

# 비대칭 축소 및 확대가 가능한 조합 보간 알고리즘의 실시간 처리를 위한 하드웨어 설계

## Hardware Design for Real-Time Processing of a Combinatorial Interpolation Scaler with Asymmetric Down-scaling and Up-scaling

한 시 연<sup>\*†</sup>, 정 세 민<sup>\*†</sup>, 손 정 현<sup>\*</sup>, 이 재 성<sup>\*</sup>, 강 봉 순<sup>\*★</sup>

Si-Yeon Han<sup>\*†</sup>, Semin Jung<sup>\*†</sup>, Jeong-Hyeon Son, Jae-Seong Lee, Bong-Soon Kang<sup>\*★</sup>

### Abstract

Recently, various video resolution formats have emerged, and digital devices have built in dedicated scaler chips to support them by enlarging or reducing the resolution of input videos. Therefore, the performance and hardware size of scaler chips are important. In this paper, the combinatorial interpolation scaler algorithm proposed by Han is used to design the hardware using the line memory structure with dual-clock proposed by Han and Jung. The proposed hardware is capable of real-time processing in QHD environments, designed using Verilog, and validated using Xilinx's Vivado 2023.1. We also verify the performance of Han's proposed algorithm with a quantitative numerical evaluation of the proposed hardware.

### 요 약

최근 다양한 영상의 해상도 포맷이 등장하였고, 디지털 기기는 이를 지원하기 위해 입력 영상의 해상도를 확대 또는 축소하는 전용 스케일러 칩을 내장하고 있다. 따라서 스케일러 칩의 성능과 하드웨어 크기는 중요하다고 할 수 있다. 본 논문에서는 Han이 제안한 조합 보간 스케일러 알고리즘을 Han, Jung이 제안한 Dual-clock을 가지는 라인 메모리 구조를 이용해 하드웨어 설계를 진행하였다. 제안하는 하드웨어는 QHD 환경에서 실시간으로 처리가 가능한 구조로, Verilog를 이용해 설계되었으며 Xilinx Vivado 2023.1을 이용하여 검증하였다. 또한 Han이 제안한 알고리즘과 하드웨어의 정량적 수치 평가 비교를 통해 성능을 검증하였다.

*Key words* : scaler, combinatorial, line memory, real-time, verilog

---

\* Dept. of Electronics Engineering, Dong-A University

★ Corresponding author

Email : bongsoon@dau.ac.kr, Tel : +25-51-200-7703

† These authors contributed equally to this work.

※ Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2023R1A2C1004592)

Manuscript received Feb. 26, 2024; revised Mar. 14, 2024; accepted Mar. 15, 2024.

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.

## I. 서론

멀티미디어 기술의 발전으로 Full HD, 4K Ultra HD 등 다양한 영상 포맷이 등장하였다. 이에 따라 스마트폰, 태블릿 PC, TV와 같은 다양한 디지털 기기에는 영상의 해상도를 변환(축소 또는 확대)하기 위한 전용 스케일러 (scaler) 칩을 내장하고 있다. 이처럼 스케일러 칩은 디지털 기기에 필수적으로 포함되기에 설계되는 칩의 하드웨어 크기 및 성능은 중요하다고 할 수 있다.

