Design Methodology of Variable Latency Adders with Multistage Function Speculation

Yongpan Liu Yinan Sun Yihao Zhu Huazhong Yang
Tsinghua National Laboratory for Information Science and Technology
EE Dept., Tsinghua University, Beijing, 100084, P.R. China
Email: {ypliu,yanghz}@tsinghua.edu.cn

Abstract— Increasing circuit delay range due to process variations, temperature and voltage fluctuations and input characterization makes the traditional worst-case fault-avoidance design methodology no longer sustainable. As an alternative, the average-case fault-detection design methodology is generating interest. Among existing solutions, function speculation design with error recovery mechanisms is quite promising due to its high performance and low area overhead. Previous work had focused on two-stage function speculation and thus lacked a systematic way to address the challenge of the multistage function speculation approach. For the first time, this paper proposes a multistage function speculation structure and applies it in a novel adder. We deduced the analytical performance and area models for the design and validated them in our experiments. Based on those models, a general methodology is presented to guide design optimization. Both analytical proofs and experimental results show that the proposed adder's delay and area has a logarithmic and linear relationship with its bit number,respectively. Compared with the DesignWare IP, the proposed adder provides the same performance with 6–16% area reductions under different bit number configurations.

Keywords— Multistage Function Speculation, Variable Latency Adder, Design Methodology

I. Introduction

Binary adders are one of the most frequently used arithmetic units, and are widely used in other advanced design units, such as multipliers and dividers. Previous work [1] had given the delay–area relationship in traditional adders, which indicated that no adders with traditional architectures can be implemented with sub-logarithmic delay. However, almost correct variable latency adders [2] can provide much better performance and energy efficient solutions. This is due to the fact that the worst-case design styles in traditional adders will lead to larger design margins, while variable latency adders permit the average-case design style and better performance. When a certain level of calculation error is acceptable – as in application areas such as data mining and machine learning – the variable latency adders can provide even better performance and smaller area.

Among the emerging variable latency adders, two kinds of architectures are proposed. The first one is the telescopic units and timing error-masking circuits, in which all timing errors are detected and removed in the design time. The shortcoming of this approach is that extensive computations are required to synthesize the exact function to cover all those input vectors causing timing violations. The complexity of the problem is NP-complete [3], whose usage are often limited to small or medium-size circuits. Furthermore, the synthesized functions [4] usually occupy a large portion of chip area (over 20%).

The second approach is function speculation, in which only part of input vectors causing timing violations are considered in the approximate function. Since speculative architectures may sometimes predict wrong logic values, error detection and recovery mechanisms [5] are needed for correct operation. This approach is more area efficient and easy to find the speculation function by covering only parts of the problematic input vectors. The penalty for the approximation is that the circuit needs more clock cycles to recover from a misprediction. It is obvious that the gain is positive as long as the mispredict ratio is below a certain threshold.

Previous works had studied how to use function speculation in circuit design. In an asynchronous design, Nowick [6] presented a low-latency adder with speculative completion. In synchronous circuits, Lu [7] proposed a method to replace a complete logic function with a simplified approximation circuit that mimics the original one. These techniques can effectively increase a processor's clock frequency. Chen et. al. [8] proposed a variable-latency adder to overcome the negative bias temperature instability by automatically shifting data capturing clock edge on critical timing paths. Verma et. al. [2] proposed a very fast adder under an extreme low function speculation error ratio (0.01%) with error detection units. Baneres et. al. [9] proposed a practical approach to synthesize two-stage variable latency circuits.

Although these works had provided various approaches for specific designs using function speculation, the general problems in speculative system design remain unsolved; these problems include analytical performance and area models for the multistage function speculation designs, design methodology for performance maximization, and the tradeoff between performance and area.

To our best knowledge, this work is the first to systematically solve design challenges in the structured circuits such as adders with multistage function speculation. Our contributions are listed as below:

1. We propose a multistage function speculation structure

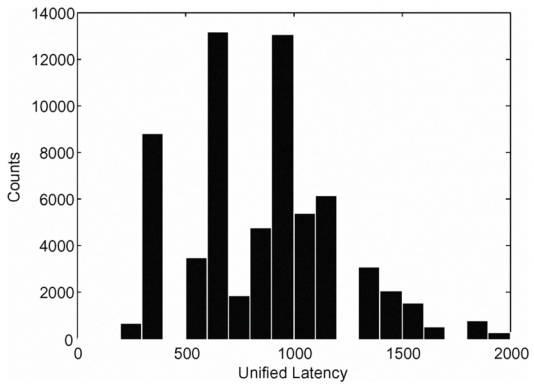

Fig. 1. Delay histogram of a 8-bit ripple carry adder

and apply it in a novel adder. We deduce the analytical performance and area models for the design and validate them in our experiments.

