

2016년도 졸업논문

GPR 신호 획득률 개선을 위한 RTS 및 다중 ADC에 관한 연구

A Study on RTS and Multiple ADC for Improvement
of GPR Signal Acquisition Rate

한국산업기술대학교

전자공학과

임영인, 송민지, 윤서영

GPR 신호 획득률 개선을 위한 RTS 및 다중 ADC에 관한 연구

A Study on RTS and Multiple ADC for Improvement
of GPR Signal Acquisition Rate

2017年 2月

한국산업기술대학교

전자공학과

임영인, 송민지, 윤서영

GPR 신호 획득률 개선을 위한 RTS 및 다중 ADC에 관한 연구

指導教授 오 재 곤 (인)

이 論文을 졸업논문으로 提出함.

2016年 12月

한국산업기술대학교

전자공학과

임영인, 송민지, 윤서영

목 차

그림 목 차	i
표 목 차	ii
수식 목 차	ii
국문 요약	iii
제 1 장 서론	1
제 1 절 개발 배경 및 목적	1
제 2 절 개발 목표	8
제 2 장 본론	
제 1 절 시스템 개발을 위한 이론	9
(1) ADC	9
(2) RAM	10
(3) UART	11
제 2 절 시스템 구성	14
(1) 1ksps급 RTS	14
(2) 200Msps급 RTS	24
(3) 100Msps급 RTS 32회 누적	27
(4) 다중 ADC를 이용한 200Msps급 RTS 32회 누적	36
제 3 절 시스템 구현	40
(1) 1ksps급 RTS	41
(2) 200Msps급 RTS	42
(3) 100Msps급 RTS 32회 누적	44
(4) 다중 ADC를 이용한 200Msps급 RTS 32회 누적	47
제 3 장 결론	49
참고 문헌	50

그 립 목 차

그림 1.1.1. GPR 탐사 모식도	1
그림 1.1.2. GPR 탐사 방법	2
그림 1.1.3. GPR장비에 보이는 이미지	2
그림 1.1.4. GPR 측정 장비의 안테나 도식화	3
그림 1.1.5. RTS 모식도	5
그림 1.1.7. 1회의 신호에서 1개의 데이터를 획득	6
그림 1.1.8. 1회의 신호에서 2개의 데이터 획득	6
그림 2.1.1. DPRAM 도식화	10
그림 2.1.2. UART 데이터 패킷 구조	12
그림 2.1.3. UART 데이터 패킷 구조	12
그림 2.2.1. 1ksps급 RTS 시스템 블록도	15
그림 2.2.2. Trigger Control 모듈의 상태도	15
그림 2.2.3. Clock Generator 모듈의 시뮬레이션 결과	16
그림 2.2.4. ADC Model 모듈의 시뮬레이션 결과	16
그림 2.2.5. Clock generator 모듈과 ADC Model 모듈의 상태도	17
그림 2.2.6. RAM Clock generator 모듈의 시뮬레이션 결과	17
그림 2.2.7. Signal generator 모듈의 시뮬레이션 결과	18
그림 2.2.8. RAM Clock generator 모듈과 Signal generator 모듈의 상태도	18
그림 2.2.9. Write RAM 모듈의 시뮬레이션 결과	19
그림 2.2.10. Write RAM 모듈의 상태도	19
그림 2.2.11. RAM 모듈의 시뮬레이션 결과	20
그림 2.2.12. UART Control 모듈의 시뮬레이션 결과	21
그림 2.2.13. UART Control 모듈의 상태도	21
그림 2.2.14. UART TX 모듈의 시뮬레이션 파형	23
그림 2.2.15. UART TX 모듈의 상태도	23
그림 2.2.16. 200Msps급 RTS 시스템 블록도	24
그림 2.3.1. 100Msps급 RTS 32회 누적 시스템 블록도	28
그림 2.3.2. Clock generator 모듈의 상태도	29
그림 2.3.3. ADC Model 모듈의 상태도	30
그림 2.3.4. RAM Clock generator 모듈의 상태도	31
그림 2.3.5. Make end 모듈의 상태도	32
그림 2.3.6. Write RAM 모듈의 상태도1	33
그림 2.3.7. Write RAM 모듈의 상태도2	33
그림 2.3.8. UART Control 모듈의 상태도	36
그림 2.4.1. ADC 4개를 이용한 200Msps급 RTS 32회 누적 시스템 블록도	37
그림 3.3.1. 전체 시스템 테스트에 사용한 GPR 수신기 하드웨어	41
그림 3.3.2. 1ksps급 RTS 시뮬레이션 결과	41

그림 3.3.3. 1ksps급 RTS MATLAB 결과	42
그림 3.3.4. 200Msps급 RTS 시뮬레이션 결과1	43
그림 3.3.5. 200Msps급 RTS 시뮬레이션 결과2	43
그림 3.3.6. 200Msps급 RTS 시뮬레이션 결과3	43
그림 3.3.7. 200Msps급 RTS MATLAB 결과	44
그림 3.3.8. 100Msps급 RTS 32회 누적 시뮬레이션 결과1	45
그림 3.3.9. 100Msps급 RTS 32회 누적 시뮬레이션 결과2	45
그림 3.3.10. 100Msps급 RTS 32회 누적 시뮬레이션 결과3	45
그림 3.3.11. 100Msps급 RTS 32회 누적 MATLAB 결과	46
그림 3.3.12. 100MHz 32회 누적 오실로스코프 파형 결과	46
그림 3.3.13. 다중 ADC를 이용한 200Msps급 RTS 32회 누적 시뮬레이션 결과1	48
그림 3.3.14. 다중 ADC를 이용한 200Msps급 RTS 32회 누적 시뮬레이션 결과2	48

표 목 차

표 1.1.1. 2개의 ADC를 이용한 9개의 채널 신호 처리 과정	3
표 2.3.1. RAM에 저장되어있는 데이터들의 주소	34

수 식 목 차

식 2.2.1.	34
---------------	----

국 문 요 약

GPR은 송신기가 내보낸 임펄스 형태의 전자파가 반사되어 돌아온 신호를 수신하는 지하 투과 레이더이다. 현재 사용하는 GPR 장비는 느린 데이터 처리 속도로 인하여 탐사 장비의 속도가 느린 문제가 있다. 이로 인해 GPR 탐사가 도로 정체를 야기 시킨다.

본 논문은 이러한 문제점을 보완하기 위해서 GPR 수신기의 속도 향상과 유효한 데이터의 획득률을 높이기 위한 방법을 제안하고 이를 연구 및 설계하였다. 연구를 통해 고속 샘플링과 데이터 누적을 이용하여 GPR 장비를 통해 획득한 데이터에서 얻고자하는 신호가 더욱 뚜렷하게 나오도록 하여 유효 데이터의 획득률을 높였다. 또한, ADC 2개가 처리하는 데이터를 4개의 ADC를 통해 ADC 마다 처리하는 데이터의 양을 줄여 데이터 처리 속도를 증가시켰다.

본 연구에서 제안하는 GPR 장비의 속도 향상과 유효 데이터의 획득률 증가를 위한 방법을 연구하였으며, 설계 및 시뮬레이션을 통해 연구한 방법의 구현 가능성을 제시하였다.

제 1 장 서 론

제 1 절 개발 배경 및 목적

지하 투과 레이더(Ground Penetrating Radar, GPR)는 10MHz~ 수GHz 주파수 대역의 임펄스 파형을 이용하여 지하구조를 파악하거나 지하시설물을 측량할 수 있는 레이더이다.[1] GPR 탐사는 아래의 [그림 1.1.1.] 및 [그림 1.1.2.]과 같이 전자기파를 지표면이나 구조물의 표면에 입사시킨 후, 매질 경계면에서 반사되어 되돌아오는 파를 수신하여 [그림 1.1.3.]과 같이 매질 특성을 영상화함으로써 지하 매질에 존재하는 대상물의 위치, 물성, 크기, 경계 등을 찾는 탐사 기법이다. [2][3][4][5] 또한 비교적 최근의 탐사방법으로 지반조사, 구조물의 비파괴 검사, 지하시설물 측량 분야에 뛰어난 적용성을 가진다. 최근 지반이 가라앉아 생기는 싱크 홀의 발생으로 인해 GPR 탐사에 대한 관심이 높아지고 있다.[1]

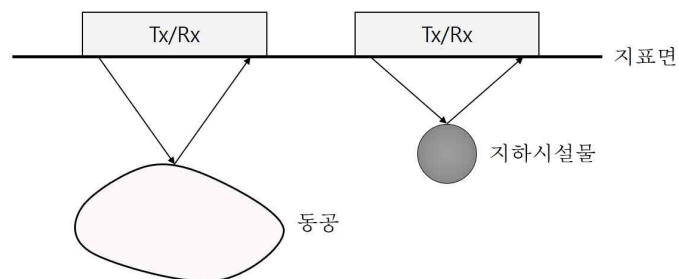


그림 1.1.1. GPR 탐사 모식도

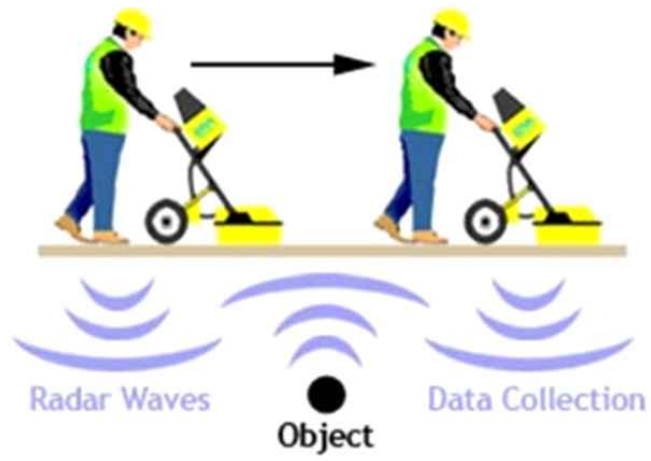


그림 1.1.2. GPR 탐사 방법

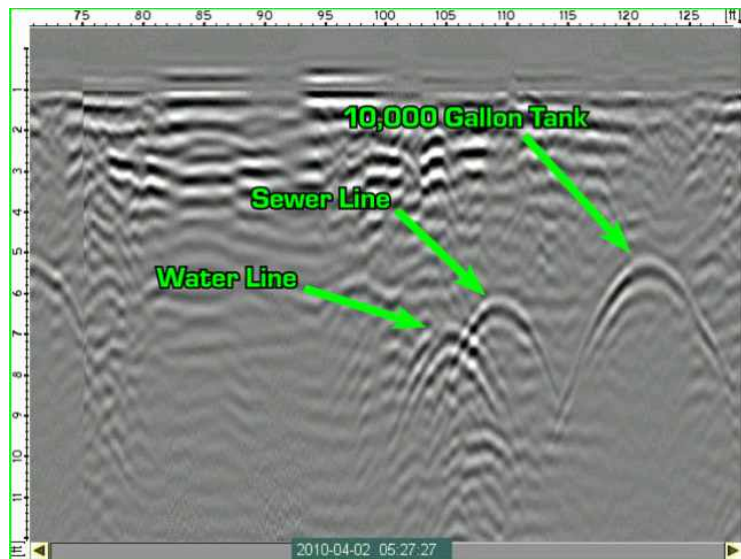


그림 1.1.3. GPR장비에 보이는 이미지

연계기업에서 사용한 GPR 측정 장비의 안테나는 [그림1.1.4]와 같이 8개의 송신 채널과 9개의 수신 채널로 구성된 다중 채널이며 2개의 아날로그-디지털 변환기(Analog To Digital Converter), ADC)를 이용하여 채널을 처리한다. 9개 채널의 신호처리는 MUX를 이용해 신호 처리 할 채널을 선택하고 두 개의 ADC로 동시에 진행한다. 그 과정은 [표1.1.1]와 같다.

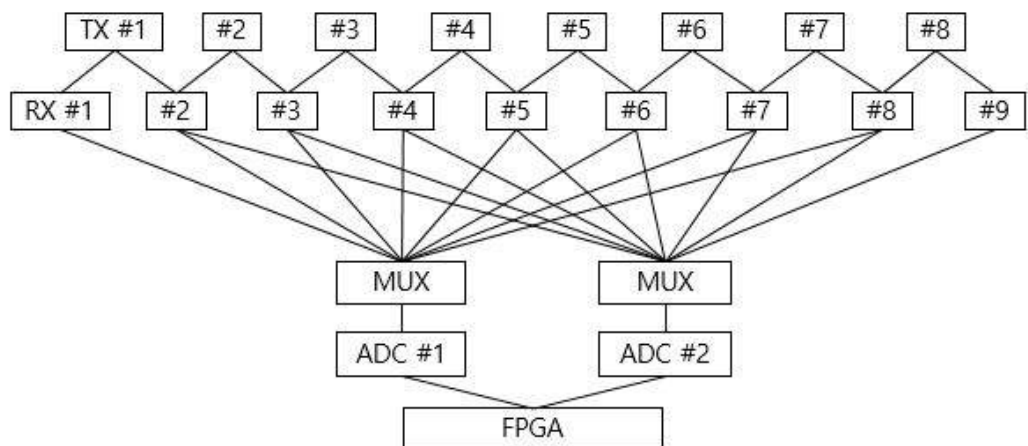


그림 1.1.4. GPR 측정 장비의 안테나 도식화

신호 처리 순서	ADC #1	ADC #2
1	Rx #1	Rx #2
2	RX #3	Rx #2
3	Rx #3	Rx #4
4	Rx #5	Rx #4
5	Rx #5	Rx #6
6	Rx #7	Rx #6
7	Rx #7	Rx #8
8	Rx #9	Rx #8

표 1.1.1. 2개의 ADC를 이용한

9개의 채널 신호 처리 과정

ADC 하나 당 처리하는 채널수가 많을수록 신호처리속도는 저하 된다. 이러한 신호처리 속도의 저하는 탐사 속도의 저하를 의미한다. 차량 장착형 다중 채널 GPR을 이용해 실제 도로에서 탐사하기 위해서는 교통상황에 영향을 끼치지 않는 60km/h 이상의 차량 속도와 고해상도를 유지해야하므로 탐사 속도와 해상도의 개선이 필요하다. 따라서 Gsps급 샘플링을 이용하여 고해상도를 유지하고 GPR 수신기의 신호처리속도를 증가시켜 탐사 속도를 증가시킬 필요성이 있다.

GPR 탐사를 통해 수신한 아날로그 신호를 처리하기 위한 샘플링 방법으로 실시간 샘플링 기법과 한 주기에 다수 데이터를 획득하는 샘플링 기법이 있다.

실시간 샘플링 기법(Real-time Sampling, RTS)은 아래의 [그림1.1.5.]와 같이 1회의 트리거 신호에 의해 원하는 모든 데이터를 얻는 샘플링 방법이다. 한 번의 샘플링 과정으로 얻고자 하는 모든 샘플링 데이터를 얻기 때문에 샘플링 시간이 매우 짧은 것이 특징이다. 또한, GHz급의 샘플링 클럭을 이용하여 샘플링을 하게 된다면 한 번의 샘플링 과정을 통해 발생하는 데이터가 많아지게 된다. 샘플링 클럭의 주파수는 신호를 복원했을 경우에 얼마나 원 신호에 가깝게 복원이 되었는지 결정하는 중요한 요소로, 샘플링 클럭의 주파수가 높을수록 복원된 신호가 원 신호와 유사하게 나오게 되며 고해상도의 이미지를 보는 것이 가능하게 된다. 샘플링 클럭의 주파수를 올리기 위해서는 보다 좋은 클럭 펄스 발생기를 사용하여야 하는데 높은 주파수의 클럭을 발생시키는 것일수록 그 가격이 굉장히 비싸지게 되어 하드웨어를 구성하는 데에 들어가는 비용이 증가하게 된다.

