Agenda

1. Behavioral Synthesis Problem
2. Area and Performance Estimation
3. Strategy for Architecture Optimization
4. Data-path Synthesis
5. Control-path Synthesis
Agenda

1. Behavioral Synthesis Problem
2. Area and Performance Estimation
3. Strategy for Architecture Optimization
4. Data-path Synthesis
5. Control-path Synthesis
Overview

- Architecture Synthesis
  - Behavioral description
    - exploration of design space
  - RTL description (structural view of data-path + logic-level specification of control unit)
- Data-path
  - interconnection of resources
- Resource
  - functional resource (ALU, adder, multiplier, ...)
  - memory resource (register, RAM, ROM, ...)
  - interface resource (bus, steering logic, I/O pad, ...)
Overview

- Behavioral model (CDFG)
- Constraints (timing, area, performance, resource binding)
- Architecture synthesis
- RTL description
- Resource (library + module generator)
- Primitives: area, delay given
- Area, delay estimated
Architectural Synthesis Problem

Place operations in TIME and SPACE

scheduling resource binding

Given a CDFG $G(V, E)$, $V = \{v_i \mid i = 0, 1, \ldots, n\}$,
$E = \{(v_i, v_j) \mid i, j = 0, 1, \ldots, n\}$
Temporal Domain: Scheduling

- Task of determining start times of operations subject to precedence constraints
- Latency = $\lambda = t_n - t_0$

start time of sink      start time of source
Definition

Scheduling is a mapping

$$\varphi: V \rightarrow \mathbb{Z}^+$$

where $$\varphi(v_i) = t_i$$, such that

$$t_i \geq t_j + d_j, \forall i, j | (v_j, v_i) \in E$$

Temporal Domain: Scheduling

$$\lambda = 5 - 1 = 4$$
Temporal Domain: Scheduling
Chaining

- multiplier: 35 ns
- others: 25 ns
- cycle time: 50 ns

\[ \lambda = 4 - 1 = 3 \]
Temporal Domain: Scheduling
Multi-cycle Operation

- multiplier: 35 ns
- others: 25 ns
- cycle time: 25 ns

\[ \lambda = 7 - 1 = 6 \]
Temporal Domain: Scheduling

- multiplier: 35 ns
- others: 25 ns
- cycle time: 40 ns

```
C-Step 1
v1 * v2 * v6 * v8 < v11
C-Step 2
v3 * v7 + v9 < v11
C-Step 3
v4 - v5
C-Step 4

\( \lambda = 5 - 1 = 4 \)
```

Throughput?
Spatial Domain: Binding

- $R = \{1, 2, ..., n_{res}\}$: set of resource types
- $\tau: V \rightarrow R$
  - one-to-many: resource (module) selection can be applied
  - many-to-one: resource sharing can be applied
- **Definition:**
  Resource binding is a mapping
  $\beta: V \rightarrow R \times \mathbb{Z}^+$
  where $\beta(v_i) = (t, r)$ denotes that $v_i$ is implemented by $r$-th instance of resource type $\tau(v_i) = t, t \in R$
- Dedicated resource binding $\rightarrow \beta$ is one-to-one
- Shared resource $\Rightarrow \beta$ is many-to-one
  $\Rightarrow$ the corresponding operations cannot execute concurrently
Resource binding can be represented by a hypergraph.
Spatial Domain: Binding Constraints

- partial binding
  - resource binding must be compatible with the partial binding

- upper bound on resource usage of each type
  - resource allocation: \( \{a_k \mid k=1, 2, \ldots, n_{res}\} \)
  - resource binding: \( \beta(v_i) = (t, r), r \leq a_t \) for each operation \( v_i \)
Agenda

1. Behavioral Synthesis Problem
2. Area and Performance Estimation
3. Strategy for Architecture Optimization
4. Data-path Synthesis
5. Control-path Synthesis
Area and Performance Estimation

- During the scheduling and binding, the area and performance cannot be determined accurately (not in physical abstraction)
- Have to estimate based on some abstract metrics and assumptions
Resource-dominated Circuit

- **mult**: 5 units, 35 ns
- **ALU**: 1 unit, 25 ns
- **cycle time**: 40 ns
- **overhead**: 1 unit

- **Dedicated resource**
  - **area**: $6 \times 5 + 5 \times 1 + 1 = 36$ units
  - **latency**: 4 cycles

- **One instance for each type**
  - **area**: $5 + 1 + 1 = 7$ units
  - **latency**: 7 cycles
General Circuits

- **Register**
  - used at cycle boundary
  - add to area and time (set-up time + propagation delay)

- **Steering logic**
  - MUX: area and time can be estimated easily
  - Bus: drivers must be considered

- **Wiring**
  - fast floor-planner can be used
  - average interconnect length = (#blocks)^α, 0 ≤ α ≤ 1

- **Control unit**
  - latency = #control steps --> #states
  - area (address space in ROM-based control units)
  - optimization using state encoding, state minimization, ...
  - hard to estimate area
General Circuits

- **Number of registers**
  - 7 intermediate variables
  - 3 loop variables (x, y, u)
  - 3 loop invariants (a, 3, dx)
  - total 13 variables
  - compute #registers considering variables’ lifetimes

Check further details in example 4.4.2 in page 158, reference textbook.
Steering Logic

