

고 신뢰성 Audio Target 주파수 발생장치에 대한 연구[☆]

A Study on the High Reliability Audio Target Frequency Generator for Industry and Military

박 창 식¹ 한 은 택¹ 김 익 재¹ 신 동 규^{1,2,3*}
Changsik Park Euntaek Han Ikjae Kim Dong-Kyoo Shin

요 약

주파수 합성기는 입력된 주파수 신호를 조작하여 필요한 주파수를 생성하는 단순한 기능을 수행하지만 통신, 제어, 감시, 의료 및 군사 분야 등의 기계장비에서 신뢰성 있는 동작을 위해서는 안정적이고 정밀한 주파수 생성이 필수적이다. 다양한 분야에서 주파수 합성은 아날로그 방식과 디지털 방식 또는 이를 혼용한 하이브리드 방식이 사용되어 왔으며, 특히 통신 분야에서는 저주파 AF(Audio Frequency)부터 고주파 마이크로파까지 각 주파수 대역에서 필요한 정밀한 주파수 합성기를 사용하고 있다. 본 논문은 AF 주파수를 사용하는 철도 궤도회로 시스템에 적용하기 위한 고도의 신뢰성이 보장된 주파수 합성기를 마이크로프로세서 사용 없이 FPGA(Field Programmable Gate Array)의 로직 회로만으로 설계하고 구현하였다. 실험결과 99.999%이상의 정확도로 Target 주파수를 발생시키는 성능을 나타내었다. 이러한 고도의 신뢰성을 갖는 AF급 주파수 합성기는 철도, 지하철 등의 교통운송 수단에 사용될 때 제동 및 신호 시스템의 안전한 운영과 계획된 운행의 오차가 줄어들어 이를 이용하는 시민의 대기시간 감소와 편의성을 높이는 데 중요한 역할을 할 것으로 판단된다.

☞ 주제어 : 주파수 합성기, 직접 주파수 합성기, 간접 주파수 합성기, 철도 궤도회로

ABSTRACT

The frequency synthesizer performs a simple function of generating the necessary frequency by manipulating the input frequency signal, but stable and precise frequency generation is essential for reliable operation in mechanical equipment such as communication, control, surveillance, medical and military fields. In various fields, analog and digital methods or hybrid methods have been used for frequency synthesis. Especially in the field of communication, a precise frequency synthesizer required for each frequency band from low frequency AF (Audio Frequency) to high frequency microwave is used. In this paper, a highly reliable frequency synthesizer is designed and implemented using FPGA(Field Programmable Gate Array) without using a microprocessor for application to a railway track circuit system using AF frequency. As a result of the experiment, it showed the performance of generating the target frequency with an accuracy of more than 99.999%. The AF-class frequency synthesizer with such high reliability is expected to play an important role in enhancing convenience, such as reducing the waiting time of citizens who use braking and signaling systems and reducing errors in planned operation when used in transportation such as railways and subways.

☞ keyword : Frequency Synthesizer, Direct Frequency Synthesizer, Indirect Frequency Synthesizer, Railway Track Circuit

1. 서 론

오늘날의 무선통신 기술은 RF(Radio Frequency) 기술

의 발전과 밀접한 연관이 있다. 무선통신은 RF 기술을 사용하여 데이터를 전송하며, RF 기술은 무선 신호를 생성, 변조, 송수신, 감지, 처리 등 다양한 작업을 수행하기 때문이다. RF를 이용한 통신에서 아날로그 방식과 디지털 방식이 있는데 공통적인 중요 요소 중 하나는 편차 없이 일정하고 정확하게 필요한 주파수를 생성하는 것이다. 이를 위해 온도 등의 주변 환경 변화에 잘 변하지 않는 수정 같은 온도계수가 낮은 재료로 발진기를 만들고 있으나, 이를 통신에 사용하기 위해서는 상황에 따라 주파수를 변동시킬 수 있어야 한다. 이에 주파수 가변이 가능하고 환경 변화에 안정적인 주파수를 얻고자 많은 연구가 진행되고 있다.

¹ Department of Computer Engineering, Sejong University, Seoul, 05006, Korea.

² Department of Convergence Engineering for Intelligent Drones, Sejong University, Seoul, 05006, Korea

³ Cyber Warfare Institute, Sejong University, Seoul, 05006, Korea

* Corresponding author (shindk@sejong.ac.kr)

[Received 24 February 2023, Reviewed 22 March 2023(R2 17 April 2023), Accepted 25 April 2023]

[☆] 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2022R1F1A1074773).

안정적인 주파수원에 대한 수요는 의료 전자기기, 건설 측정장비, 레이더, 로켓, 자율주행 차량, 공장 자동화 로봇, 철도, 항공 등 통신 분야뿐만 아니라 의료, 계측, 군사, 산업 및 기간산업을 포함하여 전자·컴퓨터·통신이 적용되는 거의 모든 분야에서 활용되고 있다. 주파수를 조합하여 만든다는 의미의 주파수 합성기(Frequency Synthesizer)는, 아날로그 방식으로는 PLL(Phase-Locked Loop)이라는 개념이 1930년대부터 개발되어 왔으나 당시 반도체 기술이 발달되지 않은 때여서 진공관이나 바이폴라 소자만으로 구현하는데 어려움이 많았다. 1950년대부터 개발이 시작된 집적회로(IC)의 집적도가 급격히 커지면서 DDFS(Direct Digital Frequency Synthesizer)로 발전되어왔다 [1].

디지털 방식인 DDFS는 1971년에 J.Tiemey가 제안하였고 반도체 집적기술이 크게 발전하는 1980년대 이후로 무선기기에 널리 사용되면서 크게 발전하였다 [2]. PLL과 DDFS 또는 DDS는 서로의 장단점을 바탕으로 분야별 요구사항에 따라 발전되어 왔으며 이 둘을 혼합한 하이브리드 방식도 이용되고 있다 [3][4][5][6][7][11].

FS(Frequency Synthesizer), 주파수 합성기는 무선통신이나 군용 감시·추적 장비 등에 사용되는 마이크로파를 이용한 FS부터, 전자제어 감시기기나 철도 등 교통시스템에서 사용하는 오디오 급의 낮은 FS까지 필요에 따라 다양하게 존재한다. 그러나 FS의 목적은 기준 주파수를 이용하여 다른 주파수를 안정되게 생성해 내는 것은 동일하다. 주파수의 정확도, 스펙트럼 순도와 튜닝 또는 스위칭 속도 등도 중요한 성능지표이다. 마이크로파 같은 높은 FS는 구현하는 방식에 따라서 발생하는 불필요한 주파수가 발생하는 잡음인 스퓨리어스 잡음이 성능을 저해하는 요인이 될 수도 있다. 대량 생산되는 스마트폰처럼 응용되는 제품에 따라서 FS를 구현하는 비용도 문제가 되므로 가급적 원가 비중을 낮추기 위해 부품 선택에 경제성을 고려하기도 한다.

반도체 집적도의 눈부신 발전으로 인한 CPLD(Complex Programmable Logic Device)와 FPGA 등의 출현으로 FS의 성능이 크게 향상되었으며, 최근에는 용도에 따라 다양한 어플리케이션을 적용할 수 있는 환경이 조성되었다. 본 논문에서는 CPU를 배제한 FPGA 기반의 고품질 산업용 FS를 제안한다.

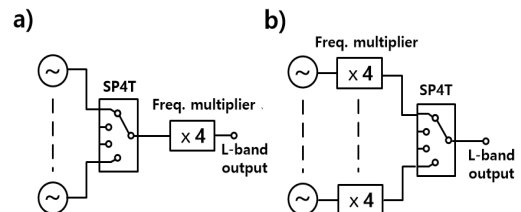
2. 관련연구 및 기술동향

전자통신기기 등에서 필수적인 주파수 합성기를 만드는 방법으로는 직접 주파수 합성(Direct FS), 간접 주파수 합성(Indirect FS)로 구분 할 수 있으며, 직접주파수 합성

방식은 DAS(Direct Analog Synthesizer)인 직접 아날로그 합성기와 DDFS(Direct Digital Frequency Synthesizer)처럼 직접 디지털 합성기가 있고, 간접주파수 합성방식에는 PLL(Phase-Locked Loop)의 위상고정루프와 DLL(Delay-Locked Loop)인 지연고정루프가 있다 [6]. 이 외에도 두 가지 방법을 혼합한 하이브리드 방식의 FS가 사용되고 있다. 주파수 생성의 아날로그 방식은 정확성은 높으나 제어가 어려운 반면, 디지털 방식은 제어는 간단하나 정확성이 떨어지는 특징이 있다.

2.1 아날로그 주파수 합성기

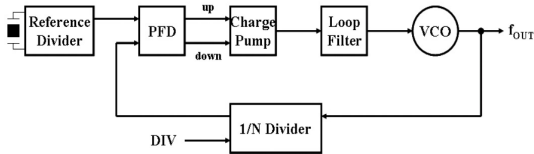
아날로그 주파수 합성기도 주파수 생성 방식에 따라 직접방식과 간접방식으로 구분할 수 있다. 직접방식은 주파수 원으로부터 직접 주파수를 생성하는 방식이고, 간접방식은 이전에 생성된 주파수를 변조하여 원하는 주파수를 생성하는 방식이다. 직접 아날로그 주파수 합성기는 기준 주파수 생성기, 혼합기, 주파수 상향/하향 변환기(Up/Down converter), 주파수 체배기(Frequency Doubler/Multiplier) 등을 결합하여 원하는 주파수를 생성하는 방식으로, 주파수 원 자체를 응용하는 방식이다. 간접 아날로그 주파수 합성기는 단순한 반전, 주파수 분할 등의 기법을 이용하여 원하는 주파수를 생성하는 방식으로, 다른 주파수 원을 상승시키는 등의 과정을 거치지 않는다. Andrzej Rokita [8]은 직접 아날로그 합성에서 수행되는 곱셈, 믹싱, 필터링 및 분할과 같은 연산에 의해 새로운 주파수가 생성될 때 발생하는 위상잡음을 저감시키는 PLL(Phase-Locked Loop) 설계로 매우 빠른 스위칭은 직접 아날로그 방식의 장점이라고 기술하고 있다. 그림 1은 [8]의 아날로그 직접 주파수 합성기의 개념도를 보여주고 있다. 각 모듈은 SP4T(Single Pole 4 Transfer) 스위치로 선택된 4개의 발전기중 하나의 신호로 구동되는 주파수 4배기(frequency quadrupler)로 공급된다.



