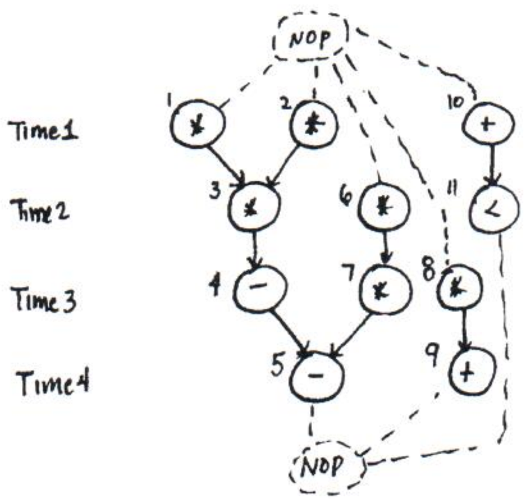
Interval Graph for scheduled sequencing graphs

Intervals $I = \{ [l_i, r_i], \quad i = 1, 2, \ldots, |I| \}$ $\qquad l_i$ = left edge, r_i = right edge

given scheduled sequencing graph
create a set of intervals for each resource type



Multipliers		ALUs	
V_1	[1,2]	V_4	[3,4]
V_2	[1,2]	V_5	[4,5]
V_3	[2,3]	V_9	[4,5]
V_6	[2,3]	V_{10}	[1,2]
V_7	[3,4]	V_{11}	[2,3]
V_8	[3,4]		

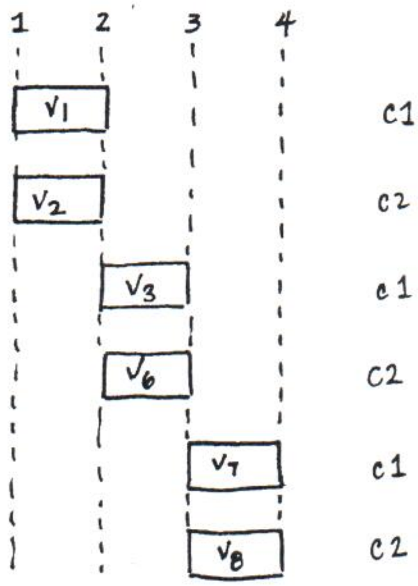
intervals capture same information as conflict graph and supports multicycle latencies.

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 colored ) do {
        S = Ø;
        r = 0;                          /* initialize coordinate of rightmost edge in S */
        while ( ∃ an element in L whose left edge coordinate is larger than r) do{
            s = First element in the list L with l_s ≥ r;
            S = S ∪ {s};
            r = r_s;                    /* update coordinate of rightmost edge in S */
            Delete s from L;
        }
        c = c + 1;
        Label elements of S with color c;
    }
}
```

ALGORITHM 2.4.7

try our multiplier example.



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

sort in order of l_i

$c = 0$

$s = \{\ \}$
$r = 0$

first element in L with $l_i \geq 0$
$s = v_1$
$S = \{\ \} \cup \{v_1\} = \{v_1\}$
$r = r_1 = 2$

delete v_1 from L

first element in L with $l_i \geq 2$
$s = v_3$
$S = \{v_1\} \cup \{v_3\} = \{v_1, v_3\}$
$r = r_3 = 3$

delete v_3 from L

first element in L with $l_i \geq 3$
$s = v_7$
$S = \{v_1, v_3\} \cup \{v_7\} = \{v_1, v_3, v_7\}$
$r = r_7 = 4$

delete v_7 from L

first element in L with $l_i \geq 4$
$s = \phi$ (exit while loop)
$c = c + 1 = 1$
Label s with color c

$L = v_2, v_6, v_8$
$S = \{\ \}$
$r = 0$

$s = 2$
$S = \{\ \} \cup \{v_2\}$
$r = 2$

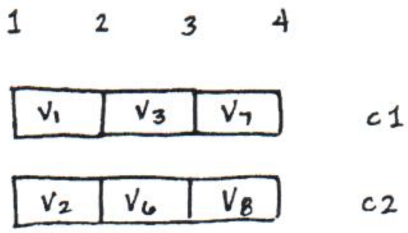
$s = 6$
$S = \{v_2\} \cup \{v_6\}$
$r = 3$

$s = 8$
$S = \{v_2, v_6\} \cup \{v_8\}$
$r = 4$

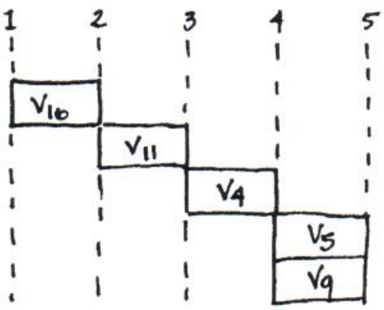
$s = \phi$ (no left edge coordinate ≥ 4)

$c = c + 1 = 2$
Label s with color c

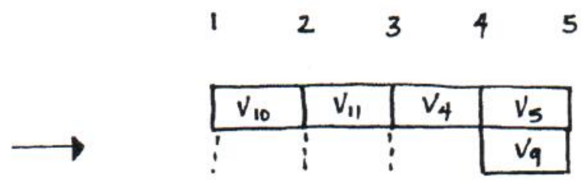
pack the graph based on colorings.
