PROCEEDINGS OF THE 1992 SUMMER COMPUTER SIMULATION CONFERENCE

JULY 27-30, 1992
JOHN ASCUAGA'S NUGGET
RENO, NEVADA

TWENTY-FOURTH ANNUAL SUMMER COMPUTER SIMULATION CONFERENCE

Edited by
Paul Luker

Conference Sponsor
The Society for Computer Simulation

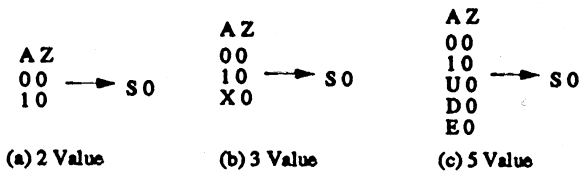


Figure 2: Pattern Transformations of Example

Let *SSP* be the set of simulation patterns. For complete verification, SSP should be the set of all possible patterns. The *set of reduced patterns(SRP)* is defined as the set of simulation patterns which uses the non-dominating logic value(S), and which represents the original or exhaustive set of simulation patterns.

For example, consider two value logic, and, complete verification of a 3 input AND gate. 8 patterns are required, as shown in Figure 3 (a). However, when the

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(a) SSP

A	B	C	Z
0	S	S	0
S	0	S	0
S	S	0	0
1	1	1	1

(b) SRP

Figure 3: SSP and SRP of 2 Value 3 Input AND

non-dominating set value is considered, only 4 patterns are necessary. The reduced set of patterns is shown in Figure 3 (b). The pattern 0SS includes; 000, 001, 010, and 011 input patterns, the output is 0. Instead of using 8 patterns, we can use the new 4 patterns for simulation while keeping the same level of accuracy. It is, clearly, understood that these 4 patterns contain the original 8 patterns; but for the simulation environment, they are simulated as 4 patterns, with the obvious savings of simulation time.

S simulation is a simulation technique which uses SRP rather than SSP. When we represent the function of a design, for each output, we can draw Karnaugh maps. Let c and n be the number of primary inputs and outputs, respectively. Then, there are c Karnaugh maps. In each Karnaugh map, there are l^n blocks where l is the number of logic values. If all Karnaugh maps are correct, we can say that the design is completely verified. In conventional simulation, with SSP, each block represents a simulation pattern. For each pattern, all outputs are considered simultaneously. However, in S Simulation, for each primary output, the blocks in the Karnaugh maps are merged. Then, using these minized patterns(SRP), simulation is executed for each output.

S PATTERN GENERATION ALGORITHM

The set of reduced patterns can be generated from the user's specification; in the form of truth tables, boolean equations, or some set of simulation patterns. If the specification is partial, the set of reduced patterns could also be partial. However, partial descriptions can be expanded to complete pattern sets and then reduced, if computing resources are available.

Firstly, consider a complete specification. Generating the new pattern for 2 logic values, can be easily explained through the use of Karnaugh maps, as one approach. For example, consider a 2 input AND gate. Figure 4 shows the Karnaugh map for this case. In this representation,

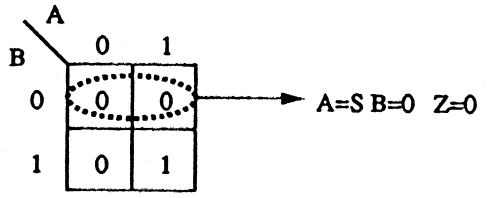


Figure 4: Karnaugh Map of 2 Value 2 Input AND

each box represents a pattern. The represented function is $Z = AB$ or $\bar{Z} = \bar{A} + \bar{B}$. For exhaustive simulation, 4 patterns are necessary for testing. However, as shown in Figure 4, the two patterns(A=0 B=0, Z=0 and A=1 B=0, Z=0) can be represented as A=S B=0, Z=0, hence, one pattern is reduced. In the above two equations, each product term represents a pattern. Since each pattern is a vector of all primary inputs, if any input is not specified, then it is regarded as an S. Therefore, the set of simulation patterns can be represented by the union of the patterns with $Z = 0$ and the patterns with $Z = 1$, for two value logic.

