VLSI Testing and Design for Testability Assignment 3

311510173 魏子翔

1. How to compile codes

   Go to the folder /podem, then enter "*make*" in command line to compile all codes.

   Then, enter "*./atpg -simulator circuit.cc -input circuit.input circuit.bench*" in command line to generate circuit.cc file. Type "*g++ circuit.cc*", and we will get the file "a.out".

   Finally, typing "*./a.out*" will generate a file "circuit.out".

2. The algorithm and idea of my codes

   A. In this assignment, I create the function called *void CompiledCodeSimulator(const char* cc_file)* to execute the a compiled code simulator word.

   B. Apply function *ReadNextPattern()* so that I can get the value of PI from function *GetValue1()* and *GetValue2()* in *main()*.

   C. In *Evaluate()*, using the variable *Queue[]* defined in circuit.h to get the relation between each gate, then the logic computation could be completed.

   D. Use *PIGate(i)* and *POGate(i)* defined in circuit.h to finish *PrintIO()*.

3. Results and discussion

   A. Pack different number of patterns into a simulator run:

   I test 4 different circuits (c17, c432, c1908, c7552) to compare CPU times and memory usage respectively. There are 1M patterns in each pattern files. We can see that when the larger packed pattern we used, the lower CPU time we spend. However, there isn't obvious change in memory usage.

(1). c17.bench

Number of packed pattern = 16	Number of packed pattern = 8
total CPU time = 6.04 5681 503 421 28 0 170 0 3115101+ 124700 0.0 0.0 -plogicsim -input ../c17_1 3115101+ 124701 46.4 0.0 3115101+ 124727 0.0 0.0 1 3115101+ 124729 0.0 0.0 1 Average memory usage: 0 Maximum memory usage: 2144	total CPU time = 6.08 5681 503 421 28 0 170 0 3115101+ 124258 0.0 0.0 %M\n ./atpg -plogicsim -inp 3115101+ 124259 43.4 0.0 ench 3115101+ 124287 0.0 0.0 1 3115101+ 124289 0.0 0.0 1 Average memory usage: 0 Maximum memory usage: 2144

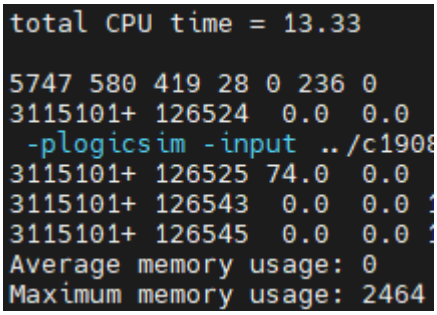
Number of packed pattern = 4	Number of packed pattern = 1
total CPU time = 6.26 5681 503 421 28 0 170 0 3115101+ 126368 0.0 0.0 -plogicsim -input ../c17_1 3115101+ 126369 48.1 0.0 3115101+ 126396 0.0 0.0 1 3115101+ 126398 0.0 0.0 1 Average memory usage: 0 Maximum memory usage: 2144	total CPU time = 6.76 5681 503 421 28 0 170 0 3115101+ 128147 0.0 0.0 -plogicsim -input ../c17_ 3115101+ 128148 48.2 0.0 3115101+ 128174 0.0 0.0 3115101+ 128176 0.0 0.0 Average memory usage: 0 Maximum memory usage: 2144

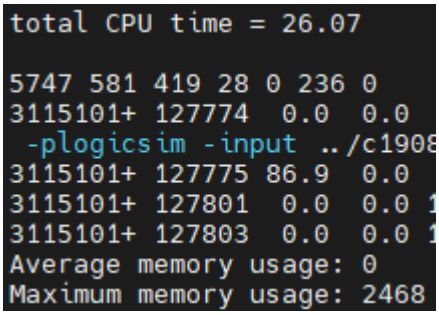
(2). c432.bench

Number of packed pattern = 16	Number of packed pattern = 8
total CPU time = 7.67 5714 527 419 28 0 203 0 3115101+ 126434 0.0 0.0 -plogicsim -input ../c432_ 3115101+ 126435 47.9 0.0 3115101+ 126449 0.0 0.0 1 3115101+ 126451 0.0 0.0 1 Average memory usage: 0 Maximum memory usage: 2252	total CPU time = 7.83 5714 527 419 28 0 203 0 3115101+ 125954 0.0 0.0 -plogicsim -input ../c432_ 3115101+ 125955 48.9 0.0 3115101+ 125986 0.0 0.0 1 3115101+ 125988 0.0 0.0 1 Average memory usage: 0 Maximum memory usage: 2252
Number of packed pattern = 4	Number of packed pattern = 1
total CPU time = 8.11 5714 527 419 28 0 203 0 3115101+ 124360 0.0 0.0 -plogicsim -input ../c432_ 3115101+ 124361 50.6 0.0 3115101+ 124378 0.0 0.0 1 3115101+ 124380 0.0 0.0 1 Average memory usage: 0 Maximum memory usage: 2256	total CPU time = 12.77 5714 527 419 28 0 203 0 3115101+ 127730 0.0 0.0 -plogicsim -input ../c432_ 3115101+ 127731 79.8 0.0 3115101+ 127746 0.0 0.0 1 3115101+ 127748 0.0 0.0 1 Average memory usage: 0 Maximum memory usage: 2256

