Proceedings of the 10th WSEAS International Conference on CIRCUITS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp318-323)

Statistical Makespan Analysis in Asynchronous Datapath Synthesis*

KOJI OHASHI & MINEO KANEKO
School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292
JAPAN

Abstract: This paper proposes statistical schedule length (makespan) analysis for evaluating a schedule and a datapath during asynchronous datapath synthesis. In order to handle the randomness of delay variation mathematically, the execution time of each operation is modeled by a stochastic variable, and an algorithm to calculate the distribution of total computation time of a scheduled asynchronous datapath is presented. The proposed statistical analysis handles three correlations; (C1) correlation between delays on different modules and nets, (C2) structural correlation (re-convergent paths in a scheduling graph), and (C3) correlation induced by resource sharing.

Key–Words: Asynchronous system, Datapath synthesis, Statistical analysis, Asynchronous schedule

1 Introduction

To design a cost effective high performance asynchronous system for a specified application, optimization of a datapath in register transfer level is an important design step. Scheduling and resource binding (assignment) are major subtasks in datapath synthesis not only for synchronous systems but also for asynchronous systems. Several synthesis systems for asynchronous systems have been proposed [1, 2, 3].

In the evaluation phase (of a schedule and a datapath) of these systems, they introduce a specious constant delay for the execution time of each operation, such as typical delay, maximum delay or minimum delay, and compute typical (maximum or minimum) makespan by using typical (maximum or minimum) execution delay. If there is no random delay variation between modules and delays of all modules vary uniformly, the above constant delay model seems to be an acceptable way to handle delay variations. However, due to local supply noise, local variations of temperature, crosstalk between wires, local manufacturing imperfections, etc., functional delay and transmission delay can vary easily, and the calculation of minimum, typical, and maximum total computation time based on the constant delay model is unacceptable for those random delay variations. For a synchronous system, the effect of delay variations is masked by the clock period and delay margins. On the other hand, for an asynchronous system, it affects the total computation time of an application directly.

*A preliminary version of this paper has appeared in [7].

In this paper, we propose a statistical schedule length (makespan) analysis method for evaluating a schedule and a datapath during asynchronous datapath synthesis. We model the execution time of each operation as a stochastic variable having the normal distribution, and handle three correlations; (C1) correlation between delays on different modules and nets, (C2) structural correlation (re-convergent fanouts), and (C3) correlation induced by resource sharing (depends on resource binding).

We present an algorithm to calculate the distribution of total computation time of an application algorithm considering correlations (C1), (C2) and (C3) under given a schedule and resource assignment. The proposed algorithm can be incorporated into a synthesis system to synthesize asynchronous datapaths having optimized statistical performance.

2 Scheduling Graph in Datapath Synthesis

A "scheduling graph" $G_S = (V_S, A_S)$ is a graph which represents precedence constraints between operations (see Fig. 1(a)). The precedence relations in G_S come from two different sources, one is mandatory precedence relation specified by a target application algorithm, and the other is optional precedence relation brought by scheduling. The arcs representing the latter are called "disjunctive arcs" in this paper. V_S is the set of the start and the end of operations, and we denote the start node and the end node of each opera-

Proceedings of the 10th WSEAS International Conference on CIRCUITS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp318-323)

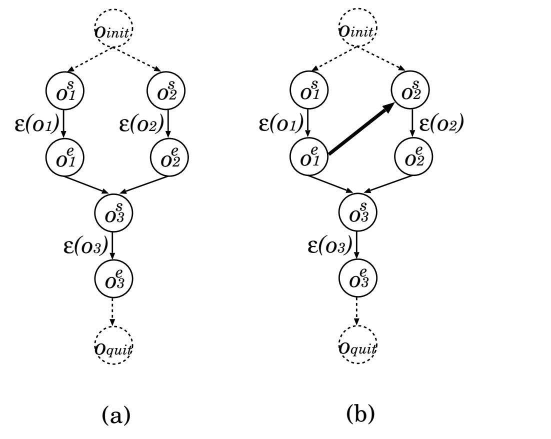

Figure 1: (a) Scheduling graph, and (b) scheduling graph having a disjunctive arc.

