A Design Method for Heterogeneous Adders

Jeong-Gun Lee[1], Jeong-A Lee[2], Byeong-Seok Lee[2], and Milos D. Ercegovac[3]

[1] Computer Laboratory, University of Cambridge, UK
Jeong-Gun.Lee@cl.cam.ac.uk
[2] Department of Computer Engineering, Chosun University, Republic of Korea
[3] Dept. of Computer Science, University of California Los Angeles, US

Abstract. The performance of existing adders varies widely in their speed and area requirements, which in turn sometimes makes designers pay a high cost in area especially when the delay requirements exceeds the fastest speed of a specific adder, no matter how small the difference is. To expand the design space and manage delay/area tradeoffs, we propose new adder architecture and a design methodology. The proposed adder architecture, named *heterogeneous adder*, decomposes an adder into blocks (sub-adders) consisting of carry-propagate adders of different types and precision. The flexibility in selecting the characteristics of sub-adders is the basis in achieving adder designs with desirable characteristics.

We consider the area optimization under delay constraints and the delay optimization under area constraints by determining the bit-width of sub-adders using Integer Linear Programming. We demonstrate the effectiveness of the proposed architecture and the design method on 128-bit operands.

1 Introduction

Embedded system design recently has attracted much attention from academia and industry for its great demand in market. The demand in applications is very wide and the diversity covers from the very high-performance real-time embedded systems to the very simple micro-controller based embedded systems. In consequence, design requirements and constraints varies a lot according to applications, and cost-effective optimization has to be applied to reduce their cost. Especially the cost-effectiveness is very important in the hardware parts of the embedded systems.

In this paper, we propose a cost-effective optimization of adder architecture. Addition is an essential operation in a digital embedded hardware/system and many adders such as a ripple carry adder (RCA), a carry skip adder (CSKA), a carry lookahead adder (CLA), and a parallel prefix adder (PA) have been developed utilizing different carry generation schemes to provide tradeoffs in speed, area and power consumption [1,2]. To date, these single type (homogeneous) adders were mixed in a multi-level architecture by adopting a different scheme for carry and sum generation, e.g., a carry lookahead scheme in the carry generation network and a carry select scheme for the sum generation, as in the case of a hybrid carry-lookahead/carry-select adder [3].

Y.-H. Lee et al. (Eds.): ICESS 2007, LNCS 4523, pp. 121–132, 2007.

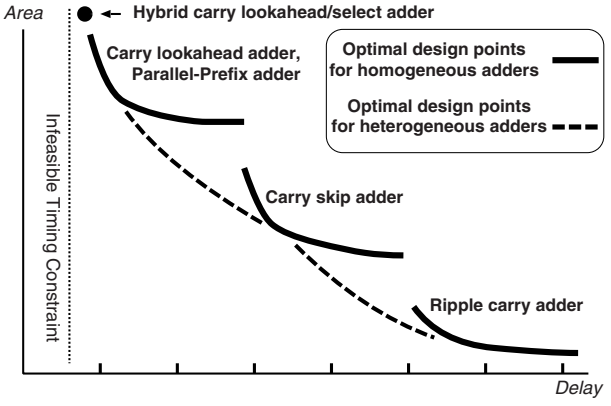


Fig. 1. Conceptual design space of adder design

For non-uniform input bit arrivals, adders consisting of sub-adders of different types have been proposed [4,5] and are typically used in the design of parallel multipliers. We do not consider here adders with non-uniform input arrivals. It is noteworthy that the hybrid adder in [3] improves the delay performance at the increased cost of area.

However, for a certain delay requirement, the additional area cost might be very big due to the characteristic of delay-area curves shown in Fig. 1. Solid lines in Fig. 1 are delay-area curves and correspond to a set of optimal design points for given conventional adder architectures [1,6]. The individual solid line is typically banana-shaped but the overall delay-area curve formed by these solid lines is not banana-shaped. Some of work [7,8] have tried to tradeoff area/delay through sparsing computational circuit and changing the amount of internal wiring/the fanout of intermediate nodes on the carry generation network in a parallel prefix adder.

