

VHDL을 이용한 고정 소수점 디지털 신호처리 알고리즘 개발 소프트웨어

(VHDL Based Fixed-point Digital Signal Processing Algorithm Development Software)

琴基一*, 成元鎔**

(Ki-Il Kum and Wonyong Sung)

要約

고정소수점 디지털 신호처리 알고리즘의 설계 환경이 VHDL(VHSIC Hardware Description Language)을 기반으로 개발되었다. 본 설계 환경에서는 디지털 신호처리 알고리즘을 블록 다이어그램으로 표현한 뒤, 부동소수점 VHDL 프로그램으로 변환한 후 모의실험을 하여서 알고리즘을 검증한다. 다음에, 부동소수점 알고리즘은 적절한 소수점의 위치와 단어길이를 가지는 고정소수점 VHDL 프로그램으로 자동 변환된다. 이때, 소수점의 위치와 단어 길이는 신호처리 시스템의 설계 사양에서 주어진 성능을 만족하게 하면서 최소의 하드웨어를 사용하도록 결정된다. 부동소수점 알고리즘의 고정소수점 변환을 빠르게 하기 위해, 같은 단어길이를 가지게 될 신호들을 묶는 (group) 방법과 설계 공간을 빠르게 탐색하기 위한 방법들이 연구되었다. 변환된 고정소수점 VHDL 프로그램은 상용화된 VHDL 합성기를 사용하여 바로 ASIC등의 하드웨어로 구현할 수 있다. 따라서, 본 설계 환경은 신호처리 알고리즘의 개발 부터 고정소수점 구현까지의 작업을 일관되게 처리할 수 있게 해 준다.

Abstract

Software for the development of fixed-point digital signal processing algorithms is developed based on VHDL(VHSIC Hardware Description Language). At this developed CAD software, a floating-point VHDL program for the simulation of a digital signal processing algorithm is obtained from the signal flow graph. Then, the floating-point program is automatically converted to a fixed-point version based on the system level performance measure such as SQNR (Signal to Quantization Noise Ratio) and the cost model of the hardware components employed. In this step, methods for reducing the search space are studied to minimize the design time. The fixed-point VHDL program can be implemented in ASIC's by using commercially available logic synthesizers. Therefore, this software can provide a seamless design environment from the development of floating-point digital signal processing algorithms to the fixed-point implementation of them.

*正會員, 서울대학교 制御計測工學科
(Dept. of Control and Instrumentation,
Seoul Nat'l Univ.)

**正會員, 서울대학교 制御計測工學科 및 半導體

共同研究所
(Dept. of Control and Instrumentation,
and ISRC, Seoul Nat'l Univ.)

接受日字: 1994年 2月 12日

1. 서론

디지털 신호처리 알고리즘을 고정소수점 연산 하드웨어를 이용해서 ASIC등으로 구현할 때에는, 각 신호를 표현하는 단어길이(word-length)가 길어질수록 하드웨어 비용이 증가함은 물론 동작 속도에서도 불리해지므로 가능한 한 단어길이를 짧게 해야 한다. 그러나, 단어길이가 짧아질수록 양자화 오차가 커지고 오버플로우가 일어날 확률이 커져 시스템의 성능은 떨어지게 된다. 그러므로 이 두 가지, 즉 하드웨어의 비용 및 설계 성능을 만족하는 적절한 단어길이를 결정해야 한다. 종래에는 부동소수점 데이터 형식으로 표현된 신호처리 알고리즘을 고정소수점 형식으로 바꾸기 위해서 L_p , norm 등을 이용하여 단어 길이를 구하는 방법들이 쓰였는데, 이 방법은 IIR 또는 FIR 디지털 필터와 같은 선형 시불변 시스템(linear time-invariant system)에만 적용될 수 있다.^[1] 한편 최근에 고정소수점 계산으로 알고리즘을 시뮬레이션 할 수 있는 설계 환경들이 개발되었으나^[2] 아직까지는 일반적인 신호처리 알고리즘에 대해 적절한 단어길이를 할당하는 연구는 되어 있지 않다. 또한 기존의 디지털 신호처리 시스템의 구현을 위한 설계 환경에서는 알고리즘이 신호흐름도(signal flow graph)등으로 표현된 후 부동소수점 형식으로 모의 실험되는 반면에, ASIC으로 구현하기 위한 하드웨어는 VHDL(VHSIC Hardware Description Language)등의 언어로 표현되어야 하는 문제점이 있다.^[3] 본 연구된 방법에서는 신호흐름도로 표현된 디지털 신호처리 알고리즘을 부동소수점 VHDL 프로그램으로 바꾸어서 모의실험한 후, 내부 신호의 최적 단어길이를 모의실험을 이용하여 구한 후, 이를 이용하여 고정소수점 VHDL 프로그램을 생성한다.

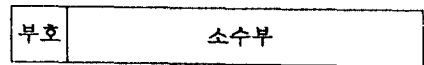
본 논문의 구성은 다음과 같다. 제 II장에는 유한 단어길이 디지털 신호처리 시스템의 표현 및 성능 평가 방법이 설명되었고, 제 III장에는 VHDL을 이용한 신호처리 시스템의 표현과 모의실험 방법이, 제 IV장에는 디지털 신호처리 시스템의 최적 단어길이를 구하는 과정이 설명되어 있다. 제 V장에는 본 개발된 알고리즘을 이용해서 LMS 적응 알고리즘을 채용한 channel identification system의 설계 예를 보이고 제 VI장에서 본 연구의 결론을 맺는다.

II. 유한 단어길이 시스템의 표현 방법 및 성능 평가

1. 신호의 고정소수점 표현

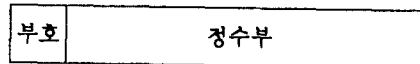
지금까지 고정소수점 데이터 형식으로는 소수형

(fractional format)과 정수형(integer format)이 널리 쓰여 왔는데, 전자는 그림 1에 보이는 바와 같이 소수점이 sign 바로 밑에 있고, 후자는 맨 마지막 비트(LSB : Least Significant Bit) 다음에 있는 경우이다. 이 방식들에서는 표현할 수 있는 수의 영역이 소수형의 경우에는 -1 과 1 사이의 값, 그리고 정수형의 경우에는 정수로 고정되어 있으므로, 고정 소수점 구현을 위해서는 신호처리 알고리즘을 다시 스케일링(scaling)해야 하는 불편함이 있다. 이러한 문제를 해결하기 위하여 그림 1에 보는 바와 같이 소수점을 각 신호마다 독립적으로 주는 혼합형(mixed binary-point format)이 채용되었다.^{[2][7][8]} 즉, 이 방법에서는 그림 1에서 보는바와 같이 소수점은 임의의 위치에 있을 수 있기 때문에 정수부와 소수부의 단어길이를 모두 가질 수 있다. 이때 정수부와 소수부의 단어길이에 부호 1 비트를 더하면 전체 단어 길이와 같으며, 각 부분의 단어길이는 음수를 포함하는 정수의 값을 가진다. 예를들어, 정수부의 단어길이가 4이고 소수부의 단어길이가 6이라면, 이 신호는 -16 과 +16사이의 값을 2^6 의 양자화 레벨로 나타낸다.