영상의 해상도를 변환하는 데 이용되는 대표적인 알고리즘으로 nearest neighbor interpolation(NNI)[1], bilinear interpolation(BLI)[2], bicubic interpolation(BCI)[3]가 있다. 이 알고리즘들은 NNI, BLI 그리고 BCI 순으로 수식이 점점 복잡해지고, 연산하는 데 필요한 픽셀 수는 각각 1개, 2개 그리고 4개로 증가한다. 따라서 알고리즘의 성능은 BCI, BLI 그리고 NNI순으로 우수하며, 하드웨어 구현 시 발생하는 복잡도는 반대가 된다. 하드웨어 복잡도를 줄이기 위해서는 다양한 방법이 존재하는데, 그중 효과적인 방법은 복잡도가 매우 높은 라인 메모리의 사용을 줄이는 것이다. 라인 메모리는 주로 Y축 연산에 사용되는 반면, X축 연산에는 매우 작은 하드웨어 복잡도를 가지는 플립플롭 소자를 주로 이용한다. 따라서 Han은 각 알고리즘의 특성을 고려하여 X, Y축에 각각 다른 알고리즘을 적용함으로써 하드웨어 복잡도를 낮춘 combinatorial interpolation(CI)을 제안했다 [4]. CI 알고리즘은 Y축에 인접한 2개의 픽셀을 사용하는 modified bilinear interpolation(mBLI)[4]를 적용하였다. mBLI는 라인 메모리를 1개만 사용하기에 3개를 사용하는 BCI보다 메모리 자원을 줄일 수 있다. 반면 X축에는 플립플롭 소자를 사용하기에 알고리즘은 복잡하지만, 성능이 가장 우수한 BCI를 적용했다. 따라서 CI 알고리즘의 성능은 BCI보다는 떨어지고 mBLI보다는 높다[4]. 또한 Han, Jung은 스케일러를 하드웨어로 구현 시 사용하는 라인 메모리 종류 및 clock 개수에 따른 이미지 처리 속도를 비교하였다[5]. 이를 통해 dual-clock을 사용하고 라인 메모리로 Dual-Port RAM(DPRAM)을 사용할 경우 실시간 처리 속도가 가장 빠른 것을 보였다[5]. 따라서 본 논문에서는 Han이 제안한 CI 알고리즘을 dual-clock을 사용하는 DPRAM 구조로 QHD (2560×1440) 환경에서 실시간 처리가 가능한 하드웨어로 구현하였다. 이때 실시간 처리 기준은 PAL과 NTSC 시스템[6]을 만족하는 30 frames per second (fps) 이상이다.

본 논문의 구성은 다음과 같다. 2장에서는 CI 알고리즘의 수식을 간략히 설명한 후 하드웨어 구조에 관해 설명한다. 3장에서는 Xilinx 합성 결과를 제시하여 QHD 환경에서 실시간 처리가 가능함을 보인다. 또한 Han이 제안한 CI 알고리즘과 본 논문에서 설계한 하드웨어의 정량적 수치 평가 비교를 진행한 후 4장에서 본 논문의 결론을 서술하여 마무리한다.

## II. 본론

### 1. 조합 보간 알고리즘

Han이 제안한 CI 알고리즘[4]에 자세히 설명되어 있듯이, X, Y 축으로 보간 계산을 수행하기 위해서는 현재 유효한 픽셀 위치( $x_{nx}$ ,  $y_{ny}$ )와 보간이 필요한 위치 ( $x$ ,  $y$ )이 필요하다. 이를 구하기 위해 X, Y축의 확대 및 축소비율인  $xratio$ 와  $yration$ 에 따른 증가 값  $xinc(nx)$ ,  $yinc(ny)$ 가 필요하며, 수식 1~2[4]를 통해 구할 수 있다. 이때  $xinc(nx)$ 의 초깃값  $xinc(0)$ 는 0이고,  $yinc(ny)$ 의 초깃값  $yinc(0)$ 는 수식 3[4]를 사용해 연산한다. 계산된 각 축의 증가 값을 이용하여 수식 4~5 [4]와 같이 유효한 픽셀 위치( $x_{nx}$ ,  $y_{ny}$ )와 보간이 필요한 위치( $x$ ,  $y$ )를 구할 수 있다.