(packed graph)



Try left edge for ALL intervals



packed graph



$L = v_{10}, v_{11}, v_4, v_5, v_9$
$c = 0$

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

$s = 10$
$S = \{ \} \cup \{v_{10}\} = \{v_{10}\}$
$r = 2$

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

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

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

$c = 1$
$c_{10} = 1$, $c_{11} = 1$, $c_4 = 1$, $c_5 = 1$

$L = v_9$

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

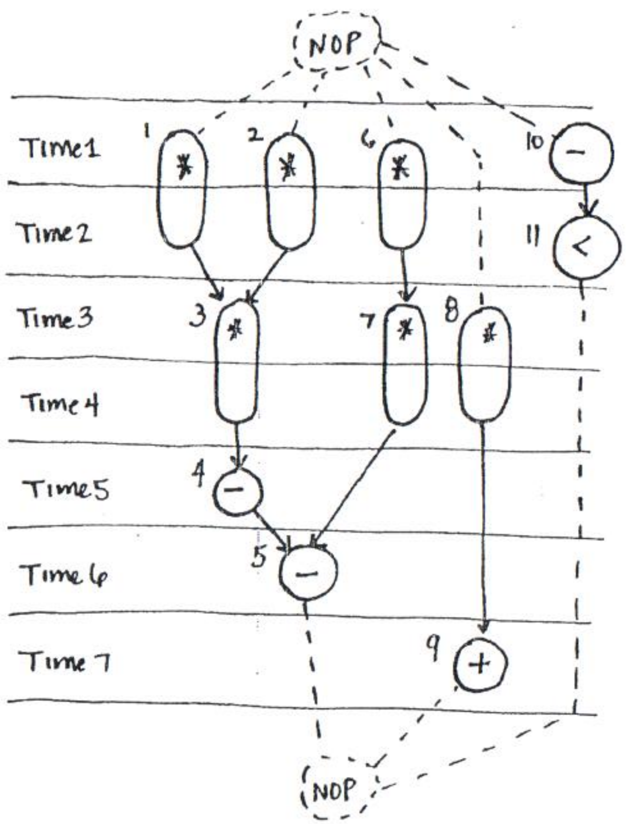
$s = 9$
$S = \{ \} \cup \{v_9\}$
$r = 5$

$c = 2$
$c_9 = 2$

$L = \{ \}$
nothing left to color so we can terminate

What about multicycle latencies?