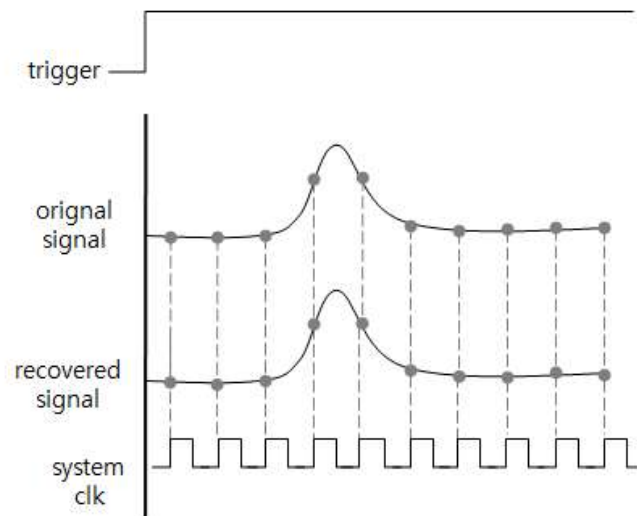


그림 1.1.5. RTS 모식도

ETS(Equivalent-time Sampling)은 트리거 신호가 1회 발생하면 하나 또는 일부의 첫 번째 샘플 데이터를 획득하여 원하는 N번째의 샘플 데이터를 얻을 때까지 샘플링 클럭을 조금씩 일정한 시간만큼 지연시키고 같은 과정을 반복하여 샘플링을 하는 방법이다. 즉, 일정한 시간 차이를 가지고 획득한 데이터들은 그만큼 위상 지연만큼의 해상도를 가지는 ETS가 된다. 예를 들면 1ns의 시간 지연을 가지고 획득한 데이터는 1GHz급의 ETS가 된다. ETS를 이용하여 샘플링을 할 때 샘플링 하는 시간을 줄이기 위해서는 한 번에 획득하는 데이터의 개수를 늘리면 되는데, 아래 [그림1.3.5]는 한 번 샘플링을 할 때 1개의 데이터를 획득하는 ETS이며, [그림1.3.6]는 한번 샘플링을 할 때 2개의 데이터를 획득하는 ETS를 나타낸다. 각 그림의 period는 샘플링을 하는 횟수를 나타낸 것이다.[7] 샘플링을 통해 같은 개수의 샘플 데이터를 얻으려고 할 때 1회에 1개의 데이터를 획득하는 것이 2개의 데이터를 획득하는 것보다 $N/2$ 회 더 소모한다. 이처럼 1회의 샘플링에 획득하는 데이터의 개수가 많을수록 샘플링 시간을 줄이는 것이 가능 하다.

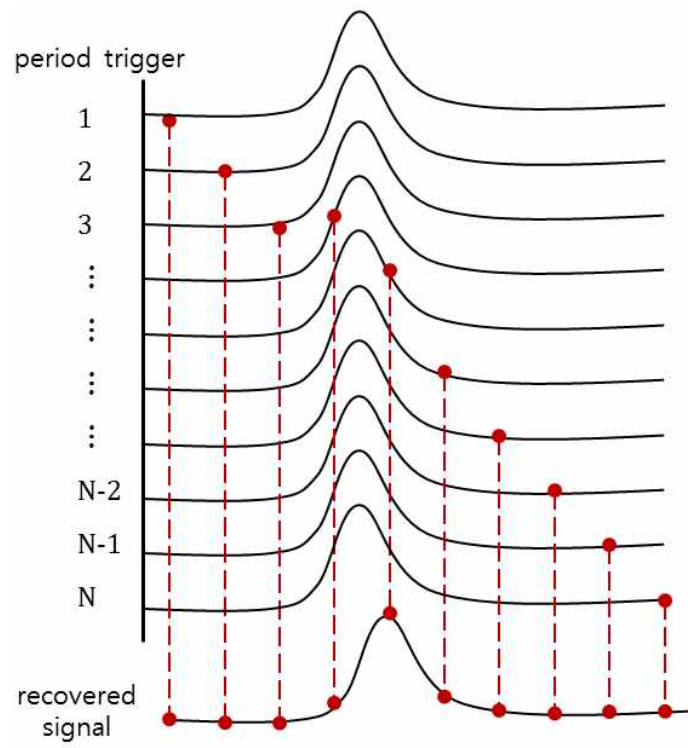


그림 1.1.7. 1회의 신호에서 1개의 데이터를 획득

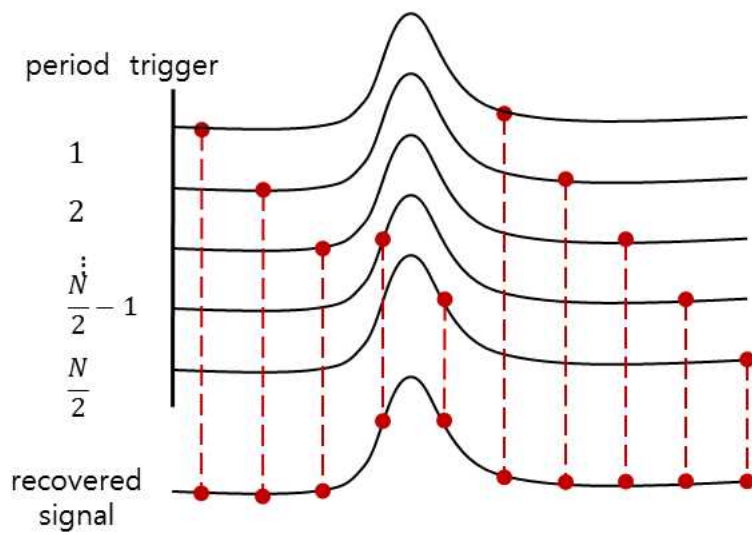


그림 1.1.8. 1회의 신호에서 2개의 데이터 획득

ETS는 RTS와는 다르게 모든 데이터를 얻기 위해서는 여러 번 샘플링을 하기 때문에 RTS보다 샘플링 진행에 걸리는 시간이 길다는 단점을 가지고 있다. 하지만 1회에 다수의 데이터를 획득함으로써 이 단점을 보완할 수 있다. 또한 ps단위의 지연 시간을 가진 ETS는 GHz급 RTS와 동일한 효과를 내는 것이 가능하다. 이러한 방법으로 GHz급 ETS는 클럭 펄스 발생기를 이용한 GHz급 RTS보다 비용이 훨씬 절감되는 방법이므로 하드웨어를 구성하기 위해 필요한 비용이 줄어들게 된다.

제 2 절 개발 목표

본 논문에서는 실제 도로에서 교통상황에 영향을 미치지 않고 GPR 탐사를 하기 위해 GPR 장비가 탑재된 차량이 60km/h의 속도로 운행할 수 있게 개발을 진행한다.

60km/h의 속도로 GPR 탐사를 해도 데이터의 지연현상이 발생하지 않도록 하기 위해 Gbps급 RTS를 구현한다. 또한 ADC 하나당 처리하는 데이터의 양을 줄여 데이터 처리 속도를 높이기 위해서 ADC를 개수를 4개로 증가시킨다. 하드웨어 제작비용을 줄이기 위해서 ETS도 사용하는데, 샘플링에 걸리는 시간을 단축시키기 위해 RTS를 이용한 Multi Point ETS를 구현한다.

이러한 개발을 통해 전체 신호처리 속도의 증가 및 GHz에 해당하는 고해상도 신호 샘플링을 가능하게 하여 GPR 수신기의 신호 획득률 개선 및 해상도 개선을 한다.

제 1 장 본 론

제 1 절 시스템 개발을 위한 이론

(1) ADC

아날로그-디지털 변환기(Analog To Digital Converter, ADC)는 연속 신호를 이산 신호로 바꾸는 샘플링, 이산 신호를 특정한 레벨 값만 갖게 하여 디지털 신호로 만드는 양자화, 그리고 디지털 신호를 디지털 시스템의 처리 방식에 적합한 코드로 표현하는 부호화의 세 과정으로 구성된다.

샘플링은 아날로그 신호를 적절한 시간 간격으로 연속 신호의 순시 값들을 취해 수열 형태의 이산 신호로 바꾸는 동작을 한다. 샘플링 과정에서 샘플링 간격 사이에 있는 아날로그 신호의 정보는 잃어버리게 되므로, 샘플링을 통해 얻은 이산 신호를 원래 연속 신호로 복원하는 것이 중요한 문제이다. 이에 대해 새넨은 샘플링 정리를 통하여 샘플링 주파수가 신호의 최대 주파수의 두 배 이상이면 이론적으로 신호의 복원이 가능하다고 설명하였다.

양자화는 샘플링에 의해 얻은 이산 신호를 크기에 대해서도 이산화 시켜 시간과 크기 모두에 대해 이산적인 값만 가지는 디지털 신호로 바꾸는 과정이다. B+1 비트로 수치를 나타낸다면, 신호 값의 범위를 2^{B+1} 개의 레벨로 나누어 원래 신호의 값 대신에 가장 가까운 양자화 레벨 값으로 대치한다. 양자화까지 마친 디지털 신호는 마지막으로 디지털 시스템이 이해하고 처리할 수 있는 형태의 이진 코드로 만들어지는데, 이를 부호화라고 한다[6].

(2) RAM

랜덤 액세스 메모리(Random Access Memory, RAM)은 임의의 데이터를 임의의 주소에 기억시켜 어느 주소에서라도 같은 속도로 정보를 읽어낼 수 있으며, 전원이 끊어지면 기록된 정보가 휘발되는 휘발성 메모리이다. RAM에 데이터를 저장하기 위해서는 저장할 공간의 주소와 데이터가 필요하고, RAM에 저장된 데이터를 읽어 오기 위해서는 원하는 저장 공간의 주소가 필요하다.

본 논문에서는 시스템 구현을 위해 사용하는 FPGA(Field-Programmable Gate Array) 내부에 DPRAM(Dual-Ported RAM)을 설계하였다. DPRAM은 RAM의 한 종류로, RAM에서 데이터를 읽어 오기 위한 입력포트와 RAM에 데이터를 쓰기 위한 출력포트가 따로 존재한다. 따라서 RAM에서의 데이터 입출력이 동시에 가능하다. 추가로 본 논문에서는 데이터 입출력이 각각 특정 클럭에 동기화되어 동작하기 때문에, 설계한 DPRAM에는 두 개의 클럭 입력포트도 포함되어 있다. 이러한 DPRAM의 구조를 [그림 2.1.1.]에서 볼 수 있다.[8] 클럭에 맞추어 DPRAM에서 데이터 입출력을 하기 위해서는 필요한 주소를 제어해주거나 데이터를 전달해주는 모듈이 필요하다. 이 모듈은 DPRAM에 쓰기 위해 다른 모듈에서 전달받은 데이터, 데이터 입출력에 필요한 주소, 시스템 동작에 필요한 특정 신호를 생성하여 DPRAM에 전달한다.

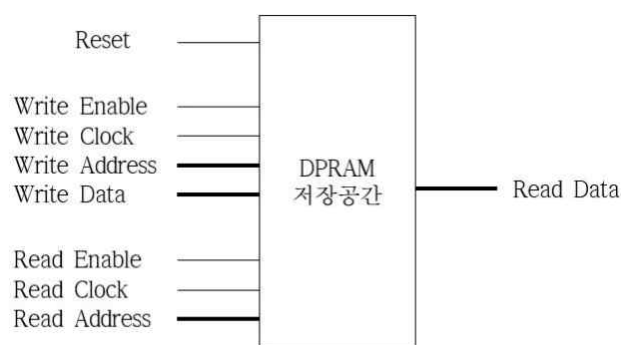


그림 2.1.1. DPRAM 도식화

본 논문에서 설계한 DPRAM에 데이터를 입력하기 위해서는 활성화 된 Enable 신호, 입력 동작의 동기화를 위한 클럭, 데이터를 입력할 주소, 입력할 데이터를 다른 모듈로부터 전달 받아야한다. 이를 이용하여 클럭에 맞춰 전달받은 주소에 전달받은 데이터를 입력한다. 또한 이와 유사하게 DPRAM에서 필요한 데이터를 읽기오기 위해서는 활성화 된 Enable 신호, 읽는 동작의 동기화를 위한 클럭, 읽어오려는 데이터의 주소를 다른 모듈로부터 전달받아야 한다. 이를 이용하여 클럭에 맞춰 전달받은 주소에 입력되어있는 데이터를 읽어 데이터 출력 포트로 출력한다.

RAM을 설계하기 위해서는 읽고 쓰려는 데이터의 길이와 양을 고려해야한다. 본 논문에서는 단계적으로 수행되는 설계에 따라 RAM에 읽고 써야하는 데이터의 길이와 양이 변화하였다. 그에 따라 필요한 저장용량에 맞춰 DPRAM의 설계를 변경하였다.

(3) UART

범용 비동기 송수신기(Universal Asynchronous Receiver-Transmitter, UART)는 비동기 데이터 전송용 인터페이스이다. UART는 데이터 전송을 위해 정해진 데이터 구조가 있는데 아래 [그림 2.1.2]과 같이 구성 되어있다. 가장 왼쪽 비트부터 Start bit, 0 bit부터 7 bit 까지 구성되어있는 Data bit, Parity bit, Stop bit로 되어 있다. UART는 비동기식 통신이기 때문에 데이터를 주고받을 때에 데이터의 시작과 끝을 알아야한다. 이를 위해 ‘START BIT’와 ‘STOP BIT’를 이용하여 데이터 패킷의 시작과 끝을 표현하는데, ‘START BIT’는 1bit, ‘STOP BIT’는 1~2bit로 설정 할 수 있다. ‘START BIT’는 데이터 전송이 없는 idle 상태에서부터 데이터 전송을 시작하는 active 상태가 되었음을 알리기 위해 ‘1’에서 ‘0’으로 변화되는 비트이다. ‘STOP BIT’는 ‘START BIT’와 반대로, active 상태에서 다시 idle 상태로 변경됨을 알리기 위해 ‘1’의 값을 갖는 비트이다. [그림 2.1.2.]와 같이 ‘START BIT’ 다음으로 ‘0 BIT’부터 ‘7 BIT’까지 데이터 비트가 있는데 이 영역이 실제로 전송할 데이터를 나타낸다. [그림 2.1.2.]은 1Byte의 데이터를 전송할 경우의 예시이다. 이 그림

을 보면 ‘7 BIT’ 다음에는 ‘Parity BIT’가 존재한다. 이것은 전송하는 데이터 패킷이 오류 없이 전송되었는지를 체크하는 패리티 비트로, 홀수 패리티와 짝수 패리티가 존재한다. 이 패리티 비트는 선택사항이며, 이 논문에서는 사용하지 않기 때문에 아래 [그림 2.1.3.]와 같이 ‘Parity BIT’가 없는 구조를 갖는다.

