CS613200 Final Project

109062630 鄭傑予

**Date Structure**

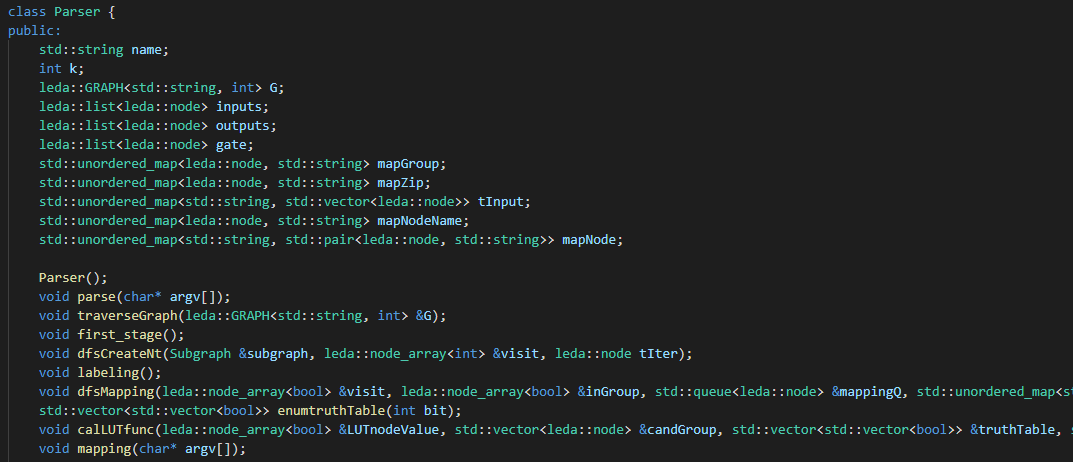
我總共建立了兩個class, 分別為Parser及Subgraph，裡面包含了c++ stl 的一些資料結構，像是unordered\_map, vector, queue等等，以及LEDA的資料結構, 像是GRAPH, node, edge, list, node\_array等等。

Parser負責讀檔寫檔, 主要的data structure維護, 以及在FlowMap演算法中各個流程的function.

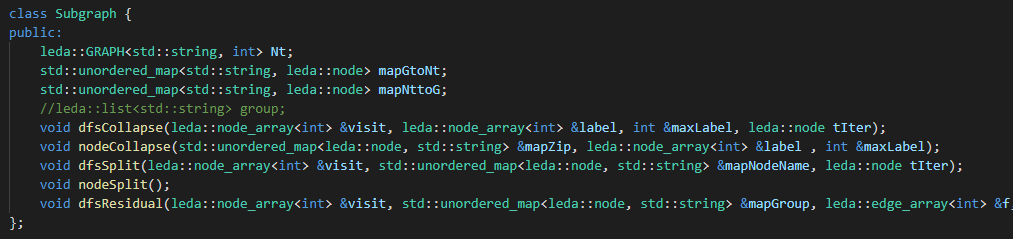
Subgraph負責在FlowMap的labeling phase中所需的data structure及function.

因為在計算子圖時所需要的data僅為暫存用的，因此特地建立這個class來降低演算法的空間複雜度。

Parser:

****

Subgraph:

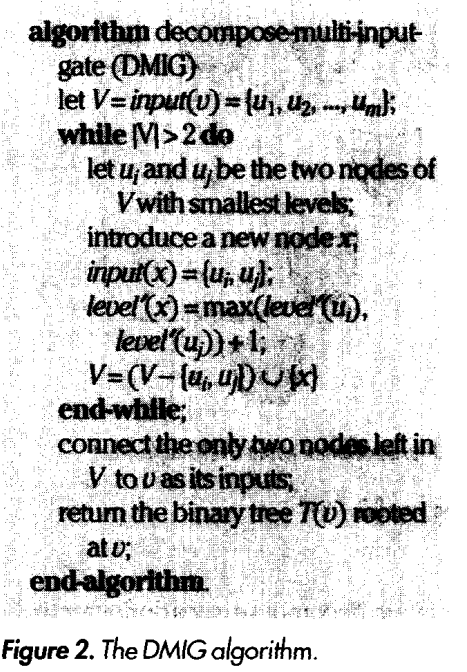
****

**Algorithm**

我參考了FlowMap: An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs這篇論文，以及跟實驗室同學互相討論，來實作這篇的演算法。

**Stage1**

我會根據FlowMap這篇論文的第三篇reference paper, DMIG來當作decomposition的演算法。



首先從graph中找出input > 2的gate, 接著調整edge weight後用shortest path演算法找出此gate的longest path, 就會得到每個input wire的level, 將最小的兩個wire接到一個新建立的gate上後，原本的gate就少了一個input, 持續做到此gate變成2-input gate.

**Stage2**

這個stage又分為Labeling Phase及Mapping Phase.

**Labeling Phase**

這個階段會依照traversal order對每個gate t標示label, label用來決定t能否在t的子圖中跟最大的label的gate包成同一個LUT, 如果能找到k-feasible cut就代表可以，找不到的話t的label被標示成maxlable+1. 因此為了判斷是否存在k-feasible cut, 會對subgraph進行dfsCollapse, nodeSplit及dfsResidual等等的functoin, 最後得到的是能跟t包成同一個LUT的set.

dfsCollapse 將t跟子圖中maxlabel的gate合併成一個node.

nodeSplit 將除了source跟t的所有node split, 建立一個分身，分身之間的edge weight = 1, 其餘的edge weight = 999.

dfsResidual 會從t進行dfs走訪，找出能跟他包進同一個LUT的set.

**Mapping Phase**

這個階段會從每個primiry output進行LUT mapping, 並將他的input wire放進queue中等待mapping, 接著會enumerate這個LUT的所有function, 找出onset之後進行寫檔，結束整個流程。

**Results**

k=3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 10aoi.sample01 | 10aoi.sample02 | 10aoi\_alu4 | 10aoi\_des |
| Level | 2 | 2 | 17 | 10 |
| LUTs | 4 | 7 | 936 | 5854 |

k=4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 10aoi.sample01 | 10aoi.sample02 | 10aoi\_alu4 | 10aoi\_des |
| Level | 1 | 2 | 12 | 7 |
| LUTs | 2 | 5 | 801 | 4238 |