

# 저전력 DSP 응용을 위한 오차보상을 갖는 가변 정밀도 승산기 코어 생성기

준회원 황 석 기\*, 이 진 우\*, 정회원 신 경 욱\*

## A module generator for variable-precision multiplier core with error compensation for low-power DSP applications

Seok-Ki Hwang\*, Jin-Woo Lee\* Associate Member, Kyung-Wook Shin\* Regular Member

### 요 약

지정된 비트 크기를 갖는 승산기 코어의 Verilog-HDL 모델을 생성하는 가변 정밀도 승산기 생성기 (VPM\_Gen; Variable-Precision Multiplier Generator)에 대해 기술한다. 사용자의 필요에 따라 승수와 피승수의 비트 수를 8-비트~32-비트 범위에서 1-비트 단위로 선택할 수 있으며, 승산결과는 8-b~64-b 범위에서 2-비트 단위로 절사할 수 있도록 함으로써 총 3,455 가지 승산기 코어를 생성할 수 있다. 승산결과가 절사되는 경우, 절사되는 부분의 회로를 제거함으로써 게이트 수와 전력소모가 각각 최대 40%와 30% 감소되도록 하였으며, 이를 통해 효율적인 저전력 승산기 코어가 구현되도록 하였다. 또한, 절사 비트 수에 따른 적응 오차보상 방법을 적용함으로써 절사 오차가 최소화되도록 하였다. VPM\_Gen에 의해 생성되는 승산기 코어는 Xilinx FPGA 보드와 논리분석기를 통하여 그 기능을 검증하였다.

Key Words : Multiplier, Multiplier Generator, Booth multiplier, Truncation multiplier, IP core

### ABSTRACT

A multiplier generator, VPM\_Gen (Variable-Precision Multiplier Generator), which generates Verilog-HDL models of multiplier cores with user-defined bit-width specification, is described. The bit-widths of operands are parameterized in the range of 8-bit~32-bit with 1-bit step, and the product from multiplier core can be truncated in the range of 8-bit~64-bit with 2-bit step, resulting that the VPM\_Gen can generate 3,455 multiplier cores. In the case of truncating multiplier output, by eliminating the circuits corresponding to the truncation part, the gate counts and power dissipation can be reduced by about 40% and 30%, respectively, compared with full-precision multiplier. As a result, an area-efficient and low-power multiplier core can be obtained. To minimize truncation error, an adaptive error-compensation method considering the number of truncation bits is employed. The multiplier cores generated by VPM\_Gen have been verified using Xilinx FPGA board and logic analyzer.

### I. 서 론

반도체 설계 및 제조기술의 급속한 발전에 따라

시스템 기능을 단일 칩에 집적시키는 SoC (System-on-Chip) 기술이 전자산업의 핵심 기술로 부각되고 있다. SoC는 마이크로프로세서, 메모리, I/O 인터페이스

\* 금오공과대학교 전자공학부 VLSI 설계 연구실 (kwshin@kumoh.ac.kr)

논문번호 : KICS2004-12-311, 접수일자 : 2004년 12월 09일

※본 논문은 2004년도 금오공과대학교 교내학술연구비 지원에 의한 연구결과의 일부임.

※반도체설계교육센터(IDEC)의 CAD Tool 지원에 감사드립니다.

이스, application-specific 매크로 등 다양한 기능블록들을 포함하는 고집적/고성능 반도체로서 설계와 검증이 매우 복잡하고 많은 시간이 소요된다. SoC의 효율적인 설계를 위한 다양한 설계 방법론들이 시도되고 있으며, 최근에는 반도체 설계자산 (Intellectual Property; IP)을 이용한 SoC 설계방법이 궁극적인 해결책으로 인식되고 있다. 반도체 IP는 “SoC 설계에 사용될 수 있도록 가공과 검증이 완료된 표준화된 설계 자산”으로 정의되며, 아날로그/디지털 회로의 물리적 라이브러리에서부터 마이크로프로세서/마이크로컨트롤러 코어 및 관련 소프트웨어 모듈에 이르기까지 매우 광범위한 반도체 설계 자산을 포함한다.<sup>[1-3]</sup>

최근에는 응용분야 및 시스템 사양에 따라 특정 기능블록의 HDL (Hardware Description Language) 모델을 자동으로 생성해주는 모듈 생성기 IP에 대한 관심이 증대되고 있으며, 메모리 생성기, 비터비디코더 생성기, 승산기 및 가산기 생성기 등이 모듈 생성기 IP의 대표적인 예이다. 특히, 승산기는 응용분야 또는 시스템에 따라 요구되는 비트 수가 매우 다양하므로, 비트 수가 고정된 승산기 IP는 활용범위가 매우 제한적이다. 따라서 비트 수를 지정하면 해당 승산기의 HDL 모델이 자동으로 생성되는 모듈 생성기 IP의 필요성이 증대되고 있으며, 국내·외에서 개발된 예로는 복소수 승산기 생성기 PCMUL\_GEN<sup>[4]</sup>와 미국 Xilinx사의 Multiplier Generator<sup>[5]</sup> 등이 있다.