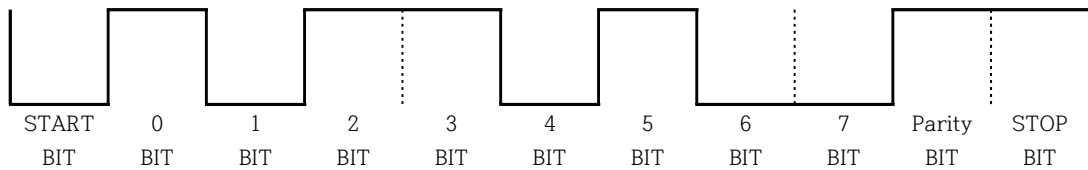


그림 2.1.2. UART 데이터 패킷 구조

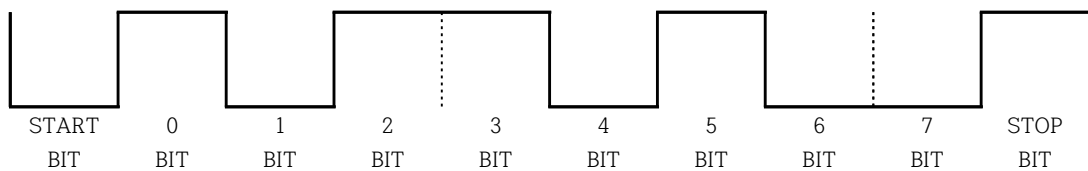


그림 2.1.3. UART 데이터 패킷 구조

데이터를 전송 할 때 데이터를 정렬하는 방식을 엔디언이라 하는데, 이 엔디언은 정렬하는 방법에 따라 보통 빅 엔디언과 리틀 엔디언으로 나누게 된다. 빅 엔디언은 최상위 바이트(MSB, Most Significant Byte)부터 차례대로 저장하는 방식이고, 리틀 엔디언은 최하위 바이트(LSB, Least Significant Byte)부터 차례대로 저장하는 방식이다. 본 논문에서 사용하는 하드웨어는 ‘Altera’사의 cyclone1 이다. LSB 우선의 바이트 정렬방식에 의해서 리틀 엔디언 방식으로 데이터를 정렬 하였다.[9]

데이터를 송수신 할 때에 상호간에 정해진 프로토콜도 중요하지만 상호간의 데이터를 주고받는 통신 속도도 중요하다. Baud rate는 직렬 통신을 하는 UART의 데이터 속도이며 1초 동안에 전송되는 비트의 수를 의미한다. 이것은 $\text{baud rate} = \text{bits/sec} = \text{bps}(\text{bits per second})$ 와 같이 나타낸다. 본 논문에선 115200bps의 baud rate를 사용하였다.

제 2 절 시스템 구성

개발 목표를 위한 시스템을 구현하기 위해, 기본이 될 초기 시스템부터 단계별로 구성하였다. 시스템의 모듈 구성은 기존 GPR 수신기의 사양과 구성에 맞추었다.

(1) 1Ksps급 RTS

1Ksps급 RTS 시스템의 구성을 위해 ‘Trig Control’, ‘Clock generator’, ‘ADC Model’, ‘RAM Clock generator’, ‘Signal generator’, ‘Write RAM’, ‘RAM’, ‘UART Control’, ‘UART TX’ 모듈을 설계하였다.

GPR 수신기 하드웨어 외부에서 입력된 트리거 신호를 Trigger Control 모듈에 통과시켜 이 신호를 이용하는 모든 모듈에 전달한다. 이후 Signal generator 모듈, Clock generator 모듈, RAM Clock generator 모듈 등이 동작을 시작한다. Clock generator 모듈을 통해 만들어진 샘플링 클럭을 ADC Model 모듈 등에서 이용해 아날로그 신호를 샘플링 데이터로 변환한다. Write RAM 모듈은 이 데이터를 전달받아 주소, Enable 신호 등과 함께 RAM에 전달한다. RAM은 Write RAM 모듈에 의해서 저장 공간에 데이터를 쓰거나, UART Control 모듈에 의해서 저장된 데이터를 읽어내는 작업을 한다. UART Control 모듈은 UART TX 모듈에서 전송을 위해 필요한 데이터나 신호를 전달하고, UART TX 모듈은 이를 이용하여 데이터를 외부로 전송한다. UART TX 모듈의 전송이 완료되면 이 시스템의 1회 동작이 완료된 것이다. 이러한 1ksps급 RTS 시스템의 블록도를 아래 [그림 2.2.1.]에서 볼 수 있다.

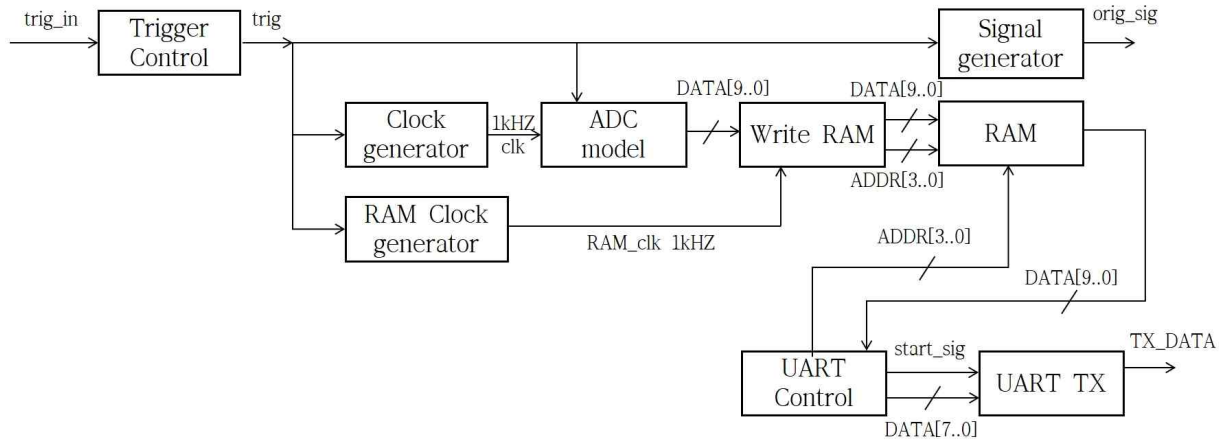


그림 2.2.1. 1ksps급 RTS 시스템 블록도

시스템의 동기를 맞추기 위해 트리거 신호를 발생시켜 일부 모듈 동작의 시작을 제어해야 하는데, 이를 위해 Trigger Control 모듈을 이용한다. 시스템 동작을 위해서는 GPR 수신기 하드웨어에 푸시버튼을 연결하여 외부에서 트리거 신호를 입력해야한다. 이때 물리적으로 연결되는 버튼의 특성상 트리거 신호가 여러 번 튀어서 발생할 수 있다. 트리거 신호가 여러 번 발생하면 전체 시스템 동작이 완전하게 이루어지지 않을 수 있으므로 이를 방지하기 위해 Trigger Control 모듈을 설계하였다. 이 모듈은 푸시버튼을 1회 눌렀을 때 트리거 신호가 1회만 발생할 수 있도록 한다. 이러한 Trigger Control 모듈의 상태 동작을 [그림 2.2.2.]의 상태로 확인할 수 있다.

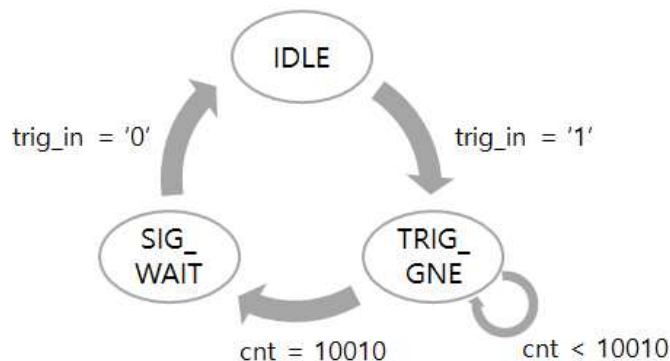


그림 2.2.2. Trigger Control 모듈의 상태도

Clock Generator 모듈은 GPR 수신기의 시스템 클럭(20MHz)을 이용하여 원하는 주파수의 샘플링 클럭을 생성하기 위한 모듈이다. 이 모듈의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, 트리거 신호가 입력되면 COUNT 상태로 이동한다. COUNT 상태에서는 시스템 클럭을 이용하여 만든 샘플링 클럭을 특정 개수(해당 시스템에서는 X=16개)까지 카운트하여 1kHz 샘플링 클럭을 출력하고 IDLE 상태로 이동한다. 이러한 Clock Generator 모듈의 시뮬레이션과 상태 동작을 [그림 2.2.3.]과 [그림 2.2.5.]에서 확인할 수 있다.



그림 2.2.3. Clock Generator 모듈의 시뮬레이션 결과

ADC Model 모듈은 GPR 수신기 하드웨어에서 칩으로 존재하는 ADC의 역할을 대체하며, 시뮬레이션 상에서 RAM의 정상 동작을 확인하기 위해 필요한 데이터를 전달한다. 이 모듈의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, 트리거 신호가 입력되면 COUNT 상태로 이동한다. COUNT 상태에서는 샘플링 클럭의 상승 엣지에 맞춰 10bit의 카운트 값 [0...14,0]을 순차적으로 총 16개의 데이터를 출력하고, 특정 개수(해당 시스템에서는 X=14개)까지 카운트하면 IDLE 상태로 이동한다. 이러한 ADC Model 모듈의 시뮬레이션과 상태 동작을 [그림 2.2.4.]과 [그림 2.2.5.]에서 확인할 수 있다.

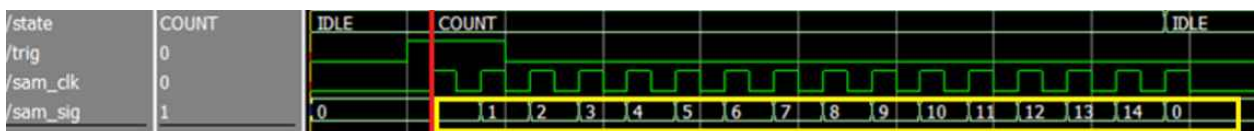


그림 2.2.4. ADC Model 모듈의 시뮬레이션 결과

Clock geneartor 모듈과 ADC Model 모듈은 상태 동작이 동일하므로 아래 [그림 2.2.5.]와 같은 동일한 상태도를 갖는다.

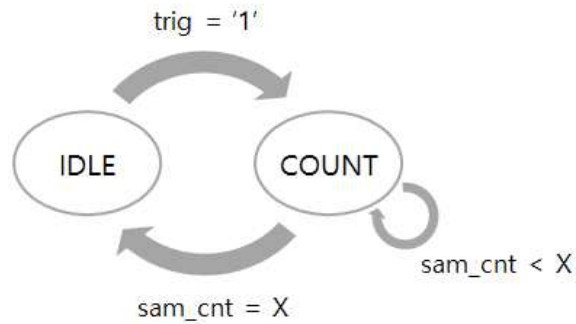


그림 2.2.5. Clock generator 모듈과
ADC Model 모듈의 상태도

RAM Clock generator 모듈은 RAM에서 사용할 클럭을 생성한다. 이 모듈의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, 트리거 신호가 입력되면 COUNT 상태로 이동한다. COUNT 상태에서는 시스템 클럭을 이용하여 1kHz RAM Clock을 출력하며, UART Control 모듈에서 발생하는 RAM END 신호를 전달받게 되면 IDLE 상태로 이동한다. 이러한 RAM Clock generator 모듈의 시뮬레이션과 상태 동작을 [그림 2.2.6.]과 [그림 2.2.8.]에서 확인할 수 있다.



그림 2.2.6. RAM Clock generator 모듈의 시뮬레이션 결과

Signal generator 모듈은 트리거 신호가 입력되면 GPR 수신기 하드웨어 내 ADC에서 변환될 1bit 아날로그 신호를 생성한다. 이 모듈은 시뮬레이션 및 하드웨어에서의 동작 확인을 위한 모듈로, 실제 탐사 시 GPR 수신기에서는 외부에서 수신되는 신호를 ADC에서 변환한다. 이 모듈의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, 트리거 신호가 입력되면 COUNT 상태로 이동한다. COUNT 상태에서는 시스템 클럭을 이용하여 500Hz 신호를 출력하며, RAM END 신호를 전달받게 되면 IDLE 상태로 이동한다. 이러한 Signal generator 모듈의 시뮬레이션과 상태 동작을 [그림 2.2.7.]과 [그림 2.2.8.]에서 확인할 수 있다.



그림 2.2.7. Signal generator 모듈의 시뮬레이션 결과

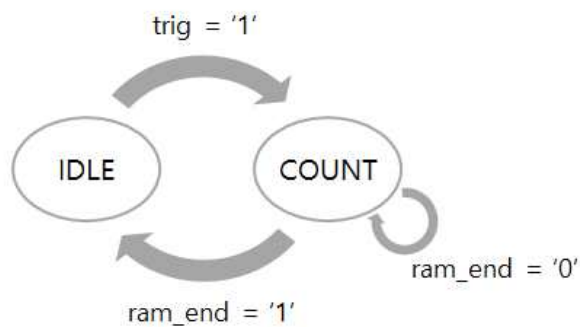


그림 2.2.8. RAM Clock generator 모듈과

Signal generator 모듈의 상태도

Write RAM 모듈은 RAM에 데이터를 쓰기 위해 필요한 주소와 RAM에 입력하기 위해 ADC로부터 전달받은 데이터를 RAM에 전달한다. 또한 데이터를 쓰기 위해 필요한 기타 신호들을 생성하여 RAM에 전달하는 역할을 한다. 이 모듈의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, 트리거 신호가 입력되면 WRITE_CNT 상태로 이동한다. WRITE_CNT 상태에서는 RAM Clock을 이용하여 카운트하고, 이 카운트를 이용하여 Write Enable 신호 및 데이터를 입력할 공간의 주소를 생성하여 RAM 모듈에 전달한다. 특정 개수 (해당 시스템에서는 16개)까지 카운트하면 데이터 쓰기 완료로 의미하는 done 신호를 출력하고 IDLE 상태로 이동한다. 이러한 Write RAM 모듈의 시뮬레이션과 상태 동작을 [그림 2.2.6.]과 [그림 2.2.10]에서 확인할 수 있다.

