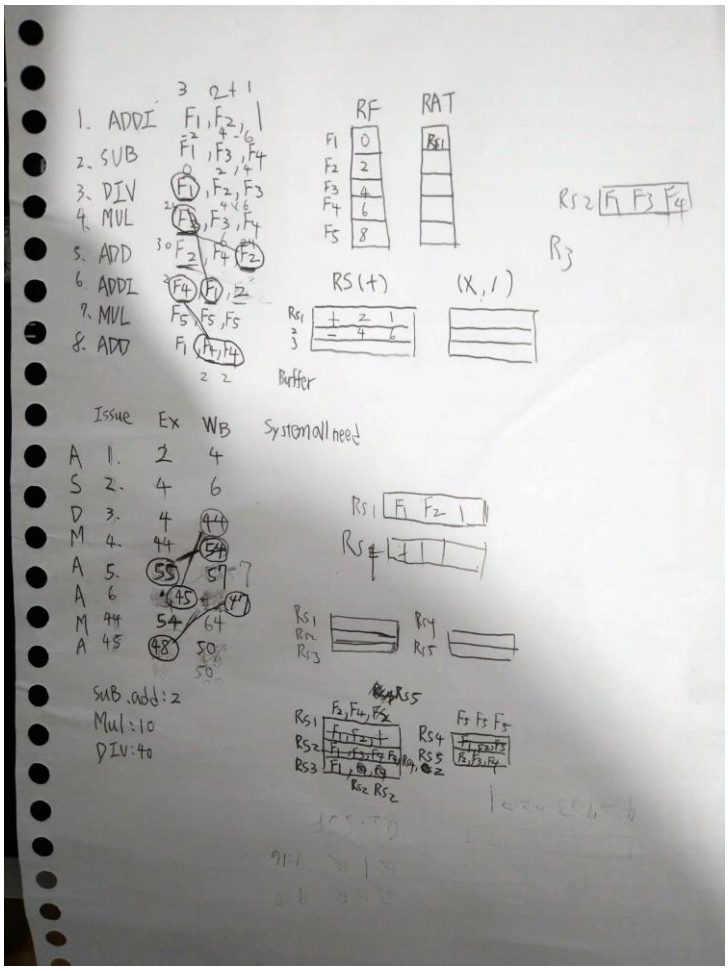
文字說明:



先用手寫直接把表建出來(預設 ADD,SUB,ADDI 需要兩個 cycle,MUL 需要 10cycle,DIV 需要 40 個 cycle),建完表格後可以知道需要 64 個 cycle 去完成這個 input 所以設 for(1~64),再來就是找有變化的 cycle 所以把表格內的數字都記下來只要 if 有遇到(changedetect)就會進行一次 output,然後先用 detectwriteback(clock);去找每個 cycle 結束時要 wr 的 RS 然後每當 issue 表格裡面的第一直行數字被遇見時要寫到 RAT 和 RS,然後 buffer_alu();去判斷哪些 Inst 要 excute 何時放在 Buffer 裡面。

程式碼:

```
#include<iostream>
#include<string.h>
#include<vector>
using namespace::std;
const int Num_ADD_RS = 2;
const int Num_SUB_RS = 2;
const int Num_MULT_RS = 10;
const int Num_DIV_RS = 40;
```

```
string input[8] = { "ADDI F1, F2, 1","SUB F1, F3, F4","DIV F1, F2, F3","MUL F2, F3, F4","ADD F2, F4, F2","ADDI F4, F1, 2","MUL F5, F5, F5","ADD F1, F4, F4" };
	int F1 = 0,F2=2,F3=4,F4=6,F5=8;
	int RS1 = 0, RS2 = 0, RS3 = 0, RS4 = 0, RS5 = 0;
	string F1_RAT = "	", F2_RAT = "	", F3_RAT = "	", F4_RAT = "	", F5_RAT = "	";
	string RS1_1="	", RS1_2="	", RS1_3="	", RS2_1="	", RS2_2="	", RS2_3="	", RS3_1="	", RS3_2="	", RS3_3="	";
	string RS4_1 = "	", RS4_2 = "	", RS4_3 = "	", RS5_1 = "	", RS5_2 = "	", RS5_3 = "	";
	string ALU_plus = " ", ALU_mult = " ";
bool changedetect(int time){
	if (time == 1 || time == 2 || time == 3 || time == 4 || time == 5 || time == 6 || time == 44 || time == 45 || time == 47 || time == 48 ||time == 50	|| time == 54 ||time == 55 || time == 57 || time == 64)
		return true;
	else
		return false;
}
void buffer_alu(int a,int time) {//excute
	if (a == 0) {
		if (2==time||time==3) {
			ALU_plus = "(RS1) 2+1";
		}
		else if (4==time|| time== 5) {
			ALU_plus = "(RS2) 4-6";
		}
		else if (55==time||time == 56) {
			ALU_plus = "(RS1) 6+24";//F2=6*4=24 wr
		}
		else if (45==time||time == 46) {
			ALU_plus = "(RS2) 0+2";//
		}
		else if (48== time||time == 49) {
			ALU_plus = "(RS2) 2+2";//F4=
		}
		else
			ALU_plus = "empty";
```

```
	}
	else if (a == 1) {
		if (time >= 4 && time <= 43) {
			ALU_mult = "(RS4) 2/4";
		}
		else if (time >= 44 && time <= 53) {
			ALU_mult = "(RS5) 4*6";
		}
		else if (time >= 54 && time <= 63) {
			ALU_mult = "(RS4) 8*8";
		}
		else{
			ALU_mult = "empty";
		}
	}
}
void issue(int time) {
	if (time == 1) {//放 RS1(4)
		F1_RAT = "RS1";
		RS1_1 = "+";
		RS1_2 = "2";
		RS1_3 = "1";
	}
	else if (time==2) {//放 RS2(6)
		F1_RAT = "RS2";
		RS2_1 = "-";
		RS2_2 = "4";
		RS2_3 = "6";
	}
	else if (time == 3) {//放 RS4(44)
		F1_RAT = "RS4";
		RS4_1 = "/";
		RS4_2 = "2";
		RS4_3 = "4";
	}
	else if (time == 4) {//放 RS5(54)
		F2_RAT = "RS5";
		RS5_1 = "*";
```

