Lab 2: Scheduling

List Scheduling (1/3)
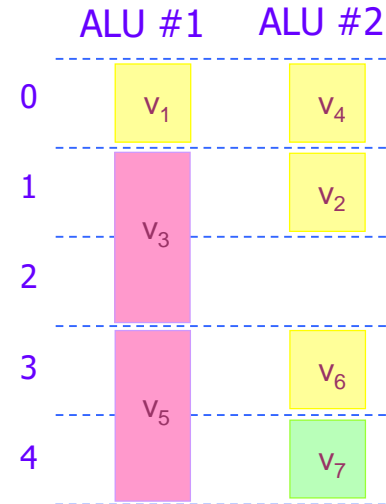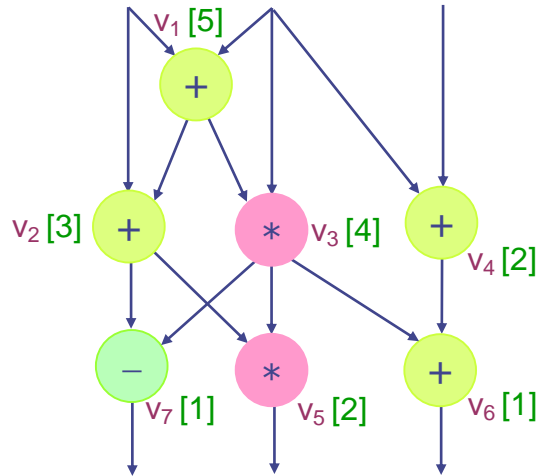
■ **List Scheduling**

  ◆ A resource-constrained scheduling method

  ◆ Start at time zero and increase time until all operations have been scheduled

  ◆ The ready list contains all operations that can start their execution at the current time step or later

  ◆ If more operations are ready than there are resources available, use some priority function to choose, e.g. the longest-path to the output node $\Rightarrow$ *critical-path list scheduling*

List Scheduling (2/3)

INSERT_READY_OPS (V, $PList_{t_1}$, $PList_{t_2}$, ..., $PList_{t_m}$);

Cstep = 0;

while ($PList_{t_1} \neq \phi$) or ... or ($PList_{t_m} \neq \phi$)) do

    Cstep = Cstep + 1;   /* $PList_{t_k}$: priority list for operation type t_k */

    for k = 1 to m do   /* N_{t_k}: number of function units performing operation of type t_k */

        for funit = 1 to N_k do

            if $PList_{t_k} \neq \phi$ then

                SCHEDULE_OP($S_{current}$, FIRST($PList_{t_k}$), Cstep);

                $PList_{t_k}$ = DELETE($PList_{t_k}$, FIRST($PList_{t_k}$));

            endif

        endfor

    endfor

    INSERT_READY_OPS (V, $PList_{t_1}$, $PList_{t_2}$, ..., $PList_{t_m}$);

endwhile

List Scheduling (3/3)

t	Ready List
0	{ v_1 [5], v_4 [2] }
1	{ v_3 [4], v_2 [3] }
2	ϕ
3	{ v_5 [2], v_6 [1], v_7 [1] }
4	{ v_7 [1] }

Lab 2: Scheduling

■ 下載並安裝**Dev-C++**

■ 參閱**List Scheduling**程式範例

■ 撰寫**List Scheduling**的**C/C++**程式

■ 以**List Scheduling**程式進行**DFG1**以及**DFG2**的排程

■ 以**List Scheduling**程式進行**RGB to YUV**的排程

■ 撰寫並繳交實驗報告

#define Size 200

```
struct cdfg{                struct readylist{            struct alu{
    int op;                     int state;                   int mult;
    int op1;                    int op;                      int add;
    int op2;                    int num;                 }alulist[Size];
    int result;                 int dis;
}sample1[Size];             }list1[Size],temp;
```