2. Based on the analytical models, we prove a theorem to show the necessary condition to acquire the optimal performance for the multistage function speculation adder under uniform input distribution. By giving out the average-case cycle expression of the optimal design, we show that there are two optimal regions in the design space and verify those observations by experiments.

3. By analytical proofs and experimental results, we show that the proposed adder's delay and area has a logarithmic and linear relationship with its bit number, respectively. Compared with the DesignWare IP, the proposed adder provides the same performance with 6-16% area reductions under different bit number configurations.

The rest of the paper is organized as follows. Section **2** describes the motivation of our work. We present our model formulation in Section **3**. The main contribution, the design methodology of adders with multistage function speculations, is described in Section **4**. Section **5** presents experimental results. We conclude in Section **6**.

II. Motivation

This section provides the motivation of the proposed variable latency adder with multistage function speculation. We first discuss the delay characterization of a ripple carry adder to show the necessity of the average-case design style. Furthermore, the advantages and challenges in adders with multistage function speculation are detailed.

A. Delay analysis

Given a uniform input distribution, the delay histogram of an 8-bit ripple carry adder is shown in Figure 1. The worst-case delay is almost one magnitude longer than that of the best case. It should be noted that the critical path is not activated in most cases except for very small portions of input patterns. i.e. the activated ratio of the critical path in Figure 1 is less than 1%. In fact, the active probability of effective carry chain in the ripple carry adder decreases exponentially with the length of the carry chain. Therefore, the average-case design style is quite promising to boost the adder's frequency and performance.

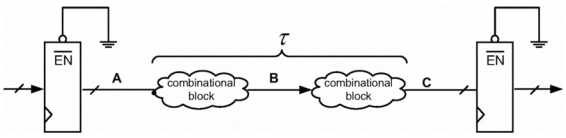

Fig. 2. Original fixed latency architecture

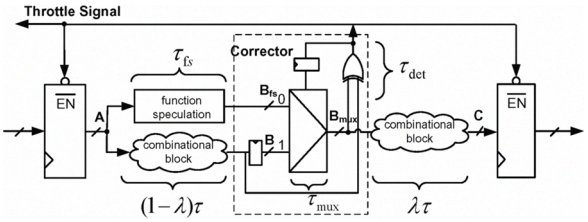

Fig. 3. Two-stage function speculation architecture

B. Multistage function speculation

This paper will mainly focus on the adder architecture with function speculations to realize the average-case performance. Figure 2 and 3 showed the original circuit and a circuit with two-stage function speculation. We denote the critical delay of the fixed latency circuit in Figure 2 as τ. The critical delay of the circuit with two-stage function speculation in Figure 3 is denoted as $\tau' = max\{(\tau_{fs} + \tau_{mux} + \lambda\tau), ((1-\lambda)\tau + \tau_{det})\}$. Generally, the value of τ_{fs}, τ_{mux} and τ_{det} are relatively small compared with the delay τ of combinational circuits, which consists of hundreds of logic gates. Therefore, the critical delay τ' of the circuit with two-stage function speculation can be reduced as much as half of the original delay τ in theory. The latency of the circuit is one clock cycle $1/\tau'$ when no speculation errors are detected. Once a speculation error is detected by the corrector, the latency of the circuit becomes two clock cycles $2/\tau'$. The throttle signal will be asserted until the correct values appear on the output. If the speculation error ratio is below a certain threshold, the variable latency circuits can obtain significant performance improvements compared with the original circuit. We call this architecture as two-stage function speculation. Baneres, et. al., [9] illustrated the variable latency design by two-stage function speculation in combinational circuits. The drawback of their design methodology is that it is only applicable for two-stage function speculation, which limits further reduction of the cycle time.

We show the architecture of a circuit with multistage function speculation in Figure 4. By inserting $n-1$ function speculation points into the critical path, the critical delay of the circuit with $m-1$ stages function speculation is equal to $\tau' = max\{(\tau_{fs,j-1}+\lambda_j\tau), (\tau_{fs,j-1}+\tau_{det})\}$ in which $j = 2, 3, ..., m$ and $\sum_{i=1}^{m} \lambda_i = 1$. As we can see, when the speculation delay $\tau_{fs,j-1}$ can be omitted compared with the original circuit delay τ and no speculation errors occurred, the theoretical lower bound of the circuit's optimal latency is τ/m. However, the circuit's latency with error recovery is μ times τ/m. The value of μ depends on the distribution

