Heuristic Scheduling: List Scheduling for MRLC

Dr. Chandan Karfa

Department of Computer Science and Engineering

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

List Scheduling Algorithms - Recap

- Algorithm 1: Minimize latency under resource constraint (ML-RC)
  - Resource constraint represented by vector ***a*** (indexed by resource type)
    - Example: two types of resources, MULT (a_1=1), ADD (a_2=2)

- Algorithm 2: Minimize resources under latency constraint (MR-LC)
  - Latency constraint is given and resource constraint vector ***a*** to be minimized

Define:

- The candidate operations $U_{l,k}$
  - those operations of type k whose predecessors have already been scheduled early enough (completed at step l):

    $U_{l,k} = \{ v_i \subseteq V: type(v_i) = k$ and $t_j + d_j \leq l$, for all j: $(v_j, v_i) \subseteq E$
- The unfinished operations $T_{l,k}$
  - those operations of type k that started at earlier cycles but whose execution has not finished at step l *(multi-cycle opearations):*

    $T_{l,k} = \{ v_i \subseteq V: type(v_i) = k$ and $t_i + d_i > l \}$
- Priority list
  - List operators according to some heuristic urgency measure
  - Common priority list: labeled by position on the longest path in decreasing order

Recap -- List Scheduling Algorithm 1: ML-RC

Minimize latency under resource constraint

LIST_L (G(V,E), **a**) { // resource constraints specified by vector **a**

l = 1

repeat {

for each resource type k {

$U_{l,k}$ = candidate operations available in step l

$T_{l,k}$ = unfinished operations (in progress)

Select $S_k \subseteq U_{l,k}$ such that $|S_k| + |T_{l,k}| \leq a_k$

Schedule the S_k operations at step l

}

$l = l + 1$

} until v_n is scheduled

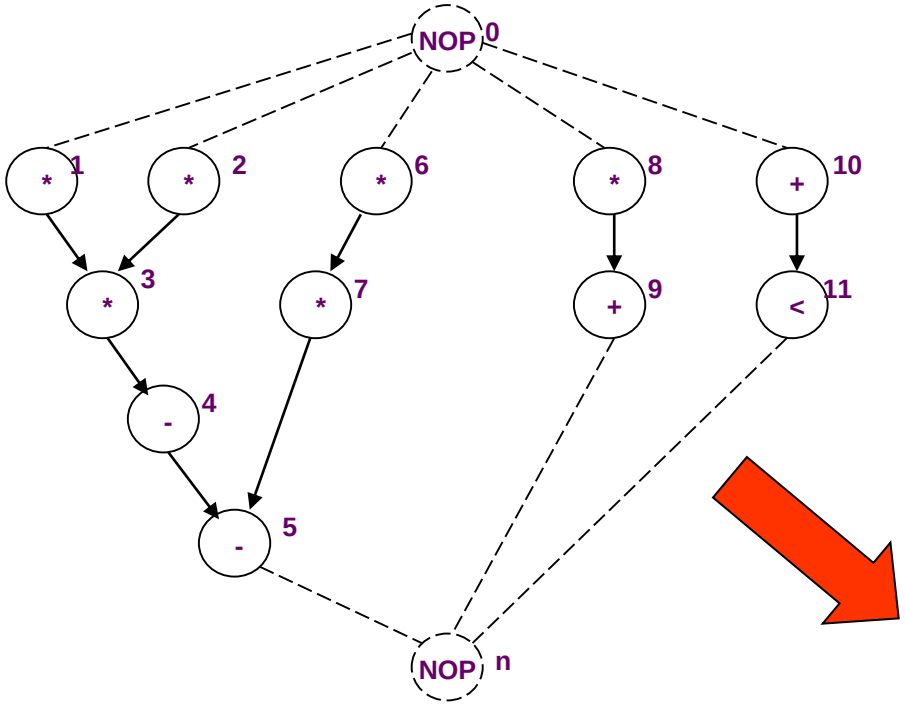
}

Note: If for all operators *i*, $d_i = 1$ (unit delay), the set $T_{l,k}$ is empty

List scheduling algorithm for MRLC

```
LIST_R( G(V, E), λ) {
    a = 1;
    Compute the latest possible start times t^L by ALAP ( G(V, E), λ);
    if (t_0 < 0)_L
        return (Ø);
    l = 1;
    repeat {
            for each resource type k = 1, 2, …, n_res {
                Determine ready operations U_{l,k};
                T_{l,k} = unfinished operations (in progress)
                Compute the slacks { s_i = t_i – l     for all v_i ϵ U_{lk}};
                Select S_k ⊆ U_{l,k} with zero slack and update a s.t. |S_k| + |T_{l,k}| = a_k;
                Schedule the candidate operations in U_{l,k} not needing additional resources;
            }
            l = l + 1;
    } until (v_n is scheduled) ;
    return (t, a);
}
```