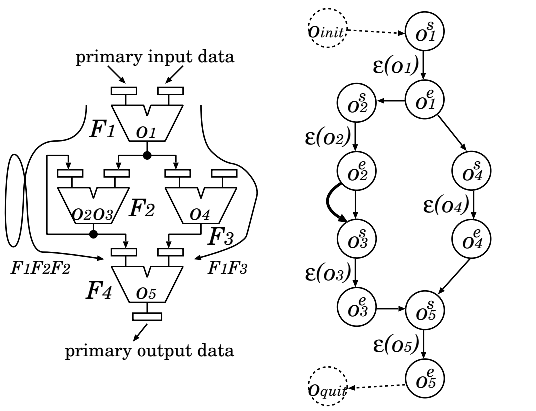

Figure 2: Correlations induced on a datapath.

tion o_i as o_i^s and o_i^e, respectively. A_S is the set of arcs which reflect precedence constraints, and each arc has variable weight. The weight of an arc (o_i^s, o_i^e) corresponds to the execution time of operation o_i, which is denoted by $\varepsilon(o_i)$. For the other arcs, the weights of them are 0.

The disjunctive arcs are introduced in G_S to avoid a collision between operations or between data. That is, if two operations (data) are assigned to the same functional unit (register), then one lifetime precedes the other. This constraint can be resolved by adding disjunctive arcs to G_S. Fig 1(b) shows an example of the scheduling graph, which is obtained by adding a disjunctive arc (o_1^e, o_2^s) to the scheduling graph in Fig 1(a). By the disjunctive arc (o_1^e, o_2^s), we intend that the execution of o_2 starts after the execution of o_1 is completed. Note that different decisions yield different sets of disjunctive arcs (i.e., different schedules), and to find an optimum set is assignment constrained scheduling problem [3]. The optimality of the schedule depends on the evaluation of the schedule.

3 Statistical Delay Model

Figure 2 shows a simple demonstrative example. The right hand side shows a scheduling graph, and the left hand side shows a datapath architecture. Basically, the execution time $\varepsilon(o_i)$ of each operation o_i is given by the functional unit F_j to which o_i is assigned, and is considered to vary randomly. When we compute the distribution of the path length of path $o_1^s o_1^e o_2^s o_2^e o_3^s o_3^e$ (which corresponds to the walk $F_1F_2F_2$ in the datapath architecture), we will take account of the correlation in type (C1) between $\varepsilon(o_1)$ and $\varepsilon(o_2)$, and between $\varepsilon(o_1)$ and $\varepsilon(o_3)$, which reflects the correlation of the performances of F_1 and F_2. We will also take account of the correlation in type (C3) between $\varepsilon(o_2)$ and $\varepsilon(o_3)$, which reflects the correlation of the performances for different inputs and at different time instances on the same functional unit F_2. On the other hand, when we look at two paths $o_1^s o_1^e o_2^s o_2^e o_3^s o_3^e$ and $o_1^s o_1^e o_4^s o_4^e$, we will take account of the correlation in type (C2) between path lengths of those paths, since both path lengths include $\varepsilon(o_1)$ in common.

In this paper, the execution time $\varepsilon(o_i)$, which is dependent on the performance of the functional unit to which o_i is assigned, is modeled by a stochastic variable having the normal distribution as done in [5, 6], and the schedule length of a schedule is also considered to be distributed randomly. In what follows, the weight of each arc e in A_S is denoted by $w(e)$, and the normal distribution of $w(e)$ is denoted by $N(\mu(e), \sigma^2(e))$, where $\mu(e)(= E[w(e)])$ and $\sigma^2(e)(= V[w(e)])$ are the mean and the variance of the distribution, respectively. For two arcs $e_i = (o_i^s, o_i^e)$ and $e_j = (o_j^s, o_j^e)$ in A_S, we consider the correlation coefficient $\rho(e_i, e_j)(= R[w(e_i), w(e_j)])$ between $w(e_i)$ and $w(e_j)$. $\rho(e_i, e_j)$ reflects the correlation of performances of two functional units to which o_i and o_j are assigned. Operations o_i and o_j may share the same functional unit, and in such case, we can tune $\rho(e_i, e_j)$ to become larger than that for non-sharing case if necessary. Let $l(v)$ be the longest path length from the source node o_{init} to a node v in a scheduling graph G_S.