multipliers = 2 cycles
alu = 1 cycle

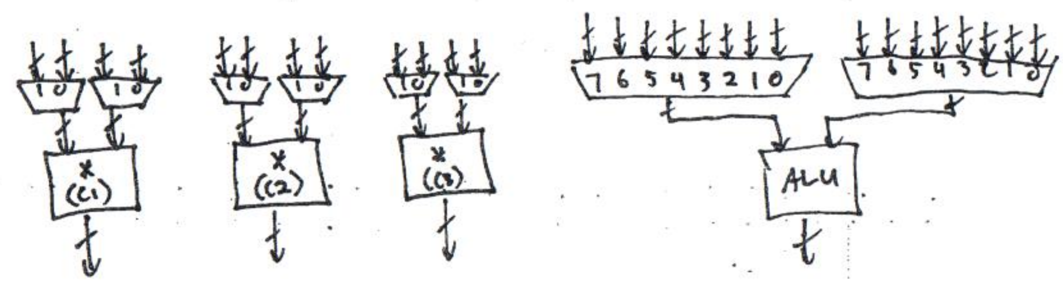


Bindings:

Multipliers:
c1 = {v_1, v_3}
c2 = {v_2, v_7}
c3 = {v_6, v_8}

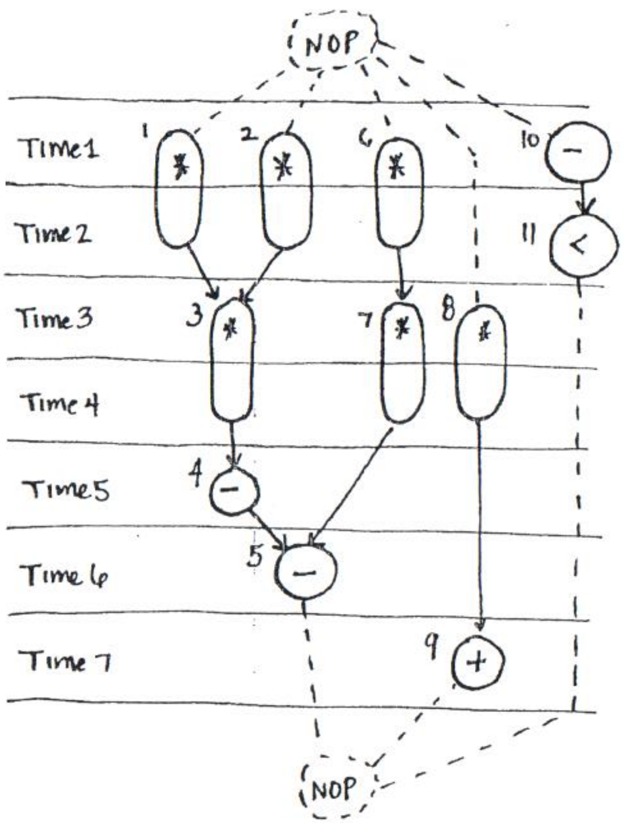
ALU:
c4 = {v_{10}, v_{11}, v_4, v_5, v_9}

Datapath (so far):



What about multicycle latencies?

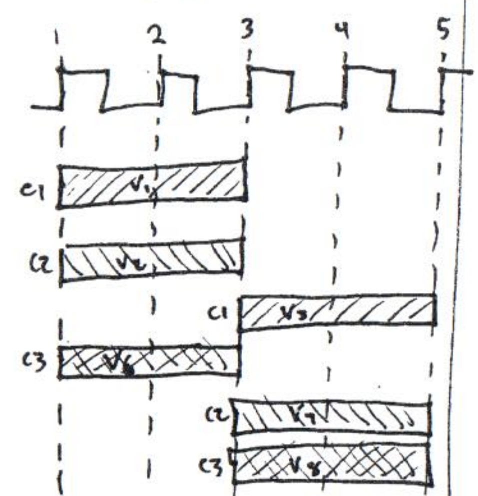
multipliers = 2 cycles
alu = 1 cycle



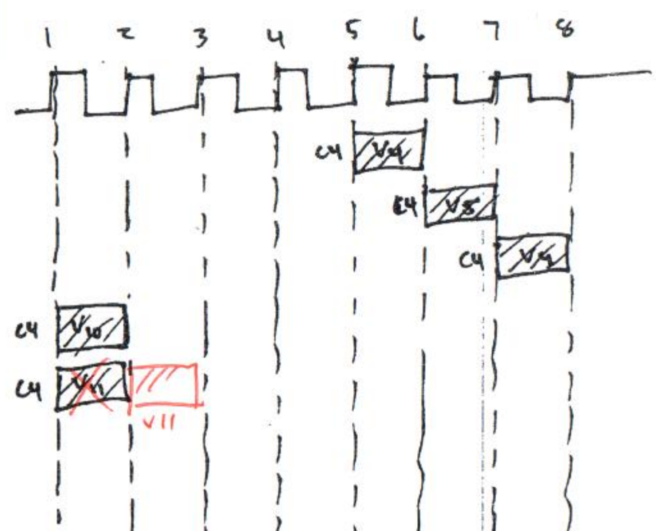
Interval Graph:

Multipliers
v_1 [1,3]
v_2 [1,3]
v_3 [3,5]
v_6 [1,3]
v_7 [3,5]
v_8 [3,5]

ALUs
v_4 [5,6]
v_5 [6,7]
v_9 [7,8]
v_{10} [1,2]
v_{11} [2,3]



