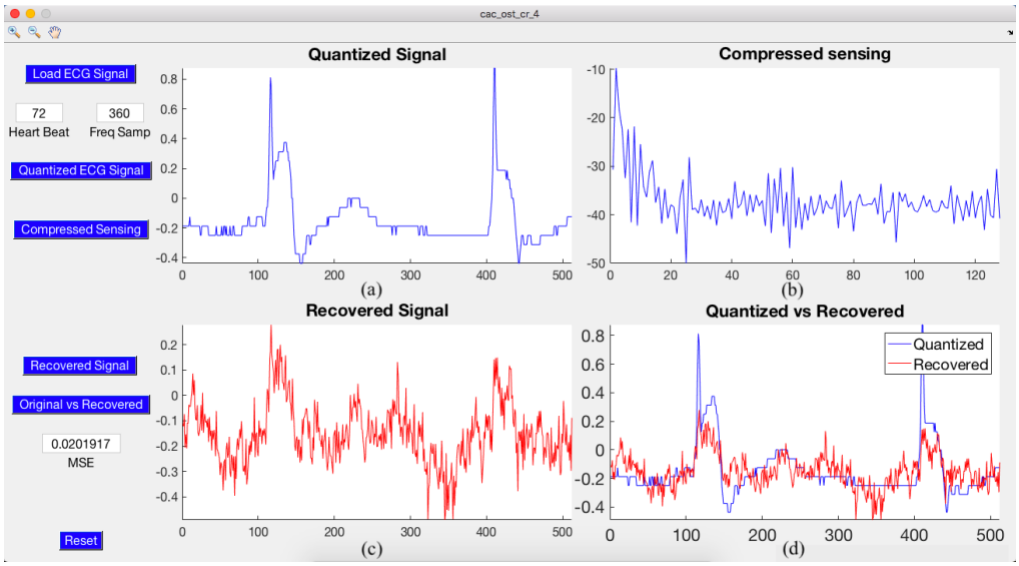


Figura 4.50 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

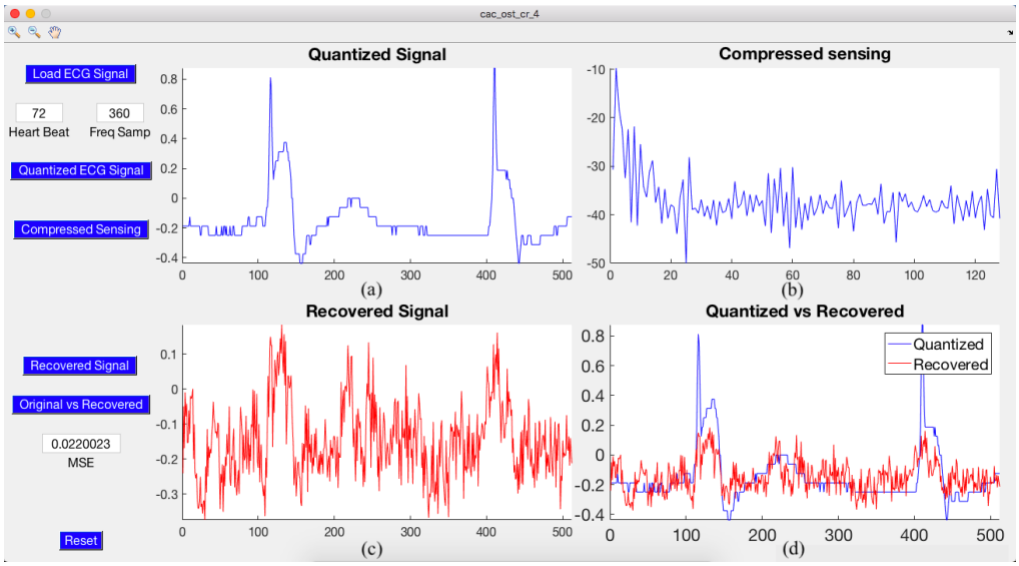


Figura 4.51 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

4.b Arquitectura digital CAC-SS para CR_1 y $\mathbf{f}_2$: Comprime la señal en ocho. Los resultados en la recuperación de la señal son:

a) La Figura 4.58 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.02661700$.
b) La Figura 4.59 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.02644540$.

5.a Arquitectura digital CAC-SS para CR_2 y $\mathbf{f}_1$: Comprime la señal en cuatro. Los resultados en la recuperación de la señal son:

a) La Figura 4.60 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.01652850$.
b) La Figura 4.61 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.01691960$.

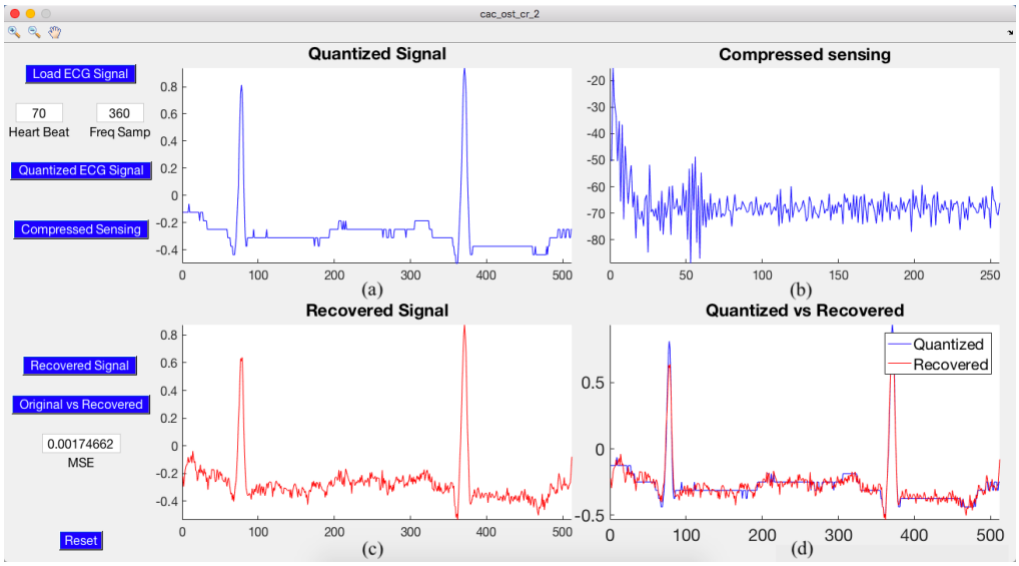


Figura 4.52 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

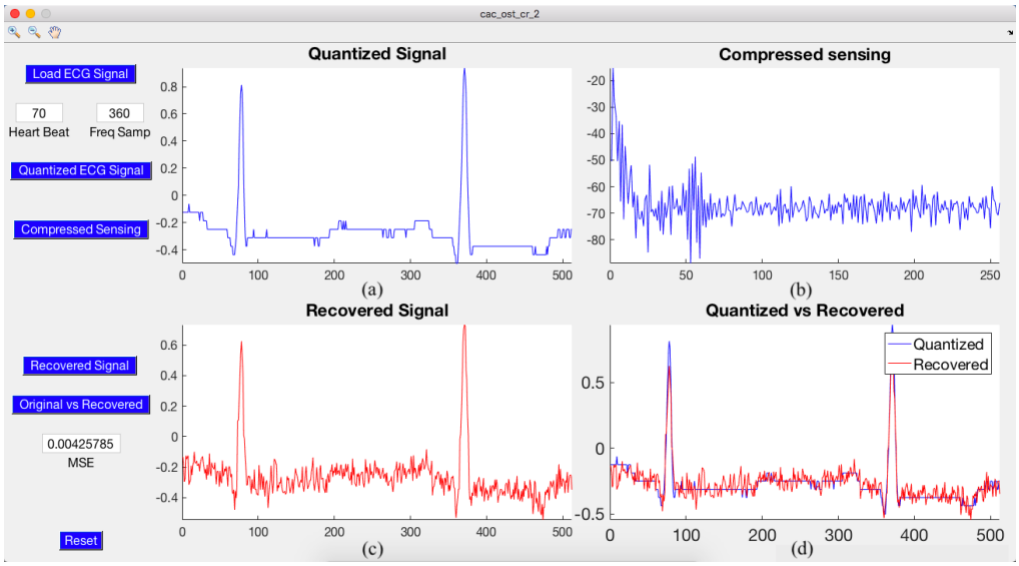


Figura 4.53 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

5.b Arquitectura digital CAC-SS para CR_2 y $\mathbf{f}_2$: Comprime la señal en cuatro. Los resultados en la recuperación de la señal son:

a) La Figura 4.62 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.01187290$.
b) La Figura 4.63 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.01614500$.

6.a Arquitectura digital CAC-SS para CR_3 y $\mathbf{f}_1$: Comprime la señal en dos. Los resultados en la recuperación de la señal son:

a) La Figura 4.64 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.00261408$.
b) La Figura 4.65 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.00474148$.

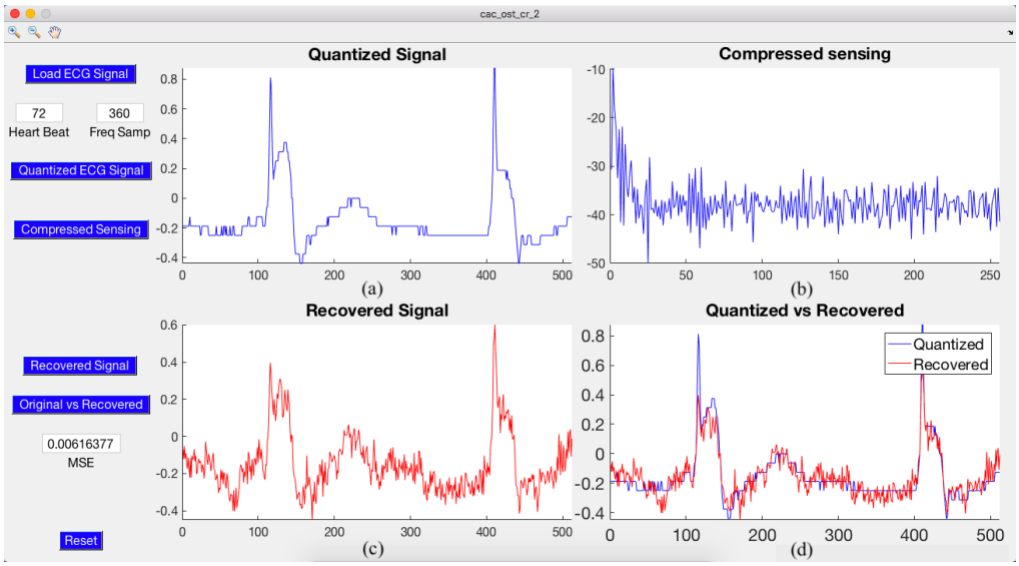


Figura 4.54 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

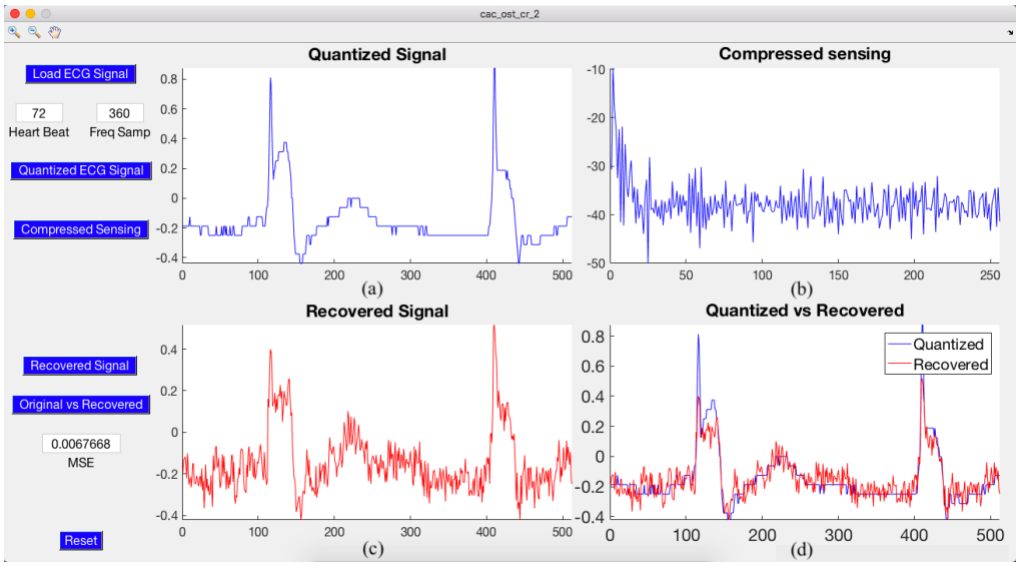


Figura 4.55 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_3$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

6.b Arquitectura digital CAC-SS para CR_3 y $\mathbf{f}_2$: Comprime la señal en dos. Los resultados en la recuperación de la señal son:

a) La Figura 4.66 muestra los resultados con $\mathbf{\Psi}_1$ y se observa el $MSE = 0.00261350$.
b) La Figura 4.67 muestra los resultados con $\mathbf{\Psi}_2$ y se observa el $MSE = 0.00439140$.

