

# DSSS 동기탐색을 위한 이중 데이터 흐름 경로를 갖는 정합필터

정희원 송명렬\*

## A Matched Filter with Two Data Flow Paths for Searching Synchronization in DSSS

Myong-Lyol Song\* *Regular Member*

요 약

본 논문에서는 DSSS (Direct Sequence Spread Spectrum) 수신기에서 초기동기 탐색에 사용될 수 있는 정합필터에 대해서 연구하였다. 하드웨어기술언어 (HDL)로 표현될 수 있는 단일 데이터 흐름 경로를 갖는 정합필터가 설명되었다. 필터 연산의 처리시간을 개선하기 위해 데이터의 흐름이 이중으로 표현될 수 있도록 식이 정리되고 이와 연관된 하드웨어 모델이 제시되었다. 제안된 모델은 고속 처리를 위해 병렬처리와 파이프라인을 기반으로 하고 일련의 메모리, 곱셈기, 누산기로 구성된 두 개의 데이터 흐름 경로가 평행하게 배열된 구조이다. 제안된 모델에 대해 성능을 분석하였고 단일 데이터 흐름 경로 구조의 정합필터와 비교하였다.

Key words : synchronization, DSSS, matched filter, data flow path, parallelism, HDL

### ABSTRACT

In this paper, the matched filter for searching initial synchronization in DSSS (direct sequence spread spectrum) receiver is studied. The matched filter with a single data flow path is described which can be presented by HDL (Hardware Description Language). In order to improve the processing time of operations for the filter, equations are arranged to represent two data flow paths and the associated hardware model is proposed. The model has an architecture based on parallelism and pipeline for fast processing, in which two data flow paths with a series of memory, multiplier and accumulator are placed in parallel. The performance of the model is analyzed and compared with the matched filter with a single data flow path.

### I. 서론

최근에 초고속 DSP와 고속이면서 집적도가 높은 FPGA의 등장으로 통신시스템에서 변조와 복조를 디지털 연산으로 실행하는 것이 가능하게 되었다. DSSS 통신 시스템에서도 데이터를 대역확산 및 고

주파 대역으로 변조하는 송신기능과 고주파 대역의 신호의 복조 및 역확산과정에 포함되어 있는 연산 요소들을 DSP로 구현하거나 하드웨어기술언어 (Hardware Description Language : HDL)로 표현하여 FPGA나 ASIC (Application Specific Integrated Circuit)으로 구현하는 것이 가능하다<sup>[5][6][7][8]</sup>.

\* 호서대학교 정보통신공학과 (mlsong@dogsuri.hoseo.ac.kr)  
 논문번호 : 030418-0926, 접수일자 : 2003년 9월 29일

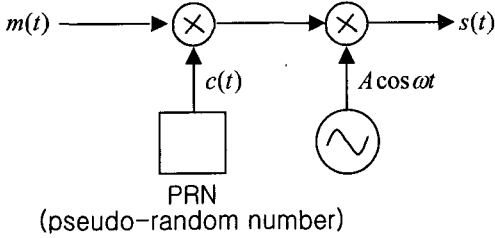


그림 1. DSSS 송신기

DSSS 통신 시스템에서 수신기에 수신된 대역확산 신호에 대하여 정합필터 방식으로 동기를 탐색할 때, 정합필터에 입력되는 데이터의 저장, 저장된 데이터들을 이용한 곱셈, 덧셈 등과 같은 연산 및 이에 수반되는 메모리에 대한 포인터들의 갱신이 반복적으로 수행된다. 이러한 정합필터의 동작을 FPGA에 구현하기 위해 기존의 단일 데이터 흐름 경로를 갖는 모델에서는 데이터의 흐름이 입력된 데이터가 저장된 메모리로부터 데이터를 읽어서 대역확산코드와 곱하고 그 결과가 덧셈기를 통해서 누적되는 1개의 데이터 흐름 경로 상에 존재하는 연산요소들이 파이프라인 방식으로 각 단계의 데이터에 대해 동시에 동작하도록 고안되었다<sup>[4][7]</sup>. 그런데 이 방식은 입력되어 저장되어 있는 각 데이터마다 적용되는 데이터의 흐름이므로, 정합필터의 출력 1개의 값을 구하기 위해 사용되는 데이터의 개수(원도우 크기  $W$ )만큼 반복되어야 한다. 따라서 이렇게 정합필터가 단일 데이터 흐름 경로를 갖는 구조에서는 대역확산코드의 코드율(code rate)이 윈도우 크기와 구현되는 회로가 허용하는 최대 클럭주파수에 의해 제한된다.