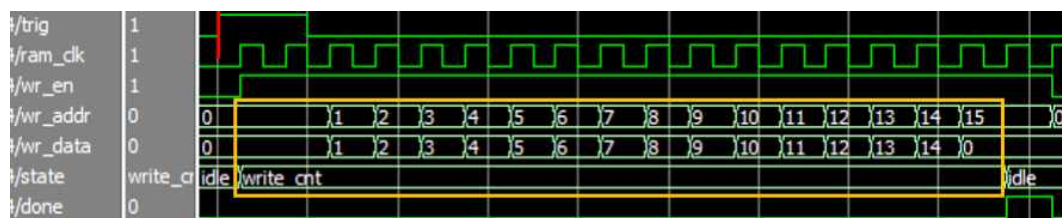


그림 2.2.9. Write RAM 모듈의 시뮬레이션 결과

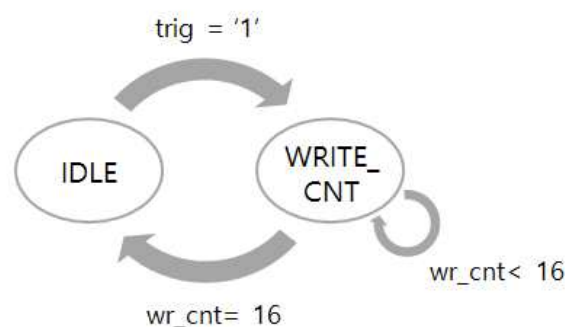


그림 2.2.10. Write RAM 모듈의 상태도

RAM 모듈은 FPGA 내부에 설계한 DPRAM으로, 다른 모듈에서 주어지는 클럭과 주소에 맞추어 데이터를 쓰거나 읽는 역할을 한다. 해당 시스템에서는 10bit 데이터를 16개 저장하므로 160bit의 용량을 가진 DPRAM을 설계하여 사용하였다. 주어진 주소를 이용하여 데이터를 메모리에 쓰는 과정과, 주어진 주소에 저장되어 있는 주소를 읽어 출력하는 과정에는 약간 차이가 있다. 데이터를 쓸 때에는 주소와 데이터가 클럭의 상승 엣지에 맞추어 주어지고, 주어진 데이터를 메모리의 저장하는 과정도 상승 엣지에 맞추어 이루어진다. 그러나 데이터를 읽을 때에는 주소가 클럭의 상승 엣지에 맞추어 주어지지만, 메모리에서 읽어내는 과정은 하강 엣지에 맞추어 이루어진다. 이 두 과정의 차이점과 RAM 모듈의 시뮬레이션을 [그림 2.2.11.]에서 볼 수 있다.

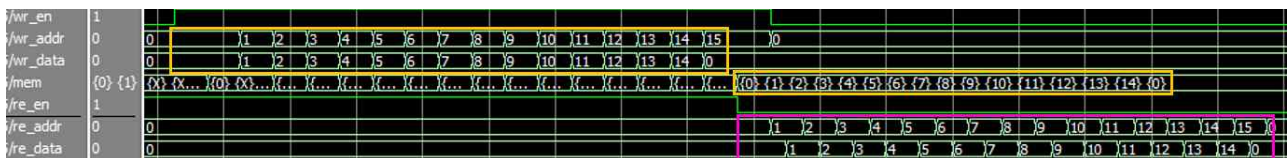


그림 2.2.11. RAM 모듈의 시뮬레이션 결과

UART Control 모듈은 RAM 모듈에서 데이터를 읽어온 후 UART를 통해 외부로 전송하기 위해, 데이터를 8bit로 분할하여 UART TX 모듈로 전달하는 역할을 한다. 이 모듈의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, Write RAM 모듈에서 전달받은 done 신호에 의해 ADDR_INC 상태로 이동한다. ADDR_INC 상태에서는 RAM에 전달할 주소 값을 하나 증가시킨 후 READ_DATA 상태로 이동한다. READ_DATA 상태에서는 RAM에서 읽어온 데이터를 전달받고 LSB_DATA 상태로 이동한다. LSB_DATA 상태와 MSB_DATA 상태에서는 RAM 모듈에서 전달받은 16bit의 데이터를 2개의 8bit 데이터로 분할하고, UART TX 모듈이 작동해도 된다는 의미의 Start Signal과 함께 UART TX 모듈로 전송한다. 또한 UART TX 모듈에서 Busy 신호를 전달받으면 DATACOUNT 상태로 이동한다. DATACOUNT 상태에서는

UART TX 모듈은 UART 프로토콜에 맞추어 데이터를 1bit씩 전송하는 역할을 한다. 이를 이용하여 GPR 수신기에서 처리한 데이터를 외부에 저장하거나 컴퓨터 프로그램 등을 이용하여 시각적으로 확인할 수 있다. 본 논문에서는 처리한 데이터를 컴퓨터에서 매트랩으로 확인하기 위해 UART TX 모듈을 이용하여 컴퓨터로 전송하였다. 이 모듈의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, UART Control 모듈에서 전달받은 Start Signal에 의해 START 상태로 이동한다. START 상태에서는 데이터 전송을 준비하며, 출력 신호 '0' 을 전송하고 바로 SEND 상태로 이동한다. SEND 상태에서는 UART Control 모듈에서 전달받은 8bit의 데이터를 한 bit씩 전송하며 8번 카운트하고, 카운트가 완료되면 PARITY 상태로 이동한다. 해당 시스템에서는 패리티를 사용하지 않으므로 PARITY 상태에서는 출력 신호 '0' 을 전송한 후 STOP 상태로 이동한다. STOP 상태는 데이터 전송 완료 상태를 의미하며 출력 신호 '1' 을 전송하고 IDLE 상태로 이동한다. 이러한 UART TX 모듈의 시뮬레이션과 상태 동작은 [그림 2.2.14.]과 [그림 2.2.15.]에서 확인할 수 있다. 시뮬레이션 파형에서 상단 사각형 내부를 보면, Start Signal을 전달받아 START 상태를 시작으로 전송준비를 하는 것을 확인할 수 있다. 또한 하단 사각형 내부를 보면, 8bit 데이터를 UART 프로토콜에 맞춰 1bit씩 전송하는 것과 UART TX 모듈의 데이터 전송 진행여부를 표시하는 busy 신호의 생성을 확인할 수 있다.

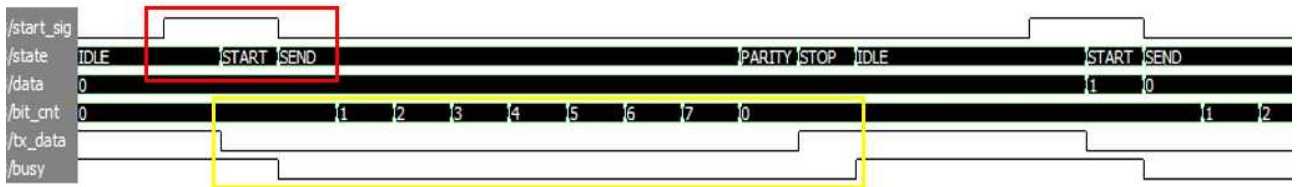


그림 2.2.14. UART TX 모듈의 시뮬레이션 파형

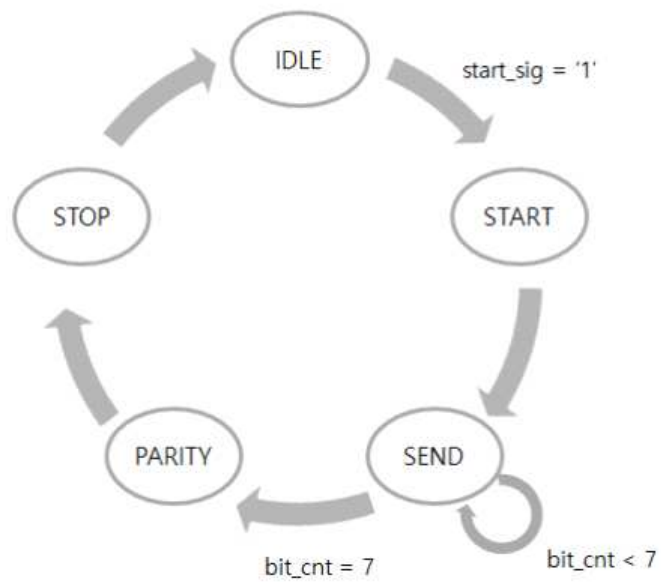


그림 2.2.15. UART TX 모듈의 상태도

(2) 200Msps급 RTS

1ksps급 RTS 시스템을 설계한 후, 더 빠른 신호처리속도와 고해상도의 신호 샘플링을 위해 200Msps급 RTS 시스템을 설계하였다. 200Msps급 RTS 시스템의 구성을 위해 ‘Trig Control’, ‘ALTPLL’, ‘Clock generator’, ‘ADC Model’, ‘RAM Clock generator’, ‘Signal generator’, ‘Write RAM’, ‘ALTDPRAM’, ‘UART Control’, ‘UART TX’ 모듈을 설계하였다.

200Msps급 RTS 시스템은 1ksps급 RTS 시스템과 전체 동작은 유사하다. 그러나 아날로그 신호의 주파수, 샘플링 클럭의 주파수, 처리하는 데이터의 개수 및 용량, 각 모듈에서 다루는 데이터의 비트 수 등에서 변경 사항이 있다. 아날로그 신호의 주파수는 10MHz이고 샘플링 클럭의 주파수는 200MHz를 사용하며, 10bit의 데이터를 2048개 처리한다. 이러한 200Msps급 RTS 시스템의 블록도를 아래 [그림 2.2.16.]에서 볼 수 있다.

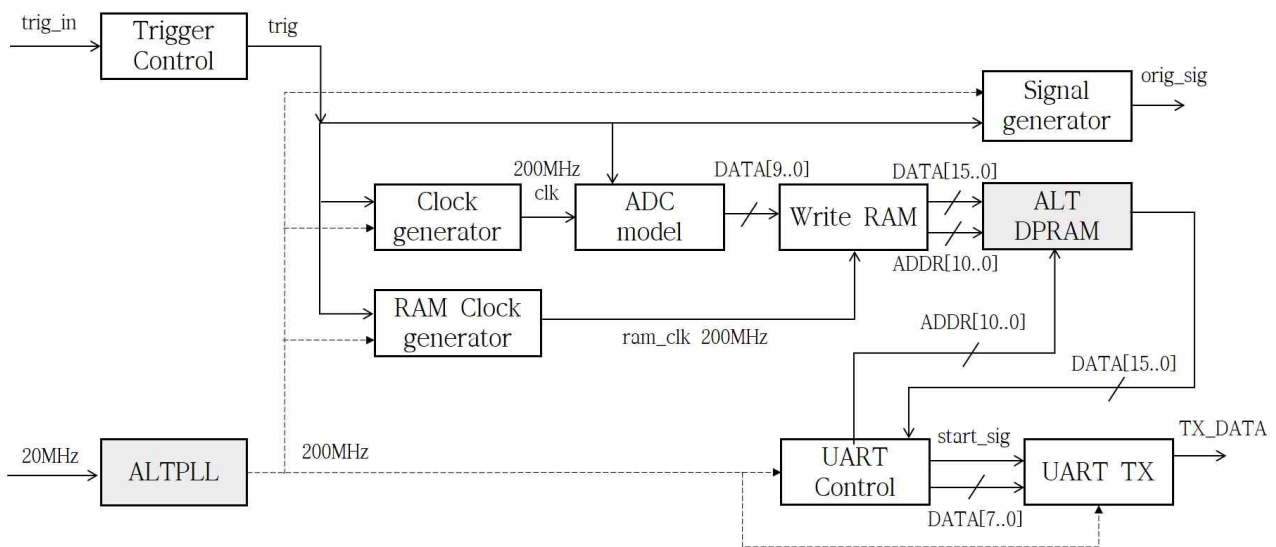


그림 2.2.16. 200Msps급 RTS 시스템 블록도

200Msps급 RTS 시스템에서는 1ksps급 RTS 시스템의 구성에서 ALTPLL 모듈을 추가하고 RAM을 ALTDPRAM 모듈로 대체하였다. ALTPLL과 ALTDPRAM은 Altera에서 제공하는 I/O로, Megawizard를 이용하여 사용자가 필요로 하는 I/O를 불러낼 수 있다. 또한 원하는 형태로 Setting하여 해당 모듈을 사용할 수 있다. 설계 시 주의해야하는 점은 칩 안에서 제공하는 메모리 공간이 한정되어 있고, 그 공간 안에서 설계가 가능하다는 점이다. 한정된 메모리 개수가 초과되면 합성은 가능하지만 배치가 불가능하기 때문에 ALTDPRAM 설계 시 이를 유의하여야한다. ALTPLL은 칩 내부에 전용 공간이 따로 있고, 본 논문에서 사용하는 EPIC3T100C8 칩에서는 사용가능한 ALTPLL의 개수가 한 개로 제한되어 있다.

위상동기회로(Phase-Locked Loop, PLL)은 입력 신호와 출력 신호에서 되먹임 신호와의 위상차를 이용해 출력신호를 제어하는 시스템을 말한다[10]. 입력된 신호를 체배·분주하여 출력 신호의 주파수를 조절하는 것이 목적이며, 입력신호로 기존의 시스템 클럭(20MHz)을 이용하여 출력신호로 새로운 시스템 클럭(200MHz)을 얻기 위해 ALTPLL을 사용하였다. 이 200MHz 클럭을 기존 모듈의 시스템 클럭으로 사용하였고, 이때 로직 간의 시스템 클럭 지연을 최소화하기 위해 FPGA 내부의 클럭 라인 전용 공간인 global buffer를 사용하였다.[8]

200Msps급 RTS 시스템에서는 1ksps급 RTS 시스템에 비해 입출력 데이터의 총 용량이 매우 커졌다. 이에 따라 설계해야하는 RAM의 용량도 커졌지만, FPGA 내에서 사용할 수 있는 메모리 공간의 한계에 이르면서 ALTDPRAM 모듈을 사용하는 것으로 대체하였다. ALTDPRAM에서 입출력하는 데이터 비트 수는 2의 제곱수로만 지정 가능하므로, 10bit의 데이터를 처리하기 위해 16bit의 데이터 입출력 포트를 지정하였다. 또한 2048개의 16bit 데이터 저장을 위해 적합한 용량을 선택하여 ALTDPRAM을 설정하여 사용하였다.

Clock generator 모듈에서는 Trigger Control 모듈에서 전달받은 트리거 신호에 맞추어 200MHz의 새로운 시스템 클럭을 샘플링 클럭으로 ADC Model 모듈에 전달한다. 이 모듈의 상태 동작은 1Ksps급 RTS 시스템에서 Clock Generator 모듈의 상태 동작과 유사하지만, 샘플링 해야 하는 데이터의 개수가 증가하였으므로 카운트해야하는 클럭의 개수가 증가하였다.

ADC Model 모듈은 1ksps급 RTS 시스템에서의 ADC Model 모듈과 역할이 동일하며 상태 동작이 유사하지만, 샘플링 해야 하는 데이터의 개수가 증가하였으므로 카운트해야하는 클럭의 개수가 증가하였다.

RAM Clock generator 모듈은 1ksps급 RTS 시스템에서의 RAM Clock Generator 모듈과 역할 및 상태 동작이 동일하다.

Signal generator 모듈도 1ksps급 RTS 시스템에서의 Signal generator 모듈과 역할 및 상태 동작이 동일하다. 그러나 출력하는 1bit 아날로그 신호를 10MHz의 주파수로 변경하였다. 이는 GPR 수신기 하드웨어에 부착된 ADC 칩의 사양에 더 적합한 아날로그 신호를 입력하기 위함이다.

Write RAM 모듈은 1ksps급 RTS 시스템에서의 Write RAM 모듈과 역할이 동일하며 상태 동작이 유사하다. 그러나 처리해야하는 데이터의 개수가 증가하였으므로 RAM 클럭을 이용한 카운트의 개수가 증가하였고, 데이터 입출력을 위해 필요한 주소의 개수가 더 증가하였으므로 주소 포트의 용량도 증가하였다.

