Allocation

Chapter 8

Source: Gajski, Dutt, Wu, Lin

"High−Level Synthesis"

Kluwer Academic Publishers, 1992

Allocation

- **Selection** of components to be used in RT design
- **Binding** of hardware structures (units, regs, connections) to behavioral operators and variables
- **Define target DP architecture**

  Clocking, Busing,..
- **Approaches**

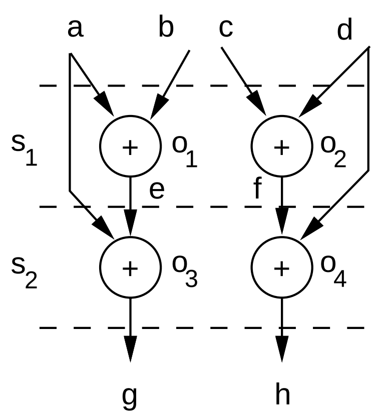
  Greedy

  Decomposition

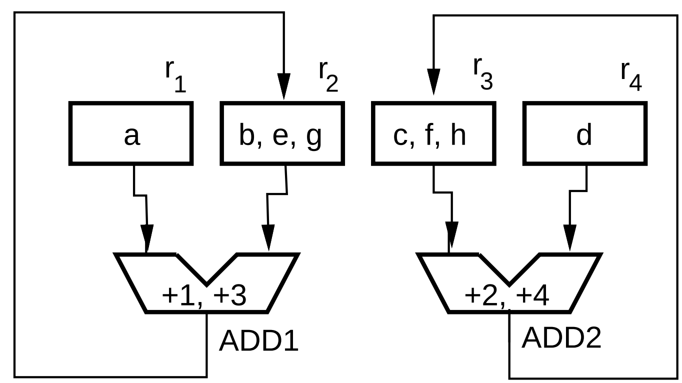
  Iterative

Unit Selection and Binding: Example