Here we consider the problem to find the mean $E[l(o_{quit})]$ and the variance $V[l(o_{quit})]$ of $l(o_{quit})$ of the sink node o_{quit} in G_S.

4 Statistical Schedule Length Analysis

4.1 Overview

In this paper, we propose an algorithm which computes the mean and the variance of the longest path length from o_{init} to each node v in topological order of

Algorithm: Statistical Schedule Length Analysis

Step 1: Depending on the binding, $\mu(e)$ and $\sigma^2(e)$ are assigned to each arc e in G_S. In addition, $\rho(e_i, e_j)$ is assigned to every pair of arcs e_i and e_j.

Step 2: All nodes in G_S are sorted in topological order.

Step 3: If $l(o_{quit})$ is computed, then the mean $E\left[l(o_{quit})\right]$ and the variance $V\left[l(o_{quit})\right]$ are outputted. Otherwise, a node v is selected from G_S in topological order.

Step 4: If $v = o_{init}$, we set $E[l(v)] = V[l(v)] = 0$ and the correlation coefficients $R[l(v), l(v)] = 1$ and $R[l(v), w(e)] = 0$ for each arc $e \in A_S$, respectively, and go to Step 3. Otherwise, for each incoming arc of v $e_i = (u_i, v)$ $(i = 1, 2, \cdots, f(v))$, we calculate the mean $E[l_i^t(v)]$ and the variance $V[l_i^t(v)]$ of $l_i^t(v)$ using the probabilistic equivalent of $l_i^t(v) = l(u_i) + w(e_i)$ ("ADD" operation).

Step 5: For each node $x \in V_S$ whose $l(x)$ has been already computed, we calculate the correlation coefficient $R\left[l_i^t(v), l(x)\right]$ between $l_i^t(v)$ and $l(x)$.
Similarly, for each arc $y \in A_S$ we calculate the correlation coefficient $R\left[l_i^t(v), w(y)\right]$ between $l_i^t(v)$ and $w(y)$.

Step 6: We recursively calculate the mean $E[l_i(v)]$ and the variance $V[l_i(v)]$ of $l_i(v)$ using the probabilistic equivalent of $l_i(v) = \max\left[l_{i-1}(v), l_i^t(v)\right]$ for $i \geq 2$ and $l_1(v) = l_1^t(v)$ ("MAX" operation).
In addition, the correlation coefficients $R[l_i(v), l(x)]$ and $R[l_i(v), w(y)]$ between $l_i(v) = \max\left[l_{i-1}(v), l_i^t(v)\right]$ and $l(x)$ of each node x whose $l(x)$ has been already computed and $w(y)$ of each arc y, respectively, are calculated.
By computing $E[l_i(v)]$, $V[l_i(v)]$, $R[l_i(v), l(x)]$ and $R[l_i(v), w(y)]$ recursively, we have the mean $E[l(v)] = E[l_{f(v)}(v)]$ and the variance $V[l(v)] = V[l_{f(v)}(v)]$ of $l(v)$, the correlation coefficient $R[l(v), l(x)] = R[l_{f(v)}(v), l(x)]$ between $l(v)$ and $l(x)$ of each node x whose $l(x)$ has been already computed, and the correlation coefficient $R[l(v), w(y)] = R[l_{f(v)}(v), w(y)]$ between $l(v)$ and $w(y)$ of each arc y. Go to Step 3. ■

Figure 3: Statistical schedule length analysis algoritm.

the precedence relation given by G_S. The algorithm also takes account of three types of correlation (C1), (C2) and (C3).

Figure 3 shows our proposed algorithm, in which $l_i^t(v)$ denotes the longest path length from o_{init} to v passing through the ith incoming arc of v, and $l_i(v)$ denotes the longest path length from o_{init} to v passing through one of 1st to ith incoming arcs of v, that is $l_{f(v)}(v) = l(v)$, where $f(v)$ is the in-degree of v.