ALTDPRAM 모듈은 위에서 언급한 바와 같이 1ksps급 RTS 시스템에서 설계하였던 RAM 모듈을 대체한 것이다. ALTDPRAM 모듈은 사양을 설정하여 배치하게 되면 자체적으로 작동 과정이 설정된다. 따라서 모듈 설정 시 그림파일로 제공하는 과형을 확인하여 모듈이 어떻게 작동하는지 이해하고 다른 모듈의 설계를 고려해야한다.

UART Control 모듈은 UART 통신을 하기 위해서 RAM에서 읽어온 데이터를 8bit씩 나누어 UART에 차례로 보내는 모듈이다. 데이터는 리틀 엔디안 방식으로 정렬하고 LSB부터 데이터를 전송한다. 앞서 설명했던 1ksps급 RTS와 동일한 모듈을 사용하였다.

UART TX 모듈은 1ksps급 RTS 시스템에서의 UART TX 모듈과 역할 및 상태 동작이 동일하며, 1ksps급 RTS와 같은 모듈을 사용하였다.

(3) 100Msps급 RTS 32회 누적

200Msps급 RTS 시스템을 설계한 후, 데이터 처리 결과에서 잡음을 제거하기 위해 누적을 사용하여 해당 시스템을 구성하였다. 사용가능한 ALTDPRAM의 최대 용량을 고려하여, 60개의 데이터를 32번 누적하여 1,920개의 데이터를 처리 할 수 있는 100Msps급 RTS 시스템을 설계하였다. 이 누적은 이전에 설계했던 100Msps급 RTS 시스템을 변형하여 설계하였다. 100Msps급 RTS 및 데이터 60개의 32회 누적 시스템의 구성을 위해 ‘Trig Control’, ‘ALTPLL’, ‘Clock generator’, ‘ADC Model’, ‘RAM Clock generator’, ‘Signal generator’, ‘Write RAM’, ‘ALTDPRAM’, ‘Make end’, ‘UART Control’, ‘UART TX’ 모듈을 설계하였다.

100Msps급 RTS 및 32회 누적 시스템은 200Msps급 RTS 시스템과 전체 동작의 형태는 비슷하다. 그러나 샘플링 클럭의 주파수, 처리하는 데이터의 개수 및 용량, 누적 기능추가 등에서 변경 사항이 있다. 샘플링 클럭의 주파수는 100MHz를 사용하고 60개의 10bit의 데이터를 처리한다. 또한 32회 누적을 위해 필요한 모듈을 추가적으로 설계하여 구성 하였다. 이러한 100Msps급 RTS 32회 누적 시스템의 블록도를 아래 [그림 2.3.1.]에서 볼 수 있다.

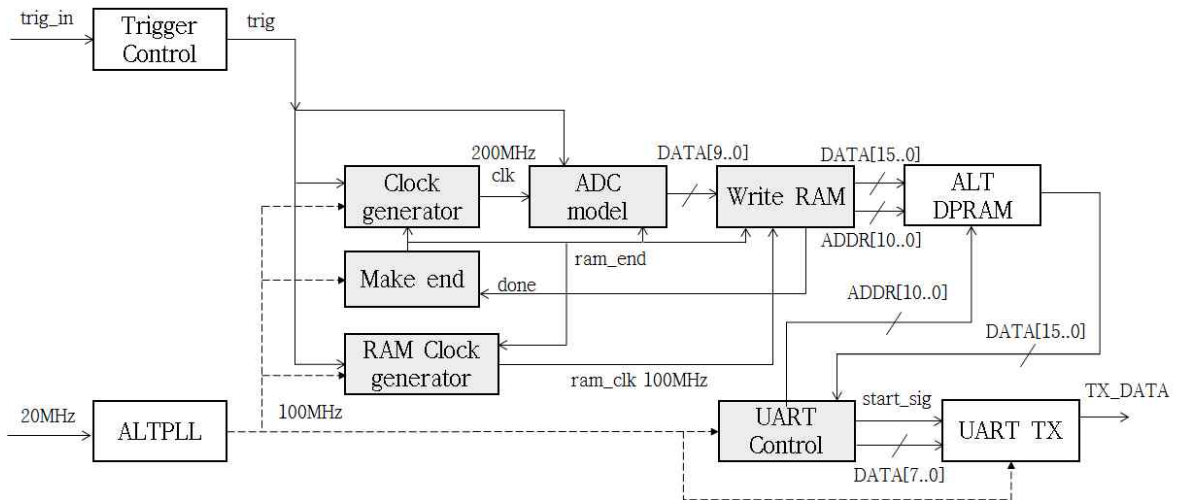


그림 2.3.1. 100Msps급 RTS 32회 누적 시스템 블록도

Clock generator 모듈은 Trigger Control 모듈에서 전달받은 트리거 신호에 맞추어 100MHz의 ADC에서 사용할 클럭을 생성한다. 이 모듈의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, 트리거 신호가 입력되면 COUNT 상태로 이동한다. COUNT 상태에서는 샘플링 클럭을 특정 개수(해당 시스템에서는 60개)까지 카운트하여 100MHz 샘플링 클럭을 출력한다. 그리고 이 과정을 한 번으로 카운트하여 카운트 값이 31 미만이면 WAIT_32 상태로 이동하고, 그 값이 31이 되면 IDLE 상태로 이동하여 총 32번의 누적을 진행할 수 있도록 하였다. WAIT_32 상태에서는 RAM의 쓰기가 끝났다는 의미의 Make end 모듈에서 보내줄 RAM END 신호를 위해 대기하며, 신호를 전달받으면 다시 COUNT 상태로 돌아가 샘플링 클럭을 출력하는 과정을 수행한다. 이러한 Clock generator 모듈의 상태 동작을 [그림 2.3.2.]의 상태도로 확인할 수 있다.

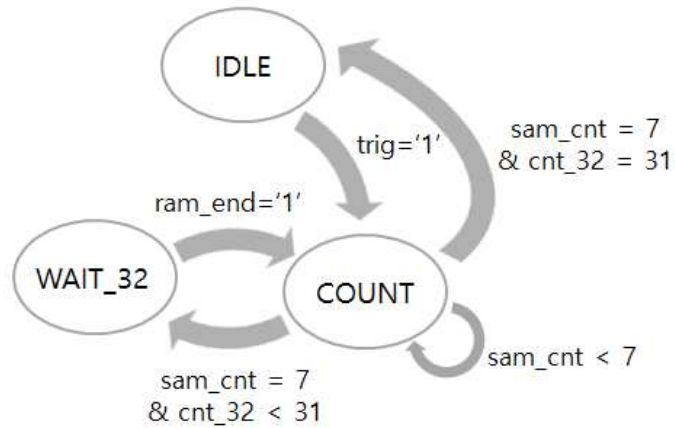


그림 2.3.2. Clock generator 모듈의 상태도

ADC Model 모듈은 앞서 설계한 두 가지 시스템에서의 ADC Model 모듈과 역할이 동일하다. 이 모듈의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, 트리거 신호가 입력되면 COUNT 상태로 이동한다. COUNT 상태에서는 60개의 10bit 데이터를 출력하기 위해 1부터 60까지 카운트를 하며, Write RAM 모듈로 done 신호를 받으면 WAIT_32 상태로 이동한다. WAIT_32 상태는 32회 누적을 위해 대기하는 상태로, COUNT 상태에서 이동해오면서 증가하는 카운트가 31 미만이면 상태를 유지하고 31이 카운트되면 IDLE 상태로 이동한다. 또한 이 상태에서 Make end 모듈로부터 RAM END 신호를 받게 되면 COUNT 상태로 다시 이동하게 된다. 이러한 ADC Model 모듈의 상태 동작을 [그림 2.3.3.]의 상태도로 확인할 수 있다.

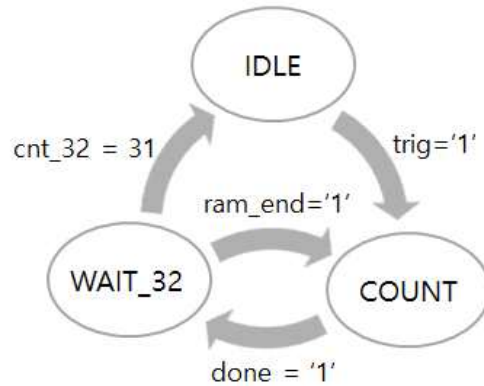


그림 2.3.3. ADC Model 모듈의 상태도

RAM Clock generator 모듈은 ALTDPRAM에서 사용할 클럭을 생성한다. 이 모듈의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, 트리거 신호가 입력되면 COUNT 상태로 이동한다. COUNT 상태에서는 시스템 클럭을 이용하여 100MHz RAM Clock을 출력하며, Write RAM 모듈로부터 60개의 데이터 한 세트가 입력되었다는 의미로 '1'의 done 신호를 전달받으면 WAIT_1 상태로 이동한다. WAIT_1 상태에서는 Write RAM 모듈의 데이터 1회 쓰기 과정이 완료되었다는 의미로 done 신호가 '0'으로 바뀐 것을 확인한 후 WAIT_2 상태로 이동하게 된다. WAIT_2 상태는 32회 누적을 위해 대기하는 상태로, 카운트 값이 31이 되면 IDLE 상태로 이동한다. 그러나 31 미만이면 RAM END 신호를 위해 대기하다가 신호를 전달받으면 다시 COUNT 상태로 이동하여 RAM Clock을 출력한다. 이러한 RAM Clock generator 모듈의 상태 동작을 [그림 2.3.4.]의 상태로 확인할 수 있다.

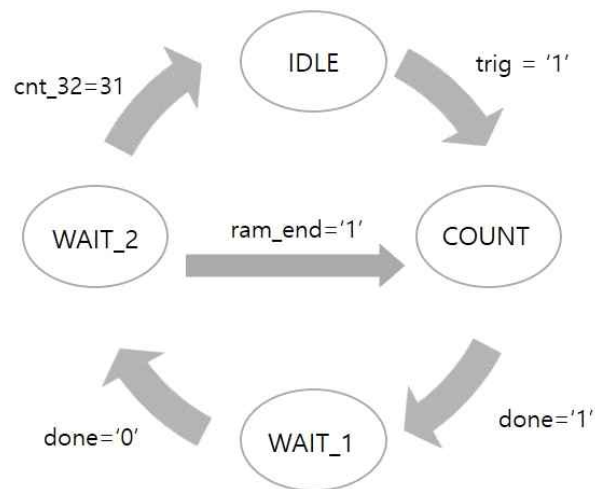


그림 2.3.4. RAM Clock generator 모듈의 상태도

Signal generator 모듈은 200Msps급 RTS 시스템의 Signal generator 모듈의 역할 및 상태 동작이 동일하다.

Make end 모듈은 해당 시스템에서 32회 누적을 위해 추가 설계된 모듈이다. Write RAM 모듈로부터 한 데이터 세트의 입력 완료를 의미하는 done 신호를 전달받으면, 다음 데이터를 생성 및 출력해도 된다는 의미로 RAM END 신호를 출력한다. 이 모듈의 초기 상태인 WAIT_D1 상태는 '1' 인 done 신호를 기다리는 상태로, '1' 인 done 신호를 전달받게 되면 WAIT_D0 상태로 이동한다. WAIT_D0 상태는 누적횟수가 32회 미만일 때 done 신호가 '0' 으로 복귀되기를 기다리는 상태로, 내부 카운터를 이용하여 일정 길이의 RAM END 신호를 출력한 뒤 WAIT_D1 상태로 이동한다. 이러한 Make end 모듈의 상태 동작을 [그림 2.3.5.]의 상태도로 확인할 수 있다.

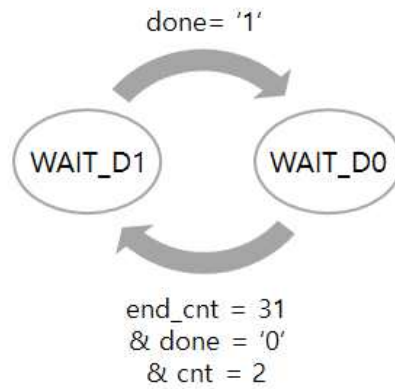


그림 2.3.5. Make end 모듈의 상태도

Write RAM 모듈은 해당 시스템에서 누적을 위해 두 가지 상태머신으로 설계하였다. 첫 번째 상태머신의 초기 상태인 IDLE 상태에서는 모듈의 신호들이 초기화되며, 트리거 신호가 입력되면 WRITE_CNT 상태로 이동한다. WRITE_CNT 상태에서는 RAM Clock을 이용하여 60개의 데이터를 카운트하고 DONE_DELAY 상태로 이동한다. 또한 WRITE_CNT 상태에서는 ALTDPRAM에 데이터를 입력하기 위해 필요한 Enable 신호, 주소를 생성하여 ALTDPRAM에 전달한다. 또한 ADC Model 모듈에서 10bit로 전달받은 데이터를 16bit로 변환하여 ALTDPRAM에 전달한다. DONE_DELAY 상태에서는 60개 데이터를 32회 누적한 개수인 1,920이 모두 카운트되었으면 IDLE 상태로 이동하고, 카운트 값이 1,920 미만이면 WAIT_END 상태로 이동한다. WAIT_END 상태에서는 전체 데이터가 1,920로 모두 카운트되었다면 바로 IDLE 상태로 이동한다. 그러나 카운트가 1,920 미만인 상태에서는 RAM END 신호가 전달될 때까지 상태를 유지하고, 신호를 전달받으면 WRITE_CNT 상태로 이동하여 과정을 반복하게 된다. 두 번째 상태머신의 초기 상태인 IDLE_2 상태에서는 첫 번째 상태머신에서 1,920개의 데이터를 모두 카운트했을 때 생성되는 done flag 신호를 전달받는다. 이 신호가 '1'이 되면 모든 데이터의 입력이 완료되었다는 의미의 all done 신호가 생성되고 DONE_OUT 상태로 이동한다. DONE_OUT 상태에서는 UART Control 모듈에서 전달받는

Clock Out 신호가 '1'로 입력되면 초기 상태인 IDLE_2 상태로 돌아가게 된다. 이러한 Write RAM 모듈의 상태 동작을 [그림 2.3.6.]과 [그림 2.3.7.]의 상태로 확인할 수 있다.

