

향상된 재구성능력을 가진 고속 어레이 구조

이 재 익, 김 진 상, 조 원 경, 김 영 수
경희대학교 전자정보대학

전화 : 031-201-2196 / 핸드폰 : 016-796-1880

Fast Array Architecture with Improved Reconfigurability

Jae-Ic Lee, Jinsang Kim, Won-Kyung Cho and Youngsoo Kim
School of Electronics and Information Kyung Hee University, Kyungkido, 449-701
E-mail : lunchbox2@nate.com

Abstract

The reconfigurable architecture is increasingly important for design of multi-mode communication systems and computation-intensive DSP systems. The proposed coarse-grain architecture is based on a reconfigurable processing element consisting of a MAC unit, a register file, a context data register, and PE interconnect control blocks. The main feature of the proposed architecture is the loop context which enables faster configuration. Also, we propose another area-efficient reconfigurable architecture with improved reconfigurability. The SystemC modeling results show that the proposed architecture can reduce 9 clock cycles of 2D DCT compared to existing architectures.

I. 서 론

다양한 DSP알고리즘 및 통신알고리즘의 고속연산을 위해서는 알고리즘을 처리할 전용 ASIC 칩이 필요하다. 그러나 ASIC을 이용한 시스템은 프로그래밍 능력이 떨어지며, 경우에 따라 많은 수의 칩으로 구성되어 하드웨어 비용 및 소비전력이 증가한다. 이와 같은 문제점은 재구성 가능한 하드웨어를 이용하여 해결할 수 있으며, 다양한 알고리즘이 적용적으로 처리되어야 하는 SDR(Software Defined Radio)과 같은 응용분야에서는 재구성 가능한 시스템이 필수적이다[1]. 재구성

가능한 하드웨어는 재구성 단위를 기준으로, 1비트 단위로 프로그램이 가능한 fine-granular 구조와 word 단위로 재구성이 가능한 coarse-grain 구조로 분류할 수 있다[2-5]. FPGA와 같이 1비트 단위로 재구성이 가능한 fine-granular 구조는 제어신호가 많은 적용 알고리즘에 유리한 구조이며, coarse-grain 구조는 연산량이 많은 워드단위의 적용 알고리즘을 처리하는데 적합하다.

본 논문은 coarse-grain 구조를 기본으로 한 재구성 가능한 하드웨어 구조에 대한 연구이며, 제안된 구조는 다수의 PE(processing elements)가 상호 연결된 배열 구조이다. 각각의 PE는 독자적인 연산기 유닛과 내부 저장장치를 가지고, context라 불리는 프로그램 가능한 명령어에 의해 매핑되어 data 흐름과 연산기 동작을 제어 받는다. 본 논문에서는 재구성 가능한 하드웨어 구조를 설계하고, DCT 알고리즘을 적용하여 유용성을 검증하고자 한다.

제안된 구조는 SystemC의 RTL로 설계되었으며, 행과 열방향의 각각 2개의 전역버스를 통해 PE array에 data공급 및 출력이 고속으로 이루어지며, data broadcast 동작으로 Matrix * vector 연산에 유리하다. 각 PE는 Address를 가지며 broadcast mode 혹은 개별지정 mode을 가진다. 또한 loop context을 적용하여, 적은 수의 context 전달시간으로 기존의 재구성 가능한 하드웨어에 비해 동작속도가 향상됨을 확인하였다. 또한 coarse-grain구조의 재구성능력을 향상시킬 수 있는 4-bit 단위의 재구성 가능한 구조를 제안한다. 본 논문의 구성은 다음과 같다. II절에서는 고속의 재

구성가능한 어레이 구조를 제안하며 III절에서는 SystemC를 이용한 모델링 결과에 대하여 논의한다. IV절에서는 4-bit 단위의 재구성 가능한 구조를 제안하며 V절에서는 결론을 맺는다.

II. 재구성 가능한 고속의 어레이 구조

제안된 어레이 구조는 co-processor 로 이용될 수 있는 구조로써, matrix * vector, 반복연산 및 다양한 통신 및 멀티미디어 data의 고속처리에 유리하게 설계되었다.

그림1은 제안된 PE (processing element) array의 적용한 재구성 가능한 시스템의 구성의 예이다.