DFG1
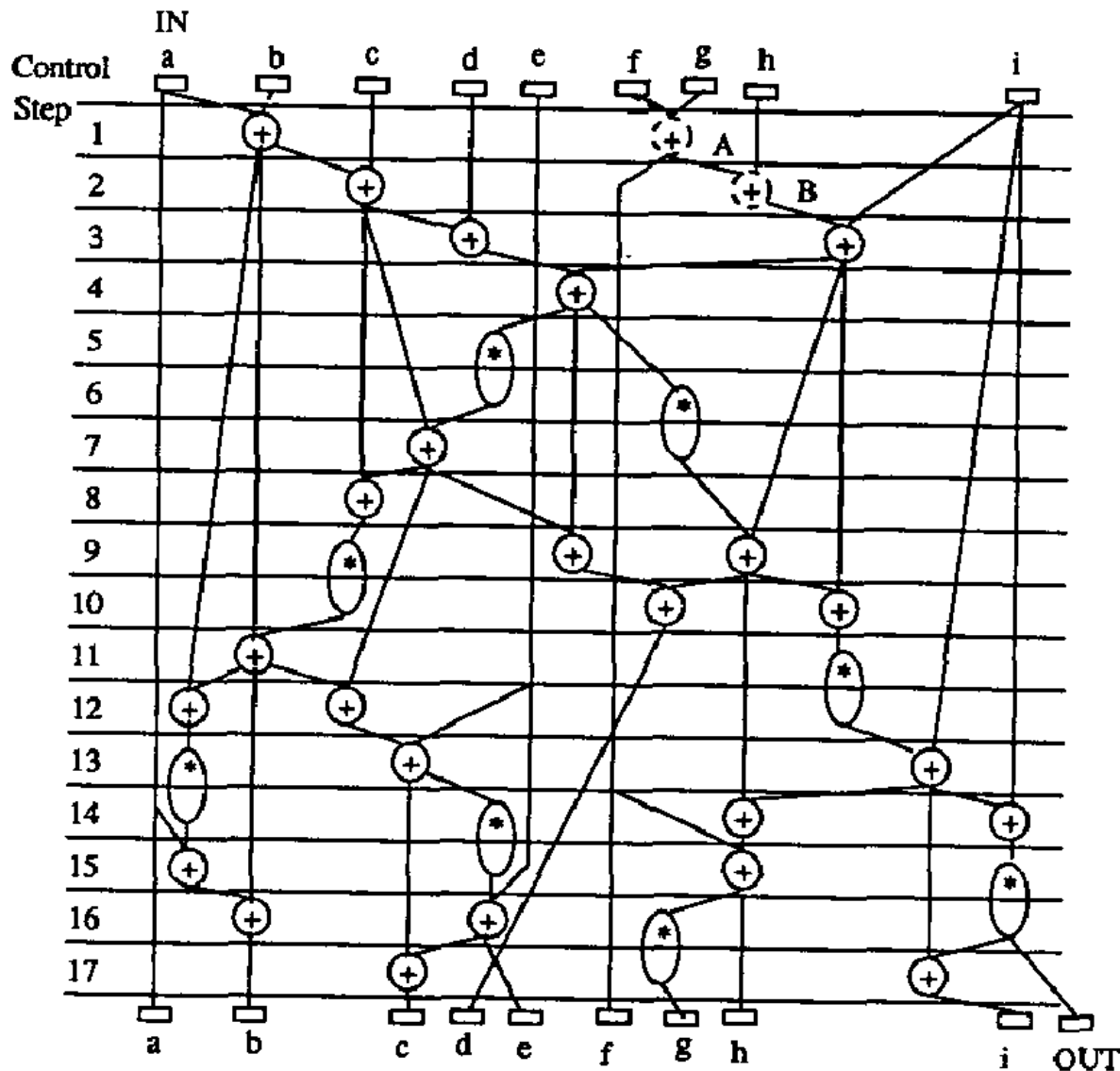
Resource constraints:

