

모바일 그래픽 응용을 위한 파이프라인 구조 특수 목적 연산회로의 하드웨어 설계

최병윤*

Hardware Design of Pipelined Special Function Arithmetic Unit for Mobile Graphics Application

Byeong-yoon Choi*

Department of Computer Engineering, Dongeui University, Busan 614-714, Korea

요 약

3차원 그래픽 API인 OpenGL과 Direct3D를 효율적으로 처리하기 위해 sine, cosine, 역수, 역제곱근, 지수 및 로그 연산을 처리하는 부동소수점 연산회로를 설계하였다. 고속 연산과 2 ulp 보다 작은 오차를 만족시키기 위해 2차 최대최소 근사 방식과 테이블 룩업 방식을 사용하였다. 설계된 회로는 65nm CMOS 표준 셀 조건에서 2.3-ns의 최대 지연시간을 갖고 있으며, 약 23,300 게이트로 구성된다. 최대 400 MFLOPS의 연산 성능과 높은 정밀도로, 설계한 연산회로는 3차원 모바일 그래픽 분야에 효율적으로 적용 가능하다.

ABSTRACT

To efficiently execute 3D graphic APIs, such as OpenGL and Direct3D, special purpose arithmetic unit(SFU) which supports floating-point sine, cosine, reciprocal, inverse square root, base-two exponential, and logarithmic operations is designed. The SFU uses second order minimax approximation method and lookup table method to satisfy both error less than 2 ulp(unit in the last place) and high speed operation. The designed circuit has about 2.3-ns delay time under 65nm CMOS standard cell library and consists of about 23,300 gates. Due to its maximum performance of 400 MFLOPS and high accuracy, it can be efficiently applicable to mobile 3D graphics application.

키워드 : 최대최소 알고리즘, 부동 소수점, 연산회로, 3차원 그래픽 SoC, 멀티쓰레딩

Key word : minimax algorithm, floating-point, arithmetic unit, 3-dimensional graphics SoC, multithreading

접수일자 : 2013. 06. 18 심사완료일자 : 2013. 07. 09 게재확정일자 : 2013. 07. 18

* **Corresponding Author** Byeong-Yoon Choi (E-mail:bychoi@deu.ac.kr, Tel:+82-51-890-1706)

Department of Computer Engineering, Dongeui University, Busan 614-714, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2013.17.8.1891>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

최근 스마트폰과 태블릿 PC 등 모바일 디바이스의 급격한 증가와 보편화로 휴대용 단말기용 실감나는 3차원 게임과 엔터테인먼트 소프트웨어, 컴퓨터 비전 분야의 수요가 급격히 증가하고 있다. NVIDIA와 AMD사에 의해 주도되고 있는 그래픽 프로세서는 최근 특화된 그래픽 분야를 뛰어 넘어 범용 분야에 적용할 수 있는 구조를 갖춘 GPGPU 구조를 발표하고 있다. 최신 그래픽 프로세서는 내부적으로 SIMD 구조, 멀티쓰레드 구조, VLIW 구조 등의 명령어 및 데이터 수준 병렬 처리 기술이 적용되고 있다[1-2]. 3차원 그래픽 프로세서는 기존 범용 응용 프로세서와 달리 삼각함수, 지수와 로그, 역수 값, 역 제곱근 값에 대한 높은 연산 빈도 특성으로, 이를 하드웨어로 고속 구현하는 것이 필수적이다. 그래픽 프로세서에 필요한 특수목적 연산은 일반적으로 룩업 테이블 방식, 테일러 근사방법, 다항식 근사 기법, 반복연산에 의한 수렴 방법 등으로 구현된다[3-6]. 면적 측면과 정밀도에서는 기본 연산의 반복 작업을 활용한 수렴 연산 방법이 효율적이지만, 그래픽 분야는 범용 응용 분야와 달리 초월 함수의 사용 빈도가 높아 정확성이 다소 떨어지더라도 연산 시간이 짧은 방식이 필요하다. 따라서 본 연구에서는 그래픽 분야의 속도와 정밀도 특성을 반영하여 최대 오차를 최소화하는 최대최소(minimax) 다항식 근사와 룩업 테이블 방식을 결합하는 방식을 채택하였다. 본 논문에서는 고성능 모바일 그래픽 분야에 적용을 위해 높은 연산 성능과 정밀도를 갖춘 파이프라인 구조의 특수목적용 연산회로를 회로를 설계하고 성능을 분석하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 부동 소수점 특수 목적 연산회로(FP-SFU)의 설계 사양을 살펴보고, 3 장에서는 초월 함수의 2차 최대최소 근사 방법을 기술하고, 4 장에서는 FP-SFU의 하드웨어 설계를 기술하였으며, 5 장에서는 설계한 회로에 대한 검증과 성능 분석을 하였으며 마지막으로 결론을 기술하였다.

