

논문 2006-43TC-3-11

# UWB MAC의 Time Slot 동기를 통한 시스템 성능 개선

## ( System Performance Improvement of IEEE 802.15.3a By Using Time Slot Synchronization In MAC Layer )

오 대 건\*, 정 정 화\*\*

( Dae-Gun Oh and Jong-wha Chong )

### 요 약

본 논문은 UWB (Ultra Wide Band) 시스템의 성능 개선을 위해서 Superframe 주기를 이용한 MAC(Medium Access Control) 계층 time slot 동기 알고리즘을 제안한다. Multi-band OFDM Alliance (MBOA) 에서 제안한 UWB시스템에서는 Time Slot의 동기를 위해서 Medium Access Slot (MAS) 와 MAS사이의 guard time에 단말기들 간의 MAC 계층 주파수 오프셋으로 야기될 수 있는 시간 오차의 최대값인 MaxDrift를 더해지게 된다. MaxDrift를 더한 만큼 MAS에서 데이터를 전송할 수 있는 시간이 줄어들게 되므로 각각의 MAS에 MaxDrift를 더해주는 방식은 전체 시스템 성능의 저하를 가져오게 된다.

본 논문에서는 시스템의 성능을 높이고자 time slot동기를 guard time을 증가시키는 방식이 아닌, Superframe주기로 전송되는 연속된 Beacon Frame을 수신하여 주파수 오프셋 값을 estimation하여 보정해주는 방법을 제안한다. Piconet을 초기화시킨 Device는 내부 clock을 이용해서 Superframe주기로 Beacon을 전송을 하므로, Piconet에 접속하려는 단말기들은 연속된 Beacon을 수신하여 Piconet을 생성한 단말기의 MAC계층과 수신한 단말기와의 MAC계층 주파수 오프셋을 구할 수 있다. 각각의 수신 단말기에서 측정된 상대적 주파수 오프셋 값을 내부적으로 estimation한 각각의 MAS의 position에 가감시켜 Piconet을 생성한 단말기에서 estimation한 MAS position에 동기를 맞출 수 있다. 제안된 알고리즘을 통해서 단말기들 간의 최대 주파수 오프셋 값과 관계없이 MaxDrift로 인해서 낭비되는 시간을 각 MAS당 1clock 이내로 줄일 수 있다.

제안된 알고리즘을 하드웨어로 합성한 결과 390개의 Logic Cell이 소모되었으며, 시뮬레이션 결과 최대주파수 오프셋이 20ppm, 40ppm, 80ppm일 때 MAS당 오차범위가 main clock의 1clock이내였으며 기존의 방법에 비해서 각각 1%, 2%, 4%의 throughput이 향상되었다.

### Abstract

In this paper, we propose the algorithm to reduce guard time of UWB MAC time slot for throughput gain. In the proposed draft by multiband ofdm alliance (MBOA), Guard time of each medium access slot (MAS) is composed of shortest inter-frame space (SIFS) and MaxDrift which is the time caused by maximum frequency offset among devices.

In this paper, to reduce guard time means that we nearly eliminate MaxDrift term from guard time.

Each device of a piconet computes relative frequency offset from the device initiating piconet using periodically consecutive transferred beacon frames. Each device add or subtract the calculated relative frequency offset to the estimated each MAS starting point in order to synchronize with calculated MAS starting point of the device initiating piconet. According to verification of simulations, if the frequency offset estimator is implemented with 8 decimal bit, the ratio of the wasted time to Superframe is always less than 0.0001.

**Keywords :** UWB (Ultra Wide Band), MaxDrift = Maximum Drift , MAS = Medium Access Slot , Beacon

### I. 서 론

최근 유비쿼터스 센서 네트워크에 대한 관심이 증가

하면서 AD-HOC 시스템에 관한 연구가 활발히 진행되고 있다. 블루투스과 Zigbee, UWB와 같은 통신 시스템이 현재 주목을 받고 있으며, 이 중 UWB 통신시스템은 광대역 초고속 통신 시스템으로 블루투스나 Zigbee와 같은 타 시스템에 비해 throughput이 뛰어나 무선 USB, 홈 네트워킹 등의 응용분야에 솔루션으로 각광받

\* 학생회원, \*\* 정회원, 한양대학교 정보통신학과  
(The College of Information & Communications ,  
Hanyang University )

접수일자: 2005년7월21일, 수정완료일: 2006년3월16일

고 있다.

현재 XstreamSpectrum 사에서 제안한 DS(Direct-Sequence)를 이용한 CDMA 방식과 Multiband사에서 제안한 Multiband-OFDM 방식이 표준안으로 제안되고 있지만 아직까지도 표준안이 확정되지 않았다.

MBOA에서 제안한 Physical Layer의 OFDM 방식은 송. 수신기간의 주파수 오프셋이 BER에 미치는 영향이 크기 때문에 주파수 오프셋을 estimation하고 보정하는 알고리즘과 하드웨어 구조에 대한 연구가 활발히 되고 있지만, MAC 계층에서 발생하는 주파수 오프셋에 대한 연구는 상대적으로 미비하다. AD-HOC 네트워크에서 MAC 계층 주파수 오프셋은 modem을 통한 데이터의 전송에는 직접적으로 영향을 미치지 않지만, 시분할 다중접속 방식으로 데이터를 전송할 때 시스템의 throughput을 떨어뜨리게 된다. MBOA에서는 제안한 시스템에서는 MAC 계층의 발생 가능한 최대 주파수 오프셋 차이를 고려하여 각각의 Time Slot인 MAS의 Guard Time에 MaxDrift라는 마진을 두게 된다. 이 MaxDrift라는 시간 값으로 인해서 데이터의 전송시간이 줄어들게 됨으로 전체 시스템의 성능을 떨어뜨리게 된다. 각각의 MAS의 MaxDrift값은 식 1에서 정의된 것과 같이 Superframe이 시작된 후 시간이 지날수록 점점 커지는 특성을 가진다. MaxDrift로 인해서 전체 Superframe중 얼마만큼의 시간이 낭비되는지 알아보기 위해서 모든 MAS의 MaxDrift의 합을 Superframe 시간으로 나눈 MaxDriftRatio값을 구하면, MaxDriftRatio값은 MAS의 수와 주파수 오프셋에 의해서만 영향을 받게 되는 것을 알 수 있다. 즉, MAS의 수가 많을수록, 주파수 오프셋이 클수록 낭비되는 시간이 많아진다. 즉, 기존의 MBOA에서 제안한 MAC 계층의 Time Slot동기를 MaxDrift를 더해서 해결하는 방식은 throughput의 낭비를 가져올 뿐만 아니라, MAS의 수에 따라서 낭비되는 시간이 영향을 받게 된다. 이러한 문제점을 극복하고자 제안된 알고리즘은 Piconet을 초기화시킨 단말기에서 전송하는 연속된 Beacon을 모든 단말기에서 수신하여, 상대적 주파수 오프셋을 구하여 Piconet을 초기화시킨 단말기의 MAS의 시작 시간에 동기를 맞추는 데 있다.