- dedicated: no MUX
- shared resources:
  - 5 operand pairs for mult
  - 5 operand pairs for ALU
  - shared registers also need MUXs
- 1 mult, 1 ALU, 2 registers for intermediate variables:
  - four 5-way MUXs + two 2-way MUXs
1. Behavioral Synthesis Problem
2. Area and Performance Estimation
3. Strategy for Architecture Optimization
4. Data-path Synthesis
5. Control-path Synthesis
Strategy for Architecture Optimization

- Three criteria
  - Area
  - Latency
  - Cycle-time

- Trade-off
  - Area/latency (fixed cycle-time)
  - Cycle-time/latency (fixed area)
  - Cycle-time/Area (fixed latency)
Area/Latency Optimization

- Given cycle-time
  - resource-constrained minimum-latency scheduling
  - latency-constrained minimum-resource scheduling
- Scheduling before/after binding
  - circular dependency
  - solve jointly or iteratively
  - resource-dominated: scheduling before binding (DSP)
  - control-dominated: binding before scheduling
Area/Latency Optimization

Example

area <= 20 units  mult : 35 ns, 5 units
latency <= 8      ALU : 25 ns, 1 unit
area overhead = 1 unit

cycle-time = 40 ns

(#mult, #ALU) = (1, 1) --> latency = 7, area = 7
(#mult, #ALU) = (1, 2) --> latency = 7, area = 8
(#mult, #ALU) = (2, 1) --> latency = 5, area = 12
(#mult, #ALU) = (2, 2) --> latency = 4, area = 13

cycle-time = 30 ns --> mult in multi-cycle operation

(#mult, #ALU) = (2, 1) --> latency = 8, area = 12
(#mult, #ALU) = (3, 1) --> latency = 7, area = 17
(#mult, #ALU) = (3, 2) --> latency = 6, area = 18
Area/Latency Optimization

Example

<table>
<thead>
<tr>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 ns</td>
</tr>
<tr>
<td>40 ns</td>
</tr>
</tbody>
</table>

Cycle-time

(a, b)

(a not a Pareto point)

(a Pareto point)
Cycle-Time/Latency Optimization

- Fixed area: after binding
- Cycle-time/latency trade-off by multi-cycle operation or chaining
- Example

  - constraints:
    - $20 \text{ ns} \leq \text{cycle-time} \leq 50 \text{ ns}$
    - $\text{latency} \leq 8$

  - Dedicated resource
    - cycle-time (ns) latency
      - $t > 35$ 4
      - $25 < t < 35$ 6
      - $20 < t < 25$ 8
      - $t = 50$ 3

  - 1 mult + 1 ALU
    - $t > 35$ --> latency = 7

  - mult : multi-cycle operation
  - ALU : multi-cycle operation
  - mult, ALU : multi-cycle operation
  - ALU : chaining

  - mult : 35 ns, 5 units
  - ALU : 25 ns, 1 unit
Cycle-Time/Latency Optimization

- **Dedicated resource**
  - cycle-time (ns) latency
  - $t > 35$ 4 mult : multi-cycle operation
  - $25 < t < 35$ 6 mult, ALU : multi-cycle operation
  - $20 < t < 25$ 8 ALU : chaining
  - $t = 50$ 3
- **1 mult + 1 ALU**
  - $t > 35$ $\rightarrow$ latency = 7
Cycle-time/Area Optimization

- fixed latency (after scheduling)
- resource sharing needs multiplexers

  less area  delay → cycle-time increase
Cycle-time/Area Optimization

Example

- dedicated (no MUX)
  
  latency = 4
  
  area = 36, cycle-time = mult delay = 35 ns
Cycle-time/Area Optimization Example

4 mult + 2 ALU
Latency = 4
Area
- 2 * 3-way MUX: area = 0.3 unit, delay = 3ns
- 6 * 2-way MUX: area = 0.2 unit, delay = 2ns

total area = 4 * 5 + 2 * 1 + 1 + 2 * 0.3 + 6 * 0.2 = 24.8

cycle-time = max (25 + 3, 35 + 2) = 37 ns

1 mult + 1 ALU
latency = 7
2 * 6-way MUX + 2 * 5-way MUX

total area = 1 * 5 + 1 * 1 + 1 + 2 * 0.6 + 2 * 0.5 = 9.2

cycle-time = 35 + 6 = 41 ns
Cycle-time/Area Optimization

**Example**

4 mult + 2 ALU
Latency = 4
Area
- 2 * 3-way MUX: area = 0.3 unit, delay = 3ns
- 6 * 2-way MUX: area = 0.2 unit, delay = 2ns
total area = 4 * 5 + 2 * 1 + 1 + 2 * 0.3 + 6 * 0.2 = 24.8
cycle-time = max (25 + 3, 35 + 2) = 37 ns

1 mult + 1 ALU
latency = 7
2 * 6-way MUX + 2 * 5-way MUX
total area = 1 * 5 + 1 * 1 + 1 + 2 * 0.6 + 2 * 0.5 = 9.2
cycle-time = 35 + 6 = 41 ns
Agenda

1. Behavioral Synthesis Problem
2. Area and Performance Estimation
3. Strategy for Architecture Optimization
4. Data-path Synthesis (Chapter 4.6)
5. Control-path Synthesis
Agenda

1. Behavioral Synthesis Problem
2. Area and Performance Estimation
3. Strategy for Architecture Optimization
4. Data-path Synthesis
5. Control-path Synthesis (Chapter 4.7)
Summary

- Architectural synthesis optimization is multi-objective optimization
  - Area
  - Latency
  - Cycle-time
- Trade-off must be made
Thank You!