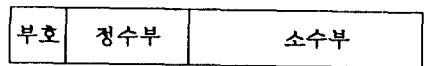
^ 소수점

(a) 소수형 표현



^ 소수점

(b) 정수형 표현



^ 소수점

(c) 사용된 방법

그림 1. 고정소수점 수의 표현 형식

Fig. 1. The formats of fixed point numbers.

정수부의 단어길이가 크면 표현할 수 있는 신호의 최대 크기가 증가한다. 따라서, 정수부의 단어길이 w_i 는 그 신호의 최대 크기, 즉 범위에 의해 아래 식 1과 같이 결정 된다.

$$w_i = \lceil \log_2 r(x) \rceil \quad (1)$$

단, x 는 x 보다 작지 않은 최소의 정수이다.

여기서 각 신호들의 범위 $r(x)$ 는 그 신호의 평균과 표준편차를 이용하여 다음과 같은 식으로 추정한다.^[7]

$$r(x) = m(x) + n \cdot \sigma(x) \quad (2)$$

위의 식에서 $m(x)$ 는 어떤 신호 x 의 평균값이고 $\sigma(x)$ 는 표준편차이다. 여기서 n 은 보통 4 에서 16 사이의 값을 사용하는데, 이 값이 작은 경우에는 단어 길이가 짧아지나 오버플로우의 가능성이 커진다.

한편, 소수부 단어길이가 짧아질수록 그 신호에는 양자화 오차가 더 많이 혼합되어 있게 된다. 이때, 소수부 단어길이를 w_i 라 하면, 양자화 오차는 반올림의 경우 -2^{-w_i} 과 2^{-w_i} 사이에 있게 된다. 이러한 양자화 오차는 아날로그 신호를 디지털 신호로 바꾸거나, 계산의 결과로 늘어나는 자리수를 제한할 때에 생기게 된다.

2. 고정소수점 구현의 성능 평가

디지털 신호처리 시스템에서는 내부의 단어길이를 작게 할수록 양자화 영향에 의해 시스템의 성능이 나빠지므로, 최소 단어길이는 시스템의 요구 성능과도 밀접한 관계가 있다. 예를 들어, 동일한 주파수 특성을 가지는 디지털 필터라도 고충실도가 필요한 오디오 응용은 통신용보다 더 긴 단어길이를 요구할 수 있다. 그러므로 시스템의 요구성능을 평가하는 기준이 필요하고, 그 기준을 설계하는 시스템에 나타내야 한다.

예를 들어, 첫째로 선형 시불변 시스템을 비롯하여 많은 시스템에서는 신호대 양자화 잡음비(SQNR: Signal to Quantization Noise Ratio)를 고정소수점 구현의 평가 척도로 사용할 수 있다. 여기에서, 기준이 되는 신호는 부동소수점 연산방법으로 얻어진 결과를 사용하고, 평가되는 신호는 고정소수점 연산방법으로 얻은 결과이다. 이때, 양자화 잡음은 부동소수점 구현 결과와 고정소수점 구현 결과의 차이이다. 그림 2에 IIR 필터 구현시의 양자화 효과를 측정하기 위한 신호흐름도를 보인다. 여기서 IIR 필터 블록은 고정소수점 연산으로 계산하며 부동소수점 모의 실험의 결과(FLOUT)는 미리 계산하여 저장한 후에 사용한다.

두번째로 비선형이나 시변 시스템 또는 기타 복잡

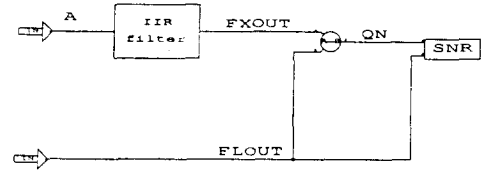


그림 2. IIR 필터의 양자화 효과를 측정하는 신호 흐름도

Fig. 2. A signal flow block diagram for measuring the quantization effect of an IIR filter.

한 시스템의 경우에는 양자화 잡음의 효과를 단순히 신호대 잡음비로 평가할 수 없다. 예를 들어 적응 필터의 경우 부동소수점으로 계산한 error 신호와 고정소수점으로 계산한 error 신호의 차이는 큰 의미가 없다. 이 경우 error를 최소화 하도록 빨리 수렴시키는 것이 목적이므로 평가 기준을 수렴 속도, 또는 수렴후 error의 파워 등으로 정하는 것이 적당하다. 그림 3에 수렴한 후의 error의 파워(mean squared error)로 적응 필터의 양자화 효과를 측정하기 위한 신호흐름도를 보인다. 그림에서 timer에 설정한 시간이 지나면 Z 신호는 ERR 신호의 제곱을 계속 누산한다. 그 값을 누산한 sample 수로 나누면 ERR 신호의 제곱의 평균, 즉 파워가 된다.

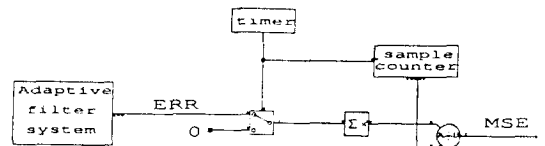


그림 3. 적응 필터의 양자화 효과를 측정하는 신호 흐름도

Fig. 3. A signal flow block diagram for measuring the quantization effect of an adaptive filter.

그 밖에 ADM이나 ADPCM 같은 coder-decoder 시스템에서는 원래 부호화 하기 전의 신호와 부호기와 복호기를 거쳐 재생시킨 신호와의 비교를 통해 그 차이를 최소화 하는 것을 평가 기준으로 삼을 수 있다.