$$xinc(nx) = \frac{1}{xratio} + xinc(nx-1) \quad (1)$$

$$yinc(ny) = \frac{1}{yration} + yinc(ny-1) \quad (2)$$

$$yinc(0) = \frac{1}{yration} + 0.5 \times (1 - \frac{1}{yration}) \quad (3)$$

$$x_{nx} = floor\{xinc(nx)\}, x = xinc(nx) - x_{nx} \quad (4)$$

$$y_{ny} = floor\{yinc(ny)\}, y = yinc(ny) - y_{ny} \quad (5)$$

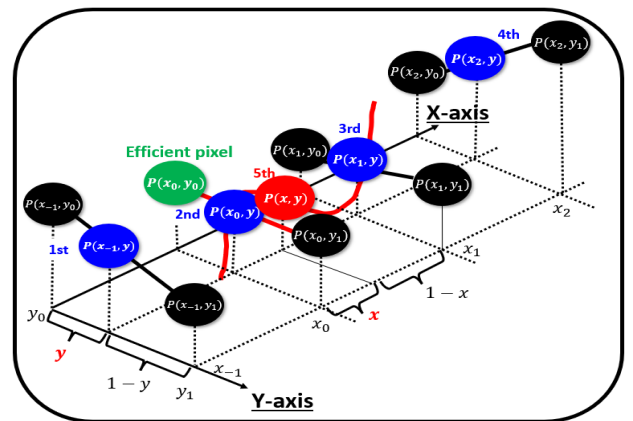


Fig. 1. Combinatorial interpolation algorithm.

그림 1. 조합보간 알고리즘

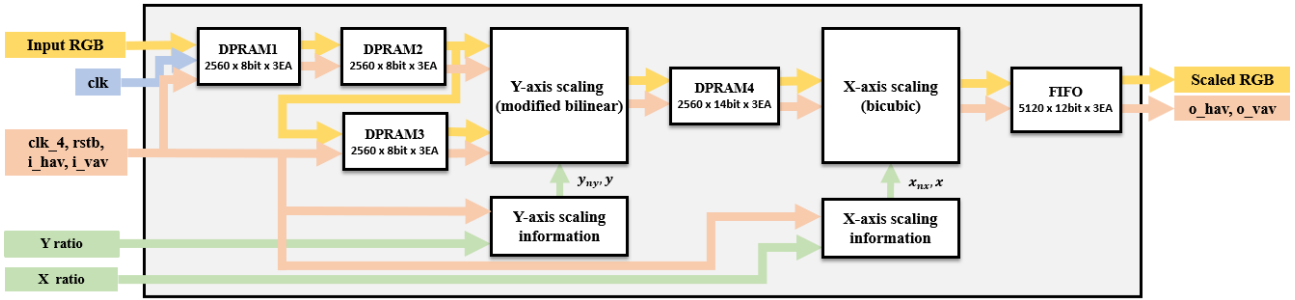


Fig. 2. Block diagram of combinatorial interpolation hardware.  
 그림 2. 조합 보간의 하드웨어 블록도

위의 수식에서  $nx = 0, 1, \dots, \text{floor}\{x \text{ ratio} \times xsize\}$ ,  $ny = 0, 1, \dots, \text{floor}\{y \text{ ratio} \times ysize\}$ 로 나타낼 수 있다.  $\text{floor}(\cdot)$  함수는 소수점을 버리는 연산을 수행하며  $xsize, ysize$ 는 입력 영상의 크기이다.

그림 1과 같이 보간이 필요한 위치  $(x, y)$ 의 픽셀값을  $P(x, y)$ , 현재 유효한 픽셀 위치  $(x_0, y_0)$ 에서의 픽셀값을  $P(x_0, y_0)$ 라 가정하자.  $P(x, y)$ 는 그림 1에서 볼 수 있듯이, 4번의 Y축 mBLI를 진행한 후 한 번의 X축 BCI를 수행하여 구할 수 있다. 먼저 Y축 mBLI는 Y축으로 인접한 라인의 픽셀 2개를 사용하며, 수식 6[4]과 같다.

$$P(x_0, y) = P(x_0, y_0) \times (1 - y) + P(x_0, y_1) \times y \quad (6)$$

수식 6에서  $P(x_0, y_1)$ 은  $P(x_0, y_0)$ 의 Y축 위치  $y_0$ 에 +1 해준  $y_1$  위치의 픽셀값이고,  $P(x_0, y)$ 는 그림 1과 같이  $x_0$ 에서의 mBLI 결과값이다. X축 BCI는 X축으로 인접한 픽셀 4개를 사용하며, 이는  $P(x_0, y_0)$ 의 X축 위치  $x_0$ 에 -1, +1, +2를 해준  $x_{-1}, x_1, x_2$ 이다. 수식 6을 이용해 해당 위치에서의 mBLI를 수행하면 그림 1과 같이  $P(x_{-1}, y), P(x_0, y), P(x_1, y), P(x_2, y)$ 를 구할 수 있다. 이 픽셀값들과 수식 7~8[4]을 사용하여 X축 BCI 연산을 수행하면 최종 보간이 필요한 위치인  $(x, y)$ 에서의 픽셀값  $P(x, y)$ 를 구할 수 있다.