(그림 1) 아날로그 직접 주파수 합성기의 구성도
(Figure 1) Block diagram of analog direct frequency synthesizer

이에 비해 간접 주파수 합성기는 널리 알려지고 많이 사용되는 PLL(Phase-Locked Loop) 방식이 있다. PLL(Phase-Locked Loop)은 입력 신호에 대한 VCO(Voltage-Controlled Oscillator)의 출력 신호를 비교하고 VCO의 주파수를 조절하여 입력 신호와 VCO 출력 신호의 위상 차이를 일정하도록 유지하는 기술이다. 간접 주파수 합성기에서는 PLL을 이용하여 원하는 주파수를 생성하는데, 이 때 PLL의 구성 요소들이 주파수 합성기의 성능을 결정한다.

윤광섭 등의 [1]에서 제안된 PLL 시스템은 정수-N PLL의 구성 요소들을 포함하고 있다. Reference Divider에서 생성된 기준 클록은 PFD(Phase-Frequency Detector)에서 VCO 출력 신호와 비교되어 업/다운 신호를 생성하고, CP(Charge Pump)는 이 업/다운 신호를 전류로 변환하여 LF(Loop Filter)로 전달한다. LF는 전류를 전압으로 변환하고 VCO의 주파수를 제어하는데 사용된다. 그림 3의 1/N Divider는 VCO의 출력 신호를 N으로 나누어 최종적으로 원하는 주파수를 생성한다. 더 세밀한 주파수 제어를 가능하게 하기 위해 정수와 분수를 조합하여 사용하는 분수-N(fractional-N) 분주기를 사용하기도 한다.



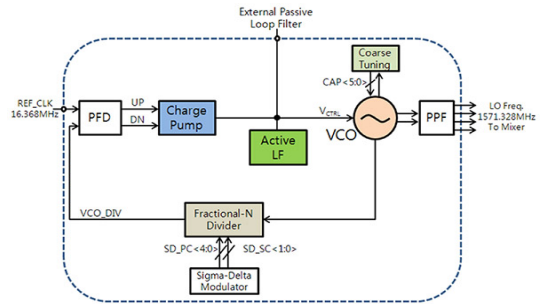
(그림 3) PLL 주파수 합성기의 기본 구조(1)

(Figure 3) Basic structure of PLL frequency synthesizer

이와 같은 PLL 시스템의 장점은 LF 동작으로 인해 스펙트럼 신호레벨이 감소하고 직접 아날로그 주파수 합성기에 비해 간단하다는 것이다. 그러나 직접 아날로그 방식에 비해 주파수 스위칭 시간이 늘어나고 위상잡음이 높다는 것은 단점이라고 할 수 있다. LF 대역폭 안에서 주파수 합성기의 위상잡음 성능은 $\lambda = \lambda_{PFD} + 10\log N$ 으로 표시할 수 있으며 λ_{PFD} 는 위상검출기에 입력되는 기준주파수, 위상검출기, LF 및 피드백 1/N 분주기의 누적 위상잡음이다.

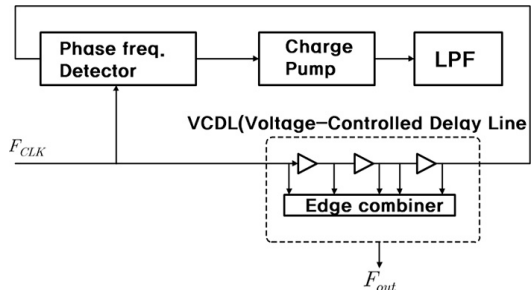
Yuchen Wang, Xuguang Bao, Wei Hua은 [9] PLL의 우수한 위상고정 능력을 이용한 영구자석 동기모터(PMSM)의 회전자 위치를 정확히 알아내는 것에 PLL을 적용하고 있다. 3상 신호의 위상분석은 일반적으로 동기식 기준

시스템을 근간으로 하는데 동기식 레퍼런스 시스템-PLL(SRF-PLL)은 3상 시스템에서 위상, 주파수 및 진폭을 추출하는데 가장 널리 사용되는 기술이다. 이 연구에서는 비대칭 위상쉬프트 신호를 2상 고정 좌표계에 매핑하기 위한 위상전이 PLL을 이용하여 구성한다. PLL을 이용하여 GPS 수신기용 저전력 주파수 신디사이저의 설계에 적용한 김상우 등의 연구 [10]에서 전통적인 프랙셔널-N 분주기를 적용하여 주파수 합성기를 연구하였다. 그림 4는 [10]에서 주파수 합성기의 블록도를 나타낸다. 위상검출기인 PFD, 차지펌프인 CP, 액티브 저역 필터, VCO, 프랙셔널-N 분주기와 시그마-델타 모듈레이터를 사용함을 볼 수 있다.



(그림 4) PLL 주파수 합성기의 블록도

(Figure 4) Block Diagram of a PLL Frequency Synthesizer



(그림 5) DLL 기반의 FS 블록도(6)

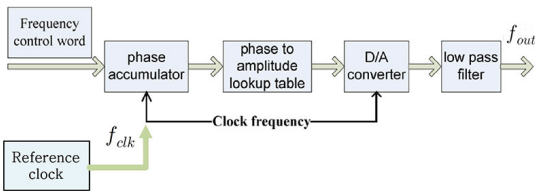
(Figure 5) FS block diagram based on DLL

그림 5는 DLL 기반의 FS 블록도로서 PLL의 VCO 대신에 VCDL(Voltage-Controlled Delay Line)을 가지는 점을 제외하고는 PLL과 매우 유사하여 연구자에 따라서는 PLL의 한 부류로 정의하기도 한다. DLL의 아이디어는 기본적으로 시스템의 클록 신호가 여러 단을 거치면서 필연적으로 발생하는 지연과 관련된 에러를 해결하고자

고안되었다. DLL 시스템은 잡음이 낮고 위상 축적이 없는 장점에도 불구하고 프로그램이 가능하지 않고 제한된 공급인자와 동작 시 발생하는 높은 전력 소모로 인해 일반적으로 FS 적용은 권장되지 않는다 [6].

2.2. 디지털 주파수 합성기

디지털 기술의 발전을 적극 활용한 DDFS는 일반적으로 DDS라고 불리기도 하는데, 이는 직접 디지털 주파수 합성기(Direct Digital Frequency Synthesizer)라고 한다. 그림 6에 간단한 기본 구성도를 보여주고 있다 [11].



(그림 6) Direct Digital Frequency Synthesizer 기본 구성도

(Figure 6) Direct Digital Frequency Synthesizer Basic Diagram

DDS는 기준 클럭 소스(Reference clock), 위상 누산기(Phase Accumulator), 위상진폭변환 및 LUT(phase to amplitude Look-up Table), DAC(Digital to Analog Converter), LPF(Low Pass Filter) 및 출력 주파수를 제어하는 FCW(Frequency Control Word) 부분으로 구성되어 있다.

DDS는 기준 클럭 주파수(f_{clk})에서 최종적으로 출력신호(f_{out})를 만들어낸다. 주로 디지털 제어로 이루어지는 이 과정은 매우 빠르고 직접 아날로그 방식에 비해 높은 스위칭 속도를 제공한다. DDS는 클럭 소스 자체의 위상잡음이 포함되어 있음에도 낮은 위상 잡음특성을 나타낸다.

그림 6에서 주파수 Control Word는 위상 누산기에서 입력되어 계산된 누산기의 합에 추가되어 만들어진 값 X 를 N 비트 가산기에서 구현하여 그 결과를 누산 레지스터에 공급한다. 한편으로는 샘플주소로서 보내진 값이 위상진폭 변환회로에서 상응하는 파형데이터가 어드레스 값에 따라 출력된다. 파형데이터는 D/A 변환기와 LPF로 아날로그 파형으로 변환되어 출력된다.

이와 같은 DDS의 최대 장점은 위상 누산기로 인한 미세한 주파수 분해 능력으로 헬츠(Hz) 레벨의 출력 주파수가 만들어진다. 그러나 사용가능한 대역폭의 제한과

스푸리어스 성능은 단점으로 작용한다. 또한 가장 높은 주파수는 나이퀴스트 정리에 의해서 클럭 주파수의 절반 이내로 한정되고 양자화 및 DAC변환 오류로 인한 높은 스푸리어스 잡음이다.

A.A. Alsharef 등은 [12]에서 그림 6의 전형적인 DDS를 FPGA(Field Programmable Gate Array)에 구현하였다. FPGA는 반도체 집적화의 진보로 태어난 소자로서 개별 로직 소자가 아닌 내부에 CLB(Configurable Logic Block)라는 단위 블록들로 이루어진 소자로 사용자가 이를 프로그램 하여 원하는 입력과 출력으로 사용할 수 있어서 하드웨어 회로의 복잡성을 줄이고 신뢰도를 높일 수 있다. FPGA를 이용한 DDS는 베릴로그 코드로 구성된 LUT로 설계되었다. 이 설계 역시, PA, LUT, D/A로 구성되어 있으며 RTL(Register Transfer Language) 모델로 시뮬레이션하고 있다. Matt Bergeron and Alan N. Willson, Jr.은 [13]에서 FPGA상에서 1-GHz DDS에 대하여 연구하였다. FPGA로 구현된 고속 직교 DDS는 사인 및 코사인 출력의 크기를 왜곡하지 않는 새로운 승수 기반 각도 회전 알고리즘을 기반으로 한 이 알고리즘은 FPGA에 존재하는 DSP 슬라이스에 잘 매핑되어 있도록 설계되었고, 이 장치는 Xilinx Virtex-7 장치에서 구현되며 이전에 ASIC 설계에서만 달성할 수 있었던 성능인 1GHz에서 54.9mW의 전력을 소비한다. FPGA를 이용한 또 다른 연구는 M.SaberSaber, M.Elmasry, M.EldinAbo- Elsoud가 연구한 [14]의 Quadrature DDS에서도 3.96mW의 낮은 전력소모와 성능이 우수한 59dBc의 스푸리어스를 보이며 주파수 분해능이 1.5kHz인 주파수 합성기를 제안하였다. FPGA 상에서 동작 중 저전력 구현을 위해서 ROM을 사용하지 않는 구조로 연구되었다. 그림 6과 같이 전형적인 DDS의 구조에서 Phase에서 Amplitude 변환기의 단점을 보완한 간단한 접근법은 LUT라는 기능의 ROM을 사용하는 것이다. 그러나 다음 수식에서 보듯이,

$$f_{out} = \frac{W}{2^L} F_{clk} \quad (1)$$

L : 누적기의 비트 수

W : 입력 주파수 워드의 비트 값

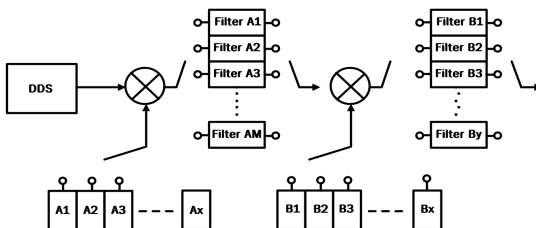
일반적으로 미세한 주파수 튜닝을 달성하기 위해서는 큰 L 값을 필요로 하기에 적절한 성능을 유지하면서 ROM 크기를 제한하는 몇 가지 기술이 고안되었는데, 그 중 한 가지 방법이 Sine 함수의 4분의 1파 대칭을 사용하