In this paper, we propose a heterogeneous adder architecture, which corresponds to a pareto-optimal delay-area curve as denoted by the dashed lines in Fig. 1. The heterogeneous adder architecture permits better design tradeoffs in terms of area and delay. The same approach is expected to provide better tradeoffs when considering power consumption. The heterogeneous adder can be described as an n-bit adder where various bit-width blocks of carry propagation adders such as RCA, CSKA and CLA are connected iteratively using carry-in and carry-out signals of block adders. We will describe in this paper an Integer Linear Programming (ILP) based methodology to configure a heterogeneous adder. Our formulated ILP produces the best choice of types and precisions of sub-adders for two cases: (i) area-constrained delay optimization, and (ii) delay-constrained area optimization. Compared to the work [7,8], our approach provide more high-level view of arithmetic optimization without considering low-level circuit issues such as fanout size and wiring complexity. In addition, our formulation provide a more systematic optimization method through '*mathematically*' modelled adder delay and area.

The remainder of this paper is organized as follows. Heterogeneous adder architecture and its characteristics are explained in detail in Section II. In Section III, we address the mathematical modeling of the heterogeneous adder for its optimization. Experimental results are presented in Section IV to show the effectiveness of the proposed method. Finally, in Section V, we conclude this paper with summary of work and possible future work.

2 Proposed Architecture and Advantage

The architecture of a heterogeneous adder allows to combine different types of adder implementations with various precision to form a single heterogeneous adder. While the conventional implementation selects only one implementation in a given library, the proposed bit-level implementations with different sub-adders can explore an extended design space allowing more find-grained area-delay tradeoffs.

For example, in the Synopsis design library [12], many sub-adders of different precision are available and we can use those sub-adders to make an adder to satisfy design constraints manually which is quite laborious and inefficient. The proposed method provides an automated procedure for determining a best configuration of sub-adders and their precisions under given design constraints. The procedure is fast and allows efficient design exploration.

Definition [**Heterogeneous Adder**]. A ***S***ub ***A***dder $SA_i(n_i)$ is an n-bit sub-adder whose carry propagating scheme is denoted by SA_i. When the number of available sub-adders is I, an n-bit heterogeneous adder is defined as an n-bit adder which concatenates $SA_i(n_i)$ where $1 \leq i \leq I$. $SA_i(n_i)$ uses the carry-out signal of $SA_{i-1}(n_{i-1})$ as its carry-in signal. The sum of all n_i should be equal to n.

In Fig. 2 we show an n-bit heterogeneous adder with three different sub-adders, i.e., $I = 3$ with $n_1 + n_2 + n_3 = n$. In general, I can be any number according to the available adder designs. For better understanding, in this paper, we use three sub-adders for our example of heterogeneous adder. As we explore design space of a heterogeneous adder with at most three different sub-adders under either delay constraint or area constraint condition, it produces a solution with one, two sub-adders or three sub-adders. If the heterogeneous adder has two sub-adder components, this can be denoted as zero for a certain i in n_i.

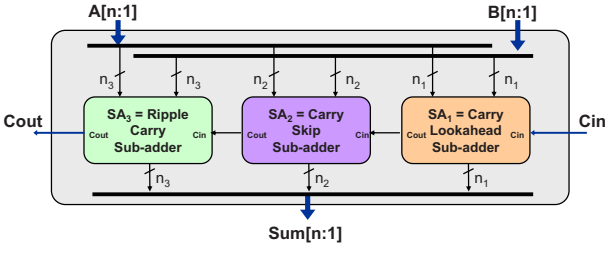


Fig. 2. A heterogeneous adder example

If n_i is found to be zero by the optimization procedures, then it means that SA_i type is not allocated to a heterogeneous adder and, in this case $SA_i(n_i)$ is simply a wire conveying a carry signal from $SA_{i-1}(n_{i-1})$ to $SA_{i+1}(n_{i+1})$. In this point of view, it is obvious that $SA_0(n_0)$ and $SA_{I+1}(n_{I+1})$ are carry-in and carry-out ports, respectively.