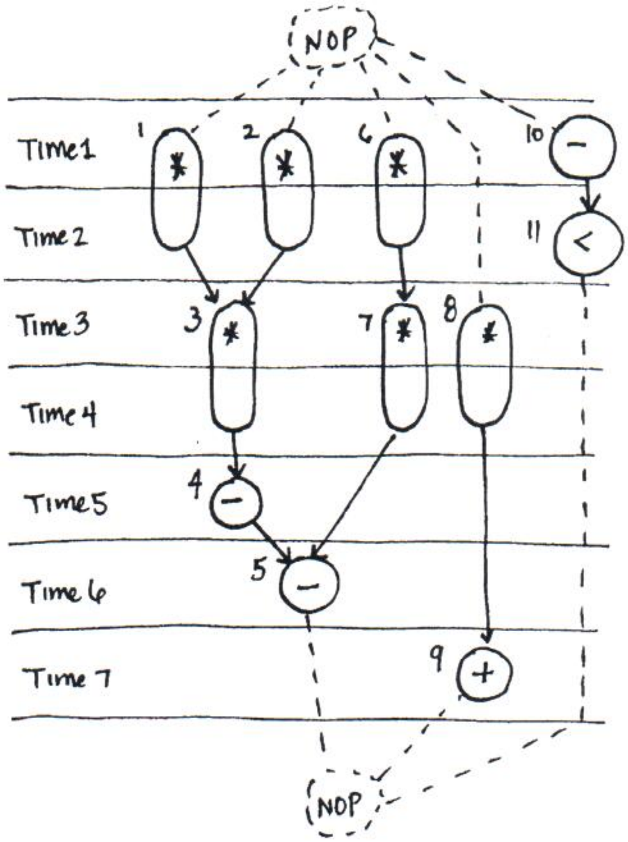
Bindings:
$c1 = \{v_1, v_3\}$
$c2 = \{v_2, v_7\}$
$c3 = \{v_6, v_8\}$



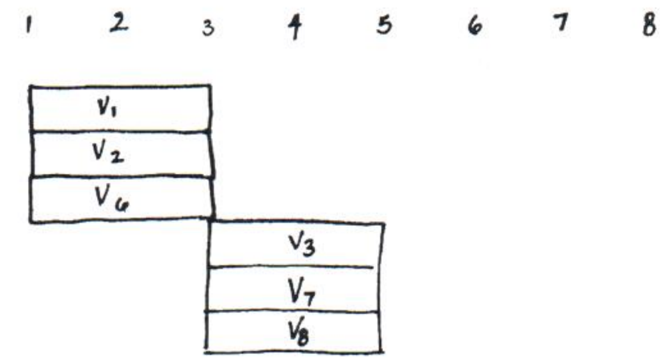
Bindings:
$c4 = \{v_{10}, v_{11}, v_4, v_5, v_9\}$

What about multicycle latencies?

multipliers = 2 cycles
alu = 1 cycle

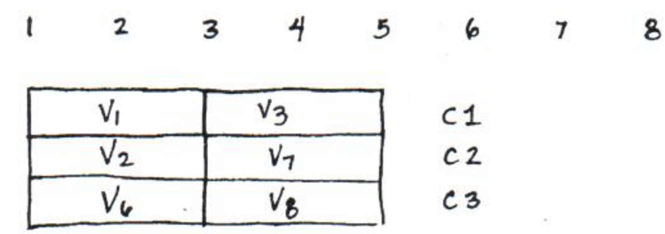


Multipliers

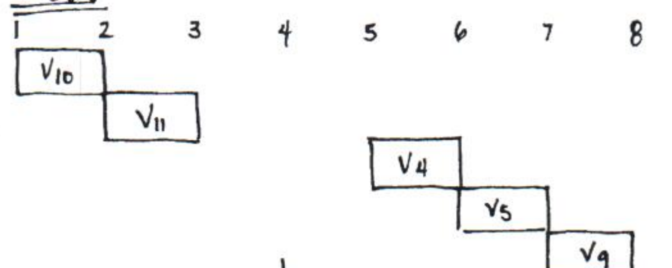


↓

packed graph

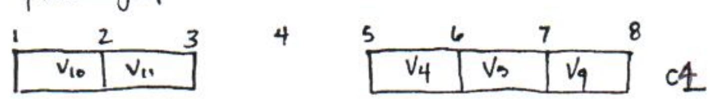


ALUs



↓

packed graph



Register Sharing

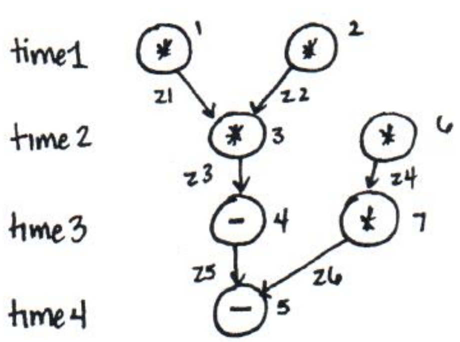
What about registers that hold values of the variables?