*	+
1	1
1	2
2	1
2	2

DFG2
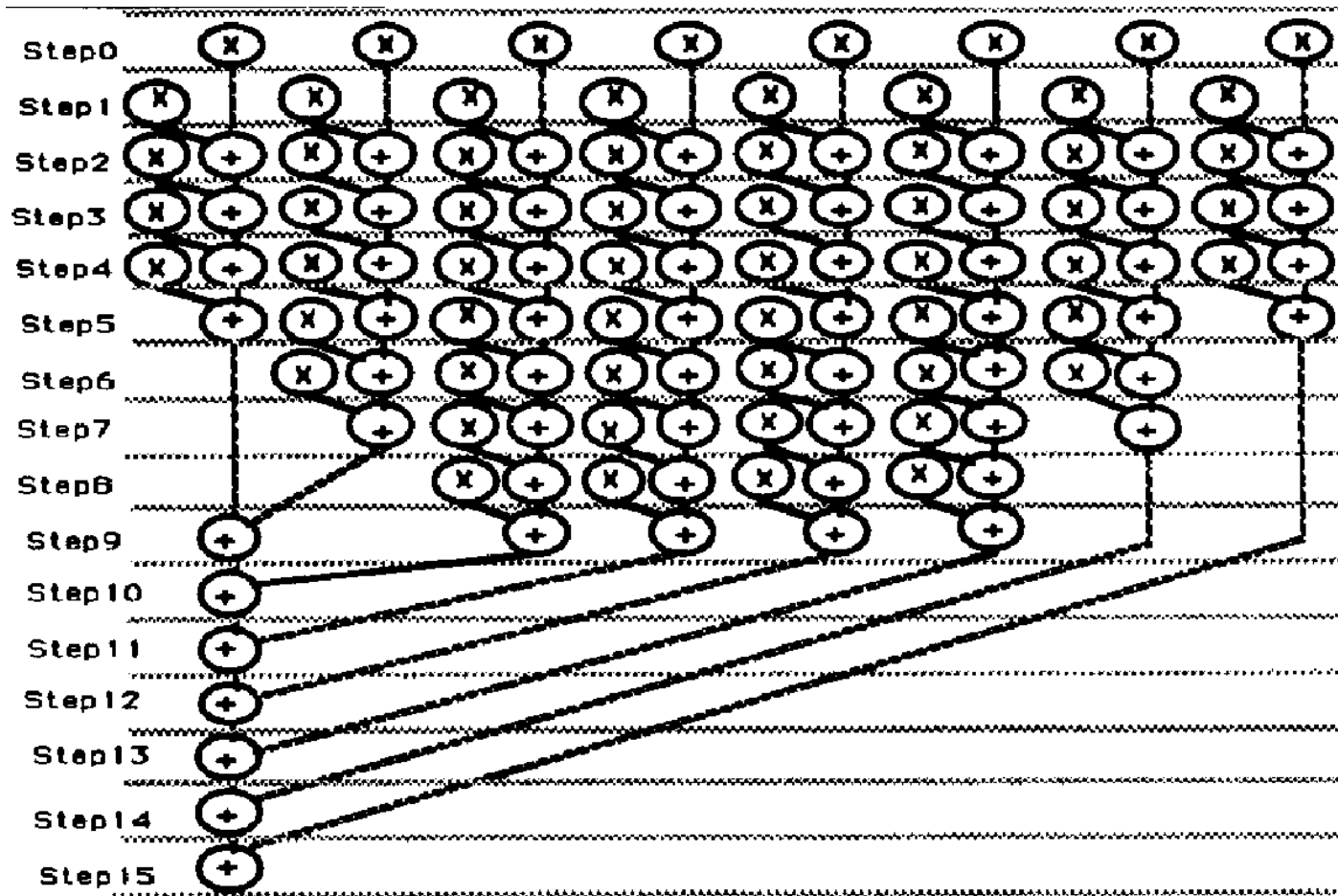
Resource constraints:

*	+
1	1
1	2
2	1
2	2
3	1
3	2
1	3
2	3
3	3

RGB to YUV

Resource constraints:

*	+
1	1
1	2
1	3
2	1
2	2
2	3
3	1
3	2
3	3

R 0.299 G 0.587 B 0.114

*1   *2   *3

+4

+5

Y

R 0.169 G 0.331 B 0.5

*6   *7   *8

128

+9   +10

−11

U

R 0.5 G 0.419 B 0.081

*12   *13   *14

128

−15   −16

+17

V
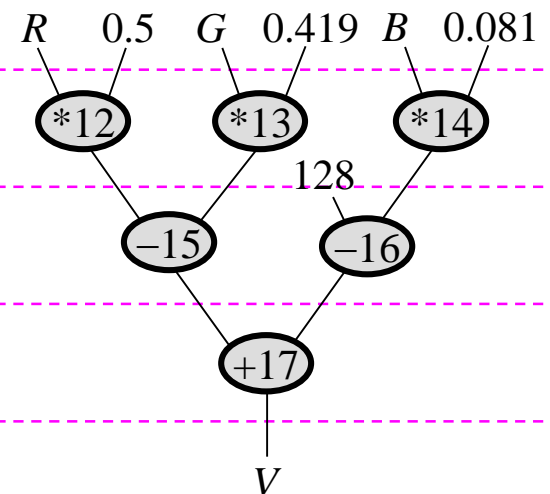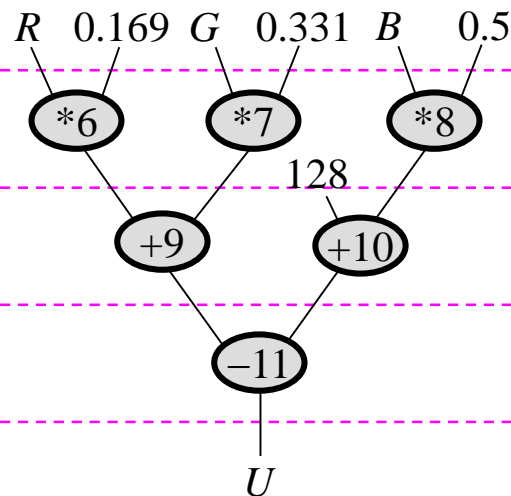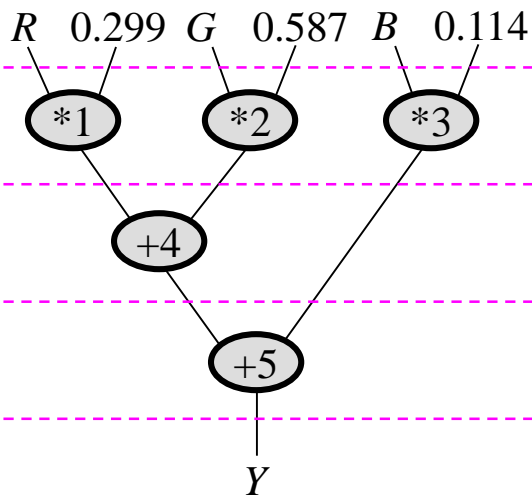
實驗報告+程式碼

■ **實驗報告**及**程式碼**以壓縮檔繳交，每位同學均須繳交

■ 實驗報告壓縮檔請以實驗編號及自己的學號姓名命名，例如：**Lab2_M999999999**陳小華**.rar**，於規定時間內上傳至"中山大學網路大學-作業評量區"繳交

■ 實驗報告內容包含

◆ 實驗主題、實驗日期、學號姓名

◆ 實驗內容、過程及結果

⊕ 實驗內容、程式簡要說明…

⊕ 實驗畫面、DFG 、 Scheduling 結果…

⊕ 實驗結果及分析

◆ 實驗心得