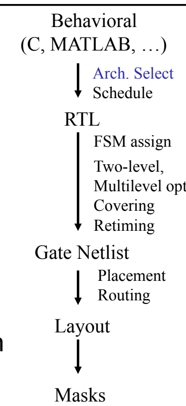
# ESE535: Electronic Design Automation

Day 10: February 18, 2015
Architecture Synthesis
(Provisioning, Allocation)



# Today

- Problem
- Brute-Force/Exhaustive
- Greedy
- Estimators
- Analytical Provisioning
- ILP Schedule and Provision



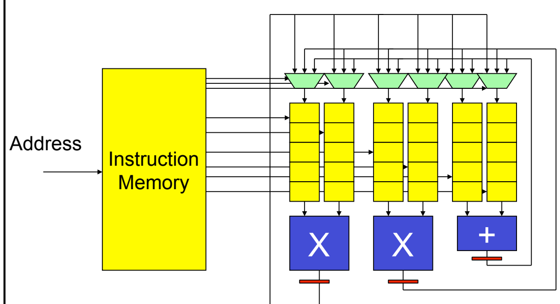



# Previously

- General formulation for scheduled operator sharing
  - VLIW
- Fast algorithms for scheduling onto fixed resource set
  - List Scheduling
- More extensive algorithms for time-constrained
  - Force Directed, Branch-and-Bound



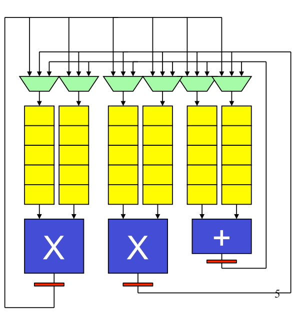
# VLIW






# Today

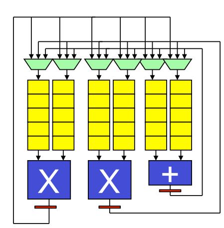
- How do we determine the set of resources?






# Today: Provisioning

- Given
  - An area budget
  - A graph to schedule
  - A library of operators
- Determine:
  - Delay minimizing set of operators
    - Or delay-achieving set of operators
  - i.e. select the operator set



## Exhaustive

1. Identify all area-feasible operator sets
   – E.g. preclass exercise
2. Schedule for each
3. Select best

- → optimal
- Drawbacks?

7



## Exhaustive

- How large is space of feasible operator sets?
  – As function of
    - operator types – O
      – Types: add, multiply, divide, ….
    - Maximum number of operators of type m

$$m^O$$

8



## Implication

- Feasible operator space can be too large to explore exhaustively

9



## Greedy Incremental

- Start with one of each operator
- While (there is area to hold an operator)
  – Which single operator
    - Can be added without exceeding area limit?
    - Schedule (maybe list-schedule?)
    - Calculate benefit (maybe $\Delta T/\Delta A$?)
    - Pick largest benefit
  – Add one operator of that type
- How long does this run?
  – $T_{schedule}(E)$* O(operator-types * A)

10



## Greedy Incremental

- Work Preclass with greedy incremental
  – For each step
    - half class evaluate each candidate resource

11

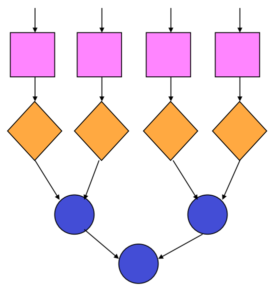


## Greedy Incremental

- Start with one of each operator
- While (there is area to hold an operator)
  – Which single operator
    - Can be added without exceeding area limit?
    - Schedule (maybe list-schedule?)
    - Calculate benefit (maybe $\Delta T/\Delta A$?)
    - Pick largest benefit
  – Add one operator of that type
- Weakness?

12

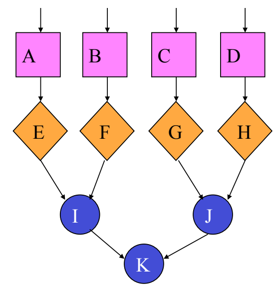
# Example



Find best 5 operator solution.



# Example



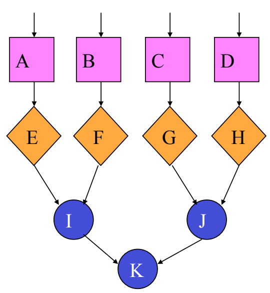

Find best 5 operator solution.