3. 연산소자의 하드웨어 비용

단어길이를 줄이는 가장 큰 목적은 하드웨어 비용을 최소화 하는것이다. 디지털 신호처리 알고리즘을 전용 VLSI로 구현시 하드웨어 비용은 VLSI chip의

면적 또는 사용되는 게이트(gate)의 수라고 생각될 수 있다. 따라서, 가산기, 승산기, 또는 지연기 등의 하드웨어의 비용이 단어길이의 변화에 의해 어떻게 달라지는지 미리 알려져 있어야만 최적화 설계를 할 수 있다. 이 하드웨어의 비용은 ASIC회사의 cell-library나 반도체공정의 설계규칙등에 따라 각각 다르다. 일반적으로 가산기, 상수 승산기, 지연기 등은 단어길이에 비례하는 하드웨어를 필요로하며, 승산기는 곱해지는 두 수의 단어길이의 곱에 비례한 하드웨어가 소요된다고 모델할 수 있다. 필요한 하드웨어의 양을 계산하는 구체적인 예로 VLSI Technology 사의 cell library를 사용한 예를 표 1에 보인다.⁹⁾

표 1. VLSI Technology 사의 cell library를 이용할 때 필요한 하드웨어량

Table 1. A model of the hardware cost using the cell library of VLSI Technology Inc.

연산 소자	4 비트	8 비트	16 비트	모델
가산기(ripple carry)	74	161	333	20w
가산기(carry select)	139	309	633	38w
지연기(D flip-flop)	72	144	288	18w
상수 승산기 (계수에 0이 아닌비트가 4개)	192	415	-	50w
승산기 (x 4비트)	279	668	-	75w
(x 8비트)	668	1595	-	180w
(x n비트 추정값)				20nw

한편, 위에 모델한 하드웨어의 양은 대략적인 비례값으로서, 정확한 값은 구현하는 하드웨어의 구조(architecture)에 따라 많이 달라진다. 예를 들어, 신호의 sampling frequency가 상당히 낮아서 하나의 하드웨어 소자를 시분할하여서 구현할 때에는, 상대적으로 곱셈기나 가산기의 값이 싸진다. 그러나 지연 소자는 시분할 방식으로 구현할 수 없으므로 값이

줄지 않는다. 지연 소자로 외부의 RAM등을 사용할 경우에는 비트당 가격이 작다고 가정할 수 있다. 그리고 쉬프트(shifter)의 경우, 시분할 구조를 이용한 구조에서는 barrel-shifter를 필요로 하나, full-array 구현에서는 단지 connection만 바꾸면 되므로 일체의 추가 하드웨어 부담이 없다.

Ⅲ. VHDL을 이용한 신호흐름도의 표현 및 모의 실험

본 설계 환경에서는 신호처리 알고리즘을 그림 4에 보이는 것과 같은 블록 다이어그램 또는 신호흐름도(signal flow block diagram)를 사용하여 표현한다. 이와같이 신호흐름도를 이용하여 신호처리 알고리즘을 표현하는 방법은 현재 Gabriel⁴⁾이나 Ptolemy¹⁶⁾, SPW⁵⁾ 등에서 이용되고 있다. 본 설계 환경에서는 신호흐름도의 입력에는 Viewlogic사의 Workview¹¹⁾를 이용하고, 모의 실험은 Synopsys사의 VHDL simulator¹⁶⁾를 이용하여 수행하였다.

1. 신호흐름도의 VHDL 프로그램 변환

VHDL은 VLSI등 하드웨어의 모델과 시뮬레이션을 위하여 개발된 언어이므로 부동소수점부터 논리 회로 단위까지 광범위하고 자유로운 데이터형식을 지원하고 있다. 또한, 순차 계산은 물론 동시 계산(concurrent assignment)기능을 가지고 있으므로 신호처리 블록다이어그램을 별도의 스케줄링 없이 모의 실험하는 것이 가능하다. 그리고, VHDL 프로그램은 현재 상용화되어 있는 많은 논리합성 소프트웨어들을 이용하여 ASIC으로 빠르게 구현할 수 있다.

¹⁶⁾[9][11]

VHDL은 행위(behavior)와 구조(structure) 기

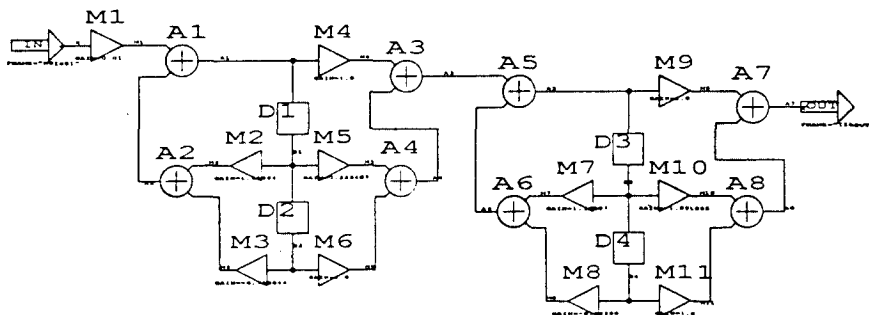


그림 4. 4차 IIR 필터의 신호 흐름도

Fig. 4. A signal flow block diagram of a fourth order IIR filter.

술(記述) 능력, 그리고 그들의 혼합된 사용을 지원한다. 행위 기술은 하드웨어의 동작 행위 또는 기능을 묘사하는 것이고, 구조 기술이란 그러한 동작을 이루게 하는 각 부품(instance)들의 연결 상태를 나타내는 것이다. 본 개발 환경에서 가산기나 승산기, 지연기 같은 디지털 신호처리의 가장 기본 부품들은 행위 기술 방법으로 표현된다. 예를 들면, 그림 4에 이용되는 가산기의 부동 소수점 모델은 그림 5-(a), 그리고 고정 소수점 모델은 그림 5-(b)와 같다. VHDL 시뮬레이터에 따라서는 내부 변수들을 초기화 해 주는 code가 추가로 필요할 수도 있다.

```
entity ADD is
  Port (
    A : In    real;
    B : In    real;
    Y : Out   real );
end ADD;
architecture behav of ADD is
  begin
    process(a,b)
    begin
      y <= a + b;
    end process;
  end behav;
```

(a)

```
entity ADD is
  Port (
    A : In    signed(7 downto 0);
    B : In    signed(7 downto 0);
    Y : Out   signed(7 downto 0));
end ADD;
architecture behav of ADD is
  begin
    process(a,b)
    begin
      y <= a + b;
    end process;
  end behav;
```

(b)

그림 5. 가산기의 VHDL 모델 (a) 부동소수점 가산기 모델 (b) 고정소수점 가산기 모델

Fig. 5. VHDL models of adders (a) A floating point adder model, (b) A fixed point adder model.