본 논문에서는 디지털 통신 및 신호처리 시스템에 사용되는 다양한 형태의 승산기를 생성하는 가변 정밀도 승산기 생성기 (Variable-Precision Multiplier Generator; VPM\_Gen) IP를 개발하였다.

VPM\_Gen은 승수와 피승수의 비트 수뿐만 아니라 승산결과 비트 수까지 가변시킬 수 있어 승산기가 사용되는 모든 응용분야에서 사용될 수 있다. 승수와 피승수는 8-비트~32-비트 범위에서 1-비트 단위로 선택할 수 있다. 특히 고정점 (fixed-point) 연산방식을 갖는 시스템에서는 승산결과 중 불필요한 하위비트를 제거해서 데이터 패스의 비트 폭에 맞추어야 하므로, 본 논문의 VPM\_Gen은 이와 같은 고정점 연산의 특성을 고려하여 승산결과를 지정된 비트 수로 절사할 수 있는 가변 정밀도 승산기를 생성한다.

고정점 연산방식에서 승산결과 비트의 하위비트 일부를 잘라버리는 경우, 절사방법에 따라 승산결과 오차와 회로의 복잡도 등이 영향을 받는다. 승산결

과를 정확하게 계산한 후, 불필요한 하위비트를 잘라버리는 방법은 절사오차는 작지만 승산결과 모든 비트를 계산하므로 면적과 전력소모 측면에서 낭비를 갖는다. 한편, 절사되는 하위비트의 회로를 생략하는 방법은 칩 면적과 전력소모가 감소되는 장점이 있는 반면에 오차가 커지는 단점을 가지며, 따라서 절사오차를 보상하기 위한 적절한 방법이 필요하다.<sup>[6-11]</sup> 본 논문의 VPM\_Gen은 승산결과를 8-비트~64-비트 범위에서 2-비트 단위로 절사할 수 있으며, 절사되는 비트 수에 따라 가변적으로 오차를 보상하는 방법을 적용함으로써 절사오차가 최소화되도록 하였다.

2장에서는 가변 정밀도 승산기의 구조와 가변 오차보상 방법에 대해 기술하고, 3장에서는 VPM\_Gen의 구조와 특징을, 그리고 4장에서는 VPM\_Gen에 의해 생성된 승산기 코어의 검증 및 성능평가에 관해 기술한다.

## II. 가변 정밀도 승산기 구조 및 오차보상

본 장에서는  $m$ -비트 $\times n$ -비트 승산의 출력  $(m+n)$ -비트 중, 사용자의 필요에 따라 최대  $(m+n)/2$ -비트까지 절사하는 가변 정밀도 승산기 구조와 이때 발생하는 절사오차를 보상하기 위한 오차보상 방법에 대해 기술한다.

### 2.1 가변 정밀도 승산기의 구조

$n$ -비트 승수  $Y = y_{n-1}y_{n-2} \cdots y_1y_0$ 에 대한 radix-4 Booth 인코딩은 식(1)과 같이 정의되며,  $y_{-1}=0$ 이고,  $Q_k = \{+2, +1, 0\}$ 는 승수 3-비트 그룹의 비트 조합에 따라 결정된다. Booth 알고리즘을 적용한  $m$ -비트 $\times n$ -비트 승산은 식(2)와 같이 표현되며, 이를 dot diagram으로 나타내면 그림 1과 같다. 식(1)과 식(2)에 의해  $(m+1)$ -비트의 부분곱  $n/2$ 개와 부분곱의 부호를 나타내는 신호  $s_k$ 가  $n/2$ 개 생성되며, 이들 부분곱을 가산하여  $(m+n)$ -비트의 승산결과가 얻어진다.

$$Y = \sum_{k=0}^{n/2-1} (-2y_{2k+1} + y_{2k} + y_{2k-1}) \cdot 2^{2k} \quad (1)$$

$$= \sum_{k=0}^{n/2-1} Q_k \cdot 2^{2k}$$

$$P = \sum_{k=0}^{n/2-1} Q_k \cdot X \cdot 2^{2k} \quad (2)$$

데이터 패스의 비트 수가 고정된 고정점 연산방식에서는  $(m+n)$ -비트의 승산결과 중 하위비트 일부를 잘라버리고 나머지 상위비트만 사용되므로, 절사되는 비트 수를  $\tau$  (단,  $0 \leq \tau \leq (m+n)/2$ ) 라고 하면, 절사 후의 승산결과는  $(m+n-\tau)$ -비트가 된다. 그림 1에서 보는 바와 같이, radix-4 Booth 알고리즘에 의해 생성된 부분곱은 2-비트씩 쉬프트된 형태를 가지므로,  $\tau$ 는 2-비트 단위로 선택되도록 한다. 한편, 그림 1에서  $\tau$ 에 따른 cut-line 오른쪽의 부분곱 연산을 제거하면, 전체 승산기 회로가 단순해져 면적과 전력소모를 줄일 수 있다. 그러나  $\tau$ 가 증가함에 따라 부분곱의 절사되는 비트들이 증가하므로, 승산결과 오차도 증가하게 되어 적절한 형태의 오차보상 방법이 필요하게 된다.