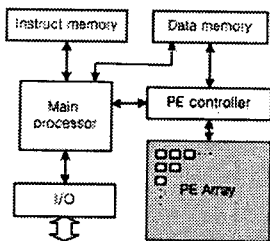


그림1. 제시된 PE Array를 적용한 시스템 구성도

2.1 PE의 상세구조

각 PE는 12bit 곱셈기와 24bit 가감산기 및 논리연산 장치를 가지며, Mux을 통해 입출력 데이터패스 연결을 지정한다. Mux 및 연산기는 context data에 의해 제어되며 PE 구조는 그림2 와 같다.

(1) 입력과정

PE내부를 재구성해주는 context는 연산 전에 PE에 공급된다. PE의 context 공급은 ROW_A bus 및 COL_A bus을 통해 이루어진다. context는 ROW_A bus 및 COL_A bus를 사용하여, 열 또는 행 방향으로 broadcast 될 수 있으므로 같은 행이나 열의 PE는 한번의 context broadcast로 동일한 context을 공급 할 수 있다. 각 PE에 서로 다른 context을 부여할 경우, context의 상위 4bit를 이용하여 addressing mode 방식의 context 공급이 가능하다. 이를 위해 PE의 BUS input 단에 address decoder를 두어 broadcast 되는 context인지, 또는 자신의 주소의 context인지 판단한다. 처리될 data는 ROW_A, ROW_B, COL_A, COL_B의 bus을 통해 12bit의 data가 입력된다. context와 마찬가지로 broadcast 혹은 addressing mode로 들어오는 data을 address decoder을 통해 자신의 data 인지 확인 후 입력받는다. context와 data의 구분은 입력의

최상위 1bit로 결정되며, context일 경우는 각각의 PE가 가지고 있는 context data에 저장된다. 알고리즘에 따른 일련의 동작은 context로 변환되어 PE에 공급되는데, 이런 일련의 동작을 순서대로 context data에 저장한다. context data는 PC(program counter)을 하나씩 증가시키면서 저장된다.

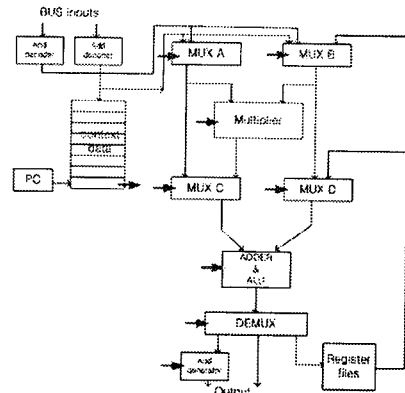


그림2. 제안된 PE의 상세구조

(2) 동작과정

context 입력이 끝난 후, start signal을 통해 PE의 연산이 시작된다. 모든 PE는 clock에 동기되어 동작하며, 매 clock마다 context data에 저장되어 있는 context data에서 PC값이 가리키는 context을 load하여 PE내부를 재구성한다. 재구성 대상은 MUX A, MUX B, MUX C, MUX D, Multiplier, Adder&ALU, DEMUX 이며, 이들의 동작 및 data path의 경로가 context에 의해 결정된다. 재구성과 동시에 처리할 data을 입력하고, 재구성된 PE의 동작이 수행된다. Load된 context가 loop context일 경우, 재구성 대상의 제어 신호대신 점프할 PC값과 반복횟수를 가진다. Loop context는 반복적인 일련의 context 처리에 적은 수의 context data 와 context 공급시간을 얻을 수 있어, context 실행순서를 loop 동작에 용이하게 설계할 경우 효율적인 동작을 수행한다.

(3) 출력과정

처리된 data의 출력경로는 PE 하단의 DEMUX에 의해 정의된다. 출력값은 ROW_A, ROW_B, COL_A, COL_B bus로 출력되거나, 내부 register files에 저장되어 누적 data연산이나, 차기 연산의 입력으로 이용될 수 있다.

각 PE는 자신의 출력 data를 행방향의 다른 PE에 전달할 수 있다. context에 의해 특정 PE로 출력하거나, 행 bus을 통해 broadcast하여, 자신이 속한 행의

모든 PE에 결과값의 전달이 가능하다. 또는 PE controller에 최종 결과값을 전달 할 수 있다. 이는 DEMUX 다음 단계 Address generator에 의해 이루어지며, 결과 값에 4bit의 address header를 추가함으로써 수행된다. Address generator에 의해 생성된 4bit address는 입력과정에 설명한 address mode에서의 address와 동일하다. Address generator 통한 self-broadcast를 이용하면 2D DCT 알고리즘과 같은 응용에서 1D DCT 후, 데이터 재배열을 위한 변환메모리가 필요없으므로 유용하다.

2.2 PE 어레이 구조

제시된 PE array는 8*8의 64개의 PE로 구성되며 PE array의 전체 구조는 그림3 과 같다.

