VLSI Testing PA3 Report

R05921058

楊承翰

1. Testcase result

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| circuit number | number of gates | number of TDFs | number of detected faults | number of undetected faults | fault coverage |
| C432 | 245 | 1110 | 3 | 1107 | 0.27 % |
| C499 | 554 | 2390 | 1324 | 1066 | 55.4 % |
| C880 | 545 | 2104 | 608 | 1496 | 28.9 % |
| C1355 | 554 | 2726 | 426 | 2300 | 15.6 % |
| C2670 | 1785 | 6520 | 3997 | 2523 | 61.3 % |
| C3540 | 2082 | 7910 | 861 | 7049 | 10.9 % |
| C6288 | 4800 | 17376 | 13180 | 4196 | 75.9 % |
| C7552 | 5679 | 19456 | 14939 | 4517 | 76.8 % |

1. Code explanation
2. First we need to generate fault list (generate\_tdf\_fault\_list) using similar algorithm as the original one. The difference is that we only collapse equivalent faults for BUF and NOT gates, because some equivalent stuck-at faults are not equivalent TDFs. For example, in c17.ckt, g2’s input from PI3 s-a-0 is equivalent to g2’s output s-a-1, but when applying pattern T’01111 1’, the first fault is not activated, while the second fault is activated and detected.
3. To record if a fault is activated in the first pattern, I add a member variable to struct FAULT to mark if the fault is activated.

|  |
| --- |
| struct FAULT {  ...  short activate;  ...  }; |

1. Simulate the first pattern and mark the activated faults

|  |
| --- |
| for (i = 0; i < ncktin; ++i) {  nv = ctoi(vector[i]);  sort\_wlist[i]->value = nv;  }  for (i = 0; i < ncktwire; ++i) {  if (i < ncktin) sort\_wlist[i]->flag |= CHANGED;  else sort\_wlist[i]->valie = 2;  }  sim();  for (f = flist; f; f = f->pnext\_undetect) {  if (f->fault\_type == sort\_wlist[f->to\_wlist]->value)  f->activate = TRUE;  else  f->activate = FALSE;  } |

1. Apply second pattern and run fault simulation (transition\_sim\_v2). The only change in fault simulation is that we need to additionally check if a fault is activated.

|  |
| --- |
| for (i = 0; i < ncktin; ++i) {  if (i == 0) nv = ctoi(vector[ncktin]);  else nv = ctoi(vector[i - 1]);  sort\_wlist[i]->value = nv;  }  for (i = 0; i < ncktwire; ++i) {  if (i < ncktin) sort\_wlist[i]->flag |= CHANGED;  else sort\_wlist[i]->valie = 2;  }  flist = transition\_sim\_v2(flist, num\_of\_current\_detect); |