그러므로 본 논문에서는 정합필터에서 연산의 처리시간을 개선하기 위한 이중 데이터 흐름 경로를 갖는 정합필터의 모델을 제안한다. 제안된 모델은 일련의 메모리, 곱셈기, 누산기로 구성된 데이터 흐름 경로가 두 개 존재하고 각 경로가 메모리의 서로 다른 주소에 저장된 데이터에 대해 동시에 작용하므로 정합필터의 출력을 계산하는데 소요되는 시간이 반으로 줄어들게 된다.

2장에서는 DS방식의 대역확산된 신호에 대해 초기동기를 탐색할 수 있는 정합필터의 원리와 연산 그리고 기존의 단일 데이터 흐름 경로 구조의 정합필터에 대해서 설명하고 분석한다. 3장에서는 정합필터의 처리시간을 개선하기 위해, 이중 데이터 흐름 경로 구조를 갖을 수 있도록 정합필터의 연산식

을 정리하고 이를 하드웨어로 구현하기 위한 구조를 제안한다. 4장에서는 제안된 정합필터의 성능을 처리시간의 관점에서 분석하고 5장에서 결론을 맺는다.

## II. 초기동기 획득을 위한 정합필터의 구조 및 분석

### 1. 정합필터를 이용한 초기동기 획득 시스템

그림 1에 DSSS 통신 시스템의 송신기 구조를 나타냈다. 전송할 데이터  $m(t)$ 는 +1 또는 -1 이라 하자. 송신기는 반송파를 대역확산 코드에 의해 BPSK방식으로 변조하여 전송하고 대역확산 코드의 대역폭은 기저대역의 데이터의 대역폭에 비해 상당히 크다고 가정한다. 그러면 송신기에서 전송되는 신호는 식 (1)과 같이 표현된다.

$$s(t) = \pm A c(t) \cos \omega t \quad (1)$$

여기서  $c(t)$ 는 대역확산 코드를 의미하고  $A$ 와  $\omega$ 는 각각 반송파의 진폭과 각주파수를 의미한다.

한편, 송신기와 수신기의 위치가 고정되어 있다고 가정하면 도플러효과로 인한 반송파 주파수의 변화는 없다. 따라서 신호  $s(t)$ 가 전송채널을 거쳐서 수신기에 도달되면 잡음, 위상지연, 감쇄가 고려되어야 하므로 수신기에 도달된 신호  $r(t)$ 는 식 (2)와 같이 표현될 수 있다.

$$r(t) = P c(t - T_d) \cos(\omega t + \phi) + n(t) \quad (2)$$

여기서  $P$ 는 수신된 신호의 전력,  $c(t - T_d)$ 는 대역확산코드  $c(t)$ 가 전송채널을 통해 전파되는 과정에서  $T_d$ 만큼 지연된 것,  $\omega$ 와  $\phi$ 는 각각 반송파의 주파수와 위상지연, 그리고  $n(t)$ 는 송신기에서 전송된 신호가 수신기에 도착할 때까지 통신채널에 유입되는 잡음을 의미한다.

본 논문에서는 식 (2)와 같이 표현되는 수신된 신호에 대해 초기동기의 탐색을 위해 일반적으로 사용되는 그림 2와 같은 구조의 정합필터방식 초기 동기 탐색기를 고려한다<sup>[1][2][9]</sup>.

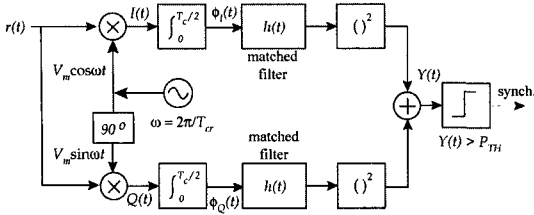


그림 2. 정합필터 방식 초기동기 탐색기

수신기의 국부발전기 국부 발전기에서 발생된 신호  $V_m \cos \omega t$ 와  $V_m \sin \omega t$ 를 수신된 신호  $r(t)$ 에 곱하면 I채널신호  $I(t)$ , Q채널 신호  $Q(t)$ 를 얻고 이들 각각을 적분기를 통과시키면, 식 (3)과 같이 시간이 지연된 확산코드와 반송파의 위상성분만 남는다.

$$\begin{aligned} \phi_I(t) &= A c(t - T_d) \cos \phi \\ \phi_Q(t) &= A c(t - T_d) \sin \phi \end{aligned} \quad (3)$$