본 논문에서 설계되는 승산기는 그림 2와 같은 구조를 가지며, 승산결과 오차의 절사 비트 수  $\tau$ 가  $(0 \sim h)$ -비트 (단,  $h=(m+n)/2$ ) 범위에서 2-비트 단위로 선택됨에 따라  $(m+n-\tau)$ -비트의 승산결과를 출력하는 가변 정밀도를 갖는다. 일반적인 Booth 승산기와 다른 점은, 그림 1에 표시된 cut-line 오른쪽의

절사되는 부분곱을 처리하는 회로 (Booth 디코더, carry-save adder 그리고 carry-propagation adder)가 제거되어 회로가 크게 단순화된다는 것이다. 또한, 절사에 의해 발생된 오차를 보상하기 위한 오차보상 및 보정 회로 (EC&C)가 추가되었으며, 오차보상 방법 및 EC&C 회로에 대한 자세한 내용은 2.2 절에서 설명된다.

부분곱 생성블록은 Booth 인코더와 디코더로 구성되며, Booth 인코더는  $n$ -비트 승수를 radix-4 Booth 인코딩하여  $n/2$ 개의 3-비트 신호를 생성하여 Booth 디코더로 보낸다. Booth 디코더는 인코더에서 입력되는 신호를 받아  $m$ -비트의 피승수로부터  $n/2$ 개의  $(m+1)$ -비트 부분곱을 생성하여 부분곱 가산기로 보낸다. Booth 인코더/디코더는 Compact & Race-free 방식을 사용함으로써 회로를 간소화하였으며, 아울러 디코더 출력신호의 불필요한 논리치이를 방지함으로써 저전력 특성을 갖도록 설계되었다.<sup>[4]</sup> 생성된 부분곱은 캐리보존 가산기 (carry-save adder; CSA)에 의해 가산되고, 그 결과는 캐리전파 가산기 (carry propagation adder; CPA)를 거쳐 최종 출력이 얻어진다.

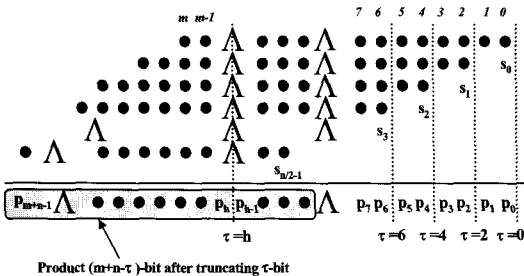


그림 1.  $m$ -비트 $\times n$ -비트 Booth 승산의 dot diagram

### 2.2 가변 정밀도 Booth 승산기의 오차보상

본 절에서는 승산결과 중 절사되는 비트 수  $\tau$ 가  $(0 \sim (m+n)/2)$ -비트의 범위에서 가변되는 가변 정밀도 승산기의 오차보상 방법에 대해 설명한다.

일반적으로, 절사승산기의 오차보상 방법은 고정오차보상 방법과 근사화된 캐리의 합을 이용하여 오차를 보상하는 가변오차보상 방법으로 구분된다. 절사된 하위 비트의 부분곱에서 발생할 수 있는 캐리의 확률 분석을 토대로 고정된 보상 값을 구하는 방법<sup>[7]</sup>은 부가적인 회로를 사용하지 않고 오차보상이 가능하다는 장점은 있으나, 오차보상에도 불구하고 절사오차가 비교적 크다는 단점이 있다. 한편, 절사된 부분곱에서 발생할 수 있는 캐리의 합을 생성하고 이를 가산해 줌으로써 절사오차를 최소화하는 다양한 방법들이 제안되고 있다.<sup>[6,8-11]</sup> 그러나 문헌에 발표된 기존의 오차보상 방법들은  $\tau=(m+n)/2$ 로 고정된 경우의 fixed-width 배열 승산기 구조에 대한 보상 방법만을 제시하고 있으며,  $\tau$ 가  $0 \sim (m+n)/2$ 의 범위에서 가변되는 Booth 승산기 구조에 대한 보상 방법은 제시되지 못하고 있다.

본 논문에서는 가변 정밀도 Booth 승산기에 적용하기 위한 새로운 오차보상/보정 알고리즘을 제안한다. 제안되는 방법은 절사되는 비트 수  $\tau$ 에 따라 가변

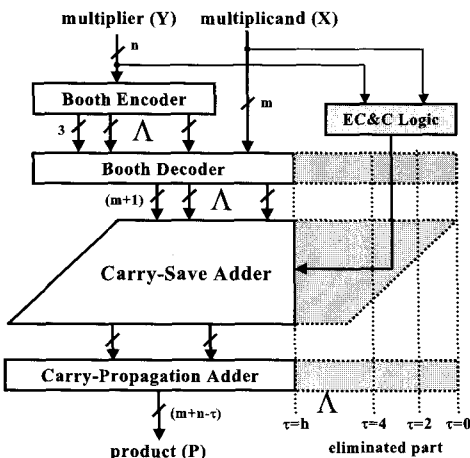


그림 2. 오차보상을 갖는 가변 정밀도 승산기 구조

적으로 보상 값을 생성하고, 이를 부분곱 가산에 포함시키는 가변오차보상 방법이다. 먼저,  $\tau=h$  (단,  $h=(m+n)/2$ )로 고정된 경우의 오차보상은 아래의 식(3)과 같이 정의되며, 이는 절사된 최하위 비트에서  $(h-1)$ -번째 비트까지 캐리의 합을 나타낸다. 한편, 절사된 부분의 최상위  $(h-1)$ -번째 비트 위치의 부분곱 합을 나타내는  $\beta$ 는 식(4)와 같이 정의된다.