$$P(x, y) = ax^3 + bx^2 + cx + d \quad (7)$$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} -0.5 & 1.75 & -1.75 & 0.5 \\ 1 & -2.5 & 2 & -0.5 \\ -0.5 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P(x_{-1}, y) \\ P(x_0, y) \\ P(x_1, y) \\ P(x_2, y) \end{bmatrix} \quad (8)$$

2. 하드웨어 설계

본 논문에서 설계한 하드웨어는 X축, Y축 방향으로 각각 0.1~2배까지 비대칭적으로 해상도 변환이 가능하게 설계하였다.

가. Overall block diagram

그림 2는 dual-clock을 사용하는 DPRAM구조[5]로 설계한 CI 알고리즘의 하드웨어 블록도 이다. 입력 RGB 영상의 픽셀값은 input horizontal active video(i\_hav), input vertical active video(i\_vav)에 따라 영상의 왼쪽 상단에서 오른쪽으로 이동하며 들어온다. 본 논문에서 설계한 하드웨어는 X, Y축 방향으로 각각 최대 2배 확대까지 지원하므로 입력보다 최대 4배 확대된 데이터가 생성될 수 있다. 따라서 입력 주파수 클록인 clk보다 4배 빠른 clk\_4로 회로의 주파수 도메인을 변경해 주기 위해 DPRAM1[5]을 사용한다. 그런 다음, Y축 최대 2배 확대 시 동일한 라인이 최대 2번까지 사용되기에 DPRAM2 [5]를 사용한다. 또한 Y축 mBLI는 인접한 라인의 픽셀을 2개 사용하기 때문에 입력을 한 라인 지연시키기는 용도로 DPRAM3을 사용한다. 한편, Y축 mBLI는 Y축 유효 위치  $y_{nx}$ 와 보간이 필요한 위치  $y$  값이 필요하며, 이는 Y-axis scaling information 블록에서 연산한다. 해당 블록에서 구한  $y_{nx}, y$ 와 DPRAM2~3을 이용해 Y-axis scaling 블록에서 mBLI를 수행한다. mBLI 결과는 X축 최대 2배 확대 시 동일한 픽셀값이 최대 2번 사용되기에 DPRAM4에 저장한다. X축 또한 Y축과 마찬가지로 X-axis scaling information 블록을 통해  $x_{nx}, x$ 를 구한 후 X-axis scaling 블록에서 BCI를 진행한다. 아래 그림 3은 앞에서 설명한 각 DPRAM1~4의 타이밍 다이어그램을 나타낸 것이다.

마지막으로 연산이 끝나면 추후 외부 다른 모듈에서 결과 영상을 사용하기에 용이하도록 first-in, first-out (FIFO) 메모리 소자를 추가해 주었다[5]. FIFO를 통해 output horizontal active video(o\_hav), output vertical active video(o\_vav)의 동기에 맞춰 최종 스케일 된 데이터가 출력된다. 그림 2에 표시된 DPRAM들의 크기는 입력이 QHD(2560×1440)이고 8 bits 데이터라