- 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 the interval from its birth to its death

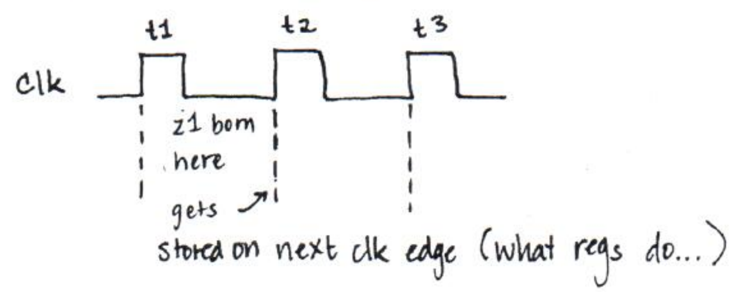
birth – time at which value is generated as an output of an operation

death – latest time at which variable is referenced as an input to another operation

(assume variables with multiple assignments within one model are aliased, so each variable has a single lifetime interval in the frame of reference corresponding to the sequencing graph entity where its used)



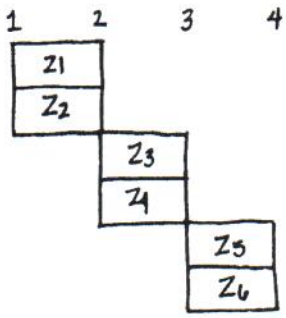
sequencing graph fragment



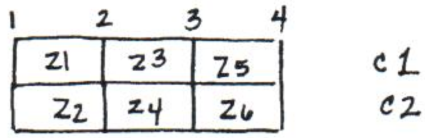
~~z_1 = [1, 2]~~	$z1 = [2, 3]$
~~z_2 = [1, 2]~~	$z2 = [2, 3]$
~~z_3 = [2, 3]~~	$z3 = [3, 4]$
~~z_4 = [2, 3]~~	$z4 = [3, 4]$
~~z_5 = [3, 4]~~	$z5 = [4, 5]$
~~z_6 = [3, 4]~~	$z6 = [4, 5]$

Graphs all move to the right by one time unit !

variable intervals



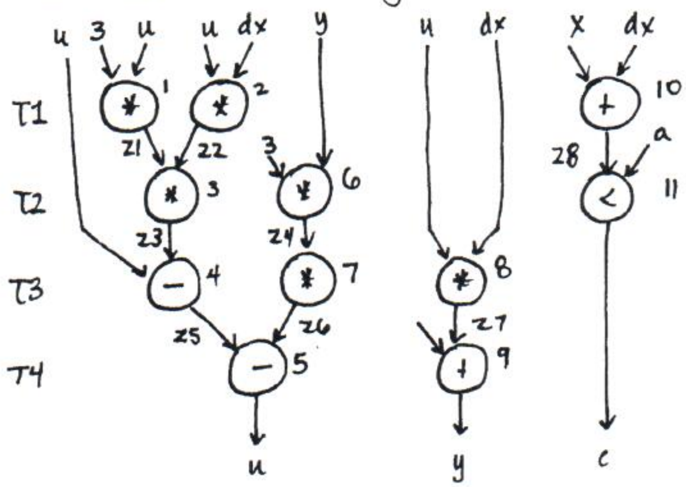
packed graph

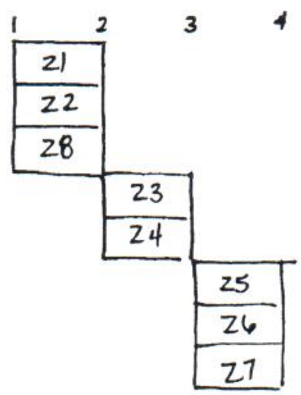


need 2 registers

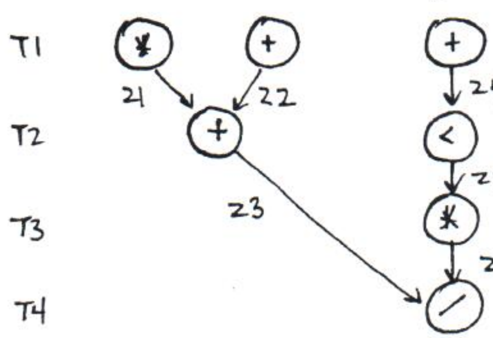
reg1 – z1, z3, z5
reg2 - z2, z4, z6

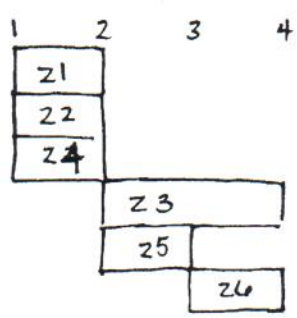
what about the following example?