여 Sine 진폭에 필요한 각도의 수를 4로 줄이는 것이다. 위상 누산기의 출력을 잘라내는 것은 스피리어스 같은 잡음을 불러오지만 이 접근법은 L에 대해 매우 큰 값이 필요한 미세한 주파수 분해능을 달성하므로 일반적으로 사용되어지고 있다. LUT 기반의 FS에서 메모리 크기를 줄이기 위해서 다양한 각도 분해 방법이 제안되었다. 이들은 일반적으로 ROM을 여러 개의 작은 단위로 나누는 것으로 구성되며, 각 단위는 절단된 위상 누산기 출력의 일부로 처리된다. 각 하위 ROM에서 검색된 데이터가 추가되어 Sine 곡선 근사 값을 산출한다. 따라서 [14]에서 제안된 구조는 Sine 함수를 선형 세그먼트로 나누고, 각 세그먼트는 선형 방정식을 가지며, 이 방정식의 값은 추가 하드웨어를 통해 얻어진다고 기술하였다.

Wenjun Chen 등은 [15]를 통해서 DDS의 성능개선을 CORDIC(Coordinated Rotation Digital Computer) 알고리즘으로 구현하는 방법에 대하여 연구하였다. 이들은 XILINX의 FPGA를 이용하여 소용량의 ROM으로 반복 병합하여 출력지연을 감소시키는 방법을 사용하였다. 이들 방법은 350MHz 높은 주파수에서 SFDR 86.76dB 정현파를 구현하는 것을 볼 수 있다. Yixiong Yang 등은 [16]에서 전통적인 DDS의 LUT-ROT(Rotation) 아키텍처를 제안한다. 2GHz 대의 DDS를 속도와 면적의 최적화를 위하여 파이프라인화한 LUT로 11.7mW/GHz를 0.016mm² 면적을 구현하였다.

2.3. 하이브리드 주파수 합성기

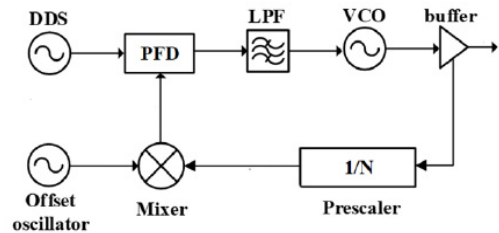
아날로그와 디지털 주파수 합성기의 구조를 혼합한 형태는 직접방식 및 간접방식 모두에서 연구되었다. 먼저 앞서 언급한 Direct Analog 주파수 합성에서 DDS를 이용하는 것으로서 그림 7과 같이,



(그림 7) DDS가 부가된 DAFS 개념도
(Figure 7) Conceptual diagram of DAFS with DDS added

아날로그 방식에서는 입력부에 DDS를 추가하여 설계의 복잡성과 전체 구성요소를 감소시킬 수 있다. 다음으로는 PLL 시스템에서 프렉셔널-N 분주기 대신에 DDS를 삽입하는 방안이 있다. 일종의 PLL과 DDS의 혼합 시스템이라 할 수 있다. 디지털 하이브리드 PLL 주파수 합성기의 위상잡음을 분석한 연구 [3]에서는 최소 위상잡음을 얻기 위한 입력 잡음, 양자화 오류로 인한 D/A 변환 잡음, VCO 잡음 소스의 수학적모델을 유도하고 분석하였다.

[4]는 DDS와 PLL을 기반으로 한 고해상도 빔포밍 수신기의 성능을 향상시키기 위해서 14비트의 DDS를 적용하여 높은 위상 분해능을 구현하였다. [4]에서는 기존의 DDS만을 사용해서 멀티 빔포밍을 구현하는 것을 그림 8과 같이 DDS-PLL을 이용하여 구현하였다.



(그림 8) DDS-PLL 블록 다이어그램
(Figure 8) DDS-PLL Block Diagram

Akila Gothandaraman, and Syed K.의 [18]은 PLL을 모두 디지털로 구현하여 아날로그의 단점을 제거한 ADPLL(An All-Digital Frequency Locked Loop)을 제안하였다. 이는 고속 주파수 획득이 가능한 모든 디지털 주파수 고정 루프(ADPLL)를 주파수 합성하는 알고리즘으로 하드웨어 비용 및 아키텍처가 감소하고, 전체 디지털화가 가능하며, 설계 및 구현이 용이한 펄스 출력 직접 디지털 주파수 합성기(DDFS)가 된다. 또한 적응형 위상 추정기를 제안하여 이 연구 DDFS는 16bit 이진 가중제어를 가지고 있으며 시뮬레이션 결과, ADPLL은 50MHz와 500MHz 사이의 주파수 범위에서 동작할 수 있음을 보여준다.

3. 제안 알고리즘

3.1. 필요성

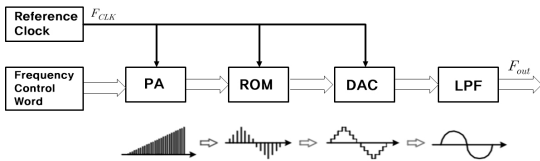
지금까지 관련연구에서 살펴본 바와 같이 주파수 합

성기(Frequency Synthesizer)는 여러 가지 방식이 각각 장·단점에 맞게 사용되고 있다. 2장에서 살펴본 바와 같이 이 논문에서 연구하고자 하는 FPGA 기반의 DDS는 신뢰성과 개발의 편의성 등으로 점차 그 이용범위가 늘어나고 있다. 본 논문에서는 여러 가지 응용분야 중 철도시스템에 적용하는 연구를 수행하였다. 철도나 지하철은 전동차의 선로 주행을 정확히 감지하여 특정 구간의 열차 제어 및 감시를 수행하는 AF(Audio Frequency) 케도회로 장치를 사용하고 있다.

AF 케도회로장치는 특정한 가청주파수를 기준으로 하여, 30Hz 이하의 주파수를 변조하여 전송하는 변조-복조 전송방식의 장치이다. 이 장치는 아날로그 방식으로 안정화가 완료된 후 20~30년이 경과하여 보다 개선된 장치가 필요한 상황이 되었다. 본 논문은 가청주파수를 정확하고 안정적으로 발생시키는 장치를 제작하기 위해 원하는 가청주파수를 생성시키는 Audio Frequency Generation Equipment의 설계 및 제작을 연구한다. 이를 위해 CPU를 배제한 순수한 논리회로 구조의 FPGA를 이용한 DDS를 제안한다. 이는 신뢰성 있고 정밀한 철도 케도회로 주파수 발생을 유도하여 오류로 인한 오동작을 방지한다. 구현한 주파수 발생장치는 시뮬레이션을 통해 그 효과성을 검증한다.

3.2. 제안 시스템의 구조

제안하는 시스템은 그림 9에서 DDS의 기본적인 블록 다이어그램[6][11]을 나타내고 있다. 이 블록 중에서 DAC 와 LPF를 제외한 부분을 FPGA로 구현한다.



(그림 9) 제안 시스템 기본 구조도

(Figure 9) Basic structure of the proposed system

그림 9와 같이 PA(Phase Accumulator)에 입력된 기준 클럭 F_{CLK} 와 FCW(Frequency Control Word)는 주파수 제어워드의 입력에 의해 출력주파수가 결정된다. PA의 출력은 ROM에 저장되어 있는 LUT(Look-Up Table)에서 정현파 진폭 값으로 변환된다. 이들 출력은 DAC(Digital Analog Converter)와 LPF를 이용하여 순수한 정현파 주파

수가 생성된다. 이 때의 주파수는 다음의 수식으로 정의된다 [13].

$$F_{out} = \frac{FCW}{2^L} F_{CLK} \quad (2)$$

FCW : 주파수 제어 워드

L : PA의 비트수

이 때 주파수의 해상도는 다음 수식처럼 기준 클럭 주파수가 작고 PA의 비트수가 클수록 증가하는 경향이 있다. DDS의 주파수 분해능은 기준 클럭 주파수를 누산기의 2^N 비트로 나눈 값으로 정의할 수 있으며 이는 아래 수식으로 나타낼 수 있다.

$$\Delta F = \frac{F_{CLK}}{2^L} \quad (3)$$

따라서 정밀한 주파수 파인튜닝을 위해서는 큰 L값과 ROM의 크기를 적절히 제한 할 필요가 있다.

3.3 Target 발생 주파수

그림 9의 제안 시스템에서 기준 클럭인 F_{CLK} 의 안정성은 매우 중요하다. 따라서 기준 클럭을 발생시키기 위해 높은 주파수의 X-tal을 사용한다. 이 소자는 산업용으로 공급되며 -40℃ ~ +120℃의 온도범위를 갖고 있어 매우 안정적이다. 이 X-tal로 발생한 기준 주파수 기반 하에 케도회로에 필요한 1kHz ~ 6kHz의 주파수를 발생시키며 이를 측면에 장착된 TWS(Thumb-wheel Switch)로 가변 할 수 있다. 발생하는 주파수 목표 정확도가 0.05%로 TWS의 값을 변경시켜서 원하는 주파수를 얻을 수 있다.

이처럼 발생된 주파수는 여러 분야에 응용할 수 있으나 본 연구에서는 철도 AF케도회로에 사용하는 주파수를 발생하였다. AF케도회로 주파수는 FSK(Frequency Shift Keying)를 변조하여 송신하고 수신은 이를 복조하여 전송된 주파수를 검지 및 분석함으로써 해당 케도회로 구간에 열차가 있는지를 판단한다. 즉 주파수가 검지 되면 해당 케도회로 구간에 열차가 없음으로 판단하고 검지가 안 되면 열차가 있음으로 판단한다.

