VLSI testing - Assignment 3

309510133 - Cheng-Cheng Lo

Different Pack Numbers

parallel logic simulation with different packed input numbers

c17.bench (5 inputs, 2 outputs, 14 gates)

- c17.input (24 test cases)

pack number	1	4	8	16
CPU time	0.000138	0.000181	7.1e-05	8.6e-05
Max. memory	828	820	812	828
Total gate evaluations num.	216	54	27	18
Avg. gate evaluations num. (/pattern)	9	2.25	1.13	0.75
Avg. gate evaluations **/** gates number (%)	64.29	16.07	8.07	5.36
Avg. gate evaluations **/** non-input gates number (%)	100	25	12.5	6.26

The total gate evaluations number is calculated as follows.

- n : number of input patterns (24 in this example)
- x : packed input number (1, 4, 8, 16 respectively)
- c : number of gate evaluations for a **single pattern**

$$total\ \ gate\ \ evaluations\ \ num. = c * \lceil \frac{n}{x} \rceil$$

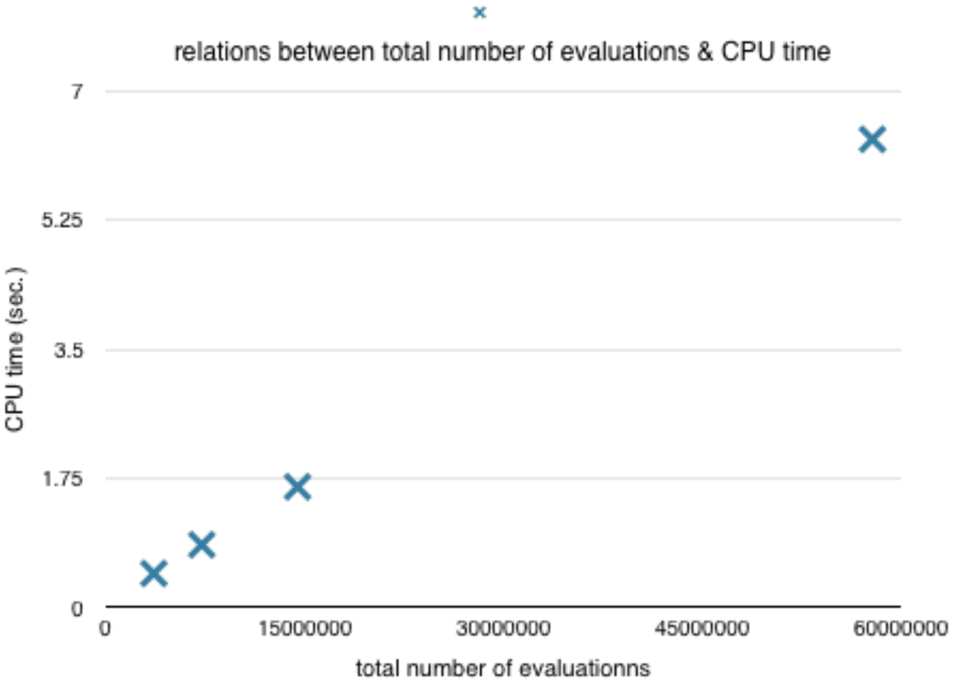
It's a quite small test case, which each data can be greatly affected by other factors (e.g. the time for parsing may account for a large part). Let's see the example below.

c7522.bench (207 inputs, 108 outputs, 5994 gates)

- c7552-10000.input (10000 input patterns)

pack number	**1**	**4**	**8**	**16**
total CPU time	6.34783	1.63688	0.85108	0.462134
Max. memory	2572	2588	2568	2576
Total gate evaluations num.	57870000	14467500	7233750	3616875
Avg. gate evaluations num. (/pattern)	5787.00	1446.75	723.38	361.69
Avg. gate evaluations **/** gates number (%)	96.55	24.14	12.07	6.03
Avg. gate evaluations **/** non-input gates number (%)	100	25	12.5	6.25

In this example, n = 10000, x = 1, 4, 8, 16 for different cases and c = 5787. Total number of gate evaluations is calculated as the formula above.



- When several input pattern are packed into one, the total number of evaluations greatly decreases and hence reduce the execuiton time.
- When the pack number is 1, we are simply doing compiled code simulation where total number of evaluations equals to **total number of gates - number of input gates**.
- Avg. gate evaluations **/** gates number (%) is **average number of gate evaluations divided by total number of gates**. If we replace the divisor by **total number of gates - number of input gates** (i.e. total number of non-input gates), the number will be perfectly 100, 25, 12.5, 6.25 for each cases.
- It seems that there are no absolute relations between memory usage and pack numbers.

Event-driven vs. Compiled Code

c7522.bench

- c7552-10000.input (10000 input patterns)

	Event-driven (-logicsim)	Compiled code (-simulator)
total CPU time	0.963047	0.429604
Max. memory (MB)	2524	13416

Compiled code runs faster than event-driven since it pack serval input patterns into one, which greatly reduces the number of evaluations. However, it also uses more memory since all patterns are inside the code; event-driven reads line-by-line and print the result after each calculation. Also, for the compiled code case, the size of the code generated is 6.3 MB, which takes a while to compile.

Build

```
make
./atpg -simulator c17.cc -input c17.input c17.bench
g++ c17.cc
./a.out
```