JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 21, No. 4 : 781~788 Apr. 2017

다중 디지털 신호의 비교를 위한 병렬 기법의 VLSI 설계

서영호[1] · 이용석[2] · 김동욱[3*]

VLSI Design of Parallel Scheme for Comparison of Multiple Digital Signals

Young-Ho Seo[1] · Yong-Seok Lee[2] · Dong-Wook Kim[3*]

[1]Ingenium College of Liberal Arts, Kwangwoon University, Seoul 01897, Korea
[2]Korea Electronics Technology Institute, Gyeonggi-do 13509, Korea
[3*]Department of Electronic Materials Engineering, Kwangwoon University, Seoul 01897, Korea

요 약

본 논문에서는 여러 디지털 신호의 크기를 비교하기 위한 알고리즘 및 디지털 회로를 제안한다. 제안하고자 하는 알고리즘은 여러 입력을 동시에 비교한 후에 간단한 디지털 논리 함수를 이용하여 그 입력들 중에서 가장 큰 값(혹은 가장 작은 값)을 검출하는 방법을 제공할 수 있다. 이 방식의 단점은 하드웨어 자원이 증가하는 것인데, 이를 위해 중복된 논리동작을 재사용하는 방법을 제안한다. 제안하고자 하는 방식은 회로 속도의 증가, 즉 지연시간의 감소에 초점을 맞추었다. 제안한 비교 알고리즘은 HDL로 구현한 후에 Altera사의 Cyclone III EP3C40F324A7 FPGA 환경에서 실험하였다. 4입력의 경우에 1.20배의 하드웨어 자원을 사용하면서 1.66배 만큼 동작 속도를 증가시킬 수 있다. 또한 8입력의 경우에는 2.15배의 하드웨어 자원을 사용하면서 2.29배로 동작 속도를 증가시킬 수 있다.

ABSTRACT

This paper proposes a new algorithm for comparing amplitude between multiple digital input signals and its digital logic architecture. After simultaneously comparing multiple inputs, the proposed algorithm can provide the information of the largest (or smallest) value among them by using a simple digital logic function. The drawback of the method is to increase hardware resource. To overcome this we propose a reuse method of the overlapped logic operation. The proposed method focuses on enhancing the operational clock frequency, in other words decreasing combinational delay time. After implementing the comparing method with HDL (hardware description language), we experiment on it with environment of Cyclone III EP3C40F324A7 FPGA of Altera Inc. In case of 4 input signals, it can increase the operational speed as mush as 1.66 times with 1.20 times the hardware resource. In case of 8, it can also have 2.29 times the clock frequency and 2.15 times the hardware resource.

키워드 : 비교기, VLSI, 상위 수준, 논리 회로, 하드웨어 설계, 다중입력

Key word : comparator, VLSI, high-level, logic circuit, hardware design, multiple input

Received 04 November 2016, **Revised** 23 November 2016, **Accepted** 12 December 2016

* **Corresponding Author** Dong-Wook Kim(E-mail:dwkim@kw.ac.kr, Tel:+82-2-940-5167)
Department of Electronic Materials Engineering, Kwangwoon University, Seoul 01897, Korea

Open Access https://doi.org/10.6109/jkiice.2017.21.4.781 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/li-censes/ 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

비교기는 두 개의 숫자의 크기를 비교하는데 사용되는 기본적이고 본질적인 연산 유닛으로 회로에서 매우 중요한 조합회로 요소이다[1,2]. 정렬 연산은 본질적으로는 많은 양의 데이터를 오름차순 혹은 내림차순으로 정렬하기 위해서 수많은 비교 연산을 수행한다[3]. 특히 비교 함수는 디지털 신호 처리와 영상 및 음성처리와 같은 영역들에서 많이 사용된다[4,5]. 예를 들어 영상처리 분야에서 객체 인식 알고리즘에서 방대한 비교 연산을 사용한다[6]. 또한 3D 그래픽스와 같은 데이터 집중적인 응용분야에서 매우 많이 사용된다[7]. 비교기는 병렬 컴퓨팅[8], 다중 접근 메모리[9] 및 다중처리와 같은 영역에서 중요한 역할을 하는 정렬 네트워크에도 사용된다[10]. 일반적인 범용 프로세서뿐만 아니라 비전 프로세서[6] 및 미디어 프로세서[11]와 같은 특수 목적의 프로세서도 많은 비교 명령어를 포함하고 있다.