FSK는 주파수 편이 변조로 진폭이 일정한 주파수에 따라 데이터가 0과 1사이에 서로 다른 주파수를 선택하는 변조방식이다.

주파수 편이를 f 라 하면 FSK 변조 신호는

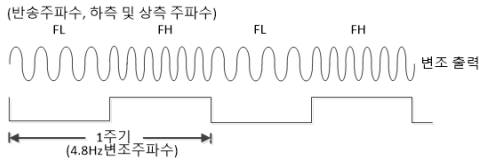
$$S(t) = A \cos 2\pi(f_C - \Delta f)t : 0 \leq t \leq T : 1$$

$$S(t) = A \cos 2\pi(f_C + \Delta f)t : 0 \leq t \leq T : 0$$

가 된다.

여기서 A 는 FSK의 진폭이며, f_C 는 반송주파수의 중심주파수이고, Δf 는 편이주파수이다.

이의 수신은 반송파인 두 개의 정현파 주파수를 사전에 인지하여 해당되는 주파수를 추출하고 변조주파수를 복원하는 것을 말하며 이를 복조라고 한다. 이 방식은 ASK (Amplitude Shift Keying) 보다 전송 불량률이 좋고 회로도도 비교적 간단하여 철도에서 사용하는 AF 케도회로장치 등 전송 장비에 많이 활용한다. 철도의 케도회로장치에서 변조주파수는 4.8Hz로 고정이며 반송주파수는 복복선과 상·하선을 감안하여 8개의 중심주파수가 있고, 중심주파수의 17Hz 상측 편이주파수와 17Hz 하측 편이주파수의 2개 주파수를 변조시킨다. 철도의 케도회로 주파수 조합은 그림 10과 같다.



(그림 10) 철도에 사용되는 케도회로 주파수 조합
(Figure 10) Track circuit frequency composition used in railways

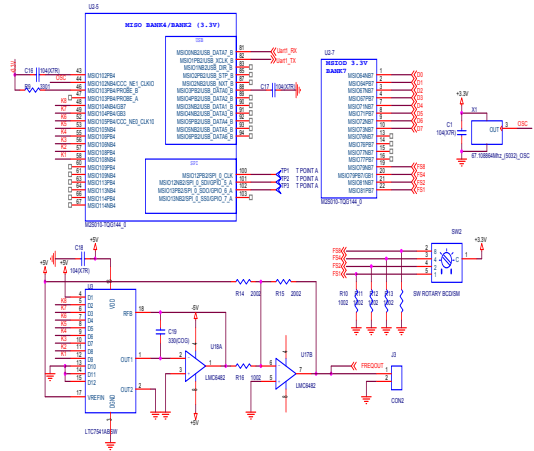
이들 케도회로 주파수는 유럽 등과 우리나라에서 사용하고 있으며 이에 대한 Bombardier 규격 [20]에서는 송·수신 주파수 배열을 아래 표 1과 같이 규정하고 있다.

(표 1) Bombardier 케도회로 사용주파수 배열
(Table 1) Bombardier track circuit frequency array

주파수명	중심주파수(Hz)	주파수 구분	주파수(Hz)±5%	비고
A	1699	하측주파수(FL)	1682	1699-17
		상측주파수(FH)	1716	1699+17
B	2296	하측주파수(FL)	2279	2296-17
		상측주파수(FH)	2313	2296+17
C	1996	하측주파수(FL)	1979	1996-17
		상측주파수(FH)	2013	1996+17
D	2593	하측주파수(FL)	2576	2593-17
		상측주파수(FH)	2610	2593+17
E	1549	하측주파수(FL)	1532	1549-17
		상측주파수(FH)	1566	1549+17
F	2146	하측주파수(FL)	2129	2146-17
		상측주파수(FH)	2163	2146+17
G	1848	하측주파수(FL)	1831	1848-17
		상측주파수(FH)	1865	1848+17
H	2445	하측주파수(FL)	2428	2445-17
		상측주파수(FH)	2462	2445+17

1) 알고리즘 설계

AF케도회로 주파수를 발생시키는 알고리즘에 필요한 회로는 그림 11과 같다.



(그림 11) 알고리즘 구현에 필요한 회로도
(Figure 11) Circuit diagram required to implement the algorithm

좌측 상단부의 M2S010은 FPGA로 이루어져 있고 우측 상단부는 기준 클럭 발생기로서 67.108864Mhz의 X-tal를 사용한다. 좌측 하단부는 디지털 아날로그 변환기와 LPF를 나타내고 있다. 우측 중간에는 로타리 스위치로서 주파수를 선택하는 TWS이다.

8bit로 구성된 FPGA 출력을 D/A converter인 아날로그 디바이스 사의 AD7541을 사용하여 아날로그 신호로 변환된다.

TWS의 선택에 의해 Bombardier 규격에 따른 케도주파수(A~H의 상측 및 하측) 중에서 해당된 1개의 주파수가 발생된다. 이는 같은 논리를 작성하여 각각 상측과 주파수와 하측과 주파수를 발생하는 두 개의 주파수 출력으로 구성할 수도 있다.

LUT(Look-Up Table) 값이 내장돼 있는 ADC_A_OUT라는 로직 블록은 FPGA로 구현되며 이 로직 블록에 의해서 256 스텝의 디지털 출력이 생성된다. 이 결과는 디지털 아날로그 변환 IC인 AD7541 입력에 인가되어 아날로그 신호로 출력된다.

출력되는 아날로그 정현파 신호를 얻기 위해서는 FPGA 출력은 DCOUNT13이라는 로직 블록으로 D-FF (Flip Flop)을 13번 수행하여 50:50의 듀티비를 갖도록 구

성하였다. 그림 10의 SW2로서 0~F 까지 16개의 주파수를 선택할 수 있도록 하였으며 스위치 위치에 따른 주파수 출력은 표 1과 같이 설계하였다.

2) FPGA 로직 블록

가. 입출력 포트 구성

입출력 포트는 입력부로 오실레이터 clock, 리셋, 주파수 선택 스위치 입력이 있으며 출력부로 8bit 디지털 주파수, 3개의 테스트 단자가 있도록 설계하였다.

```
entity PCSFRGEN is
port(Clk      : in std_logic; -- 67108864MHz Clock
     ACLR_IN  : in std_logic; -- Reset
     SEL_IN   : in std_logic_vector(3 downto 0);
     ROM_OUTA : out std_logic_vector(7 downto 0);
     TEST_P1  : out std_logic;
     TEST_P2  : out std_logic;
     TEST_P3  : out std_logic
);
end PCSFRGEN;
```

나. 컴포넌트 구성

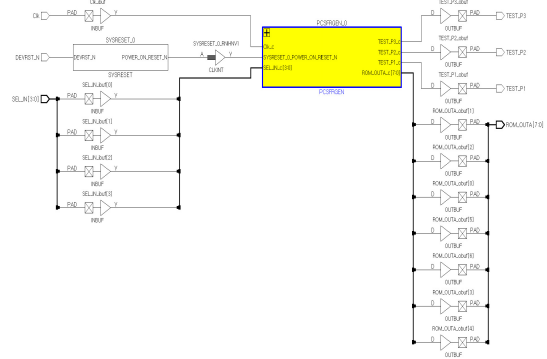
내부 로직을 구성하는 컴포넌트 구성은 PCSFR, Value_filter, ADC_A 및 클록 버퍼로 구성하였으며 클록 버퍼를 제외한 컴포넌트는 아래와 같이 구성하였다.

```
component PCSFR is
port( Aclr      : in std_logic;
      CLK      : in std_logic;
      WREN     : in std_logic;
      SEL      : in std_logic_vector(3 downto 0);
      Q        : out std_logic_vector(7 downto 0)
);
end component;
```

```
component Value_filter is
port(
  Aclr  : in std_logic;
  Clk   : in std_logic; -- 4Mhz
  SEL_IN : in std_logic_vector(3 downto 0);
  SEL_OUT : out std_logic_vector(3 downto 0)
);
end component;
```

```
component ADC_A_BLK is
port ( ADC_CT : in std_logic_vector(7 downto 0);
       ADC_A  : out std_logic_vector(7 downto 0);
       PCOUNT : out std_logic
);
end component;
```

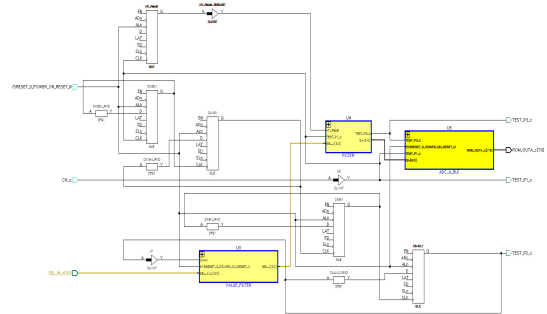
다. 구성 로직 컴파일
FPGA 내부 사용한 전체를 컴파일 한 블록을 캡처하여 아래 그림에 표시하였다.



(그림 12) 컴파일 전체 블록도
(Figure 12) block diagram of compilation

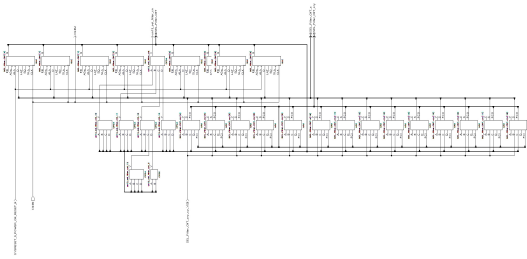
그림 12의 컴파일 전체 블록은 입출력을 위한 단자가 PCSFRGEN 블록에 연결되어 있다.

PCSFRGEN 블록의 내부는 다음 그림 13과 같이 구성된다.

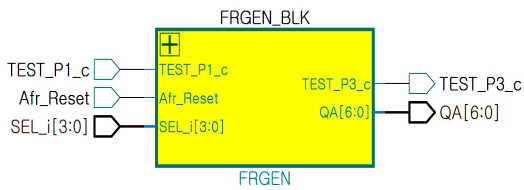


(그림 13) PCSFRGEN 컴파일 블록도
(Figure 13) PCSFRGEN compilation block diagram

그림 14는 입력부의 스위치 입력을 chattering 방지를 위해 작성된 Value_filter 로직 블록과 주파수 발생부인 그림 15 FRGEN 로직 블록 그리고 SINE Wave 출력을 위한 8bit look-up table을 내장시킨 ADC_A_OUT 로직 블록으로 구성된다. 구성 각각의 컴파일 블록을 아래에 나타낸다.