The three and five value logic cases, for a 2 input AND gate, are shown in Figure 5.

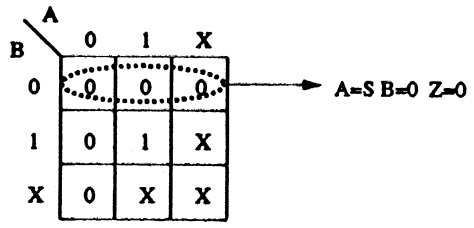
If the specification is given in the form of boolean equations, the reduced number of patterns can also be derived. Consider that the inputs are given in boolean equations, such as;

$$Z = ABC$$

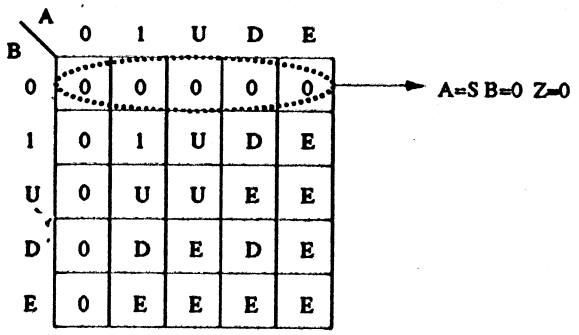
where Z is the output of a 3 input AND, and A, B, C are the inputs. Then the negation of this equation is

$$\begin{aligned} \bar{Z} &= \overline{ABC} \\ &= \bar{A} + \bar{B} + \bar{C} \end{aligned}$$

From these two equations, 4 patterns can be derived, as shown in Figure 3 (b).



(a) 3 Value Logic



(b) 5 Value Logic

Figure 5: Karnaugh Maps of Multi-Value 2 Input AND

Now, consider the process of reduced pattern generation. This is a merging procedure of several patterns, according to the logic values which are used. There are two kinds of mergings; matching and cover. Let l be the number of logic values. For example, if three value logic is used, three patterns are simultaneously compared. The l patterns are *matched* over input B if;

(1) the expected outputs are the same,
(2) all l patterns have the same logic values except for input B, and
(3) all l patterns have different logic values on B.

Then, l patterns are said to be *matching* and are merged into a new pattern, which has the same logic values, except that the B value is now an S. In Figure 2 (b), the 3 patterns satisfy all three conditions. Therefore they are merged into a new pattern.

Let's consider two patterns, P_1 and P_2. A pattern P_1 is said to be a *cover* of pattern P_2 if;

(1) P_1 and P_2 have the same expected outputs,
(2) P_1 and P_2 have the same logic values except on input B, and
(3) P_1 has an S value on B and P_2 has a non S value on B.

Then, since the pattern P_2 is a redundant pattern, it is deleted from the pattern set.

The algorithm to generate the SRP is shown in Figure 6. In the algorithm, patterns are generated for each

```
for all primary outputs
  do
    choose l patterns which have the same
    output values
    if these are the matching
      delete these patterns and
      add new pattern which has an S value
      on a matched pin
    if there exists a cover
      delete the covered patterns
  until no more new pattern
end
```

Figure 6: The Algorithm for SRP Generation

primary output, individually. Therefore, SRP is the union of disjoint sets of patterns for each primary output. Furthermore, for each output, a matching or a cover is searched. This is continued, recursively, until no more new S pattern is generated. If we consider the 2 value case, it is easy to generate the set of SRP patterns. However, for 3, 5, or multiple valued logic, it is more somewhat difficult.

