ECE 695R: SYSTEM-ON-CHIP DESIGN

Module 3: Behavioral Synthesis
Lecture 3.14: Resource Sharing - Algorithms

Anand Raghunathan
raghunathan@purdue.edu

Fall 2014, ME 1052, T Th 12:00PM-1:15PM
Algorithms for Resource Sharing

- Left-edge algorithm
- Graph-based algorithms
- Optimizing interconnect using commutativity
Resource Sharing: Left-edge Algorithm

- Originally proposed to solve the channel routing problem in layout (1971)
  - Adapted to solve binding in behavioral synthesis by Kurdahi and Parker (1987)

- Pre-processing step: Perform lifetime analysis to represent operation (variable) lifetimes as intervals
Resource Sharing: Left-edge Algorithm

```
LEFT_EDGE(Variable Lifetimes L) {
    Sort elements of L in ascending order of start time \( t_i \); /* start time == left edge*/

    \text{reg\_index} = 0;
    \text{do}\{ /* each iteration creates one register*/
        S = 0; r = 0;
        while (there exists a variable whose left edge is \( \geq r \)) do {
            s = first element in L with \( l_s \geq r \);
            S = S \cup \{s\};
            r = r_s;
            delete s from L;
        }
        bind all variables in S to register \text{reg\_index};
        \text{reg\_index} = \text{reg\_index} + 1;
    } until (all variables have been bound to registers);
}
```

Variable lifetimes represented as intervals
Left-edge Algorithm: Example

Intervals

Register binding:
Left-edge Algorithm: Optimality and Complexity

• For acyclic DFGs, left-edge algorithm is optimal (will result in minimum number of registers / FUs).

• Complexity is dominated by sorting step (n log n).
Graph Algorithms for Resource Sharing

• Graph Coloring Formulation
  – Construct **Conflict Graph**
    • Vertices = Variables/operations
    • Edge => **cannot** share the same register/FU
  – Minimum coloring of the graph corresponds to a binding with minimum no. of resources
    • Graph Coloring: Assign colors to all vertices; two vertices that have an edge between them cannot have the same color
Graph Algorithms for Resource Sharing

• Clique Partitioning Formulation
  – Construct **Compatibility Graph**
    • Vertices = Variables/operations
    • Edge => can share the same register/FU
  – Minimum clique partitioning of the graph corresponds to a binding with minimum no. of resources
Question

• The left-edge algorithm is optimal and requires polynomial time
• Graph coloring and minimum clique partitioning are NP-Hard
• Contradiction?
Exploiting Commutativity (Port Assignment)

Mux inputs Before Swapping

Case 1

After Swapping

Case 1

Mux inputs Before Swapping

Case 2

After Swapping

Case 2
False Loops through Resource Sharing

• Combinational logic is normally acyclic
  – There exist some functions for which minimal implementation has structural loops
  – Most EDA tools do not handle circuits with combinational loops

• Resource sharing can introduce “false loops”

Example:

Need to impose additional constraints during resource sharing to avoid combinational loops
Resource Sharing: Summary

• Resource Sharing: Mapping Scheduled behavior into structure
  – FU binding: mapping operations to functional units
  – Register binding: mapping variables to registers
  – Interconnect generation: Generating the interconnect (muxes/buses, wires) to connect the registers and FUs