II. 설계 사양

특수 목적 연산장치는 멀티쓰레드 구조의 모바일 3차원 그래픽프로세서에 내장을 목표로 하고 있어서,

다음 설계 사양을 갖는다.

첫째, 응용 분야인 OpenGL/ES,과 Direct3D 문서를 바탕으로 사용빈도를 고려하여 6가지 연산($\frac{1}{x}$, $\frac{1}{\sqrt{x}}$, $\log_2 x$, 2^x , $\sin(x)$, $\cos(x)$)을 구현한다.

둘째, 범용 마이크로프로세서의 부동 소수점 보조프로세서와 달리 IEEE-754 표준 반올림을 모드를 만족하지 않는 대신에, 1~2 ulp의 오차 범위의 정밀도를 만족시키도록 한다.

셋째, 응용 분야의 정밀도를 고려하여 IEEE-754[7] 단일 정밀도 부동소수점 데이터 형식만 지원한다.

넷째, 2 ulp의 오차를 허용하고, 높은 연산 성능을 위해 파이프라인 구조의 2차 최대최소 근사방식을 사용하였다.

다섯째, 그래픽 프로세서의 경우, 오버플로우와 언더플로우 발생 시 중단이 없는 실시간 처리를 위해 포화 연산 방식을 지원한다.

III. 함수의 2차 최대최소 근사 방법

3.1. 초월함수에 대한 연산 기법

초월함수에 대한 일반적인 다항식 근사 연산 기법은 그림 1과 같이 크게 3 단계로 나누어진다. 사전 처리 단계는 근사에 필요한 다항식 수를 최소화하고 계산을 효율적으로 수행하기 위해, 입력 인자의 범위를 축소시킨다[8]. 두 번째 단계는 다항식 근사를 수행하는 과정이다. 그리고 마지막 단계는 첫 번째 단계에서 수행한 범위 제한 조건을 보정하여 원하는 최종 결과를 생성하는 후처리 과정이다.

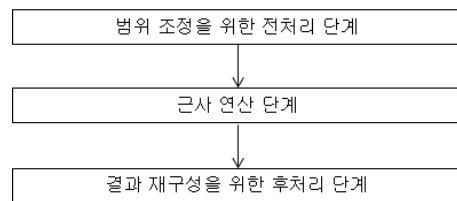


그림 1. 초월 함수의 3-단계 연산
Fig. 1 Three-step calculation of elementary functions

3.2. 인자 범위 조정을 위한 전처리 작업

표 1은 다항식 근사 동작 전에 입력 인자의 범위와 근

사 연산을 한 경우 결과 범위를 나타낸다. 표에서 ‘[’ 와 ‘]’ 표시는 경계 값을 포함하며, ‘(’ 과 ‘)’는 경계 값을 포함하지 않는다.

표 1. 전처리를 통한 입력과 결과 범위
Table. 1 Input and result range by preprocessing

연산	전처리를 거친 입력 범위, (x)	결과의 범위, (f(x))
1/x	[1, 2)	(0.5, 1]
1/√x	[1/4, 1)	(1, 2]
sin(2πx)	(-1, 1)	[-1, 1]
sin(2πx)	(-1, 1)	[-1, 1]
log ₂ x	[1, 2)	[0, 1)
2 ^x	[0, 1)	[1, 2)

표 1에서 역 제곱근(1/√x)의 경우 지수 필드가 짝수일 경우 소수부를 2-비트 우측 이동시키고, 지수 필드가 홀수일 때는 소수부를 1-비트 우측 이동시키고, 지수 값을 각각 2와 1만큼 증가시키는 형태로 얻게 된다. sin(x)와 cos(x) 함수의 경우 삼각함수의 주기성을 이용하여 식 (1)과 같이 IEEE 부동 소수점 형식의 입력 XF를 먼저 (-2π, 2π) 범위를 갖는 x_r로 변환한 후, x_r 값을 2π로 나누어 (-1, 1) 범위의 x로 만들어 근사 동작에서 사용한다. 단, 이러한 입력 인자의 범위 변환 동작은 별도의 부동소수점 연산장치(FP-AU)라는 연산장치에서 수행된다. x = x_r / 2π 이므로 특수 목적 연산장치에서는 sin(2πx) 연산을 수행한다.