The time complexity of the proposed algorithm is evaluated as follows. For each node v selected in topological order, the mean $E[l(v)]$, the variance $V[l(v)]$, and the correlation coefficients $R[l(v), l(x)]$ and $R[l(v), w(y)]$ for each node and each arc in $G_S = (V_S, A_S)$ are calculated, and thus the computation time required in Step 5 and 6 is $\mathcal{O}(f(v) \cdot |A_S|)$, where $f(v)$ is the number of incoming arcs of a node v. Since $\sum_{v \in V_S} f(v) = |A_S|$, the total time complexity is $\mathcal{O}(|A_S|^2)$.

4.2 Maximum of Two Stochastic Variables

Given two stochastic variables ξ and η having the normal distributions $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$, respectively, and the correlation coefficient $R[\xi, \eta] = \rho$ between ξ and η, the mean $E[\nu]$ and the variance $V[\nu]$ of the stochastic variable $\nu = \max[\xi, \eta]$ are obtained as follows [4]. Unless $\sigma_1 - \sigma_2 = \rho - 1 = 0$,

$$\begin{aligned} E[\nu] &= \mu_1\Phi(\alpha) + \mu_2\Phi(-\alpha) + a\varphi(\alpha), \qquad (1) \\ V[\nu] &= (\mu_1^2 + \sigma_1^2)\Phi(\alpha) + (\mu_2^2 + \sigma_2^2)\Phi(-\alpha) \\ &\quad + (\mu_1 + \mu_2)a\varphi(\alpha) - E^2[\nu], \qquad (2) \end{aligned}$$

where

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right), \ \Phi(x) = \int_{\infty}^{x} \varphi(t)dt, \quad (3)$$

$$a = \sqrt{\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho}, \ \alpha = \frac{\mu_1 - \mu_2}{a}. \quad (4)$$

Given a stochastic variable τ having the normal distribution, and correlation coefficients $R[\xi, \tau] = \rho_1$ and $R[\eta, \tau] = \rho_2$, the correlation coefficient $R[\tau, \nu]$ between τ and $\nu = \max[\xi, \eta]$ is obtained as follows [4].

$$R[\tau, \nu] = \frac{\sigma_1\rho_1\Phi(\alpha) + \sigma_2\rho_2\Phi(-\alpha)}{\sqrt{V[\nu]}} \quad (5)$$

It is well known that the following equations hold for stochastic variables.

$$\begin{aligned} E[i+j] &= E[i] + E[j], \qquad (6) \\ V[i+j] &= V[i] + V[j] + 2C[i,j], \qquad (7) \\ C[i,j] &= \sqrt{V[i]V[j]}R[i,j], \qquad (8) \\ C[i+j,k] &= C[i,k] + C[j,k], \qquad (9) \end{aligned}$$

where $C[x, y]$ denote the covariance between x and y.

4.3 ADD operation

In Step 4, for each arc $e_i = (u_i, v)$ $(i = 1, 2, \cdots, f(v))$, we compute the longest path length $l_i^t(v)$ from o_{init} to v passing through e_i. Using $l_i^t(v) = l(u_i) + w(e_i)$ and equations (6), (7) and (8), we calculate the mean $E[l_i^t(v)]$ and the variance $V[l_i^t(v)]$ of $l_i^t(v)$ as follows.

$$\begin{aligned} E\left[l_i^t(v)\right] &= E[l(u_i)] + \mu(e_i), \\ V\left[l_i^t(v)\right] &= V[l(u_i)] + \sigma^2(e_i) \\ &\quad + 2\sqrt{V[l(u_i)]\sigma^2(e_i)}R[l(u_i), w(e_i)]. \end{aligned}$$

In Step 5, for a node $x \in V_S$ whose $l(x)$ has been already computed, we calculate the correlation coefficient $R[l_i^t(v), l(x)]$ between $l_i^t(v)$ and $l(x)$ from equations (8) and (9) as follows.

$$\begin{aligned} &R\left[l_i^t(v), l(x)\right] \\ &\quad = \frac{\sqrt{V[l(u_i)]}R[l(x), l(u_i)] + \sigma(e_i)R[l(x), w(e_i)]}{\sqrt{V\left[l_i^t(v)\right]}}. \end{aligned}$$

Note that we set $R[l(x), l(u_i)] = 1$, if $x = u_i$.

