DSnP Final Project

FRAIG

B07705049 資管二 林稜凱

email: b07705049@ntu.edu.tw

0. Class design of CirGate

```
class CirGate:
        class CirPiGate:        public CirGate
        class CirPoGate:        public CirGate
        class CirAigGate:       public CirGate
        class CirUndefGate:     public CirGate

class member:
        unsigned    _gateID;            bool        _invPhase0;
        unsigned    _lineNo;            bool        _invPhase1;
        unsigned    _fanin0;            string      _symbol;
        unsigned    _fanin1;

        static unsigned         _globalRef;     // DFS
        unsigned                _ref            // DFS
        vector<unsigned>        _fanout;

        bool                    _sim            // 儲存 simulate 時的值
        vector<unsigned>*       _fecGroup;      // 指向 fecGroup，可能
                                                   與其他 gate 共用

class function:
        void DFS();                 // for constructing dfsList
        bool simulateByDFS();       // 在 simulate 時用 DFS 來算出值
```

1. Sweeping

演算法：

```
sweep()
begin
        從各個 PO 出發做 DFS，紀錄會經過的 gate
        for all 沒有在 DFS 經過的 gate do:
                removeGate(gate);
end

removeGate(gate) {
        for all fanin of gate do:
                移除 fanout 中的此 gate;
                if fanout of the fanin become empty then:
                        _unused.append(this fanin);
        delete gate;
}
```

在 remove gate 時，不需對此 gate 的 fanout 做處理，因為若此 gate 會被移除，代表此 gate 的 fanout 也會被移除。

2. Optimization

演算法：

```
optimize()
begin
        while(true)
        {
                for all fanout of const0 do:
                        令此 fanout 有兩個 fanin: const0 和 fanin
                        if 此 fanout 有一個 fanin 是 const0 then:
                                replaceByGate(fanout, const0);
                        else if 此 fanout 有一個 fanin 是 const1 then:
                                replaceByGate(fanout, fanin);

                for all aig in dfsList do:
                        if 此 aig 的兩個 fanin 相同 then:
                                if 兩個 fanin 的 _inv 值相同 then:
                                        replaceByGate(aig, fanin);
                                else if 兩個 fanin 的 _inv 值相反 then:
                                        replaceByGate(aig, const0);

                當沒有新的 gate 被移除時，跳出 while 迴圈。
        }
end

replaceByGate(gate1, gate2) {   // replace gate1 by gate2
                // 將 gate1 的 fanout 與 gate2 連接
                // 將 gate1 從 _aigList 等地方移除
}
```

3. Strash

演算法：

```
begin
        map := stl/unordered_map<hashKey, CirGate*>
        for all aig in _dfsList do:
                generate a hashKey using two fanins of the aig
                if !map[hashKey] then:
                        map[hashKey] = &aig;
                else:
                        replaceByGate(aig, map[hashKey]);
        regenerateDFSlist();
end
```

4. Simulation

演算法：

begin

```
fecPairs := queue<vector<CirGate*>>
if no simulation before then:
        vector<CirGate*> v := vector of all aigs in _dfsList;
        fecPairs.push(v);

while(!stopping_criteria)
{
        // get pattern (randomly generate or from the SAT value of FRAIG or by
           file)
        // simulate (update the values of all gates by the pattern)
        v = fecPairs.pop();
        // 將 v 裡的 gate 依照 simulate 結果分到一個或多個 vector(s) 中
        // 將這些 vector(s) push 進 fecPairs 中
}
```

end

stopping_criteria：

- 連續 30 次沒有產生新的 fecPair 或 simulation 次數超過 4 * M。

5. Fraig

演算法：

```
begin
        SatSolver solver;
        solver.initialize();
        generateProofModel(solver);

        vector<string> SATpattern;

        for all aig in _dfsList do:
                if the aig does not belong to a fecPair then:
                        continue;
                if the fecPair of the aig does not have a leading gate then:
                        let the aig be the leading gate of the fecPair;
                else:
                        solve the SAT problem (aig != leading gate)
                        if !result (UNSAT) then:
                                replaceByGate(aig, leading gate);
                        else
                                save the values to SATpattern;
                regenerateDFSlist();
end
```