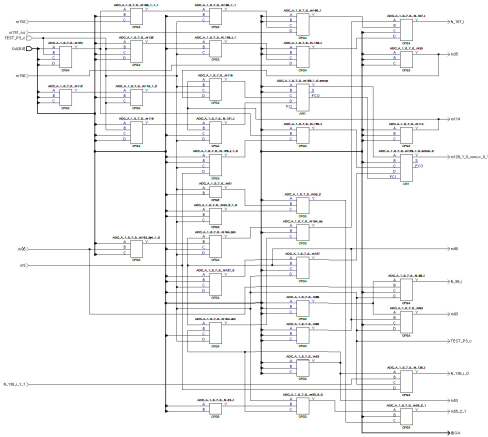


(그림 14) Value filter 컴파일 블록도
(Figure 14) Value_filter compilation block diagram



(그림 15) FRGEN 컴파일 블록도
(Figure 15) FRGEN compilation block diagram

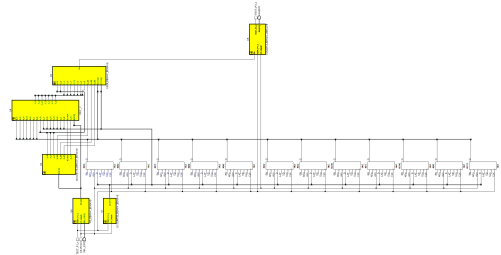
그림 16은 이 블록의 세부 컴파일된 블록이며, 세부내용을 그림 17, 18에 나누어 표시하였다.



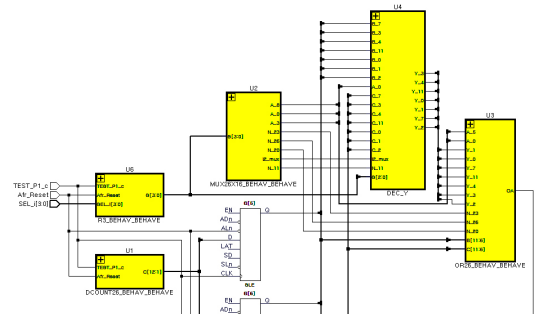
(그림 16) ADC_A_OUT(Look-up Table) 컴파일 블록도
(Figure 16) ADC_A_OUT (Look-up Table) compilation block diagram

그림 15 FRGEN 블록은 주파수 발생을 위한 초기 로직들의 집합이라고 할 수 있다. 이의 각 부의 로직 구성 alc 컴파일된 캡처 회로는 아래와 같다.

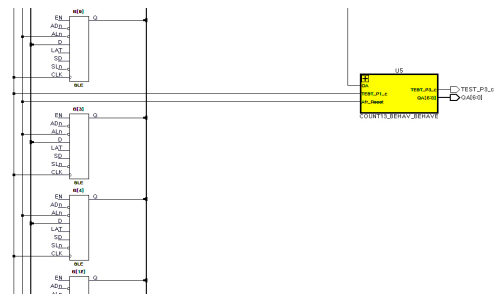
```
entity FRGEN is
port( RESETn : in std_logic;
      CLK : in std_logic;
      SEN : in std_logic;
      SI : in std_logic_vector(3 downto 0);
      Q : out std_logic_vector(12 downto 0)
);
end FRGEN;
```



(그림 17) FRGEN 컴파일 블록도
(Figure 17) FRGEN compilation block diagram



(그림 18) FRGEN 컴파일 부분블록도1
(Figure 18) FRGEN compilation partial block diagram 1

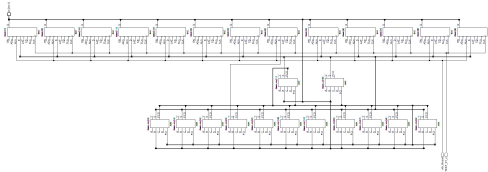


(그림 19) FRGEN 컴파일 부분블록도2
(Figure 19) FRGEN compilation partial block diagram 2

내부의 컴포넌트의 구성은 아래와 같다.

```

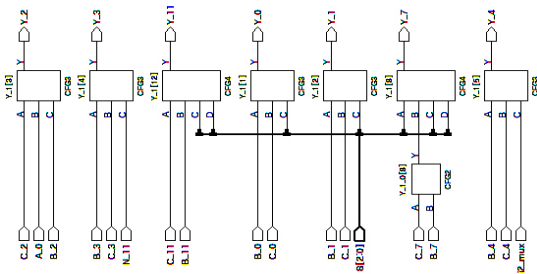
component DCOUNT26 is
port( RESETn : in std_logic;
      Clock : in std_logic;
      Q : out std_logic_vector(25 downto 0)
);
end component;
    
```



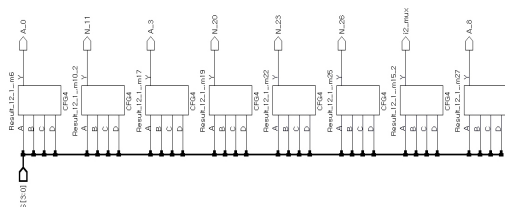
(그림 20) DCOUNT26 컴파일 블록도
(Figure 20) DCOUNT26 compilation block

```

component DEC_Y is
port(A : in std_logic_vector(25 downto 0);
      B : in std_logic_vector(25 downto 0);
      C : in std_logic_vector(25 downto 0);
      Y : out std_logic_vector(25 downto 0));
end component;
    
```



(그림 21) DEC_Y 컴파일 블록도
(Figure 21) DEC_Y compilation block diagram



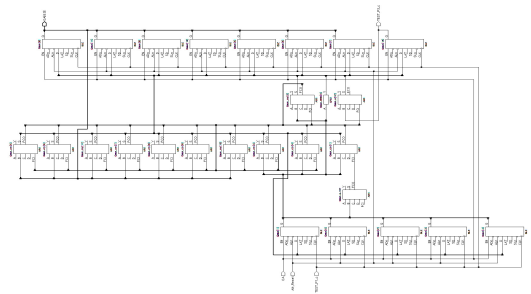
(그림 22) MUX26x8 컴파일 블록도
(Figure 22) MUX26x8 compilation block diagram

```

component MUX26X16 is
port(Data0_port : in std_logic_vector(25 downto 0);
      Data1_port : in std_logic_vector(25 downto 0);
      Data2_port : in std_logic_vector(25 downto 0);
      Data3_port : in std_logic_vector(25 downto 0);
      Data4_port : in std_logic_vector(25 downto 0);
      Data5_port : in std_logic_vector(25 downto 0);
      Data6_port : in std_logic_vector(25 downto 0);
      Data7_port : in std_logic_vector(25 downto 0);
      Data8_port : in std_logic_vector(25 downto 0);
      Data9_port : in std_logic_vector(25 downto 0);
      Data10_port : in std_logic_vector(25 downto 0);
      Data11_port : in std_logic_vector(25 downto 0);
      Data12_port : in std_logic_vector(25 downto 0);
      Data13_port : in std_logic_vector(25 downto 0);
      Data14_port : in std_logic_vector(25 downto 0);
      Data15_port : in std_logic_vector(25 downto 0);
      Sel0 : in std_logic;
      Sel1 : in std_logic;
      Sel2 : in std_logic;
      Sel3 : in std_logic;
      Result : out std_logic_vector(25 downto 0));
end component;
    
```

```

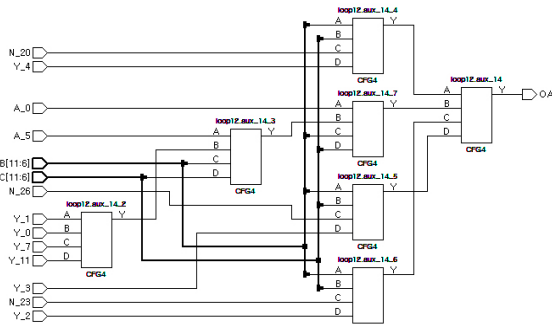
component count13 is
port(Clock : in std_logic;
      Q : out std_logic_vector(12 downto 0);
      RESETn : in std_logic;
      Enable : in std_logic);
end component;
    
```



(그림 23) COUNT13 컴파일 블록도
(Figure 23) COUNT13 compilation block diagram

```

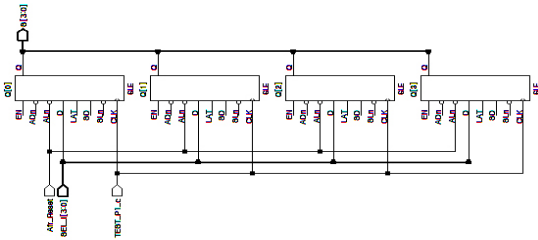
component OR26 is
port(Data : in std_logic_vector(25 downto 0);
      Result : out std_logic);
end component;
    
```



(그림 24) OR26 컴파일 블록도
(Figure 24) OR26 compilation block diagram

```

component R3 is
port(Data : in std_logic_vector(3 downto 0);
  Enable : in std_logic;
  RESETn : in std_logic;
  Clock : in std_logic;
  Q : out std_logic_vector(3 downto 0));
end component;
    
```



(그림 25) R3 컴파일 내부 블록도
(Figure 25) R3 compile internal block diagram

3) 웨도회로 사용 주파수 발생

목표 주파수 생성을 위하여 사용되는 13bit는 그림 23의 FPGA로 처리한 up 카운터의 MSB(Most Significant Bit)쪽으로 갈수록 정확하게 생성되므로 상위 8bit를 사용하였고 LSB(Least Significant Bit) 5bit를 사용하여 듀티비가 50:50이 되도록 설계하였다. 그리하여 생성된 주파수는 아래의 수식에 따르게 된다.

$$LC\ Frequency(F_{LCF}) = \frac{k}{2^5} \quad (4)$$

LC : Line Circuit으로서 웨도회로

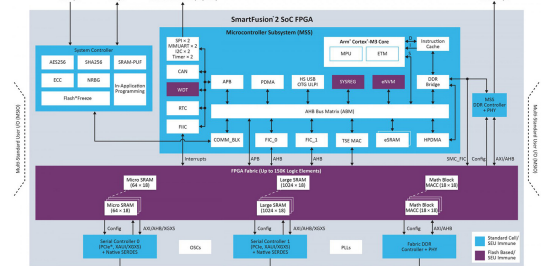
k : 목표 주파수로 추출된 bit

출력된 상위 8bit는 DAC 회로(AD7541)에 연결되어 Analog Sine Wave 출력으로 변환된 Audio Frequency 출력으로 생성되어 완전한 정현파 주파수가 출력된다.