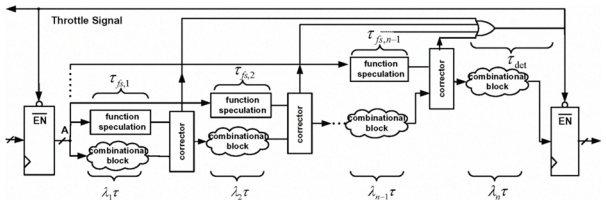

Fig. 4. Multistage function speculation architecture

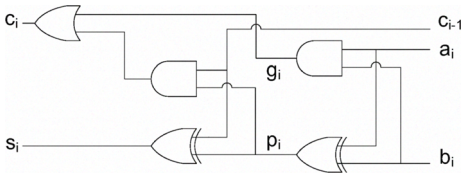

Fig. 5. Circuit diagram of a full adder

of speculation errors ranging from 2 to m.

As made clear by Figure 1, arithmetic units such as adders present large delay variations under different input patterns. It is quite promising to deploy multistage function speculation techniques in such circuits for performance improvements. This paper will try to attack the unsolved design challenges in the adder architecture with multistage function speculation.

III. Adder's Architecture and Model

This section will discuss the structure of the variable latency adder with multistage function speculation. We first describe the structure of a ripple carry adder, which is the basis of our proposed adder. Second, we illustrate the variable latency adder with multistage function speculation and related circuits. Third, we establish the adder's performance and area models.

A. Ripple carry adder's architecture

An N bit ripple carry adder consists of N 1-bit full adders, which is shown in Figure 5. a_i and b_i are the adder's inputs, c_{i-1} is the carry input, s_i is the sum output, and c_i is the carry output. By defining the intermediate generate signal $g_i = a_i b_i$ and propagate signal $p = a_i \oplus b_i$, s_i and c_i can be written as followings:

$$s_i = p_i \oplus c_{i-1} \quad (1)$$

$$c_i = g_i + p_i c_{i-1} \quad (2)$$

By cascading N 1-bit full adders in series (i.e., connecting the carry output of one adder to the next one's carry input), an N-bit ripple carry adder is constructed. Expanding the carry output c_N by p_i and g_i, we can get the following equation:

$$c_N = g_N + p_N g_{N-1} + p_N p_{N-1} g_{N-2} + \ldots + p_N p_{N-1} \ldots p_1 c_0 \quad (3)$$

Equation 3 shows that the carry input of (N-k)th full adder would affect the final carry out c_N, if and only if the following two equations are satisfied.

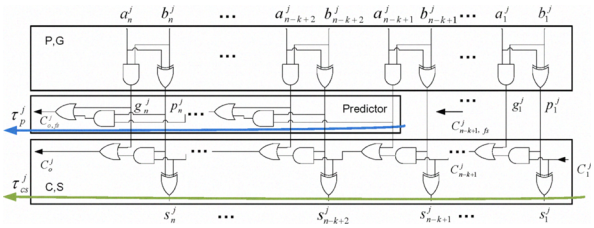

Fig. 6. A n-bit ripple carry adder with k-bit predictor

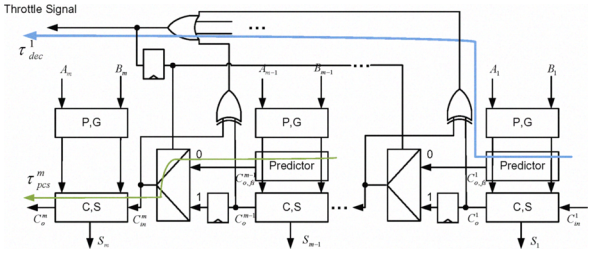

Fig. 7. Multistage function speculation adder diagram

$$g_N + p_N g_{N-1} + \ldots + p_N p_{N-1} \ldots p_{N-k+2} g_{N-k+1} = 0 \quad (4)$$

$$p_N p_{N-1} \ldots p_{N-k+1} = 1$$

where k is the bit number of the predictor. Given a uniform input distribution, the valid probability of Equation 4 decreases exponentially with the increment of k. Therefore, it is possible to speculate the final carry output, with a few full adders' inputs, with a low error ratio.

B. Multistage function speculation adder

We will illustrate the proposed multistage function speculation adder in details. Suppose the original N-bit carry ripple adder consists of m-stage n-bit sub adders, in which $N = m * n$. We first discuss the standard structure of an n-bit sub ripple carry adder with a k-bit predictor. After that, we describe the multistage function speculation adder including m sub adders. Finally, the work flow of the adder will be explained.

