1. ­封面頁: 內容如下，請自行調整格式

|  |
| --- |
| **物件導向軟體開發技術-期末專案**  成績座號: 30 學號: 406637123  姓名: 卓妏育 |

1. Part I: 系統描述

請依照如下次序撰寫，儘量以條列方式撰寫

1. 系統架構圖: 若有自行設計，請劃出類別圖，功能方塊圖等
2. 資料結構設計:  
   使用HashMap對應每個閘道的輸出。
3. 演算法設計:   
   依照Output產生遞迴來計算結果，考慮電路可能是交叉輸入，無法分別計算。
4. 與Basic Sim差異之處: (必填)

使用遞迴的方式來算output的值，並且把儲存的容器變成HashMap，執行的速度沒有變快比原本的慢了一點，但是可以處理混亂電路並且答案都是對的。

1. Part II: 程式碼

* 列印所有程式碼(可加上行號)，自行設計部分，以灰底呈現

1. package Main;
2. import java.util.\*;
3. import java.io.\*;
4. public class Main {
5. static HashMap<String, Integer> gateOut = new HashMap<String, Integer>() ;
6. static HashMap<String, String[]> gateDef = new HashMap<String, String[]>() ;
7. static ArrayList<String> inputList = new ArrayList<String>() ;
8. static ArrayList<String> outputList = new ArrayList<String>() ;
9. //static ArrayList<String[]> gateList = new ArrayList<String[]>() ;
11. public static void main(String[] args) throws Exception {
12. long start = System.currentTimeMillis() ;
13. String benchFile = "c:/ISCAS85/c432.bench.txt" ;
14. parseBenchFile(benchFile) ; // build data structure
15. String ipFile = "c:/ISCAS85/c432\_1k\_ip.txt" ;
16. String opFile = "c:/ISCAS85/c432\_1k\_op.txt" ;
17. simulation(ipFile,opFile) ; // generate the result
18. System.out.printf("Total time=%.3f sec(s)\n",
19. (System.currentTimeMillis()-start)/1000.0) ;
20. }
21. // 模擬器函數
22. public static void simulation(String ipFile, String opFile) throws Exception {
24. BufferedReader br = new BufferedReader(new FileReader(ipFile)) ;
25. BufferedWriter bw = new BufferedWriter(new FileWriter(opFile)) ;
27. String ipvs = "", opvs = "" ;
29. // 讀取輸入訊號檔案(xxx\_ip.txt)，並進行邏輯閘運算，產生輸出。
30. while ((ipvs=br.readLine())!=null) {
31. // clear all gateOut
32. for (String gName : gateOut.keySet())
33. gateOut.replace(gName, 2);
35. // Input Data
36. ipvs = ipvs.trim() ; // 如"01010011"的輸入
37. fillInput(ipvs) ; // 將ipvs分解為0, 1, …，填入電路的輸入腳
39. // Output Data
40. opvs = "";
41. /\*
42. Runnable r[] = new Runnable[size];
43. Thread t[] = new Thread[size] ;
44. for (int i = 0 ; i < size; i++) { // evaluate gates one by one
45. rr[0] = 0;
46. String gName = outputList.get(i);
47. int k = i;
48. r[i] = ()-> new BasicSim().doSim(gName, k);
49. t[i] = new Thread(r[i]) ;
50. t[i].start();
51. t[i].join();
52. }
53. for (int j = 0; j < size; j++)
54. opvs = (rr[j] == 0 ? "0": "1");
56. \*/
57. for (int i = 0 ; i < outputList.size(); i++) { // evaluate gates one by one
59. String gName = outputList.get(i);
60. opvs += (doSim(gName) == 1 ? "1": "0");

//依照每個邏輯閘特性，進行運算

1. }
3. bw.write(ipvs + " " + opvs) ; // 蒐集的輸出，並寫至檔案
4. bw.newLine();
5. }
6. br.close(); bw.close() ;
7. }
8. public static void fillInput(String ipvs) {
10. if (ipvs.length() != inputList.size()) {
12. throw new java.lang.RuntimeException("Input Size mismatch:"+ipvs.length()+","+inputList.size()) ;
13. }
15. // set input gates value
16. for (int i = 0 ; i < ipvs.length(); i++) {
18. int q = ipvs.charAt(i)=='0' ? 0: 1;
19. gateOut.replace(inputList.get(i), q) ;
20. }
21. }
22. // ------ Gate Value Evaluation Fuctions ----------
23. public static int AND(int[] gateInfo) {
24. for (int i = 0; i < gateInfo.length; i++)
25. if (gateInfo[i] == 0)
26. return 0;
27. return 1;
28. }
29. public static int OR(int[] gateInfo) {
31. for (int i = 0; i < gateInfo.length; i++)
32. if (gateInfo[i] == 1)
33. return 1 ;
34. return 0;
35. }
36. public static int XOR(int[] gateInfo) {
38. int v1 = gateInfo[0], v2 = gateInfo[1];
39. return (v1==v2)? 0: 1;
40. }
42. public static int NAND(int[] gateInfo) { return (AND(gateInfo) == 0 ? 1: 0); }
43. public static int NOR(int[] gateInfo) { return (OR(gateInfo) == 0 ? 1: 0); }
44. public static int NXOR(int[] gateInfo) { return (XOR(gateInfo) == 0 ? 1: 0); }
46. // return 0 or 1
47. public static int doSim(String gName) {
48. int v = gateOut.get(gName);
49. if (v == 0 || v == 1)
50. return(v);
52. String[] p = gateDef.get(gName) ;