$$\sin(XF) = \sin(x_r) = \sin(2\pi x), \quad (1)$$

여기서

$$\begin{aligned} XF &= \text{IEEE 부동 소수점 형식 데이터} \\ x_r &= XF - n \times 2\pi, \\ n &= \lfloor XF/2\pi \rfloor = \text{floor}(XF/2\pi) \\ x_r &\in (-2\pi, 2\pi) \\ x &= \frac{x_r}{2\pi} \\ x &\in (-1, 1) \end{aligned}$$

지수 함수(2^x) 연산의 경우도 식(2)을 통해 IEEE 부동 소수점 형식의 데이터를 부호화 정수(SXI)와 [0, 1) 사이의 값, x로 변환한다. 이렇게 하면 2^x가 부동 소수

점 형식의 정규화된 결과를 출력하므로, SXI가 최종 결과의 지수 값이 된다.

$$X = 2^a \times (\pm 1.X_F) = SXI + PXF = SXI + x \quad (2)$$

여기서,

X = IEEE 부동 소수점 형식 데이터

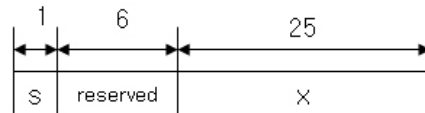
$$X = 2^a \times (\pm 1.X_F) = SXI + PXF$$

$$SXI = \lfloor 2^a \times (\pm 1.X_F) \rfloor = \text{floor}(X)$$

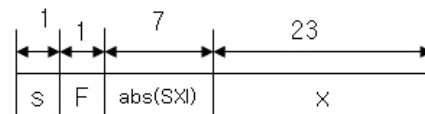
$$PXF = 0.X_F = x \in [0, 1), X_F = \text{소수부 값}$$

$$2^X = 2^{SXI} \times 2^x$$

sin(x)/cos(x)와 2^x 연산의 경우 사전처리 작업을 거쳐 그림 2의 입력 형식을 사용한다. 그림 2(a)의 경우 x값의 범위로 25 비트를 사용하는데, 그 이유는 FP-AU에서 원주율(π) 값으로 근사 값이 사용되기 때문에 정밀도를 향상시키기 위해 25-비트 소수부를 생성하도록 하였다. 그림 2(b)의 경우는 식(3)에 따라 [0, 1)의 범위를 갖는 x가 소수부에 저장되고, SXI 값은 부호-크기 형식으로 변환되어 S와 7-비트 abs(SXI) 필드에 저장된다. F=1인 경우는 지수의 절대 값이 매우 큰 값을 나타내며, 근사 동작에 무관하게 부호 비트에 따라 결과를 0 혹은 최대 양의 값으로 포화 연산을 처리하기 위한 용도로 사용된다.



(a)



S = SXI의 부호
F = 0: 정상적인값
F = 1: 매우 큰 절대값을 갖는 SXI

(b)

그림 2. 삼각함수와 지수 연산에 대한 FP-SFU의 입력 형식

(a) 삼각함수 (b) 지수 함수

Fig. 2 Input data format for trigonometric and exponential function (a) trigonometric function (b) exponential function

입력 범위 조정을 필요로 하는 일부 연산의 변환 작업을 FP-AU에서 별도로 처리하는 것은 FP-SFU에서 수행되는 6가지 동작이 모두 동일한 5단 파이프라인 단계로 처리하기 위함이다.