Figure 6 shows the n-bit ripple carry adder with a k-bit predictor, which is one stage of the proposed adder. As we can see, this block includes three parts: the PG unit for propagate and generate signals, the CS unit for carry and sum signals, and the predictor. The PG and CS units are the same as those in the ripple carry adder. The k-bit predictor copies the kth right side carry chain of the original adder. Supposing the block is the jth stage in the m-stage function speculation adder, we denote the inputs of the PG unit as $A_j = (a_1^j, a_2^j, \ldots, a_n^j)$ and $B_j = (b_1^j, b_2^j, \ldots, b_n^j)$. The carry output and input of the CS unit are C_o^j and C_{in}^j, respectively. The output of the predictor is $C_{o,fs}^j$ and the sum output is $S_j = (s_1^j, s_2^j, \ldots, s_n^j)$.

In Figure 7, the m-stage function speculation adder consists of m blocks. The predictor in each stage will speculate a carry output $C_{o,fs}^j$ (the block in the last stage does not have a predictor) and this signal will be given to its next

stage as a carry input via a multiplex. The original carry output C_o^j will also be given out to the next stage via the same multiplex. The select signal of the multiplex is decided by the flip-flop, which stores the comparison result to indicate if the speculation is correct, or not.

As Figure 4 demonstrates, the clock cycle is limited by the longest stage delay, which is much shorter than the original carry chain delay. In most cases, the adder can finish its calculation in one cycle when the speculation value in each stage is correct. In other cases, the predictor produces some erroneous values. The errors can be detected by the XOR gate, which compares the speculated carry output $C_{o,fs}^j$ with the exact carry output C_o^j. In such cases, the adder has to spend 2 to m clock cycles to complete the calculation. Under the worst case, m clock cycles are needed to generate the correct results at the output, since there are $m-1$ predictors to be verified and corrected. The following section will characterize the adder's performance when certain speculation errors occur.

C. Performance model

The average delay of m-stage function speculation can be described as below:

$$\tau_{ave} = \sum_{\mu=1}^{m} P_\mu(\mu\tau_{cyc}) \tag{5}$$

where τ_{cyc} is the length of the adder's clock cycle, P_μ is the probability of the adder to complete the calculation in μ cycles. The clock cycle τ_{cyc} is decide by the critical path delay in the multistage function speculation. We can obtain the expression of τ_{cyc} as follows:

$$\begin{aligned} \tau_{cyc} &= max\{\tau_{pcs}^i, \tau_{dec}^i\}, i = 1, 2, \ldots, m \\ \tau_{pcs}^i &= \tau_{cs}^i + \tau_{mux}^{i-1} + \tau_p^{i-1} \\ \tau_{dec}^i &= \tau_{cs}^i + \tau_{xor}^i + \tau_{or} \end{aligned} \tag{6}$$

As Figures 6 and 7 have shown, τ_{cs}^i is the critical delay of the ith CS unit; τ_p^i is the critical delay of the ith predictor; and τ_{mux}, τ_{xor} and τ_{or} are the delay of logic gates, respectively.

Before deducing the probability P_μ, we first define several terms. A conflict occurs when the XOR gate detects an inconsistent output value between the predictor and the CS unit. A transfer occurs when the input change makes the output of the CS unit flip. The adder is settled when no conflicts exist. The probability of conflict and transfer occurrence in the ith stage are denoted as α_i and β_i, respectively. We will discuss how to characterize the value of α and β in the adder. As Figure 6 has shown, the carry input for the jth k-bit predictor $C_{n-k+1,fs}^j$ is zero. A conflict will occur if and only if the corresponding carry input C_{n-k+1}^j for the CS unit is one and both values can propagate to the outputs of the predictor and CS unit. Based on Equation 4,the confliction condition can be expressed as follows:

$$\begin{aligned} C_{n-k+1}^j &= 1; \\ p_i &= 1, n \geq i \geq n-k+1; \\ g_i &= 0, n \geq i \geq n-k+1; \end{aligned} \tag{7}$$

Assuming A and B are in the uniform distribution and have no correlations, the probability $P(p_i = 1, g_i = 0)$ is 2^{-1} and $P(C_{n-k+1}^j = 1)$ is also 2^{-1}. Thus,the conflict probability α_i of a k-bit predictor in the ith stage can be expressed as:

$$\alpha_i = 2^{-1} * 2^{-k} \tag{8}$$

Similarly, the transfer probability β_i in the ith stage with a n-bit CS unit can be obtained as follows:

$$\beta_i = 2^{-n} \tag{9}$$

Now, we will deduce the probability P_μ of the adder to complete the calculation in μ cycles. Given a m-stage function speculation adder, the ith stage will generate a conflict with the probability α_i. The conflict can be transferred to the next stage with the probability β_i. We assume that each stage generates a conflict and transfers the conflict to the next stage independently. Experimental results in Section **5** will validate this assumption is reasonable. Let us focus on the ith stage, its conflict can be transferred to the following $m-i$ stages. The probability of this conflict to be eliminated within μ cycles can be expressed by the following equations:

$$P(C_i > \mu) = \alpha_i\beta_{i+1}\beta_{i+2}\ldots\beta_{i+\mu-1} \tag{10}$$

$$P(C_i \leq \mu) = 1 - \alpha_i \prod_{j=i+1}^{i+\mu-1} \beta_j \tag{11}$$

where C_i is the cycle number for the ith stage to eliminate its conflict and $i+\mu$ is always less than m because there are at most $m-i$ stages for the conflict of the ith stage to be transferred. Considering all stages are independent, the probability for the adder to eliminate all conflicts within μ cycles is

$$\begin{aligned} P(C \leq \mu) &= \prod_{i=1}^{m-\mu} P(C_i \leq \mu) \\ &= \prod_{i=1}^{m-\mu} (1 - \alpha_i \prod_{j=i+1}^{i+\mu-1} \beta_j) \end{aligned} \tag{12}$$

where C represent the cycle number for the adder to eliminate all conflicts. According to Equation 6 and 12, Equation 5 can be rewritten as:

$$\begin{aligned} \tau_{ave} &= \sum_{\mu=1}^{m} P(C = \mu)(\mu\tau_{cyc}) \\ &= \sum_{\mu=1}^{m} (P(C \leq \mu) - P(C \leq \mu - 1))(\mu\tau_{cyc}) \\ &= (m - \sum_{\mu=1}^{m-1} \prod_{i=1}^{m-\mu} (1 - \alpha_i \prod_{j=i+1}^{i+\mu-1} \beta_j))\tau_{cyc} \end{aligned} \tag{13}$$

When each stage uses the same circuit configuration and α_i,β_i can be denoted as α and β, Equation 13 can be written as

$$\tau_{ave} = (m - \sum_{\mu=1}^{m-1} (1 - \alpha\beta^{\mu-1})^{m-\mu})\tau_{cyc} \tag{14}$$

When $m = 3$, we have $\tau_{ave} = (1 + 2\alpha + \alpha\beta - \alpha^2)\tau_{cyc}$.

D. Area model

As Figure 4 has shown, the proposed m-stage adder with k-bit predictors consists of $N = m * n$ full adders, $m-1$ correctors and $m-1$ predictors. Due to the adder's regular structure, the area of the circuit is proportional to its cell number, which can be expressed by the following equation:

$$A_{msad} = NA_{fa} + (m-1)A_{cor} + (k-1)(m-1)A_{pre} \tag{15}$$

TABLE I

TYPICAL AREA PARAMETERS FOR A 128-BIT 16-STAGE WITH 4-BIT PREDICTOR MULTISTAGE FUNCTION SPECULATION ADDER

Unit name	Unit number	Area per unit	Area ratio
Full adder A_{fa}	128	86.5 $\mu m^2/bit$	80%
Predictor A_{pre}	45	19.9 $\mu m^2/bit$	18%
Corrector A_{cor}	15	103 $\mu m^2/stage$	2%

where A_{msad} is the area of the multistage function speculation adder, A_{fa} is the area of a full adder, A_{cor} is the area of a corrector and A_{pre} is the area of a predictor. The typical unit area is shown in Table I and the values are extracted based on the $0.18\mu m$ HJTM process. The total area of a 128-bit 16-stage function speculation adder with 4-bit predictor is $13157\mu m^2$. The predictor and the corrector occupy 18.1% of the total area.

IV. Design Methodology

Based on the models in Section **3**, we will illustrate a design methodology for multistage function speculation adders under different configurations.

A. Optimal performance design

Given an N-bit multistage function speculation adder, this paragraph will decide the stage number m and the bit number k_i for each predictor to maximize the performance. In order to gain tractability, we remove the predictor area as a factor and suppose that K, the total number of predictor bits, is a constant as follows:

$$K = \sum_{i=1}^{m-1} k_i \tag{16}$$

We now give a theorem for optimal performance design.

Theorem 1: For an m-stage N-bit multistage function speculation adder with total K-bit predictor under uniform input distribution, the bit number of each stage and predictor should be N/m and $K/(m-1)$, respectively. This condition is necessary to obtain optimal performance under the first order approximation.

