

향상된 설계공간을 갖는 혼합 가산기 구조와 최적화

이덕영^o, 이정아, 이정근, 이상민^{*}

강원대학교 컴퓨터학과^{*}, 조선대학교 전자계산학과, 캠프리지대학교 컴퓨터팀
riverlike@kangwon.ac.kr^o

Extending the Design Space of Adder Architectures and Its Optimization

Deok-Young Lee^o, Jeong-A Lee, Jeong-Gun Lee, Sang-Min Lee
Dept. of Computer Science, Kangwon Nation University^o

요 약

본 논문에서는 다양한 캐리 전달 방식(carry propagation scheme)이 단일 가산기 설계를 위하여 복합적으로 사용되는 가산기 구조를 제안하며, 이를 통하여 보다 향상된 delay-area trade-off 점들을 갖는 설계공간을 생성한다. 제안된 구조의 가산기는 각기 다른 캐리전달 방식의 하부 가산기 블록들을 캐리 입/출력 신호를 선형으로 연결한 구조이며, 기존의 단일 캐리전달 방식의 가산기와 달리, 다양한 delay-area trade-off 특성을 갖는 여러 종류의 캐리전달 방식을 비트 수준에서 조합하여 사용함으로써 보다 상세한 delay-area 설계공간을 생성해 낼 수 있다. 그러나, 제안된 가산기 구조의 설계공간은 다양한 캐리전달 방식이 비트 수준에서 활용되므로, 활용 가능한 설계 조합은 설계하고자 하는 가산기의 비트 폭과 고려하는 캐리전달 방식의 수에 비례하여 폭발적으로 증가하게 된다. 따라서, 제안된 가산기의 효율적이며, 자동화된 설계공간 탐색 방법이 요구된다. 본 논문에서는 이를 해결하기 위하여 정수 선형 프로그래밍(Integer Linear Programming, ILP) 방법을 이용하여 제안한 가산기의 최적화 문제를 형식화함으로써 효과적인 설계공간의 탐색 방법을 제안하였다.

1. 서 론

디지털 시스템 설계자들의 목표는 최소의 개발비용으로 최고 성능을 발휘하는 디지털 시스템을 설계하는데 있다. 디지털 시스템의 성능은 가산기의 성능에 따라 좌우된다. 이는 가산기가 디지털 시스템에서 가장 기본적으로 필수적인 연산회로의 하나이기 때문이다. 따라서 지난 수십 년간 디지털 회로 설계에 있어 중요한 설계 요소인 속도(speed)와 회로의 크기(area), 그리고 전력 소모량(power consumption)을 고려한 다양한 분석과 연구가 가산기 설계에 관하여 진행되어 왔다 [1].

가산기의 연산 속도 및 구현 비용은 캐리의 전달지연을 처리하는 방식에 영향을 받는다. 캐리 전달 문제를 해결하기 위하여, 비트들을 블록으로 그룹화하고 블록수준에서 캐리를 스킵하는 방식부터 캐리의 입력의 두 경우, 0과 1인 경우를 미리 계산해두고 실제 캐리가 입력되었을 때 해당되는 결과값을 선택하는 방식 및 Look-ahead와 같은 방법 등 전달 속도를 향상시키는 다양한 설계 방법들이 있다 [2, 3]. 최근, 고속의 캐리 전달을 위하여 두 개의 고속 캐리 전달 방식의 가산기가 조합된 형태로 개발된 바 있다 [4]. 가산기로 많이 사용되는 대표적인 것은 Ripple Carry Adder(RCA), Carry Skip Adder (CSKA), Carry Lookahead Adder(CLA) 등이다. 이와 같은 가산기 구조는 속도, 회로의 크기, 소모 전력량이 설계의 제약에 따라 선택되어 사용된다. 이와 같이 시스템 설계과정에서 목적 시스템의 특성에 적합한 가산기 구조를 찾는 것을 가산기의 설계공간 탐색(Design Space Exploration)이라고 한다 [5]. 본 논문에서는 가산기의 설계공간을 탐색하기 위한 제약조건으로 연산속도(delay)와 회로크기(area)를 고려하였다. 연산속도와 회로크기는 Trade-off 관계를 갖는다. 단일 타입의 가산기들은 그 캐리 전달 방식의 특성을 나타내는 Trade-off 관계를 갖으며, 이 관계를 delay-area 곡선으로 표현할 수 있다 [1, 6].