제안된 알고리즘을 통해서 주파수 오프셋과 MAS의 수에 관계없이 기기들 간의 Time Slot의 동기화를 1clock 이내로 가능하게 하므로 MaxDriftRatio값을 매우 작은 일정한 값으로 수렴시킬 수 있게 된다.

본 논문의 II장에서는 MBOA에서 제안한 UWB 시

스템에 대해서 살펴보도록 할 것이며 III장에서는 주파수 오프셋으로 인해서 발생하는 오버헤드인 MaxDrift가<sup>[2]</sup> UWB 시스템의 throughput에 얼마만큼 영향을 주는지 살펴보고, MaxDrift를 제거함으로써 얻을 수 있는 데이터 처리량의 이득을 살펴볼 것이다. IV장에서는 Time Slot동기 알고리즘을 알아볼 것이며, V장에서는 알고리즘을 하드웨어로 구현시킨 것의 블록 다이어그램과 Behavioral simulation의 결과를 확인할 것이다.

## II. 본 론

### 1. MBOA에서 제안한 UWB 시스템의

#### Superframe과 Guard Time

UWB 시스템은 근거리에서 있는 단말기들을 이용하여 Ad-hoc network를 형성한다. 이렇게 형성된 network(piconet)에 속한 단말기들은 시분할다중접속 방식으로 데이터를 주고받는다.

그림 1에서, Superframe의 길이는 65536us이며, 256개의 Media Access Slot으로 구성이 되며 MAS(Media Access Slot)는 256us이다. 그림 2에서 보듯이 Superframe은 Beaconing Slot과 PCA(Prioritized Control Access)와 DRP(Distributed Reservation Protocol)로 구성되어 있다. Beaconing Slot에서 Piconet상의 단말기들은 자신의 정보(Mac Address, 단말기 ID, Beaconing Slot Position 등)를 Beacon Frame을 통해서 전송한다. 또한 Beacon Frame을 통해서 각각의

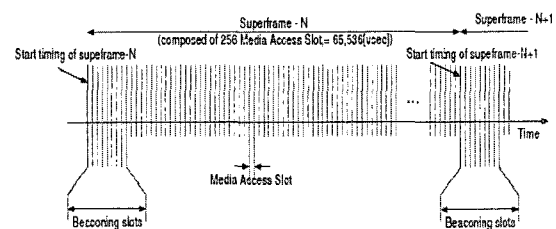


그림 1. MBOA에서 제안한 Superframe  
Fig. 1. The superframe structure proposed by MBOA.

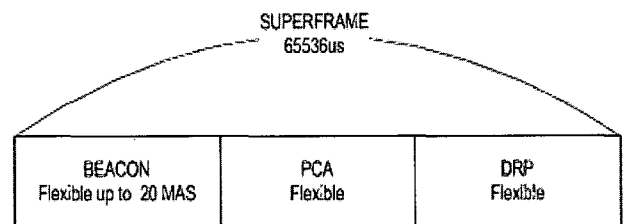


그림 2. Superframe의 구성요소  
Fig. 2. Components of superframe.

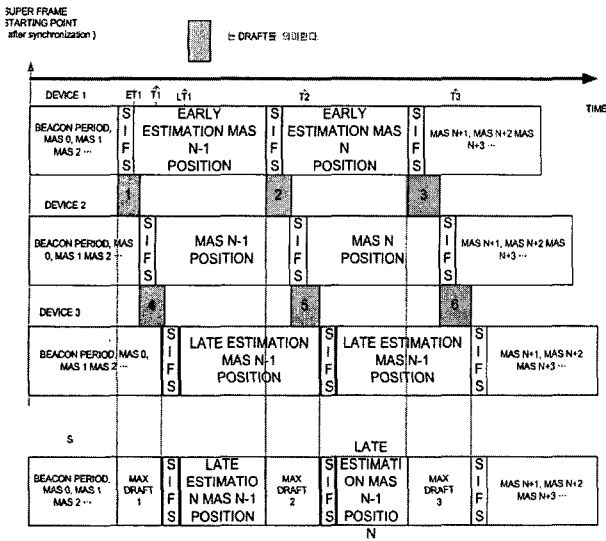


그림 3. 주파수 오프셋이 다른 단말기들의 MAS 위치 추정

Fig. 3. The conventional MAS starting position estimation methods.

DRP를 어떤 단말기가 사용할 것인지 결정을 한다. 그림 1에서는 보이지는 않지만 MAS와 MAS사이에는 Guard Time이 존재하며 Draft에서는 식1과 같이 Guard Time을 정의하고 있다.

$$\text{Guard Time} = \text{SIFS} + \text{MaxDrift} \tag{1}$$

$$\text{MaxDrift} = \text{Max Frequency Offset} * \text{interval}$$

MaxDrift값이 주파수 오프셋으로 인해서 발생하는 시간이며 주파수 오프셋값이 크고 Superframe의 시작 지점에서 멀어질수록 MaxDrift값은 커지게 된다.

MBOA에서 제안한 Draft에서는 Superframe의 시작 부분(BPST = Beacon Period Start Time)에서 단말기들 간의 synchronization을 한다.<sup>[2]</sup>

그림 3에서, Superframe의 시작점에서 단말기들이 동기 이후, 각각의 단말기가 추정한 MAS의 위치가 서로 다른 것을 알 수 있다. 이러한 추정의 차이는 단말기 간의 주파수가 정확히 일치하지 않기 (주파수 오프셋) 때문에 발생하는 것이다. 이러한 추정의 차이로 발생하는 것이 MaxDrift이다.

MaxDrift없이 SIFS만으로 Guard Time을 구성하게 되면 어떠한 일이 발생하는지 살펴보자. 먼저 Beaconsing Period동안 다음과 같은 시나리오를 가정해보자.

- (1) Device1이 N-1번째의 MAS를 Beacon Frame을 통해서 예약함.