여기서  $T_c$ 는 확산코드의 1 chip에 해당하는 시간이고 반송파 주기의 정수배로 설정된다. 그리고  $A$ 는  $P/2 \cdot T_c/2$ 이다. 신호  $\phi_I(t)$ 와  $\phi_Q(t)$  각각을 임펄스응답이  $h(t) = c(-t)$ 인 정합필터를 통과시키고 자승 후 서로 더하면 출력  $Y(t)$ 에는 식 (4)와 같이 반송파의 위상성분이 사라지게 된다.

$$Y(t) = A^2 \cdot \{c(t - T_d) * c(-t)\}^2 \quad (4)$$

따라서 출력  $Y(t)$ 에서는 반송파의 위상과 관계없는 수신된 확산코드  $c(t - T_d)$ 와 역확산에 사용되는 수신기 자체의 코드  $c(t)$ 만의 자기상관함수에 의해 초기동기의 탐색여부가 결정된다.

## 2. 정합필터에서 연산

그림 2에서 I채널과 Q채널에 위치한 정합필터에 각각  $\phi_I(t)$ 와  $\phi_Q(t)$ 가 입력되는데, 두 정합필터의 임펄스응답이 동일하므로, 여기에서는 정합필터의 동작 설명을 위해  $\phi(t)$ 와  $m(t)$ 를 각각 정합필터의 입력과 출력신호,  $h(t)$ 를 정합필터의 임펄스응답으로 표현한다. 임펄스응답을 대역확산코드의 항으로 표현하면 정합필터의 출력은 식 (5)와 같이 컨볼루션(convolution)으로 표현된다.

$$m(t) = \phi(t) * h(t) = \phi(t) * c(-t) \quad (5)$$

이것을 컨볼루션의 정의식으로 표현하면 식 (6)과 같이 표현된다.

$$m(t) = \int_{t_0}^t \phi(\tau) c(\tau - t) d\tau \quad (6)$$

식 (6)에서  $t = 0$ 를 대입하면,

$$m(0) = \int_{t_0}^0 \phi(\tau) c(\tau) d\tau \quad (7)$$

가 된다. 이 식에서  $c(\tau)$ 는  $c(\tau - 0)$ 과 같은 표현이므로 코드의 위상지연이 영인 대역확산코드를 의미한다. 그런데 정합필터의 입력신호  $\phi(t)$ 에는 송신기의 대역확산코드의 지연된 성분  $c(t - T_d)$ 가 포함되어 있으므로 식 (8)과 같이 정리된다.

$$m(0) = K \int_{t_0}^0 c(\tau - T_d) c(\tau) d\tau \quad (8)$$

여기서  $K$ 는 적분변수와 무관한 반송파의 위상성분이나 진폭과 같은 성분을 나타낸다.

식 (8)에서  $m(0)$ 는 시각  $t = 0$ 에서 정합필터의 출력이다. 이 식의 계산은 구간  $(t_0, 0)$ 동안 정합필터에 입력된 신호  $\phi(t)$ 와 수신기에서 자체적으로 발생하는 대역확산코드  $c(t)$ 의 곱을 적분하는 것이다. 그런데 대역확산코드  $c(t)$ 는 수신기 자체적으로 발생되므로 수신기에서 코드의 위상을 임의로 지정할 수 있다. 따라서  $m(0)$ 는 위상이 0으로 설정된 수신기의 확산코드  $c(t)$ 를 기준으로 할 때  $\phi(t)$ 에 포함되어 있는 지연되어 수신된 송신기 확산코드의 지연  $T_d$ 의 정도를 나타내는 수단이 될 수 있다.

이제 식 (7)을 디지털 연산으로 수행하기 위해서는 정합필터에 입력되는 신호  $\phi(t)$ 를 일정한 속도로 표본화해야 한다.  $T_s$ 를 정합필터에 입력되는 신호에 대한 표본화 주기라 하고,  $\phi_k$ 를 시각  $t = kT_s$ 에서 정합필터의 입력신호  $\phi(t)$ 의 표본화

된 신호라 한다. 그리고  $m_k(0)$ 를 시각  $t = kT_s$ 에서 계산된 정합필터의 출력이라 하자.  $m_k(0)$ 를 계산하기 위해서는  $\phi_k$ 를 포함하여 과거에 입력되었던 신호들이 필요하다. 그러나 과거의 신호를 저장할 메모리의 용량과 출력 계산을 위한 연산량 증가로 인해 개수가  $W$ 로 제한된 표본  $\{\phi_{k-W+1}, \phi_{k-W+2}, \dots, \phi_k\}$ 만  $m_k(0)$ 의 계산에 이용한다. 그리고 기준위상이 되는 수신기의 대역확산코드에 대한 표본값은  $\{c_{-W+1}, c_{-W+2}, \dots, c_{-1}, c_0\}$ 을 적용한다. 여기서  $c_{-i} = c(-iT_s)$ 이다. 그러면 식 (7)에 대응하는 이산신호입력에 대한 정합필터의 출력  $m_k(0)$ 는 식 (9)와 같이 디지털 연산으로 처리할 수 있게 되고  $\Phi_k$ 와  $C$ 의 내적이 된다.