La Figura 4.68, Figura 4.69 y Figura 4.70 comparan las gráficas de la señal ECG real $\mathbf{f}_1$ [24] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos respectivamente de ambas arquitecturas digitales propuestas, mientras que la Figura 4.71, Figura 4.72 y Figura 4.73 comparan las gráficas de la señal ECG real $\mathbf{f}_2$ [24] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos respectivamente.

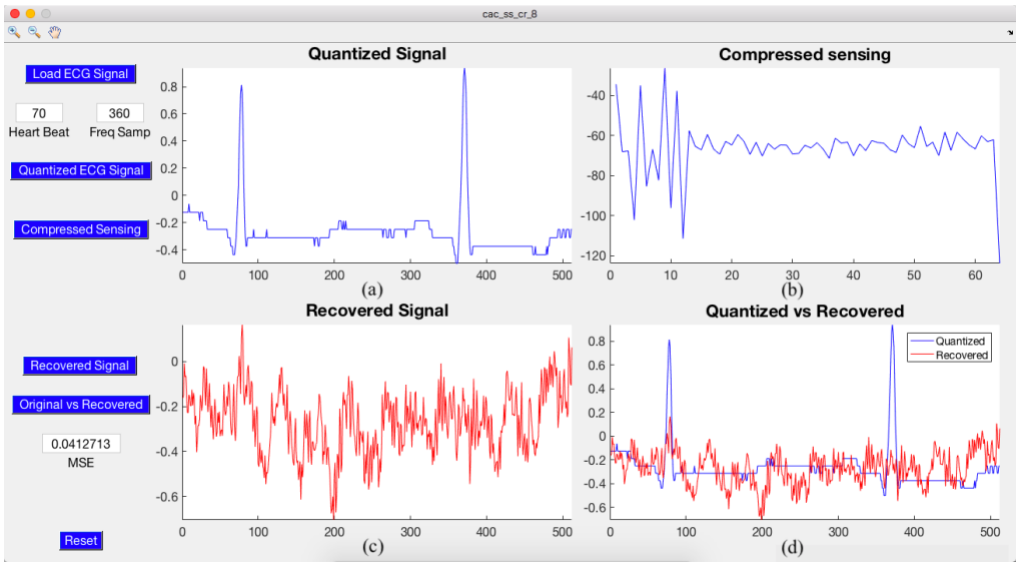


Figura 4.56 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

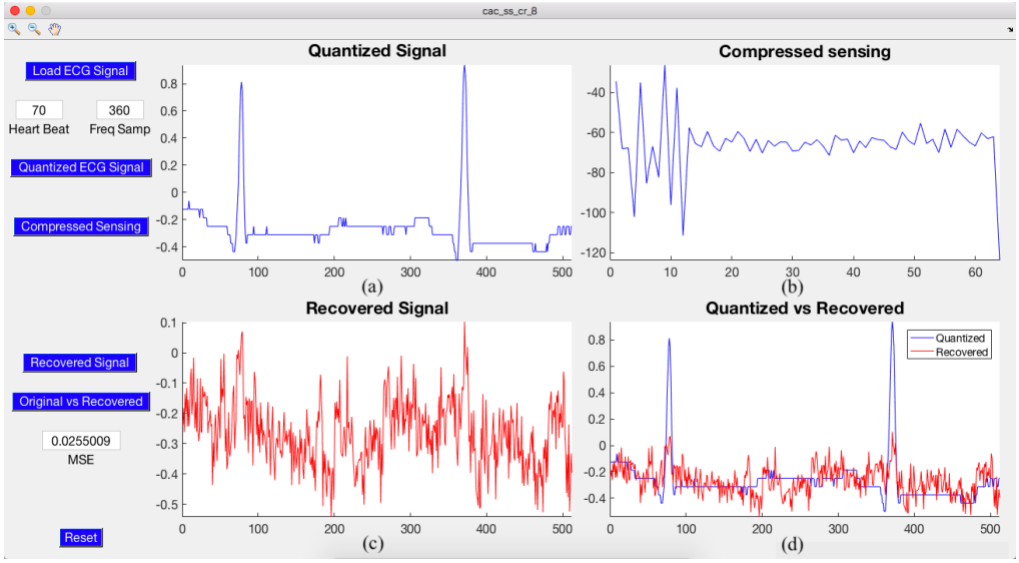


Figura 4.57 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

La menor tasa de error en la recuperación de la señal $\mathbf{f}_1$ la presenta la arquitectura digital CAC-OST cuando la razón de compresión es de dos y la dispersión de la señal ECG es DCT, mientras que la menor tasa de error en la recuperación de la señal $\mathbf{f}_2$ la presenta la arquitectura digital CAC-SS cuando la razón de compresión es de dos y la dispersión de la señal ECG es DCT. La Tabla 4.18, Tabla 4.19 y Tabla 4.20 contrastan la tasa de error en la recuperación de la señal $\mathbf{f}_1$ para razón de compresión de ocho, cuatro y dos respectivamente, mientras que la Tabla 4.21, Tabla 4.22 y Tabla 4.23 contrastan dicha tasa de error en la recuperación de la señal $\mathbf{f}_2$ para razón de compresión de ocho, cuatro y dos respectivamente.

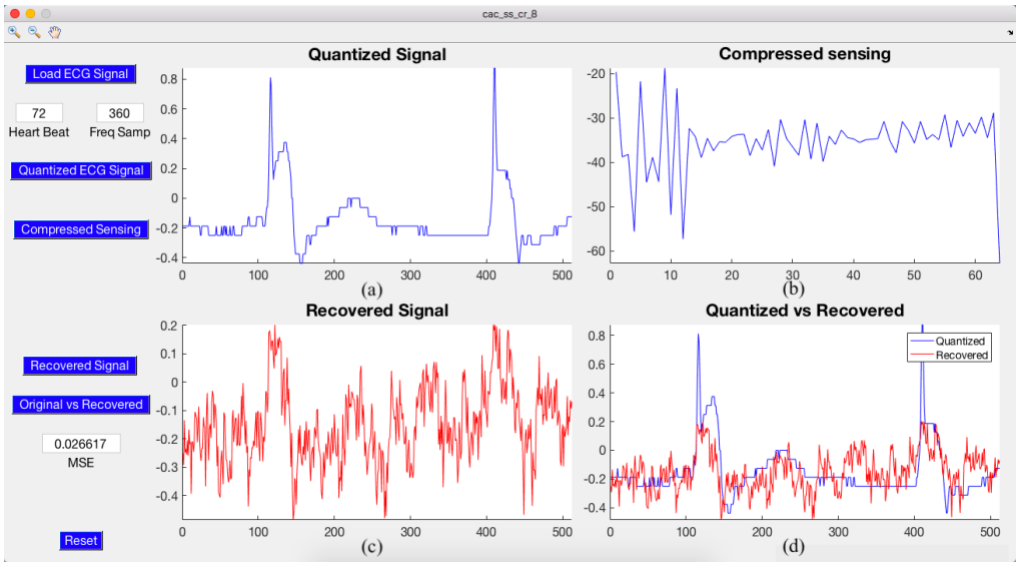


Figura 4.58 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

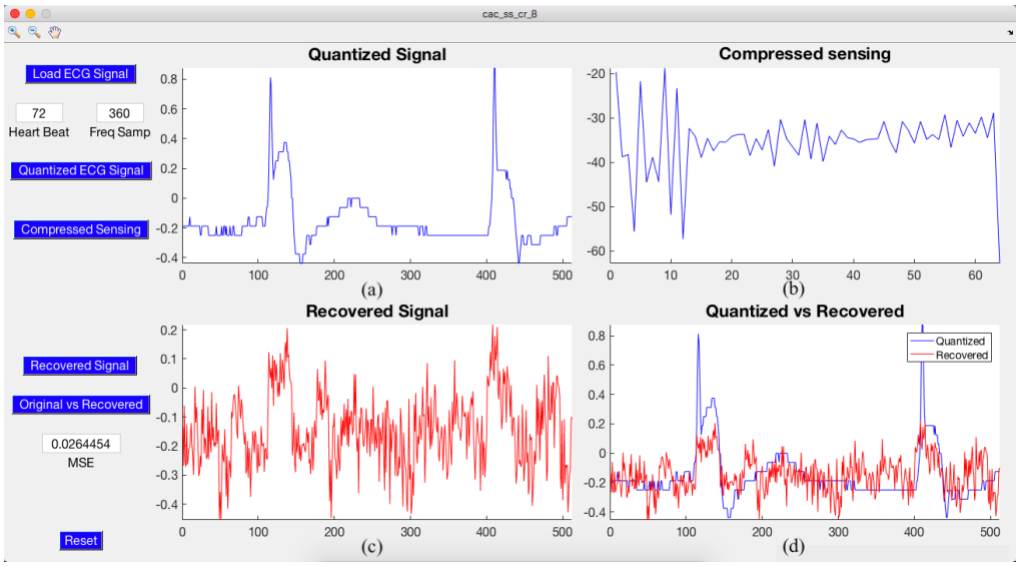


Figura 4.59 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_1, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

4.4.2 Señal EMG real

Las señales de entrada utilizadas en este escenario fueron tomadas de la base de datos de *PhysioBank* [24]. Las señales consideradas para estas pruebas son:

a) $\mathbf{f}_3$ es EMG *Myopathym*: Paciente masculino de 57 años con miopatía debido a polimiositosis. La Figura 4.74 muestra la señal cuantificada.
b) $\mathbf{f}_4$ es EMG *Neuropathym*: Paciente masculino de 62 años con dolor lumbar crónico y neuropatía debido a una radiculopatía L5 derecha. La Figura 4.75 muestra la señal cuantificada.

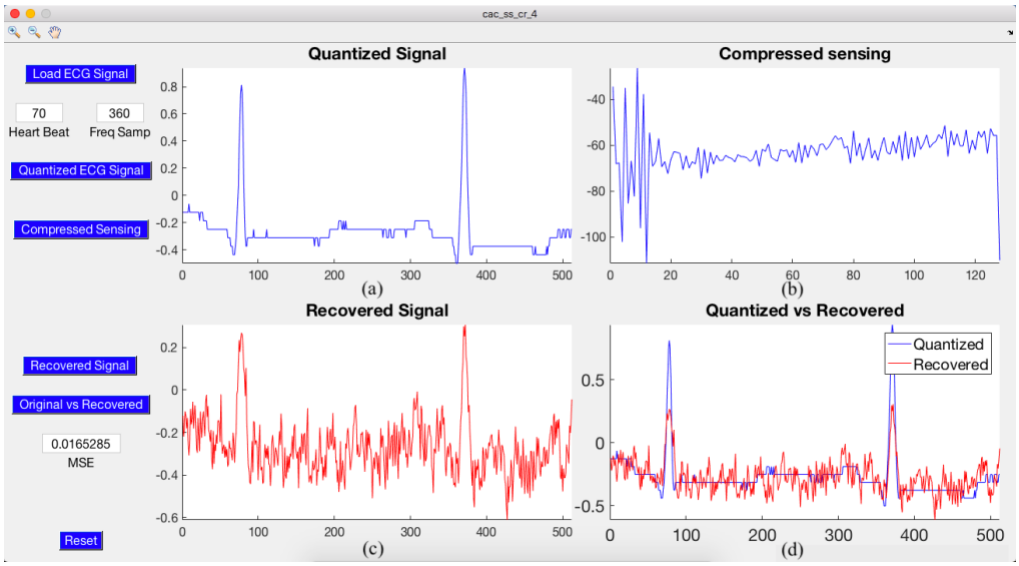


Figura 4.60 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

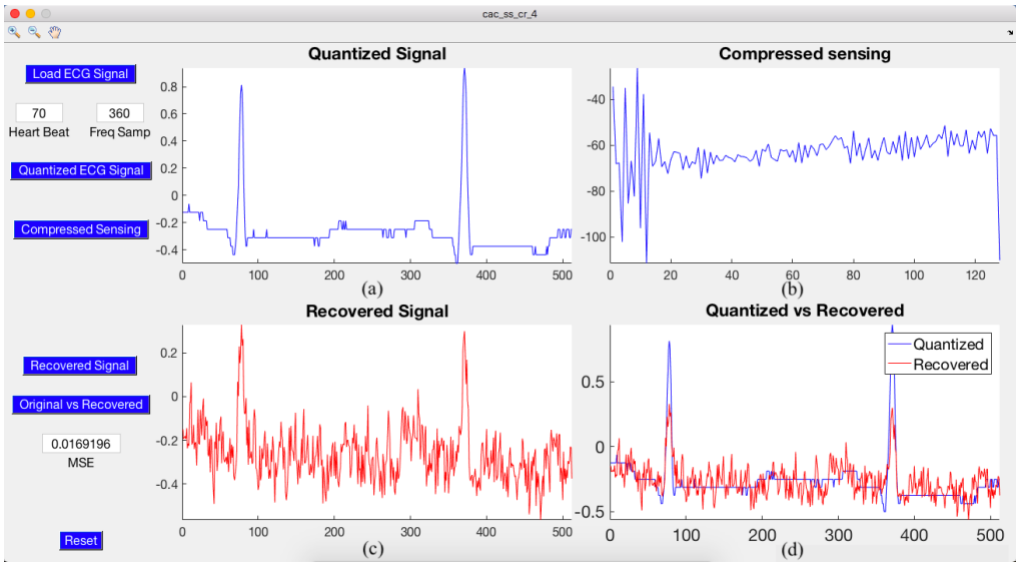


Figura 4.61 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

La Figura 4.76, Figura 4.77 y Figura 4.78 comparan las gráficas de la señal EMG real $\mathbf{f}_3$ [24] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos respectivamente de ambas arquitecturas digitales propuestas, mientras que la Figura 4.79, Figura 4.80 y Figura 4.81 comparan las gráficas de la señal EMG real $\mathbf{f}_4$ [24] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos respectivamente.