$$\alpha_{h-1} \cong \bar{\alpha}' = \begin{cases} \sum_{k=1}^{\frac{h}{2}-1} x_{2k-1}y_{(h-1)-(2k-1)} + 1, & \text{if } \beta = 0 \\ \sum_{k=1}^{\frac{h}{2}-1} x_{2k-1}y_{(h-1)-(2k-1)}, & \text{if } \beta > 0 \end{cases} \quad (3)$$

$$\beta = \sum_{i+j=h-1} x_i y_j = x_{h-1}y_0 + x_{h-2}y_1 + \dots + x_0y_{h-1} \quad (4)$$

식(3)과 식(4)는 그림 2에서  $\tau$ 의 cut-line으로 절사되는 부분의 계산이 생략됨에 따라 발생하는 오차를 보상한다. 그러나 radix-4 Booth 알고리즘은 승수 Y의 3-비트 그룹이 갖는 비트 조합에 따라  $Q_k$ 가 결정되어 부분곱을 생성하므로, 다음과 같은 세 가지 사항이 추가적으로 고려되어야 한다. 첫째, 식(1)과 식(2)의 Booth 인코딩으로 생성되는 부분곱은  $(-2y_{2k+1} + y_{2k} + y_{2k-1}) \cdot X \cdot 2^{2k}$ 로 표현되므로, 그림 1에 표시된  $\tau$ 의 cut-line으로 절사하게 되면 cut-line 바로 우측 즉,  $(\tau-1)$ -번째 열의 부분곱 항들은 비트 가중치 (weight)가 동일하지 않게 된다. 둘째, 절사되는 부분 중  $-2y_{2k+1} \cdot X$ 을 2의 보수 수치계에서 양수로 변환하기 위한 보정이 고려되어야 한다. 셋째, 생성된 부분곱의 부호를 나타내는  $n/2$ 개의  $s_k$ 가  $\tau$ 의 cut-line에 의해 절사되므로 이에 대한 보정도 필요하다. 본 논문에서는 이와 같은 세 가지 사항을 반영하기 위해 식(5)와 같이 정의되는 보정 항  $\sigma_h$ 가 추가적으로 포함되도록 하였다.

$$\sigma_h = \sum_{k=1}^{\frac{h}{2}-1} (y_{2k-1} + y_{2k} + y_{2k+1} + x_{2k-1}y_{(h-1)-(2k-1)}) \quad (5)$$

한편, 식(3)과 식(5)에서 정의된 오차보상 및 보정 항을  $\tau$ 가  $(0 \sim h)$ -비트의 범위에서 가변되는 가변 정밀도 Booth 승산기에 적용하기 위해서는 각각 식(6)과 식(7)과 같이 수정되어야 한다.

$$\alpha_{\tau-1}^* \cong \alpha' = \begin{cases} 0, & \text{if } \beta = 0 \\ \sum_{k=1}^{\frac{\tau}{2}-1} x_{2k-1}y_{(\tau-1)-(2k-1)} - 1, & \text{if } \beta > 0. \end{cases} \quad (6)$$

$$\sigma_{\tau}^* = \sum_{k=1}^{\frac{\tau}{2}-1} (y_{2k-1} + y_{2k} + y_{2k+1}) + \bar{y}_0 + y_1 + \sum_{k=1}^{\frac{\tau}{2}-1} x_{2k-1}y_{\tau-(2k-1)} \quad (7)$$

그림 3은 제안된 오차보상 방법을 회로로 구현한 회로도이다. 그림 3-(a)는  $\tau=h$ 인 경우 즉,  $(m+n)$ -비트의 승산결과 중 하위  $h=(m+n)/2$ -비트를 절사하는 경우에 적용되는 보상 항 식(3)을 회로로 구현한 것이며, 그림 3-(b)는  $\tau$ 가  $(0 \sim h)$ -비트 범위에서 가변되는 경우의 보상 항 식(6)을 구현한 것이다. 그림에서 보는 바와 같이, 오차보상 회로는 승수와 피승수로부터  $h/2$ 개의 보상신호  $c_i$ 를 생성하고, 생성된 보상신호는 부분곱 가산회로에서 가산된다. 제안된 보상회로는 단순한 OR 게이트와 AND 게이트로 구현되므로 오차보상을 위해 추가되는 하드웨어가 매우 작다는 장점을 갖는다.