- *Unit Selection:* 2 adders, 4 registers
- *Mapping Behavior to RT Structure:*

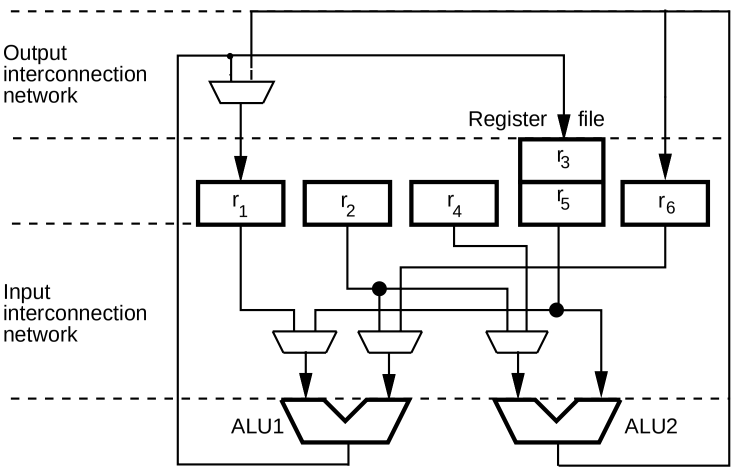


DFG

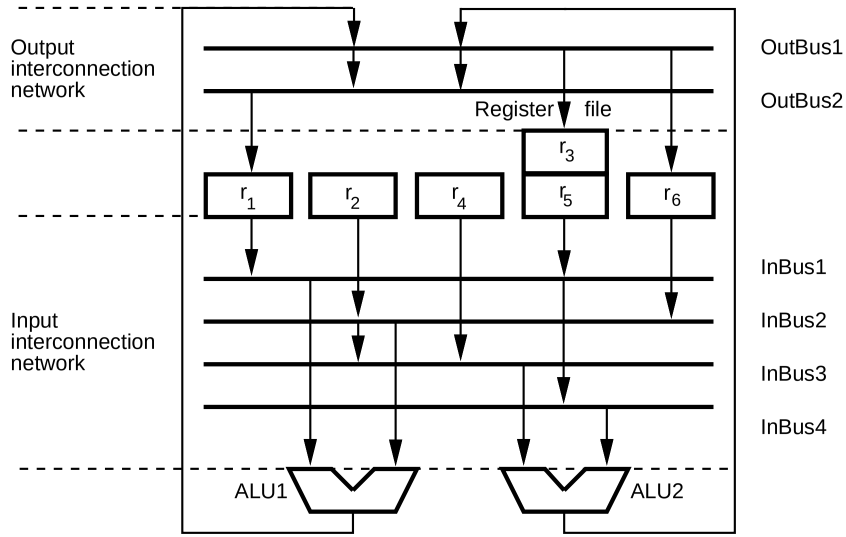


Allocated RT−Structure

Point−to−Point DP Interconnection Architectures



Mux−oriented DP

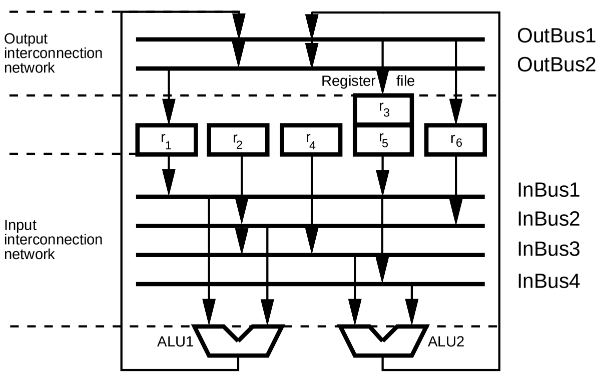


Bus−oriented DP

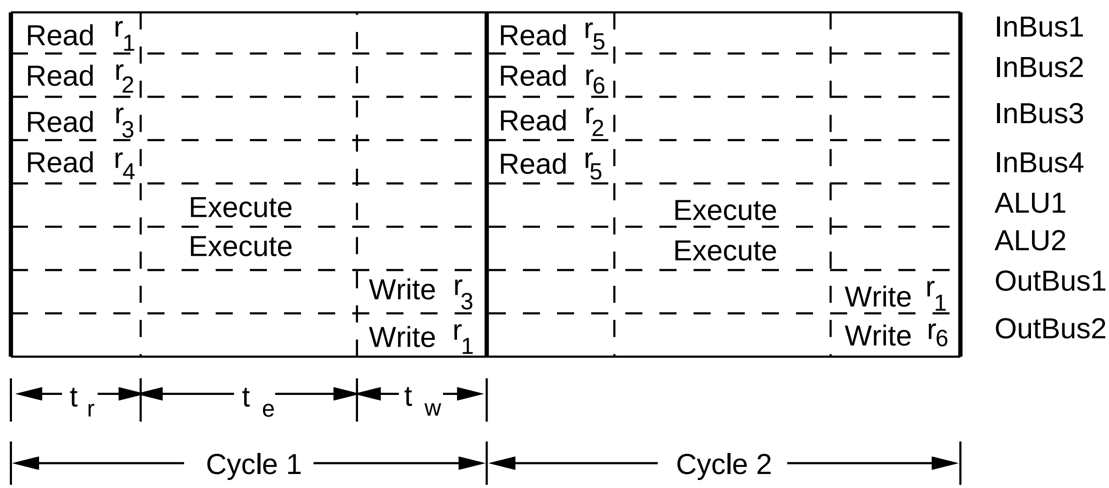
Register−transfers

s1: r3 <= ALU1(r1,r2); r1 <= ALU2(r3,r4);
s2: r1 <= ALU1(r5,r6); r6 <= ALU2(r2,r5);
s3: r3 <= ALU1(r1,r6)