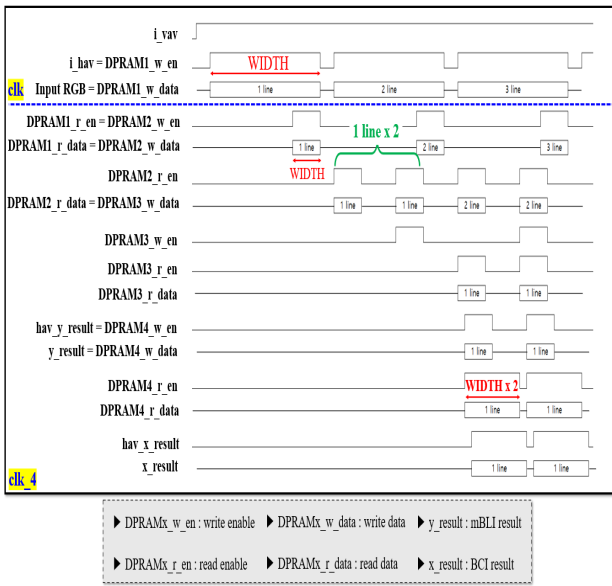


Fig. 3. Timing diagram of DPRAM.  
그림 3. DPRAM의 타이밍 다이어그램

가정한 후 설정하였다. 그림 2의 Y축, X축 연산 결과 비트 수와 이후 설명할 하드웨어 구조의 비트 설정 기준은 본 논문의 3장 2절에서 설명할 예정이다.

나. Y-axis, X-axis scaling information

그림 4는 그림 2의 Y-axis scaling information으로 수식 2, 3, 5를 하드웨어로 구현한 것이다. 먼저, 입력인 Y ratio는 0.1~2배를 지원하도록 3.10 비트(정수부 3비트, 소수부 10비트)로 설정해 주었다. 입력된 Y ratio 통해  $i_{yinc}$ 를 계산하고 이는 수식 2의 나눗셈 항이다. Y축 증가 값인  $yinc$ 는 수식 3과 같이 초깃값을 따로 계산해야 한다. 따라서  $i_{yinc}$ 를 이용해 Y start calc.에서 초깃값인  $ystart$ 를 계산해 준다. 계산된  $i_{yinc}$ ,  $ystart$ 와 0중 Multiplexer를 통해 선택된 값이 Y축 방향으로의 증가 값인  $yinc$ 가 된다. 수식 2와 같이  $yinc$ 는 이전 값을 계속 더해주는 형태이기에 누산기를 사용해 구현할 수 있다. 이때 유효 픽셀 위치  $y_{nx}$ 는 QHD 입력 기준 1440까지 가능하기에 필요한 정수부 비트 수는 11비트가 된다. 따라서  $yinc$ 는 최대 11.10비트가 되기에 11.10비트 크기의 누산기가 필요하다. 11.10비트(총 21비트)와 같이 큰 비트 수 연산은 하드웨어 구현 시 동작 속도를 저하하는 주요 원인이 된다. 따라서 이를 방지하기 위해 그림 4와 같이 정수부, 소수부 비트를 각각 분할하여 누산기를 적용함으로써 기존보다 적은 비트 수 연산이 가능하게끔 설계하였다.

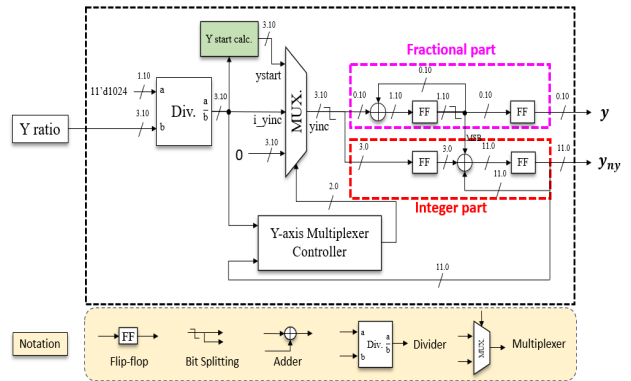


Fig. 4. Y-axis scaling information.  
그림 4. Y축 스케일링 정보

아래 그림 5는 그림 2의 X-axis scaling information으로 수식 1, 4를 하드웨어로 구현한 것이다. 그림 4와 거의 동일하나 초깃값 연산이 빠지고, 유효 픽셀 위치는 QHD 기준 2560까지 가능하기에 정수부가 1비트 증가한 12비트이다.