그림 1은 delay-area 곡선들로 표현된 가산기의 설계공간을 나타낸다. 각 가산기의 delay-area 곡선은 캐리 전달 방식에 따라 다른 특성을 나타낸다. CLA는 다른 가산기들에 비하여 지연시간은 작지만 회로크기는 매우 크다. 반면, RCA는 지연시간은 크지만 회로크기는 작다. 그림 1에서 실선으로 표현된 각 가산기들은 Trade-off 곡선으로 하나나 형의 그래프를 형성하지만, 전체 시간 축에서 보면, 세 개의 하나나 형 그래프가 연결된 형태로서 area-delay 곡선은 계단 형의 Trade-off 곡선을 형성하게 된다.

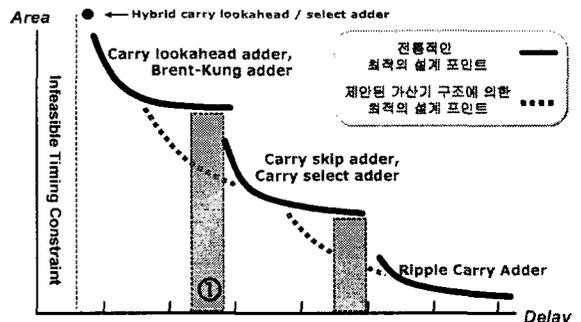


그림 1. 가산기의 설계공간 (Delay-Area Trade-off 관계)

이와 같은 현상은 delay와 area의 관계가 반비례 관계를 형성하지 못하는 비이상적인 결과를 초래한다. 설계의 시간제약이 area가 수직적으로 변하는 직선 구간의 ①과 ②로 표시된 영역에 위치하게 되면, 설계자는 가산 연산시간의 제약조건이 약화되고 있음에도 불구하고 최적화가 완만하게 이루어져 실질적인 이득을 얻지 못하게 될 수 있다. 따라서 그림 1의 점선으로 기술된 것과 같이 전체 시간 축 상에서 보다 반비례에 가까운 최적의 가산기 설계공간을 구성해야 할 필요가 있다. 확장된 설계공간에서는 목적시스템의 제약조건을 최대한 만족시키는 가산기를 탐색할 수 있다.

2. 관련연구

2.1 가산기의 종류와 특성

구조적 측면에서, 기본적으로 몇 가지의 캐리 전달 방식으로 구분될 수 있다 [1, 2]. 대표적인 가산기의 구조로 다음 세 가지의 가산기를 간략히 기술하고자 한다.

√ Ripple Carry Adder (RCA) : 가산기 종류 중 가장 간단한 구조를 띠며, 전가산기의 캐리 입출력을 선형으로 연결한 것이다. 따라서 n 비트 RCA의 경우 약 n*(전가산기의 지연시간)의 가산 처리 시간이 요구된다. 반면, 회로 사용량은 특별한 캐리 가속 회로가 없으므로 매우 적은 실리콘 영역 내에 구현될 수 있다.

√ Carry Skip Adder (CSKA) : CSKA는 특정 블록의 비트 패턴 정보를 이용하여, 그 블록으로 캐리 입력이 인가되었을 경우, 블록 내의 비트별 캐리 전파 없이 다음 블록으로 캐리 신호를 전파하는 캐리 전달 방식을 갖는다. 이를 위하여 각 블록은 캐리를 전파시킬 수 있는 입력 패턴을 검출하기 위한 부가 회로가 필요하다. RCA보다 향상된 가산 처리 속도를 갖고 있으나, 부가 회로의 추가로 인하여 회로 사용량은 증가된다.