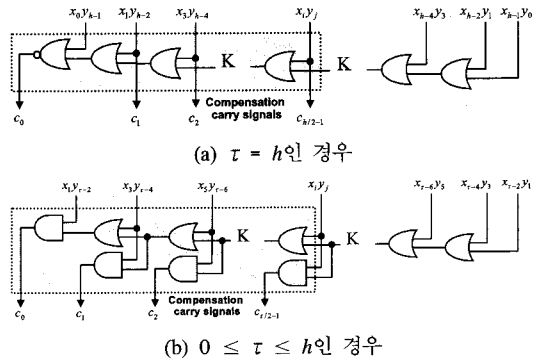


그림 3. 가변 정밀도 승산기의 오차보상 회로

그림 4는 승수와 피승수가 모두 8-비트이고, 승산 결과의 하위 8-비트를 절사하는 경우 (즉,  $\tau=8$ 인 경우)에 대한 승산기 회로를 보인 것이다. 그림 4-(a)의 dot diagram에서 cut-line 우측의 절사되는 부분곱에 대한 연산회로는 실제 회로에서 구현되지 않으며, 대신 오차보상 및 보정회로로 대체된다. 그림 4-(b)에서 오차보상회로는 그림 3-(a)의 회로에  $h=8$ 을 대입하여 얻어진 것이다. 한편, 그림 4-(b)에서 보정(correction)회로는 앞에서 설명된 식(5)를

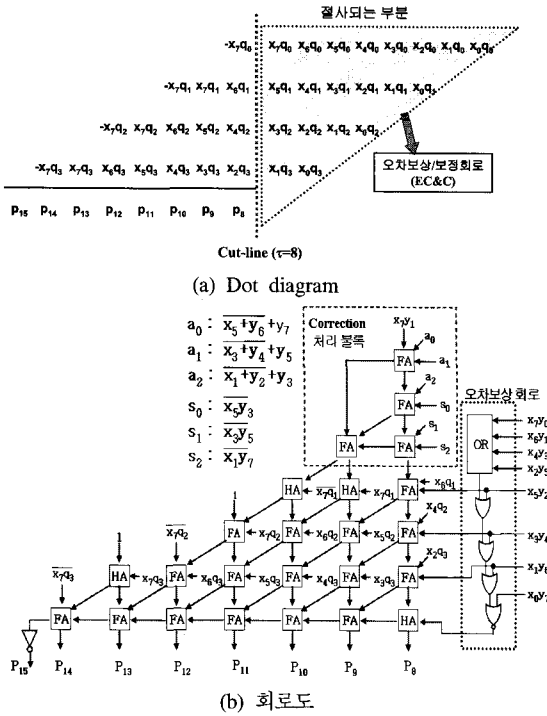


그림 4. 오차보상을 갖는 8-비트×8-비트 Booth 승산기( $\tau=8$ 인 경우)

구현한 것이며, NAND, NOR 게이트와 전가산기로 구현된다.

### III. 가변 정밀도 승산기 생성기 (VPM\_Gen)

본 장에서는 2장에서 설명된 오차보상 방법과 절사 승산기 구조를 갖는 가변 정밀도 승산기 생성기 (VPM\_Gen)에 대해 설명한다.

VPM\_Gen는 사용자가 승수와 피승수의 비트 수  $m$ 과  $n$ 을 각각 8-b~32-b 범위에서 1-b 단위로 지정하고,  $(m+n)$ -비트의 승산결과 중 절사되는 비트 수  $\tau$ 를 2-비트 단위로 지정하면 해당 사양을 만족하는 승산기의 Verilog-HDL 모델을 생성하는 기능을 갖는다. 이때 생성되는 승산기는 2장에서 설명된 구조 및 오차보상회로가 적용되어 구현된다.

VPM\_Gen에서 절사 승산기의 Verilog-HDL 모델을 생성하는 내부 흐름도는 그림 5와 같다. 승산기의 입력 비트 수가 지정되면 이들 중 작은 값을 승수의 비트 수로, 큰 값을 피승수의 비트 수로 결정한다. radix-4 Booth 승산기는 부분곱의 개수가 승수의 비트 수에 비례하므로, 승수의 비트 수가 작을수록 부분곱 가산회로가 단순해지며 승산속도를 향상시킬 수 있다.

