

# SDR을 위한 W-CDMA 업링크 소프트웨어 모뎀 구현

## Implementation of W-CDMA Uplink Software Modem for SDR

백동명(D.M. Baek) SDR연구팀 연구원  
조권도(K.D. Joh) SDR연구팀 선임연구원  
김진업(J.U. Kim) SDR연구팀 책임연구원, 팀장

다양한 이동통신기기들을 한 시스템에 수렴시킬 수 있는 기술로서 SDR 기술이 각광받고 있다. 본 논문은 W-CDMA 물리계층 업링크의 트래픽 채널을 DSP로 구현하여 베이스밴드 프로세싱 하는 것을 목적으로 한다. 이러한 소프트웨어 모뎀은 초기화, 소스 데이터 발생, 스프레딩, 스크램블링, 출력단 등으로 이루어진다. 기존의 FPGA, ASIC 등으로 구현된 하드웨어 모뎀을 소프트웨어적인 DSP로 구현할 때 생기는 주요 문제들을 고찰하였다. 로드 밸런싱, 동시성과 실시간성, 버퍼 스킴, 멀티 태스킹, 인터럽트 관리, OVFS 및 스크램블링 코드의 복소수 연산 등이다. 전통적인 구조는 FPGA와 DSP 혼합체인데 각각 칩레벨 프로세싱, 심볼 프로세싱을 담당한다. FPGA와 DSP 혼합체 구조를 넘어서 멀티 DSP를 이용한 병렬 처리기법, 또는 reconfigurable 칩을 개발해서 칩레벨 및 심볼 프로세싱을 한 번에 할 수 있는 개발제품도 출시되었다.

## I. 서론

SDR이란 Software Defined Radio의 약자로서 multi mode, multi band, multi standard의 특징을 가지고 있다. 국방분야에서 먼저 시작된 SDR은 이제 SDR 포럼이란 국제기구를 통해서 표준화가 진행되고 있다. 그 기술의 핵심에는 DSP 칩의 비약적인 발전이 있다. 최근에는 기구산하에 R&D WG가 만들어져 하드웨어 표준화가 논의되는 실정이다.

본 고는 W-CDMA 물리계층을 기존의 FPGA, ASIC에서 재구성 가능한 DSP로 바꿀 때 고려할 점을 다룬다. 즉 업링크(uplink)에만 한정시키고 MAC, RLC, RRC 등의 상위계층, 다운로드 구조를 가능케 하는 2, 3계층, 물리계층 중 채널 코딩은 제외하였다.

물리채널 중에서도 트래픽 채널인 DPCH에 포커스가 있다. 변복조를 담당하는 모뎀을 소프트웨어화(DSP 구현)할 때, reconfigurable한 구조를 구현할 때의 문제점들을 기술함에 목적이 있다. II장은 W-CDMA 규격의 변복조 개요에 대해서, III장은 소프트웨어 모뎀 일레로써 W-CDMA 업링크 송신단을 구현하는 것을 보였으며, IV장은 본문으로서 소프트웨어 모뎀으로 구현시 고려되어야 할 여러 문제를 기술하였으며, V장은 결론을 맺었다.

## II. W-CDMA 변복조기 개요

W-CDMA의 물리계층 프레임 구조는 전체 10ms 시간 내에 38400개의 칩이 들어 있으며(3.84MHz)

2560개의 칩이 존재하는 15개의 슬롯으로 구분된다. 이는 업링크의 경우 심볼이 SF 값만큼 증가된 것이므로 2560/SF(Spreading Factor)의 심볼이 존재한다. SF 값은 다운링크의 경우 4~256 사이의 2<sup>n</sup> 값으로 7개 값을 가진다. 테스트베드의 송신기의 경우 사용자의 전송속도 선택(실제로는 SF 값을 선택하는 효과)에 따라서 트래픽 발생기(예, 음성 및 데이터)의 벌크 비트를 SF 값만큼 반복시킨 후(대역폭 증가가 이루어짐) OVFSF 코드와 스크램블링 코드를 곱해서 프레임의 구성하게 된다.