√ Carry Lookahead Adder (CLA) : CLA는 임의의 블록 내에서 고속의 캐리 생성을 위하여 캐리 생성 식을 확장한다. 이와 같이 확장된 고속의 캐리 생성 식은 Lookahead 블록 회로로 구현되어 빠른 캐리를 생성하지만, 많은 회로 영역을 필요로 한다. 더욱이 다층 레벌로 CLA가 구현되는 경우, RCA나 CSKA에 비해 많은 회로 사용량이 요구된다.

이외에도 고성능의 Prefix 가산기와 다양한 수-시스템(Number System)에 따라 수많은 가산기의 구조가 가능하다 [1, 2, 3]. 표 1은 세 가지 가산기의 비트 폭 n 에 따른 delay와 area의 asymptotic complexity를 보여주고 있다[2].

표 1. n -비트 가산기의 Delay와 Area

Adder Type	Delay	Area
Ripple Carry Adder(RCA)	$O(n)$	$O(n)$
Carry Skip Adder(CSKA)	$O(\sqrt{n})$	$O(n)$
Carry Lookahead Adder(CLA)	$O(\log n)$	$O(n \cdot \log n)$

2.2 정수 선형 프로그래밍

선형 프로그래밍(Linear Programming, LP)의 분야는 최적화에 있어 많은 연구가 진행되어 온 분야이다 [7]. 특히 선형 시스템의 해가 정수라는 제약 갖는 정수 선형 프로그래밍(ILP)은 전산학을 넘어서 산업공학 및 전자공학 등 많은 분야에서 최적화 전략으로 사용되고 있다.

ILP의 목표는 주어진 제한 조건들을 만족하는 정수 변수들에 대한 목적함수를 최소화하거나 최대화하는 것이며, 이 때 목적함수 및 제약 조건은 변수들의 선형 조합(linear combination)으로 기술되는 선형 함수이다. ILP의 일반적인 형식은 다음과 같다.

$$\begin{aligned} \text{Minimize} & : Ax \\ \text{Constraints} & : Bx \leq C, x \geq 0 \end{aligned}$$

A 는 비용 벡터(cost vector)이고, B 는 제한조건 행렬(constraint matrix)이고, C 는 제한조건들의 컬럼 벡터(column vector)이며 x 는 정수 변수들의 벡터이다.

3. 혼합 가산기 구조 제안

목적 시스템의 특징을 최대한 만족시키는 가산기를 탐색하기 위하여, 캐리 전달 방식(carry propagation schemes)이 서로 다른 가산기 블록들이 선형으로 연결된 혼합 가산기 구조를 제안한다.

제안하는 혼합 가산기는 가산기 블록 $CA_i(n_i)$ 는 캐리 전달 방식(carry propagation scheme)이 CA_i 형태의 n_i -비트 가산기로 정의한다. 고려하고 있는 캐리 전달 방식의 종류가 I 가지 일 경우, 제안된 n -비트 가산기는 I 개의 $CA_i(n_i)$ 가 선형적으로 연결된 n -비트 가산기로 정의 된다 ($1 \leq i \leq I$). 그림 2와 같이 $CA_i(n_i)$ 의 Carry-in은 $CA_{i-1}(n_{i-1})$ 의 Carry-out 신호이고, n_i 의 모든 합은 n 이다. 그림의 경우, $I=3$ 이고 $n_1+n_2+n_3=n$ 이다. $n_i=0$ 일 경우 $CA_i(n_i)$ 는 $CA_{i-1}(n_{i-1})$ 에서 $CA_{i-1}(n_{i+1})$ 로 캐리를 전달하는 도선(wire)이 된다.