Similarly, for each arc $y \in A_S$ we calculate the correlation coefficient $R[l_i^t(v), w(y)]$ between $l_i^t(v)$ and $w(y)$ as follows.

$$R\left[l_i^t(v), w(y)\right] = \frac{\sqrt{V[l(u_i)]}R[l(u_i), w(y)] + \sigma(e_i)\rho(e_i, y)}{\sqrt{V\left[l_i^t(v)\right]}}.$$

where $\rho(e_i, y)$ is the correlation coefficient between $w(e_i)$ and $w(y)$ of arcs e_i and y in A_S.

4.4 MAX operation for Multiple Stochastic Variables

In Step 6, instead of $l_i(v) = \max[l_1^t(v), l_2^t(v), \cdots, l_i^t(v)]$, we recursively calculate $l_i(v) = \max[l_{i-1}(v), l_i^t(v)]$ for $i \geq 2$ and $l_i(v) = l_i^t(v)$ for $i = 1$. From (1) through (9), the mean $E[l_i(v)]$ and the variance $V[l_i(v)]$ of $l_i(v)$ are calculated as follows. In addition, the correlation coefficients $R[l_i(v), l(x)]$ and $R[l_i(v), w(y)]$ between $l_i(v) = \max[l_{i-1}(v), l_i^t(v)]$ and $l(x)$ of each node x whose $l(x)$ has been already computed and $w(y)$ of each arc y, respectively, are calculated as follows.

$$\begin{aligned} E[l_i(v)] &= E[l_{i-1}(v)]\Phi(\alpha) + E[l_i^t(v)]\Phi(-\alpha) + a\varphi(\alpha), \\ V[l_i(v)] &= \left(E^2[l_{i-1}(v)] + V[l_{i-1}(v)]\right)\Phi(\alpha) \\ &\quad + \left(E^2[l_i^t(v)] + V[l_i^t(v)]\right)\Phi(-\alpha) \\ &\quad + \left(E[l_{i-1}(v)] + E[l_i^t(v)]\right)a\varphi(\alpha) - E^2[l_i(v)], \end{aligned}$$

$$\alpha = \frac{E[l_{i-1}(v)] - E[l_i^t(v)]}{a},$$

$$a = \sqrt{\begin{array}{l} V[l_{i-1}(v)] + V[l_i^t(v)] \\ -2\sqrt{V[l_{i-1}(v)]}\Big(\sqrt{V[l(u_i)]}R[l_{i-1}(v), l(u_i)] \\ +\sigma(e_i)R[l_{i-1}(v), w(e_i)]\Big) \end{array}},$$

$$R[l_i(v), l(x)] = \frac{\begin{array}{c}\sqrt{V[l_{i-1}(v)]}R[l_{i-1}(v), l(x)]\Phi(\alpha) \\ +\sqrt{V[l_i^t(v)]}R[l_i^t(v), l(x)]\Phi(-\alpha)\end{array}}{\sqrt{V[l_i(v)]}},$$

$$R[l_i(v), w(y)] = \frac{\begin{array}{c}\sqrt{V[l_{i-1}(v)]}R[l_{i-1}(v), w(y)]\Phi(\alpha) \\ +\sqrt{V[l_i^t(v)]}R[l_i^t(v), w(y)]\Phi(-\alpha)\end{array}}{\sqrt{V[l_i(v)]}}.$$

To obtain a, we compute the correlation coefficient $R[l_{i-1}(v), l_i^t(v)]$ between $l_{i-1}(v)$ and $l_i^t(v)$ as follows.

$$R\left[l_{i-1}(v), l_i^t(v)\right] = \frac{\begin{array}{c}\sqrt{V[l(u_i)]}R[l_{i-1}(v), l(u_i)] \\ +\sigma(e_i)R[l_{i-1}(v), w(e_i)]\end{array}}{\sqrt{V[l_i^t(v)]}}.$$

Thus correlation coefficients $R[l_i^t(v), l(x)]$ and $R[l_i^t(v), w(y)]$ in Step 5 and $R[l_i(v), l(x)]$ and $R[l_i(v), w(y)]$ in Step 6 are also computed.