La menor tasa de error en la recuperación en ambas señales, $\mathbf{f}_3$ y $\mathbf{f}_4$, la presenta la arquitectura digital CAC-OST cuando la razón de compresión es de dos y la dispersión de la señal EMG es DCT. La Tabla 4.24, Tabla 4.25 y Tabla 4.26 contrastan la tasa de error en la recuperación de la señal $\mathbf{f}_3$ para razón de compresión de ocho, cuatro y dos respectivamente, mientras que la Tabla 4.27, Tabla 4.28 y Tabla 4.29 contrastan dicha tasa de error en la recuperación de la señal $\mathbf{f}_4$ para

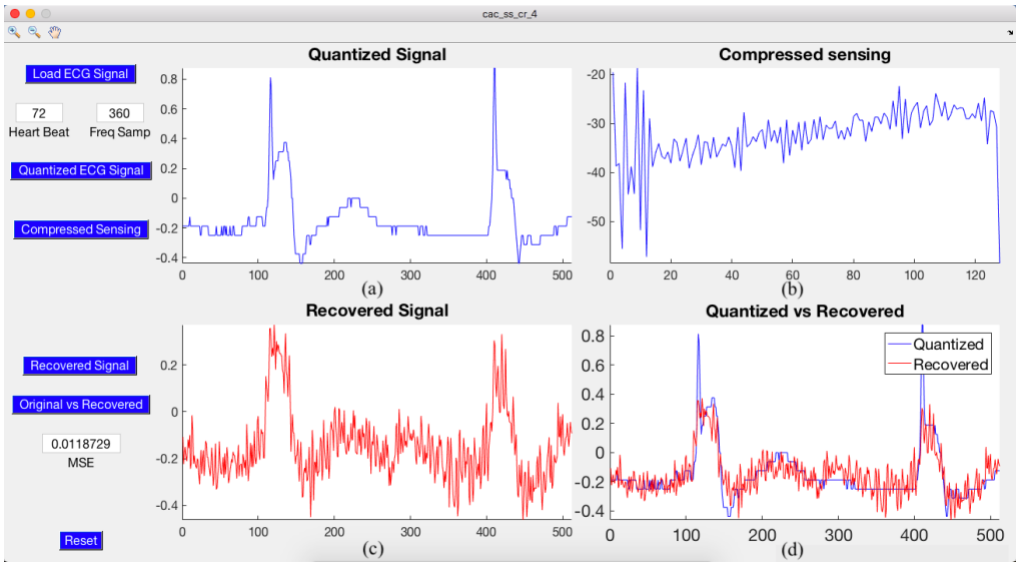


Figura 4.62 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

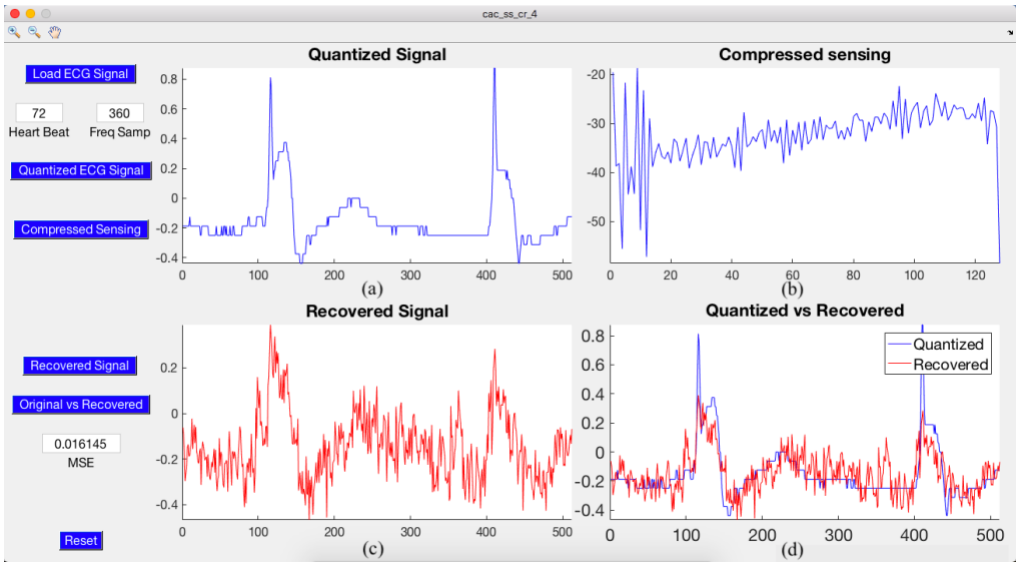


Figura 4.63 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_2, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

razón de compresión de ocho, cuatro y dos respectivamente.

4.4.3 Audio

Las señales de entrada utilizas en este escenario fueron tomadas de la base de datos de *SampleSwap* [25]. Las señales consideradas para estas pruebas son:

a) $\mathbf{f}_5$ es brilliant.wav.
b) $\mathbf{f}_6$ es arresting.wav.

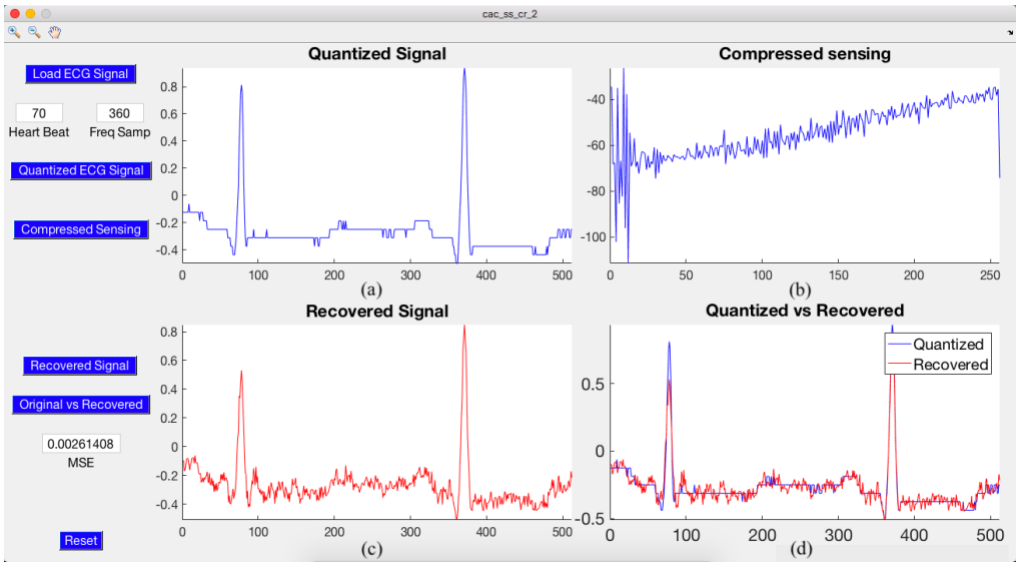


Figura 4.64 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

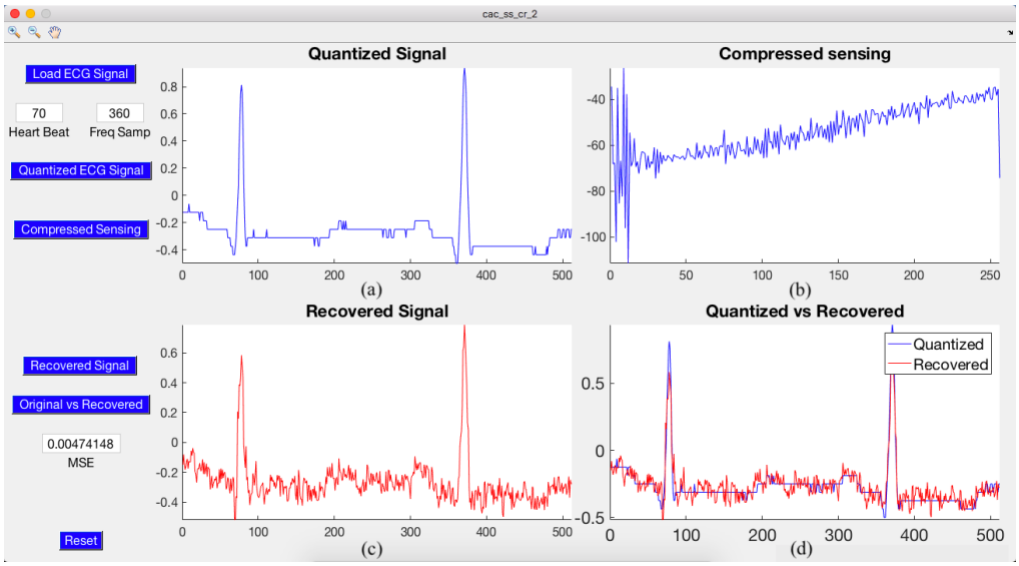


Figura 4.65 (a) Señal ECG real $\mathbf{f}_1$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

La Figura 4.82, Figura 4.83 y Figura 4.84 comparan las gráficas de la señal de audio $\mathbf{f}_5$ [25] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos respectivamente de ambas arquitecturas digitales propuestas, mientras que la Figura 4.85, Figura 4.86 y Figura 4.87 comparan las gráficas de la señal de audio $\mathbf{f}_6$ [25] *vs* señal recuperada para razón de compresión de ocho, cuatro y dos respectivamente.

La menor tasa de error en la recuperación de las señales $\mathbf{f}_5$ y $\mathbf{f}_6$ la presenta la arquitectura digital CAC-SS cuando la razón de compresión es de dos y la dispersión de la señal de audio es DCT. La Tabla 4.30, Tabla 4.31 y Tabla 4.32 contrastan la tasa de error en la recuperación de la señal $\mathbf{f}_5$ para razón de compresión de ocho, cuatro y dos respectivamente, mientras que la

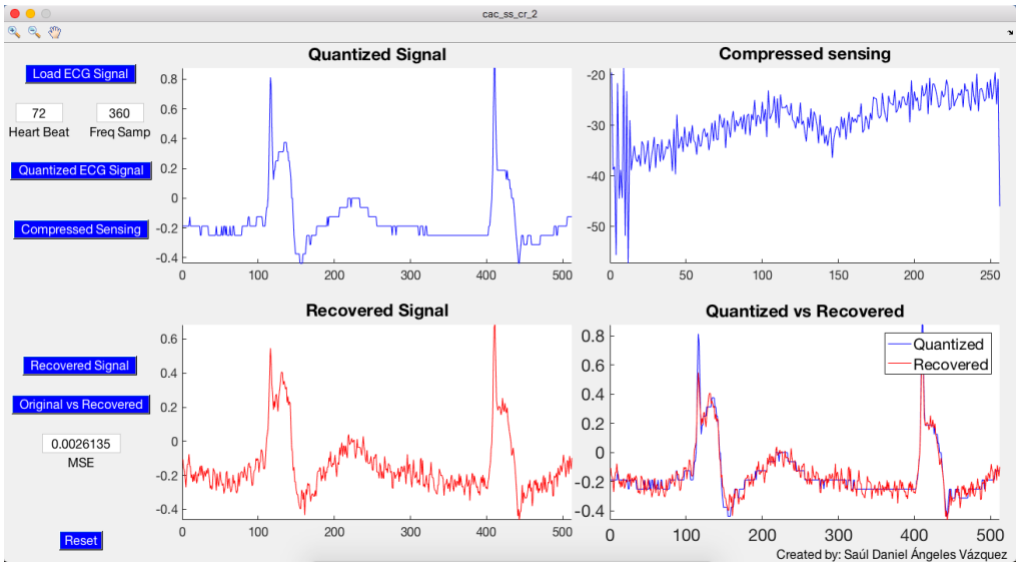


Figura 4.66 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_1$ y (d) Señal original *vs* señal recuperada.

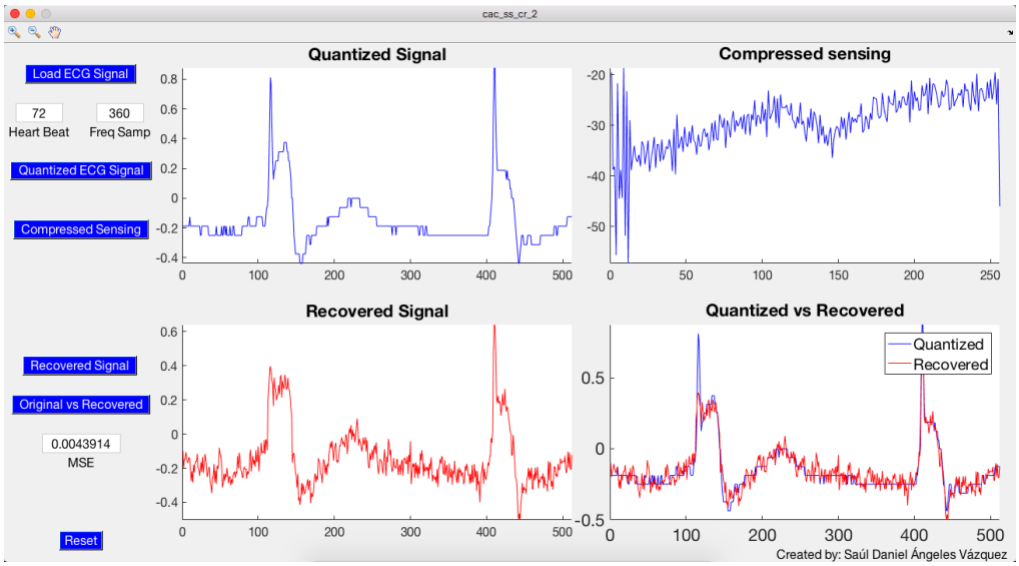


Figura 4.67 (a) Señal ECG real $\mathbf{f}_2$ [24], (b) Muestreo compreso con $\mathbf{\Phi}_4$ y CR_3, (c) Recuperación con $\mathbf{\Psi}_2$ y (d) Señal original *vs* señal recuperada.