• **Property of Heterogeneous Adder:** First, the heterogeneous adder always has a lower cost (a lower delay) than a single type homogeneous adder for a given total delay (cost). Homogeneous adder is a subset of heterogeneous adder according to our definition. Consequently, in the worst case, the heterogeneous adder is implemented by a single homogeneous adder and, in other cases, a mix of different types of sub-adders implements the heterogeneous adder. Heterogeneous adder covers a bigger design space which in turn results a lower cost (delay) design for a given total delay (cost).

Second, the order of sub-adders has an impact on the delay of a heterogeneous adder. The processing delay of the heterogeneous adder is determined by finding a maximum value among the completion times of all sub-adders. In general, there is a time difference between carry generation delay and sum generation delay in each sub-adder. Depending on the order of sub-adders, the carry generation of sub-adders positioned at a MSB part can overlap sum generation of sub-adders at a LSB part.

Now, the question, "How to allocate bits for each sub-adder optimally and efficiently?" becomes an important design problem for a heterogeneous adder architecture. Here, 'optimal' means the minimum area under a delay constraint or minimum delay under an area constraint. The 'efficiency' relates to the efficiency of the automated procedure of finding optimal solutions.

To consider the advantages of the heterogeneous adder, which is optimized at the bit level, consider Fig. 3 showing the delays of three single type adders synthesized using Synopsis tools with $0.25\mu m$ CMOS library [11,12]. Even though there are timing variations, which can be controlled by the circuit or synthesis optimization, in general, there will be delay ranges falling between delay curves of various single type adders. This can be seen in Fig. 1.

When specific timing constraints fall into the uncovered delay ranges, a designer considering the design space consisting of single type homogeneous adders has no choice but to select another adder producing speed higher than necessary at an extra unavoidable cost. When an expanded design space provided by a heterogeneous adder is used, one can explore design points in the uncovered delay ranges, marked by a shadow region in Fig. 3, and select faster designs with smaller area.

3 ILP Based Optimization

3.1 Problem Formulation

With the target adder architecture, delay-constrained area optimization problem can be written as "find the bit widths, $n_1, n_2, \ \ldots \ , n_I$ of sub-adders to minimize

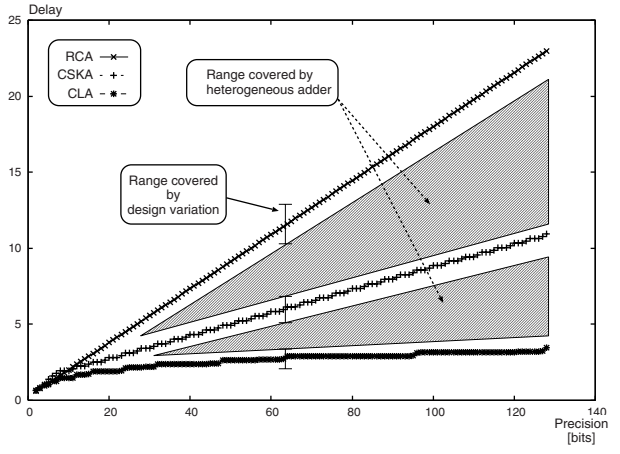


Fig. 3. Delay plot for three different types of adders

the area of n-bit heterogeneous adder, while satisfying a constraint that the total adder delay should be less than ub", where ub denote the upper bound of the total delay of the heterogeneous adder. The area-constrained delay optimization can be specified in a similar manner.