지금까지의 디지털 비교기에 대한 연구를 2가지 측면에서 고려해 보고자 한다. 첫 번째는 비교기의 구현 수준이다. 여기에는 트랜지스터 수준 혹은 레이아웃 수준과 같은 하위 수준에서의 구현과 논리 회로 수준 혹은 HDL 혹은 RTL(register transfer level) 수준과 같은 상위 수준에서의 구현이 있다. 두 번째는 비교기의 입력 개수이다. 많은 연구들은 주로 두 개의 입력 신호에 대한 크기를 비교하는 구조에 대한 것이고, 일부 4개 이상의 입력 신호에 대한 비교기의 알고리즘 및 구조에 대한 것이다. 본 논문은 상위 수준에서 4개 이상의 다중 입력을 비교하는 경우에 초점을 맞춘다.

전통적인 비교기는 오직 두 개의 입력만 체크할 수 있게 한다. 입력의 수가 두 개 이상일 때, 데이터들은 순차적으로 비교된다. 즉, 현재 신호는 다음 신호와 비교되고, 이는 차례차례 다음 신호와 비교된다. 이러한 비교기 네트워크는 종종 캐스케이드 구조를 사용하는데, 이 캐스케이드 구조는 연산 시간을 증가시키기 때문에 고속 시스템에 적합하지 않다. 영상 처리 분야에서, 쿼드 이진 숫자들을 동시에 비교하는 것은 종종 발생하는 상황이다[6,11]. 더욱이, 2개 이상의 입력에 대한 비교기를 구현하는 것은 향후 프로세서와 SIMD(single instruction multiple data) DSP(digital signal processing)의 발전에 많은 영향을 미칠 수 있다. 이진 비교기는 연산 논리 회로에서 중요한 블록이고, x86 명령어의 디코딩과 같은 많은 디지털 시스템에서 광대한 응용분야를 갖는다. 다중입력 다중출력(multiple-input-multiple-output, MIMO) 디코딩 알고리즘은 과도한 반복 이진 비교연산을 요구하기 때문에 차세대 통신 시스템을 위한 다중입력 다중출력 기술의 출현은 저전력 및 고성능 비교기의 필요성을 더욱 심화시켰다[12].

두 수의 크기를 비교하기 위해 비교기를 구현 하는 방법은 여러 가지가 있다. 비교기를 구현하기 위한 직접적이고 쉬운 방법들 중의 하나는 병렬 덧셈기를 사용하는 것이다. 다른 방법은 직접적으로 비교 함수를 전개하는 방정식에 기초하여 구현될 수 있다. 이런 방식들은 전통적인 방식이라 할 수 있고, 면적의 효율성과 전력 소모의 측면에서 많은 단점을 가지고 있다. 이를 극복하기 위해 지금까지 여러 연구가 진행되었다[13-25]. Wang은 all-N-transistor(ANT) 동적 CMOS 논리를 이용한 고성능 트리-구조 비교기를 제안하였고[13], 트랜지스터 수를 줄이기 위해 고팬인(high fanin) 동작 회로를 제안하였다[14]. ANT 회로를 위해 요구되는 파이프라인 때문에 단일 사이클 동작에 적합하지 않다. Huang과 Wang은 우선순위 인코딩 알고리즘을 사용하여 [13]보다 16%의 성능 향상을 보이는 단일-사이클 이단-위상(two-phase) 비교기를 제안하였다[15]. Cheng은 효율적인 구현을 위해 변형된 1의 보수 원리와 CSA (carry save adder) 방식을 연구하였다[16]. Tsui는 병렬 MSB(most significant bit) 검사 방법을 이용하여 비교기를 제안하였는데, [17]에서는 정적인 우선순위 인코더를 이용하였고, [18]에서는 MUX기반의 구조를 이용하였다. [19]에서는 고성능 및 저면적 디지털 비교기를 위한 비트단위 경쟁논리회로(bitwise competition logic, BCL)가 제안되었다. 이 회로는 산술연산을 사용하지 않고 MSB로부터 첫 번째 1의 위치를 이용하여 두 정수를 비교한다. Corsonello는 트리-기반 비교기를 제안하였는데, 비교 과정을 용이하게 하기 위해 다이나믹 맨체스터 구조를 이용한다[20,21]. Chaudhury는 2개의 수를 비교할 수 있는 디지털 이진 비교기를 제안하고 있다. 두 개의 이진수를 비교하기 위한 Look-Ahead Logic을 고안하여 비교기의 속도를 향상시켰다[22]. Sachdev는 2의 최대 스택 높이를 구성하기 위해 전-인코딩 특징을 가지는 새로운 트리 구조의 비교기를 제안한다. 이 설계는 특히 저전력을 이루기 위해 패스 트랜지스터 및/혹은 정적 로직의 구현에