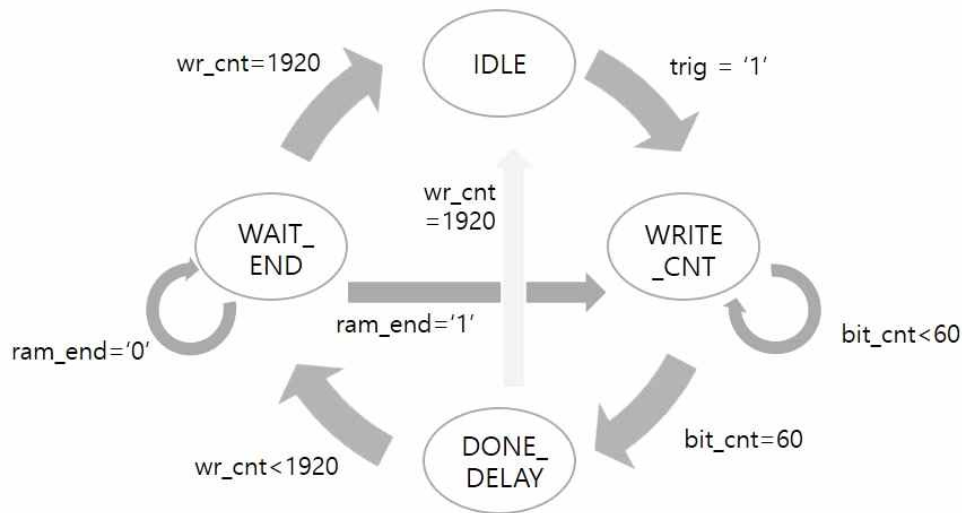


그림 2.3.6. Write RAM 모듈의 상태도1

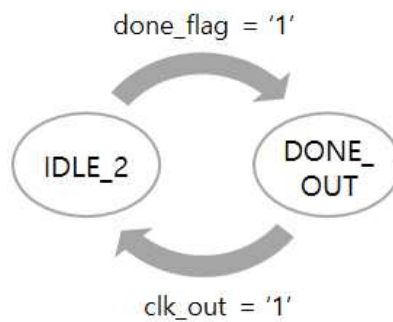


그림 2.3.7. Write RAM 모듈의 상태도2

ALTDPRAM 모듈은 60개의 데이터를 32회 누적한 총 1,920개의 데이터를 저장하기 위해 2,048bit의 용량으로 설정하였다.

UART Control 모듈은 앞서 설명했던 1ksps급 RTS처럼 UART 통신을 위해 RAM에서 꺼내 온 데이터를 8bit씩 나누어 UART에 차례로 보내는 모듈이다. 하지만 여기서는 누적 기능이 추가가 되었기 때문에, RAM에 저장되어있는 데이터들 중에서 필요한 데이터들을 꺼내와 평균필터를 이용하여 데이터를 추출하게 된다. 여기서 평균필터는 모든 데이터들의 합을 합한 후 합한 데이터의 개수만큼 나누어준다. 아래의 식(2.1)은 N개의 데이터를 평균 필터한 것이다.

$$\frac{data1 + data2 + data3 + \dots + data(N-1) + data(N)}{N} \quad \text{식 2.1.1.}$$

data : 데이터, *N*: 데이터 수

RAM에 저장되어있는 데이터들의 주소를 표로 표현을 하였는데 아래 [표 2.3.1]과 같다.

세트 줄	1	2	3	4	5	6	7	...	31	32
1	0	60	120	180	240	300	360	...	1800	1860
2	1	61	121	181	241	301	361	...	1801	1861
3	2	62	122	182	242	302	362	...	1802	1862
4	3	63	123	183	243	303	363	...	1803	1863
5	4	64	124	184	244	304	364	...	1804	1864
6	5	65	125	185	245	305	365	...	1805	1865
...
60	59	119	179	239	299	359	419	...	1859	1919

표 2.3.1. RAM에 저장되어있는 데이터들의 주소

[표 2.3.1.]에서 ‘세트’는 트리거 신호 1회에 의해서 들어온 60개의 데이터의 묶음을 의미한다. 각 세트의 세로줄에 쓰인 번호는 샘플링 데이터들이 RAM에 저장되었을 때 부여 받은 주소이며, 총 32번의 누적을 위해 32번 트리거 신호를 이용하여 32세트의 샘플링 데이터를 저장한다. ‘줄’은 32회의 트리거 입력에 의해 샘플링을 하는 동안, 동일한 순서의 샘플링 한 클럭에 의해 샘플링 된 데이터의 묶음을 의미한다. 각 줄의 가로줄에 쓰인 번호들이 동일한 순서의 샘플링 한 클럭에 의해 샘플링 된, 같은 값의 샘플링 데이터가 저장되어 있는 RAM의 주소이다. 예를 들어, ‘세트’ 1번에 있는 0~59번 주소는 처음 트리거 신호가 들어왔을 때 생성된 샘플링 데이터들이 10bit씩 각 주소에 저장되어있는 것이고, ‘줄’ 1번의 0, 60, 120, 180, 240, 300, 360, ..., 1800, 1860번 주소는 모두 각 트리거 신호가 들어왔을 때 첫 샘플링 클럭으로 생성된 10bit의 샘플링 데이터들이 저장되어있는 주소이다.

이 모듈은 아래 [그림 2.3.8.]과 같이 구성되어 있다. RAM에서 모든 데이터의 저장이 완료되었다는 all done 신호의 값으로 ‘1’이 들어오면 상태 동작이 시작된다. ADDR_INC를 이용하여 RAM에 필요한 주소를 요청하고, READ_DATA에서 RAM을 통해 데이터를 받는다. 그 후 SUM에서 이전 데이터들과 받은 데이터들을 더하는 작업을 32번 반복한다. 모두 더한 값을 누적 횟수인 32로 나누어 LSB 우선으로 전송하는 방식을 이용하여 LSB 데이터 8bit를 전송한다. 그 후 UART_TX 모듈에서 받은 데이터의 전송이 완료된 상태가 확인이 되면 남은 MSB 데이터를 UART_TX 모듈에 전송한다. ADDR_INC 상태부터 MSB_DATA 상태까지 위에서 설명했던 내용과 같이 60번 반복하면 마지막에 요청하는 주소의 값이 1979이 된다. 주소의 값이 1979가 되면 모든 샘플링 데이터 값을 전송한 것으로 보고, Control 모듈의 동작은 IDLE 상태로 넘어가 다음 all done 신호가 들어올 때까지 기다리게 된다.

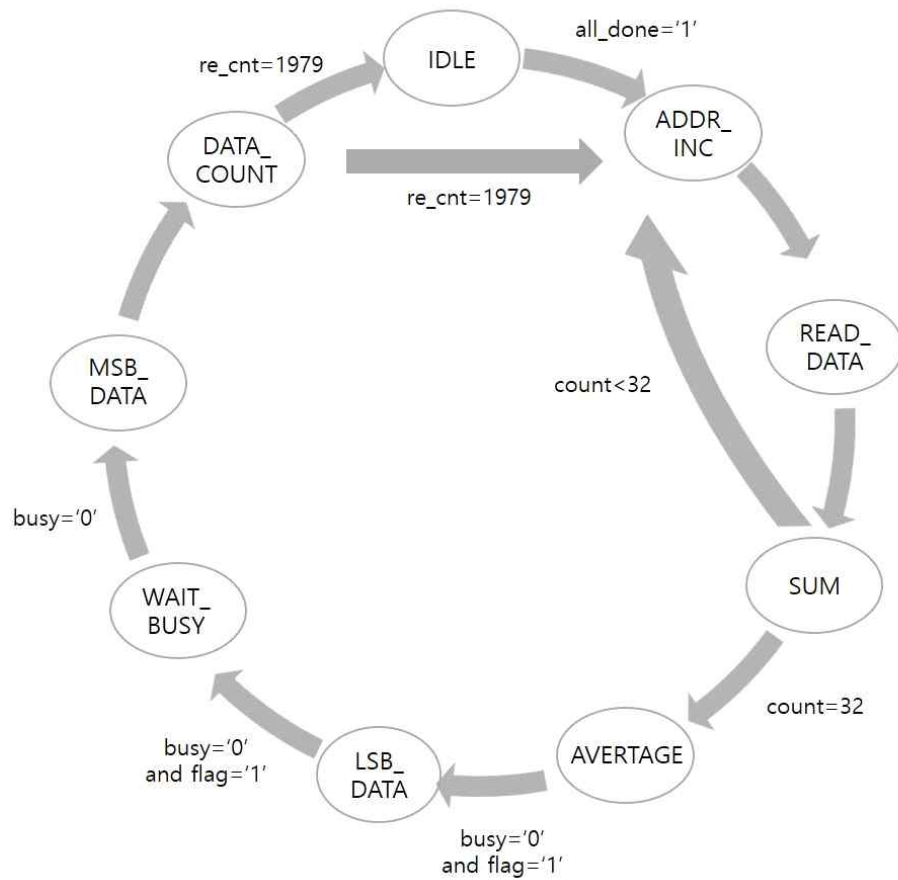


그림 2.3.8. UART Control 모듈의 상태도

UART TX 모듈은 데이터 전송을 위해 사용하는 모듈로, 1ksps급 RTS 시스템에서 사용한 UART TX 모듈과 같은 모듈이다.

(4) 다중 ADC를 이용한 200Msps급 RTS 32회 누적

32회 누적 시스템을 설계한 후, 데이터의 샘플링 속도를 증가시키기 위해 1개의 ADC를 이용하여 진행한 조건에서 4개의 ADC를 사용하여 시스템을 구성하였다. 모든 조건은 32회 누적 시스템과 동일하게 60개의 데이터를 32번 누적하여 데이터를 처리하도록 하고 샘플링 주파수는 200MHz를 이용하여 설계하였다.

이 시스템을 구성하기 위해 ‘Trig Control’, ‘ALTPLL’, ‘Clock generator’, ‘ADC Model 1’, ‘ADC Model 2’, ‘ADC Model 3’, ‘ADC Model 4’, ‘RAM Clock generator’, ‘Signal generator’, ‘Write RAM 1’, ‘Write RAM 2’, ‘Write RAM 3’, ‘Write RAM 4’, ‘ALTDPRAM 1’, ‘ALTDPRAM 2’, ‘ALTDPRAM 3’, ‘ALTDPRAM 4’, ‘Make end’, ‘UART Control’, ‘UART TX’ 모듈을 설계하였다.

전체 시스템은 100Msps급 RTS를 이용한 데이터 32회 누적 시스템과 동일하지만 ADC의 개수가 4개로 증가하면서 ADC에서 데이터를 받아 메모리에 저장하는 데에 영향을 미치는 ‘Write RAM’ 과 ‘ALDDPRAM’ 모듈이 함께 4개로 증가하여 각 모듈에 들어가는 신호를 제어하기 위해 믹스와 디믹스를 이용하여 설계하였다. 이렇게 설계한 ADC 4개를 이용한 200Msps급 RTS 32회 누적 시스템의 블록도는 아래 [그림 2.4.1.]이다. 설계한 모듈의 수가 많아 블록도의 가독성을 위해 클럭에 관련된 Trig Control 모듈, Clock generator 모듈, RAM Clock generator 모듈을 생략하여 첨부하였다.

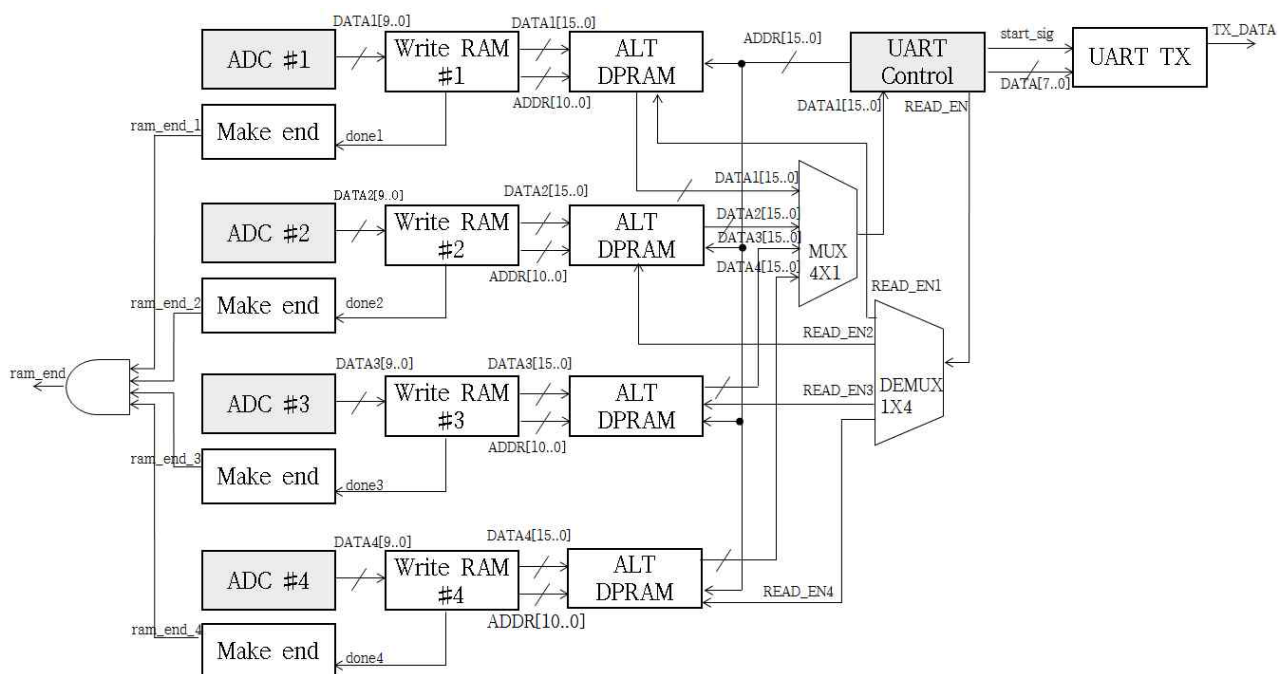


그림 2.4.1. ADC 4개를 이용한 200Msps급 RTS 32회 누적 시스템 블록도

Clock generator 모듈은 Trigger Control 모듈에서 전달받은 트리거 신호에 맞추어 200MHz의 ADC에서 사용할 클럭을 생성하는 모듈이다. 이 모듈은 앞서 설계한 (3) 100Msps 급 RTS 32회 누적 시스템과 동일한 모듈을 사용하였다.

ADC Model 모듈은 앞서 설계한 3) 100Msps급 RTS 32회 누적 시스템과 동일한 모듈을 사용하였다. 전체 시스템의 특성상 4개의 ADC를 사용하기 위해 모듈을 총 4번 사용하였다. 시뮬레이션의 결과 데이터를 한 눈에 확인 할 수 있도록 각 ADC마다 나오는 데이터의 값을 다르게 설정하였다.

RAM Clock generator 모듈은 ALTDPRAM에서 사용할 클럭을 생성한다. 이 모듈은 앞서 설계한 100Msps급 RTS 32회 누적 시스템의 RAM Clock generator모듈과 동일한 모듈을 사용하였다.

Signal generator 모듈은 트리거 신호가 입력되면 GPR 수신기 하드웨어 내 ADC에서 변환될 아날로그 신호를 생성한다. 이 모듈은 앞서 설계했던 100Msps급 RTS 32회 누적 시스템의 Signal generator 모듈과 동일한 모듈을 사용하였다.

Make end 모듈은 해당 시스템에서 32회 누적을 위해 추가 설계된 모듈이다. Write RAM 모듈로부터 한 데이터 세트의 입력 완료를 의미하는 done 신호를 전달받으면, 다음 데이터를 생성 및 출력해도 된다는 의미로 RAM END 신호를 출력한다. 앞서 설계한 100Msps급 RTS 32회 누적 시스템의 Make end 모듈과 같은 모듈을 사용하였다.