제안하는 혼합 가산기는 캐리 전달 방식과 캐리 전달 방식이 적용되는 비트의 폭에 따라 다양하게 조합될 수 있다. 목적 시스템의 제약조건을 최대로 만족하는 가산기를 찾기 위하여, 가능한 모든 가산기 조합을 탐색하는 것은 많은 시간이 소모된다. 적합한 가산기를 탐색하기 위한 최적화가 필요하다.

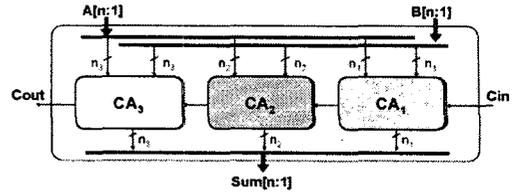


그림 2. 제안하는 혼합 가산기 구조

본 논문에서 최적화 문제는 가산기 블록에 비트 폭을 할당하는 문제로 정의하였다. 가산기 블록에 적용되는 캐리 전달 방식의 종류와 개수는 설계자에 의해 임의로 선택될 수 있도록 하였다.

4. ILP 기반의 최적화

주어진 시간 제약조건을 최대한 만족하면서 회로크기가 최소인 가산기를 탐색하는 것은 " n -비트 가산기에 대하여, 각 가산기 블록들의 최적 비트 폭, n_1, n_2, \dots, n_I 를 찾는 것"이다.

4.1 시간 모델

연산 종료시간에 대한 시간모델을 설명하기 위하여, 3개의 가산기 블록을 사용한다고 가정하자. 가산기 블록에 캐리전달 방식이 다른 RCA, CSKA, CLA를 사용하였다. 그림 3은 각 가산기 블록에서의 carry 전달과 sum 계산 시간을 나타낸 것이다. 가산기 블록에 할당된 비트 폭과 캐리 전달 방식의 위치에 따라 연산 종료시간에 차이가 발생한다. 그러므로, n -비트 가산기의 연산 종료시간은 아래의 알고리즘과 같이 첫 번째 비트에서 마지막 비트까지의 carry 전달 시간과 중간 중간 발생하는 각 $CA_i(n_i)$ 의 sum 생성 시간 중 최대 시간을 함한 것으로 정의한다.

```

Sum[0] = Carry[0] = 0;
For i = 1 to I
Begin
Sum[i] = Carry[i-1] + Dni,sCAi;
Carry[i] = Carry[i-1] + Dni,cCAi;
End
CompletionTime = Maxi { Sum[i] };
    
```

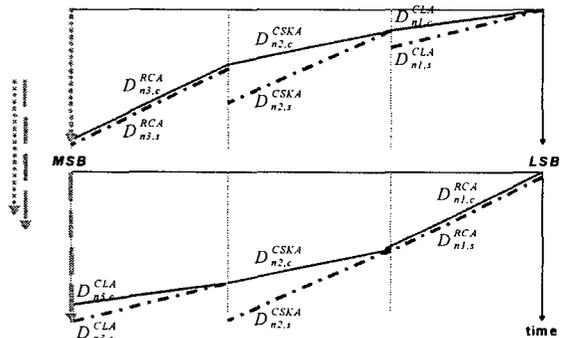


그림 3. 가산기 블록에서의 연산 종료시간 비교

4.2 ILP 형식화

연산 종료시간을 최적화하는 문제는 다음과 같이 형식화 될 수 있다. $\delta(i)$ 는 LSB로부터 i 번째 위치한 가산기 블록의 캐리 전달 방식을 나타낸다.

$$\begin{aligned} \text{Objective} & : \text{Minimize } \text{Max}_i \{ \text{DELAY}(\delta(i)) \} \\ \text{Constraints} & : \sum_{i=1}^I \text{AREA}(\delta(i)) < UB \\ & : \sum_{i=1}^I n_i = n \end{aligned}$$