(3). c1908.bench

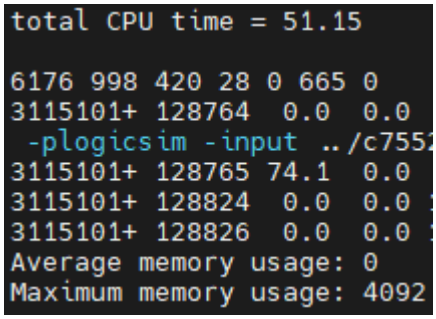
Number of packed pattern = 16	Number of packed pattern = 8
total CPU time = 8.82 5747 597 436 28 0 236 0 3115101+ 124418 0.0 0.0 -plogicsim -input ../c1908 3115101+ 124419 51.8 0.0 3115101+ 124435 0.0 0.0 1 3115101+ 124437 0.0 0.0 1 Average memory usage: 0 Maximum memory usage: 2464	total CPU time = 10.44 5747 598 436 28 0 236 0 3115101+ 126039 0.0 0.0 -plogicsim -input ../c1908 3115101+ 126040 61.4 0.0 3115101+ 126056 0.0 0.0 1 3115101+ 126058 0.0 0.0 1 Average memory usage: 0 Maximum memory usage: 2468
Number of packed pattern = 4	Number of packed pattern = 1



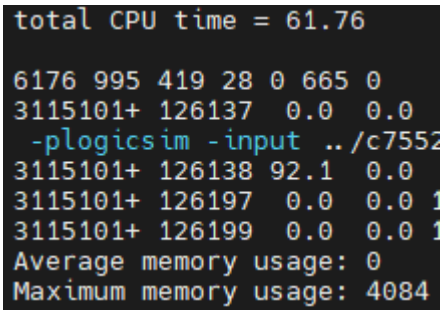


(4). c7552.bench

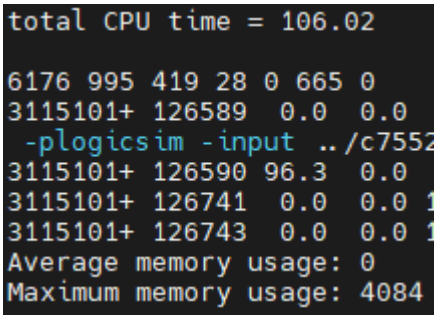
Number of packed pattern = 16



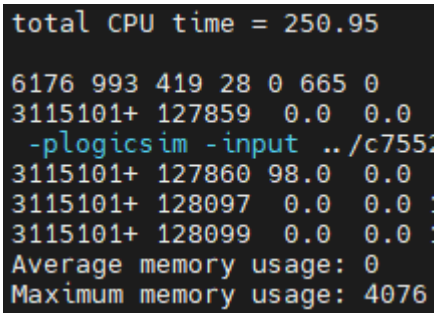
Number of packed pattern = 8



Number of packed pattern = 4



Number of packed pattern = 1



B. Calculate the average number of gate evaluations for different number of packed pattern:

The average number of gate evaluations				
	Number of packed pattern			
Circuit	16	8	4	1
c17.bench	0.5906	1.1224	2.2961	7.1011
c432.bench	13.5322	25.8283	51.4210	148.1900
c1908.bench	43.4230	83.5353	160.9616	415.5621
c7552.bench	350.1672	686.6690	1356.4649	3320.1058

The percentage of average gate evaluations					
	Number of packed pattern				Total gates
Circuit	16	8	4	1	
c17.bench	4.2186%	8.0171%	16.4007%	50.7221%	14
c432.bench	4.6987%	8.9682%	17.8545%	51.4549%	288
c1908.bench	4.6195%	8.8867%	17.1236%	44.2087%	940
c7552.bench	5.8420%	11.4559%	22.6304%	55.3905%	5994

The average number of gate evaluations has correlation with the CPU times. We can see the table upon. When CPU times increases, the average number of gate evaluations also increments obviously. Moreover, the more packed number we apply, the lower average number of gate CPU evaluates.

C. A compiled code simulator

There are 10000 patterns in each pattern.

A compiled code simulator based on a fixed number (16) of parallel patterns		
Circuit	Compiled	Origin
c17.bench	Total CPU Time = 0.04 Average memory usage: 0 Maximum memory usage: 1956	total CPU time = 0 Average memory usage: 0 Maximum memory usage: 2120
c432.bench	Total CPU Time = 0.1 Average memory usage: 0 Maximum memory usage: 3576	total CPU time = 0.04 Average memory usage: 0 Maximum memory usage: 2232
c1908.bench	Total CPU Time = 0.12 Average memory usage: 0 Maximum memory usage: 3460	total CPU time = 0.07 Average memory usage: 0 Maximum memory usage: 2444
c7552.bench	Total CPU Time = 0.45 Average memory usage: 0 Maximum memory usage: 12876	total CPU time = 0.36 Average memory usage: 0 Maximum memory usage: 4068

The CPU times the compiled code simulators spend are larger than the original ones. And the memory usage are also bigger than the original ones, when the circuit get complexity. I think that it's because the compiled code just do logic computation only, so the memory usage relies on pattern numbers