The computation of $\max[l_{i-1}(v), l_i^t(v)]$ would be an approximated one for $i \geq 3$, since each $l_{i-1}(v)$ no longer has the normal distribution. The accuracy of our computation will be checked through experiments, which will be reported in section 5.

5 Experimental Results

5.1 Verifying the Accuracy

The proposed statistical analysis algorithm is implemented using C program language on a 1GHz Pentium III personal computer, and is applied to six datapaths and schedules (*Datapath A, Datapath B, Datapath C, Datapath D, Datapath E* and *Datapath F*), which are synthesized in [3]. We use the module library in [6], and model the delays of an adder and a multiplier with $N(7.5, 0.69)$ and $N(16, 2.78)$, respectively. If two operations o_i and o_j are assigned to the same functional unit, we set the correlation coefficient $\rho(e_i, e_j) = 1$ between $w(e_i)$ and $w(e_j)$ for two arcs $e_i = (o_i^s, o_i^e)$ and $e_j = (o_j^s, o_j^e)$ in a scheduling graph G_S. Otherwise the correlation coefficient is set to 0.0, 0.3, 0.6, or 0.9, which is shown in the column of "Corr".

Table 1 shows the results of the Monte Carlo simulation and our statistical analysis. In the columns indicated by E and $\sqrt{V}$, the means and the standard deviations of the distributions of total computation time are shown, respectively. The relative errors to the results of the Monte Carlo simulation are also shown in the same table. As we can see from this table, our method provides the mean and the standard deviation with the relative errors 0.10 % and -0.64 %, respectively, on average.

To compare the shape of probability density function, we compute a large number of samples of total computation time for *Datapath E* using the Monte Carlo simulation, and the results are shown in Fig. 4 as the histogram of total computation time. The solid curve in the same figure represents the normal distribution with the mean and standard deviation obtained from our statistical analysis. From the figure, we can see that our result is very close to the distribution obtained by the Monte Carlo simulation.

5.2 Synthesis Examples

Next, we consider the problem to find a datapath and a schedule with minimum mean total computation time

Proceedings of the 10th WSEAS International Conference on CIRCUITS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp318-323)

Table 1: Total computation time for datapaths and schedules in [3].

		Monte Carlo			Ours			Error [%]	
	Corr	E [ns]	$\sqrt{V}$	time [s]	E [ns]	$\sqrt{V}$	time [s]	E	$\sqrt{V}$
Datapath A	0	64.28	4.01	1.41	64.61	4.03	$\leq$ 0.01	+0.51	+0.50
	0.3	63.82	4.60	1.44	63.96	4.56	$\leq$ 0.01	+0.22	-0.87
	0.6	63.38	5.57	1.43	63.48	5.56	$\leq$ 0.01	+0.16	-0.18
	0.9	62.70	6.40	1.41	62.74	6.40	$\leq$ 0.01	+0.06	0.00
Datapath B	0	71.86	4.22	1.21	72.09	4.08	$\leq$ 0.01	+0.32	-3.32
	0.3	71.00	6.38	1.23	71.20	6.32	$\leq$ 0.01	+0.28	-0.94
	0.6	70.51	5.80	1.18	70.64	5.77	$\leq$ 0.01	+0.18	-0.52
	0.9	70.17	6.78	1.18	70.27	6.76	$\leq$ 0.01	+0.14	-0.29
Datapath C	0	77.93	4.81	1.35	78.00	4.88	$\leq$ 0.01	+0.09	+1.46
	0.3	77.69	5.88	1.37	77.69	5.88	$\leq$ 0.01	0.00	0.00
	0.6	77.47	7.02	1.37	77.46	7.02	$\leq$ 0.01	-0.01	0.00
	0.9	77.18	8.00	1.35	77.17	8.01	$\leq$ 0.01	-0.01	+0.13
Datapath D	0	87.54	6.27	1.30	87.59	6.14	$\leq$ 0.01	+0.06	-2.07
	0.3	87.16	7.35	1.27	87.17	7.26	$\leq$ 0.01	+0.01	-1.22
	0.6	86.71	7.95	1.31	86.73	7.96	$\leq$ 0.01	+0.02	+0.12
	0.9	86.55	8.97	1.32	86.55	8.95	$\leq$ 0.01	0.00	-0.22
Datapath E	0	132.54	8.30	3.01	132.50	8.36	0.02	-0.03	+0.72
	0.3	132.27	10.46	2.98	132.25	10.51	0.02	-0.02	+0.48
	0.6	131.96	11.74	3.08	131.96	11.76	0.02	0.00	+0.17
	0.9	131.68	13.59	3.03	131.67	13.59	0.02	-0.01	0.00
Datapath F	0	136.54	7.95	3.03	136.79	7.41	0.02	+0.18	-6.79
	0.3	135.33	10.09	3.00	135.57	9.92	0.02	+0.18	-1.68
	0.6	134.60	12.01	2.99	134.78	11.92	0.02	+0.13	-0.75
	0.9	132.71	13.61	3.01	132.75	13.61	0.02	+0.03	0.00
Average								+0.10	-0.64