$$m_k(0) = \sum_{i=0}^{W-1} \phi_{k-W+1+i} \cdot c_{-W+1+i} \quad (9)$$

$$= \Phi_k \cdot C$$

여기서  $\Phi_k = \{\phi_{k-W+1}, \phi_{k-W+2}, \dots, \phi_k\}$ 이고  $C = \{c_{-W+1}, c_{-W+2}, \dots, c_{-1}, c_0\}$ 이다.

새로운 데이터가 입력될 때마다 식 (9)와 같은 과정을 반복하면, 디지털 연산에 의한 정합필터의 출력은 식 (10)에 표현된 원소와 같은 값을 차례로 갖게 된다.

$$\{\dots, \Phi_{k-1} \cdot C, \Phi_k \cdot C, \Phi_{k+1} \cdot C, \dots\} \quad (10)$$

### 3. 단일 데이터 흐름 경로 구조의 정합필터

초기동기 탐색기는 송신기에서 전송한 파형에 포함되어 있는 대역확산코드의 위상이 수신기 자체의 대역확산코드에 대해  $T_c/2$ 이내에 위치하는가를 판정해야 한다. 따라서 정합필터에 입력되는 신호에 대한 표본화 주기  $T_s$ 는 식 (11)과 같은 조건을 만족해야 한다.

$$T_s \leq T_c/2 \quad (11)$$

한편, 식 (9)를 풀어서 쓴 후, 덧셈의 순서를 바꾸면 식 (12)와 같이 정리된다.

$$m_k(0) = \sum_{i=0}^{W-1} \phi_{k-i} \cdot c_{-i} \quad (12)$$

이제, 정합필터의 표본화 주기를  $T_s = T_c/2$ 라 가정하면, 확산코드 1 chip의 시간 동안 2개의 표본과 확산코드값이 필요하므로 정합필터의 출력은 식 (13)과 같이 표현될 수 있다.

$$m_k(0) = \sum_{i=0}^{W-1} \phi_{k-i} \cdot c_{-\lfloor i/2 \rfloor} \quad (13)$$

여기서  $\lfloor x \rfloor$ 는  $x$ 보다 크지 않은 최대 정수를 의미한다.

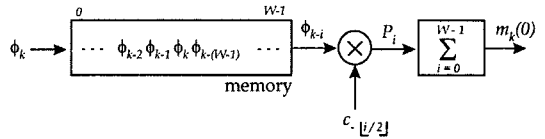
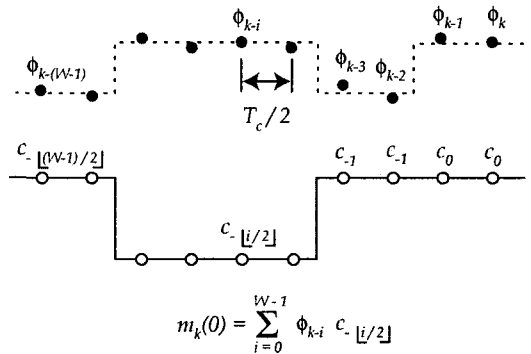


그림 3. 단일 데이터 흐름 경로의 정합필터 구조

그림 3에 식 (13)을 하드웨어로 구현하기 위한 단일 데이터 흐름 경로 구조의 정합필터를 나타냈다. 이 단일 데이터 흐름 경로 구조는 입력과 출력 사이에서 하나의 경로를 따라 배치된 메모리, 곱셈기, 누산기와 같은 연산 요소들을 데이터가 차례대로 통과하게 되어 있다.



$$m_k(0) = \sum_{i=0}^{W-1} \phi_{k-i} \cdot c_{\lfloor i/2 \rfloor}$$

그림 4. 단일 데이터 흐름 경로 구조의 정합필터에서 연산 방법

그림 4에 그림 3의 단일 데이터 흐름 경로 구조의 정합필터에서 식 (13)에 대한 계산과정을 나타냈다. 여기에서 정합필터의 출력  $m_k(0)$ 는 가장 최근에

입력된 신호  $\phi_k$ 에서 시작하여 가장 오래 된 신호  $\phi_{k-W+1}$ 의 순서로 계산된다.