2. 범위 추정 및 의사 고정 소수점 VHDL 모델
고정소수점 변환을 하기 위해서는 각 신호의 정수부 단어길이와 소수부 단어길이를 결정해야 한다. 정수부 단어길이는 신호의 범위에 의해 식 1과 같이 결정되는데, 이 범위는 해당 신호의 평균과 표준편차에 의해 식 2와 같이 정해진다. 평균과 표준편차의 값들은 신호의 합과 제곱합을 구하는 소자의 VHDL 모델을 시스템의 내부 신호들에 연결하여 구한다. 즉

신호흐름도를 VHDL 프로그램으로 변환하면서 각 신호에 평균과 표준편차를 구할 수 있는 소자를 추가로 연결하여 모의 실험으로 구한다.

소수부 단어길이를 결정하기 위해서는 소수부 단어길이의 변화에 의한 양자화 효과를 알아내야 한다. 양자화 효과를 모델하기 위해 의사 고정 소수점 VHDL 모델을 사용하였다. 이 모델에서 각 소자(instance) 간의 신호 전달은 부동소수점 데이터 형식으로 이루어진다. 그러므로 첫번째 단계에서 사용한 부동소수점 code의 interface를 그대로 사용할 수 있다. 실제로 고정소수점으로 양자화 되는 것은 소자(instance) 사이에 주고 받는 신호들이지만 그 값을 계산해내는 소자(component)의 모델에 양자화 모델을 추가 함으로써 원래의 부동소수점 code를 최소한으로 변경하여 고정소수점 모의실험이 가능하다. 이 양자화기 모델은 고정소수점 수로 표현할 수 있는 범위를 넘어가는 수에 대한 처리, 예를 들어 saturation이나 overflow등을 처리 하는 code도 포함하고 있다. 이 모델에서 입출력 신호는 부동소수점으로 나타내고 모든 연산은 부동소수점으로 수행되지만, 그 연산 결과는 고정소수점 연산의 결과와 거의 같다. 의사 고정소수점 VHDL 모델을 사용한 승산기의 예를 그림 6에 보인다.

```
entity MUL is
  Generic( limit : real := 0.0;
           scale : real := 0.0);
  Port (
    A : In    real;
    B : In    real;
    Y : Out   real );
end MUL;
architecture behav of MUL is
  begin
    process(a,b)
    variable yy : real;
    variable ytap : integer;
    begin
      yy := a * b;
      if limit /= 0.0 then
        if scale /= 0.0 then
          yy := yy * scale;
          ytap := integer(yy);
          yy := real(ytap); -- round off the scaled value
          yy := yy / scale;
        end if;
        assert ( yy < limit and yy > -limit )
          report "signal saturated" -- message display
          severity warning;
        if yy > limit then
          yy := limit; -- saturate the value
        elsif yy < -limit then
          yy := -limit; -- saturate the value
        end if;
      end if;
      y <= yy;
    end process;
  end behav;
```

그림 6. 의사고정소수점 VHDL 모델로 나타낸 승산기

Fig. 6. A multiplier model using pseudo fixed point VHDL.

IV. 부동소수점 알고리즘의 고정소수점 변환

1. 신호의 분할(grouping)

신호흐름도를 분석하여 내부의 신호들을 같은 단어

길이를 가지는 몇 개의 그룹으로 나누면 최적화 시켜야 하는 변수들의 수가 줄어들므로 탐색 시간을 크게 줄일 수 있다. 다음과 같은 방법에 의하여 신호들을 그룹으로 나눈다.

첫째, 신호흐름도에서 지연기의 입력 신호와 출력 신호는 같은 단어길이를 가지므로 이들을 같은 그룹으로 묶는다. 둘째, 가산기나 감산기의 입력 신호와 출력 신호도 같은 소수부 단어길이를 가지도록 같은 그룹으로 묶는다. 지연기의 입력과 출력은 시간 지연만 있을 뿐 사실은 같은 신호이고, 가산기는 두 입력 신호의 양자화 레벨이 같아야 계산이 가능하고 출력 신호도 같은 양자화 레벨을 가지므로 이들을 하나의 그룹으로 묶는다. 셋째, 곱셈기나 양자화기에 의해 연결되는 신호들은 서로 다른 단어길이를 가질 수 있으므로, 특별히 첫번째와 두번째 조건을 만족시키지 않으면 서로 다른 그룹에 속하도록 하였다. 네째로, 어느 한 신호와 첫번째의 조건에 의해 동일 그룹에 속한 신호들이 곱해져서 생긴 신호들은 동일 그룹으로 묶는다. 단, 네번째 방법은 선택적으로 적용할 수 있도록 하였다. 그림 4의 4 차 IIR 필터의 예에서 신호들을 그룹으로 나눈 결과를 표 2에 보인다.

표 2. 그림 4의 4 차 IIR 필터의 신호를 그룹으로 나눈 결과

Table 2. The signal grouping result of the IIR filter in figure 4.

그룹	신호
1	X
2	M1, A1, A2, M2, M3, D1, D2
3	M4, M5, M6, A3, A4, A5, A6, M7, M8, D3, D4
4	M9, M10, M11, A7, A8

2. 정수부 단어길이 및 소수부 단어길이의 결정

각 신호들의 정수부 단어길이는 그 신호들의 범위 (range)와 식 1을 이용하여 구할 수 있다. 이때, 그 신호들의 범위는 제 III장에서 설명한 범위 계산 VHDL 소자에서 구해진 범위를 사용한다. 각 신호는 모두 서로 독립된 정수부 단어길이를 가질수 있으나, 이 경우 시분할 구현중에 불편하므로 각 그룹내에 속한 신호는 모두 동일한 정수부 단어길이를 가지도록 하였다. 따라서, 어느 그룹의 정수부 단어길이는 그 그룹에 속한 신호중 최대의 범위를 가지는 것에 의하여 식 1에 따라 결정된다. ^{[7][8]} 그림 4의 4 차 IIR 필터의 그룹별 최대 범위를 가지는 신호들의 통계 결과와 범위, 이를 이용해서 결정한 정수부 단어 길이가 표 3에 나타나 있다.