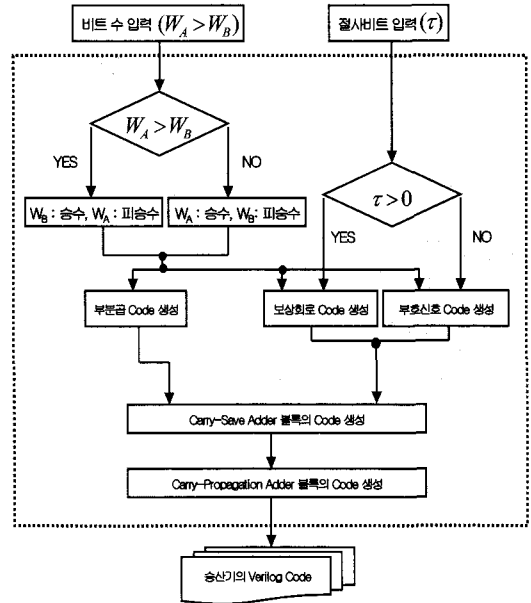


그림 5. VPM\_Gen의 흐름도

승수와 피승수의 비트 수가 확정되면, 승산결과와 절사 비트 수  $\tau$  값을 판별한다.  $\tau > 0$  이면 오차보상 회로를 생성하고,  $\tau=0$ 인 경우에는 Booth 인코더에서 생성되는 부분곱 부호신호만을 생성한다.  $\tau$  값에 따라 각 블록의 비트 수가 결정되며, 이를 기반으로 Booth 인코더 및 디코더 블록, 캐리보존 가산기 (CSA) 블록 및 캐리전파 가산기 (CPA) 블록 등을 생성하여 전체 승산기의 Verilog 모델이 완성된다. 생성되는 승산기 코어의 입/출력 핀에 대한 설명은 표 1과 같다.

표 1. I/O pin descriptions

Signal	I/O	Width	Description
clk	I	1	External clock signal
rst	I	1	Reset signal
Y	I	n	Multiplier
X	I	m	Multiplicand
P	O	$m+n-\tau$	Product Output

VPM\_Gen은 사용자의 필요에 따라 3,455가지 종류의 승산기 Verilog-HDL 모델을 생성할 수 있다. 그림 6은 VPM\_Gen의 실행화면이며, 32-비트×32-비트,  $\tau=32$ 인 승산기 코어가 생성되는 화면을 보인 것이다.

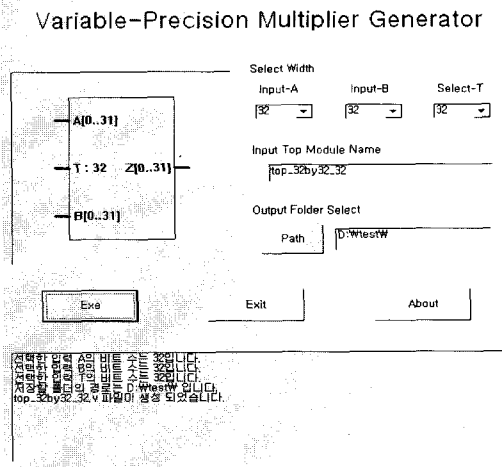


그림 6. VPM\_Gen의 실행화면

#### IV. 성능 평가 및 검증

##### 4.1 생성된 승산기 모델의 검증

VPM\_Gen의 검증은 C 언어 모델링으로부터 50,000개의 랜덤 테스트 벡터를 생성하고, 이를 Verilog testbench에서 읽어 논리 시뮬레이션을 수행하였다. 그림 7은 검증방법의 흐름을 나타낸 것이다. 승산기 검증을 위해 개발된 검증 프로그램 MTV\_Gen (Multiplier Testbench & Vector Generator)을 이용하여 testbench와 벡터파일을 생성한다. 생성된 testbench와 벡터파일, 그리고 승산기 코어의 Verilog Code를 가지고 ModelSim을 사용하여 출력파일을 만든다. 출력파일과 벡터파일을 다시 MTV\_Gen의 입력으로 넣고 최종 결과파일을 얻는다. 이와 같은 방식으로 시뮬레이션 결과와 MTV\_Gen으로 생성된 결과를 비교하여 논리동작을 검증하였다.

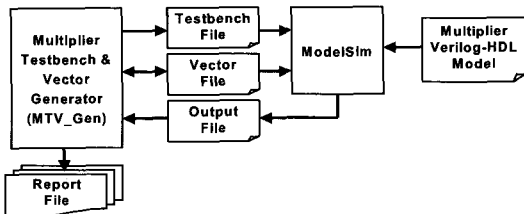


그림 7. 승산기 코어의 기능 검증 흐름도

그림 8은 개발된 생성기의 하드웨어 검증 시스템을 보인 것이며, FPGA 보드, 로직 분석기 등으로 구성된다. FPGA는 Xilinx XCV600E를 사용하였으며, 그림 9는 로직 분석기를 이용한 검증결과의 일부이다.