### III. 이중 데이터 흐름 경로 구조의 정합필터

정합필터의 출력값을 계산하기 위한 식 (13)을 하드웨어로 구현할 때, 단일 데이터 흐름 경로의 구조에서는 정합필터에 입력된 값  $\{\phi_{k-W+1}, \phi_{k-W+2}, \dots, \phi_k\}$ 에 대해서 식 (13)과 같이 곱셈과 덧셈이 순차적으로 이루어진다. 따라서 처리시간은 계산에 사용되는 데이터의 양에 비례한다. 이 논문에서는, 정합필터를 하드웨어로 구현할 때, 단일 흐름 경로 구조에서 정합필터의 출력 계산에 사용되는 것과 동일한 양의 데이터에 대해 병렬로 처리될 수 있는 요소들을 증가시킴으로써 처리시간을 단축한다.

연산요소들이 병렬로 처리되는 것을 나타내기 위해, 식 (13)의 계산을 짝수 번째에 입력된 데이터에 대한 연산과 홀수 번째에 입력된 데이터에 대한 연산으로 분리하여 식 (14)와 같이 정리할 수 있다.

$$m_k(0) = \sum_{i=0}^{W/2-1} \phi_{k-2i} \cdot c_{-i} + \sum_{i=0}^{W/2-1} \phi_{k-(2i+1)} \cdot c_{-i} \quad (14)$$

이 논문에서 제안하는 이중 데이터 흐름 경로 구조의 정합필터는 식 (14)와 같이 표현된 연산식의 첫 번째 항과 두 번째 항을 동시에 하드웨어로 처리하는 방식이다.

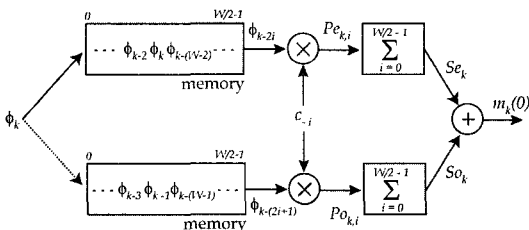


그림 5. 이중 데이터 흐름 경로 구조의 정합필터

그림 5에 제안된 이중 데이터 흐름 경로를 갖는 정합필터의 구조를 나타냈다. 제안된 구조의 정합필터는  $W/2$ 개의 입력 데이터를 저장할 수 있는 메모리와 곱셈기, 누산기가 각 데이터 흐름 경로를 구성하고 누산기의 결과가 덧셈기에서 서로 더해진다.

입력되는 데이터는 두 개의 메모리에 교대로 저장된다. 그리고 각 데이터 흐름 경로마다 메모리에 저장되어 있던 데이터를 동시에 읽어서 대역확산 코드와 곱하면 곱한 결과  $Pe_{k,i}$ 와  $Po_{k,i}$ 를 각각 얻게 된다. 그리고 이 결과는 동시에 각 경로의 누산기에서 더해진다. 이러한 동작을  $W/2$ 회 반복하면 누산기의 결과는 각각  $Se_k$ 와  $So_k$ 이 되고 각 누산기의 출력을 서로 더하면 정합필터의 출력을 얻게 된다. 그러면 동일한 개수의 입력 데이터에 대해 정합필터의 출력값의 계산시간은 단일 데이터 흐름 경로 구조에 비해서 약 1/2정도로 단축될 수 있다.

여기에서 제안된 이중 데이터 흐름 경로를 갖는 정합필터는 각 경로마다 곱셈기와 누산기를 가지므로 단일 데이터 흐름 경로에 비해서 두 배의 연산장치를 필요로 한다. 그러나 연산에 사용되는 데이터의 개수는 변화가 없으므로 필요한 메모리 크기는 단일 경로와 같고 데이터를 절반씩 나누어 동시에 처리하므로 처리시간이 감소되는 특징을 갖고 있다.

### IV. 성능분석 및 결과고찰

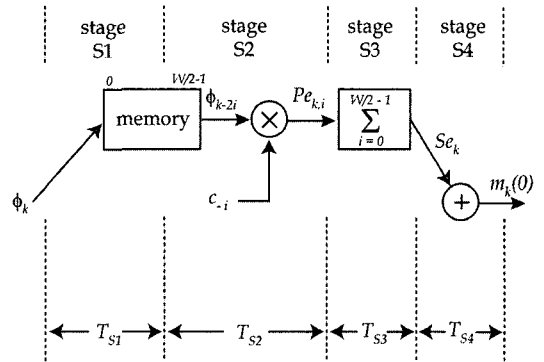


그림 6. 4단계로 구분된 상부 데이터 흐름 경로

이 절에서는 제안된 이중 데이터 흐름 경로 구조의 정합필터에서 데이터가 각 요소들을 통과하여 출력에 이르는 시간을 분석한다. 그림 5에 두 개의 데이터 흐름 경로가 상부와 하부에 평행하게 병렬로 배치되어 있는데 각 경로의 메모리에 저장되어 있는 데이터가 동시에 읽혀져서 마지막 단 덧셈기 앞의 누산기까지 각 데이터 흐름 경로를 동일한 단계를 거쳐서 이동한다. 따라서 그림 6과 같이 동일

한 두 개의 데이터 흐름 경로 중 상부 하나만 표시하여 설명한다.

