

# CS6132 Advanced Logic Synthesis

## Final Project Report

111062584 王領崧

- Design flow
  - STG Creation  
使用 C++ library fstream 和 stringstream 將 .kiss 一行行讀取進來後進行 string, int 的拆解。有宣告兩個 class (**State, Transition**) 紀錄資訊, 方便後續程式執行所需, State 包含 name, transitions, probability, encoding, Transition 包含 input/output, next state, probability。
  - State probability calculation  
所有 state 的 probability 加總為 1 之外, 每個 state 的 fan-in transition 也可以寫成一個方程式。假設有 **N** 個 state, 我們就可以列出 **N+1** 條方程式, 再透過 glpk solver, 將 N+1 條方程式設為 constraint, objective 設為 minimize 所有 state 加總的 probability, 就能解出每個 state 各自的 probability。
  - Initial Encoding of FSM  
使用講義 chapter6 裡頭 adjacency-based state assignment 的 state-pair weight assignment, 包含 fan-in / fan-out oriented, 總共有 4 種不同情況。我會依序對每一種情況, 掃過所有的 transitions, 增加符合條件 state-pair 的 weight, 完成後使用 LEDA library 中 maximum weight matching algorithm, 找出 initial state-pair solution。最後再對每組 state-pair 進行 encoding, state-pair 的兩個 state 只會有 LSB 不同。
  - Optimizing cost function  
經過 initial encoding 之後, 會將每組 state-pair 的兩個 state 設為同一個 group。我透過 simulation annealing algorithm (SA) 進行優化 cost function, 設計了兩種 perturbation, 分別為 **swap state encoding** 和 **swap group encoding**。Swap state 會替換兩個不同 group 的 state encoding, swap group 則是將兩個 group 裡面的 state encoding 進行交換。
  - SIS synthesis and power report  
透過助教提供的 sis script, 將完成 state assignment 的 blif 檔合成 -> mapping -> 最後進行 power consumption 的計算。
- Cost function

$$\sum_{\substack{\text{all pairs} \\ s, t}} w(s, t) \times \text{dist}(\text{enc}(s), \text{enc}(t))$$

$w(s, t)$ : state s 的機率 \* transition (s->t) 的機率 + state t 的機率 \* transition (t->s) 的機率。

$\text{Dist}(\text{enc}(s), \text{enc}(t))$ : State s, t 兩者的 hamming distance。

- Result  
實驗結果顯示進步幅度很有限, 並且 testcase snad 會有奇怪的 bug 無法執行, 我覺得可能是我的 cost function 定義的不太好, 但也不知道要怎麼修改了 QQ

	bbara	bbsse	bbtas	beecount	cse	dk14	dk16	ex1	s27	snad
Initial	439.5	1223	241.2	351.4	2327.7	1115.6	3027.4	3483.7	419.8	x
Final	425.6	1087	217.1	351.4	2327.7	1232	2777.4	3126.8	287.8	x

- README: project 裡面有 readme 說明如何編譯和執行我的作業