One–Phase Clocking of Point–to–Point DP



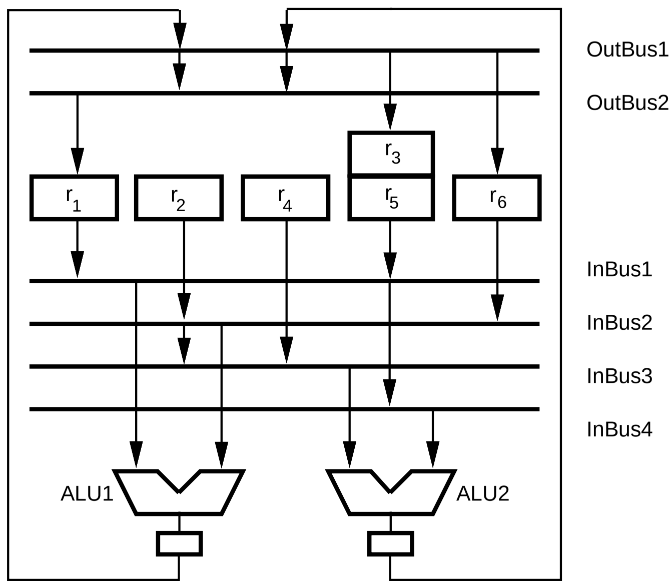
Bus–oriented DP

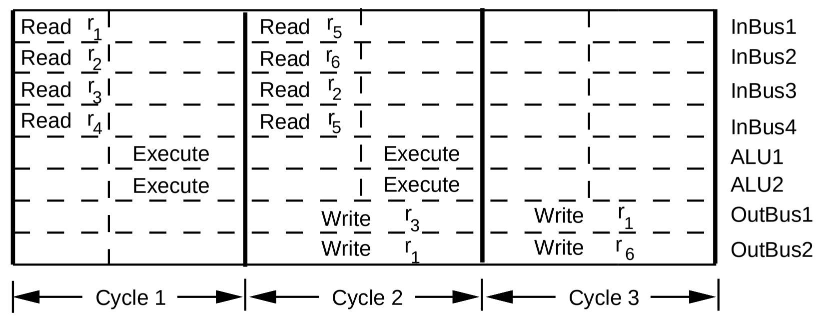


- *Requirement:* Cycle Time > tr + te + tw
- *Sequential execution*

UC Irvine

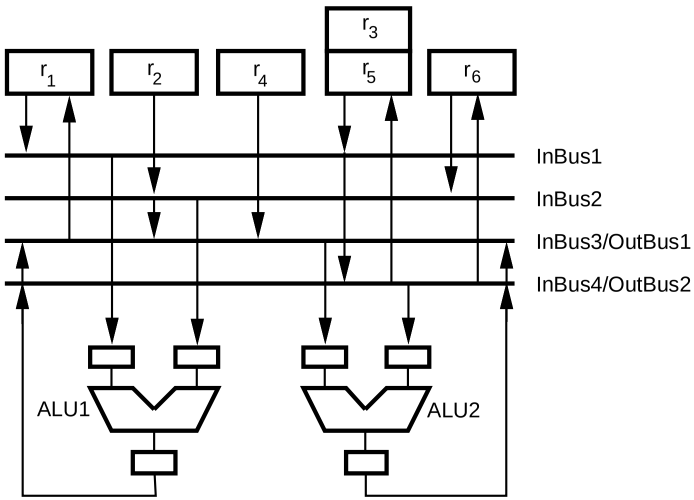
One–Phase Pipelined Datapath

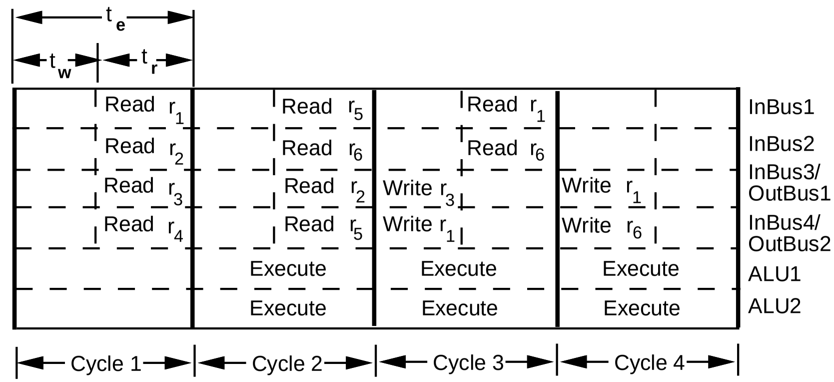




- Clock cycle = max (tr + te, tw)
- *Overlapping read and write data transfers*

Two–Phase Pipelined Datapath



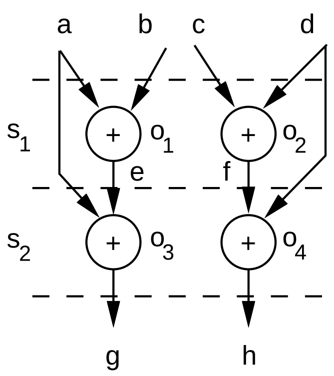


- Clock cycle = max (te, tr + tw)
- *Overlapping data transfers with FU execution*