Tabla 4.33, Tabla 4.34 y Tabla 4.35 contrastan dicha tasa de error en la recuperación de la señal $\mathbf{f}_6$ para razón de compresión de ocho, cuatro y dos respectivamente.

4.4.4 Imagen

La señal de entrada utilizada en este escenario fue tomada de [26]. La Figura 4.88 muestra la señal de entrada elegida "lena.tiff". La señal de entrada es convertida a escala de grises y escalada a un tamaño de 128x128 pixeles para realizar las pruebas. La Figura 4.89 muestra la señal de entrada $\mathbf{f}_7$.

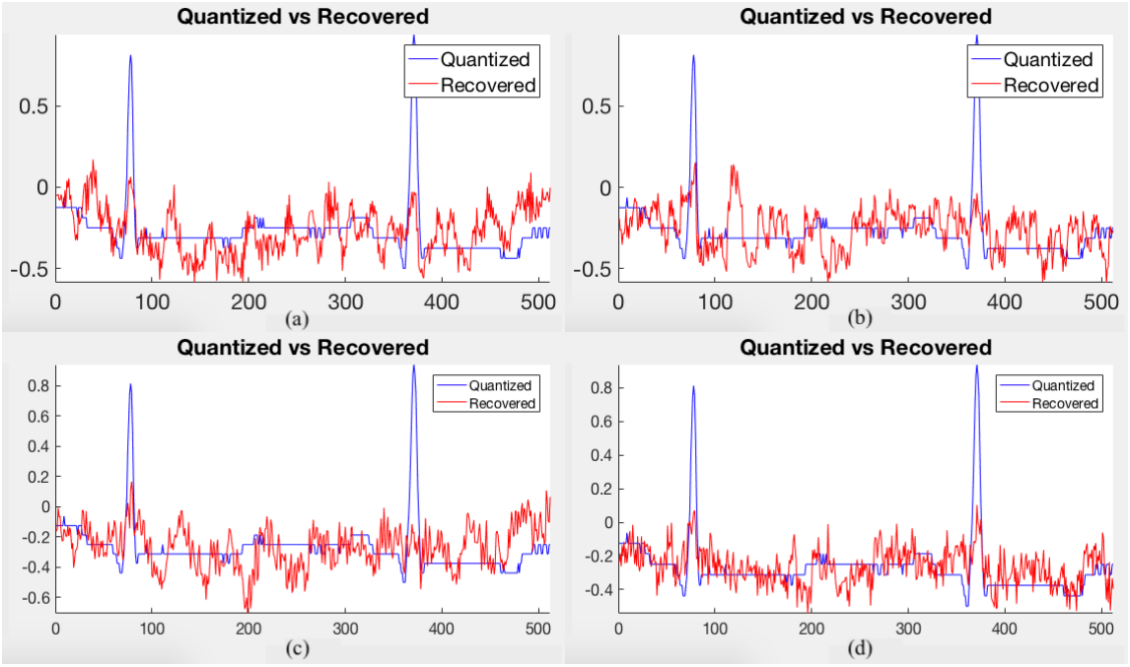


Figura 4.68 Señal ECG real $\mathbf{f}_1$ [24] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

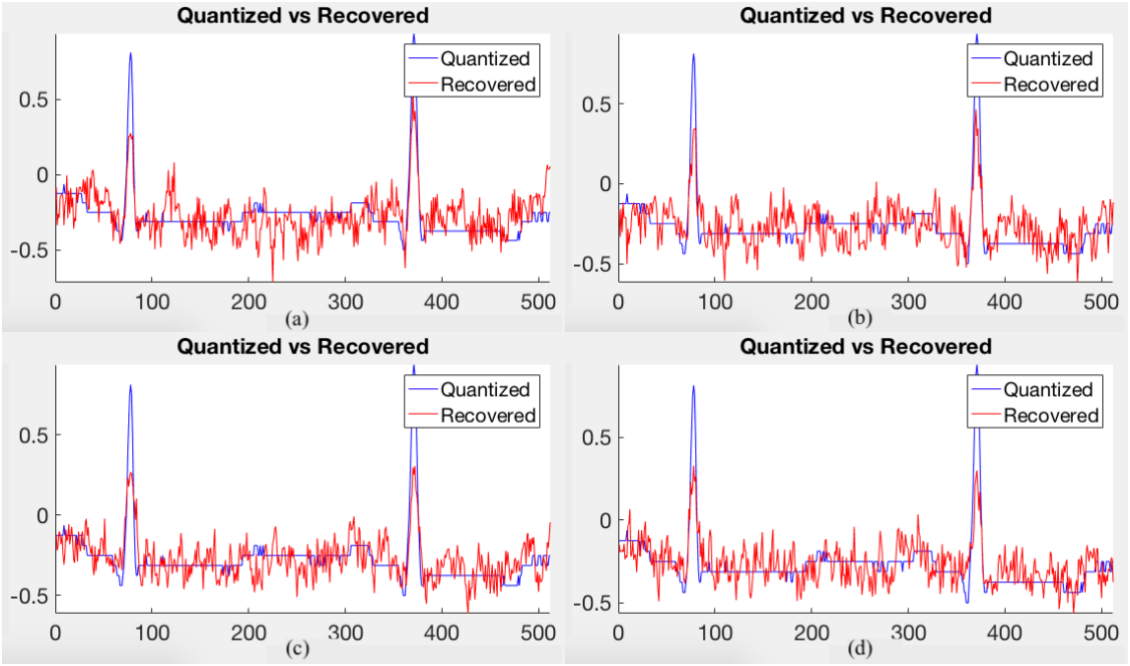


Figura 4.69 Señal ECG real $\mathbf{f}_1$ [24] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

La Figura 4.90 muestra la imagen recuperada para razón de compresión de dos. La menor tasa de error, así como la mejor proporción y relación de señal a ruido en la recuperación de la señal $\mathbf{f}_7$ la presenta la arquitectura digital CAC-SS cuando la razón de compresión es de dos y la dispersión de la señal es DCT. La Tabla 4.36 contrastan la tasa de error, la proporción máxima de señal a ruido y la relación señal a ruido en la recuperación de la señal $\mathbf{f}_7$ para razón de compresión de dos.

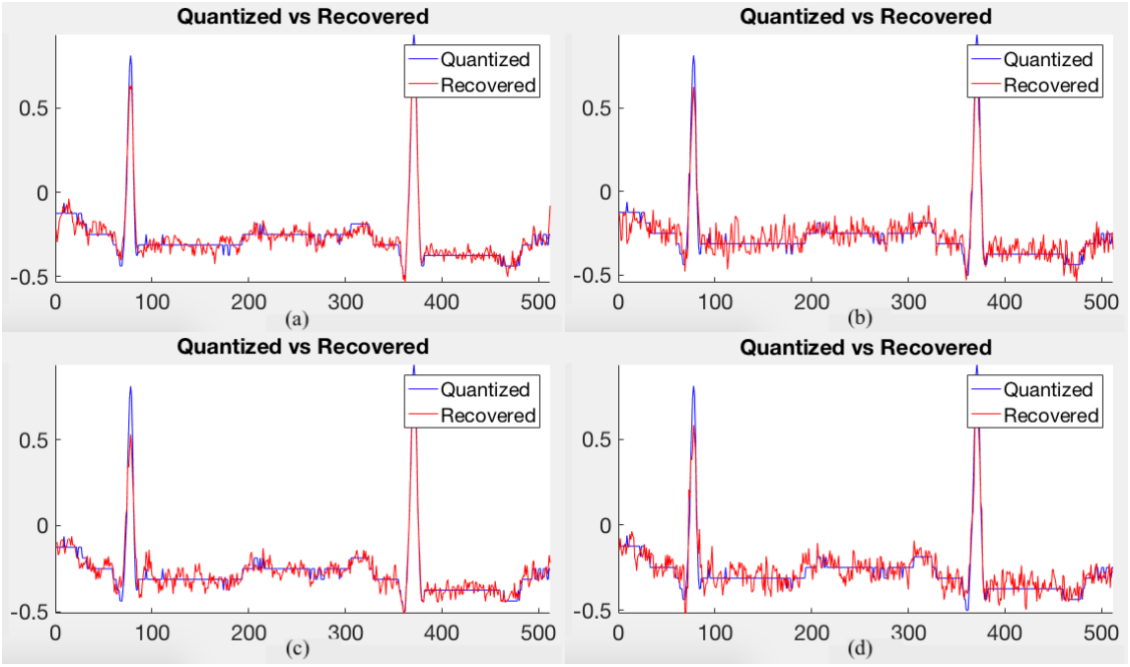


Figura 4.70 Señal ECG real $\mathbf{f}_1$ [24] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$

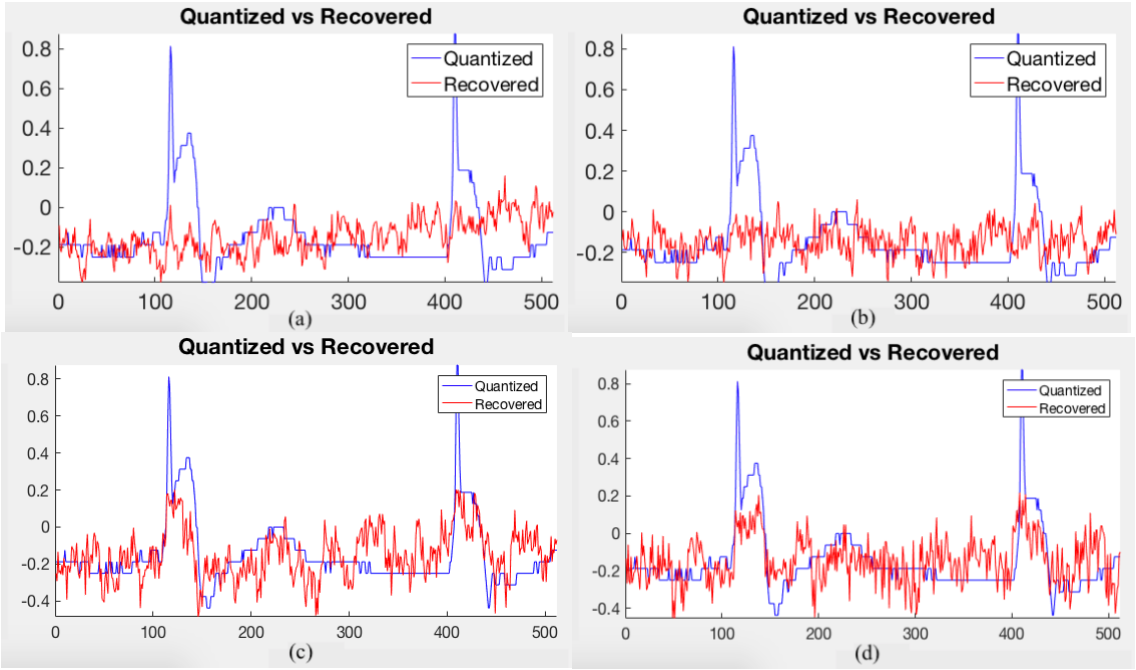


Figura 4.71 Señal ECG real $\mathbf{f}_2$ [24] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

4.5 Análisis de resultados

La arquitectura digital CAC-SS con razón de compresión de ocho presenta una reducción de cuatro por ciento en elementos lógicos y uno por ciento en registros, en comparación con la

