Clock Event Suppression Algorithm of VELVET And Its Application to S-820 Development

Yoshio Takamine, Shunsuke Miyamoto, Shigeo Nagashima
Central Research Laboratory, Hitachi, Ltd., Kokubunji, Tokyo 185, Japan

Masayuki Miyoshi, Shun Kawabe
Kanagawa Works, Hitachi, Ltd., Hadano, Kanagawa 259-13, Japan

ABSTRACT

An advanced clock event suppression algorithm for high-speed logic simulation is described.

A new signal value "Cn" and a "current clock (CC)", which indicates the current status of clock signals, has been introduced to realize this algorithm.

This algorithm suppresses about 60% of the total events, and eliminates 40% of CPU time. No overhead is needed to incorporate this algorithm using hardware support of VELVET (vectorized processing system for logic verification).

Hitachi's latest supercomputer S-820 has been developed using VELVET. The development period has been shortened to 3/4 that of the S-810.

1. Introduction

The recent rapid increase in the complexity of computers and VLSI has increased the computing time necessary for logic simulation. Advances in packaging technology have also made it increasingly important to verify logic design and to shorten the development period. A high-speed logic simulator is indispensable for carrying out tests for logic verification.

Special-purpose hardware for logic simulation has been developed and implemented for this purpose [1]-[6]. Most of it is based on parallel processing architecture. Generally speaking, parallel processor performance is very high in terms of simulation execution. However, it is difficult to make all processor elements work at their maximum level at all times. In addition, the relatively low throughput at the interface with the host processor makes it difficult to minimize total simulation time, which is most important from the viewpoint of users.

Another approach to high-speed logic simulation is to use a high-performance, general-purpose vector processor [7]-[8]. This reduces total simulation time, but it is difficult to vectorize all simulation processes using the existing vector instructions. Some hardware improvement is needed to achieve maximum performance of the processor.

Since the present objective is to achieve high performance in total simulation time, a general-purpose vector processor was applied to logic simulation. The vectorized processing system for logic verification (VELVET) including six new vector instructions for logic simulation has been implemented in the S-810 supercomputer [9].

The new instructions are very effective. However, an improved software algorithm is necessary to achieve the desired performance, which is at least one hundred times faster than conventional logic simulators. Reducing the number of events is effective in reducing CPU time. It is noteworthy that the authors' experience shows that clock signals cause about 70% of events which are evaluated in an event-driven simulator. Most of these events do not affect the behavior of the simulated logic system. The clock event suppression algorithm, which greatly reduces the number of events, has been successfully incorporated for the first time in VELVET. Thus high-speed logic simulation has been achieved through both hardware and software improvements.

This paper describes the clock event suppression algorithm and its implementation in VELVET. It also describes the development of Hitachi's latest supercomputer S-820.

2. VELVET System Overview

The VELVET system includes the six new vector instructions added to the S-810 supercomputer. The VSF (Vector Simulate Function) instruction is a most remarkable instruction. It evaluates the new output signal value for each 4-input / 1-output gate or flip-flop, and generates new events in one instruction execution cycle. This instruction treats clock signal values which have been introduced for clock event suppression. The other five instructions are used for general-purpose vector operations. They have been added to build up the power of logical operations and of pointer data accesses from main storage. Almost all processes of the simulation execution have been vectorized using the new vector instructions.

VELVET executes gate-level, zero-delay logic simulation. It also simulate logic systems described in functional-level or register-transfer-level (RTL) language. The simulation algorithm is event-driven and is based on clock event suppression. It treats four types of static signal values, 0, 1, X (unknown), and Z (nigh-impedance).

3. Clock Event Suppression Algorithm

3.1 Background of Clock Event Suppression

Today's computers and VLSI are usually designed with synchronized logic. Therefore, VELVET adopts a zero-delay simulation technique for combinational circuits. It assumes that all events are synchronized with clock signals.

Clock signals cause a great number of events at gates in clock distribution circuits or in control circuits, which control supply of the clock signals to flip-flops. However, the gates in the clock distribution circuits are only for amplifying the clock signals electrically and do not affect logical behavior. Also the events at gates in the control circuits have no logical effect as long as control signals do not change.

The authors' experience show that clock signals cause about 70% of events. They are expected to occur in cycles, and most of them do not affect the behavior of the simulated logic system. Therefore clock event suppression should be quite effective in high-speed logic simulation. The basic idea of clock event suppression is stated in [10].

3.2 Expression of Clock Signal

To realize the clock event suppression, an efficient method of distinguishing necessary events affecting the behavior of a simulated logic system from unnecessary events caused by clock signal changes is needed. The new signal value "Cn", which represents the signal whose electrical level varies in cycles, has been introduced.