Example

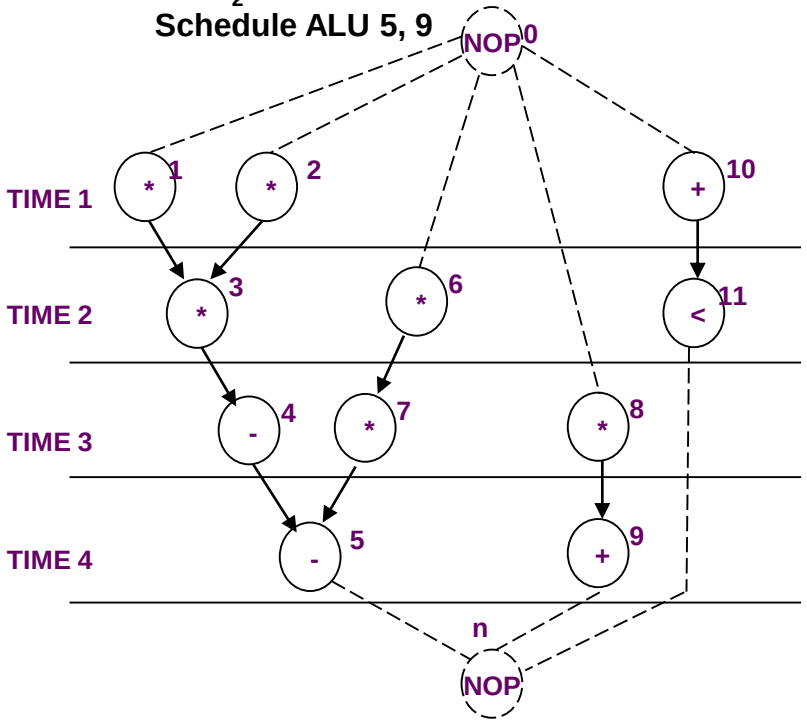


Step 1
Two multiplications on CP
Set a_1 = 2
Schedule Mult 1,2
Schedule ALU 10

Step 2
Schedule Mult 3, 6
Schedule ALU 11

Step 3
Schedule Mult 7,8
Schedule ALU 4

Step 4
Set a_2=2
Schedule ALU 5, 9



Assumptions

Unit-delay resources

Maximum latency = 4

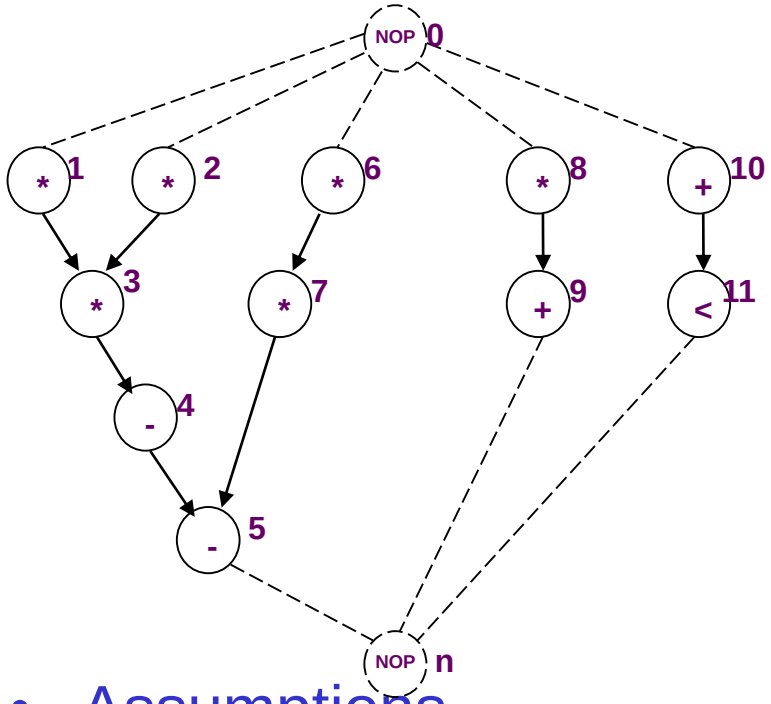
Start with :

a_1 = 1 multiplier

a_2 = 1 ALUs

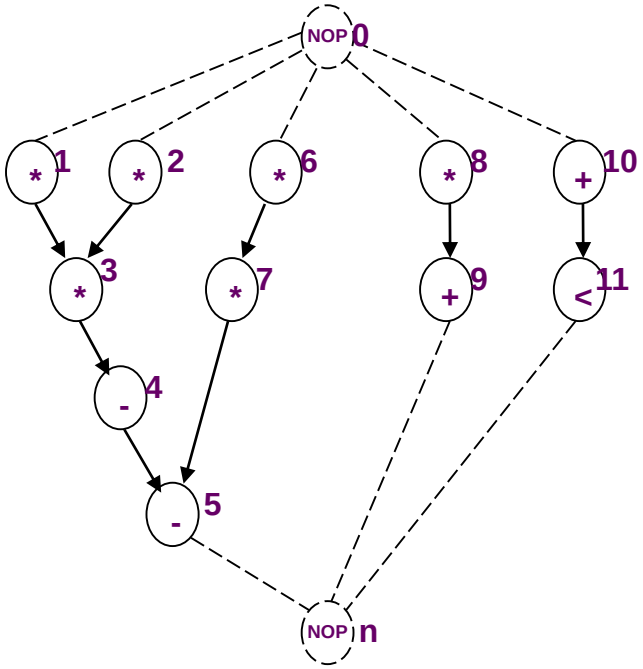
List Scheduling MR-LC – Example 2a

MRLC (with $d_1=2$, $d_2=1$)



- Assumptions
  - Operations have different delay:
    $del_{MULT} = 2$, $del_{ALU} = 1$
- *Latency L = 6*

List Scheduling MR-LC – Example 2b (Pipelined)



- Assumptions
  - Multipliers are 3-stage pipelined
- *Latency=7*

Thank You