

비트 레벨 슈퍼 시스톨릭 정렬 어레이 구현

이재진, 한강룡, 김용규, 송기용
충북대학교 컴퓨터공학과

Implementation of Bit-level Super-Systolic Array for Sorting

Jae-Jin Lee, Kang-Yong Han, Yong-Kyu Kim, Gi-Yong Song
{ceicarus, kyhan, ykkim}@archi.chungbuk.ac.kr, gysong@chungbuk.ac.kr
Dept. of Computer Engineering, Chungbuk National University

요 약

어레이 셀 내의 연산에 대한 고성능 처리는 시스톨릭 어레이의 중요한 특징이다. 본 논문에서는 시스톨릭 어레이 구조 내 셀이 또 다른 시스톨릭 어레이 구조를 가지는 슈퍼 시스톨릭 어레이 구조를 제안하고, 그 예로 비트 레벨 슈퍼 시스톨릭 정렬기의 설계 및 구현에 대하여 기술한다. 먼저 정규순환방정식으로 표현된 정렬 알고리즘으로부터 워드 레벨 1차원 평면 시스톨릭 어레이를 유도한 후 유도된 워드 레벨 시스톨릭 어레이를 슈퍼 시스톨릭 어레이로 변환한다. 위의 과정으로 유도된 비트 레벨 슈퍼 시스톨릭 정렬기를 RT수준에서 VHDL로 모델링 하여 동작을 검증하였으며, 검증된 비트 레벨 슈퍼 시스톨릭 정렬기는 Hynix에서 제공되는 0.35 μ m 셀 라이브러리와 FPGA V200E칩을 사용하여 합성 및 구현되었다.

I. 서론

정렬(sorting)은 데이터 처리의 가장 기본이 되는 연산으로 컴퓨터를 이용하는 응용분야에 널리 활용되고 있으며 성능향상을 위한 기본구조 설정 등을 포함하는 많은 연구가 진행되고 있다.

VLSI 기술의 발전은 빠른 계산이 필요하지만, 처리속도가 부응하지 못하는 문제들에 대해 시스톨릭 어레이 형태로 병렬성을 적용하는 것을 실제로 구현 가능하게 하고 있다[1].

본 논문에서는 어레이 셀 내에 대한 동시성 처리를 높이기 위해 시스톨릭 어레이 구조 내 셀이 또 다른 시스톨릭 어레이 구조를 가지는 슈퍼 시스톨릭 어레이[2] 구조를 제안하고, 그 예로 비트 레벨 슈퍼 시스톨릭 정렬기의 설계 및 구현에 대하여 기술한다.

시스톨릭 어레이는 규칙적인 데이터의 흐름을 위해 셀 내의 모든 연산들이 단위 클럭 안에 수행되어야 한다고 가정한다. 그러나, 서로 다른 연산에 대해 요구되는 지연의 차이는 무시될 수 없으며 연산 중에서 가장 큰 지연을 가지는 연산자가 시스톨릭 어레이의 단위 클럭을 결정하게 되고 이 것은 궁극적으로 시스톨릭 어레이의 성능을 떨어뜨린다. 이런 문제를 해결하기 위해 큰 지연을 가지는 셀 내의 연산자를 다시 시스톨릭 어레이로 구성하는 것이 가능하며, 이 과정은 셀이 기본적인 연산자(primitive operator)로 구성될 때까지 계속 될 수 있다.

먼저 정규순환방정식으로 표현된 정렬 알고리즘으로부터 워드 레벨 일차원 평면 시스톨릭 어레이를 유도한 후

유도된 워드 레벨 시스톨릭 어레이를 슈퍼 시스톨릭 어레이로 변환한다. 위의 과정으로 유도된 비트 레벨 슈퍼 시스톨릭 정렬기를 RT수준에서 VHDL로 모델링 하여 동작을 검증하였으며, 검증된 비트 레벨 슈퍼 시스톨릭 정렬기는 Hynix에서 제공되는 0.35 μ m 셀 라이브러리와 FPGA V200E칩을 사용하여 합성 및 구현되었다.