Let's consider the partial descriptions. There are two kinds of partial descriptions. One is that the descriptions which are not specified by the user, are not considered any longer. These patterns are deleted from SSP. The other is that the unspecified descriptions are regarded as don't care patterns. So, the outputs of these patterns are assigned to unknown values and these are included in SSP. Let's consider the first case of partial description with the 3 value case. For example, consider the specification in Figure 7. If the given function is represented as a set of

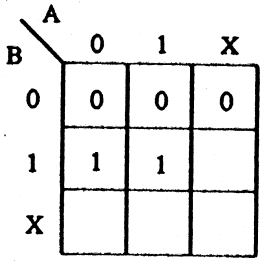


Figure 7: 3 Value K-Map of Partial Description

patterns, these are shown in Figure 8 (a). From these patterns, an SRP is derived according to the algorithm given in Figure 6. Therefore, the given 5 patterns are reduced to 3 patterns.

Consider the 5 value example as a second case of par-

A	B	Z
0	0	0
0	1	0
0	X	0
1	0	1
1	1	1
1	X	
X	0	
X	1	
X	X	

(a) Given

A	B	Z
0	0	0
0	1	0
0	X	0
1	0	1
1	1	1

(b) SSP

A	B	Z
S	0	0
1	0	1
1	1	1

(c) SRP

Figure 8: 3 Value Example of Partial Description

tial descriptions. If the given input is a boolean description such as;

$$Z = AB$$

then it is a partial description, since the outputs are not specified for the other cases. If it is the two value case, we can see that for those cases, the output is 0. However, in the 5 value case, the output can be 0, U, D, or E. Although the user doesn't provide the complete specification, there is a case that the user may want a complete verification. Since the outputs are not specified, each output is regarded as an unknown value(E). These may be more easily understood in the Karnaugh map form shown in Figure 9. In this example, only one pattern is provided,

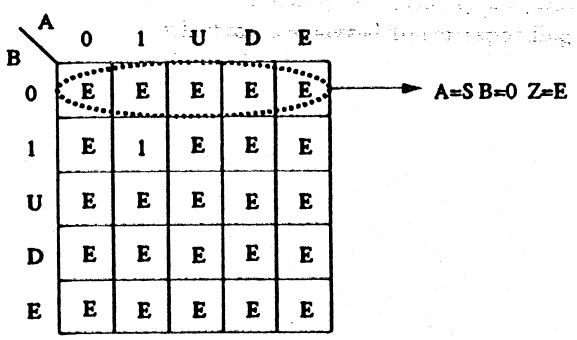


Figure 9: 5 Value K-Map of Partial Description

as shown in Figure 10 (a). To consider complete patterns, for all unspecified cases, the outputs are assigned to E. The SSP patterns are shown in Figure 10 (b). From these patterns, the SRP patterns are derived as in Figure 10 (c).

RESULTS AND REMARKS

Table 1 shows the number of SSP and SRP patterns for 2 value logic examples. The first 4 examples are the case of complete descriptions and the last 4 examples are the case of partial descriptions. These show the number

A	B	Z
1	1	1

(a) Given

A	B	Z
0	0	E
0	1	E
0	U	E
0	D	E
0	E	E
1	0	E
1	1	1
1	U	E
1	D	E
1	E	E
U	0	E
U	1	E
U	U	E
U	D	E
U	E	E
D	0	E
D	1	E
D	U	E
D	D	E
D	E	E
E	0	E
E	1	E
E	U	E
E	D	E
E	E	E

(b) SSP

A	B	Z
1	1	1
0	S	E
S	0	E
U	S	E
S	U	E
D	S	E
S	D	E
E	S	E
S	E	E

(c) SRP

Figure 10: 5 Value Example of Partial Description

of SRP patterns being much less than the SSP patterns. This, obviously, affects the total simulation time. For some designs, this is more significant than in others. For example, in the 8 to 1 multiplexer, 2^{11} patterns are required for complete exhaustive simulation. However, 54 patterns are sufficient, if an S value is considered. As was mentioned earlier, the number of patterns for exhaustive simulation of combinational logic, is l^n, where n is the number of primary inputs and l is the number of logic values. Consider the gates(AND, OR, NAND, NOR). The number of patterns using a non-dominating set value(S) is as follows.