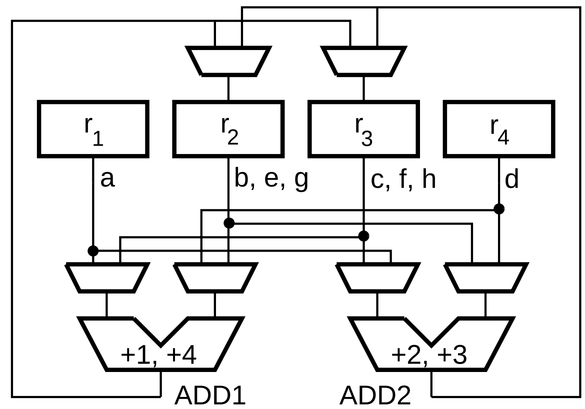
Allocation Tasks

- Unit Selection
- Functional–unit Binding
- Storage Binding
- Interconnection Binding

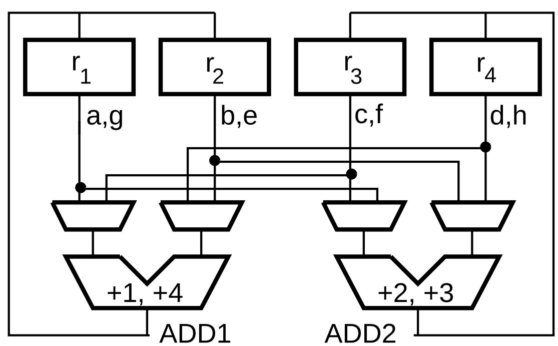
Interdependence and Ordering of Binding



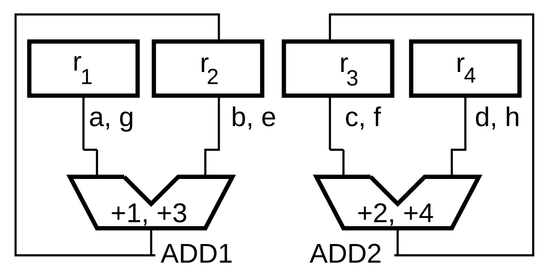
Scheduled DFG



FU Binding with 6 Muxes



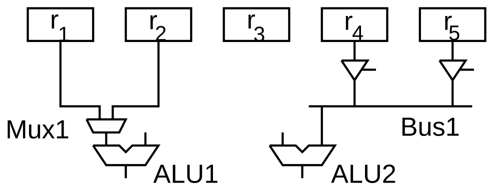
Register Reallocation: 4 Muxes



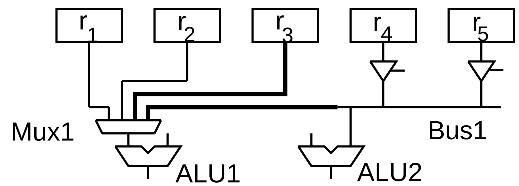
FU Rebinding: Optimal Design

UC Irvine

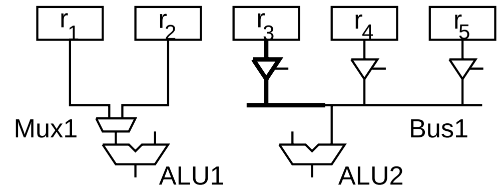
Greedy Binding: Examples



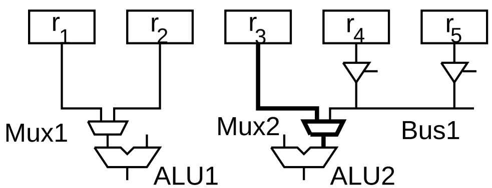
Initial partial design



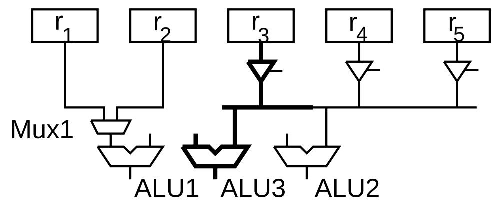
add 2 inputs to mux



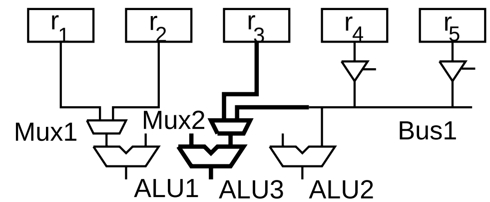
add tristate buffer to bus



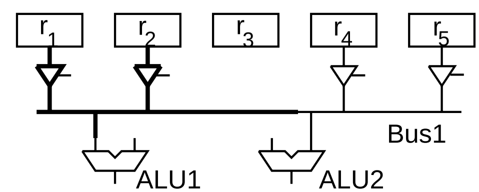