4) 하드웨어 구현

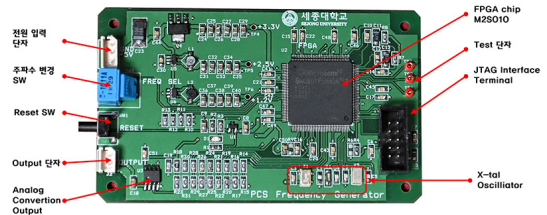
제안시스템의 FPGA 칩은 마이크로칩사의 Smart-Fusion2 SoC M2S010 [19]를 사용하였으며, 최대 사용가능한 로직 소자는 12,084개인 높은 집적도의 시스템 칩 IC로 구성되어 있다.

M2S010은 저 전력 소모 용도로 만들어졌으며 비디오 이미지처리, I/O확장 및 변환, Gigabit 이더넷 등 다목적 용도로 뛰어난 신뢰성과 보안성을 제공하는 칩 디바이스이다. ARM 계열의 MPU도 내장하고 있으나 본 제안 시스템은 고 신뢰성이 동작의 목적이어서 사용하지 않았다. 그림 26은 M2S010의 내부 블록도를 보여 주고 있다.



(그림 26) M2S010의 내부 블록도
(Figure 26) Internal block diagram of M2S010

제작한 회로의 사진을 그림 27과 같다.



(그림 27) 시험에 사용한 회로 기판
(Figure 27) Circuit board used for testing

사용 기판의 우측 시계방향으로 주요 부분의 설명을 참조하면, FPGA 칩, 테스트 단자, JTAG 인터페이스 터미

널, X-tal Oscillator이다. 좌측 상단부터 설명은, 전원 입력 단자, 주파수 변경 SW, 리셋 SW, 출력단자 및 아날로그 파형 출력 IC이다. 우측 하단의 10핀 JTAG Interface Terminal 은 FPGA에 프로그램을 입력할 때 사용한다.

이 회로를 구성한 결과 주파수 정확도에서 많은 긍정적인 특성들이 있었다. 그것은 오실레이터의 값 특히 2진수의 값을 갖는 것이면 정확한 주파수 출력을 기대할 수 있다. 실제로 크리스탈 오실레이터로 67.108864MHz를 사용하면 정확도 100%의 회로를 구사할 수 있다. 실험결과는 Octal 주파수 변경 스위치로 선택하면서 각 주파수별로 출력 되는 상태를 오실로스코프로 측정하였다. 각 주파수별 파형은 4장 실험결과에서 세부적으로 기술한다.

4. 실험결과

FPGA 로직 블록으로 처리한 목표 주파수를 최적으로 생성 할 수 있는 방법으로 거듭된 시행착오를 거친 후 다음의 수식으로 정리된 결과의 검증을 수행하였다.

$$Y = \frac{1}{CLOCK} \times 2^{13} \times 2^{26} \times Frequency \quad (5)$$

여기서

Y: 목적주파수를 생성하기 위한 최적의 주출비트

CLOCK: 본 실험에 사용한 레퍼런스 클럭

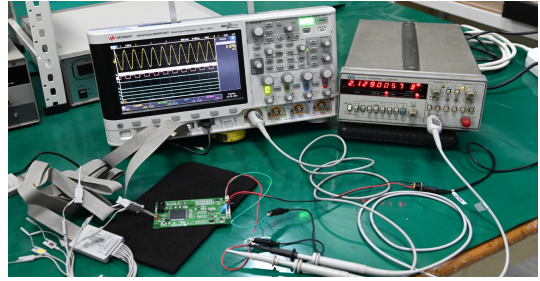
Frequency: 목표 주파수.

철도 궤도회로에 사용되는 AF대역의 DDFS를 정밀하게 만들어내는 실질적인 구조와 알고리즘을 FPGA로 구현함으로써 유럽 및 국내에서 현재 사용되고 있는 16개 주파수를 99.9980% ~ 99.9996%의 정밀도로 구현함을 보여주었으며 이를 시뮬레이션 결과로 나타내었다.

시뮬레이션 결과로 주파수 발생 편차가 목표 오차범위보다 우수한 안정되고 정확한 주파수의 출력이 만들어짐을 확인 할 수 있었다. FPGA를 이용한 기존 연구의 [11] 0~160kHz의 범위에서 5Hz의 편차를 갖는 결과에 비해서 매우 우수한 성능을 보여주고 있음을 알 수 있다. 표2와 표3에서 결과를 확인 할 수 있다.

4.1. 시뮬레이션

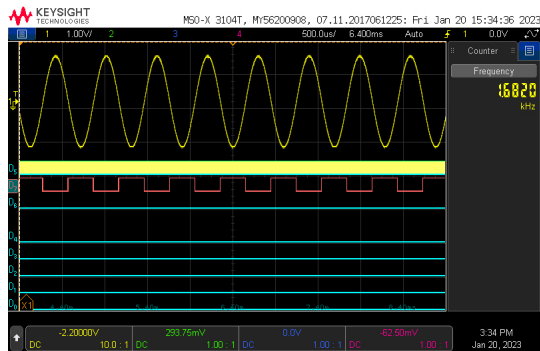
그림 28와 같이 연결하여 제작한 보드를 작동시켰다.



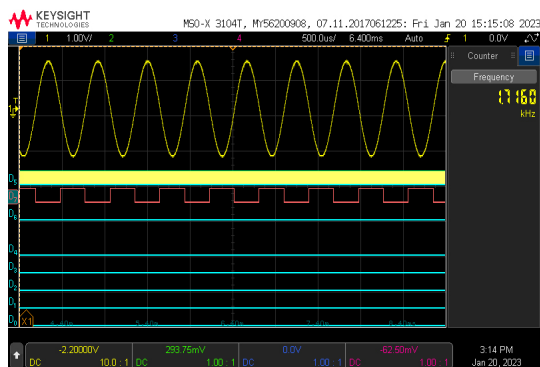
(그림 28) 시뮬레이션 (Figure 28) simulation

시험용 보드에 우측에 있는 주파수 변경용 TWS를 돌리면서 관측한 출력 결과는 다음과 같다.

1) 궤도주파수 A 시험 결과 및 파형



(그림 29) 하측 주파수 1682Hz 생성 파형 (그림 29) Lower frequency 1682 Hz generated waveform



(그림 30) 상측 주파수 1716Hz 생성 파형 (Figure 30) Upper frequency 1716 Hz generated waveform

ADC7541 A/D 변환회로 출력에 나타나는 신호파형으로 TWS(Thumb-wheel Switch)를 '0'에 위치할 때 출력되는 주파수 1682Hz 로 설계한 하측주파수 값과 비교한 정확도는 $100 - \{100 \times (1682.0007 - 1682)/1682\} = 99.9999\%$ 이며 상측 주파수 정확도는 표2와 같다.

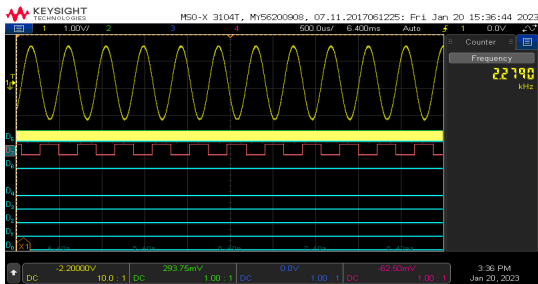
(표 2) 시뮬레이션 결과
(Table 2) simulation result

항	목표 설계 주파수(Hz)(±0.05%)	시험결과 주파수(Hz)	정확도(%)	비고
0	1682	1682.0007	99.99996838	A
1	1716	1716.0009	99.99994755	
2	2279	2279.0043	99.99981132	B
3	2313	2313.0069	99.99970169	
4	1979	1979.0036	99.99981809	C
5	2013	2013.0009	99.9996529	
6	2576	2576.0030	99.99986025	D
7	2610	2610.0052	99.99980077	
8	1532	1532.0026	99.99983029	E
9	1566	1566.0047	99.99969987	
A	2129	2129.0021	99.99990136	F
B	2163	2163.0037	99.99982894	
C	1831	1831.0037	99.99979792	G
D	1865	1865.0038	99.99979625	
E	2428	2428.0067	99.99976904	H
F	2462	2462.0056	99.99977254	

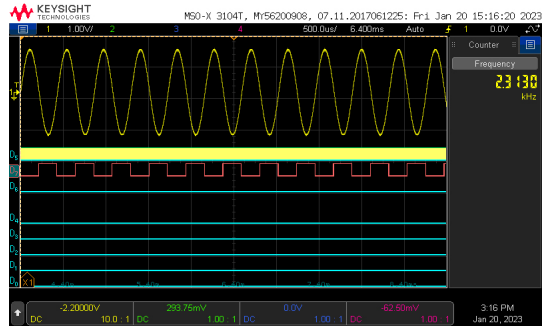
(표 3) 시뮬레이션 결과 비교
(Table 3) simulation result comparison

기존연구 및 제품과 비교	주파수 편차	정확도	비고
고 신뢰성 Audio Target 주파수 발생장치에 대한 연구	0.001~ 0.006Hz	0.001%	본 연구논문
Design and Implementation of a FPGA-based Direct Digital Synthesizer	+5Hz ~ -5Hz	0.3%	참고문헌[11]
Bombardier TI21 Track Circuit Test and Investigation Guideline	1Hz ~ 2Hz	0.06~0.12%	참고문헌[21]

2) 레도주파수 B 시험 결과 및 파형



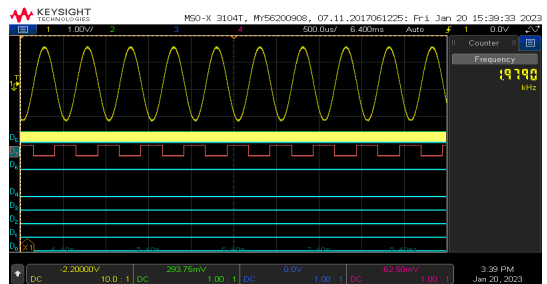
(그림 31) 하측 주파수 2279Hz 생성 파형
(Figure 31) Lower frequency 2279 Hz generated waveform



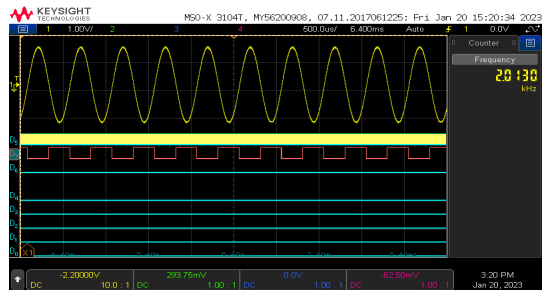
(그림 32) 상측 주파수 2313Hz 생성 파형
(Figure 32) Upper frequency 2313 Hz generated waveform

ADC7541 A/D 변환회로 출력에 나타나는 신호파형으로 TWS(Thumb-wheel Switch)를 '1'에 위치 할 때 출력되는 주파수 2279Hz 로 설계한 하측주파수 값과 비교한 정확도는 $100 - \{100 \times (2279.0043 - 2279)/2279\} = 99.9998\%$ 이며 상측 주파수 정확도는 표2와 같다.

