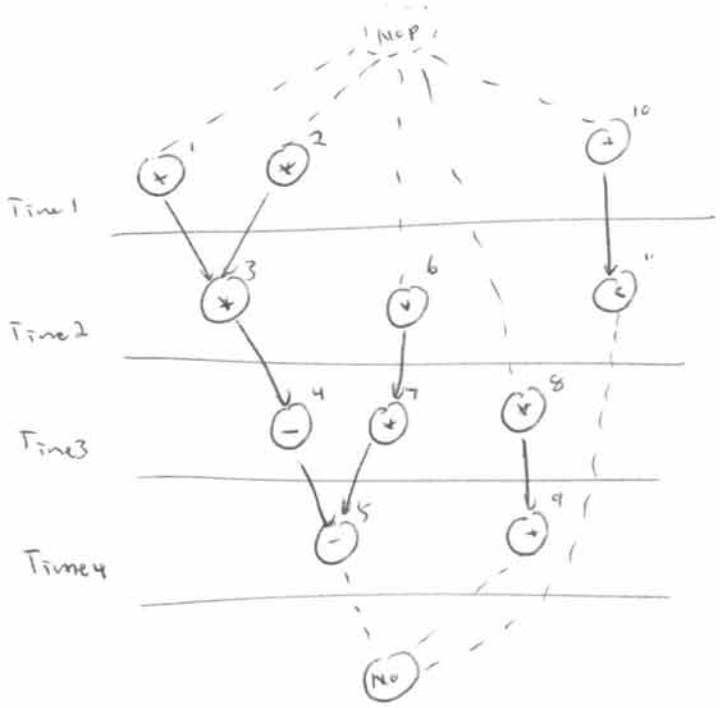


Conflict Graph for each Resource:

multipliers:

ALUs:

Interval Graphs for scheduled Sequencing Graph:

Intervals $I = \{ [l_j, r_j] ;\ j = 1, 2, \ldots, |I| \}$ l_j = left edge

r_j = right edge

Given ~~conflict~~ scheduled sequencing graph, create a set of intervals for each resource type:

for each v_i, create interval with l_j = start time of v_i and

r_j = end time of v_i

Multipliers:

[1,2], [1,2], [2,3], [2,3], [3,4], [3,4]

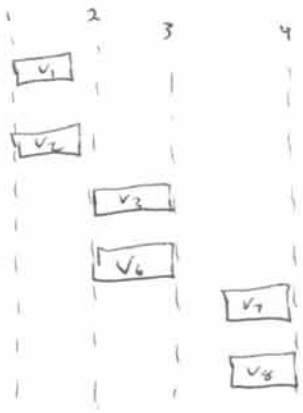
v_1 v_2 v_3 v_4 v_5 v_6

↳ captures same information as conflict graph, and supports multicycle latencies

LEFT-EDGE algorithm can optimally find vertex coloring for intervals

```
LEFT-EDGE(I) {
    Sort elements of I in a list L in ascending order of l_i
    c = 0
    while (some interval has not been scheduled) {
        S = ∅
        r = 0        // initialize coordinate of rightmost edge in S
        while (∃ an element in L whose left edge l_i ≥ r) {
            s = first element in L with l_s ≥ r
            S = S ∪ {s}
            r = r_s
            remove s from L
        }
        c = c + 1
        label elements of S with color c
}
```

Sorted Intervals for Multiplexes:



LEFT-EDGE:

$L = v_1, v_2, v_3, v_6, v_7, v_8$

$c = 0$

$S = \{\}$

$r = 0$

1 $S = v_1$

$S = \{v_1\}$

$r = 1$

2 $S = v_3$

$S = \{v_1, v_3\}$

$r = 3$

3 $S = v_7$

$S = \{v_1, v_3, v_7\}$

$r = 4$

$c = 1$

$c_1 = 1, c_3 = 1, c_7 = 1$

$L = \{v_2, v_6, v_8\}$

$S = \{\}$

$r = 0$

1 $S = v_2$

$S = \{v_2\}$

$r = 2$

2 $S = v_6$

$S = \{v_2, v_6\}$

$r = 3$

3 $S = v_8$

$S = \{v_2, v_6, v_8\}$

$r = 4$

$c = 2$

$c_2 = 2, c_6 = 2, c_8 = 2$

$L = \{\}$

Intervals for ALUs:

$[1,2]$, $[2,3]$, $[3,4]$, $[4,5]$, $[4,5]$

v_{10} v_{11} v_{4} v_{5} v_{6}

Sorted Intervals for ALUs:

LEFT-EDGE:

$L = v_{10}, v_{11}, v_4, v_5, v_6$

$c = 0$

$S = \{\}$
$r = 0$

1. $s = v_{10}$
   $S = \{v_{10}\}$
   $r = 2$

2. $s = v_{11}$
   $S = \{v_{10}, v_{11}\}$
   $r = 3$

3. $s = v_4$
   $S = \{v_{10}, v_{11}, v_4\}$
   $r = 4$

4. $s = v_5$
   $S = \{v_{10}, v_{11}, v_4, v_5\}$
   $r = 5$

$c = 1$
$c_{10} = 1, c_4 = 1, c_4 = 1, c_5 = 1$
$L = \{v_4\}$

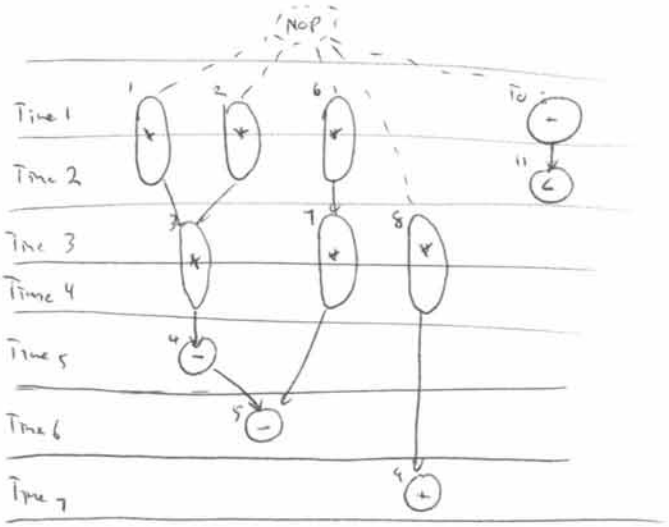
$S = \{\}$
$r = 0$

1. $s = v_4$
   $S = \{v_4\}$
   $r = 2$

$c = 2$
$c_4 = 2$
$L = \{\}$

Multicycle Latencies:

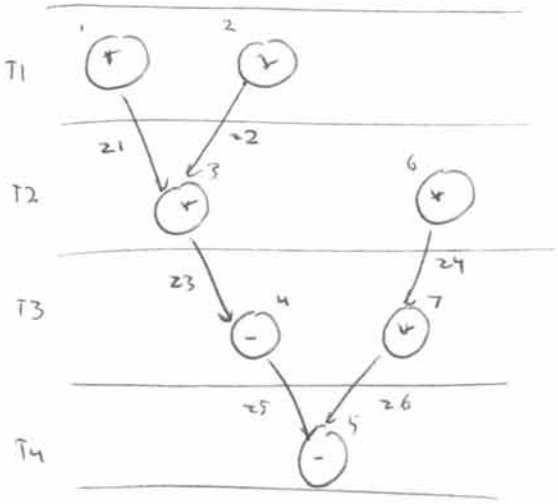
Multiplier: 2 cycles
ALU: 1 cycle



scheduled sequencing graph from LIST_L

Register Sharing:

- Edges within scheduled sequencing graph are variables that must be stored within registers
- Registers can be shared, just like resources
- Each variable has a lifetime that is an interval from its birth to its death, where the birth is the time a value is generated and death is the last time that variable is used



Intervals: $z_1 = [1,2]$
$z_2 = [1,2]$
$z_3 = [2,3]$
$z_4 = [2,3]$
$z_5 = [3,4]$
$z_6 = [3,4]$

- Use vertex coloring (LEFT-EDGE) to determine register binding

Sorted Intervals

coloring

$C_{z_1} = 1$, $C_{z_3} = 1$, $C_{z_5} = 1$

$C_{z_2} = 2$, $C_{z_4} = 2$, $C_{z_6} = 2$

Need 2 registers