Control Synthesis
Outline

• Data-path synthesis.
• Control-unit synthesis.
Data path synthesis

- Resource binding.
- Connectivity synthesis:
  - Connection of resources to: *multiplexers busses and registers*.
  - Control unit interface.
  - I/O ports.
- Physical data-path synthesis.
Example

REGISTERS

DATA-PATH

CONTROL-UNIT

enable
mux control
ALU control (+, -, <)
c

TU Delft
Control synthesis

• Synthesis of the control unit
• Logic model:
  - Synchronous FSM.
• Physical implementation:
  - Microcode (ROM, PLA).
  - Hard-wired FSM.
  - Distributed FSM.
Control synthesis

- Synthesize circuit that:
  - Executes scheduled operations.
  - Provides synchronization.
  - Supports:
    * Iteration.
    * Branching.
    * Hierarchy.
    * Interfaces.

Assumption:
- Synchronous implementation.
- Control unit is a FSM (or connection of FSM’s).
Controlling scheduled operations

- Simple model:
  - No branching, iteration, hierarchy.
  - No data-dependent delays.
- Implementation:
  - FSM-oriented design:
    - Hardware: PLAs, gates, registers.
    - One FSM state per schedule level.
  - Microcode-oriented design:
    - Hardware: ROM, PLA, counter.
FSM-based implementation

• Simple model:
  - *next-state function*: unconditional.
  - *output function*: activate operations.

• Extended model:
  - Branching and iteration:
    * Conditional next-state function.
  - Hierarchy:
    * Hierarchical FSM connection.
Example
Microcode implementation

• Horizontal microcode:
  - One bit per *activation* signal.
  - One microcode word per schedule level.
  - Maximum performance.
  - Wide words.

• Vertical microcode:
  - Encode each resource *activation* signal.
  - Shorter words.
  - One (or more) words per schedule level.
Example of horizontal microcode

```

TIME 1
*  2  *  6  *  8  +  10

TIME 2
*  3  *  7  +  9  <  11

TIME 3
-  4

TIME 4
-  5

```

Address          Microwords
---              ---------------------
00               1100010101010
01               001001010101
10               000100000000
11               000010000000

Reset            Counter          Activation signals

---              ----------          ---------------------

```
Example of vertical microcode

![Diagram of a microcode example showing nodes and time steps with corresponding activation signals and microwords.]

**Microwords**

<table>
<thead>
<tr>
<th>Microword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
</tr>
<tr>
<td>0010</td>
</tr>
<tr>
<td>0110</td>
</tr>
<tr>
<td>1000</td>
</tr>
<tr>
<td>1010</td>
</tr>
<tr>
<td>0011</td>
</tr>
<tr>
<td>0111</td>
</tr>
<tr>
<td>1001</td>
</tr>
<tr>
<td>1011</td>
</tr>
<tr>
<td>0100</td>
</tr>
<tr>
<td>0101</td>
</tr>
</tbody>
</table>

**Decoder**

Activation signals
Microcode compaction problem

• Partition ROM word into fields.
• Encode signals in each field.
• Allow for a code for NOP.
• Activation signals in each field must not be concurrent.
• Problems:
  • Minimize number of fields.
  • Minimize total ROM width.
Microcode optimization

• Conflict graph:
  - Concurrent operations.
  - Optimum *vertex coloring*
    yields minimum number of *fields*.

• Compatibility graph:
  - Non-concurrent operations.
  - Optimum *clique partitioning*
    yields minimum number of *fields*.
  - Minimum *weighted* clique partitioning
    yields minimum number of *bits*. 
### Example

<table>
<thead>
<tr>
<th>field</th>
<th>op</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>01</td>
</tr>
<tr>
<td>A</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>01</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>D</td>
<td>8</td>
<td>01</td>
</tr>
<tr>
<td>D</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>E</td>
<td>10</td>
<td>01</td>
</tr>
<tr>
<td>E</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

### Microword format

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Microwords

<table>
<thead>
<tr>
<th>01</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>00</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>01</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>01</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>01</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>00</td>
</tr>
</tbody>
</table>

### Activation signals

1, 3, 4  2  6, 7, 5  8, 9  10, 11
Hierarchical control

- Exploit the hierarchical structure of sequencing graphs.
- One controller per entity.
- Interconnected *finite state machines*.
- Handshake:
  - *activate* signals.
  - *condition* signals.
  - *reset* signals.
Example
Summary Control synthesis

- Different approaches.
- Implementations:
  - FSM, connection of FSMs or ROM.
- Techniques:
  - Bounded delays only:
    * FSM - microcode.
  - Unbounded delays:
    * Different methods to provide synchronization.