

## EE6094 CAD for VLSI Design

### Programming Assignment 2 (Due: 23:59:59, 2021/03/30)

You are asked to implement a resource-constrained scheduler to minimize the latency. Assume there are only two types of operations: **addition** and **multiplication**. An addition takes 1 time unit, and a multiplication takes 3 time units.

**Input file format:** The first four lines list the number of available adders, the number of available multipliers, total number of nodes, and total number of edges. The next  $n$  ( $n$  = total number of nodes) lines list the node information. Each of these  $n$  lines contains a unique node number followed by a symbol. There are 4 kinds of symbol: **i** represents an input node, **o** represents an output node, **+** represents an addition node, and **\*** represents a multiplication node. Note that the order of node number is **NOT** guaranteed. After that, there are  $m$  ( $m$ =number of edges) lines listing the edge information. Each of these  $m$  lines contains a node number **u** followed by another node number **v** which represents a directed edge from **u** to **v**.

**Output file format:** The first line gives the latency of your solution. The  $(i+1)$ th line lists the node(s) executed during the  $i$ th time unit in the **increasing order w.r.t the node number**. The output format is strict. The output file of your program should be named as *name.out* where *name* is the input file name.

**Algorithm:** You can use any algorithm to find best solution (minimum latency). You should be able to explain the algorithm you applied. When the condition is tight, always take the node with minimum node number.

**Requirement:** You have to write this program in C or C++. You also need to write a **makefile** which can compile and execute your program directly. I will verify your program on our workstation. You should also write a report which should at least include a description of the data structure and algorithm you used and how to execute your program. Name your file as PA2\_studID.cpp. Upload your code and report to ee-class. Some test input files will be announced on the ee-class soon.

**Score:** Your assignment will be ranked and scored according to (1) the quality of your solution, (2) the runtime of your program, (3) the readability of your source code, (4) the report you wrote, and (5) the demo session.

**Example:** Below is a sample input file and a sample output file with comments added. Fig. 1 is the corresponding DFG and Fig. 2 is the result of scheduling.

Sample input file:

```
1      // number of adders
1      // number of multipliers
13     // total number of nodes
17     // total number of edges
```

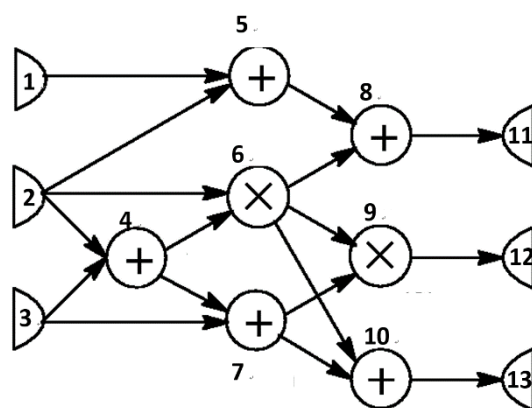
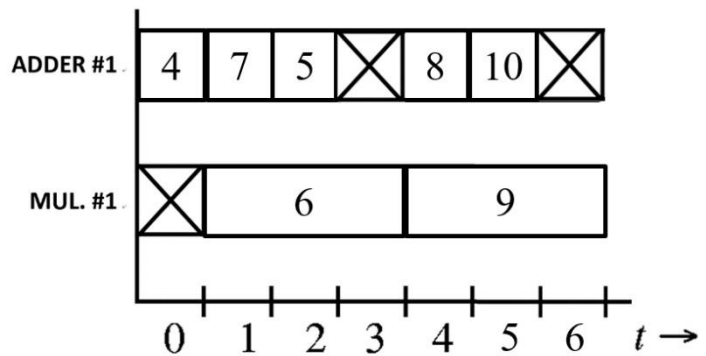


Fig. 1

```

1 i      // node 1~3 are inputs
2 i
3 i
4 +      // node 4, 5 are Add operators
5 +
6 *      // node 6 is a Multiply operator
7 +
8 +
9 *
10 +
11 o     // node 11~13 are outputs
12 o
13 o
1 5      // an edge from node 1 to node 5
2 5      // an edge from node 2 to node 5
2 4
2 6
3 4
3 7
4 6
4 7
5 8
6 8
6 9
6 10
7 9
7 10
8 11
9 12
10 13

```



Sample output:

```

7      // the latency of your solution (the beginning time of end NOP)
4      // execute the operation of node 4 at t0
6 7
6 5
6
9 8
9 10

```