(이 때, N은 시간T1 일 때의 Superframe안의 MAS의 위치 번호)

- (2) Device2와, Device3은 Device1의 Beacon Frame을 받아서 분석한 다음, N-1번째 MAS는 Device1이 쓰는 것으로 인식함
- (3) Device2가 N번째 MAS를 Beacon Frame을 통해서 예약함.
- (4) Device1, Device3은 Device2의 Beacon Frame을 받아서 분석한 다음, N번째 MAS는 Device2가 쓰는 것으로 인식함
- (5) Device3가 N+1번째 MAS를 Beacon Frame을 통해서 예약함.
- (6) Device1, Device2는 Device3의 Beacon Frame을 받아서 분석한 다음 N+1번째 MAS는 Device3에서 쓰는 것으로 인식함.

a) 위와 같은 시나리오를 가정하고 그림 3을 보게 되면 Device1은 시간 T1이전에 N-1번째 MAS가 시작되었다고 판단하여 T1 이전에 Frame전송을 시작한다. 하지만 Device2는 시간 T1에서 N-1번째 MAS가 시작되었다고 판단하기 때문에 Device1에서 전송한 Frame의 시작부분을 놓치게 되고, 이에 따라 Device2는 Frame을 수신하지 못한다. Device3도 위와 같은 이유로 Device1의 Frame을 받을 수 없게 된다.

b) N번째 MAS는 Device2가 예약을 했었다. Device2가 시간 T2때 Frame을 전송 한다. 이 때, Device1은 T2이전부터 대기상태에 있기 때문에 Device2가 전송하는 Frame을 수신하게 된다. 하지만, Device2가 전송하는 Frame이 SIFS를 제외한 MAS를 모두 사용하면 Frame의 끝 부분이 전송되기 이전에 Device1은 N번째 MAS가 끝났다고 가정하기 때문에 Frame의 일부분이 전송 되지 않게 된다. Device2와 Device3의 관계는 위에서 언급한 바와 같다.

a), b)와 같은 사건이 발생할 수 있으므로 그림3의 Device4와 같이 256개의 모든 MAS의 Guard Time에 MaxDrift값을 더해줘야 한다. 하지만 Guard Time에 포함되어 있는 MaxDrift값은 MAS의 position이 증가됨에 따라서 점점 커지게 된다. 시간의 효율성 측면에서 보았을 때 이러한 방법은 좋지 않다.

그림 3에서는 SIFS가 크게 표현되었지만, 실제로는 매우 작은 값이다.

2. MaxDrift로 인한 데이터 처리량의 감소와 Superframe의 제약

단말기들이 최대 x ppm의 주파수 오프셋의 차이를 보일 때, 식 1을 이용해서 각각의 MaxDrift를 모두 합산하여 전체 MaxDrift의 시간을 구해보면 식 2와 같다.

Main Clock : KMHz , MAS Duration : Nus  
 Superframe을 이루는 MAS의 개수 : H MAS  
 주파수 오프셋 : x ppm

$$\begin{aligned} \text{TotalMaxDrift} &= \text{MaxDrift}_1 + \dots + \text{MaxDrift}_H \\ \text{MaxDrift}_i &= (N \cdot I)_{us} * \text{kHz} + 2x(\text{ppm}) \\ \text{TotalMaxDrift} &= N * K * 2x(\text{ppm}) * \sum_{i=1}^H I \\ &= N * K * 2x * 10^{-6} * \frac{H(H+1)}{2} \end{aligned} \quad (2)$$

$$\begin{aligned} \text{SuperframeClockCount} &= (H * N)_{us} * \text{kHz} \\ &= H * N * K \\ \text{MaxDriftRatio} &= \frac{\text{TotalMaxDrift}}{\text{SuperframeClockCount}} \\ &= x * 10^{-6} * (H + 1) \end{aligned} \quad (3)$$

식 3의 MaxDriftRatio는 Superframe에서 MaxDrift로 손실되는 비율을 의미한다.

식 2, 3로부터 알 수 있는 것은 각각의 MAS의 Duration과 단말기의 Main clock은 각각의 MaxDrift에 영향을 주지만, MaxDrift의 합이(TotalMaxDrift) 차지하는 비율인 MaxDriftRatio는 오직 주파수 오프셋과 H (Superframe을 구성하는 MAS의 수)에 비례한다는 것이다.

Superframe의 주기 65536us를 256개의 MAS로 분할하며, 주파수 오프셋을 20~200ppm 정도로 가정한다면, 식 3에 의해 MaxDriftRatio는 약 0.5%~5% 정도의 값을 갖게 된다. 즉 Superframe에서 0.5%~5%는 낭비된다는 것이다.

MBOA에서 제안한 식 1에 의하여 유도된 식 2, 3은 이론적으로 아무런 문제가 없지만, MaxDrift가 이론적인 값을 가질 수 있도록 실제 MAC을 구현하는 것은 하드웨어 복잡도가 매우 커지기 때문에 현실적으로 불가능하다. 왜냐하면 Superframe이 시작되고, Beacon을 보낼 시간이 되면, 단말기들이 현재까지 버퍼에 있는 데이터 또는 앞으로 전송해야 될(Superframe 기간 내에) 데이터를 추정해서 필요한 만큼의 DRP(MAS)를 예약해야 한다. 하지만 각각의 DRP (MAS)의

MaxDrift가 다르기 때문에 DRP의 위치에 따라 보낼 수 있는 데이터의 양도 다르게 된다. DRP마다 서로 약간씩 다른 MaxDrift를 감안하여 전송할 데이터가 필요로 하는 만큼 DRP를 예약하는 것은 너무 복잡하고 시간이 오래 걸릴 수 있다. 그래서 실제 구현을 할 때에는 가장 worst한 MaxDrift를 가정하고 DRP를 예약해야 한다. Implementation 관점에서 식 2, 3을 다시 계산하면 식 4, 5와 같다.

$$\begin{aligned} \text{MaxDriftI} &= (N * I)_{us} * \text{kHz} + 2x(\text{ppm}) \\ \text{WorstMaxDrift} &= \text{MaxDriftH} \\ &= (N * H)_{us} * \text{kHz} + 2x(\text{ppm}) \\ \text{TotalMaxDrift} &= (N * H)_{us} * \text{kHz} * 2x(\text{ppm}) * H \\ &= N * K * 2x * 10^{-6} * H^2 \end{aligned} \quad (4)$$

$$\begin{aligned} \text{SuperframeClockCount} &= (H * N)_{us} * \text{kHz} \\ &= H * N * K \\ \text{MaxDriftRatio} &= \frac{\text{TotalMaxDrift}}{\text{SuperframeClockCount}} \\ &= 2x * H * 10^{-6} \end{aligned} \quad (5)$$

앞에서 언급하였듯이 MaxDriftRatio는 X, H에만 영향을 받는다. X는 고정된 Factor라고 가정하면 MaxDriftRatio는 H에 의해서만 결정된다.

이런 이유로, MaxDrift를 Guard Time에 포함시키는 방법은 Superframe을 몇 개의 MAS로 나눌 것인가에 대한 제약을 받게 된다. 왜냐하면 식 5에서 알 수 있듯이 고정된 Superframe을 수십, 수백 개의 MAS로 분할한 만큼 MaxDriftRatio의 값이 증가하기 때문이다.

그림 4를 보게 되면, Piconet1의 MAS의 Period는 PICONET2의 2배이다. 즉 Piconet1이 256개의 MAS로

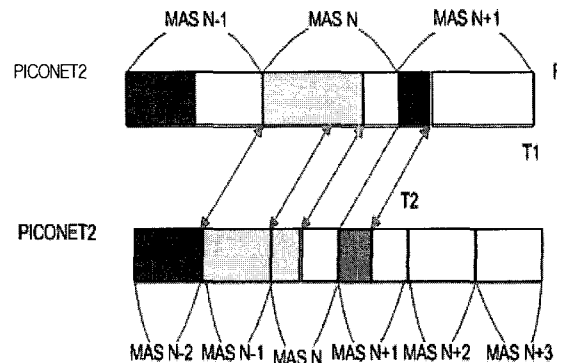


그림 4. MAS Period에 따른 데이터 처리량의 비교 구성  
 Fig. 4. The Comparison of throughput with MAS period.