적합하다[23]. 비록 많지는 않지만, 4개 이상의 입력을 비교하기 위한 비교기도 연구가 수행되었다. Hsia는 한 번에 N개의 데이터를 비교할 수 있는 비교 셀을 제안하였는데, 제안된 셀은 동적 논리 방법론에 기초한다[24]. Yoo는 빠르고 작은 면적을 갖는 64비트 네 개의 이진수를 비교기를 제안하였다. 제안한 비트-단위 비교 논리 체인(bit-wise comparing logic chain, BCLC)과 순차적 스트로브(sequential strobe, SS) 기법은 64비트 네 개의 이진 수를 1.55ns의 지연시간만에 비교할 수 있었다[25]. 최근의 연구들에서 [22]의 연구를 제외하고는 모두 하위 수준에서의 비교기를 구현한 것들이라서 본 논문의 방향과는 다소 차이를 갖는다. 또한 [24]와 [25]를 제외하고는 모두 2개의 값을 비교하는 비교기를 연구한 것이기 때문에 본 논문의 방향과 다소 차이를 갖는다.

본 논문은 상위수준, 특히 RTL(register transfer level)에서의 설계 방법론을 이용하여 4개 이상의 입력에 대한 비교를 수행할 수 있는 비교 기법 및 그에 대한 VLSI 구조를 제안하고자 한다. 본 논문은 다음과 같이 구성된다. 2장에서는 비교기와 비교기의 병렬성, 그리고 제안하는 비교 알고리즘을 나타낸다. 3장에서는 2장에서 제안한 비교 알고리즘의 VLSI 구조에 대해서 설명한다. 4장에서는 실험결과를 보이고 4장에서 결론을 맺는다.

Ⅱ. 크기 비교 알고리즘

논리 연산에서 가장 흔하게 사용되는 동작 중의 하나가 크기를 비교하는 동작이다. 디지털 논리 회로의 측면에서 고려할 때 이러한 동작은 크기 비교기(이하 비교기)를 사용하여 수행된다. 비교기는 두 개의 값 중에서 어느 것이 큰지 혹은 같은 지를 판단하는 회로이다. 디지털 논리 회로 이론의 논리 관계를 이용해서 크기를 비교하기 위한 동작을 구현할 수 있다[2].

2.1. 다채널과 병렬성

앞 절에서와 같이 살펴볼 수 있듯이, 비교 연산은 기본적으로 연산자가 2개이다. 만일 3개의 입력신호를 비교하거나 4개, 8개, 그리고 그 이상의 입력을 비교하기 위해서는 일반적으로 2개의 입력에 대한 비교기를 여러 단계로 나누어서 비교하는 신호의 개수를 반으로 줄여 나가면서 비교한 후에 최종적으로 가장 큰 값을 비교한다. 본 논문에서는 이 방법을 전통적인 방식이라 하고, 그림 1에 나타냈다.

본 논문에서 다루고자 하는 비교기는 입력이 여러 개일 경우(다입력)를 위한 것이다[24,25]. 전통적인 방법에서 가장 큰 문제는 입력의 개수가 증가하면 그에 비례하여 성능이 감소한다는 점이다. 또한 입력 신호의 비트수가 커지면 각 단계별 지연시간을 증가시켜 성능을 더욱 감소시킨다는 점도 고려해야 한다. 최근까지 SoC는 AMBA버스(AXI) 기반의 공통 버스 구조를 가지고 있다[26]. 공통 버스의 구조에서는 여러 마스터 및 슬레이브 중에서 어느 것이 버스를 사용할 것인지 결정해야하는 문제가 필연적으로 존재한다. 많은 SoC들은 버스 성능의 향상 및 신속한 응답을 위해서 파이프라인 단계가 적으면서 고속으로 동작할 수 있는 중재기(arbitor) 및 버스 매트릭스(bus matrix)를 원하고 있고, 그 동작의 핵심은 다입력 비교기에 있다.

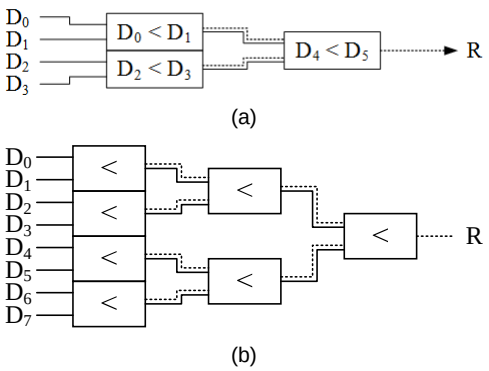


Fig. 1 Structure of multichannel digital magnitude comparator (a) 4, (b) 8 channels

전통적인 방식의 다입력 비교기의 가장 큰 문제점 중의 하나는 가장 큰 값이 여러 개일 경우에 이들을 모두 표시하기 위해서는 별도의 제어 회로를 가져야 한다는 점이다. 그림 1(b)의 구조를 살펴보면 8개의 입력들 중에서 한 개의 가장 큰 값을 찾는 것은 쉬울 것이다. 그러나 만일 2개, 혹은 4개, 혹은 7개의 값이 같다면, 어떤 값이 크기가 같고, 그 중에서 어느 것을 가지고 다음 단계에서 비교를 하는지 등에 대한 정보를 처리하는 부가적인 회로가 필요하다. 채널이 커질수록 이러한 동작은 매우 복잡하고 귀찮은 작업이 될 것이다.