3) 레도주파수 C 시험 결과 및 파형



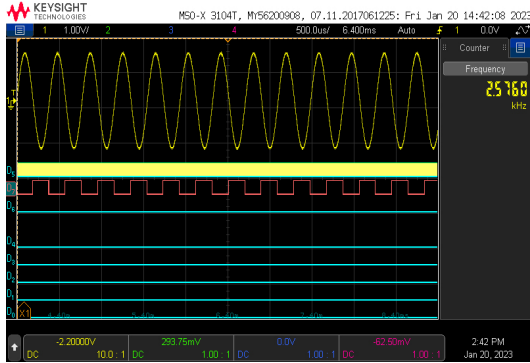
(그림 33) 하측 주파수 1979Hz 생성 파형
(Figure 33) Lower frequency 1979 Hz generated waveform



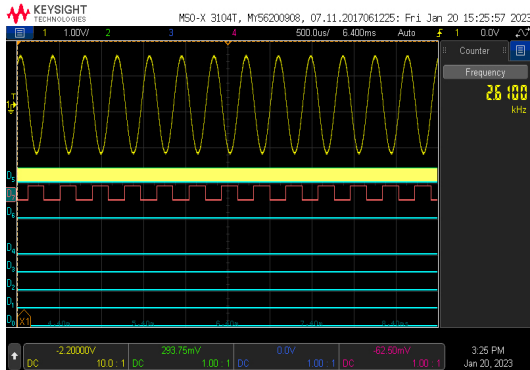
(그림 34) 상측 주파수 2013Hz 생성 파형
(Figure 34) Upper frequency 2013 Hz generated waveform

ADC7541 A/D 변환회로 출력에 나타나는 신호파형으로 TWS(Thumb-wheel Switch)를 '2'에 위치 할 때 출력되는 주파수 1979Hz 로 설계한 하측주파수 값과 비교한 확도는 $100 - \{100 \times (1979.0036 - 1979)/1979\} = 99.9998\%$ 이며 상측 주파수 정확도는 표 2에 보였다.

4) 레도주파수 D 시험 결과 및 파형



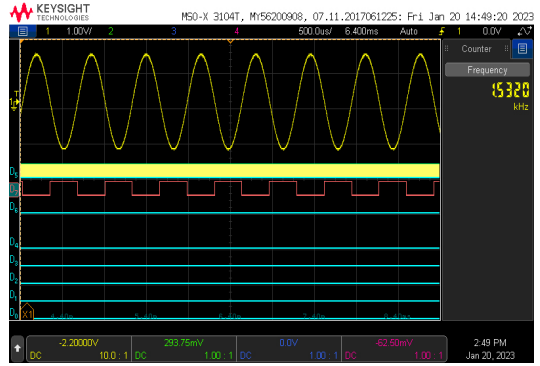
(그림 35) 하측 주파수 2576Hz 생성 파형
(Figure 35) Lower frequency 2576 Hz generated waveform



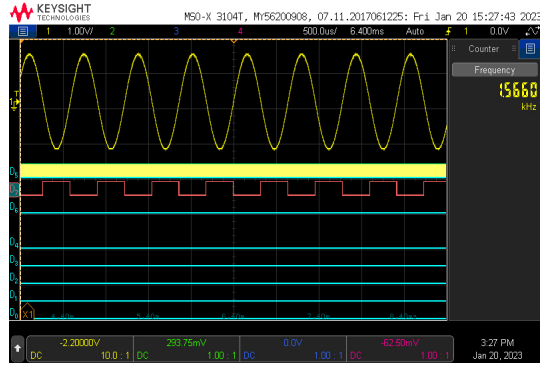
(그림 36) 상측 주파수 2610Hz 생성 파형
(Figure 36) Upper frequency 2610 Hz generated waveform

ADC7541 A/D 변환 회로 출력에 나타나는 신호 파형으로 TWS(Thumb-wheel Switch)를 '0'에 위치 할 때 출력되는 주파수 2576Hz 로 설계한 주파수 값과 비교한 확도는 $100 - \{100 \times (2576.0030 - 2576)/2576\} = 99.9998 \%$ 이며 상측 주파수 정확도는 표 2와 같다.

5) 레도주파수 E 시험 결과 및 파형



(그림 37) 하측 주파수 1532Hz 생성 파형
(Figure 37) Lower frequency 1532 Hz generated waveform

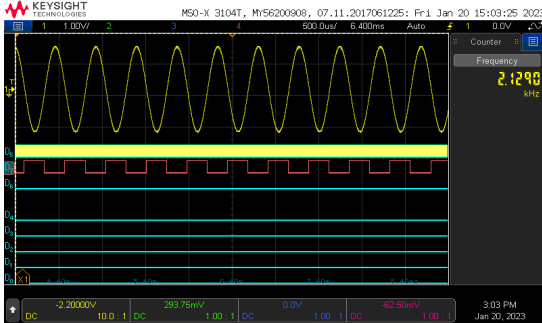


(그림 38) 상측 주파수 1566Hz 생성 파형
(Figure 38) Upper frequency 1566 Hz generated waveform

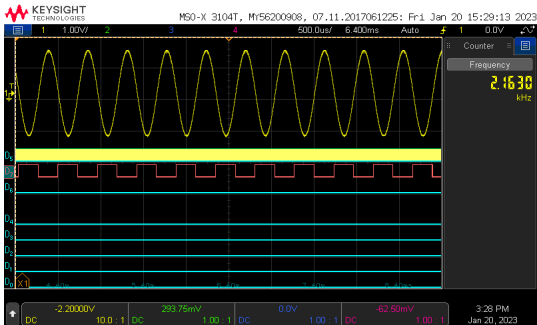
ADC7541 A/D 변환 회로 출력에 나타나는 신호 파형으로 TWS(Thumb-wheel Switch)를 '0'에 위치할 때 출력되는 주파수 1532Hz 로 설계한 주파수 값과 비교한 확도는 $100 - \{100 \times (1532.0026 - 1532)/1532\} = 99.9998 \%$ 이며 상측 주파수 정확도는 표 2와 같다.

6) 퀘도주파수 F 시험 결과 및 파형

ADC7541 A/D 변환 회로 출력에 나타나는 신호 파형으로 TWS(Thumb-wheel Switch)를 '0'에 위치할 때 출력되는 주파수 2129Hz 로 설계한 주파수 값과 비교한 확도는 $100 - \{100 \times (2129.0021 - 2129)/2129\} = 99.9999\%$ 이며 상측 주파수 정확도는 표 2와 같다.



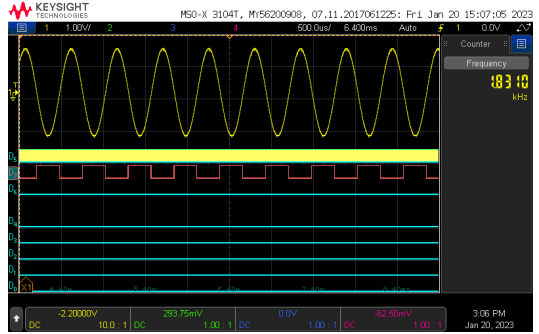
(그림 39) 하측 주파수 2129Hz 생성 파형
(Figure 39) Lower frequency 2129 Hz generated waveform



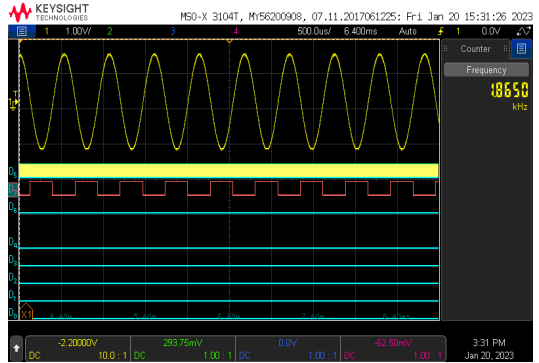
(그림 40) 상측 주파수 2163Hz 생성 파형
(Figure 40) Upper frequency 2163 Hz generated waveform

7) 퀘도주파수 G 시험 결과 및 파형

ADC7541 A/D 변환 회로 출력에 나타나는 신호 파형으로 TWS(Thumb-wheel Switch)를 '0'에 위치할 때 출력되는 주파수 1831Hz 로 설계한 주파수 값과 비교한 확도는 $100 - \{100 \times (1831.0037 - 1831)/1831\} = 99.9998\%$ 이며 상측 주파수 정확도는 표 2와 같다.



(그림 41) 하측 주파수 1831Hz 생성 파형
(Figure 41) Lower frequency 1831 Hz generated waveform

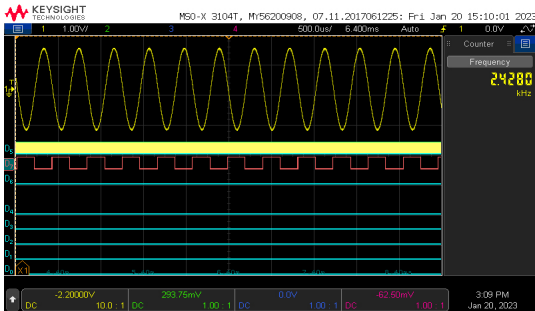


(그림 42) 상측 주파수 1865Hz 생성 파형
(Figure 42) Upper frequency 1865 Hz generated waveform

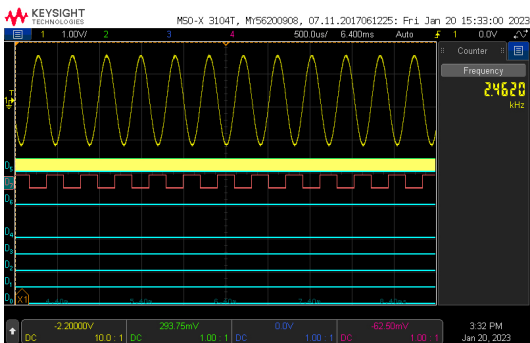
8) 퀘도주파수 H 시험 결과 및 파형

ADC7541 A/D 변환 회로 출력에 나타나는 신호 파형으로 TWS(Thumb-wheel Switch)를 '0'에 위치 할 때 출력되는 주파수 2428Hz 로 설계한 주파수 값과 비교한 확도는 $100 - \{100 \times (2428.0057 - 2428)/2428\} = 99.9997\%$ 이며 상측 주파수 정확도는 표 2와 같다.