3.2 ILP Formulation

A direct formulation of the delay-constrained area optimization can be described by the following equations:

$$\underset{(n_1,n_2,\ \ldots\ ,n_I)}{Minimize}\ \text{AREA(Heterogeneous Adder)}$$

with Constraints

1: DELAY(Heterogeneous Adder) $\leq ub$

2: Bit-Width Sum of Sub-Adders $= n$

In these equations, 'AREA' and 'DELAY' are functions computing area and delay. To model these 'AREA' and 'DELAY', we use Integer Linear Programming (ILP) method [10]. Current ILP model assumes that the order of sub-adders allocated to a heterogeneous adder is fixed. Within an ILP formula, according to the assumption, 'I' types of sub-adders (SA_i, $1 \leq i \leq I$) are placed from the least significant bit (LSB) to the most significant bit (MSB).

In the proposed ILP formulation, an integer variable, $x_{ni}^{SA_i}$, denotes the number of $SA_i(n_i)$ which is used in a heterogeneous adder. Three constants, $D_{n_i,s}^{SA_i}$, $D_{n_i,c}^{SA_i}$ and $A_{n_i}^{SA_i}$, are the sum generation/carry generation delays and area of $SA_i(n_i)$, respectively. Here, 'c' denotes carry generation and 's' stands for sum generation. More precisely, $D_{n_i,s}^{SA_i}$ and $D_{n_i,c}^{SA_i}$ are worst-case delays from the carry-in signal arrival to the output of the sum and carry-out signals, respectively.

Basically, 'i' of n_i determines the type, SA_i, so we do not need to put 'SA_i' in the notations of these variable and constants. However, for better understaning, we put 'SA_i' to their notation.

- **Area Modelling:** For a specific i-th type of sub-adder, AREA ($SA_i(n_i)$) can be derived by the linear combination of $x_{ni}^{SA_i}$ and $A_{n_i}^{SA_i}$ as follows.

$$\sum_{n_i=0}^{n} A_{n_i}^{SA_i} \times x_{n_i}^{SA_i}, \text{ where } \sum_{n_i=0}^{n} x_{n_i}^{SA_i} \leq 1.$$

The equation, '$\sum_{n_i=0}^{n} x_{n_i}^{SA_i} \leq 1$', means at most one bit-width is selected for each type of sub-adder. This equation, however, can be removed in the constraint list to allow multiple instances of each sub-adder. When $n_1, n_2, \ \ldots \ , n_I$ are given, the area of a heterogeneous adder, (sum of the sub-adder areas), can be expressed as $\sum_{i=1}^{I}$ AREA($SA_i(n_i)$) and, finally, in the following form:

$$\sum_{i=1}^{I} \sum_{n_i=0}^{n} A_{n_i}^{SA_i} \times x_{n_i}^{SA_i}, \text{ where } \sum_{n_i=0}^{n} x_{n_i}^{SA_i} \leq 1.$$

In this paper we assume that the area of a heterogeneous adder is described by the summation of areas of its sub-adders. Since sub-adders are linearly combined to make a heterogeneous adder by connecting carry signals, we believe that this assumption is reasonable.

- **Delay Modelling:** Compared to area modeling, 'DELAY (Heterogeneous Adder)' is more complex to model because the completion delay is not simply a sum of delays of sub-adders. The DELAY of a heterogeneous adder is a maximum propagation delay between many possible data propagation paths from the input signals and carry-in signal to the sum and carry-out signals of the heterogeneous adder.