3.3. 2차 minimax 다항식을 사용한 근사

그래픽 연산의 경우 다항식 근사 과정에 실제 값과 근사 값의 평균 오차를 줄이는 방식보다 최대 오차를 최소화하는 것이 중요하므로 2차 최대최소(minimax) 다항식 근사 방식[9]을 사용하였다. n-비트 입력 인자 x은 m-bit의 상위 부분 X_H 와 (n-m) 비트의 하위 부분 X_L 로 구성되며, 2차 근사다항식의 형태는 식 (3)과 같다.

$$f(x) = C_0 + C_1 X_L + C_2 X_L^2 \tag{3}$$

2^m 개의 구간으로 나뉘는 X_H 의 각 구간에 대해 최대 오차를 최소화하도록 2차 다항식으로 근사 처리를 한다. 2차 다항식의 계수 C_0, C_1, C_2 의 계수 값은 상위 필드 X_H 의 속한 범위와 연산의 종류를 사용하여 각 구간에 대해 최소 오차를 얻도록 계수를 생성한다. 식 (3)의 연산을 계산하는 하드웨어 구조는 개념적으로 그림 3과 같다. 먼저 소수부 필드 X_L 를 사용하여 계수를 생성한 후, 식(3)의 연산을 한다.

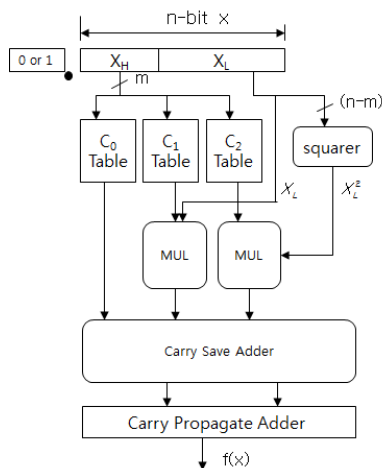


그림 3. 2차 minimax 알고리즘을 사용한 근사 연산 하드웨어 구조
 Fig. 3 Block diagram of 2nd-order minimax approximation method

2차 다항식 근사 방식의 경우 가장 중요한 사항은 계수의 비트 폭과 중간 연산 결과를 표현하는데 사용하는 비트 폭 결정이 중요하다. 본 연구에서는 참고문헌[9]에서 제시된 maple[10]을 프로그램을 변형해서, 6가지 연산에 대한 계수 C_0, C_1, C_2 를 생성하였다. 단, maple 프로그램으로 계수를 구하는 과정에 $\log_2(x)$ 함수는 수렴을 하지 못하는 오류가 발생하여, 등가 식인 $\frac{\log_2(x)}{\log_2 2}$ 를 사용하여 문제점을 해결하였다. 본 연구에서는 3차원 그래픽 분야 응용에서 요구하는 정밀도를 고려하여 $m=8$, 즉 256개의 구간에 대해 개별적인 2차 다항식을 적용하는 방식을 채택하였다. 단, 삼각 함수 sine과 cosine 함수의 경우 표 2와 같이 양의 값을 갖는 1 상한 ($0 \sim \pi/2$) 구간에 대한 계수만 저장해두고, 다른 상한의 경우 $\sin(2\pi x)$ 와 $\cos(2\pi x)$ 의 1 상한 값에서 유도하는 방식을 사용하였다. 따라서 X_H 의 상위 2-비트는 상한 결정에 사용하며, 나머지 6비트를 사용하여, 1 상한에 대한 64개의 계수 값만 저장하도록 하여 ROM 테이블의 면적을 감소시키도록 하였다.

표 2. $\sin(2\pi x)$ 과 $\cos(2\pi x)$ 계산
 Table. 2 Calculation of $\sin(2\pi x)$ and $\cos(2\pi x)$

연산	x가 속한 상한 (x의 상위 2-비트로 판단) ($0 \leq x < 1$)	결과 값 (x' : 1 상한 값) ($0 \leq x' < 1/4$)
$\sin(2\pi x)$	1	$\sin(2\pi x')$
	2	$\cos(2\pi x')$
	3	$-\sin(2\pi x')$
	4	$-\cos(2\pi x')$
$\cos(2\pi x)$	1	$\cos(2\pi x')$
	2	$-\sin(2\pi x')$
	3	$-\cos(2\pi x')$
	4	$\sin(2\pi x')$

6 가지 연산별로 C_0, C_1, C_2 계수 값의 범위가 다른데, 본 연구에서는 Maple 소프트웨어를 통해 얻은 6가지 연산의 최대, 최소 값을 바탕으로 3가지 계수를 분석하여 동일 ROM 크기를 사용하도록, C_0 계수의 경우 29-비트 (3.26 2의 보수 형식: 부호 1-비트, 2-비트 정수 비트, 26-비트 소수 비트), C_1 계수의 경우 20-비트(4.16 형식), C_2 의 경우 14-비트(6.8 형식)을 채택하였다.