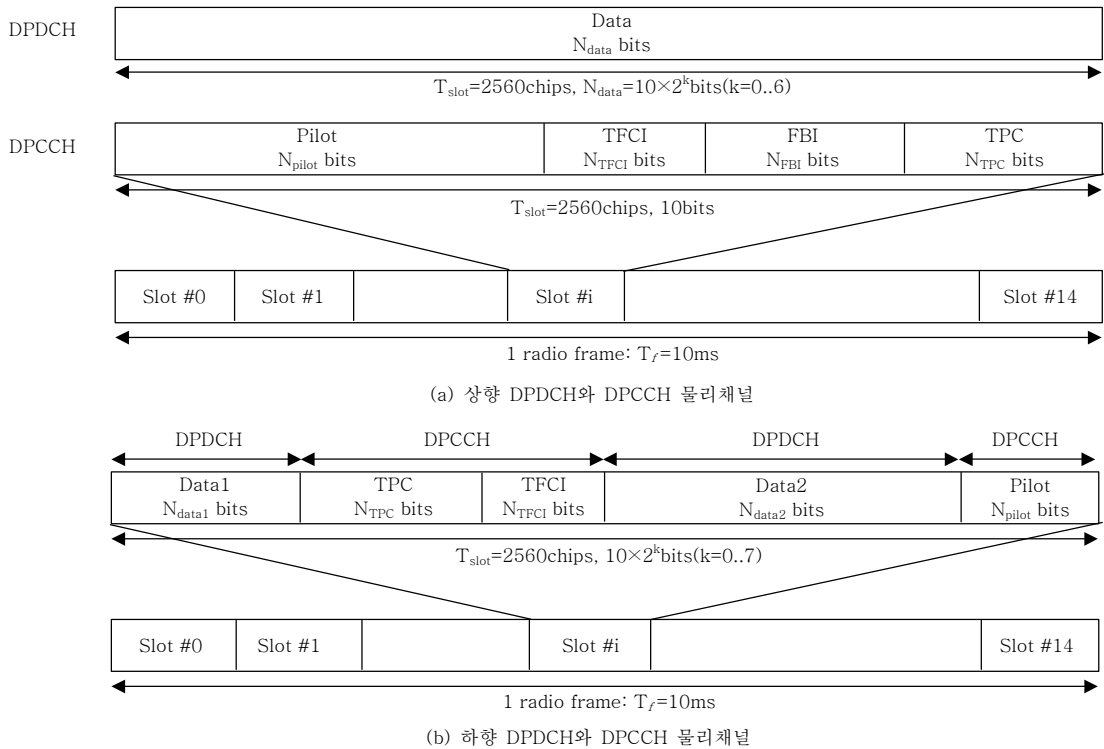
업링크의 경우는 DPDCH와 DPCCH가 분리되어 I, Q 멀티플렉싱 한다. 두 채널의 구분은 서로 다른 OVFSF 코드를 사용함으로 가능하다. 수신기에서 해당 OVFSF 코드를 곱해 적분함으로써 0, 1을 판별하게 되며 스크램블링 코드를 통해서 PN 오프셋으로 표현되어지는 시간 지연을 찾아서 동기를 맞춘다. DPDCH는 SF에 따라 프레임을 구성하며 DPCCH는 SF=256 고정으로서 각종 제어정보를 실어서 보

낸다. 제어정보란 4가지로서 Pilot 비트를 통해 공통 위상 탐색을, TFCI를 통해 전송채널의 포맷 정보를, TPC를 통해 전력제어 정보를, FBI를 통해 다이버시티에 이용되는 피드백 정보전송 정보를 보낸다. 단방향전송에 포커스를 맞춘다면 채널 에스티메이션을 위해서 알려진 Pilot 패턴을 보내도록 한다. 수신기에서는 이 패턴의 변이를 알아내어(복소수의 켈레복소수 성질에 의해) 전파 채널특성을 추정해서 채널보정을 하게 된다.

### 1. 변조기

상하향 전용채널의 프레임 구조는 (그림 1)과 같다.

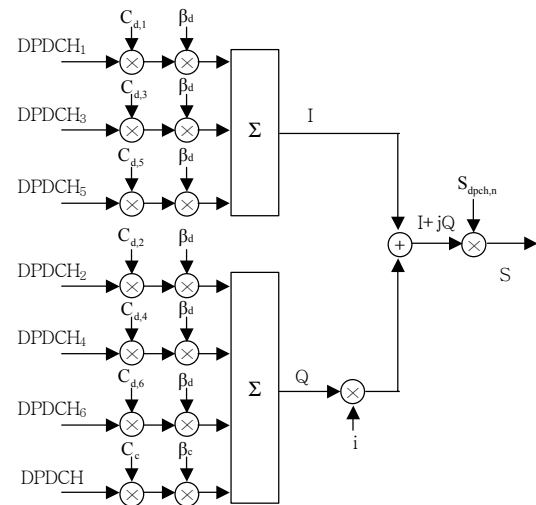
한 프레임에 15개의 슬롯을, 각 슬롯은 2560개의 칩을 가진다. 그래서 한 프레임의 칩 수는 38400개로 균일하다. 각 슬롯에는 심볼이 2560/SF 개만큼 들어 있다. SF는 4~512의 8개 값을 가진다. SF에 따라 한 프레임 내의 심볼 개수가 달라지는데 전