# Example

One of each.

| Sq | Dia | Circ |
|---|---|---|
| A | | |
| B | E | |
| C | F | |
| D | G | I |
| | H | |
| | | J |
| | | K |



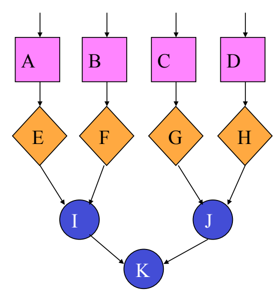

Find best 5 operator solution.



# Example

Two Squares

| Sq | Dia | Circ |
|---|---|---|
| A,B | | |
| C,D | E | |
| | F | |
| | G | I |
| | H | |
| | | J |
| | | K |



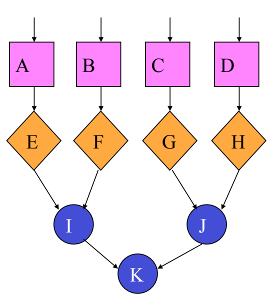

Find best 5 operator solution.



# Example

Two Diamonds

| Sq | Dia | Circ |
|---|---|---|
| A | | |
| B | E | |
| C | F | |
| D | G | I |
| | H | |
| | | J |
| | | K |



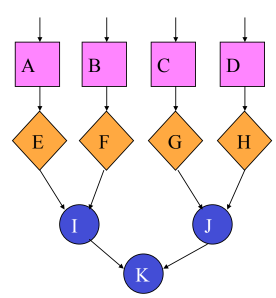

Find best 5 operator solution.



# Example

Two Circles

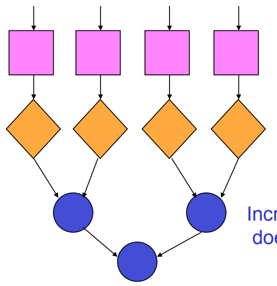
| Sq | Dia | Circ |
|---|---|---|
| A | | |
| B | E | |
| C | F | |
| D | G | I |
| | H | |
| | | J |
| | | K |




Find best 5 operator solution.

# Example

Which should greedy add?



Find best 5 operator solution.

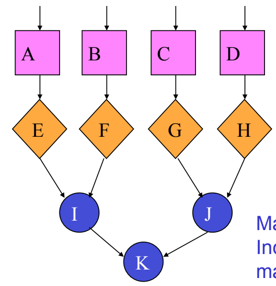
Incremental addition does not accelerate.



# Example

Two sqs
+ Two diamonds

| Sq | Dia | Circ |
|---|---|---|
| A,B | | |
| C,D | E,F | |
| | G,H | I |
| | | J |
| | | K |
| | | |
| | | |




Find best 5 operator solution.

Max effect: Incremental may not suggest next single addition.



# Analytic Formulation



# Challenge

- Scheduling expensive
  - O(|E|) or O(|E|*log(|V|)) using list-schedule
- Results not analytic
  - Cannot write an equation around them
- Bounds are sometimes useful
  - No precedence → is resource bound
  - Often one bound dominates
    - Latency bound unaffected by operator count



# Estimations

- Step 1: estimate with resource bound
  - O(|E|) vs. O(|V|) evaluation
- Step 2: use estimate in equations
  - $T=\max(N_1/M_1, N_2/M_2, \ldots.)$
- Most useful when RB>>CP



# Constraints

- Let $A_i$ be area of operator type i
- Let $M_i$ by number of operators of type i

$$\sum A_i \times M_i \leq Area$$

(start summary of variables on board)

## Achieve Time Target

- Want to achieve a schedule in T cycles
- What constraint equation does that imply? (what property must hold?)
- Each resource bound must be less than T cycles:
  - $N_i/M_i \leq T$



## Algebraic Solve

- Set of equations
  - $N_i/M_i \leq T$
  - $\Sigma A_i M_i \leq Area$
- Assume equality for time bound
- $N_i/M_i = T \Rightarrow M_i = N_i/T$

$$\frac{\sum A_i \times N_i}{T} \leq Area$$



## Rearranging

$$\frac{\sum A_i \times N_i}{T} \leq Area$$

$$\frac{\sum A_i \times N_i}{Area} \leq T$$



## Bounding T

- Gives Lower Bound on T

$$\frac{\sum A_i \times N_i}{Area} \leq T$$

Intuition: N of each is right balance given unbounded area; Scale to area available.