2.2. 제안하는 방법

본 논문에서는 두 가지 비교 방법에 대해 논의하고자 한다. 비교방법1은 제안하는 기본적인 기본 알고리즘으로 가장 큰 값들을 모두 검출 가능하다. 검출한 가장 큰 값들을 다양한 방법으로 활용이 가능하다. 제안하는 비교 방법은 동시에 자기 자신을 제외한 나머지 입력값들과 비교를 하고 그 비교한 결과들을 그림 2와 같은 매우 간단한 논리 연산을 통해 어느 값이 가장 큰지 검출할 수 있다. 그림 2에서 사용된 함수는 아래와 같이 정의한다.

- Parallel 함수 : 함수의 첫 번째 인수가 포함된 모든 문장을 두 번째 인수의 개수만큼 병렬적으로 처리 혹은 연산 수행
- Compare 함수 : 2개의 인수를 함수 내에 정의된 방법으로 비교한 후에 논리 값 반환

다음으로 가장 큰 결과가 여러 개일 경우에 약속된 위치의 데이터를 검출할 수 있도록 한다. 여러 개의 큰 값이 입력될 경우에 가장 왼쪽에 입력된 하나의 값을 가장 큰 값으로 검출하도록 하였다. 가장 오른쪽 값을 검출하고자 할 경우에는 비교하는 방향을 바꾸면 가능하다. 그림 3에 제안하는 비교방법2의 Compare 함수에 대한 유사코드를 나타내었다.

비교방법 2에서는 “크거나 같다”와 같은 비교를 사용한다. “같다”는 비교동작은 앞서 설명한 것과 같이 $x_3x_2x_1x_0$의 논리식을 사용하고 이는 “크거나”에 해당되는 논리식과 OR 연산을 거쳐야 하므로 게이트 레벨을 한 단계 혹은 그 이상 증가시키게 된다. 이로 인해 하드웨어 자원과 지연시간이 증가한다. “크거나 같다”라는 조건은 “작다”의 반전이므로 이 특성을 이용해서 비교방법 2의 첫 번째 if문의 내용을 개선할 수 있다. 그림 4에 이와 같이 개선된 방법을 나타내었다. 본 논문에서 제안하고 있는 비교방법은 기본적으로 입력들간에 병렬 비교를 사용하기 때문에 고성능에는 매우 적합하지만 그에 따른 하드웨어 자원도 증가한다. 따라서 하드웨어 자원을 줄이기 위한 시도가 필요하다. 다행히 “크거나 같다”는 조건($A \le B$)을 “작다”는 조건($A > B$)의 결과를 반전하여 사용함으로써 중복된 중간 연산을 만들 수 있고, 이를 통해 자원을 감소시킬 수 있다. 이 경우에 먼저 사용된 비교기를 그대로 사용하고 결과만 반전시켜서 사용하면 되기 때문이다. 이에 대한 회로도는 다음 장에서 나타낸다.

```
Parallel( i, all inputs ) {
  Parallel( j, all inputs except i ) {
    R_AND,i = R_AND,i & Compare( D_i, D_j)
    R_OR,i = R_OR,i | Compare( D_i, D_j)          }
  R_AND,i = R_AND,i & R_OR,i          }
Compare( D_i, D_j) {
  if ( D_i < D_j) Return 0
  else Return 1              }
```

Fig. 2 Comparing method1

```
Compare( D_i, D_j) {
  if ( i > j ) {
    if ( D_i ≦ D_j) Return 0
    else Return 1    }
  else {
    if ( D_i < D_j) Return 0
    else Return 1          }          }
```

Fig. 3 Compare function of method2

```
Compare( D_i, D_j) {
  if ( i > j ) {
    if ( D_j < D_i) Return 1
    else Return 0          }
  else {
    if ( D_i < D_j) Return 0
    else Return 1          }          }
```