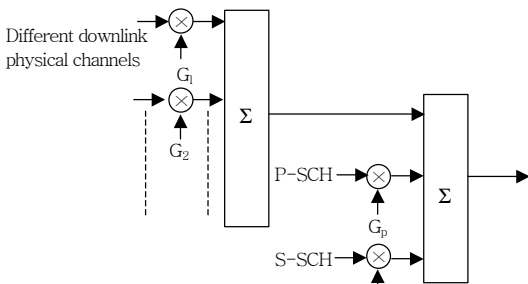
(그림 1) 상하향 DPDCH와 DPCCH 물리채널

송속도에 따라 각종 포맷이 있다. DPDCH의 주요 정보는 DPCCH가 가지고 있는데 Pilot 비트란 공통 위상 탐색에, TFCI란 전송채널의 포맷정보 표현에, FBI라는 다이버시티에 이용되는 피드백 정보 전송에, TPC란 전력제어에 이용된다. DSP 관점에서 단 순한 심볼 비트들의 행렬인데 칩, 슬롯, 프레임 주기의 시계(카운트)가 돌면서 구조를 이룬다.

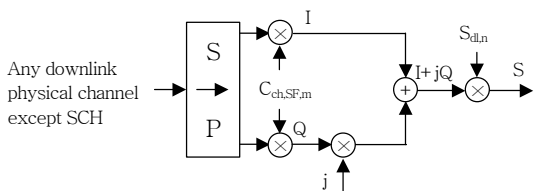
이 비트마다 채널 구별을 위해 OVFSF 코드를 곱



(a) 상향 채널의 채널화와 스크램블링화



(b) 하향 채널의 채널화



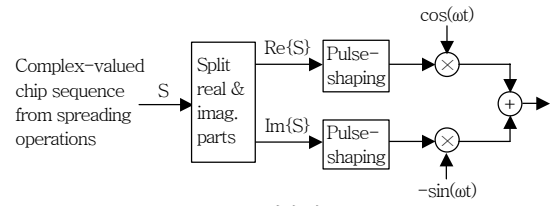
(c) 하향 채널의 스크램블링화

(그림 2) 상하향 채널의 채널화 및 스크램블링

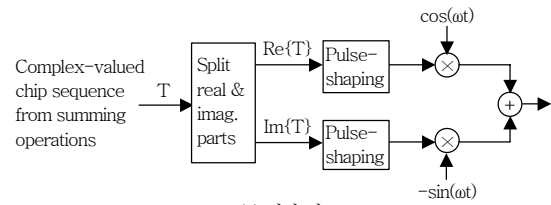
해서 SF 만큼 대역폭을 증가시킨 후, 단말 및 기지국 구분을 위한 스크램블링 코드를 곱한다. 이 코드는 m 시퀀스가 두 개 합해진 Gold 코드로서 38400 칩 주기로서 복소수 값은  $\pm 1, \pm j$  등이다(그림 2) 참조.

업링크의 DPCH인 경우는 DPDCH와 DPCCH가 서로 나누어져 I, Q 멀티플렉싱하며, 다운링크는 타임 멀티플렉싱을 해서 DPDCH와 DPCCH가 서로 합쳐져 있으나 다시 직렬 정렬에서 I, Q로 병렬 정렬함으로 역시 I, Q로 나누어짐을 알 수 있다(그림 2c) 참조.

그 후 각각 캐리어 변조시킨다. 실수 및 허수부로 나누어 주파수 트래킹은 필수적인데 주파수를 알기 전까진 coherent detection이 불가능하므로 각 스크램블링 코드를 신호마다 곱해서 제공값을 구해서 RRC 필터로서 펄스 모양을 만든다(그림 3) 참조.



(a) 상향 변조



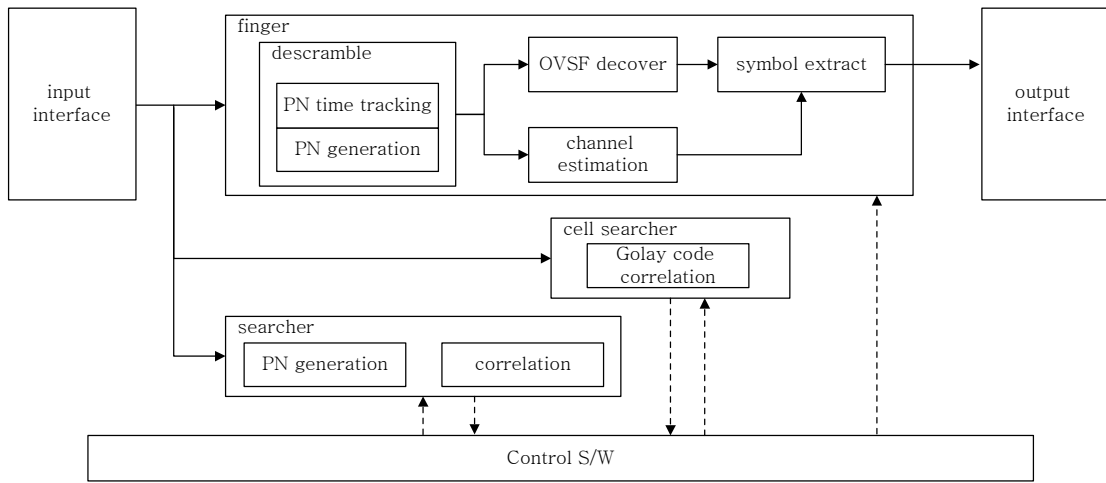
(b) 하향 변조

(그림 3) 상하향 변조

## 2. 복조기

CDMA에 대한 복조기는 잘 알려져 있다. 보통 4 배의 오버샘플링을 하여 서브칩 레벨의 동작을 한다. 각 샘플 입력을 기반으로 동작한다(그림 4) 참조.