## Preclass

- What is $T_{lower}$ for preclass?

$$\frac{\sum A_i \times N_i}{Area} \leq T$$

$$T \geq \frac{1 \times 8 + 2 \times 4}{7} = \frac{16}{7} \approx 2.3$$

$$T \geq 3$$



## Back Substitute from T to x

- $M_i = N_i/T$

$$\frac{\sum A_i \times N_i}{Area} \leq T$$

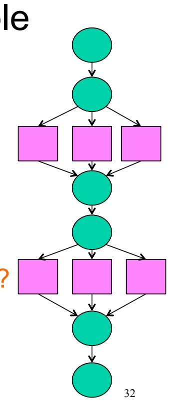
- $M_i$ won't necessarily be integer
  - Round down definitely feasible solution
  - May have room to move a few up by 1
- Reduces range may need to search
  - (just over the residual area once rounded down)

# Preclass

- $M_i=N_i/T$
- T>=3
- $M_{add}$, $M_{mpy}$ ?
- $M_{add} = 8/3 \rightarrow$ 2 or 3
- $M_{mpy} = 4/3 \rightarrow$ 1 or 2



# Counter Example



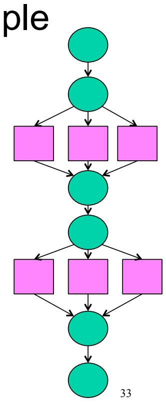
- 1 Unit each
- Area = 4 Units
- What would analytic predict?
- What is best?
- How does CP compare to RB?

- Analytic Resource Estimate
  - Most useful when RB>>CP

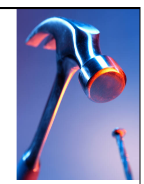


# Analytic Counter Example

- How would greedy incremental work on this one?







# ILP

Maybe we can do exhaustive,
if we formulate properly.



# ILP

- Integer Linear Programming
- Formulate set of linear equation constraints (inequalities)
  - $Ax_0+Bx_1+Cx_2 \leq D$
  - $x_0+x_1=1$
  - A,B,C,D – constants
  - $x_i$ – variables to satisfy
  - No products on variables, just linear weighted sums
- Can constrain variables to integers
- No polynomial time guarantee
  - But often practical
  - Solvers exist (significant piece on April 1 (seriously))



# ILP Provision and Schedule

Now to make it look like an ILP nail…

- Formulate operator selection and scheduling as ILP problem

## Formulation

- Integer variables $M_i$
  - number of operators of type i
- 0-1 (binary) variables $x_{i,j}$
  - 1 if node i is scheduled into timestep j
  - 0 otherwise
- Variable assignment completely specifies operator selection and schedule
- This formulation for achieving a target time T (time constrained)
  - j ranges 0 to T-1



## Target T → Min T

- Formulation targets T
- What if we don't know T?
  - Want to minimize T?
- Do binary search for minimum T
  - How does that impact solution time?



## Constraints

What properties must hold true for a solution to be valid?

1. Total area constraints
2. Not assign too many things to a timestep
3. Assign every node to some timestep
4. Maintain precedence



## (1) Total Area

- Same as before

$$\sum A_i \times M_i \leq Area$$



## (2) Not overload timestep

- For each timestep j
  - For each operator type k

$$\sum_{o_i \in FU_k} x_{i,j} \leq M_k$$



## (3) Node is scheduled

- For each node in graph

$$\sum_j x_{i,j} = 1$$

Can narrow to sum over slack window.

## (4) Precedence Holds

- For each edge from node *src* to node *snk*

$$\sum_j j \times x_{src,j} - \sum_j j \times x_{snk,j} \leq -1$$

Can narrow to sum over slack windows.



## Example (Time Permitting)

- What are the ILP equations for the preclass example?
  1. Total area constraints
  2. Not assign too many things to a timestep
  3. Assign every node to some timestep
  4. Maintain precedence



## Constraints

1. Total area constraints
2. Not assign too many things to a timestep
3. Assign every node to some timestep
4. Maintain precedence



## ILP Solver

- ILP Solver can take these constraints and find a solution (satisfying assignment)
- On Wednesday, will see how to start to make this practical



## Round up Algorithms and Runtimes

- Exhaustive Schedule
- Greedy Schedule
- Analytic Estimates
- ILP formulation



## Big Ideas:

- Estimators
- Dominating Effects
- Reformulate as a problem we already have a solution for
  - ILP
- Technique: Greedy
- Technique: ILP

## Admin

- Assignment 5 Thursday
- No class on Monday
  - Will have class on Wednesday
- No assignment 6 supplement
  - Focus on project and writeup
- Reading for Wednesday online