add mux to FU input



add FU and tristate to bus



add FU and mux



convert muxes to shared bus

CONSTRUCTIVE ALLOCATION

Algorithm 8.1

page 276

Allocation: Decomposition Methods

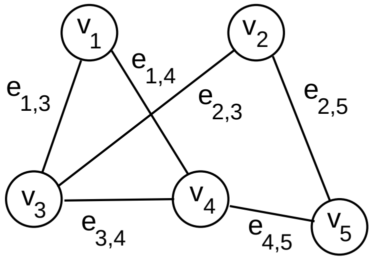
- *Clique Partitioning [TsSi86]*
- *Left−Edge Algorithm [KuPa87]*
- *Weighted Bipartite Matching [HCLH90]*

CLIQUE PARTITIONING

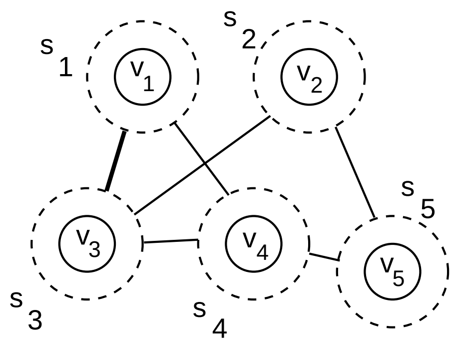
Algorithm 8.2

page 279

CLIQUE PARTITIONING EXAMPLE

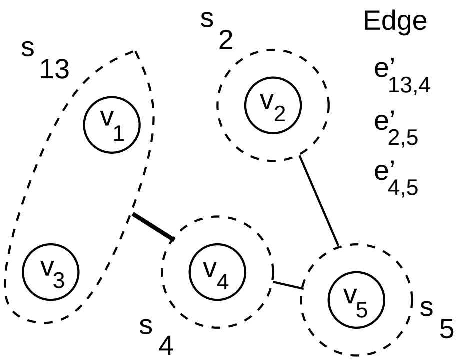


Graph G



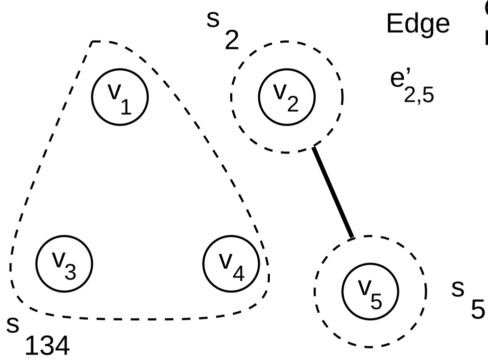
Edge	Common neighbors
$e'_{1,3}$	1
$e'_{1,4}$	1
$e'_{2,3}$	0
$e'_{2,5}$	0
$e'_{3,4}$	1
$e'_{4,5}$	0

Common Neighbors



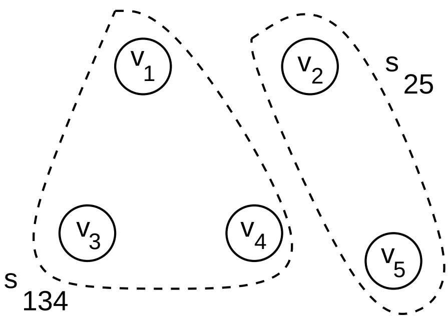
Edge	Common neighbors
$e'_{13,4}$	0
$e'_{2,5}$	0
$e'_{4,5}$	0