위의 식에서 **AREA**와 **DELAY**는 주어진 가산기 블록 $k(i)$ 의 area와 delay를 계산하는 함수이다. RCA를 제외한 CSA와 CLA의 area와 delay는 주어진 비트 폭 n_i 에 대해 비선형 함수이다. 이는 설계 공간의 탐색을 어렵게 한다. 이처럼 복잡한 ILP 문제를 해결하기 위하여, 가능한 캐리전달방식과 가능한 비트 폭에 대해 area와 delay를 계산한 후 이 값을 특정 캐리 전달 방식과 특정 비트 폭을 나타내기 위하여 도입한 변수 $x_{n_i}^{k(i)}$ 와 조합하여 전체 가산기의 회로크기와 연산종료 시간을 기술하였다. 이로써 비선형함수는 그림4와 같이 선형함수로 다시 표현할 수 있다.

Objective : Minimize CompletionTime
 = Minimize $Max_i \{S_i\}$
 = Minimize z

Constraints :

$$\sum_{k=1}^{i-1} \sum_{n_s=0}^n D_{n_s, c}^{k(i)} \cdot x_{n_s}^{k(i)} + \sum_{n_s=0}^n D_{n_s, s}^{k(i)} \cdot x_{n_s}^{k(i)} \leq z$$

for all $1 \leq i \leq I$

$$AREA = \sum_{k=1}^I \sum_{n_s=0}^n A_{n_s}^{k(i)} \cdot x_{n_s}^{k(i)} < UB$$

$$\sum_{n_s=0}^I n_i x_{n_s}^{k(i)} = n$$

$x_{n_s}^{k(i)} > 0$ and Integer
 for all $1 \leq i \leq I, 1 \leq n_s \leq n$

그림 4. 연산 종료시간 최적화를 위한 ILP 모델

5. 실험 및 분석

본 논문에서는 제안된 방법의 유용성을 증명하기 위하여 128-비트 가산기의 최적화된 설계공간의 탐색에 관한 실험을 수행하였다. 실험에는 RCA, CSA, CLA 세 가지 캐리전달 방식의 가산기 블록을 사용하였고, 각 가산기 블록들이 할당될 수 있는 비트 폭을 4, 8, 16, 32, 64, 128로 제한하였다.

상수 $D_{n_s, c}^{k(i)}$ 와 $D_{n_s, s}^{k(i)}$ 그리고 $A_{n_s}^{k(i)}$ 는 [8]에서 제안된 모델을 통해 계산된 sum과 carry 그리고 회로크기에 대한 상수 값이다. 형식화된 ILP 문제를 해결하기 위해 Ip_solve 4.0을 사용하였다. Ip_solve는 Eindhoven University of Technology에서 개발한 public-domain ILP 소프트웨어이다.

그림 5는 위에서 정의한 문제의 해를 Ip_solver를 통해 얻은 후 area-delay 곡선으로 나타난 것이다. 이 그림은 제안된 혼합 가산기 구조의 장점을 나타내고 있다. '●'로 표시된 포인트는 그림 1에서 정선 상의 포인트에 해당되고, '+'로 표시된 포인트는 바나나 모양 곡선 상의 포인트에 해당된다. '■'로 표시된 최적의 포인트는 세 가지 타입의 가산기를 조합함으로써 생성된다. 이 포인트들은 설계비용을 최적화하는데 도움을 준다. 그림 5의 ②에 표현된 설계 포인트는 제안된 구조의 가산기에 의해 새로 생성된 최적 포인트의 일부이다. 64-비트 CLA와 64-비트 CSA로 구성된 가산기는 단일 타입 가산기 CLA만으로 구성된 ①의 어떤 포인트보다도 회로크기가 작다. 또한 delay 속의 범위가 약 140에서 340사이의 범위 내에서, 제안된 혼합 가산기 구조는 단일 타입 가산기 CSA보다 회로크기가 작게 구현될 수 있다.