- Searcher: SCH 신호를 통해 1단계는 슬롯동기를, 2단계는 프레임 및 코드 그룹을 알고, 3단계에선 8개 중 하나를 선택하는 3단계의 비동기 특



(그림 4) 복조기

유의 탐색을 한다. Golay 코드와 상관을 이용한다. 초기 탐색 이후에도 핸드오버 탐색을 한다.

- MPS(다중경로탐색기): 다중 경로의 전력, 에코 타이밍, 해당 스크램블링 코드를 찾아낸다. 이 정보로서 제어기가 핑거(finger)를 할당한다.
- Estimator: 채널의 왜곡 파라미터를 CPICH 수신을 통해서 알게 된다. SF=256이므로 256칩마다 적분한 결과값인 심볼 하나씩을 추출한다.
- 핑거: 변조기에서 행했던 채널화 및 스크램블링 작업의 역 프로세싱 작업을 진행하는데 채널 Estimator를 통해 알아낸 보정치로서 각 핑거마다 곱해서 심볼을 복조한다.
- 콤파이너: 각 핑거로부터 수신한 신호를 결합하는데 MRC를 많이 사용한다.
- AFC: 정확한 시간 및 최대치를 파악함으로써 PN 타이밍을 트래킹하도록 한다. 이것을 DLL(Delay-Lock tracking Loop)이라고 한다.

### III. W-CDMA 업링크 송신기 구현

DSP 디지털 하드웨어로 어느 범위까지 구현하는냐는 문제인데 초점이 베이스밴드 프로세싱에 맞추어져 있다. 사실은 SDR 구조의 발전을 보면 ADC가 발전하면서 점차 아날로그 영역을 대체하고 있다.

sin과 cos 함수를 발생시켜 입력신호에 곱해주어서 상하향 디지털 변조를 시켜 1차 중간주파수로 바꾸는 부분, 이미지 및 앨리어싱 제거 필터, AGC(자동이득제어장치) 등이 점차 디지털화 되고 있다. 이런 부분을 SDR 포럼에서는 채널화부라고 하며 베이스밴드 프로세싱과 분리하고 있다. 본 고는 베이스밴드부만을 포커싱하고 있으나 디지털 하드웨어(DSP 혹은 FPGA)로 구성할 수 있다.

소프트웨어 모뎀은 하드웨어 모뎀을 모사한 것이므로 특징을 살려야 한다. 가장 전형적인 예는 PN 코드를 생성할 때 일어난다. 하드웨어의 경우는 시프트레지스터를 한 비트씩 이동시키면서 생성한다. 이를 소프트웨어적으로 모사한다면 시프트레지스터를 한 정수 변수로 선언하고서 연산자를 이용해서 한 비트씩 시프트시킬 것이다. 이보다는 소프트웨어적인 특성을 살려 한 비트마다 모두 정수 형태로 바꾸어서 계산량이 훨씬 편리하고 빠르다. 이유는 DSP 프로세싱의 기본이 바로 정수형이기 때문이다. 또한 한 샘플마다 계산하지 않고 내적 연산에 강한 DSP의 특징을 이용해서 블록 프로세싱을 하는 것도 소프트웨어적 모뎀의 유연성을 증가시킨다. 블록 프로세싱에서 중요한 것은 실시간 조건(dead time)에 맞게끔 사이즈를 조정하는 것인데 프레임의 한 비트를 절대적 위치보다는 모듈러 연산을 통해서 무한

루프가 형성되더라도 최대치를 넘어서는 일이 없도록 해야 한다.

본 고는 송신기만을 다루었는데 수신기는 현재 개발중이다. 수신기의 경우는 서처, 멀티서처, 핑거 등의 모듈이 서로 동시적으로 동작해야 하는데 송신기는 시퀀셜한 프로세싱이 가능해서 수신기보다는 훨씬 간단하다. 속도면에서도 송신기는 칩 레벨에서만 다루면 되지만 수신기는 동기를 맞추기 위해 서브칩 레벨까지 다루어야 하므로 계산량도 많은 차이를 보인다. 실시간적 제한이란 장벽이 수신기에겐 있다.