Proof 1: As $\beta_i \le \alpha_i$ $(k \le n)$, we can obtain the first order approximation of Equation 13 as followings:

$$\begin{aligned}\tau_{ave} &= \sum_{\mu=1}^{m} P(C=\mu)(\mu\tau_{cyc}) \\ &= (m - \sum_{\mu=1}^{m-1}\prod_{i=1}^{m-\mu}(1-\alpha_i\prod_{j=i+1}^{i+\mu-1}\beta_j))\tau_{cyc} \\ &\approx (1+\sum_{i=1}^{m-1}\alpha_i)\tau_{cyc} \end{aligned} \tag{17}$$

The optimal performance will be acquired when both τ_{cyc} and $(1+\sum_{i=1}^{m-1}\alpha_i)$ reach their minimal values simultaneously. Equation 6 has shown that τ_{pcs} is usually much larger than τ_{dec} because $\tau_p^{i-1} \gg (\tau_{xor}^i + \tau_{or})$. Therefore, the τ_{cyc} can be expressed as below:

$$\begin{aligned}\tau_{cyc} &= \max(\tau_{cs}^i + \tau_{mux}^{i-1} + \tau_p^{i-1}) \\ &= \max(n_i\tau_{cso} + \tau_{mux} + k_i\tau_{po}) \end{aligned} \tag{18}$$

where n_i and k_i is the bit number of the ith stage and predictor, respectively. τ_{cso} and τ_{po} is the 1-bit unit delay of CS unit and predictor. Since $N = \sum_{i=1}^{m} n_i$ and $K = \sum_{i=1}^{m-1} k_i$, the minimal of τ_{cyc} is obtained:

$$\tau_{cyc} \ge (N\tau_{cso}/m + K\tau_{po}/(m-1) + \tau_{mux}) \tag{19}$$

when $n_i = N/m$ and $k_i = K/(m-1)$. Next, we will show that this condition will also minimize $(1+\sum_{i=1}^{m-1}\alpha_i)$.

$$(1+\sum_{i=1}^{m-1}\alpha_i) \ge 1 + (m-1)\sqrt[m-1]{\prod_{i=1}^{m-1}\alpha_i} \tag{20}$$

The equal relationship is satisfied when $k_i = K/(m-1)$ and $\alpha_i = 2^{-K/(m-1)+1}$ under the uniform input distribution. Therefore, the optimal average-case delay can be given:

$$\begin{aligned}\tau_{ave} &= \max(n_i\tau_{cso} + \tau_{mux} + k_i\tau_{po})(1+\sum_{i=1}^{m-1}\alpha_i) \\ &\ge (\frac{N}{m}\tau_{cso} + \frac{K}{m-1}\tau_{po} + \tau_{mux}) \cdot \\ &\quad (1+(m-1)\sqrt[m-1]{\prod_{i=1}^{m-1}\alpha_i}) \\ &\ge (\frac{N}{m}\tau_{cso} + \frac{K}{m-1}\tau_{po} + \tau_{mux}) \cdot \\ &\quad (1+(m-1)2^{-K/(m-1)+1}) \end{aligned} \tag{21}$$

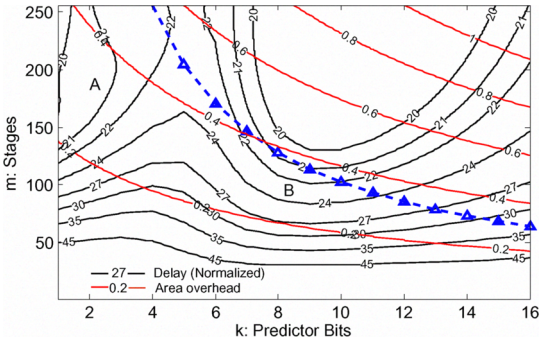

Fig. 8. Analytical performance and area curves under different k and m configurations

B. Performance and area trends

Based on Equation 21 and 15, we can plot the delay and area of a 1024-bit multistage function speculation adder under different m parameters in Figure 8. We assume that both 1-bit full adder and multiplex critical path delay are 1 unit. The area overheads are normalized with a 1024-bit ripple carry adder. The black and red line separately represents the equal delay and area overhead under different k and m configurations. When we restrict $k \le n$, the valid design region is below the dotted line with triangle marks in Figure 8. To our surprise, there are two regions: A and B for the optimal performance design candidates. Our experimental results in Section **5** will validate this observation and show their different behaviors. Furthermore, we will give a theorem on the delay and area trend of the multistage function speculation adder with the adder's bit number N.

Theorem 2: Given the total predictor bit K and the adder's bit number N, the average-case delay of optimal performance multistage function speculation adder is proportional to $O(\log_2 N)$ and the adder's area is proportional to $O(N)$.