되어있다면 PICONET2는 512개의 MAS로 구성된 것이다. Piconet1에서는 3개의 단말기가 3개의 MAS를 써서 T1이라는 시간 동안에 데이터를 전송했다. 동일한 데이터를 Piconet2에서 전송하게 되면 4개의 MAS동안에 데이터를 전송하게 되지만, Piconet1의 2개의 MAS에 해당하는 시간동안에 데이터를 모두 전송하게 된다. 그림 4에서 MAS의 주기가 짧은 Piconet2가 Piconet1보다 사용률이 높은 것을 알 수 있다.<sup>[4]</sup>

MAS Period를 짧게 하면 고정된 Superframe에서 MAS의 개수가 증가하게 되며, 이 때 위에서 언급한대로 데이터 처리량을 높일 수 있게 된다. 하지만 식 5에 의해서 MaxDriftRatio가 증가하기 때문에 MAS Period를 짧게 하는 것이 더 안 좋을 수도 있다. 본 논문에서 제안할 MaxDrift를 제거하는 알고리즘을 사용하면, Superframe를 구성하는 MAS의 Period를 짧게 해도 MaxDrift로 인한 시간의 낭비는 없다. 즉, 그림 4와 같은 MaxDrift가 고려되지 않은 상황에서, MAS Period가 짧을 때 발생하는 이득을 얻을 수 있다.

3. MaxDrift를 줄이는 알고리즘

가) Fixed Total Delay

여기서는 Transmitter의 MAC에서 Frame을 보내고 Receiver의 MAC에서 전송된 Frame을 Detection하는데 소요되는 시간(Delay)는 고정되어 있다는 것을 보이고자 한다.

그림 5는 MAC 계층과 PHY 계층의 Interface에 어떠한 Signal들로 구성되어있는지 보여준다.

그림 5의 Control Interface 부분에서 TX\_EN신호가 Enable이 되며 PHY에서는 Preamble신호를 전송을 시작하게 되며 RX\_EN신호가 Enable되었을 시 PHY에서는 Receive State를 유지하면서 안테나를 통해서 들어오는 신호들을 Detection한 다음, 수신한 Frame을 MAC-PHY interface를 통해 MAC에 전달한다.

그림 6은 Transmitter의 MAC에서 PHY로 데이터를 전송하며, Receiver의 PHY가 수신한 데이터를 Receiver의 MAC으로 전달할 때의 Timing을 보여준다. Transmitter 부분을 보게 되면, MAC에서 TX\_EN신호를 1로 인가시킨 후 TxDelay시간이 지나면 안테나를 통해서 Long Preamble신호가 전송된다. Long Preamble 신호의 길이는 고정되어 있으며, 이 때 송신된 Long Preamble 신호는 Air Interface를 통해서 Receiver로 전달이 됨으로서, Propagation Delay를 가진다. Receiver는 Transmitter에서 전송한 Long Preamble

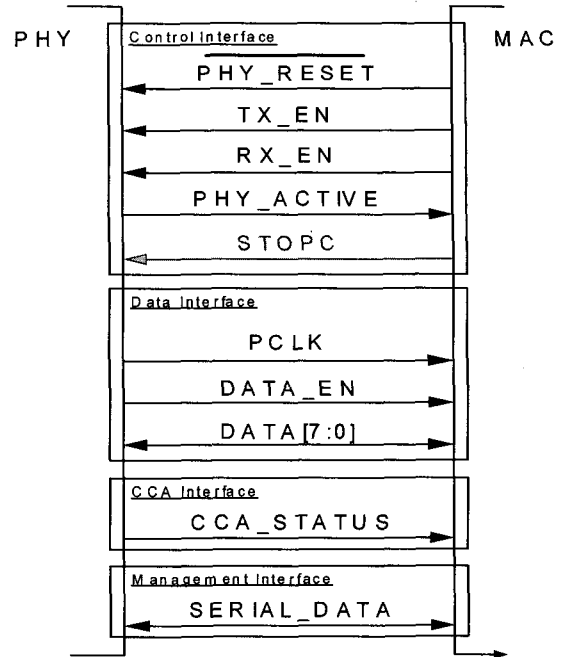


그림 5. MAC/PHY 계층 Interface 구성 Signal<sup>[3]</sup>  
Fig. 5. MAC/PHY Layer interface signals.

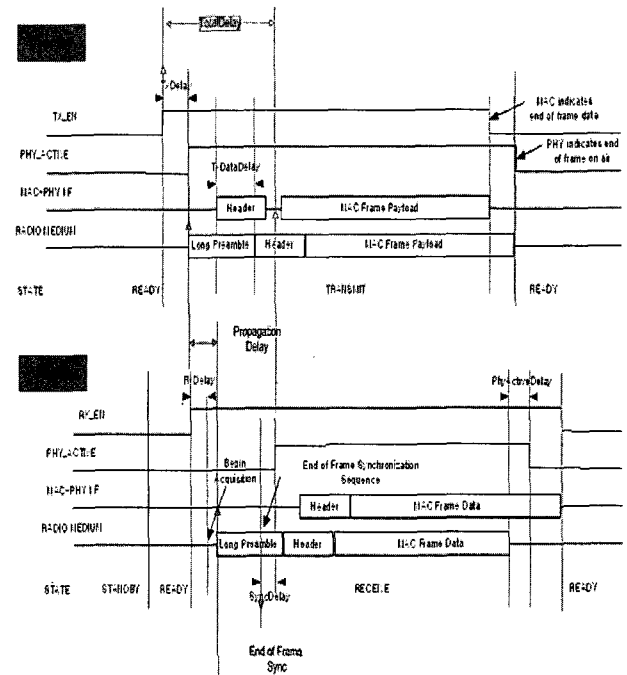


그림 6. Transmitter의 MAC에서 전송한 Frame이 Receiver MAC이 수신하기까지의 지연시간  
Fig. 6. The propagation delay from MAC Layer of transmitter to MAC Layer of receiver.