표 3. 신호의 통계 결과, 그룹의 범위 및 정수부 단어길이

Table 3. The statistics of signals, the ranges and the integer word-lengths of groups.

signal	mean	variance	range	정수부단어길이
X	3.855312e-02	3.404529e-01	2.372	2
M2	7.580257e-03	2.115712e-03	0.192	-2
M7	4.913862e-02	1.874743e-01	1.781	1
M10	-3.395561e-02	8.951978e-02	1.163	1

다음은 신호들의 소수부 단어길이는 정하는 단계이다. 각 그룹별 단어길이를 나열한 것을 단어길이 벡터 w 라 하자

$$w = (w_1, w_2, \dots, w_N) \tag{3}$$

단, N 은 그룹의 수

이때 시스템의 성능을 $p(w)$, 시스템 구현의 비용을 $c(w)$ 라 하면, 원하는 최적의 단어길이를 찾는 문제는 $p(w) \geq p_{desired}$ 를 만족시키는 w 중에서 $c(w)$ 를 최소화하는 w 를 찾는 문제가 된다. 이때 단어 길이가 짧아질수록 시스템의 성능 $p(w)$ 가 줄어든다는 가정아래 다음과 같은 두 단계의 방법을 사용한다.

첫째, 각 그룹별 최소 소수부 단어길이, $w_{f,min,i}$ 를 결정한다. 이때 그룹 i 의 최소 소수부 단어 길이로는, 그 그룹의 신호만 고정소수점 단어길이, $w_{f,min,i}$ 를 가지도록 하고, 나머지 그룹은 모두 부동소수점으로 두고 모의실험을 하여 주어진 규격을 만족하는 것이다. 이때, 최소 소수부 단어길이는 그 그룹이 가질 수 있는 최적 소수부 단어길이보다 항상 짧거나 같은 값을 가진다. 왜냐하면 다른 그룹은 모두 부동소수점으로 모의 실험했기 때문에 그 그룹의 양자화 오차만이 전체 시스템의 성능에 영향을 주기 때문이다. 그림 4의 IIR 필터의 예에서는 네 개의 그룹으로 나뉘므로 총 네 개의 최소 소수부 단어길이를 결정하여야 한다. 그림 4의 IIR 필터 예에서 찾아낸 그룹별 최소 단어길이 및 그때의 신호대 잡음비는 표 4와 같다.

표 4. 그룹별 최소 단어길이 및 그때의 신호대 잡음비

Table 4. The word-length of each group and the signal to noise ratio.

그룹	소수부 단어길이	정수부 단어길이	단어길이	신호대 잡음비
1	6	2	9	12.37
2	14	-2	13	15.21
3	10	1	12	40.27
4	8	1	10	40.59

둘째, 각 그룹들의 최소 단어길이를 가지고 전체 시스템을 모의실험한다. 그 결과가 원하는 사양을 만족하게 될 경우, 각 그룹의 최소 단어길이들이 바로 최적 단어길이이다. 그렇지 못한 경우는 각 그룹의 단어길이를 하나씩 증가 시키면서 모의실험으로 시스템의 성능을 만족 시키는 최적 단어길이를 찾는다. 이때, 어느 그룹의 단어 길이를 먼저 증가시키는 가 하는 문제는 다음의 두가지 방법중 하나를 사용한다.

i) 하드웨어의 양, 즉 $c(\mathbf{w})$,가 가장 적은 순서대로 찾는다. 즉 최소 단어길이 벡터 \mathbf{w}_{min} 부터 $c(\mathbf{w})$ 를 미리 계산, 그 값이 작은 순서대로 모의실험을 하여 $p(\mathbf{w})$ 를 구해 $p_{desired}$ 보다 커지는 \mathbf{w} 값을 찾으면 된다. 이 방법의 장점은 최적의 단어길이 찾는다. 그런데 이 방법은 최적의 단어길이를 찾기 위해 너무 많은 모의실험이 필요할 수 있다는 단점이 있다. 그림 4의 IIR 필터의 예에서 이 방법을 사용한 결과가 표 5-(a)에 있으며, 총 7 번의 모의실험이 필요하였다.

표 5. 4차 IIR 필터의 최적 단어길이 찾기

Table 5. The search of optimal word-length in the fourth order IIR filter.

(a) 최소 하드웨어 비용 순으로 찾기

(a) The search in the sequence of the minimum cost

\mathbf{w}	$c(\mathbf{w})$	$s(\mathbf{w}), \text{dB}$
(9, 13, 12, 10)	8230	36.42320
(9, 13, 12, 11)	8270	37.75603
(10, 13, 12, 10)	8280	36.56599
(9, 13, 12, 12)	8310	38.81416
(10, 13, 12, 11)	8320	38.44302
(11, 13, 12, 10)	8330	36.75763
(9, 13, 12, 13)	8350	38.83337
(10, 13, 12, 12)	8360	40.22198

(b) 성능증분/비용증분이 큰 순서로 찾기

(b) The search in the sequence of the performance increment per cost increment

\mathbf{w}	$c(\mathbf{w})$	$p(\mathbf{w})$	$\Delta p(\mathbf{w})/\Delta c(\mathbf{w})$
(9, 13, 12, 10)	8230	36.42320	
(10, 13, 12, 10)	8280	36.56599	2.855760e-03
(9, 14, 12, 10)	8506	36.56567	5.162101e-04
(9, 13, 13, 10)	8546	37.12624	2.224801e-03
(9, 13, 12, 11)	8270	37.75603	3.332070e-02
(10, 13, 12, 11)	8320	38.44302	2.244241e-02
(9, 14, 12, 11)	8546	38.35761	6.121547e-03
(9, 13, 13, 11)	8586	38.37303	5.477062e-03
(9, 13, 12, 12)	8310	38.81416	2.988704e-02
(10, 13, 12, 12)	8360	40.22198	2.922136e-02

< 선택

< 선택

ii) 하드웨어 비용의 증가($\Delta c(\mathbf{w})$)에 비해 시스템 수준의 성능향상($\Delta p(\mathbf{w})$)이 큰 순서대로 찾는다. 즉 각 그룹에 대하여 한 비트 씩 단어길이를 증가 시킨 경우에 대하여 모의실험을 하여 $p(\mathbf{w})$ 값을 얻은 다음 그중에서 $\Delta p(\mathbf{w})/\Delta c(\mathbf{w})$ 가 가장 큰 경우를 선택한다. 선택한 \mathbf{w} 에 대하여 위의 과정을 $p(\mathbf{w})$ 가 $p_{desired}$ 보다 커질 때까지 반복한다. 그런데 선택에서 탈락한 것들 중에 최적값을 찾기 위한 중간 단계가 있을 수도 있으므로 branch and bound 방법을 첨가한다. 여기서 사용한 branch and bound 방법은 어떤 단계에서 $\Delta p(\mathbf{w})/\Delta c(\mathbf{w})$ 값이 전 단계에서 선택에 탈락한 값보다 모두 작은 값이 나오면 전 단계로 되돌아가 차선의 방향으로 분기하여 계속 찾아가는 것이다. 예로 들은 4 차 IIR 필터의 경우 이 방법을 사용하여 최적 단어길이를 찾는 과정은 표 5-(b)와 같이 되는데, 총 9 번의 모의 실험을 필요로 하였다.