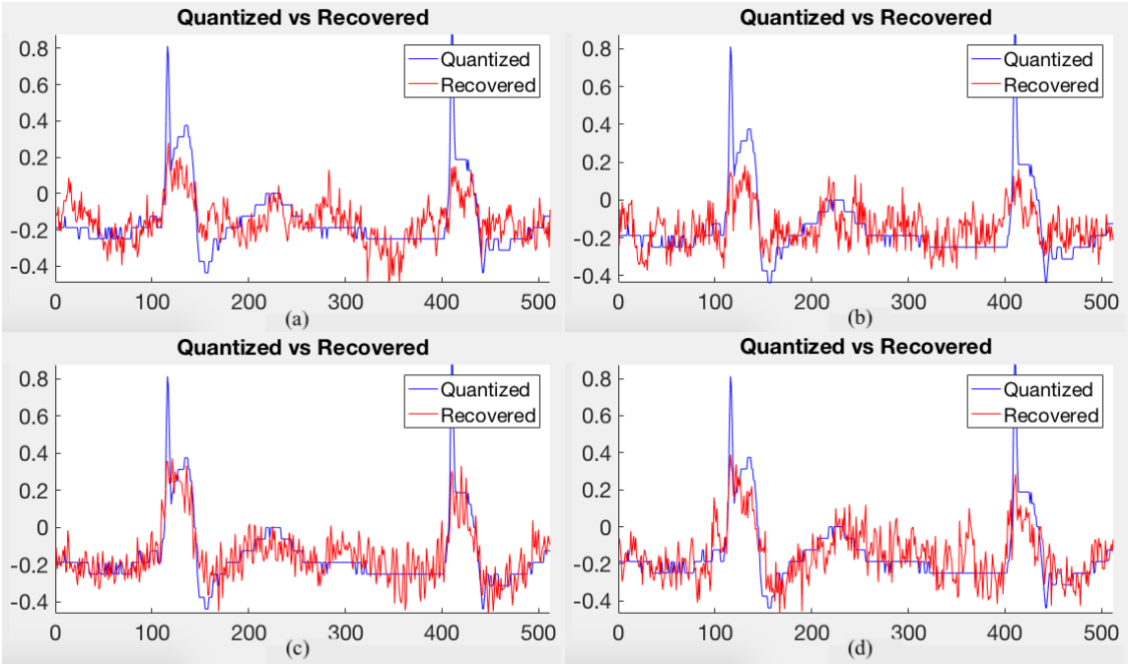


Figura 4.72 Señal ECG real $\mathbf{f}_2$ [24] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

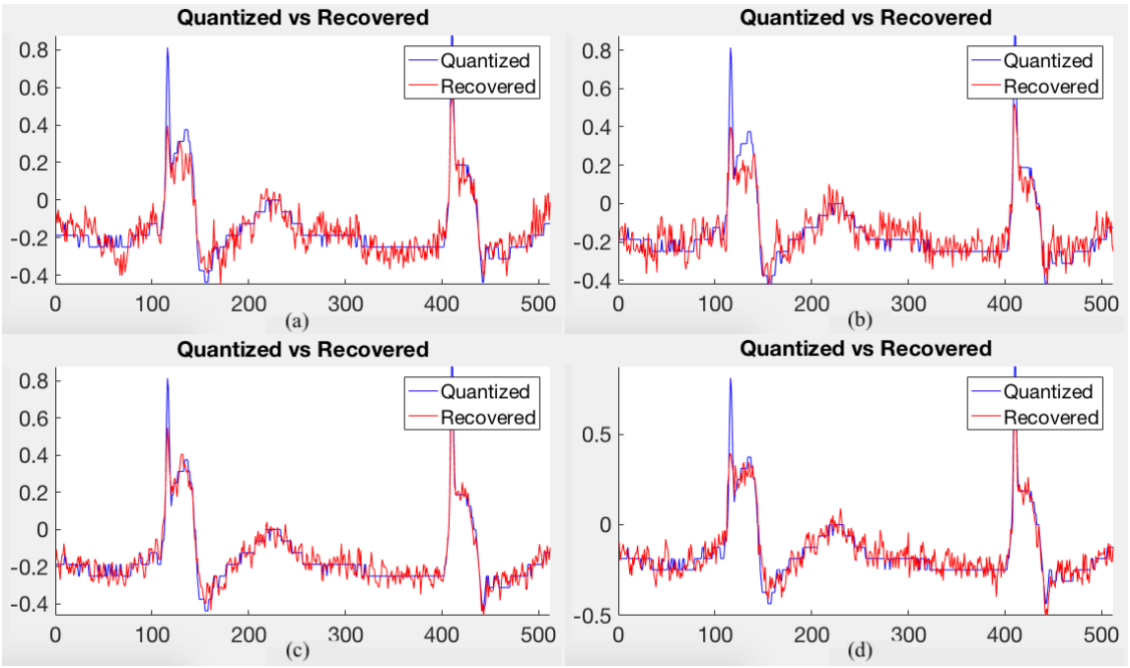


Figura 4.73 Señal ECG real $\mathbf{f}_2$ [24] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$

arquitectura digital [13]. También presenta una reducción de quince por ciento en elementos lógicos y dos por ciento en registros, en comparación la arquitectura digital CAC-OST. En términos de frecuencia máxima de operación, la arquitectura digital CAC-SS es tres por ciento y cuatrocientos ochenta y cuatro por ciento más rápida que la arquitectura digital [13] y la arquitectura digital CAC-OST respectivamente.

Tabla 4.18 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_1$ para CR_1.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal ECG real (record 100) [24]			
N	512			
M	64			
CR	8			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.04127130	0.0255090	0.03501920	0.03458600

Tabla 4.19 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_1$ para CR_2.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal ECG real (record 100) [24]			
N	512			
M	128			
CR	4			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.01652850	0.01691960	0.02056760	0.01857970

Tabla 4.20 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_1$ para CR_3.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal ECG real (record 100) [24]			
N	512			
M	256			
CR	2			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00261408	0.00474148	0.00174662	0.00425785

Tabla 4.21 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_2$ para CR_1.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal ECG real (record 100) [24]			
N	512			
M	64			
CR	8			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.02661700	0.02644540	0.04096190	0.03448580

La arquitectura digital CAC-SS con razón de compresión de cuatro presenta una reducción de cuatro por ciento en elementos lógicos y uno por ciento en registros, en comparación con la arquitectura digital [13]. También presenta una reducción de veinte por ciento en elementos lógicos y dieciocho por ciento en registros, en comparación la arquitectura digital CAC-OST. En

Tabla 4.22 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_2$ para CR_2.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal ECG real (record 100) [24]			
N	512			
M	128			
CR	4			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.01187290	0.01614500	0.02019170	0.02200230

Tabla 4.23 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_2$ para CR_3.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal ECG real (record 100) [24]			
N	512			
M	256			
CR	2			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00261350	0.00439140	0.00616377	0.00676680

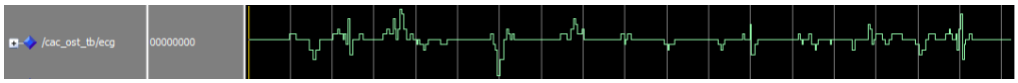


Figura 4.74 Señal $\mathbf{f}_3$ cuantificada.

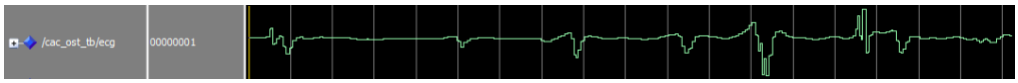


Figura 4.75 Señal $\mathbf{f}_4$ cuantificada.

términos de frecuencia máxima de operación, la arquitectura digital CAC-SS es dieciocho por ciento y novecientos sesenta por ciento más rápida que la arquitectura digital [13] y la arquitectura digital CAC-OST respectivamente.

La arquitectura digital CAC-SS con razón de compresión de dos presenta una reducción de cuatro por ciento en elementos lógicos y dos por ciento en registros, en comparación con la arquitectura digital [13]. También presenta una reducción de dieciséis por ciento en elementos lógicos y treinta por ciento en registros, en comparación la arquitectura digital CAC-OST. En términos de frecuencia máxima de operación, la arquitectura digital CAC-SS es veintisiete por ciento y un mil seiscientos cincuenta y cuatro por ciento más rápida que la arquitectura digital [13] y la arquitectura digital CAC-OST respectivamente.

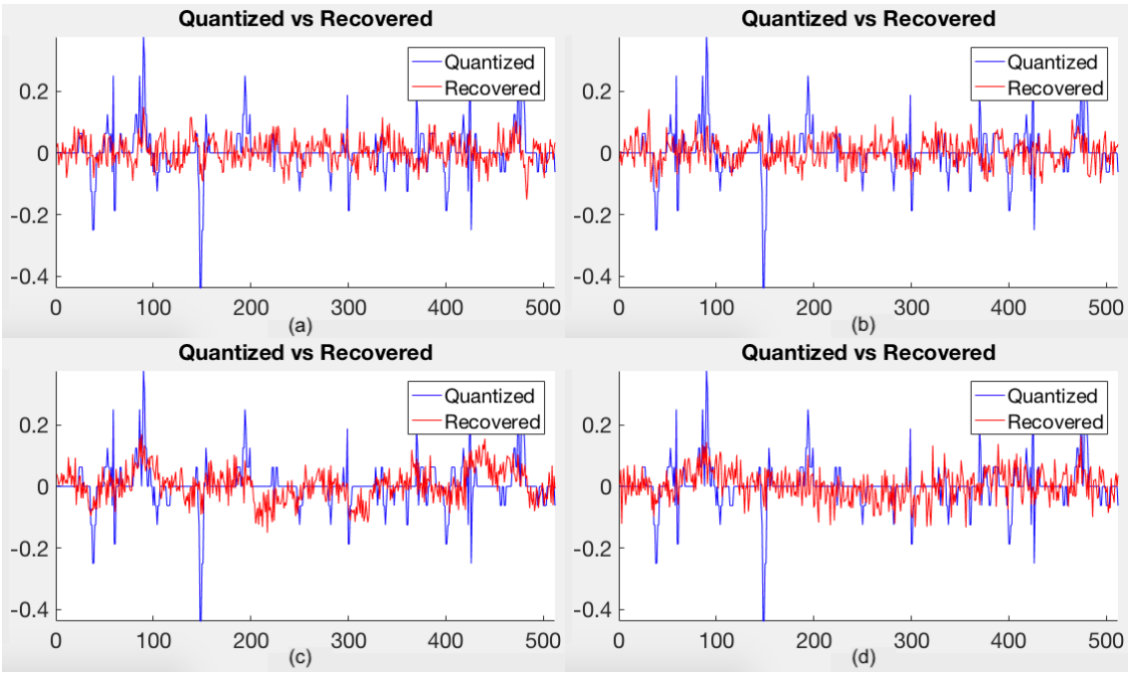


Figura 4.76 Señal EMG real $\mathbf{f}_3$ [24] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

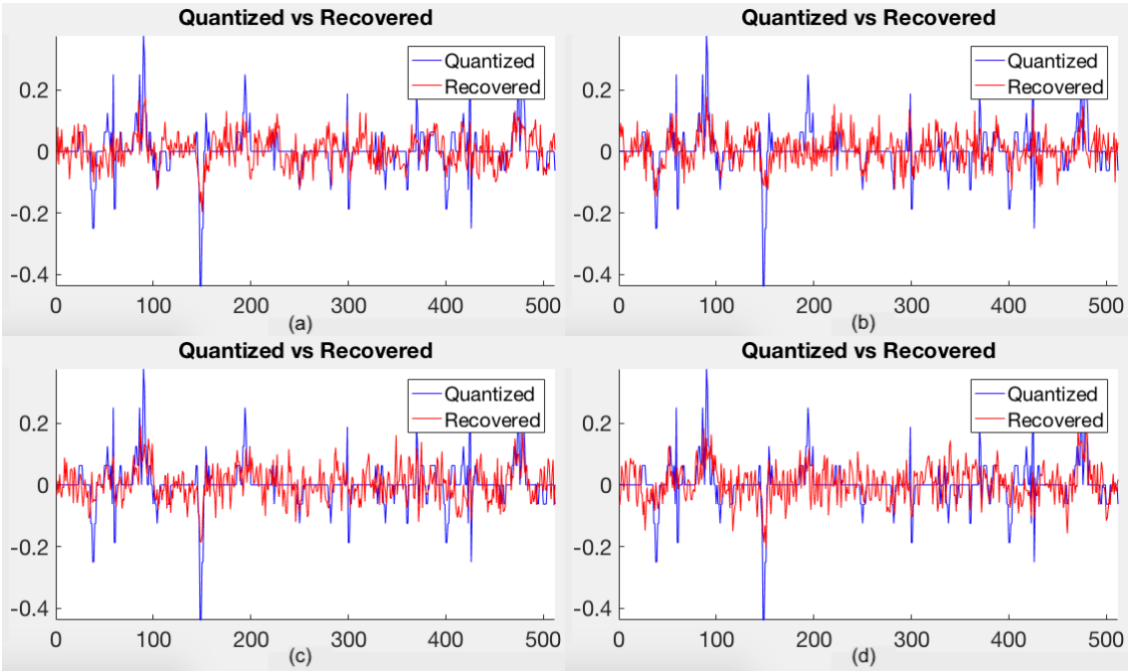


Figura 4.77 Señal EMG real $\mathbf{f}_3$ [24] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