신호를 Propagation Delay 만큼 지연된 뒤에, Acquisition 하는 것을 알 수 있다. 하지만 UWB는 10m 이내의 근거리 네트워크를 형성하기 때문에 Propagation Delay는 무시할 수 있는 매우 작은 값을 가진다. Long Preamble는 30개의 OFDM Symbol로 구성이 되며,

30개의 OFDM Symbol을 전송하는데 9.375us가 소요된다. Piconet이 어떤 Time Frequency Code를 사용해도 24개의 OFDM Symbol을 전송하게 되면 Frame Synchronization에 관한 Symbol의 전송은 끝나게 된다.<sup>[3]</sup> 이 때로부터 SyncDelay만큼의 시간이 지나면 PHY에서 Phy\_active신호를 1로 인가시키게 된다.

$$\begin{aligned} \text{TotalDelay} &= \text{TxDelay} + \text{Propagation Delay} \\ &+ 24\text{OFDMSymbol} + \text{SyncDelay} \end{aligned} \quad (6)$$

TotalDelay는 Transmitter의 TX\_EN 신호가 1로 인가된 순간부터 Receive의 PHY\_ACTIVE 신호가 1로 바뀌는데 걸리는 시간이다. TotalDelay를 구성하는 각각의 Delay중 Propagation Delay를 제외한 나머지는 모두 고정된 값을 가지며 Propagation Delay는 가변적이기는 하지만, 위에서 언급한바와 같이 매우 작은 값이므로 영향을 미치지 않는다. 그러므로 TotalDelay는 고정된 값이라 말할 수 있다.

나) 1개의 단말기가 piconet에 존재 할 때에 Relative 주파수 오프셋 추정

1개의 단말기가 piconet에 존재 할 때에 주파수 오프셋을 구하는 방법을 설명하고자 한다. UWB 시스템에서 Superframe의 길이는 65536us이며, 단말기들은 Local clock을 통해서 65536us에 해당하는 Superframe Period의 Timing을 구하게 된다. Superframe Period인 65536us와 특정한 값을 가지는 Main clock은 UWB 시스템을 사용하는 모든 단말기들이 이미 약속한 사실이다. (예를 들어 100MHz)

그림 7에서 보듯이 Dev1이 T1에서 Beacon Frame을 전송을 하며, TotalDelay 후에 Dev2는 Dev1이 전송한

Beacon Frame을 수신하게 된다. 마찬가지로 T2에서 Dev1이 Beacon1을 전송하면, TotalDelay후에 Dev2가 Dev1이 전송한 Beacon Frame을 수신하게 된다. 식 6을 통하여 증명한대로 송신 MAC이 전송한 Frame을 수신 PHY가 받아서 MAC에게 Phy\_Enable을 인가하는데 걸리는 TotalDelay는 항상 동일한 값이다.

다음과 같은 가정을 하고 그림 7을 수학적으로 계산해보겠다.

Dev1의 Main Clock : 99.8Mhz

Dev2의 Main Clock : 100Mhz

Superframe의 Period 65536us는 각 단말기의 Main Clock을 이용해서 계산한다.

그림 7에서 Dev1의 Superframe Period (65536us)를 Dev2가 Main Clock으로 Counting을 하게 되면, 6566733개의 clock이 소요됨을 식 7를 통해서 알 수 있다. Dev2는 자신이 6566733개의 clock을 Counting하는 동안에, Dev1은 6553600개의 clock을 소요하였다는 것은 이미 알고 있다. 왜냐하면 모든 단말기는 자신이 정확히 100Mhz라 생각하고 Counting을 하기 때문이다. 65536us동안 Dev1과 Dev2의 clock Count의 차이는 13133만큼 났다.

$$\text{Dev1}'65536\text{us} = \frac{1}{99.8 * 10^6} * 6553600 = 65667.33\text{us}$$

$$\begin{aligned} \text{Dev2}'\text{ClockCountOfDev1}'65536\text{us} &= \frac{656673\text{us}}{100\text{Mhz}} \\ &= 6566733 \end{aligned}$$

$$\text{Dev2}'\text{ClockCountOf}65536\text{us} = 6553600$$

$$\begin{aligned} \text{ClockOffsetBetweenDev1andDev2} &= 6566733 - 6553600 \\ &= 13133 \end{aligned}$$

그림 7에서 Dev1의 Superframe Period (65536us)를 Dev2가 Main Clock으로 Counting을 하게 되면, 6566733개의 clock이 소요됨을 식 7를 통해서 알 수 있다. Dev2는 자신이 6566733개의 clock을 Counting하는 동안에, Dev1은 6553600개의 clock을 소요하였다는 것은 이미 알고 있다. 왜냐하면 모든 단말기는 자신이 정확히 100Mhz라 생각하고 Counting을 하기 때문이다. 65536us동안 Dev1과 Dev2의 clock Count의 차이는 13133만큼 났다.

그림 7에서 볼 수 있듯이 Superframe Period의 2배동안 Searching을 하게 되면, Piconet의 존재여부를

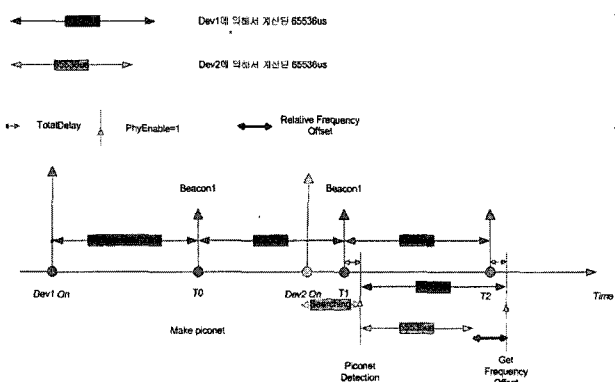


그림 7. 상대적 주파수 오프셋 추정 방법  
Fig. 7. The Proposed relative frequency offset estimation method.