Fig. 4 Modification of method2

Ⅲ. VLSI 구조

앞 장에서 제안한 알고리즘에 대한 VLSI 구조를 그림 5에 나타냈다. 입력의 수가 4인 경우를 예시하였다. 그림 5(a)는 방법1에 대한 VLSI 구조이고, 그림 56(b)는 방법2에 대한 VLSI 구조를 나타낸다. 그림에서 실선은 데이터 버스를 의미하고, 점선은 논리 연산의 결과를 전달하기 위한 배선을 의미한다. 그림 2에서 3개의 문장은 그림 5(a)의 3레벨 구조(비교기-AND/OR-AND)에 해당한다. Compare(Di, Dj)가 첫 번째 레벨을 구성하고, RAND,i및 ROR,i가 두 번째 레벨을 구성한다. 그리고 RAND,i가 다시 세 번째 레벨을 구성한다. 이 구조를

살펴보면 채널의 개수가 많아지면 첫 번째 레벨에서 비교기의 개수가 많아진다. 그러나 비교기의 증가는 자원의 증가를 야기하지 지연시간의 증가를 가져오지는 않는다. 물론 비교기 증가로 인해 배선이 늘어나기 때문에 이로 인한 배선 지연의 증가는 발생할 것이다. 그러나 그것이 성능(임계경로)에 큰 영향을 주지는 않는다. 이와 같이 채널의 증가로 인한 첫 번째 레벨에서의 지연시간 증가는 미미하다. 다음으로 두 번째 레벨을 고려한다. 비교기의 개수가 증가하면 AND와 OR 게이트의 fan-in이 증가한다. 실제 칩 제작을 위한 라이브러리에서 프리미티브 게이트들은 입력의 개수에 제한이 있다. 따라서 AND 및 OR 게이트는 케스케이딩될 것이고, 이로 인해 지연시간은 증가한다. 이 지연시간은 채널의 증가에 따라 선형적으로 발생할 것이다. 마지막으로 세 번째 레벨은 구조를 고려할 때 채널의 증가와 무관하다. 이 회로는 채널들 중에서 최대값이 여러 개일 경우에 모든 최대값을 표시할 수 있는 기능을 갖고 있다. 부가적인 회로 혹은 특별한 알고리즘을 사용하지 않으면서도 이러한 특성을 보유하고 있다는 것은 본 회로의 매우 큰 장점이다.

비교기에 규칙적으로 “크거나 같다”라는 조건을 삽입함으로써 가장 큰 값들 중의 하나(가장 오른쪽 혹은 가장 왼쪽)을 자동으로 선택할 수 있다. 물론 이 방식도 매우 좋은 결과를 가져왔지만, 앞 장에서 설명한 바와 같이 몇 게이트 수준의 지연시간의 증가를 유발한다. 따라서 “작다”는 조건으로 치환할 수 있다. 이 구조는 채널 수의 증가에 따라 하드웨어 자원량이 너무 증가하는 단점을 계속 가지고 있다. 따라서 중복되는 비교 동작을 찾은 후에 이들을 공유하는 구조를 그림 5(b)와 같이 만들 수 있다. 동작은 동일하지만 비교기가 단지 하나의 NOT 게이트로 대체된다는 것은 하드웨어 자원량의 측면에서 매우 효율적이다. 그림 6에서 확인할 수 있는 것과 같이, 입력이 N개일 경우에 그림 5(a) 및 그림 3의 구조에서 사용된 비교기의 개수는 $N(N\text{-}1)$이다. 반면에 그림 5(b)의 구조에서 사용된 개선된 방법2의 경우에는 비교기가 $N(N\text{-}1)/2$개로 다른 구조에 비해 반만 사용된다.

본 논문에서 제안하는 다중 채널을 비교하기 위한 회로의 VLSI의 구조와 동작 방식은 매우 단순하다. 뿐만 아니라 여러 응용도 가능하기 때문에 매우 발전 가능성이 많은 회로이기도 하다. 예를 들어 4채널을 위한 비교회로는 아주 사소한 회로만 추가해주면 2개의 2채널 비교 회로로 사용할 수 있다. 즉, 8채널 비교 회로인 경우에는 2개의 4채널 혹은 4개의 2채널을 위한 비교기로 얼마든지 동시에 사용할 수 있다.

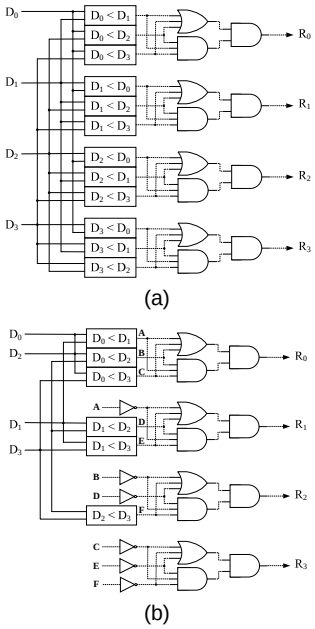


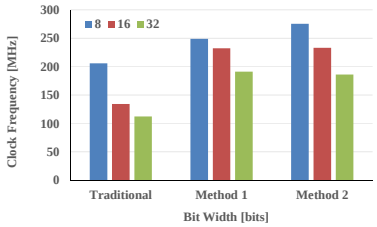
Fig. 5 VLSI architecture of the proposed comparator (a) method1, (b) method2