그림 6에서 단계 S1에서는 정합필터에 입력되는 데이터가 메모리에 저장된다. 단계 S2에서는 메모리에 저장되어 있던 데이터와 대역확산 코드가 서로 곱해진다. 단계 S3에서는 단계 S2의 계산결과가 누적되어 더해진다. 단계 S4에서는 상부와 하부경로의 누산기 결과가 서로 더해져서 정합필터의 출력을 얻게 된다. 데이터가 단계 S1, S2, S3, S4의 각각에 입력되어 각 단계의 출력에 전달되는 시간을 각각  $T_{S1}$ ,  $T_{S2}$ ,  $T_{S3}$ ,  $T_{S4}$ 라 하자.

한편, 그림 6에서 각 단계의 기능이 다르고 이를 구현할 때 필요한 게이트 수와 구조가 다르므로 각 단계에서 처리시간  $T_{S1}$ ,  $T_{S2}$ ,  $T_{S3}$ ,  $T_{S4}$ 도 서로 다르다. 그러나 제안된 정합필터를 동기식 회로로 적절히 설계하면 각 단계에서의 연산이나 전달시간이 식 (15)의 관계와 같이 1 클럭주기  $T_{CLK}$  보다 작게 되도록 조절할 수 있다.

$$\max \{ T_{S1}, T_{S2}, T_{S3}, T_{S4} \} < T_{CLK} \quad (15)$$

clock cycle	stage S1	stage S2	stage S3	stage S4	output
-2			$Pe_{k-1, W/2-1}$		
-1				$Se_{k-1}$	
0	$\phi_k$				$m_{k,1}(0)$
1		$\phi_k, c_0$			
2		$\phi_{k-2}, c_{-1}$	$Pe_{k,0}$		
3		$\phi_{k-4}, c_{-2}$	$Pe_{k,1}$		
4		$\phi_{k-6}, c_{-3}$	$Pe_{k,2}$		
...		...	...		
$W/2$		$\phi_{k-(W/2)}, c_{-(W/2-1)}$	$Pe_{k, W/2-2}$		
$W/2+1$			$Pe_{k, W/2-1}$		
$W/2+2$				$Se_k$	
$W/2+3$	$\phi_{k+1}$				$m_{k,1}(0)$

그림 7. 제안된 정합필터의 각 단계에서 나타나는 데이터

그림 7은 제안된 이중 데이터 흐름 경로구조의 정합필터의 성능을 분석하기 위해 정합필터의 각 단계에 나타나는 데이터를 클럭사이클의 진행에 따라 표현한 것이다. 이 그림은 그림 6과 같이 정합필터의 상부 경로에서 나타나고 처리되는 데이터를 표현한 것이고 하부 경로는 상부 경로와 동일한 구

조이고 또한 동시에 처리되므로 그림 5의 하부 경로에 표현된 데이터로 바꾸어서 생각하면 된다.

설명을 위해, 표본화된 신호  $\phi_k$ 가 상부경로 메모리의 입력포트에 나타나고 있는 시점 (클럭사이클 0)을 기준시점으로 정한다. 그리고 이 데이터는 클럭사이클 1이 되기 전에 버퍼에 저장된다. 항  $\phi_k, c_0$ 는 메모리에 기억되어 있던  $\phi_k$ 와 코드참조 표에 기록되어 있던  $c_0$ 가 클럭사이클 1에서 읽혀져서 단계 S2 (곱셈기)의 입력에 나타나는 것을 의미한다.  $\phi_k$ 와  $c_0$ 의 곱  $Pe_{k,0}$ 는 클럭사이클 2에서 단계 S3 (누산기)의 입력포트에 나타난다. 그리고 이와 동시에  $\phi_{k-2}, c_{-1}$ 가 읽혀져서 단계 S2의 입력에 나타난다. 이와 같은 동작을 반복하면 항  $\phi_{k-(W-2)}, c_{-(W/2-1)}$ 는 클럭사이클  $W/2$ 에서 단계 S2에 입력되고 이들의 곱셈결과  $Pe_{k, W/2-1}$ 은 클럭사이클  $W/2+1$ 에 단계 S3에 입력된다. 단계 S3에서 최종계산 결과  $Se_k$ 는 클럭사이클  $W/2+2$ 에서 단계 S4 (덧셈기)에 입력된다. 그리고 이와 동시에 하부 경로의 단계 S3의 계산결과  $So_k$ 가 단계 S4의 또 다른 입력에 나타난다. 상부와 하부 경로의 누산기의 결과들은 단계 S4에서 서로 더해진 후 클럭사이클  $W/2+3$ 에서 비로소 정합필터의 출력에  $m_k(0)$ 의 형태로 나타난다.

