Orthogonal Scan Paths for Data Path Logic

Robert B. Norwood* and Edward J. McCluskey

Center for Reliable Computing, Stanford University
Gates 236, MC 9020
Gates Building 2A
Stanford, CA 94305

Voice: (415) 723-1258
Fax: (415) 725-7398
Email: norwood@shasta.Stanford.EDU

* designated presenter

Extended Abstract

We have implemented a synthesis-for-test algorithm to implement orthogonal scan paths in data path logic. Orthogonal scan paths [Avra 92] facilitate the sharing of the functional and the test logic, thereby reducing the overhead of the scan features.

Traditional scan paths, shown in Figure 1, connect individual flip-flops within a register and then connect the registers, e.g., bit one of register one is connected to bit two of register one, and bit two is connected to bit three of register one, and so on until the last bit of register one is connected to bit one of register two. An orthogonal scan path, shown in Figure 2, is orthogonal to the traditional scan path. The flip-flops are connected in the scan path so that bit one of register one connects to bit one of register two, and bit two of register one connects to bit two of register two, and likewise for all the bits of the register. In this way, the scan path follows the normal data path flow, but is orthogonal to the traditional scan path flow. Orthogonal scan paths allow functional elements of the data path, such as adders and multipliers, to be used, with slight modifications, to implement the scan path.

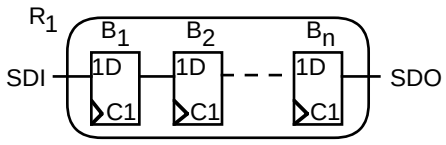

Figure 1. Traditional scan path

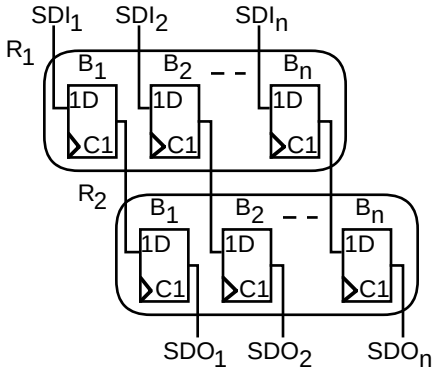

Figure 2. Orthogonal scan path

An ordering for the data path registers is determined to maximize the amount of sharing of the functional elements and minimize the amount of additional interconnect needed for the scan path. Taking the orthogonal scan path into account during such high-level synthesis operations as function binding and register allocation allow for a better final solution, but orthogonal scan paths can be applied to any data path. Once the order is determined, the functional elements are modified to allow them to be used during the scan operations. For example, an adder (A + B = C) can be used to pass data from register A to register C if the B input is forced to zero. Only a single gate per bit, along with the scan mode select, is needed to disable an input. For an adder an AND gate can be used to force a zero on an input during the scan operation. The orthogonal scan path order may be selected so that additional logic is not added to the critical path. If the

registers in the data path have enables, multiple orthogonal scan path configurations can be used to make better use of the available functional elements, though additional test mode selects may be required. The configurations are then used in series to scan a different sets of registers, while previously scanned registers hold their values.

Using the Stanford synthesis-for-test tool, TOPS, we have synthesized various benchmark circuits using this technique, and results show that orthogonal scan paths can result in no additional scan in/out pins, little, or no, additional interconnect, and only slight modifications to the functional units. This is in contrast to traditional scan paths that require additional test pins, extra interconnect, and the addition of MUXes to every flip-flop. Orthogonal scan paths also have the added benefit of reducing the length of the scan chain and thereby reducing the test vector application time.

Table 1 shows results for two benchmark circuits. Only the data path is considered; the control is assumed to be scanned in some other manner. Both orthogonal scan path circuits have much less test logic overhead than with traditional scan paths. Neither circuits requires additional pins for scanning data in or out, but test mode selects are still needed. Modifications to the control are not considered. The diffeq benchmark has one register that must be serially scanned, requiring a MUX for each bit; three functional units must also be modified The ellipf benchmark has all the registers scan with orthogonal scan; three functional units must be modified. The test application time is greatly reduced with orthogonal scan.

Table 1. Results for orthogonal scan paths

Circuit	# Registers	# Test Pins	Scan Overhead	Scan Shifts
diffeq traditional scan	7 (32-bits each)	2 scan in/out 1 scan mode	193 MUXes	192
diffeq orthogonal scan	7 (32-bits each)	2 scan in/out 1 scan mode	32 MUXes 96 gates	32
ellipf traditional scan	11 (16-bits each)	2 scan in/out 1 scan mode	176 MUXes	176
ellipf orthogonal scan	11 (16-bits each)	0 scan in/out 2 scan mode	48 gates	7

References

[Avra 92] Avra, L., "Orthogonal Built-In Self-Test," *COMPCON Spring 1992 Dig. of Papers*, San Francisco, CA, USA, pp. 452-457, February 24-28, 1992.