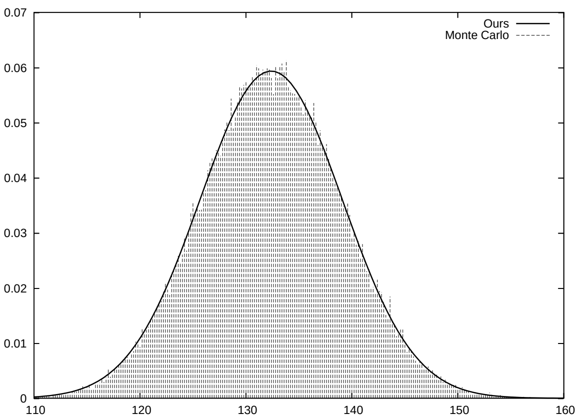

Figure 4: The distribution of total computation time for *Datapath E*, where the means of the Monte Carlo simulation and our statistical analysis are 132.54 and 132.50, respectively.

(i.e., minimum $E[l(o_{quit})]$) with maximum total computation time $T_{max} \leq T_M$ under given set of available modules and a constant T_M (the upper bound of maximum total computation time T_{max}). Note that T_{max} is computed using maximum execution time of each operation. To find such datapath and schedule, we incorporate the proposed statistical analysis into binding exploration based synthesis system [3], and set $E[l(o_{quit})]$ for the objective function to be minimized. We use three datapath synthesis benchmarks; four-order all-pole lattice filter (ALF), four-order Jaumann wave digital filter (JWF), and fifth-order elliptic wave filter (EWF) as target algorithms.

For comparison, datapaths and schedules are also synthesized by asynchronous datapath synthesis systems in [1, 2, 3], in which the conventional objective function is used. That is, the systems in [1, 2, 3] find datapaths and schedules with minimum typical total computation time T_{typi} with $T_{max} \leq T_M$. All synthesis results are evaluated by the Monte Carlo simulation, and we obtain means of the total computation time.

Table 2 shows the results obtained with several different specifications of the numbers of functional units (Add and Mul) and registers (Reg) and the upper bound of maximum total computation time T_M for each benchmark. The last column "time" shows runtime in second. Note that our results cannot be compared with the results of [1, 2] for some instance, since [1] and [2] do not produce any feasible solutions.

As we can see from the column of E, our system using the proposed statistical analysis always provides better solutions. For the case of small target algorithms or small number of functional units, the performance difference between datapaths designed by our system and conventional ones seems not so large. However, our system tends to generate better solutions than conventional ones in the mean total computation time, when the size of a target algorithm becomes larger, and the number of functional units becomes

Proceedings of the 10th WSEAS International Conference on CIRCUITS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp318-323)

Table 2: Experimental results of datapath synthesis.