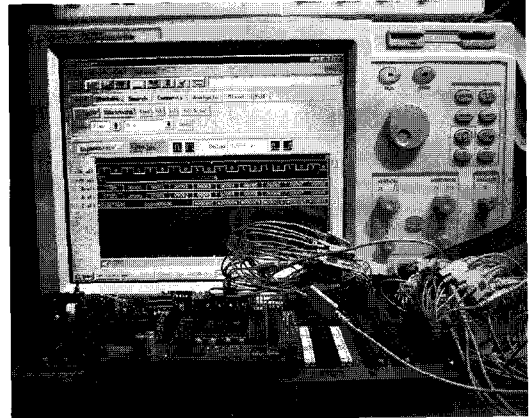


그림 8. VPM\_Gen의 검증 시스템 사진

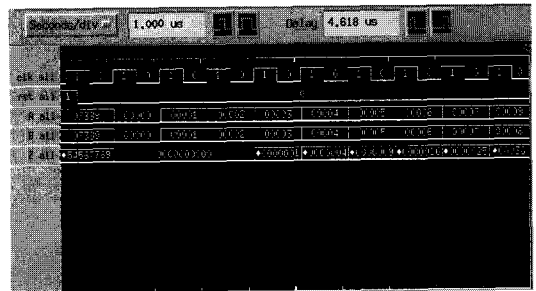


그림 9. 로직 분석기를 이용한 검증결과

##### 4.2 성능평가

생성된 승산기의 승수와 피승수의 비트 수, 그리고 승산결과의 절사비트 수에 따른 게이트 수, 전력소모 그리고 평균 절사오차 등의 성능을 평가하였다. 생성된 승산기의 면적은 Synopsys와 0.25- $\mu$ m 표준 셀 라이브러리를 이용한 논리합성을 통하여 분석하였다. 그림 10은 VPM\_Gen에서 생성되는 승산기 중, 3가지 경우 (32-비트 $\times$ 32-비트, 24-비트 $\times$ 24-비트, 16-비트 $\times$ 16-비트)에 대해  $\tau$ 를 가변시키면서 얻어진 승산기의 게이트 수를 나타낸 것이다. 32-비트 $\times$ 32-비트 승산기의 경우를 예로 들면,  $\tau=32$ 인 경우 5,800 게이트이며  $\tau=0$ 인 경우의 9,540 게이트에 비해 약 40%의 면적 감소가 얻어진다.

그림 11은 승수와 피승수의 비트 수  $m$ 과  $n$ 을 각각 8-비트~32-비트 범위에서 변화시키면서  $\tau=0$ 인 경우와  $\tau=(m+n)/2$ 인 경우의 전력소모 특성을 시뮬레이션 한 결과이다. 전력소모는 50,000개의 벡터를 사용해서 승산기 내부의 switching activity를 산출하여 Synopsys를 통해 측정하였다. 32-비트 $\times$ 32-비트 승산기의 경우,  $\tau=32$ 인 승산기의 전력소모는 약 2.22-mW이며 절사되지 않은 승산기의 전력소모 3.29-mW에 비해 약 30%의 전력 감소가 얻어진다.

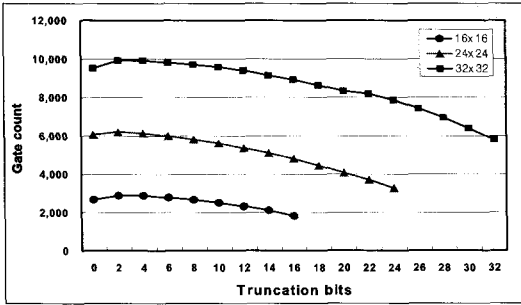


그림 10.  $\tau$ 에 따른 승산기의 게이트 복잡도

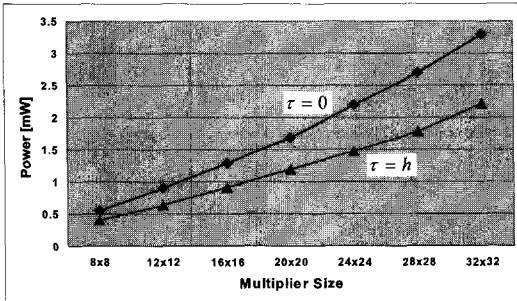


그림 11.  $\tau$ 에 따른 전력소모 특성

표 2와 표 3의 결과는 VPM\_Gen에 의해 생성된 승산기의 절사오차 특성을 분석한 결과이다. 50,000개의 테스트 벡터를 사용하여 구해진 평균오차를 오차보상이 없는 단순 절사 승산기와 비교한 것이다. 평균오차  $E_V$ 는 식(8)로 정의되며,  $P_F$ 는 절사되지 않은 승산결과를 나타내고  $P_T$ 는 절사된 승산결과를 나타내며,  $M$ 은 테스트 벡터의 수를 나타낸다. 표 2와 표 3으로부터, 2.2절에서 제안된 본 논문의 오차보상 방법은 오차보상을 갖지 않는 경우에 비해 평균오차가 최소 10%에서 최대 90%까지 감소하여 매우 우수한 성능을 나타냄을 알 수 있다.

$$E_V = \frac{\sum |P_F - P_T|}{M} \quad (8)$$

표 2.  $\tau$ 에 따른 승산기의 평균 오차 비교 (32-비트×32-비트 승산기에 50,000개의 테스트 벡터를 인가)

$\tau$	오차보상을 하지 않은 경우	본 논문의 경우
8	325.05 (1.0)	119.81 (0.37)
16	212,590 (1.0)	37,846 (0.18)
24	$8.78 \times 10^7$ (1.0)	$1.0 \times 10^7$ (0.11)
32	$4.8 \times 10^{10}$ (1.0)	$4.3 \times 10^9$ (0.09)

표 3.  $\tau=(m+n)/2$ 인 경우의 승산기 평균오차 비교 (50,000의 테스트 벡터를 인가)

비트 수	오차보상을 하지 않은 경우	본 논문의 경우
16	155,510 (1.0)	52,820 (0.34)
24	$8.58 \times 10^7$ (1.0)	$1.52 \times 10^7$ (0.18)
32	$4.8 \times 10^{10}$ (1.0)	$4.3 \times 10^9$ (0.09)

## V. 결론

일반적으로, 디지털 통신 및 신호처리 분야의 핵심 연산 장치인 승산기는 응용분야 및 시스템에 따라 사용되는 비트 수가 매우 다양하다. 본 논문에서는 매우 광범위한 응용분야에서 폭넓게 사용될 수 있도록 승수와 피승수의 비트 수를 8-비트~32-비트 범위에서 선택하고, 승산결과 비트 수를 8-비트~64-비트 범위에서 선택할 수 있도록 함으로써 총 3,455 종류의 가변 정밀도 승산기를 생성할 수 있는 승산기 생성기 VPM\_Gen을 개발하였다.