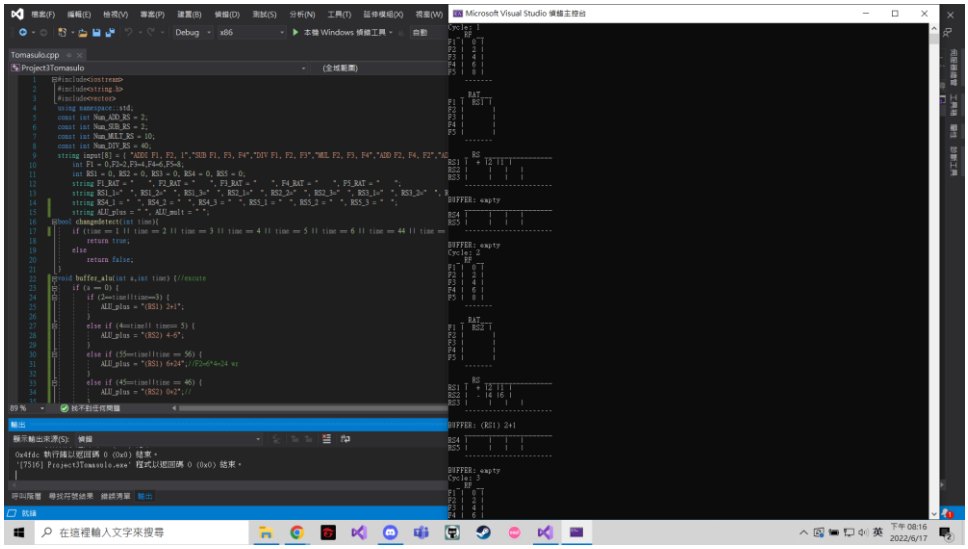
```
            RS5_2 = "4";
            RS5_3 = "6";
        }
        else if (time == 5) {//放 RS1(56)因為 RS1(4)WB
            F2_RAT = "RS1";
            RS1_1 = "+";
            RS1_2 = "6";
            RS1_3 = "RS5";
        }
        else if (time == 6) {//放 RS2(58)因為 RS2(6)WB
            F4_RAT = "RS2";
            RS2_1 = "+";
            RS2_2 = "RS4";
            RS2_3 = "2";
        }
        else if (time == 44) {//放 RS4(64)因為 RS4(44)WB
            F5_RAT = "RS4";
            RS4_1 = "*";
            RS4_2 = "8";
            RS4_3 = "8";
        }
        else if (time == 45) {//放 RS3
            F1_RAT = "RS3";
            RS3_1 = "+";
            RS3_2 = "RS2";
            RS3_3 = "RS2";

        }
}
/*void excute(int time) {
        //if(time==2)
}*/
void detectwriteback(int time) {
        if (time == 4) {//RS1=3
            //F1_RAT = "    ";不用清空因為不是他的 RSI2 覆蓋掉了所以也不用改 RF 值
            RS1_1 = "     ";
            RS1_2 = "     ";
            RS1_3 = "     ";
```

```
        F1 = 0;


    }
    else if (time == 6) {//RS2=-2
       // F1_RAT = "    ";同上不用清空因為不是他的 RSI3 覆蓋掉了所以也不用改 RF 值
        RS2_1 = "      ";
        RS2_2 = "      ";
        RS2_3 = "      ";
        F1 = 0;
    }
    else if (time == 44) {//RS4=0
        F1_RAT = "    ";
        RS4_1 = "      ";
        RS4_2 = "      ";
        RS4_3 = "      ";
        F1 = 0;
        RS2_2 = "    0    ";
    }
    else if (time == 54) {//RS5=24  同上不用清空因為不是他的 RS I8 等他執行完就覆蓋
掉了所以也不用改 RF 值只需要改 I5
        //F2_RAT = "    ";
        RS5_1 = "      ";
        RS5_2 = "      ";
        RS5_3 = "      ";
        F2 = 2;
        RS1_3 = " 24    ";
    }
    else if (time == 57) {//RS1=30
        F2_RAT = "    ";
        RS1_1 = "      ";
        RS1_2 = "      ";
        RS1_3 = "      ";
        F2 = 30;
    }
    else if (time == 47) {//RS2=2
        F4_RAT = "    ";
        RS2_1 = "      ";
        RS2_2 = "      ";
```

```
            RS2_3 = "      ";
            F4 = 2;
            RS3_2 = " 2    ";
            RS3_3 = " 2    ";
        }
        else if (time == 64) {//RS4=64
            F5_RAT = "    ";
            RS4_1 = "      ";
            RS4_2 = "      ";
            RS4_3 = "      ";
            F5 = 64;
        }
        else if (time == 50) {//RS3=4
            F1_RAT = "    ";
            RS3_1 = "      ";