(1) 동작

PE Array는 행과 열 방향의 각 2개의 bus가 각 8개의 PE에 연결되어 있다. ROW_B 와 COL_B는 PE에 data을 전달하며, ROW_A와 COL_A는 data및 context을 전달한다. 행과 열의 양방향으로 context을 전달하므로, array구조를 활용한 고속의 context 전달이 가능하다. 2D DCT의 경우, 열방향의 1D DCT후에 행방향의 1D DCT로 구현되는데, 이때 context 전달이 연산 동작 방향과 같이 열(행)방향으로 가능하므로, 열과 행 방향의 PE 재구성에 유리하다. 또한 가로와 세로에 2개의 bus를 사용하여 다수의 PE을 사용하는 2D array 구조에서 PE data 입력 및 결과의 출력을 고속화한다.

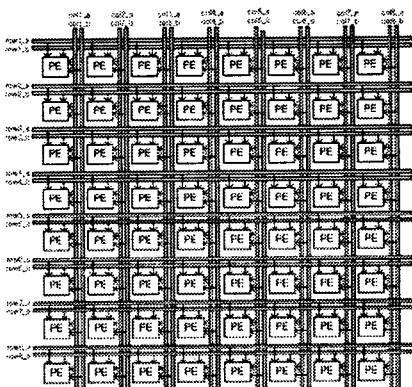


그림3. 제안된 PE Array Architecture

III. PE array 구조의 모델링 결과

3.1 Matrix * Vector operation

제안된 구조는 DSP 연산에서 많이 사용되는 matrix*vector 연산을 4개의 PE로 구현한다. 그림 4는

하나의 PE 동작결과를 나타낸다.

$$\begin{bmatrix} 5 & 9 & 4 & 2 \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \times \begin{bmatrix} 3 \\ 9 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 20 \\ 22 \\ 24 \\ 26 \end{bmatrix} \quad Z_0 = 5*3 + 9*8 + 4*2 + 2*4 = 7B$$

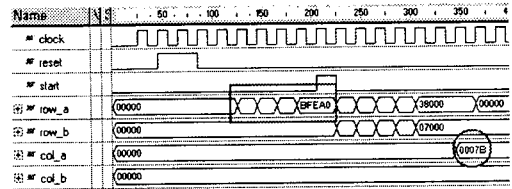


그림4. 제안된 PE Architecture

각 PE는 MAC연산을 한 cycle에 수행한다. 또한 loop context을 적용하여 4개의 context로 4x4 matrix*vector 연산을 수행한다. Context입력 후 연산 결과가 출력될 때 까지, 6 clock cycle이 필요함을 알 수 있다.

3.2 1D DCT의 적용

1D DCT는 Chen 알고리즘[2]을 이용하여, 8개의 PE로 2개의 matrix*vector를 구현하여 이루어진다.

Loop context를 적용하여, 5개의 context가 8개의 PE에 broadcast 되었고 4개 PE가 하나의 matrix*vector연산을 수행하였다. 계수값과 픽셀값의 입력시간을 포함하여 9 clock cycle 만에 1D DCT 연산결과를 출력한다.

3.3 2D DCT의 적용

그림5은 PE 64개를 이용한 8*8 block의 2D DCT 결과이다.

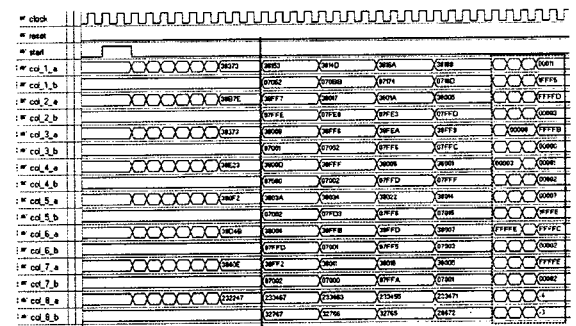


그림5. 제안된 구조의 2D DCT 수행결과

address generator를 이용하여 1D DCT 수행 후 결과값을 자신이 속한 행의 PE에 broadcast한다. 이때 COL_A와 COL_B bus를 모두 사용하여, 4 clock cycle

동안 모든 1D 결과값을 다른 PE에 전달한다. 2D DCT 결과는 COL_A와 COL_B를 이용하여 4 clock cycle동안 출력된다. 그림7은 2D DCT결과와 비교 그래프를 보여준다[3]. 2D DCT는 계수 및 픽셀 data 입력을 포함하여 28 clock cycle이 소모되며 이는 기존의 재구성 가능한 구조보다 최소 9 clock cycle 만큼 고속으로 동작됨을 확인할 수 있다.