Fig. 4 shows one possible timing path of a heterogeneous adder with three different sub-adders. There are three possible completion times for each sub-adder, (1) $D_{n_1,s}^{CLA}$, (2) $D_{n_1,c}^{CLA} + D_{n_2,s}^{CSKA}$ or (3) $D_{n_1,c}^{CLA} + D_{n_2,c}^{CSKA} + D_{n_3,s}^{RCA}$. The final completion time is evaluated by finding the maximum delay of those three different path delays. With the assumption that I sub-adders are allocated from LSB to MSB in the predefined order, the DELAY of a heterogeneous adder can be formulated in the following way.

$$\text{DELAY} = \text{Max}\{T_{Pass_1}, T_{Pass_2}, \ldots, T_{Pass_I}\}$$
$$T_{Pass_1} = D_{n_1,s}^{SA_1}$$
$$T_{Pass_i} = \sum_{k=2}^{i} D_{n_{k-1},c}^{SA_{k-1}} + D_{n_i,s}^{SA_i},\ 1 < i \leq I$$

Since n_i can be varied from 0 to n, the constant values, $D_{n_i,s}^{SA_i}$ and $D_{n_i,c}^{SA_i}$ in this equations, can be derived from $\sum_{n_i=0}^{n} D_{n_i,s}^{SA_i} \times x_{n_i}^{SA_i}$ and $\sum_{n_i=0}^{n} D_{n_i,c}^{SA_i} \times x_{n_i}^{SA_i}$ with an additional constraint, $\sum_{n_i=0}^{n} x_{n_i}^{SA_i} \leq 1$, respectively, as in the AREA modeling.

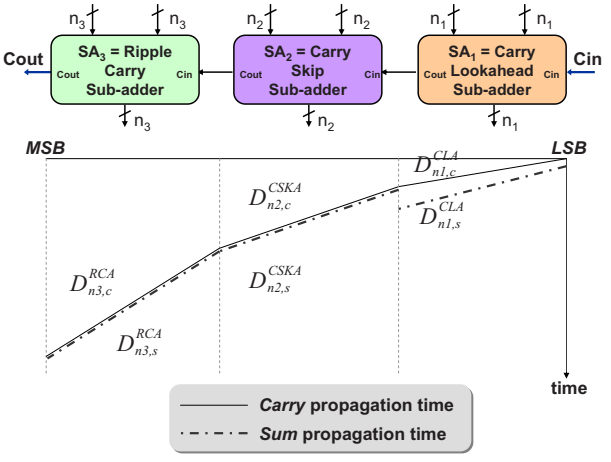


Fig. 4. Delay modeling of a heterogeneous adder

With the mathematically modeled delay and area, finally, the ILP formulation of our delay-constrained area optimization problem is described below.

$$\underset{(n_1,n_2,\ \ldots\ ,n_I)}{Minimize} \sum_{i=1}^{I}\sum_{n_i=0}^{n} A_{n_i}^{SA_i}\times x_{n_i}^{SA_i}$$

with Constraints

1: $\sum_{n_i=0}^{n} D_{n_i,s}^{SA_i}\times x_{n_i}^{SA_i} \le ub_{delay}$ for all SA_i, $(1 \le i \le I)$

2: $\sum_{k=2}^{i}\sum_{n_{k-1}=0}^{n} D_{n_{k-1},c}^{SA_{k-1}} \times x_{n_{k-1}}^{SA_{k-1}}$
$+ \sum_{n_i=0}^{n} D_{n_i,s}^{SA_i} \times x_{n_i}^{SA_i} \le ub_{delay}, \quad 1 < i \le I$

3: $\sum_{n_i=0}^{n} x_{n_i}^{SA_i} \le 1$ for all SA_i

4: $\sum_{i=1}^{I}\sum_{n_i=0}^{n} n_i \times x_{n_i}^{SA_i} = n$

Similarly, area-constrained delay optimization is formulated as.

$$\underset{(n_1,n_2,\ \ldots\ ,n_I)}{Minimize} \ d_{max}$$

with Constraints

1: $\sum_{n_i=0}^{n} D_{n_i,s}^{SA_i} \times x_{n_i}^{SA_i} \le d_{max}$ for all SA_i, $(1 \le i \le I)$

