



















```
Code
   108
109
                                                     update resource
                                            // update resource
if(resource[i] < zero_slack.size() && i == 0)
    resource[i] = zero_slack.size();
else if(i == 1)(
    if(resource[i] < zero_slack.size() + on_going_mul[level - 1].size())
    resource[i] = zero_slack.size() + on_going_mul[level - 1].size();
}</pre>
    110
   111
112
113
    114
   115
116
117
                                            //
// schedule additional operations under resource constrain
if(resource[i] > zero_slack.size()) {
   int unused_resource = 0;
}
    118
                                                       if(i == 0)
   119
120
121
                                                       unused_resource = resource[i] - zero_slack.size();
else if(i == 1 % resource[i] < zero_slack.size() + on_going_mul[level - 1].size())
break;</pre>
                                                       else
    122
                                                      unused_resource = resource[i] - zero_slack.size() - on_going_mul[level].size();

// get the operations with slack > 0

-/for(int slack = 1; slack <= max_latency - level && unused_resource > 0; slack++){

for(int slack = max_latency - level; slack >= 1 && unused_resource > 0; slack--){
   123
124
125
126
   127
128
129
                                                                 if (unused_resource == 0)
break;
                                                                 break;
vector(int) additional_operations = find_slack(slack, level, nodes, operations[i], unused_resource);
for(int j = 0; j < additional_operations.size() 66 unused_resource > 0; j++){
    nodes[additional_operations[j]].schedule(level);
    result[level - 1].push_back(nodes[additional_operations[j]].Index());
    if(i == 1){
   130
131
132
133
134
135
136
137
138
139
140
141
                                                                                      result[level + 0].push_back(nodes[additional_operations[j]].Index());
result[level + 1].push_back(nodes[additional_operations[j]].Index());
on_going_mul[level + 0].push_back(nodes[additional_operations[j]].Index());
on_going_mul[level + 1].push_back(nodes[additional_operations[j]].Index());
                                                                            unused_resource--;
scheduled_cnt++;
141
142
143
144
145
146
147
                       }while(scheduled cnt < nodes.size() && level <= max latency);</pre>
```