상향링크의 경우 채널을 획득하기 위한 RACH (랜덤 액세스 채널)이 있으며 DPCCH 처럼 전력 제어, 파워오프 정보, 피드백 정보가 담긴 것이 있는데 송수신기가 완성되었을 때 피드백이 되면서 실험해야 될 것이다. 본 고는 트래픽 채널 중 DPDCH의 변조에 관심을 두었으며 DPDCH와 DPCCH의 트래픽을 동시에 발생시키는 구조를 이룬다.

개발환경은 SDR 시스템 랙에 실장할 예정이지만 우선 DSP 모듈상의 테스트를 위해 CCS2.2 환경에서 PC와 연결해서 사용하고 있다. 언어는 C만을 사용하고, DSP는 TI 사의 6416칩을 사용한다. RTDX로서 디버깅을 하고서 랙에 실장할 예정이다.

시퀀셜한 순서는 우선 DPDCH와 DPCCH의 데이터를 모사하는 I, Q 시물레이션 데이터를 발생시킨 후에 각각 스프레딩 시켜 준다. 그 후 미리 메모리에 세팅시켜둔 스크램블링 코드를 곱해서 복소수 송신단 데이터를 발생시킨다. 그 후 실수와 허수로 갈라서 실제로 출력한다. FPGA에서 받아서 인터폴레이션, 필터링하여 DAC로 보내어준다.

일반 C로 시물레이션하는 것과는 거의 비슷하지만 차이를 보인다. TI의 GUI 개발환경인 CCS 환경이 그것을 제공하는 데 실시간성 한계를 주는 것이다. 즉 객체지향 의미를 사용한 송신단 태스크를 만들어서 주기마다 인터럽트를 발생시켜 태스크에 트리거함으로써 시물레이션하는 것은 일반 C에서는 하지 못한다. 이 환경을 통해서 일정시간 안에 프로세싱을 끝내야 하는 작업을 테스트할 수 있다.

## IV. 소프트웨어 모뎀의 고려할 점

이 장에선 소프트웨어 모뎀으로 바꿀 때 송수신단에 해당하는 여러 고려할 사항을 나열해보았다. 시퀀셜한 송신기 구조에는 이런 난제들이 적지만 수신기에서는 다양하고도 시간에 민감하게 사안이 발생한다.

### 1. FPGA와 DSP의 역할 분담

FPGA는 Gate Array를 프로그래밍할 수 있는 하드웨어로서 VHDL 언어를 이용한다. Gate Array란 and, or 등의 조합을 다루는 것이어서 비트단위의 연산이 가능하다. 하드웨어이므로 수많은 신호들을 동시에 매우 빠르게 프로세싱할 수 있는 병렬성을 가진다. 따라서 모뎀 제작에 있어서 비트 오퍼레이션을 해야 하는 OVFSF와 스크램블링 코드 복변조를 수행하는 Searcher, Rake Rx 등과 계산량이 많은 필터를 구성한다. 그러나 리로딩이 가능하다고 하지만 시간이 오래 걸리고 유연하게 재구성(reconfigurable)하기엔 시간이 많이 소요되며, 프로세스가 아닌 하드웨어이므로 좀 더 수준 높고 유연한 재구성 시스템을 프로세싱하기엔 단점이 많다.

이에 비해 DSP는 하드웨어적인 제한성이 없지는 않지만 고급언어(하이레벨 언어) C로 프로그래밍하는 프로세서이므로 유연성이 많아서 SDR 시스템의 핵심을 이루고 있다. 그런데 C의 언어 특징상 비트 단위의 연산은 매우 제한적이어서 비트 단위 연산보다는 심볼 단위의 연산에 적합하다. 계산속도는 클럭률에 좌우되는데 TI 칩중에는 600MHz로 동작되는 것이 있다. 이런 TI 칩 중 6416의 경우는 16비트 short type 연산에 최적화되어 있으므로 C언어에서 한 워드에서 1비트를 뽑아내어 short type으로 만든 후 프로세싱해서 비트연산을 한다. FPGA에 비해 매우 비효율적인 프로세싱이지만 프로세서의 속도가 높아짐에 따라 이런 방식의 계산이 점차 개선되리라고 본다. 이러한 문제를 더욱 근본적으로 해결하는 방법이 나온다면 상당히 속도를 향상시킬 수 있을 것이다. DSP의 주언어인 C는 바이트 단위 연

산을 하는 심볼프로세싱이 적합해서 대부분은 비트 단위연산은 어셈블리어의 도움을 받는 것으로 알고 있다. 더 깊게는 VHDL 언어로서 기술되지만 아직은 VHDL과 C언어의 장점을 통합한 언어가 없고 각각 다른 하드웨어에서 적용되므로 서로간 통신을 위해서는 많은 오버헤드가 불가피하므로 비트연산에 적합하지 않은 DSP C언어지만 고속의 클럭 스피드에 의지해서 바이트 연산으로 나가는 추세이다. 문제는 이런 지정된 짧은 시간(데드 타임)에 일정한 스트림을 처리하는 실시간성보다는 동시성이 문제가 된다. DSP는 구조적으로 병렬처리가 안되므로 기법을 통해서 동시처리의 효과를 주도록 해야만 된다. 이 때문에 RTOS 기법이 필요하게 된다.