EX-3 단계의 경우 EX-2 단계에서 생성한 캐리 보존 결과를 캐리선택 덧셈기를 사용하여 더하는 동작을 수행함과 동시에 $\log_2(x)$ 연산에 대한 처리 작업을 한다. 부동 소수점 데이터에 대한 \log 연산은 식(4)와 같이 계산될 수 있다. 소수부의 결과는 항상 1 미만의 소수 값을 갖는 반면 지수부 값이 정수 값이 되는데, 지수 값(a)이 양수인 경우 2개의 값의 결합시켜 결과를 구하게 되지만, 지수부가 음수인 경우 부호-크기 형식의 소수부 형식을 만들기 위해, (a-1)의 절대 값과 소수부 값의 1의 보수 값을 결합시켜 결과를 얻는다. 원래는 (a-1)의 절대 값과 소수부의 2의 보수 값을 더해 결과를 얻어야 하지만, 오차가 크지 않으므로 1의 보수를 취해 연산을 덧셈 없이 정수부와 소수부를 연결시켜 결과를 얻는 구조로 단순화시켰다.

$$f(x) = \log_2((1.X_F)2^a) = \log_2(1.X_F) + a \quad (4)$$

$$= a + \log_2(1.X_F)$$

$$\approx \begin{cases} +a.Y_F, & \text{if } a \geq 0 \\ -(|a-1|.Y_F), & \text{if } a < 0 \end{cases}$$

여기서, $Y_F = \log_2(1.X_F)$

그림 6은 단계 3의 간략화 된 하드웨어 구조를 나타낸다.

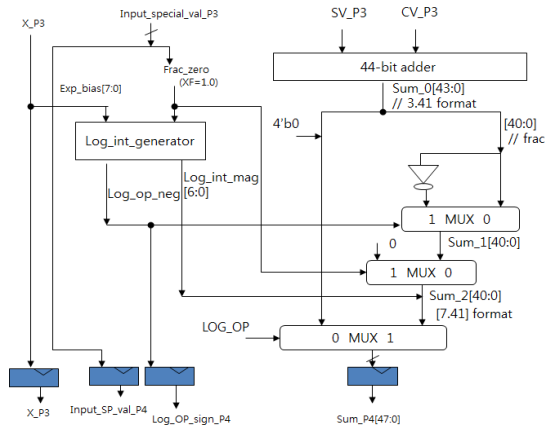


그림 6. EX-3 파이프라인의 하드웨어 구조
Fig. 6 Hardware structure of EX-3 pipeline

EX-4 단계의 주된 연산은 정규화 동작과 최종 지수부 계산 동작이다. $\sin(2\pi x)$ 와 $\cos(2\pi x)$ 함수와 $\log_2(x)$

만 상위 첫 번째 0을 감지하는 동작을 통해 복잡한 정규화 동작이 필요하고, 나머지 연산은 최대 1-비트 좌·우측 이동 동작으로 정규화가 가능하며, 정규화 결과에 따라 지수 값을 보정하게 된다. 따라서 정수부와 소수부에 대해 상위부에서 첫 번째 1의 위치를 감지하는 LOD(leading one detector)를 사용하여 생성한 이동 거리 정보는 정규화 동작과 함께 $\log_2(x)$ 연산과 삼각함수 연산에 대한 지수 값 생성에 사용된다. 그리고 나머지 연산의 경우 전용 하드웨어로 지수 값을 생성한다. 그림 7은 EX-4 단계에 대한 구조를 나타낸다. EX-5 단계는 6개 연산에 대해 특별한 조건, 예를 들면 오버플로우, 언더플로우, 1.0 혹은 0.0을 감지하여, 연산에 맞는 특별한 결과와 포화 연산 처리를 수행한다. 이러한 동작과 함께 IEEE-754 형식에 맞는 출력 형식으로 변환한 결과를 최종적으로 출력한다. 6개 연산의 결과의 부호는 간단한 회로로 생성된다.