//[gName,"nand","G224","G898"] or [gName,"not","G146"]

2. v = doSim(p[2]);
3. switch(p[1]) {
4. case "not": // [null,"not","G146"]
5. v = (v == 0) ? 1: 0;
6. gateOut.replace(gName, v);
7. return(v);
9. case "buf": // [null,"buf","G146"]
10. gateOut.replace(gName, v);
11. return(v);
12. }
14. int[] q = new int[p.length-2];
15. q[0] = v;
16. for (int j = 1; j < q.length; j++)
17. q[j] = doSim(p[2+j]);
19. switch(p[1]) {
20. case "and":
21. v = AND(q);
22. break;
24. case "or":
25. v = OR(q);
26. break;
28. case "xor":
29. v = XOR(q);
30. break;
32. case "nand":
33. v = NAND(q);
34. break;
35. case "nor":
36. v = NOR(q);
37. break;
39. case "nxor":
40. v = NXOR(q);
41. break;
43. default:
44. throw new java.lang.RuntimeException("Unknown Gate:"+p[1]) ;
45. }
47. gateOut.replace(gName, v);
48. return(v);
49. }
51. // ------ Parsing Circuit File and Build Data Structure ----------
52. //
53. // 產生 Gate 的 HashMAP
54. // INPUT: ("G953", {"","in"})
55. // GATE: ("G231", {"", "nand", "G224", "G898"})
56. // ("G206", {"", "not", "G146"})
57. // OUTPUT: ["G3","G4",...]
58. //
60. public static void parseBenchFile(String benchFile) throws Exception {
61. BufferedReader br = new BufferedReader(new FileReader(benchFile)) ;
62. String aLine = "" ;
63. String gName = "";
65. int ni = 0, no = 0, ng = 0;
66. while ((aLine=br.readLine()) != null) {
68. if (aLine.startsWith("#")|| aLine.trim().length()==0 ) continue ;
70. if (aLine.startsWith("INPUT")) {
72. //INPUT(G953)
73. String[] tt = aLine.split("\\(") ;
74. gName = tt[1].replace(")","") ;
76. gateOut.put(gName,2); // input gate = 2 (undefine)
77. inputList.add(gName); //["G1","G2",...]
78. ni++;
79. }
80. else if (aLine.startsWith("OUTPUT")) {
82. //OUTPUT(G3)
83. String[] tt = aLine.split("\\(") ;
84. gName = tt[1].replace(")","") ;
86. outputList.add(gName); //["G3","G4",...]
87. no++;
88. }
89. else {
90. //G206 = not(G146)
91. //G231 = nand(G224, G898)
92. aLine = aLine.replace(" ","") ; //G206=not(G146)
93. aLine = aLine.replace("=",",") ; //G206,not(G146)
94. aLine = aLine.replace("(",",") ; //G206,not,G146)
95. aLine = aLine.replace(")","") ;

//G231,nand,G224,G898 or G206,not,G146

2. String[] p = aLine.split(",") ; //G231 nand G224 G898
3. gName = p[0];
4. p[1] = p[1].toLowerCase(); //to lower case
5. gateDef.put(gName,p);

//("G231", ["G231", "nand", "G224", "G898"])

1. //("G206", ["G206", "not", "G146"])
2. //("G216", ["G216", "buf", "G146"])
3. gateOut.put(gName,2); //2: undefine!
4. ng++;
5. }
6. }
7. System.out.println(ni+" Input, "+no+" Output, "+ng+" Gates" );
8. br.close() ;
9. }
10. }
11. Part III : 系統測試表 (必填)

* 內容如最末頁，請單獨成為一頁
* 請具實填寫，會請助教實際執行，進行覆核

**期末專案系統測試表** 學號: 406637123 姓名: 卓妏育

1. [基本測試]: 以自行撰寫版本進行基本測試
   * 使用c432.bench+c432\_1k\_ip.txt，結果不正確者，不進行後續測試。

☑正確 □不正確

1. 電腦效能基本測試
2. 電腦配備

|  |  |  |  |
| --- | --- | --- | --- |
| CPU | RAM | OS | HDD |
| Intel(R) Core(TM) i5-5300U CPU @2.30GHz 2.30GHz | 8.00 GB | 64位元作業系統，x64型處理器 | CT240BX500SSD1 |

1. BasicSim執行時間 (打V處需填註執行時間，若無法執行，請註明# )

|  |  |  |
| --- | --- | --- |
|  | C432 | C7552 |
| 1M | 17.434 | 387.698 |
| 5M | V |  |
| 10M | 154.004 |  |

|  |  |  |
| --- | --- | --- |
|  | C432 | C7552 |
| 1M | 17.474 | 475.326 |
| 5M | V |  |
| 10M | 171.784 |  |

1. 執行時間測試: 以秒為單位，小數點後1位。

- 測試各電路執行時間時(打V處)，需先確定結果正確，否則沒有意義。

|  |  |  |  |
| --- | --- | --- | --- |
|  | C432 | C2670 | C7552 |
| 1K | 0.313 |  | 1.062 |
| 10K | V | 2.807 | 6.687 |
| 1M | 18.014 |  | 475.326 |
| 5M | V |  |  |
| 10M | 171.784 |  |  |

1. 通過步驟3者，測試能否模擬混亂化過後之電路: ☑有 □無

|  |  |  |
| --- | --- | --- |
|  | C432 | C7552 |
| 1K | 0.359 | 1.375 |
| 10K | V | 9.077 |
| 1M | 25.525 | 590.877 |