						[1]			[2]			[3]			Ours	
Bench.	Add	Mul	Reg	T_M	Corr	$T_{typi}[ns]$	E [ns]	time [s]	$T_{typi}[ns]$	E [ns]	time [s]	$T_{typi}[ns]$	E [ns]	time [s]	E [ns]	time [s]
ALF	4	2	6	82	0	62	63.90	≤1	62	63.90	≤1	62	65.25	≤1	63.63	1
					0.3		63.60			63.60			64.72		63.39	1
					0.6		63.21			63.21			64.07		63.03	1
					0.9		62.62			62.62			63.05		62.55	1
	2	2	6	92	0	69.5	72.41	≤1	69.5	72.41	≤1	69.5	72.88	≤1	71.68	1
					0.3		71.93			71.93			72.32		71.26	1
					0.6		71.32			71.32			71.62		70.98	1
					0.9		70.40			70.40			70.55		70.21	1
JWF	2	2	7	102	0	77	79.69	≤1	77	79.69	≤1	77	79.52	≤1	78.22	1
					0.3		79.15			79.15			79.01		77.41	1
					0.6		78.50			78.50			78.39		77.32	1
					0.9		77.59			77.59			77.55		77.13	1
	2	1	7	114	0	86.5	88.57	≤1	86.5	88.57	≤1	86.5	88.69	≤1	87.44	1
					0.3		88.07			88.07			88.10		86.76	1
					0.6		87.47			87.47			87.43		86.69	1
					0.9		86.73			86.73			86.62		86.54	1
EWF	3	3	12	174	0	131.5	134.68	22	131.5	134.68	11	131.5	137.24	19	132.32	31
					0.3		134.01			134.01			136.22		132.18	45
					0.6		133.20			133.20			134.97		131.89	29
					0.9		133.20			133.20			133.07		131.81	30
	2	2	12	184	0	-	-	1941	-	-	386	131.5	137.20	26	136.08	42
					0.3		-			-			136.10		135.19	44
					0.6		-			-			134.75		134.31	61
					0.9		-			-			132.83		132.33	68

larger.

On the other hand, our system runs slower than [3], since the time complexity of the proposed statistical analysis algorithm is $\Theta(|A_S|^2)$ while the complexity of constant delay analysis is $\mathcal{O}(|A_S|)$. In the future, we will develop an asynchronous datapath synthesis system, which can generate a solution quickly even if the statistical delay analysis is incorporated.

6 Conclusion

In this paper, we have proposed a statistical schedule length analysis method for evaluating a schedule and a datapath during asynchronous datapath synthesis. Statistical analysis based synthesis system generates schedules and datapaths having higher statistical performances, which are not synthesized by using conventional systems.

The normal distribution is not always adequate for the delay analysis of asynchronous datapaths. Development of a proper model and algorithms to compute statistical parameters, which reflect practical random delay variations, are left for future work.

Acknowledgements: This work is part of a program for the “21st Century COE Program” for promoting Science and Technology by Ministry of Education, Culture, Sports, Science and Technology, Japan. Also it is partly supported by the Society for the Promotion of Science, Japan, under Grant-in-Aid for Scientific Research (C), No.16560294, 2004-2005.

References:

[1] B. M. Bachman, H. Zheng, C. Merys, “Architectural Synthesis of Timed Asynchronous Systems,” Proc. International Conf. Computer Design, pp.354-363, 1999

[2] M. Kawanabe, H. Saito, M. Imai, H. Nakamura, T. Nanya, “Design Space Reduction Filter in Asynchronous Data-path Synthesis,” Technical Report of IEICE, SLDM2003-112, Vol.2003, No.120, pp.103-114, 2003.

[3] K. Ohashi, M. Kaneko, “Asynchronous Datapath Synthesis Based on Binding Space Exploration,” Proc. 17th Workshop on Circuits and Systems in Karuizawa, pp.549–554, 2004.

[4] C. Clark, “The Greatest of a Finite Set of Random Variables,” Operations Research, vol.9, pp.145–152, 1961.

[5] S. Tsukiyama, “Statistical Timing Analysis : A Survey,” Proc. 18th Workshop on Circuits and Systems in Karuizawa, pp.533-538, 2005.

[6] H. Tomiyama, H. Yasuura, “Module Selection Using Manufacturing Information,” IEICE Trans. Fundamentals, vol.E81-A, no.12, pp.2576-2584, Dec. 1998.

[7] K. Ohashi, M. Kaneko, “Statistical Schedule Length Analysis in Asynchronous Datapath Synthesis,” Proc. IEEE International Symposium on Circuits and Systems, pp.700-703, 2005.