The clock signals usually used in a simulated logic system are those of a multiphase clock, as shown in Figure 1. In conventional logic simulators, these clock signals are represented as alternate repetitions of "0" and "1". Both the rise and fall of clock signals cause events.

For clock event suppression, these clock signals are represented as "C" values. Their phase number is represented as "n", where "n" is "0", "1", ..., or "N". Inverted clock signals are represented as "-Cn". The "C" value indicates one pulse including rise and fall in each clock cycle. It permits the clock signal to be treated as a static signal, that is, clock signals do not cause events as long as they do not change their polarity or phase number, nor change to static signals.

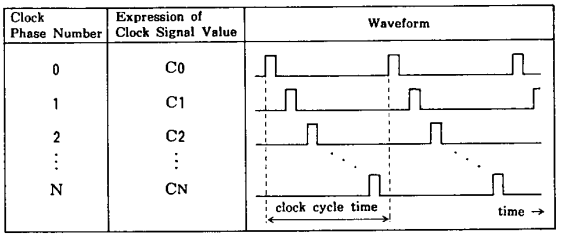


Figure 1. Expressions of Clock Signal Value

3.3 Logical Operations with Clock Signal Values

The "Current clock (CC)" has been introduced, and logical operations such as NOT, AND, and OR with clock signal values have been defined to realize the clock event suppression algorithm.

CC is a variable which indicates the status of clock signals, that is, which clock signals are actually high-level or low-level. Assuming that the clock signals of the simulated logic system are as shown in Figure 1, the value of CC changes in each clock cycle to "C0", "-C0", "C1", "-C1", ..., "CN", and "-CN" in turn. When the value of CC is "Cn", the clock signal represented as clock signal value "Cn" is actually high. At this point the clock signals represented as "-Cn", "Cp", and "-Cp" (p is not equal to n) are low, low, and high respectively. When the value of CC is "-Cn", the clock signals represented as "Cn", "-Cn", "Cp", and "-Cp" are low, high, low, and high respectively.

Logical operations with these new clock signal values have been defined as shown in Figure 2, where "n" and "p" indicate different phase numbers.

INPUT	OUTPUT
0	1
1	0
X	X
Cn	-Cn
-Cn	Cn
Cp	-Cp
-Cp	Cp

(a) NOT

I1 \ I2	0	1	X	Cn	-Cn	Cp	-Cp
0	0	0	0	0	0	0	0
1	0	1	X	Cn	-Cn	Cp	-Cp
X	0	X	X	X/0	0/X	0	X
Cn	0	Cn	X/0	Cn	0	0	1/0
-Cn	0	-Cn	0/X	0	-Cn	0	0/1
Cp	0	Cp	0	0	0	Cp	0
-Cp	0	-Cp	X	1/0	0/1	0	-Cp

(b) AND

: should be represented as alternate repetitions of static signal values.
"n" is equal to the phase number of "current clock", and "p" is not equal to "n".

Figure 2. Logical Operations with Clock Signal Value

The NOT operation with a clock signal value changes the polarity only of the input clock signal value.

The AND operation with static signal value "0" and a clock signal value, produces an output signal value "0". The AND operation with "1" and a clock signal value, produces an output clock signal value with the same phase number as that of the input clock signal value. The AND operation with "X" and a clock signal value, produces more complicated output signal values, which should be represented as alternate repetitions of "0" and "X". Also AND operation with two clock signal values of different phase numbers, produces complicated output signal values, which should be represented alternately as "0" and "1", but with more than two changes in each cycle. Therefore, these complicated values are represented by static signal values, and must be re-evaluated whenever any clock signals rise or fall.

3.4 Principles of Clock Event Suppression

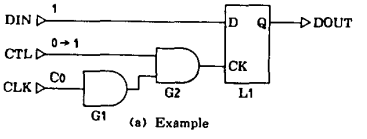
The principles of clock event suppression are illustrated in Figure 3.

In an example logic circuit, initial input signal values at DIN and CTL are "1" and "0" respectively. A clock signal represented as "C0" is input to AND-gate G2 through G1. The initial output signal value of DOUT is assumed to be "0". Before the signal value of CTL becomes "1", the output signal value of G2 is "0". When the signal value of CTL becomes "1", the clock signal is supplied to flip-flop L1 through G2, and the output signal value of L1 changes to "1".