Ⅳ. 구현 및 결과

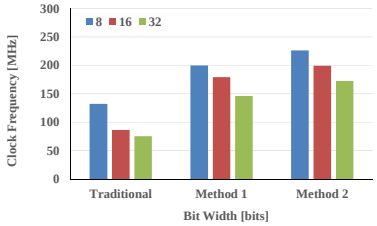
제안한 구조는 FPGA 환경에서 구현하여 특성을 검증하였다. 사용한 FPGA는 Altera사의 Cyclone III EP3C40F324A7이고, 사용한 툴은 Quatus II 9.1 버전이다. HDL을 이용하여 구현한 이후에 구현된 하드웨어의 속도와 자원량(셀 개수)를 측정하였다. 그 결과를 그림 6과 7에 나타냈다. 실험에서는 입력의 개수를 4개와 8개로 정하여 구현하였고, 입력의 비트너비는 8, 16, 32 비트로 수행하였다. 또한 전통적인 방식(Traditional), 방법1(Method 1), 그리고 방법2(Method 2)에 대해서 각각 구현하였다. 따라서 총 18가지 조건에 대해 각각 하드웨어를 HDL로 구현하고, 실험을 수행하였다.

그림 6에는 입력과 비트너비에 따른 최대 동작 주파수를 나타냈다. 이를 통해 구현된 하드웨어가 성능의

관점에서 얼마나 병렬화 효과가 있는지 확인할 수 있다. 모든 방식에서 최대 동작 주파수는 8비트의 경우가 가장 빠르고, 16비트의 경우가 가장 느리다. 또한 각 방법들을 비교하면 방법2가 가장 빠른 동작속도를 갖는다는 것을 확인할 수 있다. 4개의 입력인 경우를 살펴보면 8비트를 제외하고 16비트와 32비트는 거의 5MHz 이내로 속도의 차이가 난다.

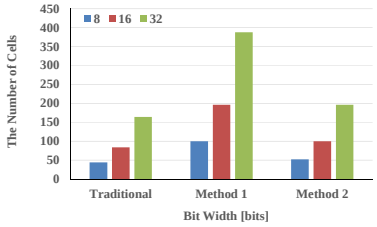


(a)

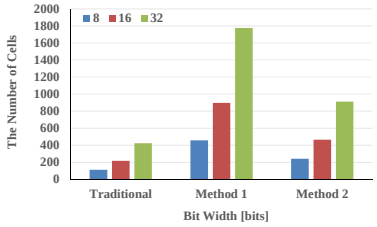


(b)

Fig. 6 Maximum clock frequency for the number of inputs and bitwidth (a) 4, (b) 8 inputs



(a)



(b)

Fig. 7 Hardware resource for the number of inputs and bitwidth (a) 4, (b) 8 inputs

즉, 동작주파수가 거의 비슷하다는 것을 확인할 수 있다. 따라서 전통적인 방식과 비교할 때 동작주파수의 증가는 54.72%와 57.80%로 거의 근소하다. 그러나 8입력의 경우에는 전통적인 방식 대비 방법1은 84%만큼 동작속도가 증가했고, 방법2는 110%만큼 동작속도가 증가했다. 전통적인 방식에 비해서 방법1은 평균 약 77MHz, 그리고 방법2는 평균 약 101MHz의 동작속도가 증가했다. 본 실험을 통해서 제안한 방법은 전통적인 방식에 비해서 매우 효과적으로 동작속도를 증가시킬 수 있다는 것을 확인하였다.

제안한 기본 방법은 병렬적으로 입력값들을 서로 비교하기 때문에 면적이 크게 증가한다. 그림 7에 면적에 대한 결과를 나타냈다. 4입력의 경우에는 전통적인 방식과 비교할 때 방법1은 평균 132% 증가하고, 방법2는 평균 19% 증가한다. 8입력의 경우에는 전통적인 방식과 비교할 때 방법1은 평균 313% 증가하고, 방법2는 평균 114% 증가한다. 4입력의 경우에 방법2는 전통적인 방식과 비교할 때 아무 미미한 자원량의 증가를 보인다. 8입력의 경우에는 방법1은 전통적인 방식의 약 4배에 해당하는 자원을 사용하고 방법2는 약 2배에 해당하는 자원을 사용한다. 8입력의 경우를 종합적으로 고려하면 방법2는 전통적인 방법에 비해서 2배의 하드웨어 자원을 사용하면서 2배만큼 동작속도를 증가시킨다. 2배의 자원을 투입해서 2배의 속도 향상을 가져왔다는 것에 대해 VLSI 설계에서 객관적인 효율성으로 이야기할 수는 없지만 상위 수준의 설계 방법론을 이용해서 속도가 중요한 응용분야를 위한 대안을 제시할 수 있다는 점은 충분히 의미가 있을 것으로 사료된다.