II. 시스톨릭 어레이

시스톨릭 어레이[1][3]는 파이프라인 어레이 구조의 새로운 범주으로써 지역적으로 연결된 프로세서들이 규칙적으로 데이터를 계산하고 시스템을 통해 데이터를 전달하는 프로세서 네트워크라고 정의된다.[4] 규칙적이고, 지역적인 데이터의 흐름을 갖으며 계산량이 많은(computation-bound) 알고리즘에 적용될 수 있는 시스톨릭 어레이는 1980년대부터 연구되기 시작하였으며, 현재는 의존성 분석과 선형 순환 반복의 시공간 또는 정규순환 방정식의 변형으로부터 자동적으로 시스톨릭 어레이 구조를 유도해 내는 연구가 진행되고 있다.

시스톨릭 어레이 디자인은 고도의 파이프라인 처리라는 점에서 폰노이만 컴퓨터와 다르다. 한 번 메모리로부터 읽어온 데이터는 하나의 PE(Processing Element)에서 사용된 후 어레이를 따라 다른 PE로 전달되어 각각의 PE에서 효과적으로 사용되며, 반면에 반복적인 방법으로 각 데이터에 대하여 다양한 연산이 수행될 수 있다. 시스톨릭 어레이의 이런 특징은 계산량이 많은 알고리즘 처리에 효과적이며, 폰노이만 컴퓨터에서 일반적으로 발생

하는 고전적인 메모리 접근 병목현상을 피할 수 있게 한다. 시스틀릭 어레이의 기본적인 구성을 그림 3에 보인다.

시스틀릭 어레이의 특징은 동시성, 모듈성과 확장성, 공간적 지역성, 시간적 지역성이라 할 수 있다. 전역 클럭에 맞추어 동시에 각각의 PE에서 데이터가 규칙적이고 반복적으로 계산되고 네트워크를 통해 전달된다는 면에서 동시성, 동질의 상호연결을 가진 독자적인 기능을 수행하는 PE로 구성되어 있으며 이러한 구성에 기인하여 무한한 확장이 가능하다는 점에서 모듈성과 확장성, 또한 커뮤니케이션 가능한 상호연결 구조를 가진다는 점과 각 PE들 사이의 데이터 전달 과정에서 단위시간 이상의 지연이 존재한다는 점에서 공간적, 시간적 지역성을 갖는다.

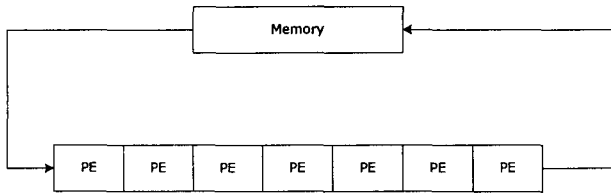


그림 3. 시스틀릭 어레이의 기본 구조

III. 비트 레블 슈퍼 시스틀릭 어레이

3.1 정렬을 위한 워드 레블 시스틀릭 어레이

정렬문제는 주어진 수열 $\{x(i)\}$ 에 대해 새로운 순열 $\{m(i)\}$ 를 얻는 것이다. 여기서 $\{m(i)\}$ 는 주어진 수열 $\{x(i)\}$ 의 원소들로 구성되어 있지만, 내림차수 순 (decreasing order) 또는 오름차수 순 (ascending order)으로 재배열된 순열이다.

정렬 알고리즘은 쉽게 지역적 순환 방정식으로 공식화되며, 선택정렬 알고리즘에 대한 정규순환 방정식은 아래와 같이 표현된다 [4].

```

FOR i from 1 to N
  ( N : Number of {x(i)} )
  FOR j from 1 to i
    m[i+1][j] = max (x[i][j], m[i][j])
    x[i][j+1] = min (x[i][j], m[i][j])
    
```

위의 정규순환방정식으로부터 얻어진 DG는 그림 4에 보인다. $x(i, 1)$ 은 주어진 순열 $\{x(i)\}$ 로 초기화되며, $m(i, i)$ 는 $-\infty$ 값으로 정해진다. 결과는 가장 오른쪽 노드, $\{m(N, j), j = 1, 2, \dots, N\}$,에서 출력되며 정렬순서는 내림차수 순이다.

그림 4의 DG에서 프로젝션을 ij -방향, $[1 \ 1]$, 으로 하고 디폴트 스케줄을 사용하면 그림 4와 같은 SFG가 얻어진다. 이때 얻어진 SFG는 버블정렬기와 일치하는 구조를 가지게 되며, 시스틀릭 어레이가 가져야 하는 특징인 동시성, 모듈성과 확장성, 공간적 지역성, 시간적 지역성을

가지므로 시스틀릭화 과정없이 바로 시스틀릭 어레이로 사용될 수 있다. 결과는 모든 노드에서 출력되며 변수 m 과 x 의 데이터 흐름은 서로 반대이다.