따라서 클럭사이클 0에서 입력되었던 데이터  $\phi_k$ 에 대한 정합필터의 출력  $m_k(0)$ 은 클럭사이클  $W/2+3$ 부터 출력에 나타나게 된다. 그러므로 메모리에 어떤 입력데이터가 저장된 후, 정합필터의 출력계산에 있어서 순수 연산에만 소요되는 시간  $T_{PROC}$ 은 식 (16)과 같이  $W/2+2$ 개의 클럭사이클이 필요하다.

$$T_{PROC} = (W/2 + 2) T_{CLK} \quad (16)$$

그리고 정합필터에 입력되는 데이터가 메모리에 저장되는 시간을 포함하면, 입력신호의 표본화 주기  $T_S$ 와 정합필터 클럭  $T_{CLK}$ 의 관계는 식 (17)과 같이 표현된다.

$$T_S \geq T_{CLK} + T_{PROC} = (W/2 + 3) T_{CLK} \quad (17)$$

한편, 단일 데이터 흐름 경로 구조에서 표본화 주기와 클럭의 관계는 식 (18)과 같이 표현되었다<sup>7)</sup>.

$$T_S \geq (W + 2) T_{CLK} \quad (18)$$

그러므로 단일 데이터 흐름 경로구조와 제안된 이중 데이터 흐름 경로 구조에서 최소 표본화 주기를 각각  $T_{SS, \min}$ ,  $T_{SD, \min}$ 이라 할 때, 이들은 다음과 같이 표현된다.

$$T_{SS, \min} = (W + 2) T_{CLK}$$

$$T_{SD, \min} = (W/2 + 3) T_{CLK}$$

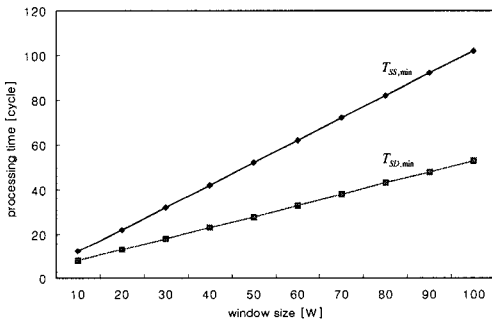


그림 8. 정합필터의 출력 계산시간 비교

그림 8에 윈도우의 크기 변화에 따른 최소 표본화 주기를 나타냈다. 그림에서 보는 바와 같이 정합필터 출력 계산에 사용되는 표본수  $W$ 가 클수록 제안된 이중 데이터 흐름 경로 구조와 단일 경로 구조의 처리시간의 차이가 점점 커지고 단일 경로 구조의 1/2에 근접함을 알 수 있다. 이 의미는 정합필터에 입력되는 데이터에 대한 표본화율의 관점에서 보면, 제안된 구조로 약 2배 정도 표본화율의 개선효과를 얻을 수 있다는 것을 의미한다.

위에서 분석한 바와 같이 처리시간이 단축되는 이유는 이중 데이터 흐름 경로로 데이터가 병렬로 처리 되고 각 경로에서 처리된 결과를 서로 더하므로 병렬처리의 단계수가 기존의 단일 데이터 흐름 경로에 비해서 1단계 늘어났지만, 각 경로에서 처리되는 데이터 수는 반으로 줄기 때문에 성능개선 효과를 얻을 수 있다.

한편, 이와 같은 방법을 데이터의 흐름경로가 두 개 보다 많은  $n$ 개인 경우로 확장하면 최초로 탐색되는 코드의 위상오차가  $T/n$  이내로 좀더 줄어

들고 처리시간도 줄어드는 효과를 얻을 수 있다. 그러나 이와 동시에 증가하는 하드웨어 성분도 고려해야 하는 문제점도 고려해야 한다.