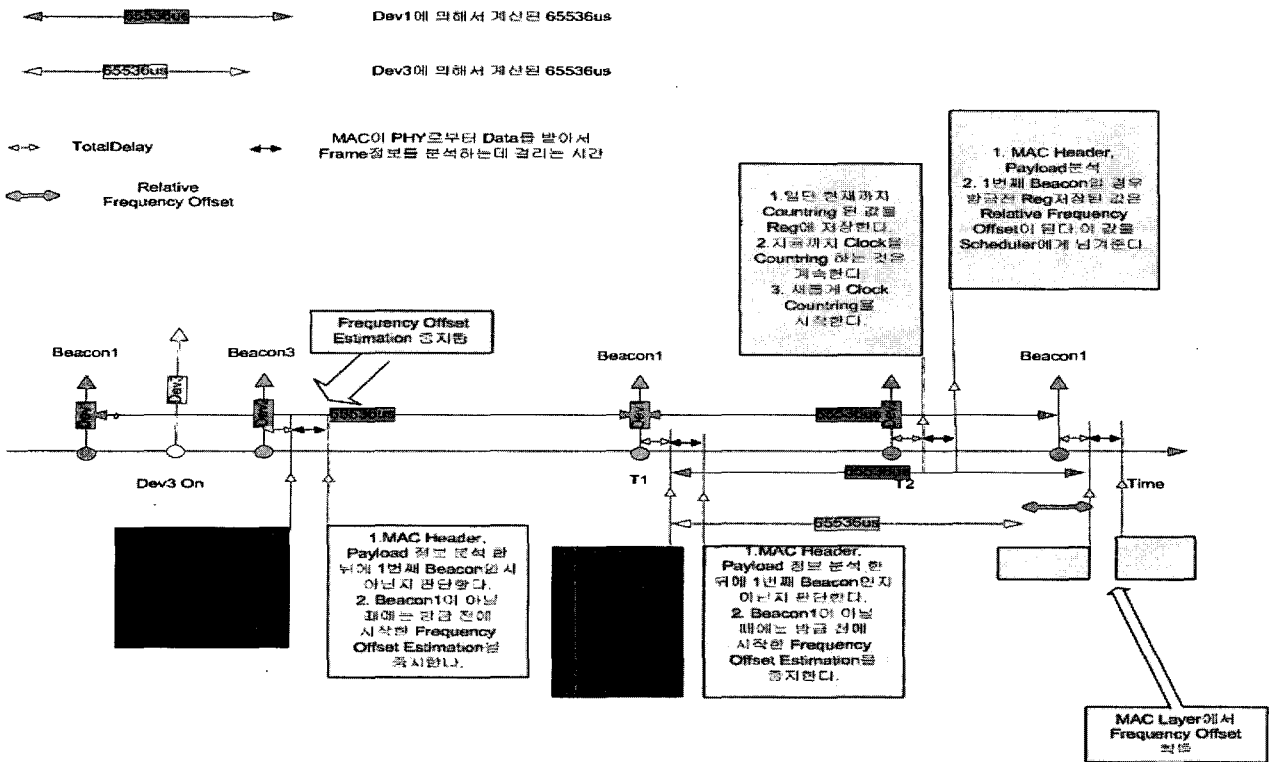


그림 8. 다수의 Device들이 Piconet에 존재할 때, 상대적 주파수 오프셋 Estimation 순서  
 Fig. 8. In a situation where several devices are in a piconet, The proposed relative frequency offset estimation sequence.

확인하며, 만약 Piconet이 존재한다면, Piconet의 BPST (Beacon Period Start Time)의 위치에서 Beacon을 전송하는 단말기와의 상대적 주파수 오프셋을 구할 수 있다.

다) Piconet에 다수의 단말기가 존재할 때의 문제

그림 7과 수식 7은 piconet에 1개의 단말기가 있다고 가정하고 상대적 주파수 오프셋을 구하였다. 그림 7과 같이 Dev1만이 Piconet에 존재하게 될 때에는, Dev2에서 Detection하게 되는 모든

신호는 Dev1이 보낸 것이라 판단할 수 있다. 하지만, 다수의 단말기가 Piconet을 형성하고 있을 때에는 어떻게 상대적 주파수 오프셋을 구하는지 살펴볼도록 하겠다.

그림 8은 Piconet에 다수의 단말기가 존재할 때, Piconet을 Searching하려는 단말기가 상대적 주파수 오프셋을 구하는 과정을 보여준다. 한 예이다. Dev1과 Dev2, 2개의 단말기가 Piconet에 존재할 때, Superframe의 시작점에서 Beacon을 전송하는 Dev1과의 상대적 주파수 오프셋을 구하는 과정이다. Superframe의 시작점에 Dev1이 Beacon을 전송할 때, Dev3은 수신된 신호가

Dev1의 Beacon인지 판단한 다음, 상대적 주파수 오프셋을 구하게 된다. 세부적인 알고리즘은 그림 9를 통하여 이해할 수 있다.

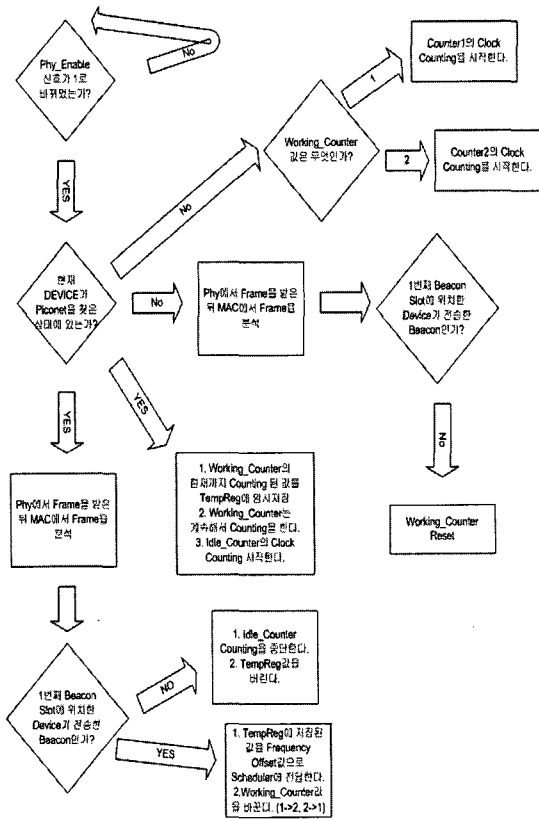
라) 상대적 주파수 오프셋을 이용한 MaxDrift의 제거

그림 7, 8을 통해서 Superframe의 시작점에서 Beacon을 전송하는 단말기와의 상대적 주파수 오프셋을 구하는 방법을 알아보았다. 여기서는 상대적 주파수 오프셋을 이용해서 MaxDrift를 제거하는 알고리즘을 살펴볼 것이다.

그림 10의 분홍색으로 표현된 부분은 Dev1과 Dev2의 MaxDrift를 나타낸다. MaxDrift를 제거하기 위해서는, 상대적 주파수 오프셋을 이용하여 동기를 맞추어야 한다.

그림 10에서, Dev1의 K번째의 MAS position  $MAS_K$ 는 식 8과 같다.

$$\begin{aligned}
 T &= \text{Dev1의 MAS Period} \\
 t &= \text{Dev2의 MAS Period} \\
 R &= \text{상대적 주파수 오프셋} / 256 \\
 MASK &= K * T = K * t + K * R \quad (8)
 \end{aligned}$$



Working\_Counter Register 값이 1이면 Idle\_Counter의 Register 값은 2  
Working\_Counter Register 값이 2이면 Idle\_Counter의 Register 값은 1이 된다.

그림 9. 그림 8에서의 상대적 주파수 오프셋을 구하는데 사용한 알고리즘  
Fig. 9. The algorithm used in Fig 8 to get the relative frequency offset.

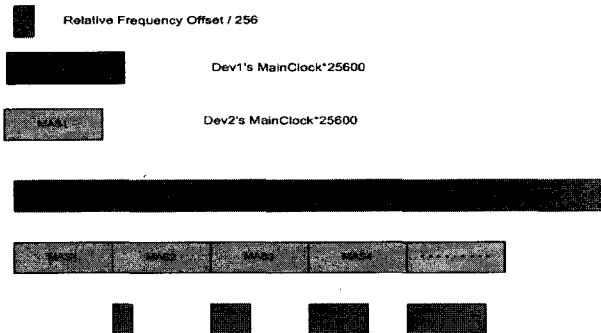


그림 10. 상대적 주파수 오프셋과 MAS Period의 관계  
Fig. 10. The relation between relative frequency offset with MAS period.