2: $\sum_{k=2}^{i}\sum_{n_{k-1}=0}^{n} D_{n_{k-1},c}^{SA_{k-1}} \times x_{n_{k-1}}^{SA_{k-1}}$
$+ \sum_{n_i=0}^{n} D_{n_i,s}^{SA_i} \times x_{n_i}^{SA_i} \le d_{max}, \quad 1 < i \le I$

3: $\sum_{i=1}^{I}\sum_{n_i=0}^{n} A_{n_i}^{SA_i}\times x_{n_i}^{SA_i} \le ub_{area}$

4: $\sum_{n_i=0}^{n} x_{n_i}^{SA_i} \le 1$ for all SA_i

5: $\sum_{i=1}^{I}\sum_{n_i=0}^{n} n_i \times x_{n_i}^{SA_i} = n$

4 Experiments

As an experiment, we consider the three types of sub-adders, RCA, CSKA, and CLA, in the design a 128-bit heterogeneous adder. In addition, possible bit-widths of sub-adders are restricted to the bit-widths from 2 to 128 bits. In this paper, variations caused by actual implementation are not considered since our focus is on bit-level optimization strategies. In actual implementation, some physical design issues including layout and interconnect will lead to alterations of the optimized design produced by our method. However, the structurally optimized design obtained from our method makes the actual physical optimization more efficient.

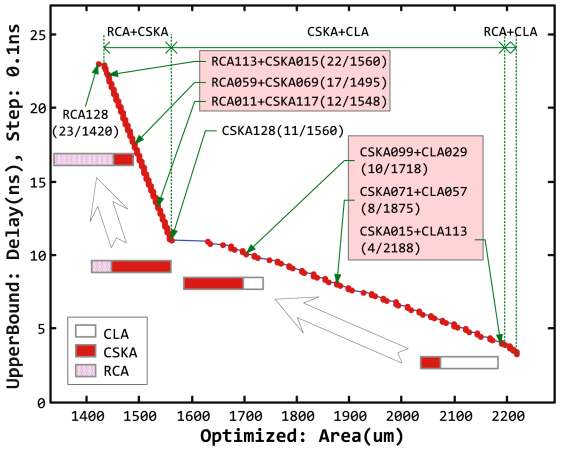
For experiment, we calculate the constant delay and area estimates, $D_{n_i,s}^{SA_i}$, $D_{n_i,c}^{SA_i}$ and $A_{n_i}^{SA_i}$ which are estimated from synthesized designs using Synopsis with $0.25\mu m$ CMOS technology [11,12]. The ILP formulation is solved using a public domain LP solver called "lp_solve" [10].

Fig. 5 depicts the advantage of the proposed adder architecture, which shows the area-delay curve of the 128-bit heterogeneous adder obtained through the LP solver. While changing delay (or area) boundary, ub, optimal heterogeneous adder configurations are searched to minimize area (or delay). Fig. 5(a) shows the case of "Delay-constrained area optimization problem" and Fig. 5(b) shows the case of "Area-constrained delay optimization problem". In this experiment, the constraint '$\sum_{n_i=0}^{n} x_{n_i}^{SA_i} \leq 1$' is removed from the constraint list to allow multiple instances of each sub-adder to get more room for optimization.

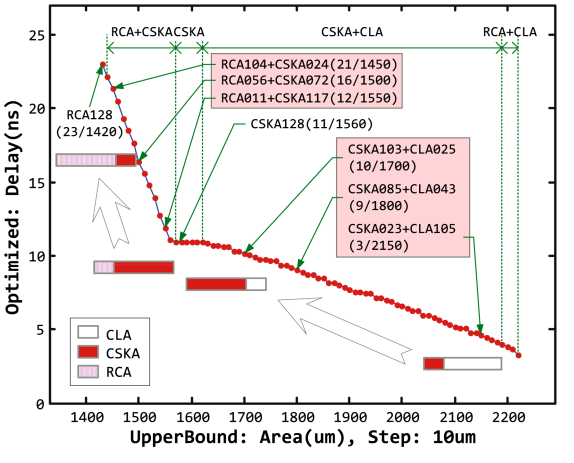
We considered all possible combination of ordering of sub-adders, that is 15 configurations of a heterogeneous adder out of 3 sub-adders, RCA, CSKA, and CLA. For better illustration purposes, in Fig. 5, we show configurations of some design points marked with their delay/area values. In the figures, two values in parentheses are 'delay' and 'area' pairs of the corresponding heterogeneous adders.