## V. 결 론

본 논문에서는 정합필터방식 초기동기 탐색기에 대해서 분석하였다. 정합필터의 동작을 디지털 연산으로 처리하기 위한 식을 정리하였다. 기존의 단일 데이터 흐름 경로 구조의 정합필터에 대해서 설명하였다. 단일 데이터 흐름 구조의 정합필터 보다 처리시간을 단축하기 위해, 정합필터의 출력값을 계산하는데 필요한 연산 요소들을 분석하여 데이터의 흐름이 이중으로 처리될 수 있는 식을 유도하였다. 그리고 유도된 연산식을 하드웨어로 구현할 때 참조할 수 있는 이중 데이터 흐름 경로 구조의 정합필터를 제안하였다.

제안된 이중 데이터 흐름 경로 구조의 성능지표로서 정합필터의 입력신호에 대한 최소 표본화 주기에 대한 분석은 기존의 단일 데이터 흐름 경로 구조에 비해서 출력계산에 사용되는 입력 데이터 수가 증가할수록 1/2로 감소하여 성능이 개선됨을 보이고 있다. 따라서 제안된 이중 데이터 흐름 구조의 정합필터는 기존 단일 경로 구조의 정합필터에 적용되는 대역확산 코드의 코드율 보다 약 2배 정도 빠른 코드율을 필요로 하는 대역확산통신 시스템에 적용할 수 있다.

한편, 제안된 구조는, 하드웨어로 구현시 정합필터의 출력을 얻는데 필요한 데이터의 처리시간을 단축하기 위해, 파이프라인과 병렬처리 개념을 기초로 고안된 것이다. 기존의 단일 흐름 경로 구조 보다 병렬로 처리할 수 있는 요소가 증가되므로 구현되는 하드웨어의 게이트 수도 증가됨을 고려해야 한다.

향후에는 제안된 구조를 verilog HDL이나 VHDL과 같은 하드웨어 기술언어로 표현하고 FPGA에 구현하여 발생될 수 있는 문제점들에 대해 연구해야 한다. 그리고 이에 대한 신뢰성을 확보하고 데이터 윈도우의 크기, 표본화율, 데이터 경로의 크기, 대역확산 코드 설정 등과 같은 정합필터의 특성을 결정하는 요소를 임의로 지정할 수 있는 IP(intellectual property)에 대한 연구가 필요하다.

참 고 문 헌

[1] Robert C. Dixon, *Spread Spectrum Systems with Commercial Applications*, 3rd Edition, John Wiley & Sons, 1994.

[2] Vladan M. Jovanovic and Elvino S. Sousa, "Analysis of Non-Coherent Correlation in DS/BPSK Spread Spectrum Acquisition", *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, Part: 1, pp. 565-573, Feb. 1995.

[3] Roger L. Peterson, Rodger E. Ziemer, and David E. Borth, *Introduction to Spread Spectrum Communications*, Prentice-Hall, 1995.

[4] Myong-Lyol Song, "The Architecture of High-Speed Matched Filter for Searching Synchronization in DSSS Receiver", *8th IEEE International Conference on Communication Systems 2002*, Vol.8, pp.883-886, Nov. 2002

[5] 김진, 조중휘, "직접확산통신을 위한 기적 대역 MODEM의 VLSI 구현", *전자공학회 논문지*, 제 33권, C편, 제 8호, pp. 541-547, 1997.

[6] 김진천, 박홍준, 임형수, 전경훈, "대역 제한된 직접 시퀀스 CDMA 확산 대역 신호를 위한 전 디지털 부호 획득 및 추적 루우프의 FPGA 구현", *전자공학회 논문지*, 제 33권, A편, 제 5호, pp. 893-899, 1996.

[7] 송명렬, "DSSS 수신기에서 동기탐색을 위한 고속 정합필터", *한국통신학회 논문지*, 제 27권, 제 10C호, pp.999-1007, 2002.

[8] 심복태, 박종현, 이홍식, 김제우, 김관옥, "Spread Spectrum 방식을 이용한 무선 LAN MODEM의 구현", *전자공학회 논문지*, 제 32권, A편, 제 1호, pp. 1-13, 1995.

[9] 이정훈, 이충웅, "비동기식 디지털 상관기를 이용한 직접부호계열 확산신호의 초기동기에 관한 연구", *전자공학회 논문지*, 제 24권, 제 1호, pp. 1-9, 1987.

송 명 렬(Myong-Lyol Song)

정회원



1985년 2월 : 연세대학교  
전자공학과 학사

1988년 2월 : 연세대학교  
전자공학과 석사

1996년 2월 : 연세대학교  
전자공학과 박사

1984년 12월~1986년 2월 :  
삼성전자 연구원

1988년 12월~1991년 7월 : 육군사관학교  
전자과 전임강사

1996년 3월~현재 : 호서대학교  
전기정보통신공학부 조교수

<주관심분야> 컴퓨터통신, 통신시스템, 디지털시스  
템 설계, 임베디드 시스템