3. 기존 연구와의 비교

디지털 신호처리 시스템의 고정소수점 하드웨어 구현시 유한 단어길이 효과를 분석하기 위해 신호흐름도 상에서 입출력과 임의의 노드 사이의 전달함수 (transfer function)를 이용하는 방법이 널리 사용되어 왔다. 입력 노드로부터 어떤 노드까지의 전달함수를 사용하여 그 노드 신호값의 범위(정수부 단어길이)를 구하고 그 노드로 부터 출력까지의 전달함수를 사용하여 그 노드 신호가 가져야 할 정밀도(소수부 단어길이)를 구하는 방법이다. 이 방법을 이용하여 정수부 단어길이와 소수부 단어길이를 정하는 과정은 다음과 같다.

신호흐름도 상의 어떤 노드의 신호 범위는 입력 신호의 범위에 전달함수의 주파수 특성상의 최대 증폭도를 곱한 값보다 작다. 이때 전달함수의 주파수 특성상의 최대 증폭도가 그 전달함수의 L_∞ norm이 된다. ¹¹ 그림 4의 필터에서는 입력으로부터 그룹 1의 신호까지의 전달함수의 L_∞ norm이 1, 그룹 2까지는 0.144, 그룹 3까지는 1.77, 그룹 4는 1 이다. 한편 입력 신호의 범위가 (-1,1)이므로 정수부 단어길이는 (0,-2,1,0)이 된다.

신호처리 블록 다이어그램 내의 각 노드에서 발생되는 양자화 오차는 서로 상관 관계가 없다고 가정할 수 있으므로 출력에 나타나는 양자화 잡음의 파워는 각 노드에서 발생하는 양자화 잡음 파워들의 가중치 합으로 나타 낼 수 있다. 그 가중치 값은 그 노드에서 출력에 이르는 전달함수에 의해 증폭되는 증폭비의 제곱이다. 그 증폭비는 주파수 특성상의 최대 증

폭도(L norm)의 상한값을 가진다. 제안한 방법과의 비교를 위해 그림 4의 필터에서는 한 그룹에 속한 신호들은 같은 소수부 단어길이를 가진다고 가정하였다. 양자화 잡음은 승산기 출력에서만 발생하는 것으로 생각하면 그룹 1은 1 곳, 그룹 2는 3 곳, 그룹 3은 5 곳, 그룹 4는 1곳에서 양자화 잡음이 발생함을 알 수 있다. 그리고 양자화 잡음이 발생하는 곳으로부터 출력까지의 전달함수의 최대 증폭도가 첫번째 그룹은 1, 두번째 그룹은 99.6, 세번째 그룹은 12.7, 네번째 그룹은 1이 되므로 각 단에서 발생하는 양자화 잡음의 파워를 p_i 라고 하면 출력이 나타나는 그 값들의 합은 다음과 같다.

$$p = 3p_4 + 161.3 * 5p_3 + 9920 * 3p_2 + p_1 \quad (4)$$

여기서 p_i 는 각 그룹의 소수부 단어길이를 n_i 라 할때 $(2^{n_i})^2/12$ 로 나타내어진다. 한편 출력의 신호 파워는 모의실험 결과 $4.43e-2$ 가 나왔으므로 신호대 잡음비가 40dB가 되기 위해 잡음의 파워 p 는 $4.43e-6$ 가 되어야 한다. 식 4와 하드웨어 비용 모델을 풀면 최적 단어길이가 계산될 수 있는데 이 문제는 비선형 최적화 문제가 된다.^[12] 출력의 양자화 잡음 중 가장 큰 비중을 차지하는 그룹 2의 소수부 단어길이를 계산하면 그 결과는 14 비트가 나오는데 모의실험으로 구한 값과 일치한다.

결론적으로 전달함수의 최대 증폭도(L_∞ norm)을 이용해서 해석적으로 얻은 단어길이를 모의실험을 사용하여 얻은 단어길이는 거의 비슷하다. 그러나 해석적으로 푸는 방법은 신호처리 블록 다이어그램 상의 모든 노드에 대해 입력으로부터 그 노드까지의 전달함수, 그 노드로부터 출력까지의 전달함수를 구해야 하는 어려움이 있을 뿐만 아니라 이 방법은 디지털 필터와 같은 선형 시불변 시스템에만 적용할 수 있다는 단점이 있다. 그런 반면에, 본 제시된 방법은 비선형과 시변 시스템 등 모두에 적용할 수 있다.

V. 구현 예

LMS 적응 필터를 사용하여 그림 7과 같은 채널 식별 시스템을 설계하였다. 모의실험에 사용한 입력 $u[n]$ 은 -1 과 +1 사이에서, $v[n]$ 은 -0.03162 와 +0.03162 사이에서 균일한 분포를 가지는 난수를 이용하였다. 이때 입력신호의 파워는 1/3이 되고 채널의 출력에 더해지는 노이즈의 파워는 1/3000이 된다. 사용한 채널은 12차 FIR 필터를 사용한 저역 통과 필터이다. 여기에서 적응 필터의 설계 사양으로 택한

기준은 필터가 수렴한 뒤의 error신호의 파워가 1/1000이하가 되는 것이다. 파워의 계산은 200 샘플 부터 500 샘플 까지의 time average를 사용하였다. 여기서 $p(w)$ 는 수렴한 이후의 error 파워로 하였다. 성능 평가를 위해 그림 3에서 설명한 성능 평가 블록을 추가하여 모의 실험하였다. 사용한 12차 적응 필터의 블록 다이어그램은 그림 8와 같다.

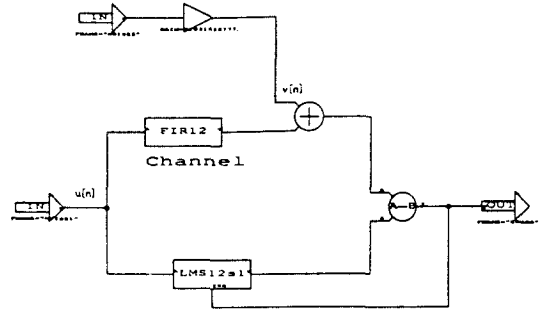


그림 7. 채널 식별 시스템

Fig. 7. A channel identification system.