VPM\_Gen은 승산결과 비트 수에 따라 가변적으로 오차가 보상되는 방법을 적용함으로써 절사오차를 획기적으로 감소시키고 아울러 면적이 최소화 되도록 설계되었다. VPM\_Gen은 C 모델링과 ModelSim 및 Synopsys를 이용한 논리합성과 시뮬레이션을 통해 검증하였다. VPM\_Gen에 의해 생성되는 승산기는 게이트 수와 전력소모 그리고 평균 절사오차 특성이 매우 우수하므로, 고정점 연산을 사용하는 각종 디지털 신호처리 및 디지털 통신 시스템의 저전력 설계에 폭넓게 이용 가능할 것이다.

## 참고 문헌

- [1] 서지선, 유희준, "IP 기반 SOC 설계기술 동향", *대한전자공학회지*, vol. 29, no. 1, pp. 25-31, 2002.
- [2] M. Keating and P. Bricaud, *Reuse Methodology Manual for System-On-A-Chip Designs* (3rd Edition), Kluwer Academic Publishers, 2002.
- [3] 강성호, 김대정, 이승준, 이찬호, "HDL을 이용한 SoC 및 IP 설계기법", *홍릉과학출판사*, 2004.
- [4] 양대성, 이승기, 신경욱, "복소수 승산기 코어의 파라미터화된 소프트 IP 설계", *한국통신*

학회 논문지, vol. 26, no. 10B, pp.266- 274, Oct. 2001.

- [5] "Multiplier Generator V6.0", <http://www.xilinx.com/ipcenter>, 2004.
- [6] 정해현, 박종화, 신경욱 "작은 오차를 갖는 절사형 Booth 승산기", 한국해양정보통신학회 논문지, vol. 5, no. 2, pp. 617-620, Oct. 2001.
- [7] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-Efficient Multipliers for Digital Signal Processing Applications," *IEEE Trans. on Circuits and Systems II*, vol. 43, no. 2, pp. 90-95, Feb. 1996.
- [8] J. M. Jou, S. R. Kuang, and R. D. Chen, "Design of Low-Error Fixed-Width multipliers for DSP Application," *IEEE Trans. on Circuits and Systems II*, vol. 46, no. 6, pp. 836-842, Jun. 1999.
- [9] S. Y. Jou, M. H. Tsai, and Y. L. Tsao, "Low-Error Reduced-Width Booth Multipliers for DSP Applications", *IEEE Trans. on Circuits and Systems I*, vol. 50, no. 11, pp. 1470-1474, Nov. 2003.
- [10] S. J. Jou and H. H. Wang, "Fixed-width multiplier for DSP application." *Proc 2000 Int. conf. Computer Design (ICDD)*, pp. 318-322, Sep. 2000.
- [11] L. D Van, S. S. Wang, S. Tengchen, W. S. Feng, B. S. Jeng, "Design of a lower error fixed-width multiplier for speech processing application," *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 3, pp. 130-133, May 1999.

黄石起(Seok-Ki Hwang)

준회원



2004년 2월 금오공과대학교 전자공학과 졸업  
 2004년 3월~현재 금오공과대학교 전자공학과 석사과정 재학 중  
 <관심분야> 정보보호 SoC 설계 반도체 IP 설계

李振宇(Jin-Woo Lee)

준회원



2004년 2월 금오공과대학교 전자공학과 졸업  
 2004년 3월~현재 금오공과대학교 전자공학과 석사과정 재학 중  
 <관심분야> 마이크로프로세서 설계, SoC 설계, 통신 및 신호 처리용 집적회로 설계

辛 卿 旭(Kyung-Wook Shin)

정회원



1984년 2월 한국항공대학교 전자공학과 졸업  
 1986년 2월 연세대학교 대학원 전자공학과(공학석사)  
 1990년 8월 연세대학교 대학원 전자공학과(공학박사)  
 1990년 9월~1991년 6월 한국 전자통신연구소 반도체연구단 (선임연구원)  
 1991년 7월~현재 금오공과대학교 전자공학부(교수)  
 1995년 8월~1996년 7월 University of Illinois at Urbana-Champaign(방문교수)  
 2003년 1월~2004년 1월 University of California at San Diego(방문교수)  
 <관심분야> 통신 및 신호처리용 SoC 설계, 네트워크 정보보호 SoC 설계, 반도체 IP 설계