LP solver produced a three sub-adder concatenation such as RCA ‖ CSKA ‖ CLA as a solution (here, we use a symbol, '‖', to denote a linear carry connection between sub-adders. For example, in the case of RCA ‖ CSKA ‖ CLA, a CLA is located to a LBS part and CSKA is used in a middle part, and RCA is located to a MSB part). However, the difference of delay (or area) between RCA ‖ CSKA ‖ CLA and CSKA ‖ CLA was so small, we eliminate this 3 sub-adder solution in the design space and keep only one or two sub-adder configurations for the heterogeneous adder design space.

The design space we obtained, as indicated in Fig. 5(a) and 5(b), shows that RCA, RCA ‖ CSKA, CSKA, CSKA ‖ CLA, RCA ‖ CLA are good candidate orderings for the solution space. The design space covered by CSKA ‖ CLA in Fig. 5 contains many solutions with various precision of CSKA and CLA sub-adders. The precision of each sub-adder shown in the Fig. 5 explains clearly that heterogeneous adder indeed allows time-area tradeoffs much better than the conventional adder design. The solutions with RCA ‖ CSKA, CSKA ‖ CLA, RCA ‖ CLA are newly introduced design points. Those newly introduced



(a) Delay-constrained Area optimization problem

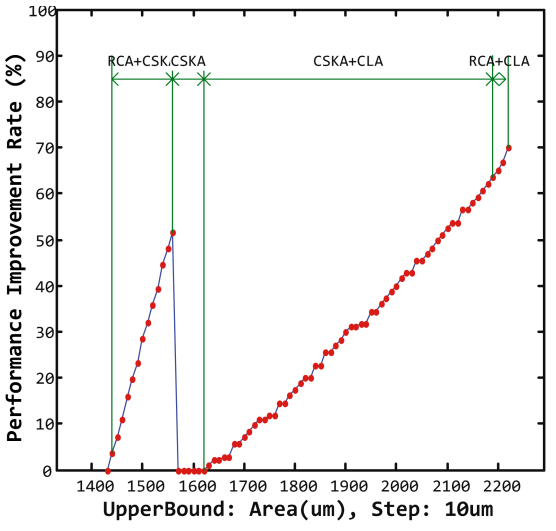


(b) Area-constrained Delay optimization problem

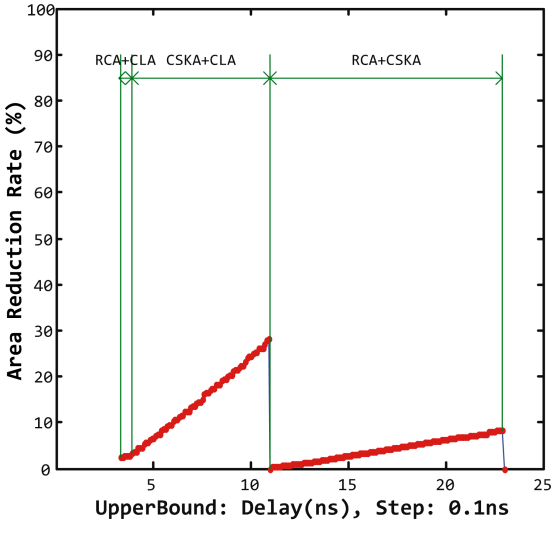
Fig. 5. A delay-area curve for a 128-bit heterogeneous adder

pareto-optimal points in these regions cannot be obtained without using heterogeneous adder architecture.