$$\begin{aligned} n+1+(l-1)^n-1 &= n+(l-1)^n \\ &< l^n \end{aligned}$$

The $n+1$ term is for the case of the output being a 0 or 1. The $(l-1)^n-1$ term is for the case of the output being one of the the other logic values(X, U, D, or E). Particularly, for two logic values, the number of simulation patterns is $n+1$. For more complex primitives, it is more difficult to determine the exact number of patterns. However, the total number of patterns in SRP is

Circuit	SSP	SRP
6 input AND	64	7
9 input AND	512	10
Full adder	32	18
8-to-1 mux	2048	54
74ls181	1000	235
Bench c499	1000	792
Bench c880	1000	894
Bench c1355	1000	640

Table 1: Number of Patterns of SSP and SRP

roughly $O(kc(l-1)^n)$, where k is a positive constant and c is the number of primary outputs in the circuit. The above equation is usually less than l^n. However, some circuits which have more primary outputs, might have more patterns than SSP, since SRP is the union of patterns for each output. Therefore, the efficiency of SRP highly depends on the number of primary inputs and the number of outputs as well as circuit topology. The more primary inputs and/or the fewer primary outputs, the more speed up the SRP approach can achieve. Therefore, we can determine the order of SRP before generation is performed. The process of generation and simulation is undertaken, only, for those designs where substantial simulation efficiency can be achieved.

Simulation speed can be further reduced using heuristics. The circuits are partitioned into a set of subcircuits which have only one primary output and primary inputs which are connected to corresponding primary output as shown in Figure 11. Then, for each output, unrelated

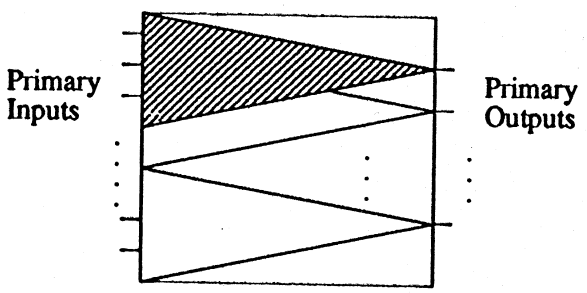


Figure 11: Partitioning of a Circuit

primary inputs are not considered. Therefore, efficient simulation for each output can be executed by removing unnecessary overhead.

Table 2 shows simulation times using SSP and SRP patterns. The SRP simulation time is much less than the SSP simulation time.

Circuit	Simulation Time of SSP	Simulation Time of SRP
6 input AND	0.08	0.02
9 input AND	0.62	0.02
full adder	0.07	0.03
8-to-1 mux	6.45	0.15
74ls181	4.98	1.00
BENCH c499	18.30	13.55
BENCH c880	25.88	21.45
BENCH c1355	32.51	20.78

Table 2: Simulation Time Using SSP and SRP

CONCLUSION

This paper has introduced a new technique that can vastly reduce simulation time for design verification. A non-dominating set value 'S' was introduced to reduce the number of simulation patterns, while keeping the same degree of testability as the specified simulation patterns. This approach is applicable for multi-valued simulations, as well as, for the two value case. Simulation is speeded up proportionately, where SRP patterns are used. Therefore, we can expand the domain where simulation is possible, as a design verification tool. Simulation results show that fast simulation can be achieved using this approach.

REFERENCES

M. A. Breuer and A. D. Friedman. 1976. in *Diagnosis and Reliable Design of Digital Systems.*

H. Chang and J. Abraham. 1987. "The Complexity of Accurate Logic Simulation." *Proc. ICCAD.*

S. Kang and S. A. Szygenda. January 1989. "Development of Reduced Time Interval Partitioned Simulation Algorithm." *IEE Computer Aided Design.*

S. Kang and S. A. Szygenda. March 1991. "Move: Model Verification System." *Proc. IEEE International Phoenix Conference on Computers and Communications.*

S. A. Szygenda. June 1972. "TEGAS2 - Anatomy of a General Purpose Test Generation and Simulation System." *Proc. 9th DAC .*

E. Ulrich. June 1983. "A design Verification Methodology based on Concurrent Simulation and Clock Suppression." *Proc. 20th DAC.*