            RS3_2 = "      ";
            RS3_3 = "      ";
            F1 = 4;
        }
}
int main() {
    int clock = 1;
        while(clock<=64){

            if( changedetect(clock)==1){
                detectwriteback(clock);
                if (clock == 1 || clock == 2 || clock == 3 || clock == 4 || clock == 5 || clock == 6
|| clock == 44 || clock == 45)
                    issue(clock);

                //if(clock)
                cout << "Cycle: " << clock<<"\n";
        cout <<"    _ RF __ "<<"\n";
        cout << "F1 |    " << F1 << " |" << "\n";
        cout << "F2 |    " << F2 << " |" << "\n";
        cout << "F3 |    " << F3 << " |" << "\n";
        cout << "F4 |    " << F4 << " |" << "\n";
        cout << "F5 |    " << F5 << " |" << "\n";
```

```
        cout << "       -------" << "\n"<<"\n";
        cout << "      _ RAT___ " << "\n";
        cout << "F1 |    " << F1_RAT << " |" << "\n";
        cout << "F2 |    " << F2_RAT << " |" << "\n";
        cout << "F3 |    " << F3_RAT << " |" << "\n";
        cout << "F4 |    " << F4_RAT << " |" << "\n";
        cout << "F5 |    " << F5_RAT << " |" << "\n";
        cout << "       -------" << "\n" << "\n";
        cout << "       _ RS ____________________" << "\n";
        cout << "RS1 |    " << RS1_1 << " |" << RS1_2 << " |" << RS1_3 << " |" << "\n";
        cout << "RS2 |    " << RS2_1 << " |" << RS2_2 << " |" << RS2_3 << " |" << "\n";
        cout << "RS3 |    " << RS3_1 << " |" << RS3_2 << " |" << RS3_3 << " |" << "\n";
        cout << "       ----------------------" << "\n" << "\n";

        buffer_alu(0,clock);
        cout << "BUFFER: "<< ALU_plus;
        cout << "\n";
        cout << "       _____________________" << "\n";
        cout << "RS4 |    " << RS4_1 << " |" << RS4_2 << " |" << RS4_3 << " |" << "\n";
        cout << "RS5 |    " << RS5_1 << " |" << RS5_2 << " |" << RS5_3 << " |" << "\n";
        cout << "       ----------------------" << "\n" << "\n";

        buffer_alu(1,clock);
        cout << "BUFFER: "<< ALU_mult;
        cout << "\n";
        }
            clock++;
        }
}
```

執行結果:


Tomasulo.cpp
Project3Tomasulo
Debug
x86
本機 Windows 偵錯工具
Microsoft Visual Studio 偵錯主控台
Cycle: 1
Cycle: 2
Cycle: 3
BUFFER: empty
BUFFER: (RS1) 2+1
0x4fdc 執行緒以返回碼 0 (0x0) 結束。
'[7516] Project3Tomasulo.exe' 程式以返回碼 0 (0x0) 結束。
在這裡輸入文字來搜尋
英
下午 08:16
2022/6/17