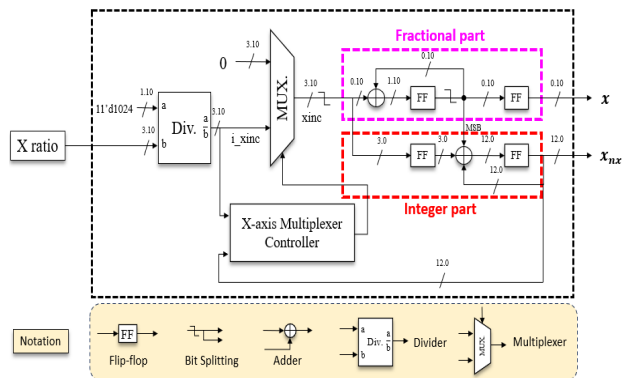


Fig. 5. X-axis scaling information.  
그림 5. X축 스케일링 정보

다. Y-axis, X-axis scaling

그림 2의 Y-axis scaling, X-axis scaling을 구현하기 위해 수식 6~8을 보면 곱셈 연산이 많은 것을 알 수 있다. 하드웨어 구현 시 큰 비트 수 간의 곱셈을 그대로 구현하면 속도 저하가 발생할 수 있다. 따라서 본 논문에서는 입력 비트 수를 기존보다 더 작은 비트 수로 분할한 후 곱셈을 진행하는 split multiplier를 사용하였다 [7]. 입력보다 더 작은 비트끼리 곱셈을 진행하기에 속도 저하 문제가 해소될 수 있지만, 플립플롭과 덧셈기가 추가되는 등 하드웨어가 커지는 단점이 있을 수 있다. Y축 mBLI의 경우 수식 6과 같이 일차 함수이기에 비교적 곱셈기의 부담이 작다. 따라서 입력 비트 수를 2개로 분할

후 곱셈을 수행하는 2-split multiplier를 사용하였고, 그림 6은 이를 구현한 것이다. X축 BCI의 경우 수식 7과 같이 3차 함수를 이용하기에 곱셈기 부담이 커진다. 따라서 입력 비트 수를 4개로 분할 후 곱셈을 진행하는 4-split multiplier를 사용하였으며, 하드웨어 구현 원리는 2-split multiplier와 동일하다.

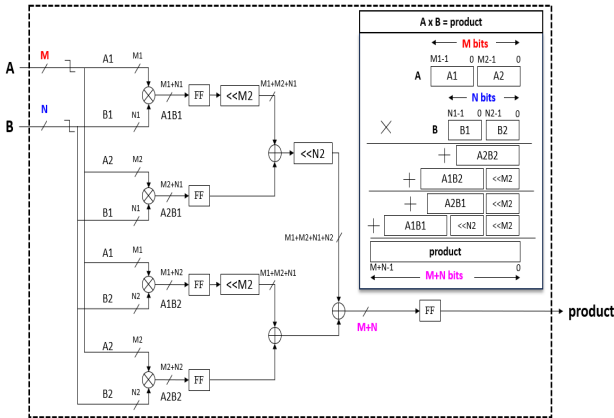


Fig. 6. 2-split multiplier.  
그림 6. 2-split 곱셈기

### III. 하드웨어 합성 결과 및 성능 평가

#### 1. 하드웨어 합성 결과

본 논문에서 설계한 하드웨어는 Verilog로 작성했고, MATLAB과 ModelSim으로 기능을 검증했다. 그런 다음 Xilinx 사의 xczu7ev-ffvc1156을 Target device[8]로 하여 Filed Programmable Gate Array (FPGA) 환경에서 합성을 진행했다.

Table 1. Xilinx synthetic result.

표 1. Xilinx 합성 결과

Xilinx Vivado 2023.1			
Device	xczu7ev-ffvc 1156-2		
Logic Utilization	Avaliable	CI Scaler	
		Used	Util.
CLB Register(#)	460,800	5,150	1.12%
CLB LUT's(#)	230,400	5,635	2.51%
Block RAMs(#)	312	24	7.69%
Min. period(ns)	-	2.217	
Max. Freq.(MHz)	-	451.060	

\* The EDA tool was supported by the IC Design Education Center (IDEC), Korea.