## 2. 실시간성과 버퍼 스킴

실시간성 조건이란 여러 이벤트가 발생할 수 있는 상황에서 주어진 주기(deadline) 안에 각 이벤트들이 정확한 결과값을 내어야 한다는 것이다. 다음 이벤트가 들어오기 전에 현재의 이벤트를 모두 처리해야 지연 및 손실이 없는 조건이다. 특히나 계산량이 많은 W-CDMA의 경우, DSP의 프로세싱 구조, 데이터 타입구조, 메모리 사이즈, 인터럽트 방식 등에 대한 이해를 바탕으로 정확한 파라미터값을 찾아내어야 가능한 일이다. 많은 프로세싱을 거치지만 직접적으로 디지털 신호를 발생하는 것이 수신부에선 바로 ADC이다. RF 신호를 샘플링이 가능한 적당한 중간 밴드로 낮추어 디지털 신호로 바꾸는 곳으로서 샘플링 주파수, 비트 수 등이 주요 파라미터이다. 이 샘플단위의 비트 추출이 바로 가장 실시간성을 요하는 hard real time의 조건을 가진다. 따라서 입력 데이터단의 인터페이스 구조, 동기화 동작이 주요 설계 이슈가 된다.

이런 실시간성을 만족하기 위한 여러 스킴 중 가장 대표적인 것이 바로 'block based processing'이다. DSP는 FPGA 처럼 병렬처리가 되지 않고 고속이지는 않지만, 샘플단위의 연산을 벗어나 블록단위의 연산이 가능함으로써 그 단점을 일부 극복할

수 있는 것이다. 한 순간의 블록 전체를 한 데드라인 시간 안에 처리하기 위해 각종 알고리즘이 도입되어 실시간성 제한을 극복할 수 있는 것이 바로 하드웨어 디자인과 다른 점이다. 이때 DSP 구조에 가장 적합한 블록 사이즈, 데이터 구조, 최적화된 알고리즘이 선정되어야 한다. 샘플들이 블록을 이루기 위해서는 버퍼 스킴이 필수적이다. 버퍼가 어느 정도 차기까지 기다렸다가 다 차면 메모리에 덤프시켜 주면서 DataReady를 전역변수, 플래그, 기타 태스크간 동기전달 수단(세마포, 메일박스 등)을 사용하게 된다.

W-CDMA의 경우 프레임 구조가 칩, (심볼), 슬롯, 프레임 등의 3 (혹은 4)단계 구조를 이루는데, 응용 프로그램에 의해 발생하는 무한 비트 스트림인데 대해 각 하부구조마다 루프를 돌면서 동작(OVSF 및 스크램블링화, 전력제어, 다이버시티 수행 등)을 수행한다. 각 심볼, 슬롯, 프레임을 이루기까지는 버퍼를 통해 모아지게 되는데 그 사이즈가 데드라인과 관련해서 알아내어야 할 중요한 파라미터이다. 한 예를 들면 다음과 같다. 복조기는 왜곡된 신호를 변별하기 위해 변조기보다 몇 배의 오버샘플링(Ne)을 한다. 보통 4배의 샘플링을 하며 ADC 성능향상을 위해 4배의 인터플레이션을 하므로 결국 8배 샘플링의 효과를 가지도록 한다. ADC에서 나오는 비트 수(보통 10~14비트) 혹은 Matched 필터를 거친 출력 비트 수가 2바이트를 넘지 않는다고 하고 1 프레임단위로 처리한다면  $Ne \times 3.84\text{MHz} \times 2\text{byte}/\text{SF}$  만큼의 버퍼를 차지한다. SF=16일 경우 3.84Mbyte의 버퍼가 필요하게 된다. 1 슬롯단위로 처리한다면 2.56 Mbyte가 소요된다. 그 블록사이즈가 너무 크다면 심볼단위 혹은 정수배의 심볼단위로 정해지게 될 것이다. I, Q 스트림이 분리되어 있다면 I와 Q 데이터의 구조(각각을 1바이트를 사용할지 1word로 사용할지 또는 서로 엮갈리거나 연속적으로 배열함)에 따라 성능이 차이가 있을 것이다. 그것을 약속에 의해 정하게 된다. 블록사이즈가 너무 작다면 block based processing의 장점이 살아나지 않을 것이며, 너무 길다면 데드라인의 조건을 넘어서게 되어 실시간성 제한을 어기게 된다. 슬롯단위의 프로세싱이

가장 적합하리라고 본다.