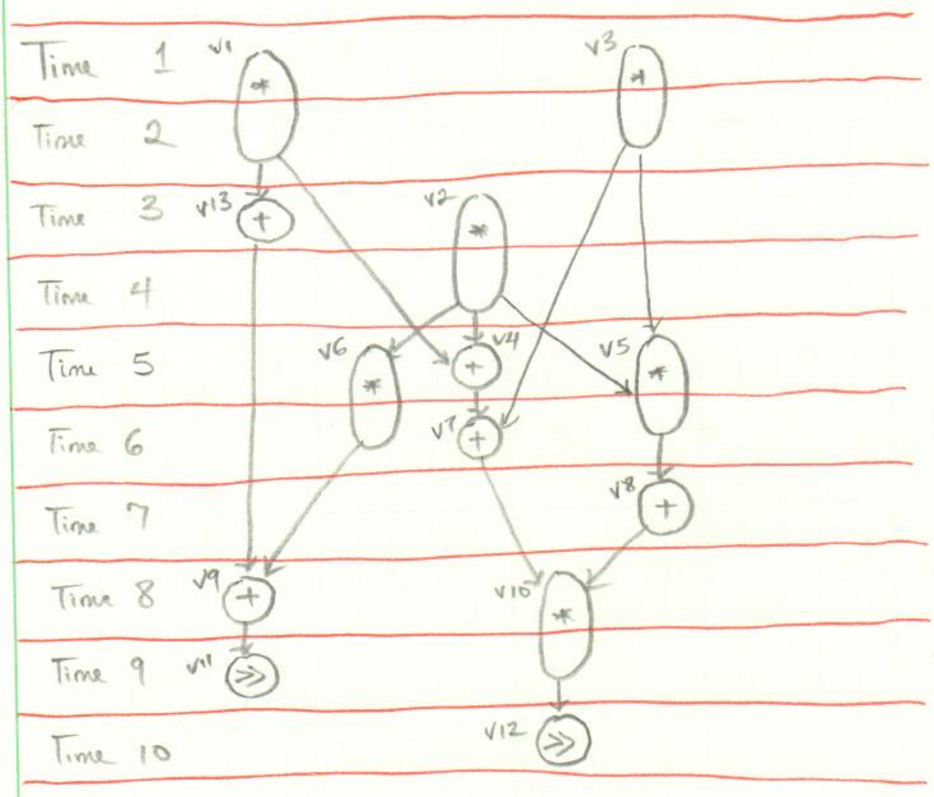
what about the following example?





you can apply left edge alg to interval graphs.

Use LeftEdge Algorithm to determine resource sharing and binding.



Determine intervals for the different vertices and vertex coloring.