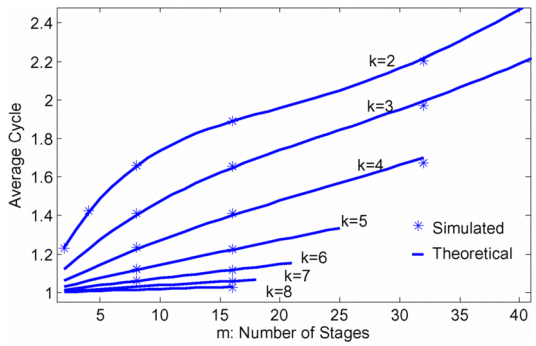

Fig. 9. Average cycles under different k and m configurations

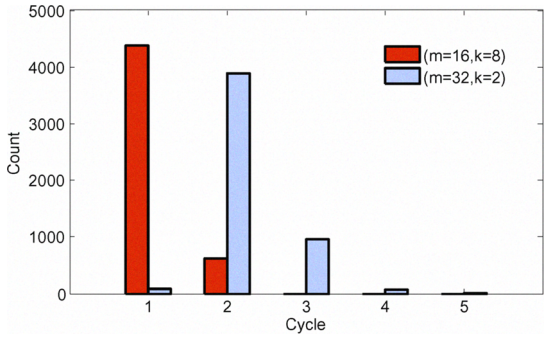

Fig. 10. Histogram of execution cycle number in two regions

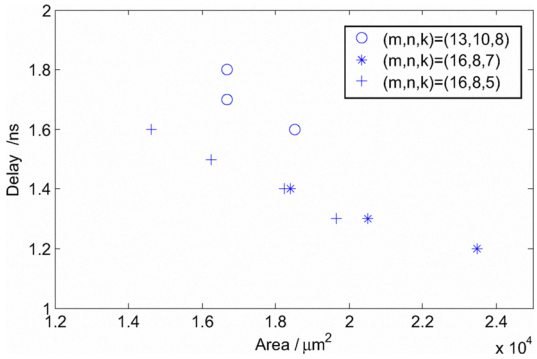

Fig. 11. Pareto delay and area curve of multistage function speculation adder

Theorem 2 is proven in [10]. It is omitted here due to the page limit.

V. Experiments

In this section, we explain our experimental setup and then verify the proposed models. Second, we analyze adder behavior in different design regions. Third, we show the delay and area relationship of the proposed adder using different stage and predictor configurations. Finally, we compare the proposed adder with other existing adders.

A. Experimental Setup

We use Verilog and DesignWare library to implement the multistage function speculation adder and the reference adder. The circuits are synthesized using a standard-cell library for HJMC 0.18μm CMOS technology. We generate uniform distributed 5000 random inputs and use ModelSim™to catch the speculation errors for performance model analysis. The bit number of the multistage function speculation adder ranges from 128 to 1024.

B. Performance model validation

Figure 9 shows average cycles under different k and m configurations. The solid line is plotted based on Equation 13. The star points are obtained by simulating the specific adder under 5000 random input data. As we can see, the values predicted by our performance models fit well with the experimental results. It verifies that our assumption that each stage generates a conflict and transfers the conflict to the next stage independently is reasonable. Furthermore, the average cycle number decreases when m decreases and k is held constant. This drop can be explained by noting that when k is held constant, the error probability of each stage remains unchanged – this means the lower stage number, m, will lead to a smaller average cycle number. However, the cycle number increases when m is held constant and k decreases. When k decreases, the speculation errors rise thus the average cycle number does so as well.

C. Behavior in different design regions

As Figure 8 has shown, there are two design regions A and B for optimal performance candidates of a 1024-bit adder. In region A, a small predictor bits and low stage numbers are used: the clock frequency can be quite high. Region B tends to use more predictor bits for higher precisions with less stages. We synthesized those two designs and profiled their execution cycle number in Figure 10. As we can see, it will cost $2-3$ cycles for design A to complete one operation in a 1024-bit adder, while design B spent $1.001-1.01$ cycle number for the same operation with much lower clock frequency. Based on the Design Compiler's results, the energy per operation for design A and B is 69.3 pJ and 129.8 pJ, respectively. Although both designs provide the same performance, design B is 1.87x more power efficient than design A due to the lower frequency and less operation cycles. Therefore, region B should be more preferable for low power applications.

D. Performance and area tradeoff

Figure 11 shows the clock frequency and area based on the m and k configurations given by Figure 8. For each configuration, we used Design Compiler to synthesize the adder under different area constraints. As previous analysis indicates that region B is preferable, we picked up two configurations of region B in the Pareto frontier and one configuration in the sub-optimal region. As we can see, the optimal performance of configuration (m=16,n=8,k=7) is better than configuration (m=16,n=8,k=5). The sub-optimal configuration (m=13,n=10,k=8) is always dominated by those configurations in the Pareto frontier. It validated that our design methodology can guide the multistage function speculation adder design effectively.