Write RAM 모듈은 ALTDPRAM에 들어갈 데이터를 제어하는 모듈이다. 데이터를 메모리에 저장하기 위해 필요한 Enable 신호, 저장할 데이터, 클럭 신호를 ALTDPRAM에 보내준다. 이 모듈은 앞서 설계한 100Msps급 RTS 32회 누적 시스템의 Write RAM 모듈과 같은 모듈을 사용하였다. 이 모듈은 동일한 Write RAM 모듈을 4번 사용하여 설계하였다.

ALTDPRAM 모듈은 60개의 데이터를 32회 누적한 총 1,920개의 데이터를 저장하기 위해 2,048bit의 용량으로 설정하였다. 이 모듈은 Write RAM 모듈과 동일하게 동일한 모듈을 4번 사용하여 설계하였다.

UART Control 모듈은 UART 통신을 위해 ALTDPRAM에서 꺼내온 데이터를 평균 필터를 통과시켜 UART 모듈로 데이터를 전송해주는 모듈이다. 앞서 100Msps급 RTS 32회 누적 시스템에서 설계한 UART Control모듈과 동일하게 사용하지만 ADC의 개수가 증가함에 따라 ALTDPRAM의 데이터를 순차적으로 가져와야한다. 이를 위하여 ALTDPRAM에 들어가는 Enable 신호를 데이터를 가져오하고자 하는 ALTDPRAM에 주어 원하는 곳에서 데이터를 가져올 수 있게 설계하였다. 또한, 원하는 ALTDPRAM을 선택하는 기준이 되는 신호를 이 모듈에서 주는 것으로 ALTDPRAM의 읽는 동작을 제어하였다.

UART TX 모듈은 데이터 전송을 위해 사용하는 모듈로, 1ksps급 RTS 시스템에서 사용한 UART TX 모듈과 같은 모듈이다.

제 3 절 시스템 구현

제 3 절 시스템 구현에서는 앞의 제 2 절 시스템 구성에서 만든 각각의 모듈을 모두 통합한 동작한 결과를 보여준다. 통합한 전체 시스템을 Modelsim^{주1)} 프로그램을 이용하여 컴파일하고 시뮬레이션을 통해 시스템 구현 결과를 확인하였다. 그 후 Quartus^{주2)} 프로그램을 이용하여 GPR 수신기 하드웨어에 프로그램을 전송하고 전체 시스템 동작을 테스트 하였다. 전체 시스템 과정을 거쳐 GPR 수신기 하드웨어 외부로 전송된 샘플링 데이터가 원 신호와 유사하게 샘플링 되었는지 확인 작업을 하기 위해 MATLAB^{주3)} 프로그램을 이용하였다.

하드웨어를 통해 전체 시스템 동작을 테스트할 때는 Modelsim으로 시뮬레이션을 확인할 때와는 다르게 ADC Model 모듈을 제거하고 테스트한다. 이는 GPR 수신기 하드웨어에 ADC가 칩으로 존재하여 ADC Model 모듈을 시뮬레이션 용도로만 설계하였기 때문이다.

이 테스트를 위해 사용한 하드웨어는 아래의 [그림 3.3.1.]을 통해 확인할 수 있다. 그림과 같이 브레드보드에 스위치를 연결하여 외부에서 트리거 신호를 입력 할 수 있도록 하였다. 스위치를 누르게 되면 시스템 설계에 의해 원 신호가 FPGA의 I/O핀에서 나오게 된다. 그 신호가 ADC에 입력되어 샘플링 클럭에 의해 샘플링 된 데이터로 출력된다. 이 샘플링 데이터는 UART선을 통해 컴퓨터로 전송되고, 이 데이터를 저장하여 MATLAB에서 그래프로 그려 샘플링 결과를 확인하는 테스트를 진행하였다.

주1) Mentor Graphics사에서 만든 HDL 설계 소프트웨어

주2) Altera에서 제공하는 논리설계 소프트웨어

주3) MathWoks 사에서 개발한 수치 해석을 위한 공학용 소프트웨어

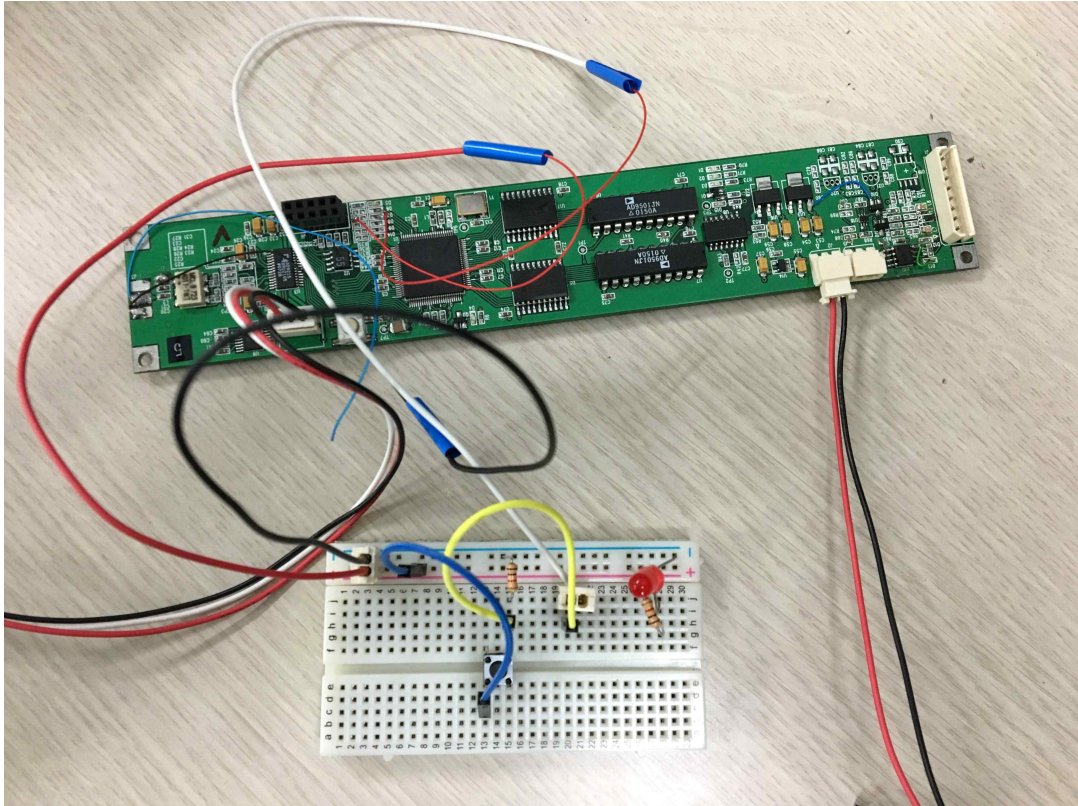


그림 3.3.1. 전체 시스템 테스트에 사용한 GPR 수신기 하드웨어

(1) 1ksps급 RTS

10MHz의 원 신호를 1kHz의 샘플링 클럭을 이용하여 샘플링 하였다. 샘플링 된 데이터가 RAM에 쓰이는 과정, 데이터를 RAM에서 읽어내는 과정, 데이터가 UART TX로 전송되는 과정을 아래 [그림 3.3.2.]를 통해 볼 수 있다.

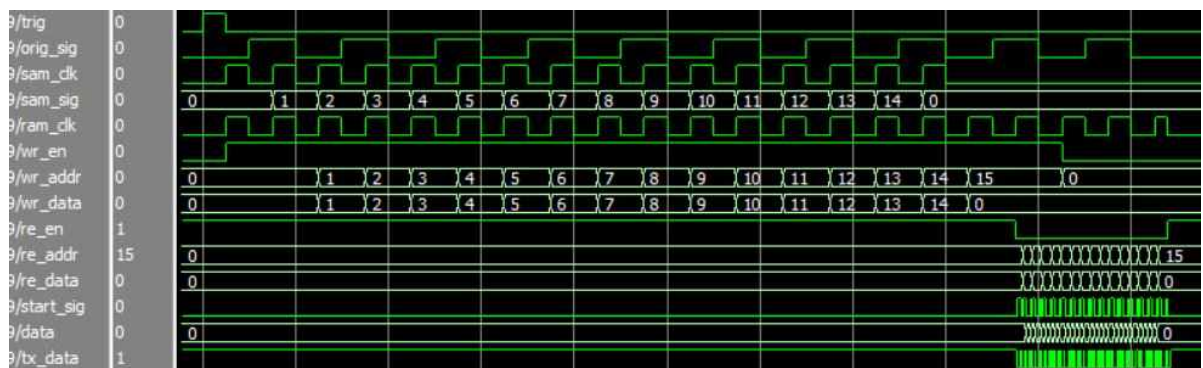


그림 3.3.2. 1ksps급 RTS 시뮬레이션 결과

전체 시스템의 시뮬레이션 결과를 바탕으로 GPR 수신기 하드웨어에서 처리되어 컴퓨터로 전송된 데이터를 MATLAB을 이용하여 그래프로 그려보았다. 그 결과를 아래의 [그림 3.3.3.]에서 확인할 수 있다.

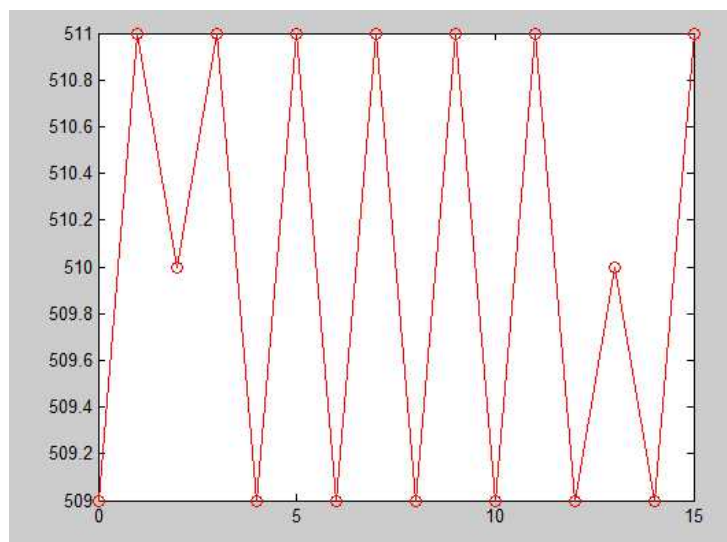


그림 3.3.3. 1ksps급 RTS MATLAB 결과

(2) 200Msps급 RTS

10MHz의 원 신호를 200MHz의 샘플링 클럭을 이용하여 샘플링 하였다. 샘플링 된 데이터가 RAM에 쓰이는 과정, 데이터를 RAM에서 읽어내는 과정, 데이터가 UART TX로 전송되는 과정을 아래 [그림 3.3.4.], [그림 3.3.5.], [그림 3.3.6.]을 통해 볼 수 있다. 샘플링 데이터의 개수가 많아 전체 시뮬레이션의 가시성 문제로, 시뮬레이션 결과 중 주요 부분의 그림을 첨부하였다.

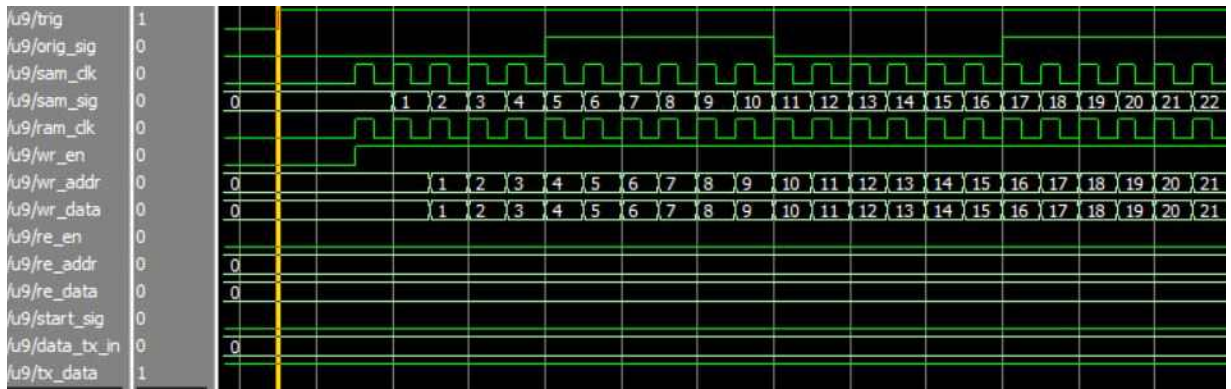


그림 3.3.4. 200Msps급 RTS 시뮬레이션 결과1

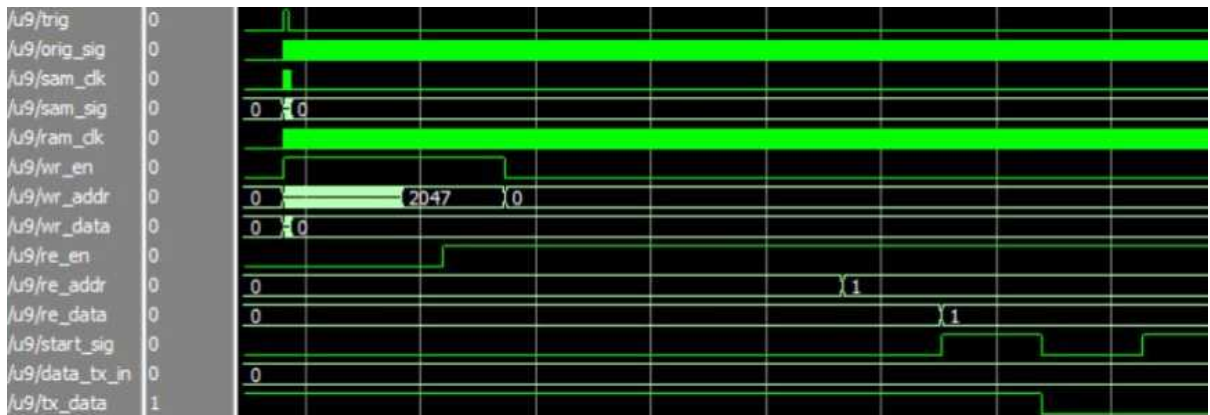


그림 3.3.5. 200Msps급 RTS 시뮬레이션 결과2

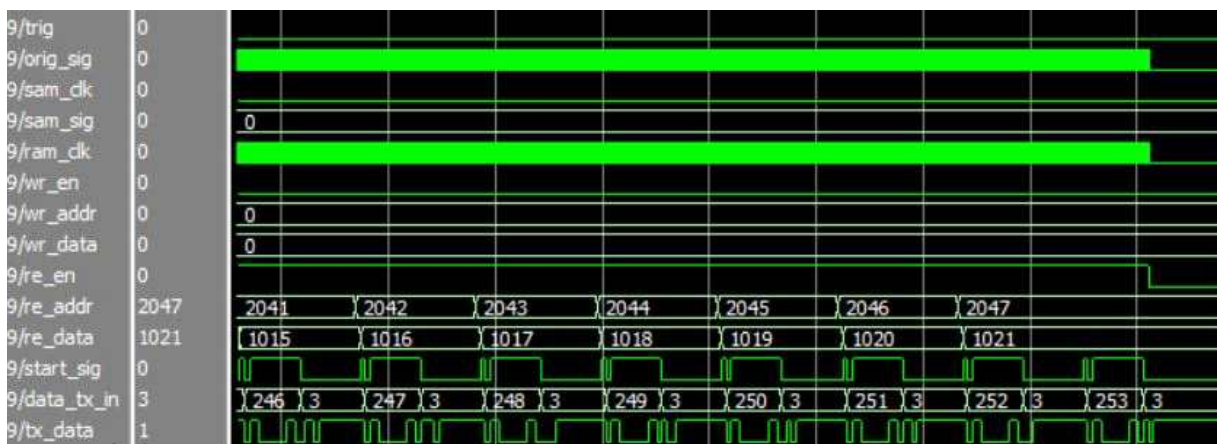


그림 3.3.6. 200Msps급 RTS 시뮬레이션 결과3

[그림 3.3.4.]에서는 원 신호의 샘플링과 RAM에 데이터가 입력되는 과정의 시작부분을 보여준다. [그림 3.3.5.]에서는 RAM에 데이터가 입력되는 과정의 완료의 부분을 보여준다. 또한 RAM에서 데이터를 읽고 UART TX로 전송하는 과정의 시작부분을 보여준다. [그림 3.3.6.]에서는 RAM에서 데이터를 읽고 UART TX로 전송하는 과정의 완료부분을 보여준다.