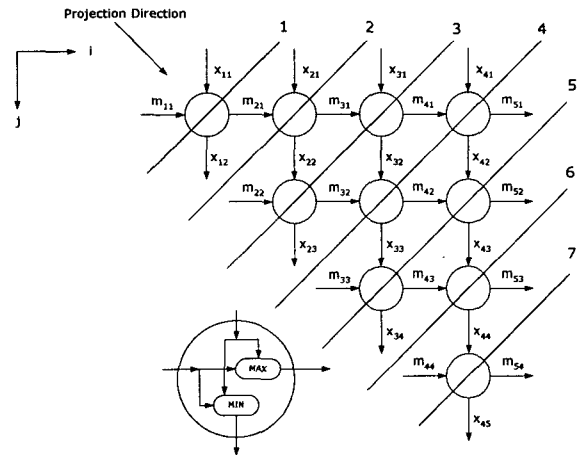


그림 4. 정렬을 위한 DG

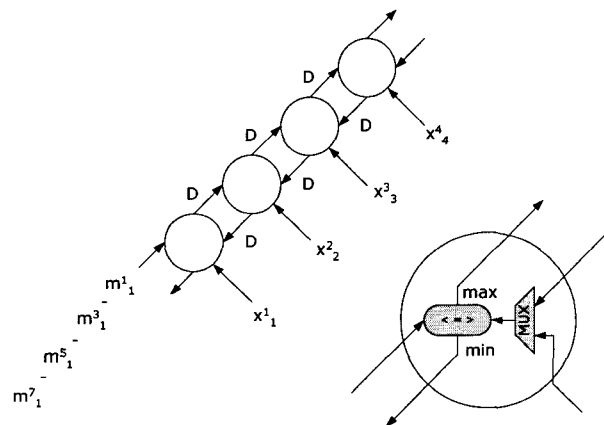


그림 5. 정렬을 위한 워드 레블 시스틀릭 어레이

그림 4에 보이는 것처럼 정렬을 위한 워드 레블 시스틀릭 어레이 셀은 비교기(comparator)를 포함하고 있다. 집적도가 높은 시스템에서 면적의 제한은 매우 중요한 요소이며, 비교기와 같이 많은 면적과 지연을 차지하는 연산자는 다시 시스틀릭 어레이를 이용하여 구현될 수 있는 후보 연산자이며 이것은 원 시스틀릭 어레이를 슈퍼 시스틀릭 어레이로 만든다.

3.2 비교 연산을 위한 비트 레블 시스틀릭 어레이

비교연산은 두 입력의 대소 관계를 판단하는 기능을 하는 연산으로 일반적으로 많은 면적과 큰 지연을 차지한다. 4-비트 두 입력 m, x 의 비교 연산에 대한 DG는 그림 4에 보이며 내부 연산은 표 4에 보인다. 그림 4에 보이는 것처럼 비교는 비트 단위로 이루어지며 두 입력

의 MSB부터 비교를 하기 때문에 대소의 판단은 가장 빠른 경우 첫 번째 노드에서 이루어 질 수 있으며 최악의 경우 마지막 노드에서 이루어진다. 그림 7의 DG를 j -방향으로 프로젝션 해서 얻어진 비교 연산을 위한 비트 레블 시스틀릭 어레이를 그림 8에 보인다. 이 때 얻어진 시스틀릭 어레이는 하나의 셀로 구성되며 셀 내부의 연산은 표 3에 보인다.

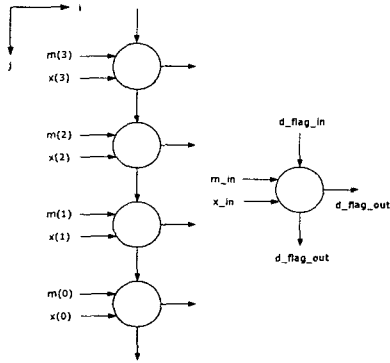


그림 6. 비트 레블 비교 연산을 위한 DG

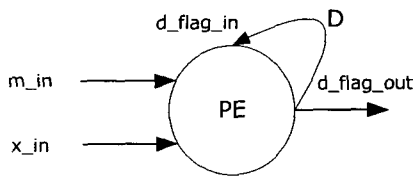


그림 7. 비교 연산을 위한 비트 레블 시스틀릭 어레이

표 3. 시스틀릭 어레이 셀의 내부 연산

d_flag_in	m_in	x_in	d_flag_out
00	0	0	00
	0	1	01
	1	0	10
	1	1	00
01	x	x	01
10	x	x	10