첫째로 블록 다이어그램내의 모든 신호를 그룹으로 분리하는 단계이다. 우선 제 4 장에서 설명한 첫번째와 두번째, 세번째 규칙에 의해서 입력 신호와 입력이 지연된 신호들이 첫번째 그룹으로 나뉘고, 출력과 출력을 계산하기 위해 더해지는 신호와 그 결과인 error신호들이 두번째 그룹, 그리고 error에 상수(step size)를 곱한 신호가 세번째 그룹으로 나누어진다. 마지막으로, 네번째 규칙에 의해서 필터의 각 계수 값들을 나타내는 신호들이 모두 네번째 그룹으로 묶였다.

표 6. 12차 LMS 적응필터 내부 신호를 그룹으로 나눈 결과

Table 6. The signal grouping result of the 12th LMS adaptive filter.

그룹	신호
1	X0, X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11
2	S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, Y, P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11
3	H0, H1, H2, H3, H4, H5, H6, H7, H8, H9, H10, H11, B0, B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11, C0, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11
4	SERR

두번째 단계로 각 신호들의 정수부 단어길이를 구한다. 각 그룹의 범위와 결정된 정수부 단어길이를 표 7에 보인다. 평균과 분산으로 추정된 범위가 실제 가능한 최대값을 포함하는지를 의사 고정소수점

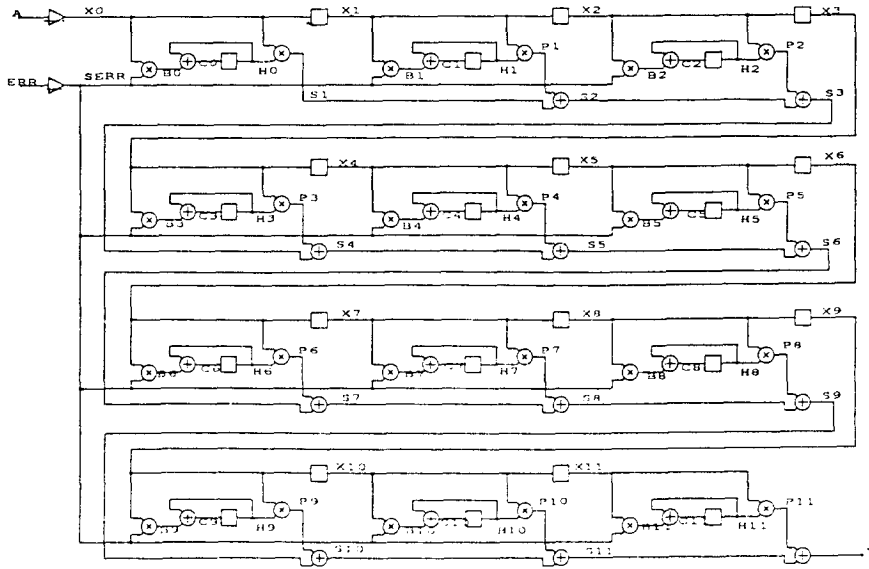


그림 8. 12차 LMS 적응 필터
Fig. 8. A twelfth order LMS adaptive filter.

VHDL 모델의 overflow 검사 기능을 이용한 모의 실험을 통해 검증을 하였다.

표 7. 신호의 통계 결과와 결정된 그룹의 범위
Table 7. The statistics of the signal and the determined ranges of the groups.

그룹	최대 범위 불 가지는 신호	평균	분산	범위	정수부 단어길이
1	X0	1.754651e-02	3.251311e-01	2.298	2
2	S10	3.180678e-02	1.105362e-01	1.361	1
3	H6	3.107040e-01	5.856484e-03	0.617	0
4	SERR	7.149953e-04	9.049274e-05	0.0373	4

세번째로 소수부 단어길이를 결정하는 단계이다. 우선 각 그룹별로 최소 단어길이를 구한다. 각 그룹별로 의사 고정소수점 VHDL 프로그램을 이용해서 모의 실험한 결과가 표 8에 나와 있다.

표 8. 그룹별 최소 단어길이 및 그때의 error 파워
Table 8. The minimum word-length of each group and the error power.

그룹	단어길이	error 파워
1	6	0.9587e-03
2	8	0.9601e-03
3	7	0.9794e-03
4	4	0.6715e-03

하드웨어 비용은 표 1의 결과를 사용하여 지연기는 18w, 가산기는 20w, 승산기는 20w₁w₂로 계산하였다. 적응 필터의 각단은 모두 같은 회로를 가지므로, 한 단의 구현에 필요한 하드웨어의 양만을 고려하면 식 5의 결과를 얻는다.

$$c(w) = 18 * w_1 + 18 * w_2 + 20 * w_3 + 20 * w_4 * w_1 + 20 * w_1 * w_2 + 20 * w_1 * w_3 + 20 * w_1 * w_4 + 20 * w_2 * w_3 + 20 * w_2 * w_4 + 20 * w_3 * w_4 \quad (5)$$

최소 단어길이 (6,8,7,4)부터 시작하여 최적 단어 길이를 찾기 위한 모의실험을 반복하였다. 최소 하드웨어 비용 순서 대로 모의 실험을 반복하면 27 번째에 원하는 성능을 가지는 단어길이 (7,9,7,4)를 찾아 낼수 있다. 그때의 하드웨어 비용 c(w)는 2112 이고 계산한 파워는 0.907445e-3이 되었다. 성능 증분 대 비용 증분의 증가가 큰 것 부터 선택하여 모의 실험을 반복하면 5 번째에 원하는 성능을 가지는 단어길이 (7,9,7,4)를 찾아낼 수 있다. 이 LMS 적응 필터의 예에서는 두가지 방법에 의한 결과가 같은데 일반적으로는 첫번째 방법에 의한 결과가 하드웨어 비용이 더 적게 든다.