Dev2의 MAC에서 식 8을 이용해서 Dev1의 MAS position을 계산하여, 동기를 맞추주면 그림 11와 같다.

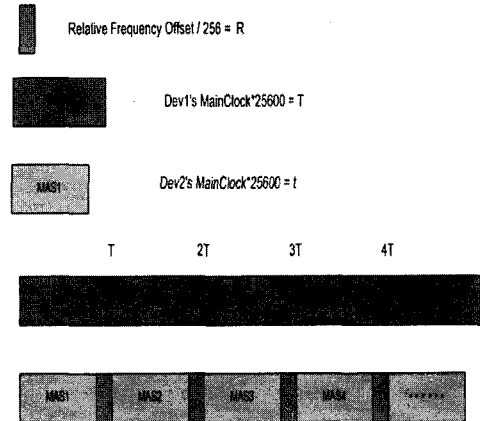


그림 11. 상대적 주파수 오프셋을 이용한 MaxDrift의 제거  
Fig. 11. The elimination of MaxDrift by using relative frequency offset.

### III. 실험 ( Implementation & Simulation )

#### 1. Block Diagram

그림 11에서 상대적 주파수 오프셋을 Acquisition하는데 기술된 알고리즘을 Verilog로 구현하였다. 그림 12에서 timerOffsetGenerator Block이 상대적 주파수 오프셋 Estimation 하는 모듈이다. schedulerTimer Module과 schedulerRegisterSet Module은 상대적 Frequency Offset 값을 timerOffsetGenerator로부터 받아서 Reservation Based Channel Access 방식으로 Frame의 전송을 가능하게 해준다.

#### 2. Synthesis 결과

Target FPGA는 EP20K200EFC672-1이며, Quartus 2,2로 합성을 하였다. MAC Scheduler 전체를 합성한

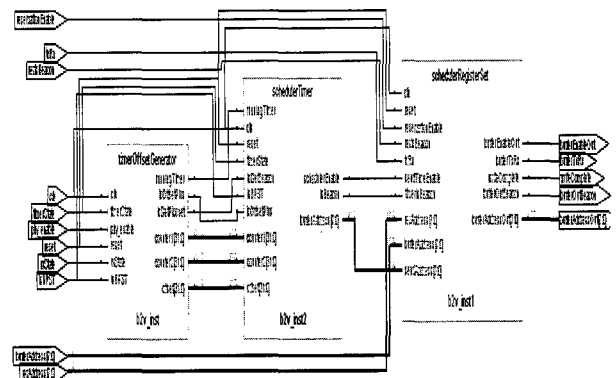


그림 12. MAC Scheduler의 블록 다이어그램  
Fig. 12. The block diagram for MAC scheduler.



결과, EP20K200EFC672-1X 전체의 Logic Cell의 83%에 해당하는 6943개를 차지하였다.

주파수 오프셋 Estimator는 390개의 Logic Cell을 사용했다. 이 값은 Scheduler를 구성하는데 드는 Logic Cell의 5%에 해당하는 작은 값이다. 주파수 오프셋 동기 알고리즘을 하드웨어로 구현하는데 드는 비용은 적

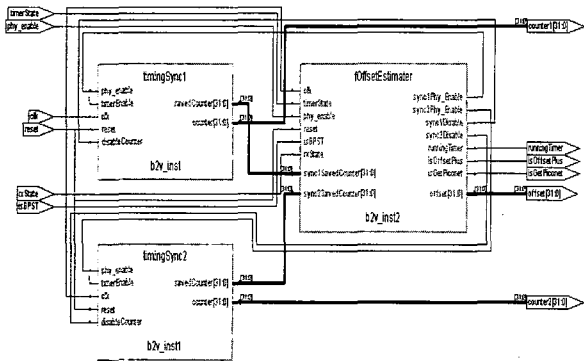


그림 13. 주파수 오프셋 측정기의 블록 다이어그램  
Fig. 13. The block diagram for relative frequency offset estimator.

Timing requirements/analysis status	No requirements
Chip name	fullScheduler
Device for compilation	EP20K200EFC672-1X
Total logic elements	6,943 / 8,320 ( 83 % )
Total pins	37 / 376 ( 9 % )
Total memory bits	0 / 106,496 ( 0 % )
Total PLLs	0 / 2 ( 0 % )
Device for timing analysis	EP20K200EFC672-1X

Logic Cells	Registers	Memory Bits	Pins	Virtual Pins	LUT-Only LCs	Register-Only LCs	LUT/Register
6943 (0)	3208	0	37	0	3735 (0)	2729 (0)	479 (0)

그림 14. MAC Scheduler의 Synthesis결과  
Fig. 14. The Synthesis result of MAC scheduler.

Timing requirements/analysis status	No requirements
Chip name	timerOffsetGenerator
Device for compilation	EP20K200EFC672-1X
Total logic elements	390 / 8,320 ( 4 % )
Total pins	105 / 376 ( 27 % )
Total memory bits	0 / 106,496 ( 0 % )
Total PLLs	0 / 2 ( 0 % )
Device for timing analysis	EP20K200EFC672-1X

Logic Cells	Registers	Memory Bits	Pins	Virtual Pins	LUT-Only LCs	Register-Only LCs	LUT/Register
390 (0)	188	0	105	0	222 (0)	64 (0)	104 (0)

그림 15. 주파수 오프셋 Estimator의 합성결과  
Fig. 15. The Synthesis result of frequency offset estimator.

다고 말 할 수 있다.

3. 시뮬레이션

Modelsim5.7f를 이용해서 시뮬레이션을 수행하였다. Testbench에서 Piconet을 만든 단말기1의 clock을 100.2Mhz로 Piconet에 Join하려는 단말기2의 clock은 100Mhz로 Modeling하였다.

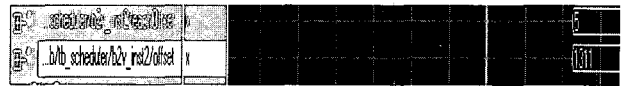


그림 16. 상대적 주파수 오프셋  
Fig. 16. Relative frequency offset.

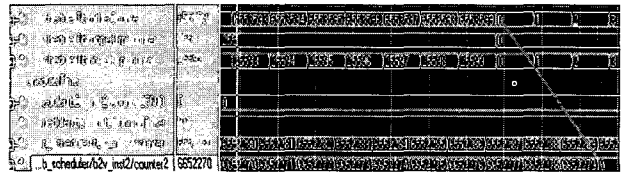


그림 17. 상대적 주파수 오프셋을 이용한 0번째 MAS Position 추정 (Target Device와 2 Clock 차이)  
Fig. 17. 0th MAS position estimation using proposed algorithm (2 clock difference from Target Device).



그림 18. 상대적 주파수 오프셋을 이용한 50번째 MAS Position 추정 (Target Device와 8 Clock 차이)  
Fig. 18. 50th MAS position estimation using proposed algorithm (8 clock difference from Target Device).