3.3 정렬을 위한 비트 레블 슈퍼 시스틀릭 어레이

3.1절에서 얻어진 정렬을 위한 워드 레블 시스틀릭 어레이의 셀 내부의 비교 연산을 3.2절에서 얻어진 비교연산을 위한 비트 레블 시스틀릭 어레이로 대체 한 후 데이터 흐름의 동기를 위한 지연을 추가하면 N 이 3인 경우 그림 9과 같은 정렬을 위한 비트 레블 슈퍼 시스틀릭 어레이가 얻어진다. 정렬을 위한 데이터 x_0, x_1, x_2 는 비트

단위로 어레이에 입력되며, MSB순으로 입력된다.

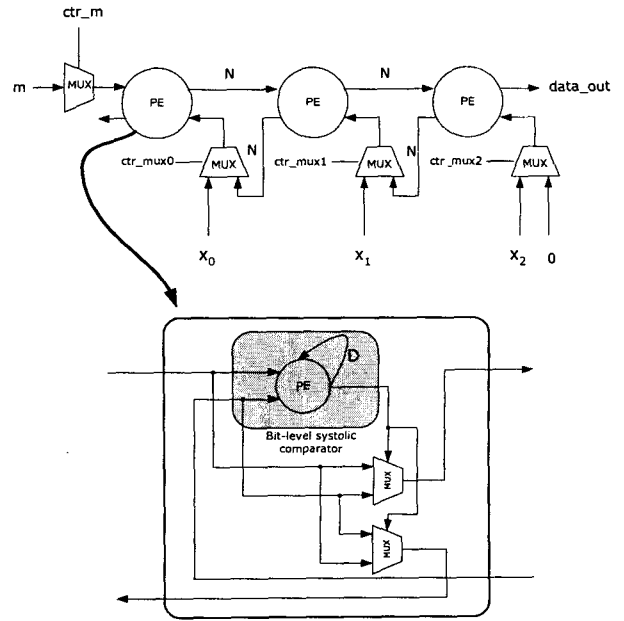


그림 8. 정렬을 위한 비트 레블 슈퍼 시스틀릭 어레이

IV. 시뮬레이션, 합성 및 구현

정규순환방정식으로 표현된 정렬(sorting)알고리즘으로부터 비트 레블 슈퍼 시스틀릭 어레이를 유도한 후 유도된 시스틀릭 어레이를 RT 수준에서 VHDL[5]로 모델링하여 Synopsys 시뮬레이션을 통해 동작을 검증하였다. 시뮬레이션 결과는 그림 9에 보인다.

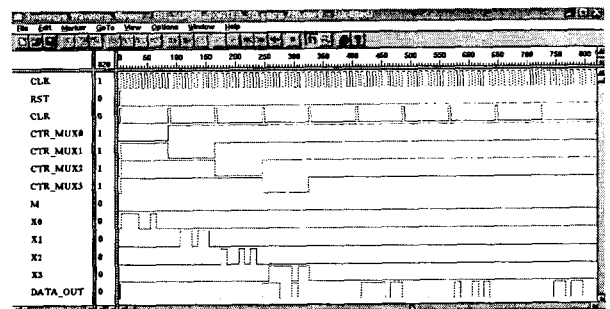


그림 9. N 이 4인 경우 시뮬레이션 결과

시뮬레이션 결과에 보이는 것처럼 N 이 4인 경우 8-비트 데이터 E4(11100100), 36(00110110), 4A(01001010), 7B(01111011)에 대하여 E4, 7B, 4A, 36 순으로 비트 레블에서 결과 값이 출력되는 것을 알 수 있다.

동작이 검증된 비트 레블 슈퍼 시스틀릭 정렬기는 Hynix에서 제공하는 0.35 μ m 셀 라이브러리를 기본으로 하는 Synopsys 디자인 컴파일러[6-7] 와 FPGA V200E

칩[8]을 사용하여 합성 및 구현되었다. N 이 4이고 각 데이터가 4-비트인 경우 비트 레벨 슈퍼 시스톨릭 정렬기의 합성 스케메틱은 그림 10에 보이며 셀 내부의 합성 스케메틱은 그림 11에 보인다. 그림 11에 보이는 것처럼 셀 내부의 비교 연산은 다시 비트 레벨 시스톨릭 어레이로 구성되어 있는 것을 볼 수 있으며 비트 레벨 시스톨릭 비교기의 셀 내부 연산은 그림 12에 보인다.