VI. 결론

신호 흐름도를 부동소수점으로 모의실험하고, 또 소수점의 위치와 최적의 단어길이를 구한 후, 이를 이용하여 고정소수점 VHDL 프로그램을 생성하는

소프트웨어가 개발되었다. 이때, 최적 단어길이는 신호대 잡음비 또는 찌그러짐의 양 등 시스템 수준의 성능 사양과 승산기와 가산기등의 하드웨어 모듈의 비용등을 토대로 최소의 비용으로 주어진 성능을 만족시키도록 정해진다. VHDL언어가 부동소수점 모의실험을 비롯하여 모든 개발 과정에 사용되었는데, 이는 신호흐름도를 별도의 스케줄링없이 모의실험하는 것을 가능케하고, 또 부동소수점과 고정소수점등 융통성있는 신호의 표현을 쉽게 지원하고, 또 그 결과를 논리합성 소프트웨어에 연결해서 바로 하드웨어로 구현할 수 있기 때문이다. 본 연구에서는 어떤 단어길이를 가졌을 때의 시스템의 성능을 모의실험(simulation)으로 구하기 때문에 선형 시스템은 물론, 비선형 및 시변 시스템등에 제한없이 적용할 수 있다. 그러나, 모의실험에는 시간이 많이 걸리기 때문에 최적 단어길이를 가능한한 최소의 모의실험으로 구하는 방법이 연구되었다. 우선, 신호흐름도를 분석하여 양자화 레벨이 같은 신호들을 미리 그룹으로 묶어 최적화 시켜야할 신호의 수를 줄인다. 그리고 최적 단어길이를 찾는 탐색 시간을 감소시키기 위해 각 그룹별 최소 단어길이를 우선 결정하여 최적 해를 찾기 위한 탐색공간을 줄이는 방법을 연구하였다. 탐색하는 방법으로는, 시간은 더 걸리나 최적 값을 찾을 수 있는 최소 하드웨어 비용 탐색과, 비록 빠르지만 근사값을 찾게 되는 최대 성능 대 비용 탐색의 두 가지 방법을 연구하였다. 본 방법은 저자들의 협력으로 미국 Alte Group(옛날 이름 Comdisco Systems)에서 Fixed-Point Optimizer라는 소프트웨어로 상품화 되었다.

參 考 文 獻

[1] Leland B. Jackson, "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters," *The Bell System Technical Journal*, Feb. 1970.

[2] Comdisco Systems, Inc., *SPW-The DSP Framework Hardware Design System User's Guide*, Jan. 1993.

[3] Roger Lipsett, Carl F. Schaefer, Cary Ussery, *VHDL: Hardware Description and Design*, Intermetrics, Inc., 1989.

[4] Edward A. Lee, Wai-Hung Ho, Edwin E. Goei, Jeffrey C. Bier, and Shuvra Bhattacharyya, "Gabriel: A Design Environment for DSP," *IEEE Trans. on*

ASSP, Vol. 37, No. 11, Nov. 1989.

[5] Comdisco Systems, Inc., *Signal Processing WorkSystem Users' Guide*, Aug. 1990.

[6] Synopsys Inc., *VHDL Compiler Reference Manual*, V2.2., 1991.

[7] Wonyong Sung, "An Automatic Scaling Method for the Programming of Fixed-Point Digital Signal Processors," *IEEE International Symposium on Circuits and Systems '91*, pp. 37-40, Singapore, Jun. 1991.

[8] 김 시현, 성 원용 "자동 스케일링 기능이 지원되는 고정 소수점 디지털 시그날 프로세서 개발 시스템," *대한전자공학회지* 29권 A편 3호, pp. 261-270, 1992년 3월

[9] VLSI Technology, Inc., *VGT350 Portable Library*, 1990.

[10] J. Buck, S. Ha, E. A. Lee, and D. G. Messerschmitt, "Ptolemy: A Mixed-Paradigm Simulation/Prototyping Platform in C++", *Proc. C++ At Work Conference*, Santa Clara, CA, Nov. 1991.

[11] Viewlogic Systems Inc., *Workview Manual*, 1989.

[12] 성 원용, 금 기일, 김 시현, "디지털 신호처리 회로도의 고정소수점 구현법," 제 5회 신호처리합동학술대회 논문집, 중앙대학교, 1992년 9월

[13] C. Bernard Shung et al, "An Integrated CAD System for Algorithm-Specific IC Design," *IEEE Trans. on CAD*, Vol. 10, No. 4, Apr. 1991.

[14] Steven E. Reyer, James A. Heinen, "Quantization Noise Analysis of a General Digital Filter," *IEEE Trans. on ASSP*, Vol. 29, No 4, Aug. 1981.

[15] Hon Keung Kwan, "Amplitude Scaling of Arbitrary Linear Digital Networks," *IEEE Trans. on ASSP*, Vol. 32, No. 6, Dec. 1984.

[16] I. D. Yun and S. U. Lee, "On the Fixed-Point-Error Analysis of Several Fast DCT Algorithms," *IEEE Trans. on Circuits and Systems for Video*

- Technology*, Vol. 3, No. 1, Feb. 1993.
- [17] Hugo De Man, Francky Catthoor, Gert Coossens, Jan Vanhoof, Jef Van Meerbergen, Stefaan Note, and Jos Huisken, "Architecture-Driven Synthesis Techniques for VLSI Implementation of DSP Algorithms," *Proc. of IEEE*, Vol. 78, No. 2, Feb. 1990.
- [18] B. Liu, "Effect of Finite Word Length on the Accuracy of Digital Filters : A Review," *IEEE Trans. Circuit Theory*, Vol. CT-18, pp. 670-677, Nov. 1971.
- [19] Alan V. Oppenheim and Ronald W. Scafer, *Discrete-Time Signal Processing*, Prentice-Hall, Inc., 1989.
- [20] Lawrence R. Rabiner and Bernard Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, Inc., 1975.
- [21] Ki-Il Kum and Wonyong Sung, "VHDL based Fixed-point Digital Signal Processing Algorithm Development Software," *Proc. of International Conference on VLSI and CAD '93*, Taejeon Korea, pp. 257-260, Nov. 1993.
- [22] Wonyong Sung and Ki-Il Kum, "Word-length Determination and Scaling Software for a Signal Flow Block Diagram," *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) '94*, Adelaide Australia, Vol. 2, pp. 457-460, Apr. 1994.

 著 者 紹 介



琴基一(正會員)

1969年 2月 20日生. 1991年 2月 서울대 제어계측공학과 졸업(공학사). 1994年 2月 서울대 대학원 제어계측공학과 졸업(공학석사). 1994年 3月 ~ 현재 서울대 대학원 제어계측공학과 박사과정 재학

중. 주관심 분야는 디지털신호처리 및 컴퓨터이용설계 등임.

成元鎔(正會員)

1955年 4月 14日生. 1978年 2月 서울대학교 전자공학과 졸업(공학사). 1980年 2月 한국과학원 전기 및 전자공학과 졸업(공학석사). 1987年 7月 미국 University of California, Santa Barbara 전기 및 컴퓨터공학과 졸업(공학박사). 1980年 2月 ~ 1983年 7月 (주) 금성사 중앙연구소 연구원. 1989年 2月 ~ 현재 서울대학교 제어계측공학과 부교수. 주관심 분야는 병렬처리 컴퓨터와 VLSI를 이용한 고속 신호처리 등임.