표 1은 Xilinx Vivado 2023.1 프로그램으로 회로를 합성한 결과이다. CI 알고리즘을 하드웨어로 구현 시 CLB Register와 CLB LUT, Block RAM의 점유율은 각각 1.12%, 2.51%, 7.69%이고, 하드웨어의 최대 동작 주파수는 451.060MHz이다. 또한 실시간 처리 만족 여부를 판단하기 위해서 fps 단위의 최대 동작 속도인 maximum processing speed (MPS)를 계산해야 하고, 이는 수식 9를 이용해 구할 수 있다[7].

$$MPS = \frac{f_{max}}{(H + HB)(V + VB)} \tag{9}$$

수식 9에서  $H$ ,  $V$ 는 각각 입력 영상의 수평, 길이, 수직 길이를 의미하며 QHD 기준으로 2560, 1440이다. 또한  $HB$ ,  $VB$ 는 각각 수평 공백, 수직 공백 구간으로 해당 설계에서 모두 1로 설정해 주었다[5].  $f_{max}$ 는 최대 동작 주파수로, 표 1에서 알 수 있듯이 451.060MHz이다. 이때 표 1의 최대 동작 주파수는 출력 clk\_4를 기준으로 측정된 주파수이고, 수식 9에 대입할 나머지 파라미터값들은 입력 기준이다. 따라서 최대 동작 주파수 또한 입력 기준으로 환산해 주어야 한다. 입력 영상의 데이터가 들어오는 동작 주파수는 clk\_4보다 4배 느리기 때문에 451.060MHz를 4로 나눈 112.765 MHz를 수식 9의  $f_{max}$ 에 대입한다. 이를 이용해 MPS를 계산하면 30.56fps로, QHD 환경에서 실시간 처리 기준인 30fps 이상을 만족한다.

#### 2. 성능 평가

본 논문에서 설계한 하드웨어는 부동 소수점 표현 방식을 사용한 CI 알고리즘과 달리, 고정 소수점 표현 방식을 사용하였다[9]. 부동 소수점 표현 방식은 정밀도가 높기에 알고리즘 성능은 우수하나, 하드웨어 구현 시 크기가 매우 크고 빠른 동작이 불가능하다. 따라서 본 논문에서는 CI 알고리즘의 결과 영상과 8비트 정수부 기준  $\pm 0.1$  least significant bit 이내의 오차만 허용하도록 신호의 비트 폭을 제한하였다. 비트 폭을 제한하는 고정 소수점 표현 방식으로 하드웨어 설계를 진행하였기에 기존 CI 알고리즘과 성능 차이가 발생한다. 따라서 하드웨어의 성능을 평가하기 위해 Peak signal-to-noise ratio (PSNR) 수치를 측정했고, CI 알고리즘과 비교하였다.

수치 측정을 위한 실험 영상으로 Han이 사용한 영상 중 그림 7의 Lena 영상과 Goldhill 영상을 사용하였다[4]. PSNR은 아래 수식 10과 같이 Mean Square Error

(MSE)를 이용해 계산하며, 수치가 높을수록 성능이 좋은 것을 의미한다[10].



Fig. 7. Experimental images.  
그림 7. 실험 영상

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \quad (10)$$

아래 표 2~3은 Han이 제안한 CI 알고리즘과 본 논문에서 설계한 하드웨어에 1.3배 확대 후 축소, 축소 후 확대를 적용한 후 PSNR을 측정하는 것이다. 수치 차이는 모두 1dB 이내로 무시할 수 있을 만한 수준으로 나타났다.

Table 2. Comparison of PSNR for up scaling and then down scaling by a factor of 1.3.

표 2. 1.3배 확대 후 축소 PSNR 비교[dB]

Image \ Method	Han[4]	Proposed
Lena	36.01	35.73
Goldhill	39.28	38.55

Table 3. Comparison of PSNR for down scaling and then up scaling by a factor of 1.3.