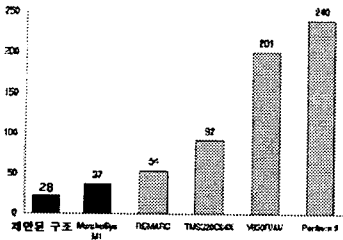


그림6. 제안된 구조의 2D DCT 성능비교

IV. 4bit의 재구성 단위의 PE 구조

지금까지 설명된 12bit PE단위의 재구성가능한 구조는 저전력과 작은 Chip size, 보다 높은 flexibility 요구하는 개인 휴대 단말기등의 Embedded 환경에 적응하기 위해 개선이 필요하다. PE의 연산단위가 작아 질수록 flexibility가 높아지며, 취급하는 데이터의 bit-width가 낮을 수록 utilization의 증가 및 사용치 않은 PE를 off 시킴으로 효율적인 전력관리가 가능하다. 반면, context 제어가 복잡해지며, 입력 데이터를 4bit단위로 분할하여 PE에 할당해야 하는 어려움이 있다.

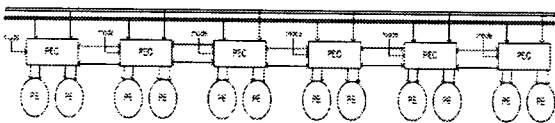


그림7. 4bit PE 단위 구조의 행 연결 모습

제안된 구조는 이의 보안을 위해 2개의 4bit PE 마다, PEC(PE controller)을 두어, 8, 16, 24 등의 8배수의 bit-width의 연산에 PEC단위의 연산동작을 수행하여, 처리될 데이터 할당과, context 공급을 보다 원활하게 하는 장점을 가진다. 또한 PEC에 merge와 free mode을 두어, merge mode는 2개의 묶음으로 PE동작을 수행하며 free mode 일때는 3개의 PE로 동작을 수행하여, 4, 12, 20 등의 4배수의 PE 동작에 유리하게 설계되었다. 그림 7은 제안된 4bit PE가 행으로 연결 구조으로, 먼저 제안된 12bit 단위의 PE구조의 장점을 수용하기 위해 그것과 유사한 PE 구조을 하고 있으며,

PEC의 연결을 통해 4bit단위의 PE구조의 단점을 해결하고자 한다.

V. 결 론

본 논문은 재구성 가능한 PE를 설계하고 이를 2D array형식으로 구성하였으며, 2D DCT 연산을 적용하여 기존의 재구성 가능한 구조보다 고속의 연산동작을 확인했다. 이는 context 가능한 행과 열방향의 전역 bus 및 PE의 address generator를 통한 self-broadcast, 2개의 bus을 통한 빠른 data 공급과 출력으로 이루어 졌다. 또한 loop context 적용으로 효율적인 PE 재구성을 실현했다. 이후의 연구단계로 4bit PE단위의 보다 효율적인 재구성 가능한 구조를 제안 하였으며, 이는 2개의 PE를 PEC을 통한 묶음단위 연산으로 이루어진다. 본 논문은 이를 위한 초기연구단계의 결과로써, 최적화된 PE 구조 탐색과 다양한 영상 및 통신 알고리즘의 동시적용을 위한 재구성 가능한 하드웨어 구조연구의 기반으로 활용될 수 있다.

감사의 글

본 연구는 한국과학재단 목적기초연구 (R01-2003-000-10149-0) 지원으로 수행되었습니다.

Reference

- [1] 김지연, 김진엽, "SDR(Software Defined Radio) TTA 저널 통권 82호 2002. 08
- [2] W-H Chen, C. H Smith and S. C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform", IEEE Trans. on Comm., Vol COM-25, No. 9, September 1977, pp. 1004-1009.
- [3] Guangming Lu, Hartej Singh, Ming-hau Lee, Nader Bagherzadeh, Fadi . Kurdahi, Eliseu M. C. Filho, Vladimir Castro Alves, "The MorphoSys Dynamically Reconfigurable System-On-Chip"
- [4] Reiner Hartenstein, "A decade of Reconfigurable Computing : a Visionary Retrospective", IEEE, 2001
- [5] Wu Jigang, Srikanthan Thambipillai, "A Run-time Reconfiguration Algorithm for VLSI Arrays", Proceedings of the 16th International Conference on VLSI Design IEEE, 2003