Finally, Fig. 6 shows the delay/area reduction rate obtained from the use of a heterogeneous adder. Delay/area reduction is observed in the interval where a heterogeneous adder interpolates a new design point effectively between two homogeneous adders. In these intervals, unless the heterogeneous adder is used, we have to use a homogeneous adder with extra cost. These intervals where delay/area reduction is observed are corresponding to the region covered by a heterogeneous adder in Fig. 3 given in Section 2. In this experiment, up to near 70% delay reduction and around 30% area reduction are obtained. However, we would like to note the improvement numbers are not absolute since this improvement



(a)



(b)

Fig. 6. Delay/area reduction rate by a 128-bit heterogeneous adder (a) Delay reduction rate, (b) Area reduction rate

is based on the assumption that the area/delay cost of a homogeneous adder we use is fixed. The improvement would be changed relatively if other circuit level optimizations like a transistor sizing are applied to homogeneous adders. Those area/delay variations of homogeneous adders are also shown in Fig. 4 as "range covered by design variations".

5 Conclusions and Future Work

For designing delay-area efficient adders, we have proposed a heterogeneous adder architecture, which consists of sub-adders of various sizes and different carry propagation schemes. The proposed decomposition into heterogeneous sub-adders allows more design tradeoffs in delay-area optimization. We also developed an ILP based technique and applied it iteratively to find an optimal configuration in selecting the type and bit-width of sub-adders as components of the heterogeneous adder.

Considering a 128-bit adder, we showed that many delay-area efficient designs could be found with the extended design space of heterogeneous adder architecture, which were not possible in a conventional, homogeneous adder design space using the proposed method with tools like Synopsis. Currently we are developing power modeling of heterogeneous adders using ILP and extending the design method to allow tradeoffs between delay, area, and power.

Acknowledgement

This work was supported in part by research funds from Chosun University, 2004 and by the Korea Research Foundation Grant funded by Korean Government (KRF-2005-D00026). The work of Jeong-Gun Lee was supported in part by IT Scholarship Program supervised by IITA & MIC, Republic of Korea.

References

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. V. Oklobdzija, B. Zeydel, S. Mathew, and R. Krishnamurthy, "Energy-Delay Estimation Technique for High-Performance Microprocessor VLSI Adders," *In Prod. IEEE Symposium on Computer Arithmetic*, pp. 272- 279, Jun. 2003
3. 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
4. V. Oklobdzija and D. Villeger, "Improving Multiplier Design by Using Column Compression Tree and Optimized Final Adder in CMOS Technology," *IEEE Trans. on VLSI*, 3(2):292-301, 1995.
5. P. F. Stelling and V. Oklobdzija, "Design Strategies for Optimal Hybrid Final Adders in a Parallel Multiplier," *Journal of VLSI Signal Processing*, Vol. 14(3) pp.321-31, 1996.
6. M.D. Ercegovac and T. Lang, "Digital Arithmetic," *Morgan Kaufmann Publishers*, 2004
7. R. Zimmermann, "Binary Adder Architectures for Cell-Based VLSI and their Synthesis," *PhD thesis*, Swiss Federal Institute of Technology (ETH) Zurich, Hartung-Gorre Verlag, 1998
8. Simon Knowles, "A Family of Adders," *In Prod. IEEE Symposium on Computer Arithmetic*, pp. 30-34, 1999

9. H.P. Williams, "Model Building in Mathematical Programming," *4th Ed., John Wiley*, New York, 1999
10. M. Berkelaar, "lp_solve - version 4.0," *Eindhoven University of Technology*, ftp://ftp.ics.ele.tue.nl/pub/lp_solve/, 2003
11. "IDEC-C221: IDEC Cell Library Data Book," *IC Design Education Center*, 2000
12. "DesignWare IP Family Reference Guide," *Synopsis Corporation*, September 12, 2005