표 3. 1.3배 축소 후 확대 PSNR 비교[dB]

Image \ Method	Han[4]	Proposed
Lena	33.62	33.45
Goldhill	33.71	33.65

#### IV. 결론

본 논문에서는 Han이 제안한 CI 알고리즘을 dual-clock을 사용하는 DPRAM 구조로 하드웨어 구현을 진행했다. 제안하는 하드웨어는 Verilog로 작성했고, 회로의 속도 향상을 위해 큰 비트 수를 작은 비트 수로 분할하여 연산하는 기법을 사용했다. 설계한 하드웨어는 Xilinx Vivado 2023.1을 통해 FPGA 환경에서 합성을

진행하였으며, 회로의 최대 동작 주파수는 451.060MHz로 측정되었다. 이때 최대 동작 주파수는 입력 클럭보다 4배 빠른 출력 클럭 기준이기에 fps 수치 연산을 위해 입력 클럭 기준으로 바꿔주었다. 따라서 입력 기준 최대 동작 주파수는 451.060MHz를 4로 나눈 112.765MHz가 된다. 이를 통해 MPS를 계산하면 QHD(2560×1440) 환경에서 30.56fps로 실시간 처리 기준인 30fps 이상을 만족하였다. 또한 CI 알고리즘을 고정 소수점 표현 방식으로 변환 후 설계를 진행하여 성능 평가를 위해서 PSNR을 측정했다. CI 알고리즘과 PSNR 차이는 1dB 이내로 무시할 수 있을 만한 수준으로 나타났다. 본 논문에서 제시한 하드웨어 구조는 하나의 모듈에서 비대칭으로 확대 및 축소가 가능하므로, 이를 바탕으로 성능이 더 우수한 하드웨어를 설계할 수 있을 것으로 기대된다.

#### References

- [1] R. Crane, *A Simplified Approach to Image Processing*, Prentice Hall, 1997.
- [2] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing using MATLAB*, Prentice Hall, 2004.
- [3] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans.*, vol.29, no.6, 1981. DOI: 10.1109/TASSP.1981.1163711.
- [4] S. Y. Han and B. S. Kang, "A Study of the Combinatorial Interpolation Algorithm for Scaler Hardware Design," *Institute of Korean Electrical and Electronics Engineers*, vol.27, no.3, 2023. DOI: 10.7471/ikeee.2023.27.3.296.
- [5] S. Y. Han, S. Jung, and B. Kang, "Analysis of the Image Processing Speed by Line-Memory Type," *Institute of Korean Electrical and Electronics Engineers*, vol.27, no.4, 2023. DOI: 10.7471/ikeee.2023.27.4.494.
- [6] K. Jack, *Video Demystified*, Elsevier Science, 2005.
- [7] D. Ngo and B. Kang, "Taylor-Series-Based Reconfigurability of Gamma Correction in Hardware Designs," *Electronics*, vol.10, no.16, 2021. DOI: 10.3390/electronics10161959.
- [8] Xilinx, "ZCU106 Board User Guide," <https://docs.xilinx.com/v/u/en-US/ug1244-zcu106-eval-bd>

[9] Sen M. Kuo, Bob H. Lee, and Wenshun Tian, *Real-Time Digital Signal Processing*, Wiley, 2013

[10] MathWorks, "PSNR," <https://kr.mathworks.com/help/vision/ref/psnr.html>

---

## BIOGRAPHY

---

### Si-Yeon Han (Member)



2023 : BS degree in Electronics Engineering, Dong-A University.  
2023~Present : MS degree course in Electronics Engineering, Dong-A University.

### Semin Jung (Member)



2023 : BS degree in Electronics Engineering, Dong-A University.  
2023~Present : MS degree course in Electronics Engineering, Dong-A University.

### Jeong-Hyeon Son (Member)



2019~Present : BS degree course in Electronics Engineering, Dong-A University.

### Jae-Seong Lee (Member)



2019~Present : BS degree course in Electronics Engineering, Dong-A University.

### Bongsoon Kang (Member)



1985 : BS degree in Electronics Engineering, Yonsei University.  
1987 : MS degree in Electrical Engineering, University of Pennsylvania.  
1990 : PhD degree in Electrical Engineering, Drexel University.

1989~1999 : Senior Staff Researcher, Samsung Electronics.

1999~Present : Prof. of Dept. of Electronic Engineering, Dong-A University.