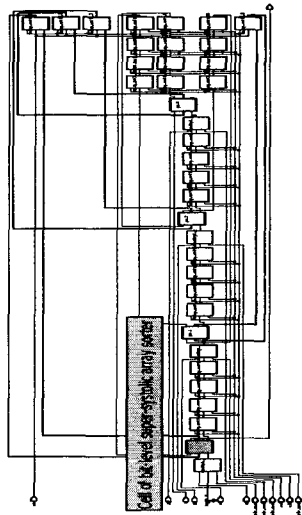


그림 10. 비트 레벨 슈퍼 시스톨릭 정렬기의 합성 스케메틱

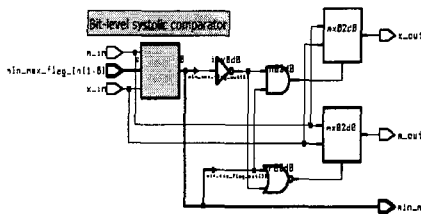


그림 11. 슈퍼 시스톨릭 정렬기의 셀 합성 스케메틱

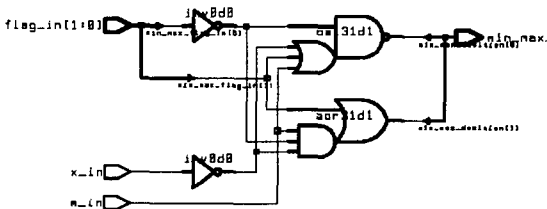


그림 12. 시스톨릭 비교기의 합성 스케메틱

슈퍼 시스톨릭 어레이의 우수함을 보이기 위해 본 논문에서 제안된 비트 레벨 슈퍼 시스톨릭 정렬기를 워드 레벨 시스톨릭 어레이와 비교 분석하였다. Synopsys 디자인 컴파일러를 이용 합성 결과를 표 4에 보이며 FPGA

V200E칩 상의 구현 결과는 표 4에 보인다.

표 4. Synopsys 합성 결과

Word Size	4		8	
	BS	WS	BS	WS
Design				
Total Area (2-input NAND)	308	373	489	853
Critical Path Delay (ns)	3.03	5.11	3.03	7.85

표 5. FPGA 구현 결과

Word Size	4		8	
	BS	WS	BS	WS
Design				
SLICES	29	44	53	84
Avg. Connection Delay	0.899	1.427	0.935	1.928
Avg. Connection Delay (10 worst nets)	1.566	2.469	1.672	4.294

BS : Bit-level super-systolic sorter.

WS : Word-level systolic sorter.

V. 결론

본 논문에서는 어레이 셀 내에 대한 동시성 처리를 높이기 위해 시스톨릭 어레이 구조 내 셀이 또 다른 시스톨릭 어레이 구조를 가지는 슈퍼 시스톨릭 어레이 구조를 제안하고, 그 예로 비트 레벨 슈퍼 시스톨릭 정렬기의 설계 및 구현에 대하여 기술하였다.

[참고문헌]

- [1] H. T. Kung and C. E. Leiserson, "Systolic array (for VLSI)", In *sparse Matrix Symposium*, pp. 256-282, SIAM, 1978.
- [2] G.Y.Song and J.J.Lee, Implementation of the Super Systolic Array for Convolution, *ASP-DAC 2003*, pp. 491-494, Jan. 2003.
- [3] H.T.Kung, "Why Systolic Architectures?", *Computer* Vol.15, No.1, pp.37-46, January 1982.
- [4] S.Y.Kung, *VLSI Array Processors*, Prentice Hall, 1988.
- [5] Y.C.Hsu, K.F.Tsai, J.T.Liu and E.S.Lin, *VHDL Modeling for Digital Design Synthesis*, Kluwer Academic Publishers, 1995.
- [6] K.C. Chang, *Digital Systems Design with VHDL and Synthesis*, IEEE Computer Society Press, 1999.
- [7] Weng Fook Lee, *VHDL Coding and Logic Synthesis with Synopsys*, Academic Press, 2000.
- [8] *Xilinx Data Book* 1995.