With the conventional event-driven algorithm, the clock signal is represented as alternate repetitions of "0" and "1". Both the rise and fall of the clock signal causes events at G1 and G2. Thus these gate must be evaluated at least twice per clock cycle. Most of these events do not affect the behavior of the circuit. The effective events are caused, only immediately after the input signal value of CTL becomes "1".

With the clock event suppression algorithm, the clock signal value "C0" implies periodical changes,

and can be treated as a static signal. The initial output signal value of G1 is also "C0". The output signal value of G2 is "0" initially, and becomes "C0" when the input signal value of CTL becomes "1". Several events occur only when the signal value of CTL changes.



(a) Example

(b) Behavior of Example

Algorithm	Signal Name	Behavior (○: event)	Frequency of Gate Evaluation
Conventional Event-Driven	CLK, CTL, G1, G2, DOUT		11
Clock Event Suppression	CLK, CTL, G1, G2, DOUT	C0 ↔ C0	2

Figure 3. Principles of Clock Event Suppression

3.5 Simulation Flow with Clock Event Suppression

Simulation flow with clock event suppression is illustrated in Figure 4.

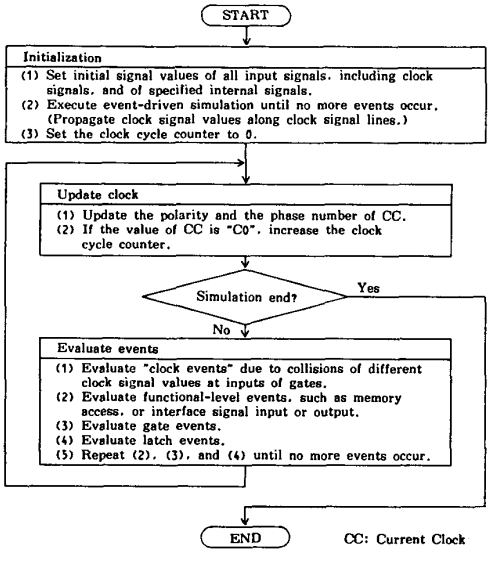


Figure 4. Simulation Flow with Clock Event Suppression

In the initialization process, all input signals and some internal signals specified by simulation test data, are set to initial values. All clock signal values are propagated along clock signal lines. Thus clock signal changes cause no events during simulation execution.

In the simulation process, the value of "current clock (CC)" is updated at the beginning of each simulation cycle. If CC is "C0", then it is the beginning of the clock cycle, and the clock cycle counter indicates an increase of one.

In each simulation cycle, a small number of "clock events" initiated by updating CC should be evaluated for gates with more than one clock signal input or with "X" value and clock signal input. The rest of the simulation flow is the same as in conventional event-driven simulation.

4. Experimental Results

4.1 Effect of Clock Event Suppression

The VELVET system has been developed based on the S-810 supercomputer. This system incorporates the clock event suppression algorithm for the first time. The complicated logical operations shown in Figure 2 are executed in one machine cycle using the VSF instruction. Therefore, no overhead is needed to incorporate the clock event suppression algorithm into VELVET.

Experimental simulation was performed for actual logic systems. A large-scale general-purpose computer, which contains over one million gates, was used as benchmark.

The effect of clock event suppression on event number reduction is shown in Figure 5.

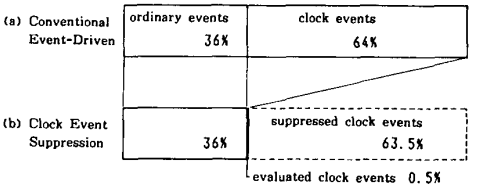


Figure 5. Effect of Clock Event Suppression (Number of Events)

The proportion of clock events caused by clock signal changes is 64% in the benchmark simulation using the conventional event-driven algorithm. Using the clock event suppression algorithm, over 99% of these clock events are suppressed. This is equivalent to 63.5% of the total events. The proportion of evaluated clock events due to collisions of different clock signal values, or of an "X" value and a clock signal value at inputs, is only 0.5% of the total.

The reduction of CPU time by clock event suppression is shown in Figure 6. Clock event suppression reduces CPU time by about 40%.

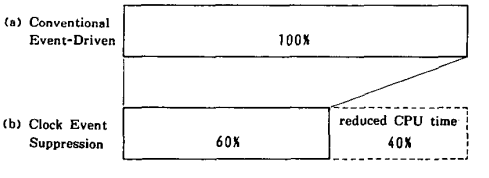


Figure 6. Effect of Clock Event Suppression (CPU Time)

4.2 S-820 Supercomputer Development with VELVET

The VELVET system was used to develop Hitachi's latest supercomputer S-820.