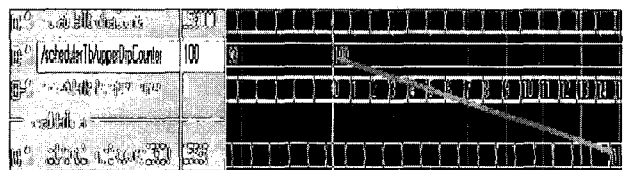


그림 19. 상대적 주파수 오프셋을 이용한 100번째 MAS Position 추정 (Target Device와 14 Clock 차이)  
Fig. 19. 100th MAS position estimation using proposed algorithm (14 clock difference from Target Device).

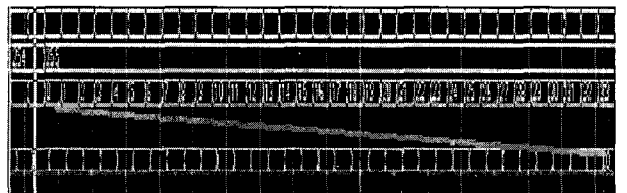


그림 20. 상대적 주파수 오프셋을 이용한 255번째 MAS Position 추정 (Target Device와 33 Clock 차이)  
Fig. 20. 255th MAS position estimation using proposed algorithm (33 clock difference from Target Device).

표 1. 주파수 오프셋 Estimator의 하드웨어 구현 시 Decimal Bit에 따른 estimation오차 범위  
Table 1. The Estimation error range due to decimal bit of the frequency offset estimator.

decimal Bit	Worst MAS ERROR	Best MAS ERROR
0	255 Clock	1 Clock
1	127 Clock	1 Clock
2	63 Clock	1 Clock
3	31 Clock	1 Clock
4	15 Clock	1 Clock
5	7 Clock	1 Clock
6	3 Clock	1 Clock
7	2 Clock	1 Clock
8	1 Clock	1 Clock

그림 16에서 상대적 주파수 오프셋이 1311 임을 알 수 있다. 그림 17-20은 상대적 주파수 오프셋을 알고 있는 상태에서 단말기1의 MAS position과 단말기2의 MAS position의 오차를 보여준다. 그림 18에서 2Clock 차이를 보인 것은, 상대적 주파수 오프셋으로 야기된 Delay가 아닌 Verilog로 구현할 때 Register로 값이 들어오는데 걸리는 Delay다.

그림 20은 255번째의 MAS의 position 오차를 보여준다. 약 33Clock정도의 오차가 발생했다. 33Clock의 오차가 발생한 이유는, 상대적 주파수 오프셋값 1311을 256으로 제산하여 각MAS마다 1311을 256으로 제산한 몫인 5를 감소하는 방식을 택하였기 때문이다. 1311을 256으로 제산할 때, 몫을 정수가 아닌, 소수점으로 표현을 하면 255번째의 MAS에서 position 추정 오차 범위를 최대 1Clock 이내로 감소시킬 수 있다.

표 1은 상대적 주파수 오프셋을 256으로 제한한 몫을 Register에 저장할 때, 몫의 Decimal Bit에 따른 Worst case의 MAS 위치 추정 에러를 보여준다.

#### IV. 결 론

본 논문에서 새롭게 제안한 Time Slot동기 알고리즘은 하드웨어로 구현시 390개의 Logic Cell이 소모되었으며, 이를 통해서 발생 가능한 최대 주파수 오프셋 값

과 관계없이 Device들간의 Time Slot 동기를 이루는 것이 가능하게 해준다. Verilog로 구현된 알고리즘을 Behavioral 시뮬레이션을 통해서 발생 가능한 최대 주파수 오프셋값과 무관하게 1clock오차 이내로 Time Slot동기를 이루는 것을 확인할 수 있었다. 또한 Superframe길이가 65536us이며 256개의 MAS로 구성된 Superframe구조에서 발생가능한 주파수 오프셋이 각각 20ppm, 40ppm, 80ppm일 때, 약 1%, 2% 8%의 전체 시스템 성능의 향상을 가져왔다. 결과를 통해서 알 수 있듯이, 본 논문에서 제안된 알고리즘은 발생 가능한 최대 주파수 오프셋값이 작고, Superframe을 구성하는 MAS의 수가 적다면, 얻을 수 있는 효과가 크지 않다는 한계를 가지고 있다.

#### 참 고 문 헌

- [1] Staderini, E.M., "UWB radars in medicine," Aerospace and Electronic Systems Magazine, IEEE , Vol.17 , pp.13-18 , Jan. 2002.
- [2] Jay O'Conor, "MBOA WIRELESS MEDIUM ACCESS CONTROL(MAC) SPECIFICATION FOR HIGH RATE WIRELESS PERSONAL AREA NETWORKS (WPANS)," MBOA MAC Specification Draft 0.6, Jul. 2004.
- [3] MBOA CONFIDENTIAL, "MBOA PHY LAYER TECHNICAL SPECIFICATION," Ver. 0v891, Jun. 2004.
- [4] Toh, C.-K.,Delwar, M., Allen, D., "Evaluating the communication performance of an ad-hoc wireless network," Wireless Communications, IEEE Transactions on , Vol.1 , pp402-414, Jul. 2002.
- [5] Bill McFarland, David su, John Thomson, "Design and Implementation of an All-CMOS 802.11a Wireless LAN Chipset," Communications Magazine, IEEE, pp160-168, Aug. 2003.
- [6] IEEE Std. 802.11a-1999, "Wireless LAN MAC and PHY Specifications - High-Speed Physical Layer in th 5GHz Band," ISO/IEC 8802-11:1999(E)/Amd 1:2000(E), New York; IEEE, 2000.
- [7] IEEE Std. 802.11b-1999, "Wireless LAN MAC and PHY Specifications," ISO/IEC 8802-11:1999(E), New York; IEEE, 1999.

## — 저 자 소 개 —



오 대 건(정회원)  
 2005년 한양대학교 미디어통신  
 학과 학사 졸업.  
 20005년~한양대학교 미디어통신  
 공학과 석사  
 <주관심분야 : IEEE802.15.3a,  
 IEEE802.15.4a, 통신칩설계>



정 정 화(정회원)  
 1975년 한양대학교  
 전자공학과 졸업 (학사)  
 1977년 한양대학교 대학원  
 전자공학과 졸업(공학석사)  
 1981년 일본 와세다대학교  
 전자공학과 졸업(공학박사)  
 1986년 6월~1987년 9월 미국 Berkeley 대학  
 박사후 과정  
 2000년 10월~2002년 8월 한양대학교  
 정보통신대학원 원장 겸 대학학장  
 2002년~2005년 대한전자공학회 부회장  
 2006년~현재 대한전자공학회 수석부회장  
 <주관심분야 : ASIC Emulation System CAD  
 for V131, HW/SW Co-Design. IEEE 802.15.3a  
 Wireless Communication System >