Assume multipliers, adders, and shifters

Multipliers

v1 [1, 3]
v2 [3, 5]
v3 [1, 3]
v5 [5, 7]
v6 [5, 7]
v10 [8, 10]

Adders

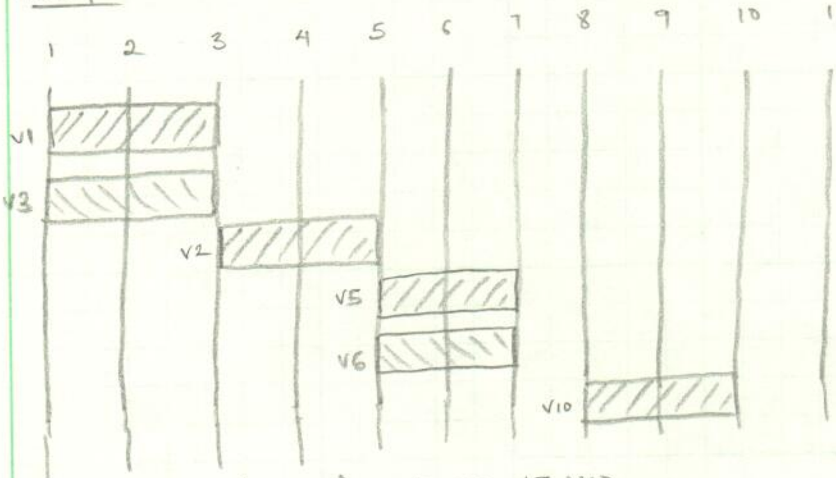
v4 [5, 6]
v7 [6, 7]
v8 [7, 8]
v9 [8, 9]
v13 [3, 4]

Logic/shift

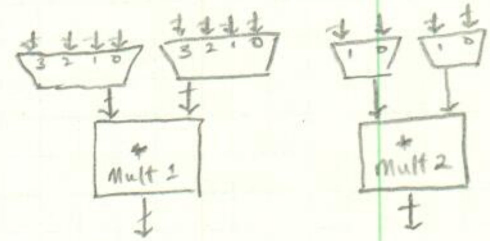
v11 [9, 10]
v12 [10, 11]

Interval graphs

Multipliers

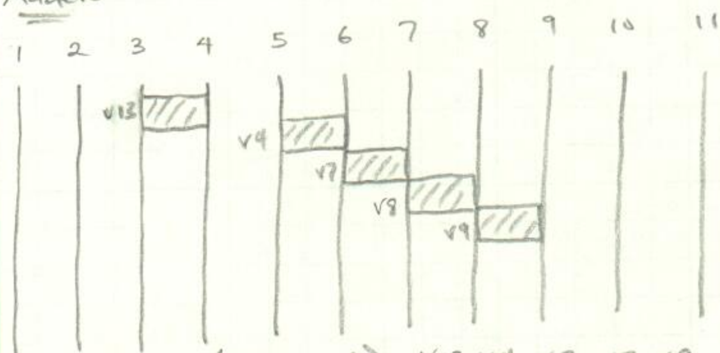


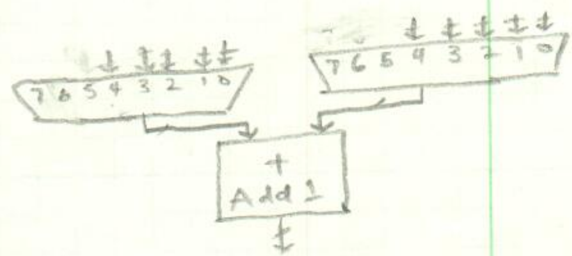
Datapath:



Color 1 (mult 1) : V1, V2, V5, V10
Color 2 (mult 2) : V3, V6

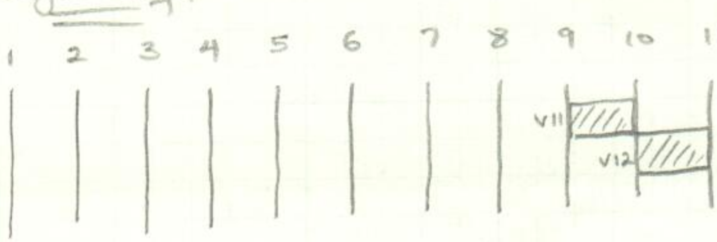
Adders

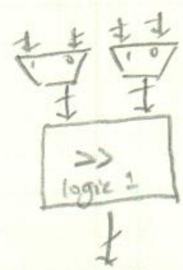