### 3. 동시성, 인터럽트 방식, 스케줄링 기법

다수의 프로그램 모듈(태스크)을 동시에 동작시킬 경우, DSP는 FPGA 처럼 데이터를 동시에 병렬적으로 처리하지 못하는 단점이 있다. 물론 아예 멀티 DSP를 이용하여 병렬처리형 구조를 만들 수는 있지만 DSP간의 통신을 위한 오버헤드가 적지 않을 것이다. 이 분야는 본 고의 초점에서 벗어나므로 생략하고 기본기능에 대한 단일 DSP 구조로만 논하고자 한다.

각 태스크의 동작이 시간순서에 따라서 시리얼 동작을 한다면 굳이 가로채기(preemption)가 되는 RTOS가 필요하지는 않다. 하나의 단점이라면 C언어 코딩에 있어서 무한 루프를 나타내는 while(1)문으로 되어 있다면 일이 없을 경우에도 계속 공회전하는 비효율성에 있다. 여기에 대한 단점을 극복하기 위한 것이 인터럽트 방식의 프로그램이다. 가장 기본적인 방법은 어셈블리어로 콘텍스트 스위치 처리 및 벡터 테이블 동작을 작성한다. 좀더 진보된 방식은 스케줄링 도구에 의한 hwi 방식으로 저레벨의 일은 OS(hardware dispatcher)가 담당하도록 한다. 이 방법은 자원을 필요할 때만 사용하는 효율성이 있으나 동시적으로 이벤트가 발생할 경우 한 개만 처리된다. 또한 그런 면까지 고려해서 프로그램을 한다면 스테이트가 워낙 복잡해져서 프로그래밍하기도 힘들 뿐더러 일처리에 대한 결과를 예상하기 힘들게 된다(Nondeterministic). 이런 하드웨어만으로는 스케줄링은 한계가 있으므로 foreground-background란 개념이 도입되었다. TI의 경우 hwi를 모사한 swi가 도입되어 데드라인이 비교적 긴(soft real time) 태스크가 시간차를 두고 동작되고 스케줄링하도록 되어 있다. 이 방법도 여전히 다수 태스크를 유연히 동작시키고 커뮤니케이션하는 것에 제약이 있으므로 태스크가 등장하였다.

다수의 태스크를 동작시키는 가장 단순한 방법은 멀티채널 응용에서 많이 사용되는 시간분할형이다.

각 태스크별로 시간순서에 따라(= time task의 인터럽트 주기에 따라) 라운드 로빙 방식으로 주어진 시간을 쪼개어서 할당하는 방법이다. 이보다 복잡한 스케줄링을 하기 위해서는 각 태스크별로 우선순위가 주어지고 조정함으로써 행해진다. 또 태스크간 동기를 맞추어주기 위해 세마포를 사용하게 된다. 이런 스케줄링 기법을 통해 시간에 민감한 작업을 짧은 데드라인의 한계 내에서 동시성 문제를 일부 해결한다.

이러한 문제를 W-CDMA의 수신부에 적용한다면 다음과 같다. (1) 오버샘플링 때문에 만들어진 서브칩, 채널화 코드로 생긴 칩, 상위계층의 정보를 추출하는 심볼, 제어 채널의 단위인 슬롯, 모든 카운트가 원점으로 돌아가는 (EPOC) 프레임은 각각 다루어야 한다. 문제는 각각의 주기적 인터럽트가 겹칠 경우 (예, 칩, 슬롯, 프레임의 시작점이 겹칠 때) 동시성 문제가 발생하는 것이다. 이때 비교적 긴 데드라인을 가진 슬롯 및 프레임 프로세싱을 약간 지연시켜서 처리해줄 스케줄러 방식이 적용된다. 물론 critical region을 피해서 슬롯 동작을 하도록 하면 동시성을 일부 피할 수는 있다. (2) 서처의 경우 초기 동기를 맞추기 위해 correlation을 취하여 (각 PN 코드를 곱해서) 최대값이 되는 PN 오프셋을 찾는다. 멀티 셀 환경에서는 늘 동작하다가 셀이 바뀔 경우 인터럽트를 통해 정보를 업데이트하므로 계산량이 무척 많다. 근거리와 이동이 적은 테스트베드 같은 단일 셀 환경이라면 초기 오프셋만 구하고 쉬는 것이므로 한 개의 서브루틴으로 처리하거나 우선순위가 낮은 idle task 같은 백그라운드 작업으로 배정함이 옳다. (3) 멀티패스 서처의 경우는 다수 경로의 PN 오프셋을 찾아서 각 핑거를 할당하기도 하고, 스크램블링 코드를 리셋시키기도 하며 slewTiming을 주기도 한다. 이때도 다수의 모듈이 데드라인 안에 동작해야 하므로 동시성이 요구된다.