이상의 실험 결과는 동일한 시간 제약조건에 주어졌을 때, 제안된 구조의 혼합 가산기가 단일 타입 가산기보다 회로 사용량이 적게 소모되므로 회로크기가 작은 가산기를 구현할 수 있다. 그러므로 제안된 혼합 가산기 구조는 주어진 시간제한 조건을 만족하면서 회로크기를 최소화함으로써 저비용 가산기의 설계를 가능하게 한다. 즉, 설계비용의 최적화가 가능하다.

6. 결론 및 향후과제

본 논문에서는 다양한 캐리 전달 방식(carry propagation scheme)을 가진 가산기 블록을 선형적으로 연결한 혼합 가산기

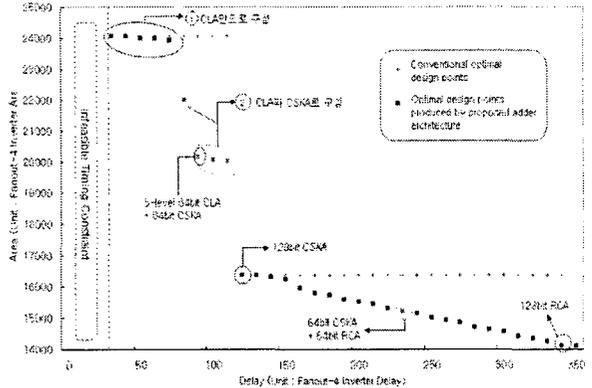


그림 17. 제안된 혼합 가산기가 생성하는 설계공간(128비트)

구조를 제안하였다. 각 가산기 블록들은 서로 다른 비트 폭으로 구성되며, 캐리 in/out 시그널을 통해 선형적으로 연결된다. 시간 제약조건이 주어졌을 때, 확장된 설계공간에서는 단일 타입의 가산기만을 고려하는 기존 설계공간에서보다 회로크기가 작은 가산기를 탐색할 수 있다. 제안된 혼합 가산기 구조는 각 가산기 블록이 서로 다른 비트 폭과 서로 다른 캐리 전달 방식을 사용하기 때문에 단일 타입의 가산기가 커버하지 못하는 지연시간 영역 내에서 최적화된 가산기를 탐색할 수 있다. 따라서 비트 수준에서 특정한 시간 제약조건을 만족하는 최적화된 가산기를 구현할 수 있다. 본 문에서는 시간 제약조건을 만족하면서 회로의 크기가 최소인 가산기를 탐색하기 위하여, 최적화 기법을 기반으로 하는 ILP (Integer Liner Programming)를 사용하였다. 실험을 통해 제안된 가산기 구조가 delay-area 관점에서 보다 향상된 trade-off 곡선을 생성함을 알 수 있었다.

참고 문헌

- [1] C. Nagendra, M.J. Irwin, R.M. Owens, "Area - time - power tradeoffs in parallel adders," In IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 43, pp. 689-702, Oct. 1996
- [2] M.D. Ercegovic and T. Lang, "Digital Arithmetic," Morgan Kaufmann Publishers, 2004
- [3] James E. Stine, "Digital Computer Arithmetic Datapath Design Using Verilog HDL," Kluwer Academic Publishers, Nov. 2003
- [4] Y. Wang, C. Pai, X. Song, "The design of hybrid carry-lookahead/carry-select adders," In IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 49, Jan. 2002
- [5] Anu Gupta, "Programmatic design space exploration through validity filtering and quality filtering"
- [6] J. Zhu and R. Kelly, "Architectural Diversity - The Key to Design Compiler Optimization," In DESIGN WARE, Synopsys Technical Bulletin, vol. 3, Q2 1998, "http://www.synopsys.com/news/pubs/dwtb/q298/frame_dwtb1.html"
- [7] H.P. Williams, "Model Building in Mathematical Programming," 4th Ed., John Wiley, New York, 1999
- [8] D.E. Williams, E.E. Jr. Swartzlander, "Parametric delay and area models for adders," In Proc. of the 36th Midwest Symposium on Circuits and Systems, pp. 863 - 870, Aug. 1993