전체 시스템의 시뮬레이션 결과를 바탕으로 GPR 수신기 하드웨어에서 처리되어 컴퓨터로 전송된 데이터를 MATLAB을 이용하여 그래프로 그려보았다. 그 결과를 아래의 [그림 3.3.7.]에서 확인할 수 있다. 위의 [그림 3.3.3.]과 비교하여, 샘플링 클럭을 증가시키면 원 신호에 더 가깝게 복원되어 해상도가 높아지는 것을 확인 할 수 있다.

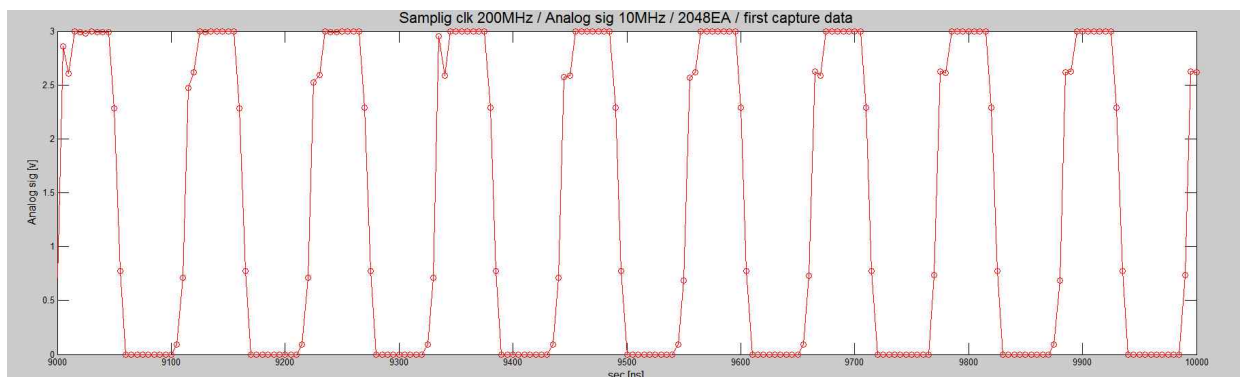


그림 3.3.7. 200Mps급 RTS MATLAB 결과

(3) 100Mps급 RTS 32회 누적

10MHz의 원 신호를 100MHz의 샘플링 클럭을 이용하여 샘플링 하였다. 60개의 데이터를 32회 누적하여 총 데이터의 개수가 1920개가 되어야하지만 가독성을 높이기 위해 8개의 데이터를 32회 누적한 결과를 그림으로 첨부하였다. 전체 시스템의 시뮬레이션 결과를 아래의 [그림 3.3.8.], [그림 3.3.9.], [그림 3.3.10.]에서 확인할 수 있다. [그림 3.3.8.]은 원 신호를 샘플링하여 얻은 샘플 데이터를 ALTDPRAM에 0번 주소부터 차례로 입력된 결과이다.

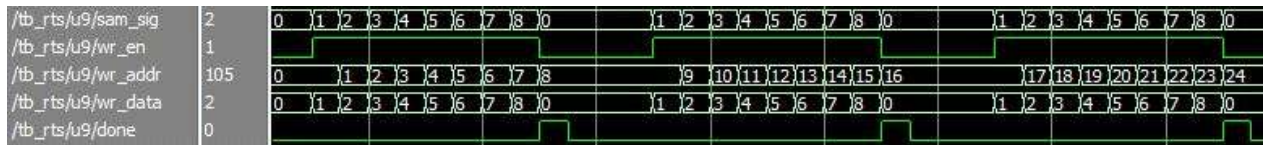


그림 3.3.8. 100Msps급 RTS 32회 누적 시뮬레이션 결과1

[그림 3.3.9.]은 ALTDPRAM에서 읽어온 데이터를 평균 필터를 통과시켜 UART TX로 전송한 것이다. [그림 3.3.9.]의 data_tx_in 신호는 UART Control 모듈에서 UART TX로 UART전송시킬 데이터를 넘겨주는 변수이다. 데이터를 읽어온 후 평균 필터를 통과시키는 시간보다 UART 전송시간이 굉장히 짧아 [그림 3.3.9.]같이 데이터가 안보이게 된다. 중간에 보이지 않는 데이터를 [그림 3.3.10.]과 같이 [그림 3.3.9.]의 앞부분 데이터를 확대하여 data_tx_in에 전송되는 값을 확인 하였다. 확인한 결과 0x00, 0x01, 0x00, 0x02의 데이터가 리틀엔디안 방식을 이용하여 0x01, 0x00, 0x02, 0x00으로 데이터가 전송된 것을 알 수 있다.

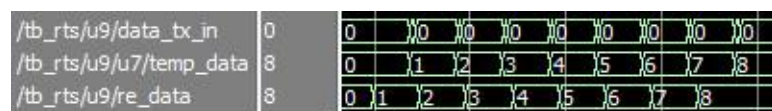


그림 3.3.9. 100Msps급 RTS 32회 누적 시뮬레이션 결과2



그림 3.3.10. 100Msps급 RTS 32회 누적 시뮬레이션 결과3

전체 시스템의 시뮬레이션 결과를 바탕으로 GPR 수신기 하드웨어에서 처리되어 컴퓨터로 전송된 데이터를 MATLAB을 이용하여 그래프로 그려보았다. 그 결과를 아래의 [그림 3.3.11.]에서 확인할 수 있다. [그림 3.3.11.]을 보면 복원된 신호가 [그림 3.3.7.]에 비해 원 신호에 가깝게 복원되지 않았는데, 이는 샘플링 개수가 2048개에서 60개로 비교적 많이 감소했기 때문이다.

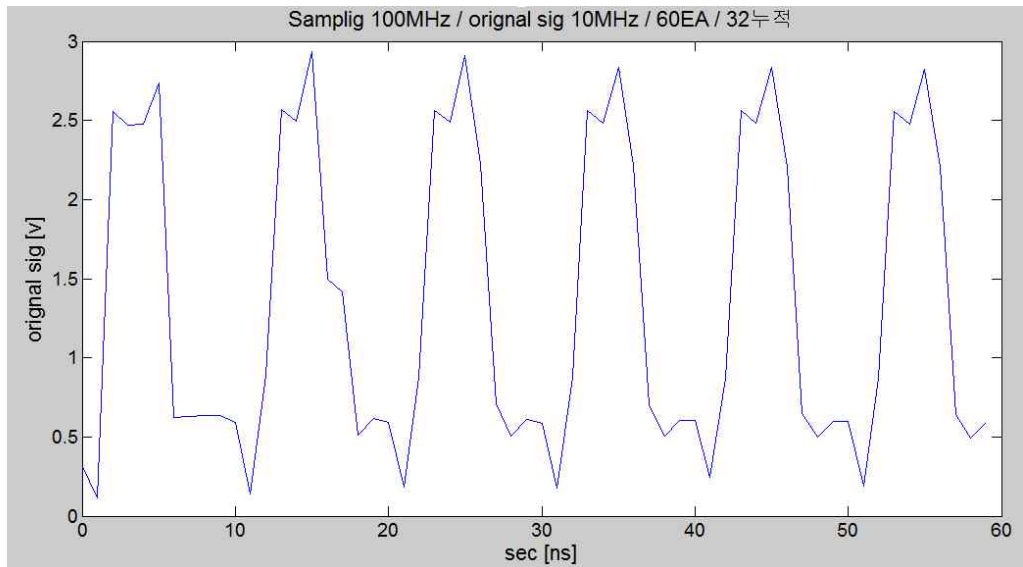


그림 3.3.11. 100Mps급 RTS 32회 누적 MATLAB 결과

100Mps급 RTS 32회 누적 시스템을 하드웨어에 적용하여 트리거 신호 및 원 신호를 오실로스코프 파형으로 관찰하였다. [그림 3.3.12.]을 통해 트리거 신호에 동기를 맞추어 원 신호가 발생하는 것을 확인할 수 있다.

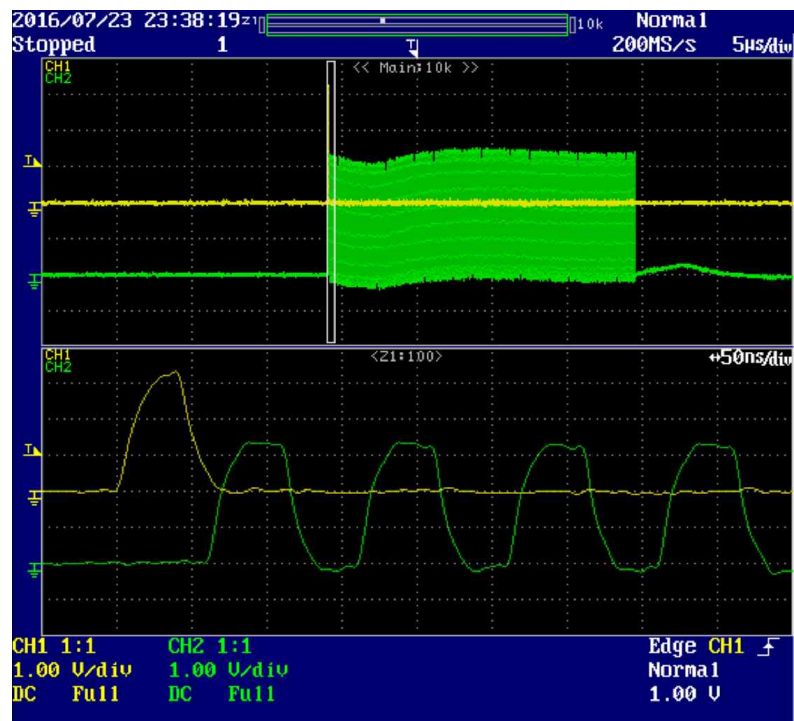


그림 3.3.12. 100MHz 32회 누적 오실로스코프 파형 결과

(4) 다중 ADC를 이용한 200Msps급 RTS 32회 누적

10MHz의 원 신호를 200MHz의 샘플링 클럭을 이용하여 샘플링 하였다. 60개의 데이터를 32회 누적하여 총 데이터의 개수가 1920개가 되어야하지만, 가독성을 높이기 위해 8개의 데이터를 32회 누적하였다. 전체 시스템의 시뮬레이션 결과는 아래의 [그림 3.3.13.]과 같으며, 이 그림과 같이 ALTDPRAM에 데이터를 쓰게 된다. 1번 ALTDPRAM부터 4번 ALTDPRAM까지 데이터가 들어가는 모습으로, 각 ALTDPRAM의 0번 주소부터 데이터가 들어가는 것을 확인하였다.

[그림 3.3.14.]는 [그림 3.3.13.]에서 확인한 ALTDPRAM에 쓰여진 데이터를 읽어오는 과정이다. 1번 ALTDPRAM의 모든 데이터를 불러와 평균 필터를 거쳐 UART로 전송하면서 다음 2번 ALTDPRAM의 데이터를 가져오는 작업을 한다. 그 결과, [그림 3.3.13.]의 데이터가 [그림 3.3.14.]에 있는 데이터와 일치하는 것을 확인하였다.

제 4 장 결 론

본 논문에서는 GPR 장비가 탑재된 차량의 탐사 속도 증가와 해상도 개선을 위해 Gsps급의 RTS 및 ETS를 구현하고 ADC의 개수를 4개로 증가시키는 것을 목표로 연구한 졸업 작품에 대해 기술하였다.

최종적으로 100Msps급 RTS 32회 누적 시스템 및 평균필터를 이용한 데이터 처리 시스템까지 구현하였고, 이를 통해 복원된 신호를 그래프로 나타내었다. 또한 4개의 ADC를 이용해 샘플링 데이터를 처리하는 시스템을 구현하여 시뮬레이션 결과를 확인하였다.

구현한 시스템을 하드웨어에 적용하여 테스트를 진행하는 과정에서 ADC에 입력되는 원신호에 노이즈가 발생하는 문제점을 발견하였다. 또한 FPGA 칩 내 사용가능한 비트 및 로직 수의 한계로 인해 총 샘플링 데이터 개수를 제한하게 되었다. 차 후, 용량이 더 큰 FPGA를 이용한다면 더 많은 샘플링 데이터를 처리하는 시스템의 구현이 가능할 것으로 기대된다.

RTS 및 다중 ADC를 통해 GPR 수신기의 전체 신호처리 속도가 증가하고 GHz에 해당하는 고해상도 신호 샘플링이 가능해질 것이다. 이러한 개선이 싱크홀 및 지하시설물 등의 정확한 탐지와, GPR 장비가 탑재된 차량의 탐사 속도 증가를 가능하게 할 것으로 예상된다.

참 고 문 헌

- [1] 이강원, 손호웅, “지형 공간정보체계 용어사전”, 구미서관, 2016. 01. 03.
- [2] 최종성, “ETS와 AGC를 적용한 GPR용 고속 샘플링 수신기 구현”, 한국산업기술대학교
산업기술·경영대학원 : 전자공학과, 2008. 12.
- [3] Worksmart, Inc.

(<http://www.worksmartinc.net/>)
- [4] 한국정보통신기술협회, IT용어사전

(<http://word.tta.or.kr/main.do>)
- [5] Dig Smart of Maine, Inc.

(<http://digsmartofmaine.com/gpr.php>)
- [6] 윤호근, “ETS를 이용한 13.3GHz 디지털 샘플러 구현”, 한국산업기술대학교 지식기반
기술·에너지대학원 : 정보통신 전공, 2011. 06.
- [7] 이철희, 「디지털 신호 처리:기본 이론부터 MATLAB 실습까지」,한빛아카데미, 2013, p150
- [8] ALTERA, Internal Memory (RAM and ROM) User Guide, 2014. 07

https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/an/an207.pdf
- [9] ALTERA, cyclone FPGA Family Data Sheet, 2003. 04, ver.1.2

https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ds/ds_cyc.pdf
- [10] 위상동기회로 - 위키백과

<https://ko.wikipedia.org/wiki/%EC%9C%84%EC%83%81%EB%8F%99%EA%B8%B0%ED%9A%8C%EB%A1%9C>