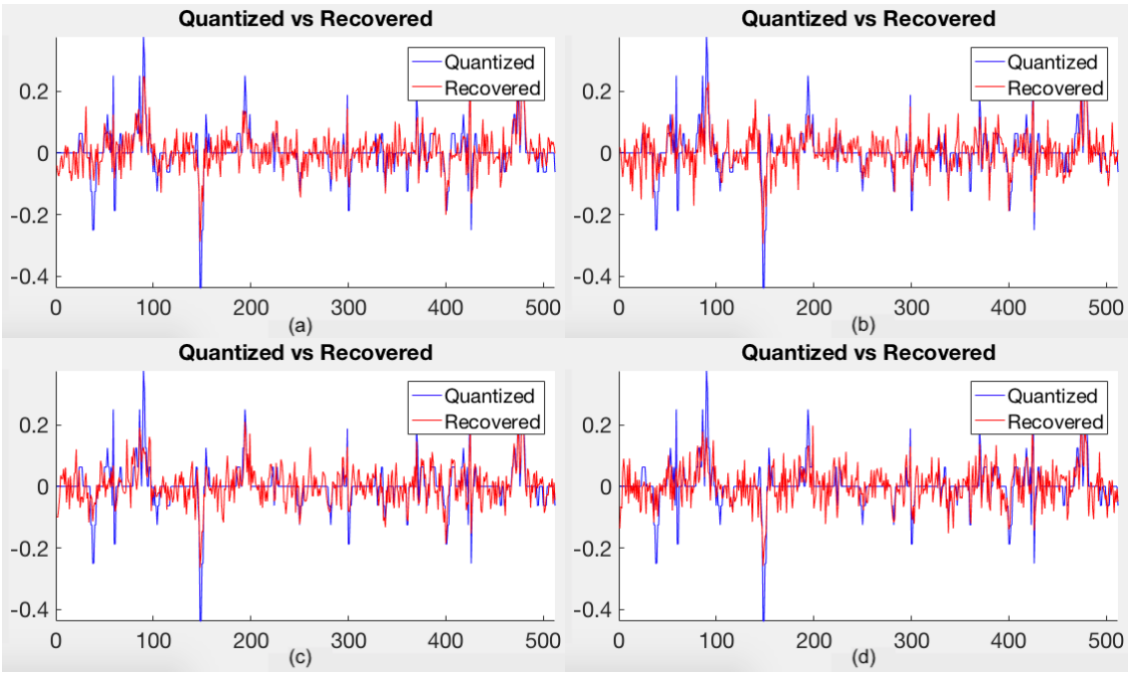


Figura 4.78 Señal EMG real $\mathbf{f}_3$ [24] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

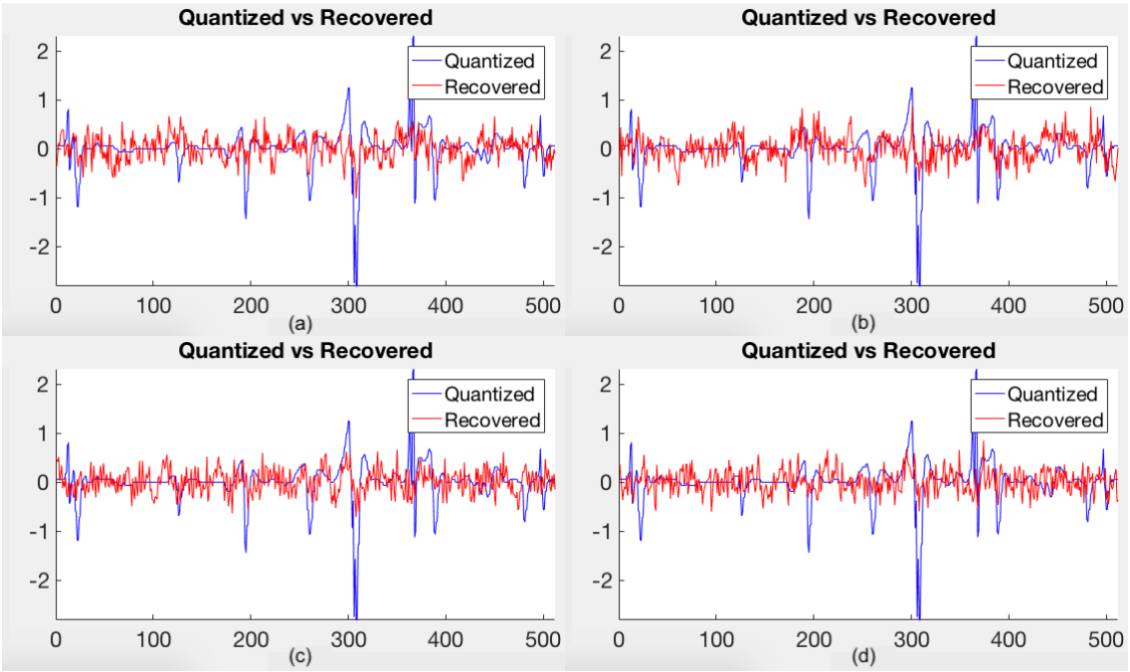


Figura 4.79 Señal EMG real $\mathbf{f}_4$ [24] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

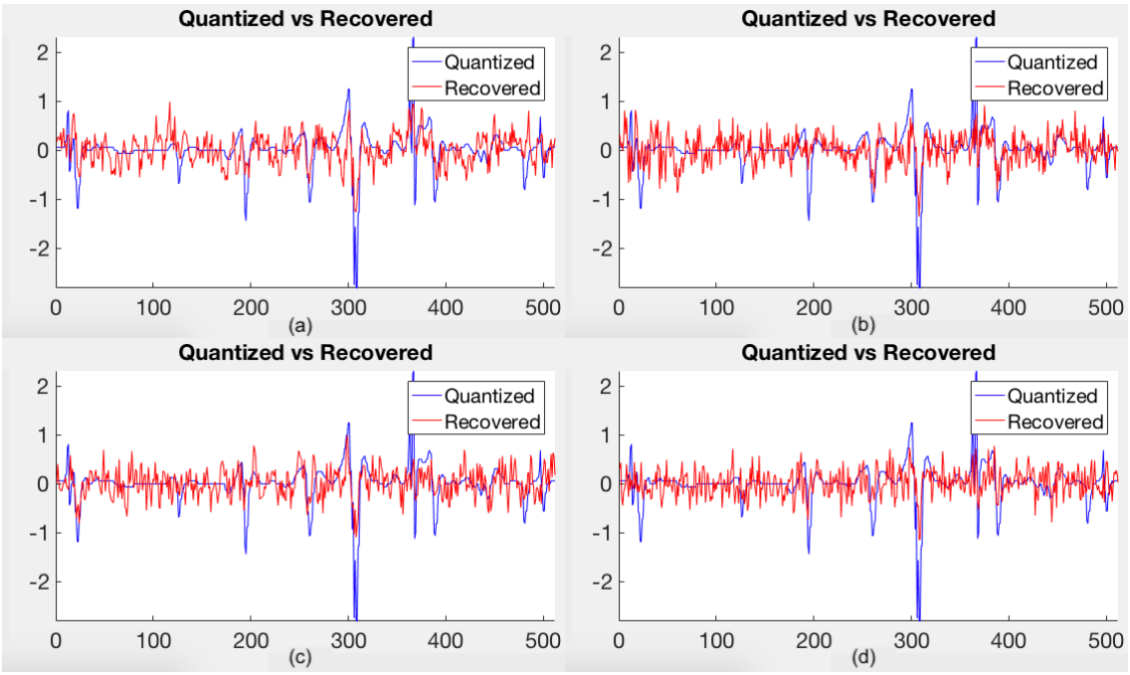


Figura 4.80 Señal EMG real $\mathbf{f}_4$ [24] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

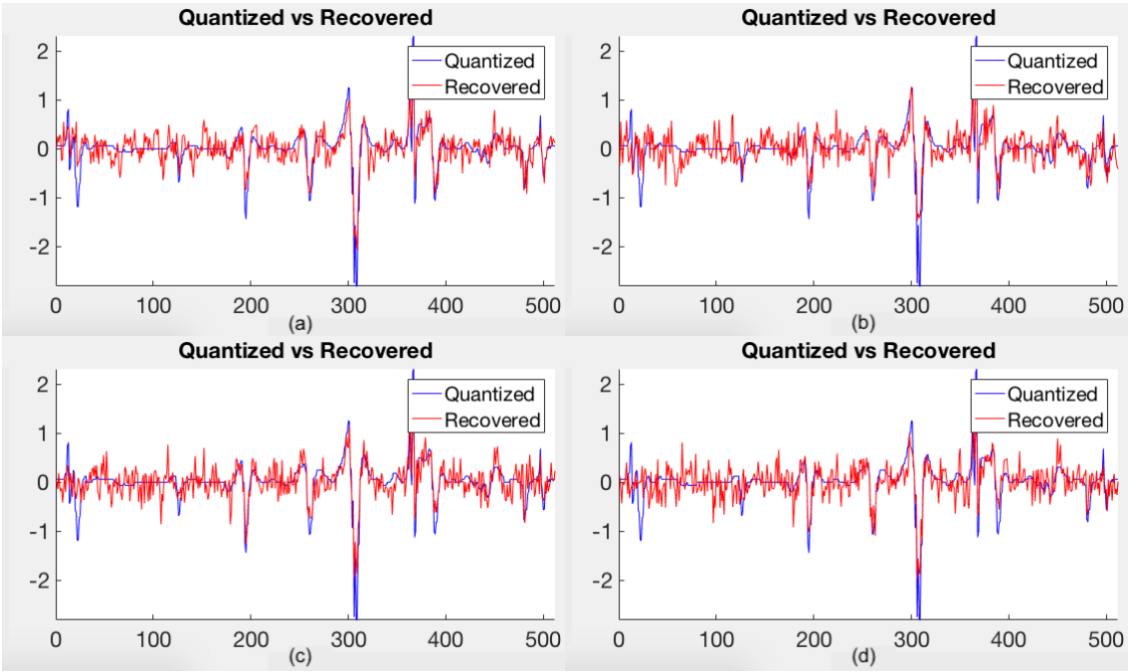


Figura 4.81 Señal EMG real $\mathbf{f}_4$ [24] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$

Tabla 4.24 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_3$ para CR_1.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal EMG real (*Myopathym*) [24]			
N	512			
M	64			
CR	8			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00608345	0.00599045	0.00571370	0.00585373

Tabla 4.25 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_3$ para CR_2.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal EMG real (*Myopathym*) [24]			
N	512			
M	64			
CR	8			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00466046	0.00513711	0.00470744	0.00536224

Tabla 4.26 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_3$ para CR_3.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal EMG real (*Myopathym*) [24]			
N	512			
M	64			
CR	8			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00288358	0.00345846	0.00281018	0.00353565

Tabla 4.27 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_4$ para CR_1.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal EMG real (*Neuropathym*) [24]			
N	512			
M	64			
CR	8			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.19843300	0.20431400	0.18244800	0.21043600

Tabla 4.28 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_4$ para CR_2.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
$\mathbf{f}$	Señal EMG real (*Neuropathym*) [24]			
N	512			
M	64			
CR	8			
$\mathbf{\Phi}$	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
$\mathbf{\Psi}$	DCT	WHT	DCT	WHT
MSE	0.162417	0.16627800	0.14863500	0.16675900

Tabla 4.29 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_4$ para CR_3.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
$\mathbf{f}$	Señal EMG real (*Neuropathym*) [24]			
N	512			
M	64			
CR	8			
$\mathbf{\Phi}$	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
$\mathbf{\Psi}$	DCT	WHT	DCT	WHT
MSE	0.0801905	0.0971769	0.0712806	0.09670100

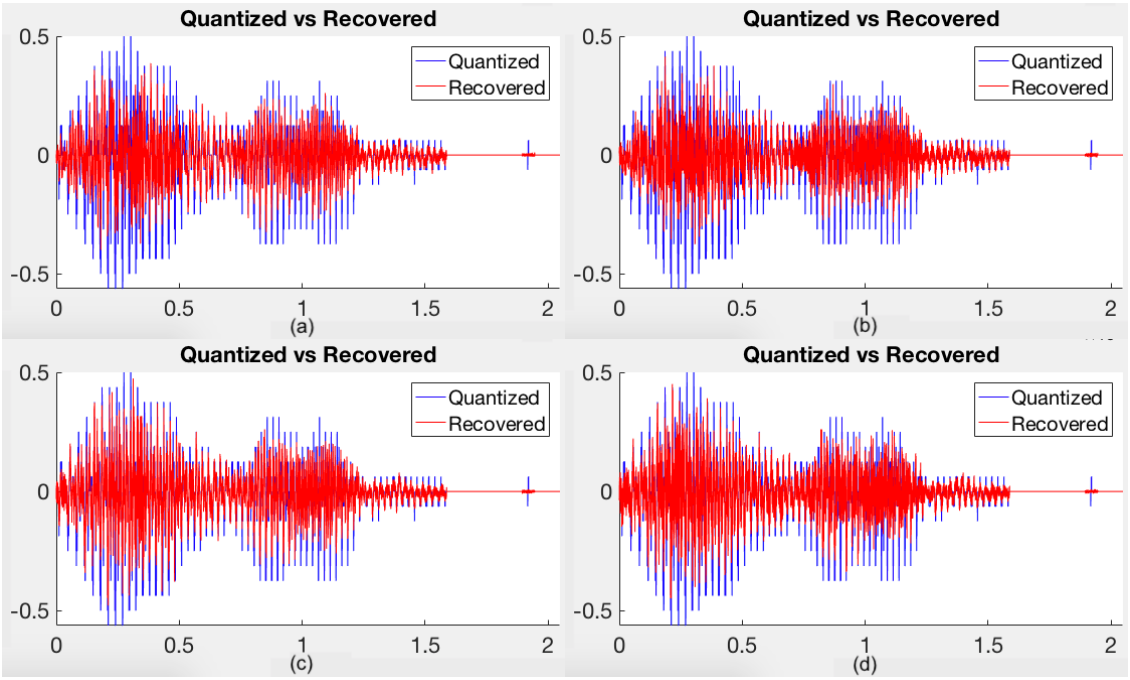


Figura 4.82 Señal de audio $\mathbf{f}_5$ [25] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