상위 수준 설계 방법론을 이용해서 RTL(register transfer level)에서 다수 개의 입력에 대한 비교기를 설계한 다른 연구가 없기 때문에 실험결과를 다른 연구사례와 비교하기는 어렵다는 점은 본 논문의 실험결과를 정리하는데 있어서 다소 아쉬운 부분으로 볼 수 있다.

Ⅴ. 결 론

본 논문에서는 다수 개의 디지털 신호의 크기를 비교하기 위한 알고리즘과 그에 대한 VLSI 구조를 제안하였다. 제안하고자 하는 알고리즘은 여러 입력을 동시에 비교한 후에 간단한 디지털 논리 함수를 이용하여 그 입력

들 중에서 가장 큰 값 혹은 가장 작은 값을 검출하는 방법을 제안하였다. 제안한 비교 알고리즘은 HDL로 구현한 후에 알테라사의 Cyclone III EP3C40F324A7 FPGA에 맵핑하여 실험하였다. 입력의 개수가 증가할수록 병렬적인 비교로 인한 하드웨어 자원의 사용율은 증가하지만 그에 따라 동작 주파수도 증가함을 확인할 수 있었다. 즉, 본 논문이 목표로 하는 고속의 비교기 구현이 가능하다는 것을 확인할 수 있었다. 본 논문에서 제안된 기법은 서론에서 설명한 것과 같이 다양한 응용분야에서 다수의 입력에 대한 비교를 수행할 수 있는 다양한 형태의 비교기로 사용될 수 있을 것으로 사료된다.

ACKNOWLEDGMENTS

The present Research has been conducted by the Research Grant of Kwangwoon University in 2016.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2014R1A2A1A11052433).

REFERENCES

[1] R. J. Tocci, N. Widmer, and G. Moss, *Digital Systems: Principles and Applications*, 11th ed. London, Pearson, 2010.

[2] I. Koren, *Computer Arithmetic Algorithms*, 2nd ed, Florida, A K Peters/CRC Press, 2001.

[3] K. Mehlhorn, *Sorting and Searching*, 1st ed. Springer-Verlag, Berlin, 1984.

[4] W. Alexander, C and M Williams, *Digital Signal Processing: Principles, Algorithms and System Design*, 1st ed, Massachusetts, Academic Press, 2017.

[5] J.W. Kang, "Disparity Vector Derivation Method for Texture-Video-First-Coding Modes of 3D Video Coding Standards," *Journal of The Korean Institute of Communication Sciences*, vol. 40. no. 10, pp. 2080-2089, Oct. 2015.

[6] S.B. Yoon, S.H. B, H.J. Park, and J.H. Yi, "Probabilistic Graph Based Object Category Recognition Using the Context of Object-Action Interaction," *Journal of The Korean Institute of Communication Sciences*, vol. 40, no. 11, pp. 2284-2290, Nov. 2015.

[7] H. Lee, K. Cho, H. Kim, S. Choi, J. Lim and J. Kim, "Electrical performance of high bandwidth memory (HBM) interposer channel in terabyte/s bandwidth graphics module," *2015 International 3D Systems Integration Conference (3DIC)*, Sendai, pp. TS2.2.1-TS2.2.4, 2015.

[8] M. K. Jaiswal, B. S. C. Varma, H. K. H. So, M. Balakrishnan, K. Paul, and R. C. C. Cheung, "Configurable Architectures for Multi-Mode Floating Point Adders," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 8, pp. 2079-2090, Aug. 2015.

[9] S. Abed, M. Al-Shayeji, S. Sultan, and N. Mohammad, "Hybrid approach based on partial tag comparison technique and search methods to improve cache performance," *IET Computers & Digital Techniques*, vol. 10, no. 2, pp. 69-76, Nov. 2016.

[10] M. Codish, L. Cruz-Filipe, M. Frank, and P. Schneider-Kamp, "Twenty-Five Comparators Is Optimal When Sorting Nine Inputs (and Twenty-Nine for Ten)," *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, Limassol, pp. 186-193, 2014.

[11] R. Woo, S. Choi, J.H. Sohn, S.J. Song, and H.J. Yoo, "A Low-Power 3-D Rendering Engine With Two Texture Units and 29-Mb Embedded DRAM for 3G Multimedia Terminals," *IEEE Journal of Solid-State Circuits*, vol. 39, no.7, pp. 1101-1109, Jul. 2004.

