Chapter 5

Conclusions

The fault detection and coverage of our behavioral test strategy is already discussed in chapter 4. Here we would look at some typical scenarios where this strategy can be applied.

One of the tasks of this project was to assist Warner Robins Air Logistics Center in their board level testing activities, i.e., development of overall board level testing strategies, applicable to legacy systems. For legacy systems, the main considerations for developing the test strategy are:

1. The original functionality of the boards should remain unchanged.
2. Additional hardware used for testing should be kept to a minimum.
3. Fault coverage should be high.
4. Fault isolation should be to the component level.
5. Technical effort required by the operator should be low.

Functional and in-circuit testing methods require expensive ATE and bed-of-nails fixture respectively. Familiarity with the ATE programming language is required for functional ATE. A new bed of nails fixture is required for a different board in case of in-circuit testing. Microprocessor emulation is not preferred since the main component i.e. the microprocessor itself cannot be tested. Memory emulation has a similar draw back in that memory cannot be tested. We can test memory separately, but this is an off-line method. Bus cycle emulation is more suited to test add-on cards to a computer rather than a general purpose microprocessor

based board. Boundary scan technique cannot be applied to legacy systems since most chips on them do not support boundary scan architecture.

Our behavioral test strategy seems to be a good candidate for application on legacy systems. We can add the test code based on behavioral test strategy in the same ROM which contains the original program provided there is sufficient space available in it. The original code should remain intact; the self-test code should be just added to it. If that is not possible, the next best approach is to add extra memory to the legacy board for self-test purposes. Our technique has the following advantages:

1. No change in the original code is required.
2. The system can be tested in the field.
3. The system is tested at full operational speed with the same clock running as during normal operation.
4. Correct timing of the bus signals are guaranteed for a fault free microprocessor since these are generated by the microprocessor and not the external tester.
5. Minimal addition of hardware. No external test setup is required for different boards. No ATE or bed of nails fixtures are required. And no interface circuitry between the ATE/bed of nails fixture and the board under test is needed.
6. The test code is in the same language as the original code so no tester language needs to be learned.

For the legacy systems there can be two possible scenarios for incorporating this self-test scheme:

1. Sufficient free memory on the system memory chip so that the self-test code can be accommodated in it.
2. Free addressable space available so that an extra memory chip containing self-test code can be added to the system. In that case, a jump to the starting address of the chip containing test code should be made at start. After completion of the test, another jump should be made to the original code in the system memory.

Following is a possible sequence of steps for incorporating test code in a legacy system:

1. Write the test program, assemble it and find how much space it takes.
2. Read the contents of the system memory and find out how much free space is available. We can put the test program in the same chip if there is enough free memory.
3. Put the jump to the test code instruction in the first location read by the processor after power on.
4. Our test code should contain a jump to the application program. The self test run would run at start and after its successful completion, jump to the application program would take place.
5. If the memory is in-circuit programmable, we can directly download the code to it, otherwise, we have to take out the chip from the circuit and then program it.

If the system memory is on a different board than the processor, we can first confirm its checksum by microprocessor emulation and then check the rest of the board through it. In that case, start small approach would be executed in the following sequence:

1. EEPROM/EPROM
2. Display
3. Microprocessor instructions to be used subsequently
4. RAM
5. Microprocessor

At the depot level, our test strategy can be used to replace the faulty component due to its good fault resolution. And if used in some aircraft, it can tell whether the system is working or not prior to being airborne.

REFERENCES

[ArmJ93] J.R. Armstrong, F.G. Gray, *Structured Logic Design with VHDL*, Prentice Hall, 1993

[BatesonJ85] J. Bateson, *In Circuit Testing*, Van Nostrand Reinhold, NY, 1985

[BroyS84] S. Broyles, "Automating functional programming for microprocessor based boards", *Proceedings of International Test Conference*, Oct. 1984, pp 730-736

[CohenB95] B. Cohen, *VHDL Coding Styles and Methodologies*, Kluwer Publishers, 1995

[FeeW77] W. J. Fee, *LSI Testing*, IEEE Computer Society, 1977

[FiliS85] S.F. Filippone, "Automating test bed fault detecting and diagnosis" *Proceedings of International Test Conference*, Nov. 1985, pp 386-392

[GoorA91] A.J. Van De Goor, *Testing Semiconductor Memories, Theory and Practice*, John Wiley & Sons, Chichester, 1991

[HaislettD82] D.W. Haislett, *A methodology for self testing microprocessors*, Masters thesis, Virginia Tech, 1982

[IEEEs90] IEEE standard Test Access Port and Boundary Scan Architecture, *IEEE Std 1149.1 – 1990*, IEEE Computer Society, 1990

[KaisM92] M.R. Kaiser, "DMA emulation and its use in microprocessor system troubleshooting", *WESON Conference Record*, Nov. 1992, pp 718-723

[LalaP97] P.K. Lala, *Digital Circuit Testing and Testability*, Academic Press, San Diego, CA, 1997

[LinM98] M.W. Lin, *"A Functional Test Planning System for Validation of DSP Circuits Modeled in VHDL"*, Ph.D. Thesis, Virginia Tech, 1998

[MiczoA86] A. Miczo, *Digital Logic Testing and Simulation*, Harper and Row Publishers, NY, 1986

[Motorola91] *M68HC11 Reference Manual*, Motorola Inc, 1991

[ScheiberS95] S. F. Scheiber, *Building a Successful Board Level Test Strategy*, Butterworth-Heinemann, 1995

[SpasovP96] Peter Spasov, *Microcontroller Technology, The 68HC11*, Prentice Hall Inc, Englewood cliffs, NJ, 1996

[TurinoJ90] Jon Turino, *Design to Test*, Van Nostrand Reinhold, New York , 1990