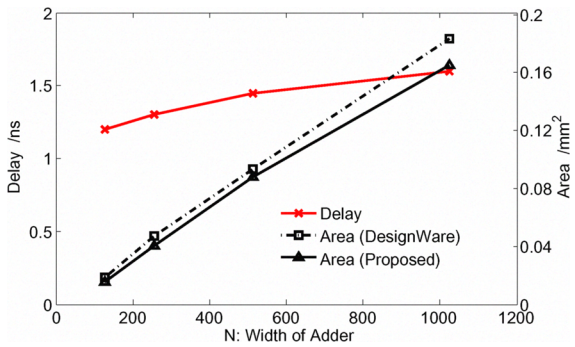

Fig. 12. Area and delay of DesignWare and multistage function speculation adder

E. Comparison with other adders

Figure 12 shows the delay and area of adders with different bit numbers from 128 to 1024. The plotted delay is for both the multi stage variable latency speculative adder and for the adder generated by Design Ware; i.e., the proposed adder and the reference adder are compared for area at equal delay points. As we can see, delay for the proposed adder obeys an $\log_2 N$ trend with the adder's bit number N. It should be noted that the delay for traditional adders is obtained by synthesis tools for optimal performance, which includes the register delay. While the delay for adders with function speculation is defined as the average-case delay. It means that the original clock delay of multistage function speculation adder would be multiplied by a constant (> 1) due to possible multiple cycle executions for fair comparison. The area values of the proposed architecture are $6 - 16\%$ smaller than those of the fastest DesignWare adders ranging from 128 to 1024. As we can see, the delay and area trends with bit number N declared in Theorom 2 are also validated in Figure 12. Besides the above advantage, when a certain level of calculation error is allowed the proposed adder can provide much better performance and smaller area. These almost correct units are quite promising for algorithm level fault-tolerant calculations, such as data mining and machine learning.

VI. Conclusions

Multistage function speculation design is quite promises to offer superior average-case performance. In this paper, we propose performance and area models for multistage function speculation adders, based on which, a general methodology is presented to guide design optimization. Both analytical and experimental results validate our models and the design methodology. Experiments showed that the proposed adder's delay and area has a logarithmic and linear relationship with its bit number, respectively. Compared with the DesignWare IP, the proposed adder provides the same performance with 6–16% area reductions under different bit number configurations.

VII. Acknowledgement

The authors would like to thank John P. Stevenson, Stanford University, for his helpful discussions and suggestions to improve the paper. This work was supported in part by the 863 Program award under 2009AA01Z130 and in part by the NSFC awards under 60976032.

References

[1] I. Koren, *Computer Arithmetic Algorithms*, Prentice-Hall Inc., New Jersey, 1993.

[2] A.K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: a new paradigm for arithmetic circuit design," in *Proc. Design, Automation and Test in Europe Conf*, 2008, pp. 1250–1255.

[3] P.C. McGeer and R.K. Brayton, *Integrating functional and temporal domains in logic design: the false path problem and its implications*, Kluwer Academic Publishers, 1991.

[4] Y.S. Su, D.C. Wang, S.C. Chang, and M. Marek-Sadowska, "An efficient mechanism for performance optimization of variable-latency designs," in *Proc. Design Automation Conf.* ACM New York, NY, USA, 2007, pp. 976–981.

[5] D. Ernst, N.S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, and K. Flautner, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. Int. Symp. Microarchitecture*, pp. 7–18.

[6] S.M. Nowick, "Design of a low-latency asynchronous adder using speculative completion," *IEE Proc. Computers and Digital Techniques*, vol. 143, no. 5, pp. 301–308, 1996.

[7] S.L. Lu, "Speeding up processing with approximation circuits," *Computer*, pp. 67–73, 2004.

[8] Y. Chen, H. Li, J. Li, and C.K. Koh, "Variable-latency adder (VL-adder): new arithmetic circuit design practice to overcome NBTI," in *Proc. Int. Symp. Low Power Electronics and Design.* ACM New York, NY, USA, 2007, pp. 195–200.

[9] D. Baneres, J. Cortadella, and M. Kishinevsky, "Variable-latency design by function speculation," in *Proc. Design, Automation and Test in Europe Conf*, pp. 1704–1709.

[10] Y. Sun, Y. Liu, Y. Zhu, and H. Yang, "Variable-latency adders using multistage function speculation," in *Technical Report*, 2009.