[12] S. S. Ameer Abbas, S. J. Thiruvengadam, and N. A. R. Kumar, "Realization of receiver architectures using VLSI DSP techniques for broadcast channel in LTE," *2014 International Conference on Embedded Systems (ICES)*, Coimbatore, pp. 12-17, 2014.

[13] C.C. Wang, C.F. Wu, and K.C. Tsai, "1 GHz 64-Bit high-speed comparator using ANT dynamic logic with two-phase clocking," *IEE Proceedings - Computers and Digital Techniques.*, vol. 145, no. 6, pp. 433–436, Nov. 1998.

[14] C.C. Wang, P.M. Lee, C.F. Wu, and H.L. Wu, "High Fan-In Dynamic CMOS Comparators With Low Transistor Count," *IEEE Transactions on Circuits and Systems I : Fundamental Theory and Applications*, vol. 50, issue 9, pp. 12161-220, Sep. 2003.

[15] C.H. Huang and J.S. Wang, "High-performance and power-efficient CMOS comparators," *IEEE Journal of*

Solid-State Circuits, vol. 38, no. 2, pp. 254-262, Feb. 2003.

[16] S.W. Cheng, "A High-Speed Magnitude Comparator With Small Transistor Count," *IEEE Proceedings of International Conference on Electronics, Circuits and Systems*, vol. 3, pp. 1168-1171, Dec. 2003.

[17] H.M. Lam and C.Y. Tsui, "High-performance single clock cycle CMOS comparator," *Electron. Letters.*, vol. 42, no. 2, pp. 75-77, Jan. 2006.

[18] H.M. Lam and C.Y. Tsui, "A MUX-based high-performance single cycle CMOS comparator," *IEEE Transactions on Circuits and Systems. II, Exp. Briefs*, vol. 54, no. 7, pp. 591-595, Jul. 2007.

[19] J.Y. Kim and H.J. Yoo, "Bitwise Competition Logic for Compact Digital Comparator," *IEEE Asian Solid Stated Circuits Conference*, Jeju, pp. 59-62, Nov. 2007.

[20] S. Perri, and P. Corsonello, "Fast low-cost implementation of single-clock cycle binary comparator," *IEEE Transactions on Circuits and Systems II, Exp. Briefs*, vol. 55, no. 12, pp. 1239- 1243, Dec. 2008.

[21] F. Frustaci, S. Perri, M. Lanuzza, and P. Corsonello, "A new low-power high-speed single-clock-cycle binary comparator," *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, Paris, pp. 317-320, May 2010.

[22] S. Deb and S. Chaudhury, "High-speed comparator architectures for fast binary comparison," *2012 Third International Conference on Emerging Applications of Information Technology (EAIT)*, Kolkata, pp. 454-457, Nov. 2012.

[23] P. Chuang, D. Li, and M. Sachdev, "A Low-Power High-Performance Single-Cycle Tree-Based 64-Bit Binary Comparator," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 2, pp. 108-112, Feb. 2012.

[24] S.C. Hsia, "High-speed multi-input comparator," *IEE Proceedings - Circuits, Devices and Systems*, vol. 152, no. 3, pp. 210-214, Jun. 2005.

[25] M. Kim, J. Y. Kim, and H. J. Yoo, "A 1.55ns 0.015 mm2 64-bit quad number comparator," *2009 International Symposium on VLSI Design, Automation and Test*, Hsinchu, pp. 283-286, Apr. 2009.

[26] ARM information center. AMBA [Internet]. Available: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.set.amba/index.html.

서영호(Young-Ho Seo)

1999년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
2001년 2월 : 광운대학교 일반대학원 졸업(공학석사)
2004년 8월 : 광운대학교 일반대학원 졸업(공학박사)
2008년 3월 ~ 현재 : 광운대학교 인제니움학부대학 정교수
※관심분야 : 실감미디어, 2D/3D 영상 신호처리, 디지털 홀로그램, SoC 설계

이용석(Yong-Seok Lee)

2015년 2월 : 광운대학교 전자재료공학과 졸업 (공학사)
2017년 2월 : 광운대학교 일반대학원 졸업(공학석사)
2017년 3월 ~ 현재 : 전자부품연구원(KETI) 연구원
※관심분야: 실감미디어, 2D/3D영상 신호처리

김동욱(Dong-Wook Kim)

1983년 2월 한양대학교 전자공학과 졸업(공학사)
1985년 2월 한양대학교 공학석사
1991년 9월 Georgia공과대학 전기공학과(공학박사)
1992년 3월 ~ 현재 광운대학교 전자재료공학과 정교수
※관심분야 : 3D 영상처리, 디지털 홀로그램, 디지털 VLSI Testability, VLSI CAD, DSP설계