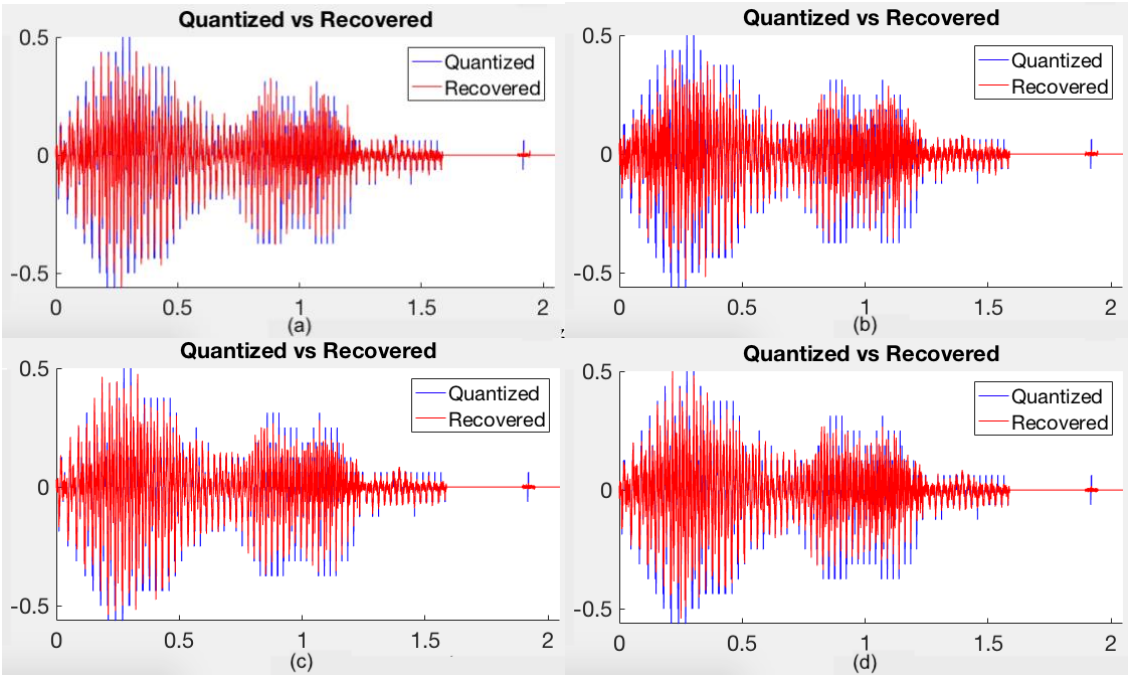


Figura 4.83 Señal de audio $\mathbf{f}_5$ [25] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

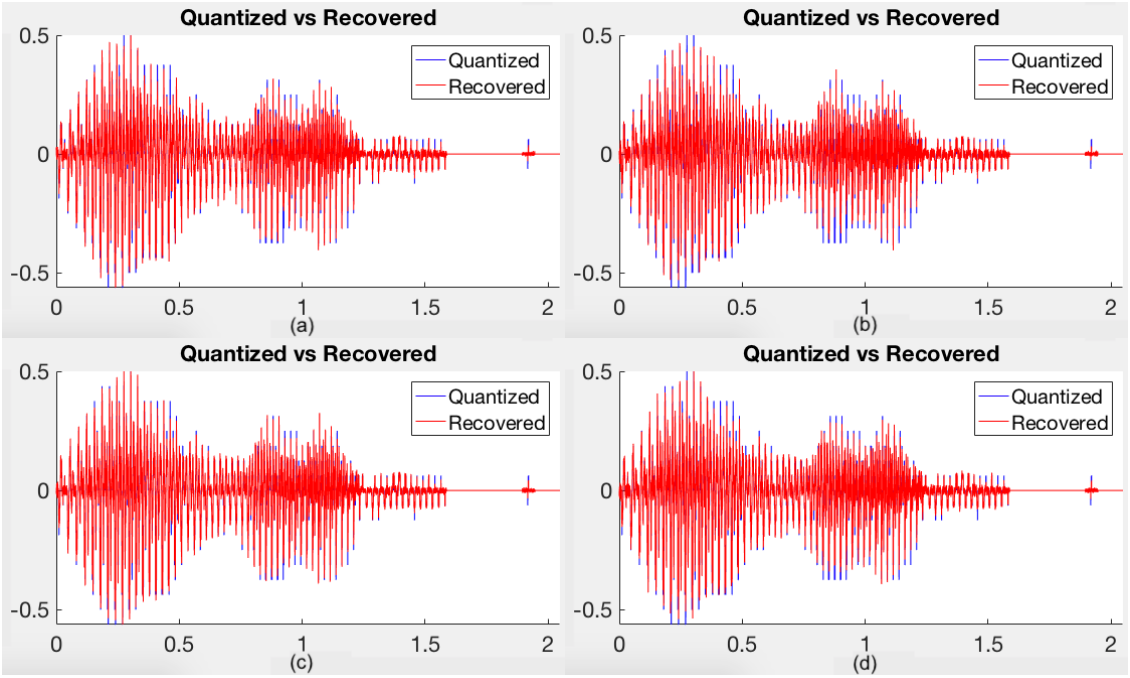


Figura 4.84 Señal de audio $\mathbf{f}_5$ [25] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

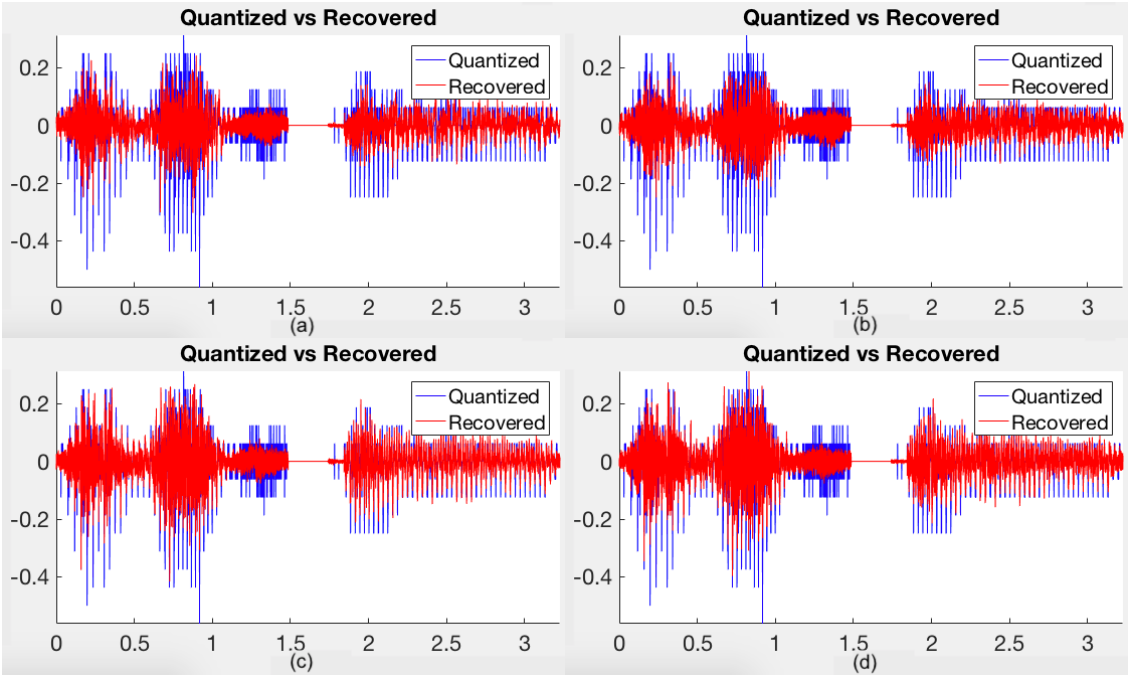


Figura 4.85 Señal de audio $\mathbf{f}_6$ [25] *vs* señal recuperada con CR_1 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

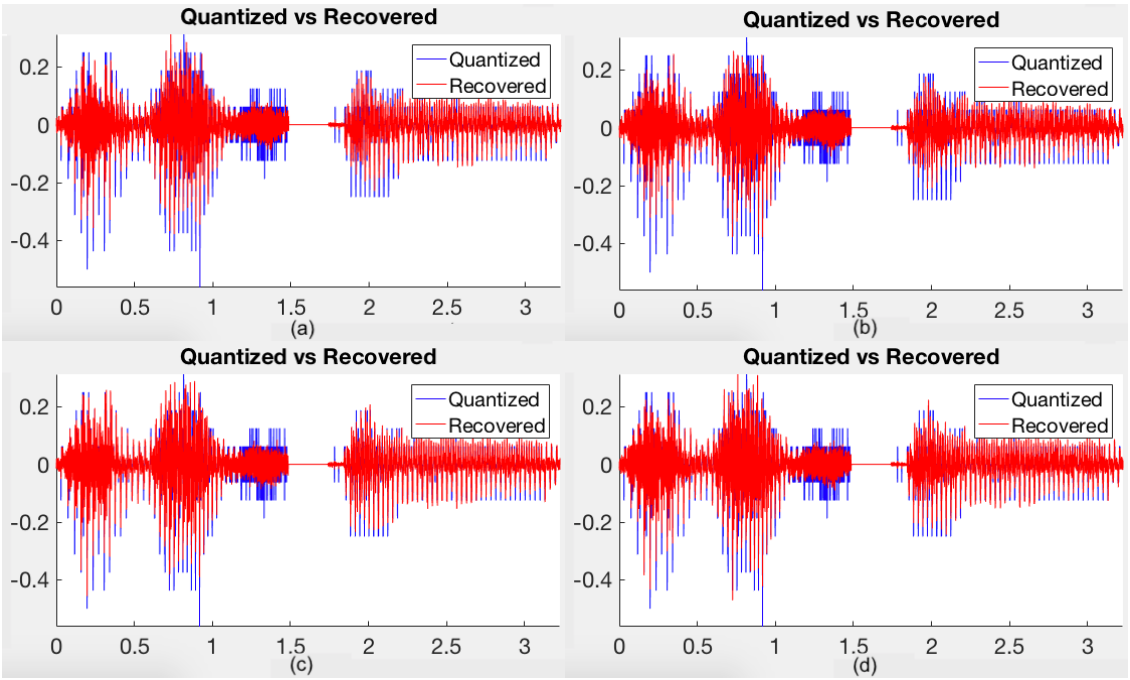


Figura 4.86 Señal de audio $\mathbf{f}_6$ [25] *vs* señal recuperada con CR_2 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

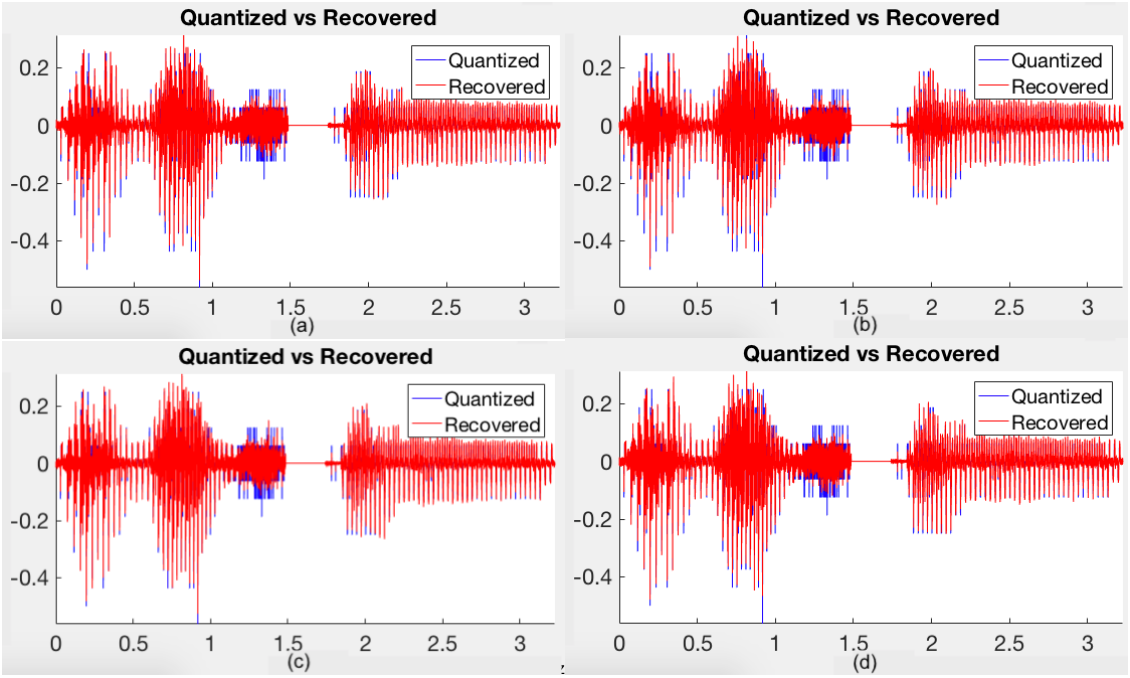


Figura 4.87 Señal de audio $\mathbf{f}_6$ [25] *vs* señal recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$, (b) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_2$, (c) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$ y (d) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_2$.