Color 1 (Adder 1) : V13, V4, V7, V8, V9

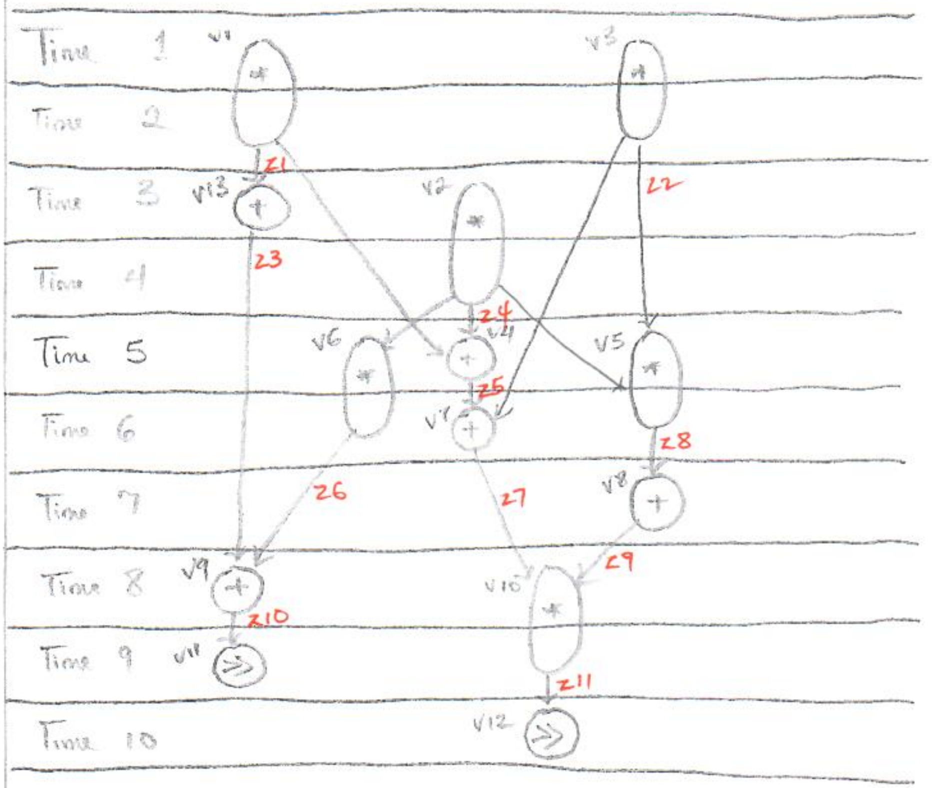
Logic shift





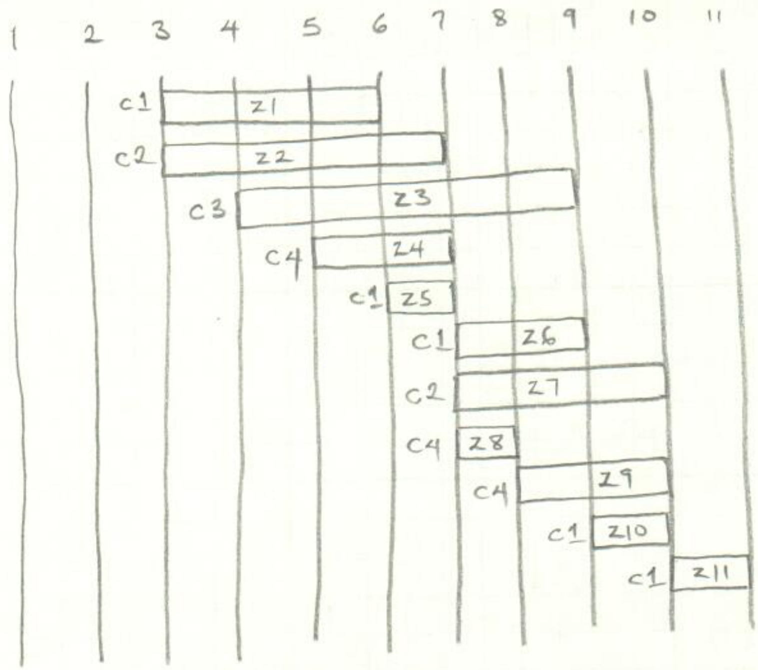
Color 1 (Logic/shift 1) : V11, V12

Use Left Edge Algorithm to determine ~~resource~~ sharing and register binding.



Determine intervals for the different vertices and vertex coloring.

Assume multipliers, adders, and shifters.



z1 [3, 6]
z2 [3, 7]
z3 [4, 9]
z4 [5, 7]
z5 [6, 7]
z6 [7, 9]
z7 [7, 10]
z8 [7, 8]
z9 [8, 10]
z10 [9, 10]
z11 [10, 11]

Color 1 (reg 1) : z1, z5, z6, z10, z11
Color 2 (reg 2) : z2, z7
Color 3 (reg 3) : z3,
Color 4 (reg 4) : z4, z8, z9

Need 4 registers.