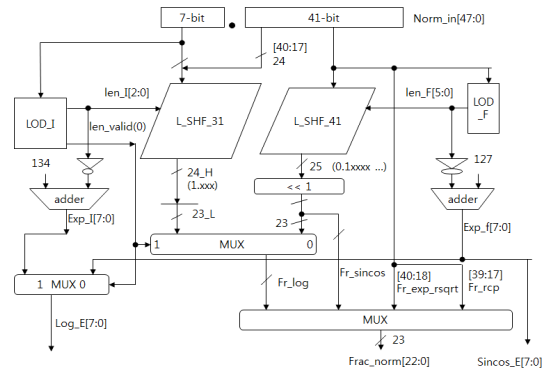


그림 7. EX-4 파이프라인의 정규화 동작
Fig. 7 Normalization operation of EX-4 pipeline

V. 설계 검증 및 성능 분석

설계된 부동 소수점 특수목적 연산회로를 검증하기 위해 C++ 언어를 사용하여 전체 회로에 대한 기능 모델을 만들어 테스트 벡터와 예상 검증 결과를 생성하였다. 기능 모델에서 얻어진 값은 Verilog HDL로 설계한 회로에서 얻어진 결과와 비교를 통해 올바른 회로 설계 검증에 활용되었다. 그림 8은 Modelsim 시뮬레이터를 사용한 $\log_2 x$ 연산에 대한 검증파형을 나타낸다.

REFERENCES

- [1] Jeong-Ho Woo, Ju-Ho Sohn, Byeong-Gyu Nam, and Hoi-Hun Yoo, *Mobile 3D Graphics Soc : From Algorithm to Chip*, John Wiley & Sons, 2010.
- [2] Chang-Hyo Yu, Kyusik Chung, Donghyun Kim, Lee-Sup Kim, "An Energy-Efficient Mobile Vertex Processor With Multithread Expanded VLIW Architecture and Vertex Caches", *IEEE Journal of solid state circuits*, vol. 42, no. 10, pp.2256-2269. oct. 2007.
- [3] Jean-Michel Muller, *Elementary Functions: Algorithms and Implementation*, Birkhauser Press, 1997.
- [4] H. C. Shin, J. A. Lee, and L. S. Kim, " A Minimized Hardware Architecture of fast Phong Shader Using Taylor Series Approximation in 3D Graphics," *Proc. Int'l Conf. Computer Design*, pp.286-291, 1998.
- [5] Ping Tak Peter Tang, "Table-Driven Implementation of the logarithm function in IEEE Floating-Point Arithmetic," *ACM Transactions on Mathematics Software*, vol. 4, no. 16, pp.378-400, Dec. 1990.
- [6] Stuart F. Oberman and Michael Y. Siu, "A High Performance Area Efficient Multifunction Interpolator", *IEEE 11th Symposium on Computer Arithmetic*, pp.272-279, 2005.
- [7] IEEE, *ANSI/IEEE Standard 754-1985: IEEE Standard for Binary Floating-Point Arithmetic*, IEEE Press, 1985.
- [8] K. C. Ng, "Argument Reduction for Huge Arguments : Good to the last Bit," *SunPro*, July 13, 1992.
- [9] Jose-Alejandro Pineiro, Stuart F. Oberman, Jean-Michel Muller, and Javier D. Bruguera, "High-Speed Function Approximation Using a Minimax Quadratic Interpolator," *IEEE Transaction on Computer*, vol.54, no.3, pp.304-318, Mar. 2005.
- [10] Waterloo Maple Inc., *Maple 14 Programming Guide*, 2010.
- [11] S. M. Quek and Larry Hu, "Apparatus for Determining Booth Recoder Input Controls Signals", *US patent*, 5,280,439, Jan. 18, 1994.
- [12] Michael J. Schulte and Earl E. Swartzlander, Jr, "Hardware Design for Exactly Rounded Elementary functions," *IEEE Transaction on Computer*, vol.43, no.8, pp.964-973, Aug. 1994.
- [13] M. Roorda, "Method to reduce the sign bit extension in a multiplier that uses the modified booth algorithm," *Electronics letters*, vol.22, no.20, pp.1061-1062, 1986.



최병윤(Byeong-Yoon Choi)

1985년 2월 : 연세대학교 전자공학과 졸업

1987년 2월 : 연세대학교 공학석사

1992년 8월 : 연세대학교 공학 박사

1993년 3월 ~ 현재 : 동의대학교 컴퓨터공학과 교수

※ 관심분야 : VLIW 프로세서 설계, RISC 설계, 정보 통신 알고리즘의 SoC 설계