As mentioned before, logic simulation aims at reducing the number of errors after manufacture and at reducing the development period. However, when the S-810 supercomputer was developed, it was difficult to verify logic design before manufacture due to lack of performance of conventional software simulator.

The VELVET system can simulate a large scale system containing several million gates over one hundred times faster than a conventional software simulator. Therefore, it has been more successful on the S-820 than on the S-810. Details showing the contrast between the S-820 and the S-810 are shown in Table 1.

Table 1. Summary of the S-820 Development

	S-810	S-820
Simulator	a gate-level software simulator	VELVET
Simulation Methods	unit simulation	unit simulation and system simulation
Relative Number of Gates for Each Simulation Job	100	500
Relative CPU Time for Each Simulation Job	100	1
Relative Length of Development Schedules	100	75
Relative Number of Logic Design Errors Detected after Manufacture	100	60

During the S-810 development, only unit simulation was available with the conventional software simulator. Each unit contains several hundred thousand gates. By the time the S-820 was being developed, system simulation containing several million gates had become available by using VELVET. Therefore, the proportion of errors detected before manufacture was very high, and the S-820 development period was reduced to 3/4 that of the S-810.

5. Conclusions

This paper has described a new clock event suppression algorithm for high-speed logic simulation, and its implementation in the vectorized processing system for logic verification (VELVET), which has been implemented in the S-810 supercomputer.

In conventional event-driven logic simulators, the clock signal values of a simulated logic system are represented as alternate repetitions of "0" and "1". Both the rise and fall of clock signals cause events. The proportion of these events is estimated to be about 70% of all events processed in an event-driven simulator.

For clock event suppression, these clock signal values are represented as static signal value "Cn". They do not cause events as long as they do not change their polarity or phase number, nor change to static signal values.

"Current clock (CC)" has been introduced to define logical operations with clock signal value "Cn". CC indicates the current status of clock signals. Most logical operations with clock signal values produce an output signal value which is a simple static signal value or a clock signal value with the same phase number as the input clock signal. There are only a few complicated cases, which must be re-evaluated whenever clock signals rise or fall. However such cases are few, and have little effect on simulation execution.

The clock event suppression algorithm has been implemented in VELVET. This algorithm suppresses over 99% of the events caused by clock signal changes. This is equivalent to 60% of the total events which are evaluated in conventional event-driven simulators. Clock event suppression eliminates about 40% of total CPU time. No overhead for incorporating the clock event suppression algorithm is needed using VELVET hardware support. Thus the high performance of VELVET, which is over one hundred times faster than conventional logic simulators, has been achieved through both hardware and software improvements.

The VELVET system incorporating the clock event suppression algorithm was used to develop Hitachi's latest supercomputer S-820. The development period has been shortened to 3/4 that of the S-810.

6. Acknowledgements

The authors would like to thank H. Horikoshi, T. Kozawa, K. Furumaya, T. Odaka and Y. Tsuchiya for giving us the opportunity of developing VELVET and for providing us the necessary support.

7. References

[1] G.F.Pfister et al., "The Yorktown Simulation Engine: Introduction", Proc. 19th DA Conf., 1982, pp.51-54.

[2] J.K.Howard et al., "Introduction to the IBM Los Gatos Logic Simulation Machine", Proc. ICCD '83, 1983, pp.580-583.

[3] T.Sasaki et al., "HAL: A Block Level Hardware Logic Simulator", Proc. 20th DA Conf., 1983, pp.150-156.

[4] S.Takasaki et al., "HAL II: A Mixed Level Hardware Logic Simulation System", Proc. 23rd DA Conf., 1986, pp.581-587.

[5] N.Amano and S.Miyamoto, "EP-1 (Event Processor-1): A Logic Simulation Machine for VLSI", Proc. ICCAD '84, 1984, pp.251-253.

[6] F.Hirose et al., "Simulation Processor SP", Proc. ICCAD '87, 1987, pp.484-487.

[7] H.E.Krohn, "Vector Coding Techniques for High Speed Digital Simulation", Proc.18th DA Conf., 1981, pp.525-528.

[8] N.Ishiura et al., "High-Speed Logic Simulation on A Vector Processor", Proc. ICCAD '85, 1985, pp.119-121.

[9] S.Nagashima et al.,"Hardware Implementation of VELVET on the Hitachi S-810 Supercomputer", Proc. ICCAD '86, 1986, pp.390-393.

[10] E.G.Ulrich et al., "Design Verification for Very Large Digital Networks Based on Concurrent Simulation and Clock Suppression", Proc. ICCD '83, 1983, pp.277-280.