Tabla 4.30 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_5$ para CR_1.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal de audio brilliant.wav [25]			
CR	8			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00483358	0.00663967	0.00818501	0.00865711

Tabla 4.31 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_5$ para CR_2.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal de audio brilliant.wav [25]			
CR	4			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00135251	0.00270277	0.00175917	0.00411680

Tabla 4.32 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_5$ para CR_3.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal de audio brilliant.wav [25]			
CR	2			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00031415	0.00069262	0.00046047	0.00097873

Tabla 4.33 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_6$ para CR_1.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal de audio arresting.wav [25]			
CR	8			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00231598	0.00288067	0.00385938	0.00423320

Tabla 4.34 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_6$ para CR_2.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal de audio arresting.wav [25]			
CR	4			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00081671	0.00135278	0.00143918	0.00216972

Tabla 4.35 Comparación de tasa de error en la recuperación de la señal $\mathbf{f}_6$ para CR_3.

Muestreo compreso	Arquitectura digital CAC-SS		Arquitectura digital CAC-OST	
f	Señal de audio arresting.wav [25]			
CR	2			
Φ	CAC-SS		CAC-OST	
Recuperación	Optimización l_1 [23]			
Ψ	DCT	WHT	DCT	WHT
MSE	0.00024240	0.00042258	0.00031718	0.00057418

Figura 4.88 Lena.tiff

Figura 4.89 Señal de entrada $\mathbf{f}_7$.

(a) (b)

Figura 4.90 Señal $\mathbf{f}_7$ recuperada con CR_3 y: (a) $\mathbf{\Phi}_3$ y $\mathbf{\Psi}_1$ y (b) $\mathbf{\Phi}_4$ y $\mathbf{\Psi}_1$.

Tabla 4.36 Comparación de tasa de error, proporción máxima de señal a ruido y relación de señal a ruido en la recuperación de la señal $\mathbf{f}_7$ para CR_3.

Muestreo compreso	Arquitectura digital CAC-SS	Arquitectura digital CAC-OST
f	Imagen en escala de grises y tamaño de 128x128 pixeles	
CR	2	
Φ	CAC-SS	CAC-OST
Recuperación	Optimización l_1 [23]	
Ψ	DCT	
MSE	133.9645	76.9713
$PSNR\ (dB)$	26.8609	29.2675
$SNR\ (dB)$	21.1814	23.5663

5 Conclusiones

Este capítulo presenta las conclusiones de la presente investigación. La sección 5.1 presenta las conclusiones generales del diseño e implementación de las arquitecturas digital CAC-OST y CAC-SS en el chip FPGA. La sección 5.2 presenta el trabajo a futuro que permitirá darle continuidad a la presente investigación.

5.1 Conclusiones

En términos generales, la arquitectura digital propuesta, denominada CAC-SS (*Cellular Automata Chaos for Streaming Signals*), es más rápida y utiliza menos hardware que la arquitectura digital del estado del arte, reportada en [13], y que la arquitectura digital propuesta para el algoritmo CAC-OST (*Cellular Automata Chaos with Original Signal Thresholding*) el cual fue recientemente propuesto en [21]. Además, las pruebas revelan que la arquitectura digital CAC-SS presenta resultados competitivos en la recuperación de señales electrocardiográficas, particularmente en la detección de arritmias (trastorno del ritmo cardiaco) porque en esos casos es esencial recuperar los picos de gran amplitud. Muestreo compreso en general es muy adecuado para recuperar componentes de la señal con mayor energía, en la que la condición de arritmia representa esos picos de gran amplitud.

Los objetivos propuestos en la sección 1.6 han sido logrados. El primer objetivo específico: "Construir una arquitectura regular" se logró siguiendo las reglas de la metodología de diseño digital síncrono. La regla primordial de diseño digital síncrono es que sólo se utilice un reloj en toda la arquitectura. Así, las arquitecturas propuestas preservan módulos simples y regulares para una operación eficiente sincronizada por una señal de reloj. El segundo objetivo específico: "Realizar una implementación computacionalmente eficiente" se logró disminuyendo los recursos de *hardware* y aumentando la frecuencia de operación máxima para comprimir la señal, mediante la reducción de complejidad de la circuitería asociada con las funciones lógicas combinacionales. El tercer objetivo específico: "Simplificar la unidad de control" se logró diseñando la arquitectura lo más sencilla posible y eliminando bloques innecesarios (memoria y multiplicadores). Logrando estos objetivos específicos, se ha logrado el objetivo general: "Diseñar una arquitectura de bajo consumo de potencia, baja utilización de recursos de *hardware* y alta velocidad de operación para muestreo compreso con base en matrices estructuradas."

En la sección 1.5 se planteó la siguiente pregunta de investigación: ¿Cómo diseñar una arquitectura de muestreo compreso que pueda procesar una señal tipo *streaming* con una implementación eficiente y que permita reducir el número de muestras para reconstruir la señal? La arquitectura digital CAC-SS responde de manera asertiva a dicha pregunta de la siguiente

manera:

1. La forma en cómo se diseñó la arquitectura de muestreo compreso para procesar una señal tipo *streaming* fue generando los coeficientes de la matriz de muestreo **Φ** mediante la regla clásica de autómata celular [22] por columna y no por fila. La generación de los coeficientes por columna permite que cuando entre una muestra de la señal, esta pueda ser multiplicada y acumulada. Una vez que entra la siguiente muestra, también puede ser multiplicada y acumulada por su respectiva columna, y así sucesivamente.
2. La implementación eficiente se demuestra con la Tabla 4.14. En dicha tabla se comparan los resultados obtenidos de las arquitecturas digitales propuestas CAC-OST y CAC-SS contra el estado del arte. Los resultados de la arquitectura digital CAC-SS son más eficientes, es decir, la arquitectura digital CAC-SS utiliza menos área (recursos de *hardware*) y es más rápida ya que su frecuencia máxima de operación es mayor.
3. La reducción en el número de muestras para reconstruir una señal se demuestra en la Tabla 4.16 y Tabla 4.17. En dichas tablas se comparan la tasa de error resultante cuando se comparan la señal original contra señal recuperada para razones de compresión de 4 y 2 respectivamente. Cuando la razón de compresión es de 4, es decir, sólo se requiere la cuarta parte de muestras de la señal, el mejor resultado lo presenta la arquitectura digital CAC-OST. Cuando la razón de compresión es de 2, es decir, sólo se requiere la mitad de las muestras de la señal, el mejor resultado lo presenta la arquitectura digital CAC-SS.

Finalmente, se han propuesto técnicas de baja complejidad computacional para codificación digital de señales electrocardiográficas para su eficiente transmisión, las cuales contribuyen al estado del arte en la emergente teoría de adquisición y procesamiento de información conocida como muestreo compreso.

5.2 Trabajo a futuro

La presente investigación permite sentar las bases en sensores inalámbricos de nueva generación para el monitoreo de la frecuencia cardiaca. La compresión y recuperación de señales electrocardiográficas mediante muestreo compreso permite el desarrollo de estos nuevos sensores. No obstante, la detección de características de una señal mediante muestreo compreso requiere que la señal sea dispersa o fácilmente compresible, entonces, en general, no todos los trastornos cardiacos podrían ser detectados mediante muestreo compreso si la señal deja de ser dispersa. Por lo anterior se recomienda que las arquitecturas propuestas sean utilizadas para padecimientos o trastornos como arritmia. Se recomienda continuar trabajando en los siguientes puntos:

- Elaborar el *layout* de la arquitectura digital CAC-SS. Esto permitirá la fabricación del chip y la realización de pruebas con pacientes.
- Realizar pruebas de recuperación utilizando como matriz de dominio o base Wavelet (Haar y Daubechies). Validar los resultados y comparar contra las bases o dominios utilizados en esta investigación.
- Diseñar nuevas matrices de muestreo basadas en filtros. Esto permitirá poder detectar otro

tipo de trastornos cardiacos.

- Diseñar la arquitectura digital para la recuperación de la señal. Esto permitirá crear aplicaciones en donde se le pueda avisar al paciente, mediante el celular, que está teniendo alguna condición o trastorno cardiaco particular.

Muestreo compreso es una teoría emergente, la cual tiene un gran campo de aplicaciones. Una de las ramas en donde se podría dar continuidad a la investigación es en señales de video. Por lo anterior, se sugiere diseñar una arquitectura digital basada en las matrices de muestreo propuestas que permitan trabajar con este tipo de señales. El desarrollo de dicha arquitectura podría implementarse como un nuevo formato de compresión superior a los que actualmente existen y se podrían probar en drones de nueva generación.

Productos derivados de la investigación

La investigación permitió desarrollar las siguientes publicaciones:

- S. D. A. Vazquez, D. E. T. Romero and M. S. Ramírez, "Digital architectures based on chaotic cellular automata for compressed sensing of electrocardiographic (ECG) signals," *20th IEEE Autumn Meeting on Power, Electronics and Computing*. ROPEC 2018. Ixtapa, Guerrero, México, November.
  BEST PAPER AWARD: ELECTRONICS TRACK.

- S. D. A. Vazquez and D. E. T. Romero, "Compressed sensing apps developed in Matlab®," September, 2018. [Online]. Available: http://www.sadavasc.com.mx/compressed-sensing/. [Accessed: 11-Dec-2018].

Referencias

[1] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, 4th Ed., New Jersey: Prentice Hall, 2007.

[2] H. Nyquist, "Certain factors affecting telegraph speed," *The Bell System Technical Journal*, vol. 3, no. 2, pp. 324 – 346, April 1924.

[3] H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the American Institute of Electrical Engineers* , vol. 47, no. 2, pp. 617 – 644, 1928.

[4] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379 – 423, July 1948.

[5] D. L. Donoho, "Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289 – 1306, April 2006.

[6] A. V. Oppenheim, A. S. Willsky and S. H. Nawab, *Signals & Systems*, 2nd Ed., Prentice Hall, 1997.

[7] E. J. Candés, J. Romberg and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489 – 509, February 2006.

[8] D. L. Donoho, M. Elad and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6 – 18, January 2006.

[9] J. A. Tropp, "Just relax: convex programming methods for identifying sparse signals in noise," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1030 – 1051, March 2006.

[10] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118 – 121, July 2007.

[11] E. J. Candés and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21 – 30, March 2008.

[12] R. Karthik, "ECG simulation using Matlab," April 2006. [Online]. Available: https://la.mathworks.com/matlabcentral/fileexchange/10858-ecg-simulation-using-matlab. [Accessed: 11-Dec-2018].

[13] F. Chen, A. P. Chandrakasan and V. M. Stojanović, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 744 – 756, March 2012.

[14] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, 4th Ed., Springer, 2014.

[15] A. M. R. Dixon, E. G. Allstot, D. Gangopadhyay and D. J. Allstot, "Compressed sensing system considerations for ECG and EMG wireless biosensors," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 156 – 166, April 2012.

[16] R. Rieger and J. T. Taylor, "An adaptive sampling system for sensor nodes in body area networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17, no. 2, pp. 183 – 189, April 2009.

[17] R. F. Yazcioglu, S. Kim, T. Torfs, H. Kim and C. Van Hoof, "A $30\mu W$ analog signal processor ASIC for portable biopotential signal monitoring," *IEEE Journal of Solid-State*

Circuits, vol. 46, no. 1, pp. 209 – 223, January 2011.
[18] Z. Zhang, T. P. Jung, S. Makeig and D. Rao, "Compressed sensing of EEG for wireless telemonitoring with low energy consumption and inexpensive hardware," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 1, pp. 221 – 224, 2013.
[19] Z. Zhang and B. D. Rao, "Extension of SBL algorithms for the recovery of block sparse signal with intra-block correlation," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2009 – 2015, April 2013.
[20] N. Attarmoghaddam, S. H. Khayat and M. Maymandi-Nejad, "FPGA implementation for power and area afficient compressed sensing for multi-channel ECG compression," in *24th Iranian Conference on Electrical Engineering (ICEE)*, pp. 1027 – 1032, 2016.
[21] J. Liu and Q. Sun, "Chaotic cellular automaton for generating measurement matrix used in CS coding," *IET Signal Processing*, vol. 11, no. 1, pp. 115 – 122, 2017.
[22] S. Wolfram, "Random sequence generation by cellular automata," *Advances in Applied Mathematics*, pp. 123 – 169, 1986.
[23] J. Candés and J. Romberg, " l_1 -magic: recovery of sparse signals via convex programming," October 2005. [Online]. Available: https://statweb.stanford.edu/~candes/l1magic/. [Accessed: 11-Dec-2018].
[24] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng and H. E. Stanley, "PhysioBank, PhysioToolkit and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. 215 – 220, June 2000.
[25] C. Becker, "Audio samples," 2001. [Online]. Available: https://sampleswap.org/index.php. [Accessed: 11-Dec-2018].
[26] M. Wakin, ""Standard test image," May 2013. [Online]. Available: https://www.ece.rice.edu/~wakin/images/ [Accessed: 11-Dec-2018].