표 2와 같이 목표하는 주파수에 99.99905%로 매우 정확한 주파수 발생을 하고 있음을 알 수 있고 0.05% 설계 목표 편차 범위를 만족하고 있음을 또한 보여준다.



(그림 43) 하측 주파수 2428Hz 생성 파형
(Figure 43) Lower frequency 2428 Hz generated waveform



(그림 44) 상측 주파수 2462Hz 생성 파형
(Figure 44) Upper frequency 2462 Hz generated waveform

5. 결 론

본 논문에서는 마이크로칩사의 FPGA를 사용하여 철도 궤도회로용 AF를 DDS를 이용하여 구현하는 방법을 제시하였다. 이 주파수 발생장치는 일반적으로 사용되는 CPU를 사용하지 않고 순수 Logic 회로만으로 구성하여 오동작의 요인을 최소화 하였으며 기간산업 분야에서 안전성을 높일 수 있는 가능성을 제시하였다. AF 대역의 DDFS를 정밀하게 만들어내는 실질적인 구조와 알고리즘을 제안함으로써 현재 철도의 궤도회로에 사용되고 있는 16개 주파수를 99.9980% ~ 99.9996%의 정밀도로 구현하였으며 이를 시뮬레이션 결과로 보여 주었다. 이로써 기존연구[11]의 5Hz 편차보다 우수한 성능을 구현하였음을 입증하였다.

이는 매우 안정되고 정확한 주파수의 출력을 생성할 수 있었으며, 이를 활용하면 철도 등 기간산업 분야에서

고도의 신뢰성을 갖는 정밀한 주파수 발생장치를 만들 수 있을 것으로 판단된다. 이러한 연구 성과는 철도 등 기간산업 분야에서 제어 시스템의 안전성과 사용자 편의성을 높이리라 기대되며, 향후 본 연구를 발전시켜 동시에 여러 개의 주파수를 발생시키는 다중 AF DDFS에 대해 연구가 진행될 경우 더 많은 다양한 산업분야에서 실용적으로 매우 유용할 것으로 기대된다.

참고문헌(Reference)

- [1] Kwangseop Yoon, Minkyu Song, Jeongjin Noh, Kangyoon Lee, "Design of Data Converters and PLL", p299~324, Hongneung Science Publishing House, 2013.
<https://doi.org/10.978.8997570/515>
- [2] J.Tierney, C.M.Radre, and B.Gold, "A Digital Frequency Synthesizer", IEEE Transactions on Audio and Electroacoustics, Vol. AU-19, No.1, March 1971.
<https://doi.org/10.1109/TAU.1971.1162151>
- [3] Heung-Gyoon Ryu and Hyun-Seok Lee, "Analysis And Minimization Of Phase Noise Of The Digital Hybrid PLL Frequency Synthesizer", IEEE Transactions on Consumer Electronics, Vol. 48, No. 2, May 2002.
<https://ieeexplore.ieee.org/abstract/document/1010136>
- [4] Dong-Chan Kim, Ye-Eun Chi, Junhyeong Park, "High-Resolution Digital Beamforming Receiver Using DDS - PLL Signal Generator for 5G Mobile Communication", IEEE Transactions on Antennas and Propagation, Vol. 70, No. 2, February 2022.
<https://doi.org/10.1109/TAP.2021.3111334>
- [5] Eliabe Duarte Queiroz, Joo Ina'cio Yutaka Ota and Jos'e Antenor Pomilio, "State-Space Representation Model of Phase-Lock Loop Systems for Stability Analysis of Grid-connected Converters", 14th IEEE International Conference on Industry Applications, 2021.
<https://doi.org/10.1109/INDUSCON51756.2021.9529609>
- [6] Salah Hassan Alkurwy, "A Novel ROM Design for High Speed Direct Digital Frequency Synthesizer", Lap Lambert Academic Publishing, Germany, 2014,
- [7] S. Gao and M. Barnes, "Phase-locked loops for grid-tied inverters: Comparison and testing", in 8th IET

- International Conference on Power Electronics, Machines and Drives (PEMD 2016), pp.1-6, April 2016.
<https://digital-library.theiet.org/content/conferences/10.1049/cp.2016.0304>
- [8] Andrzej Rokita, "Direct Analog Synthesis Modules For An X-Band Frequency Source", 12th International Conference on Microwaves and Radar. MIKON-98. Conference Proceedings (IEEE Cat. No.98EX195), Krakow, Poland, pp. 63-68 vol. 1, 1998.
<https://doi.org/10.1109/MIKON.1998.737920>
- [9] Yuchen Wang, Xuguang Bao, Wei Hua, "Implementation of Embedded Magnetic Encoder for Rotor Position Detection Based on Arbitrary Phase Shift Phase Lock Loop", IEEE Transactions on Industrial Electronics, vol. 69, no. 2, pp. 2033-2043, February 2002.
<https://doi.org/10.1109/TIE.2021.3062270>
- [10] Kim Sang-woo, Kim Joo-seong, Oh Hwan-sul, Cheon Jeong-in, Park Gyeong-seok, Go Seon-jun, Lee Kang-yoon, "Design of Low Power Frequency Synthesizer for GPS Receiver", 2022.
- [11] Changhong Shan, Zhongze Chen, Hua Yuan, Wei Hu, "Design and Implementation of a FPGA-based Direct Digital Synthesizer", IEEE, 2011,
<https://doi.org/10.1109/ICECENG.2011.6057152>
- [12] A.A Alsharif, M.A. Mohd. Ali and H. Sanusi, "Direct Digital Frequency Synthesizer Design and Implementation on FPGA", Reserach Journal of Applied Sciences, 7(8):387-390, 2012.
- [13] Matt Bergeron and Alan N. Willson, Jr., "A 1-GHz Direct Digital Frequency Synthesizer in an FPGA", 2014 IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, VIC, Australia, pp. 329-332, 2014.
<http://dx.doi.org/10.1109/ISCAS.2014.6865132>
- [14] M.Saber Saber, M.Elmasry, M.EldinAbo-Elsoud, "Quadrature Direct Digital Frequency Synthesizer Using FPGA", 2006 International Conference on Computer Engineering and Systems, Cairo, Egypt, pp. 14-18, 2006.
<https://doi.org/10.1109/ICCES.2006.320418>
- [15] Wenjun Chen, Tianya Wu, Wangwang Tang, Kai Jin, Guangming Huang, "Implementation Method of CORDIC Algorithm to Improve DDFS Performance", IEEE 3rd International Conference on Electronics Technology, 2020.
<https://doi.org/10.1109/ICET49382.2020.9119621>
- [16] Yixiong Yang, Zhibo Wang, Pei Yang, Meng-Fan Chang, Mon-Shu Ho, Huazhong Yang, Yongpan Liu, "A 2-GHz Direct Digital Frequency Synthesizer Based on LUT and Rotation", 018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, pp. 1-5, 2018.
<https://doi.org/10.1109/ISCAS.2018.8351207>
- [17] Kim Dong-shik, Lee Haeng-soo, Kim Jong-pil, Kim Seon-joo, "Design and Modeling of a DDS Driven Offset PLL with DAC", Korea Internet Broadcasting and Communication Society, Vol. 12, No. 5, pp. 1~9, October 2012.
<https://doi.org/10.7236/JIWIT.2012.12.5.1>
- [18] Akila Gothandaraman, and Syed K., "An All - Digital Frequency Locked Loop(ADFLL) with a Pulse Output Direct Digital Frequency Synthesizer(DDFS) and an Adaptive Phase Estimator", IEEE Radio Frequency Integrated Circuits Symposium, 2003,
<https://doi.org/10.1109/RFIC.2003.1213949>
- [19] Microchip, "FPGA and SoC Product Families", 2019,
<http://ww1.microchip.com/downloads/en/DeviceDoc/00002871B.pdf>
- [20] Bombardier, "EBI Track 200 TI21 Audio Frequency Track Circuit", Technical Manual, pp. 7-1~4, October 2011.
<https://docplayer.net/28867426-Ebi-track-200-ti21-audio-frequency-track-circuit.html>
- [21] Transport RailCorp, "TI21 Track Circuit Test and Investigation Guideline", p13~14, Issued December 2010.
<https://manualzz.com/doc/8332803/ti21-track-circuit-test-and-investigation-guideline--pdf->

● 저 자 소 개 ●



박 창 식(Chang-sik Park)

2006년 한국방송통신대학교 경영학과(경영학사)
2010년 성균관대학교 대학원 이동통신공학과(공학석사)
2021년~현재 세종대학교 대학원 컴퓨터공학과(박사과정 수료)
관심분야 : 에이전트, Computer Graphics & Vision, IOT Applications, Image Processing, etc.
E-mail : pcs5130@hanmail.net



한 은 택(Eun-tack Han)

1992년 한국항공대학교 항공통신공학과(공학사)
2010년 성균관대학교 대학원 이동통신공학과(공학석사)
2021년~현재 세종대학교 대학원 컴퓨터공학과(박사과정 수료)
관심분야 : AI & Intelligent System, 기계학습, Wireless Internet & Network, etc.
E-mail : shrd9600@daum.net



김 익 재(Ikjae Kim)

1995년 공군사관학교 전산학과(이학사)
2007년 국방대학원 전산정보학과(공학석사)
관심분야 : Risk Management Framework, Cybersecurity Test & Evaluation, 정보보호, etc.
E-mail : nodo2@naver.com



신 동 규(Dong-Kyoo Shin)

1986년 서울대학교 컴퓨터학과(공학사)
1992년 Illinois Institute of Technology 대학원 컴퓨터공학과(공학석사)
1997년 Texas A&M University 대학원 컴퓨터학과(공학박사)
1998년~현재 세종대학교 컴퓨터공학과 교수
관심분야 : 머신러닝, 유비쿼터스 컴퓨팅, 생체신호 데이터처리, 정보보호, etc.
E-mail : shindk@sejong.ac.kr