Supernode Creation 1



Edge	Common neighbors
$e'_{2,5}$	0

Supernode Creation 2



Supernode Creation 3

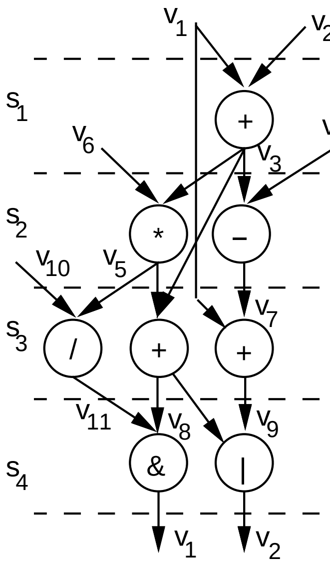
$$s_{134} = \{v_1, v_3, v_4\}$$

$$s_{25} = \{v_2, v_5\}$$

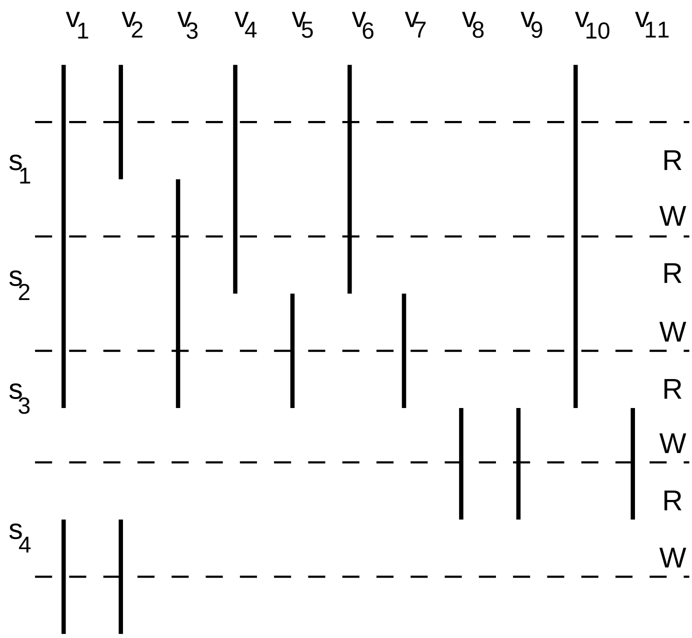
Cliques for Graph G

UC Irvine

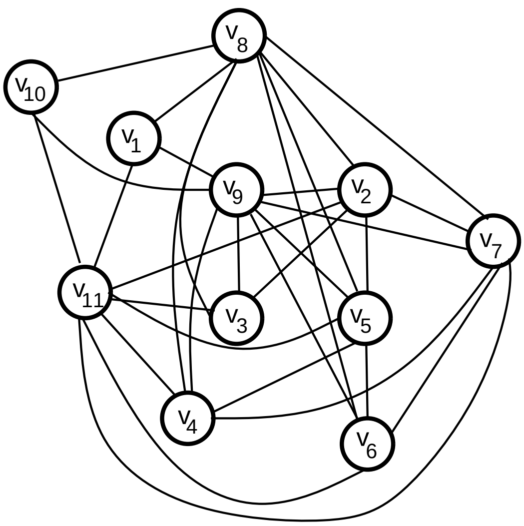
CLIQUE PARTITIONING FOR REGISTER BINDING



DFG



Lifetime intervals



Graph model

Cliques:

$r_1 = \{v_1, v_8\}$

$r_2 = \{v_2, v_3, v_9\}$

$r_3 = \{v_4, v_5, v_{11}\}$

$r_4 = \{v_6, v_7\}$

$r_5 = \{v_{10}\}$

A CP solution

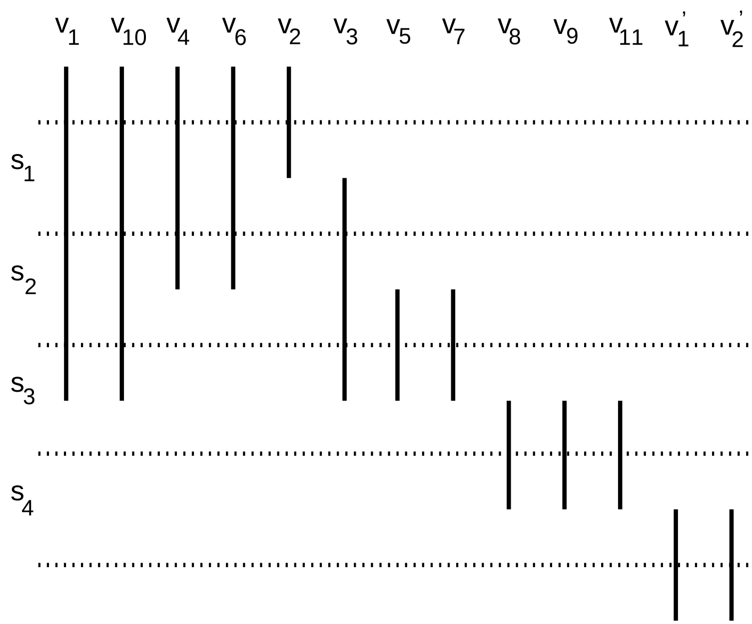
UC Irvine

REGISTER ALLOCATION USING LEFT−EDGE ALGORITHM

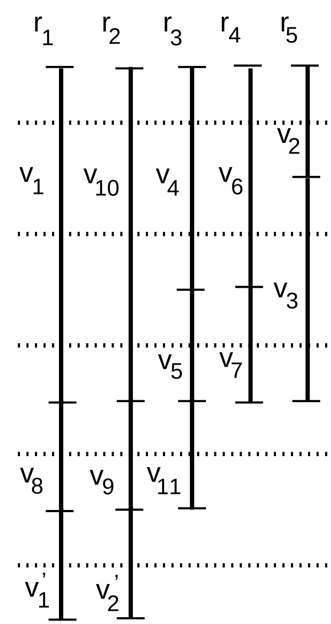
Algorithm 8.3

page 285

Register Binding with Left Edge Algorithm



Lifetimes



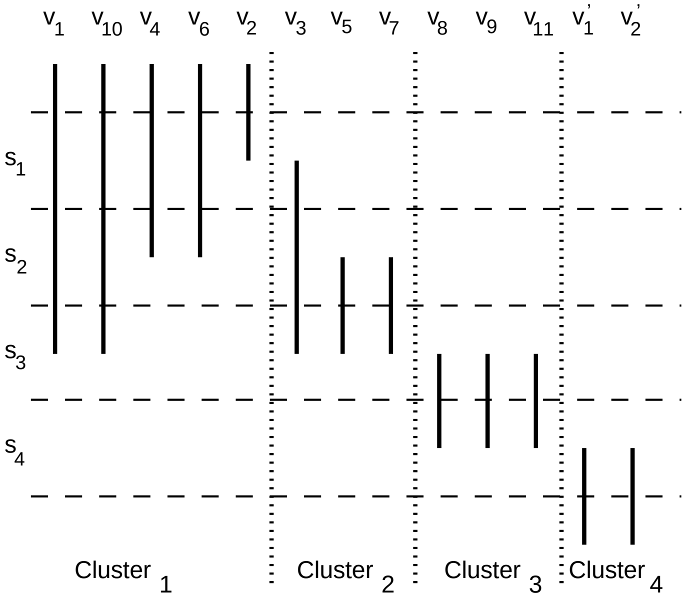
Registers

WEIGHTED BIPARTITE MATCHING ALGORITHM

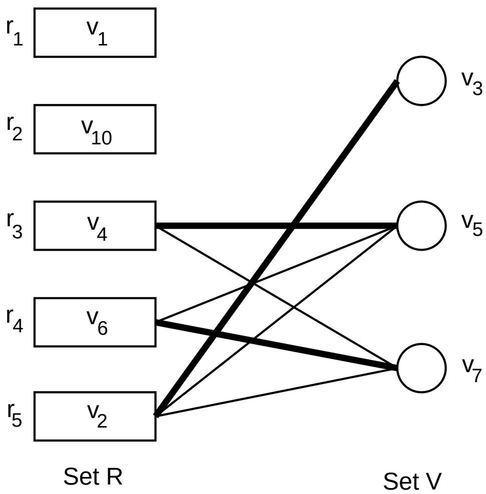
Algorithm 8.4

page 288

BIPARTITE MATCHING EXAMPLE FOR REGISTER ALLOCATION



Sorted Lifetime Intervals with Clusters



Bipartite Graph for Binding Vars in Cluster2 after Cluster1

$$r_1 = \{ v_1 , v_8 , v_1' \}$$
$$r_2 = \{ v_9 , v_{10} \}$$
$$r_3 = \{ v_4 , v_5 , v_{11} \}$$
$$r_4 = \{ v_6 , v_7 \}$$
$$r_5 = \{ v_2 , v_3 , v_2' \}$$

Final Register Binding

PAIRWISE EXCHANGE ALGORITHM

Algorithm 8.5

page 292

Interdependence of Scheduling and Allocation

- **Scheduling + Allocation => CU + DP**

  Scheduling: needs prelim component selection

  Allocation: needs "rough" schedule

- **Which one first?**

- **Iteration and Interleaving**

Allocation: Future Directions

- Interaction between scheduling and allocation
- Realistic cost functions
- Allocation of memories
- Different target architectures