### 4. 채널 코드 부분(OVSF+스크램블링 코드)

이때 스크램블링 코드는 m 시퀀스가 두 개 합해진 Gold 코드로서 값이  $\pm 1, \pm j$  값을 가지며 전 주기

중에 38400 (한 프레임의 칩 수) 만큼만 사용한다. DPDCH와 DPCCH로 들어오는 각 I, Q 비트스트림에 OVFSF 코드를 곱하고서 각각을 실수 및 허수 스트림을 보고 복소수를 만든 다음 복소수 스크램블링 코드를 곱한다. 이것을 하드웨어적으로 만든다면 복소수 계산을 각각 4개의 실수계산으로 나누어야 하지만 DSP의 C언어로서 구조체를 사용해서 복소수 연산을 할 수 있다. 두 채널이 들어오지만 복소수를 사용해서 한 채널을 이용하는 것처럼 사용할 수 있는 것이다. DSP에서 16비트 연산이 최적화되어 있으므로 1word에 16비트짜리 I, Q 데이터를 넣어서 한 채널로 전송시키고 프로세싱해서 1word에 넣어서 출력시키면 된다. 그 다음은 FPGA 하드웨어를 통해서 I, Q 분리하고 각각 FIR, RRC 등의 필터처리를 해서 RF 보드로 I, Q 스트림을 단자를 통해서 전송하면 된다. RF 보드를 통해서 캐리어 변조되어 안테나를 통해 방사된다.

그런데 이 스크램블링 코드를 만드는 것은 시프트레지스터를 이용한 비트연산이므로 심볼계산에 편한 DSP 연산으로는 적합하지가 않다. 너무 많은 클럭사이클이 소요되므로 발생시켜서 사용한다면 실시간성을 어기게 된다. 따라서 어셈블리어로 짜든지, 블록단위로 PN 코드를 발생시켜야 한다. 여기서는 가장 간단한 방법으로 LUT(Look Up Table)을 이용하는 방법을 취한다. OVFSF의 경우도 SF={4, 8, 16, 32, 64, 128, 256}의 7개가 있으므로 비트단위로 LUT를 저장하고 바이트 단위로 끄집어 넣으로써 사용할 수 있다. 그러나 그것도 메모리를 매우 많이 차지한다고 판단이 되면 하드마더 코드를 이용해서 단순 패턴을 클럭에 따라서 계속 발생시켜 데이터와 곱해주는 방식을 사용함이 더 좋을 수 있다. 업링크 단말기의 경우 OVFSF 코드를 2개 사용하므로 ALL 1, 번갈아서 0, 1 패턴을 사용하면 된다.

## V. 결론

W-CDMA 업링크의 송신단 구현을 통해서 소프트웨어 모뎀 구현시 고려되어야 할 여러 난제들을 정리해보았다. 버퍼 스킴을 이용한 블록 단위의 프로세싱을 통해서 하드웨어 프로그래밍과 다른 소프트웨어 모뎀의 특징을 잘 살려야하며, 실시간성에 대한 제한을 맞추기 위해 스케줄링을 이용하는 RTOS를 사용해야 하는 필요성이 있으며, 최대한 고속으로 계산하기 위해 수표 발생은 테이블화함이 좋다는 것을 기술하였다. 그러나 여전히 SDR의 전통적인 구조인 FPGA와 DSP의 혼합체는 서로 통신해야 하는 장애물이 남아 있다. VHDL과 C코드로 각각 프로그래밍하는 디지털 하드웨어는 서로 이질적이기 때문이다. 전자를 통해 칩 프로세싱, 후자를 통해 심볼 프로세싱을 한다고는 하지만 정보를 서로 주고받는 데서 많은 시간낭비가 예상된다.

그래서 아예 칩 자체를 reconfigurable 칩을 이용해서 칩 및 심볼 레벨 프로세싱을 없애고 일원화해서 접근하는 업체(Chameleon System, Pico Chip)도 있다. 또 하나의 흐름은 멀티 DSP를 사용한 병렬 처리 기법(예, SandBridge사)를 사용한다. 이렇게 서브 칩, 칩, 심볼 레벨의 프로세싱의 일원화가 소프트웨어 모뎀의 핵심기술이라고 본다.

## 참고 문헌

- [1] 3GPP규격, TS 25.211(v3.10.0), TS 25.213(v 3.7.0)
- [2] Thomas Keller and UbiNetic, "Implementation of Chip Rate Processing Algorithms for Third Generation W-CDMA Mobile Terminals," *VTC'01*, pp. 2745-2749.
- [3] Keld Lange, Alcatel Mobile Networks Division, Germany, "A Software Solution for Chip Rate Processing in CDMA Wireless Infrastructure," *IEEE Communication Magazine*, 2002. 11. 8.
- [4] SPRA680 - July 2000, "Implementation of a WCDMA Rake Receiver on a TMS320C62x DSP Device."
- [5] DSP/BIOS II Workshop Student Guide