Control Synthesis

Outline

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

Data path synthesis

- Resource binding.
- Connectivity synthesis:

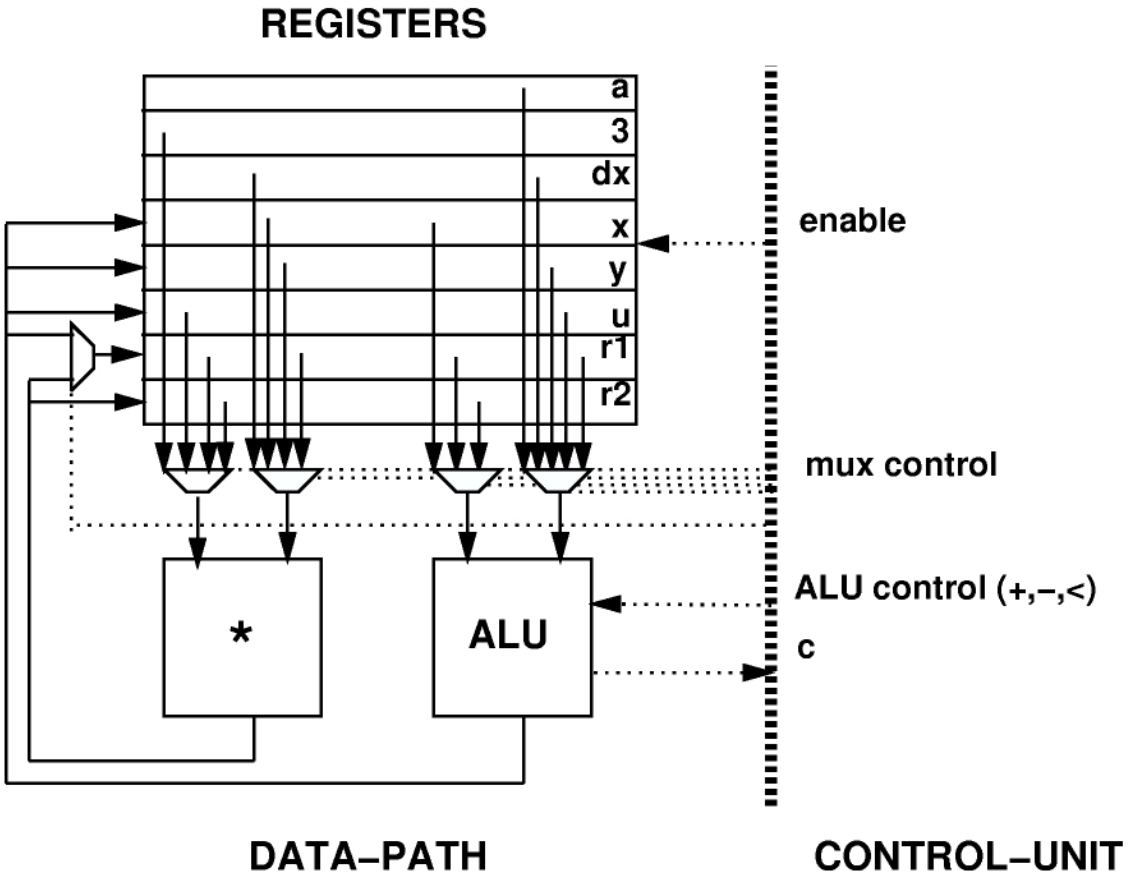
  - Connection of resources to:

    multiplexers busses and *registers.*

  - Control unit interface.

  - I/O ports.
- Physical data-path synthesis.

Example



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) .

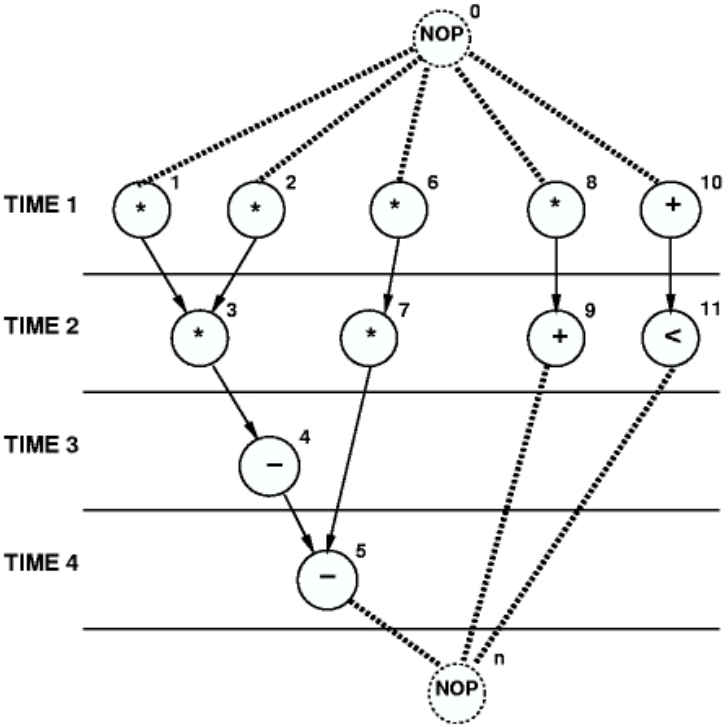
Controlling scheduled operations

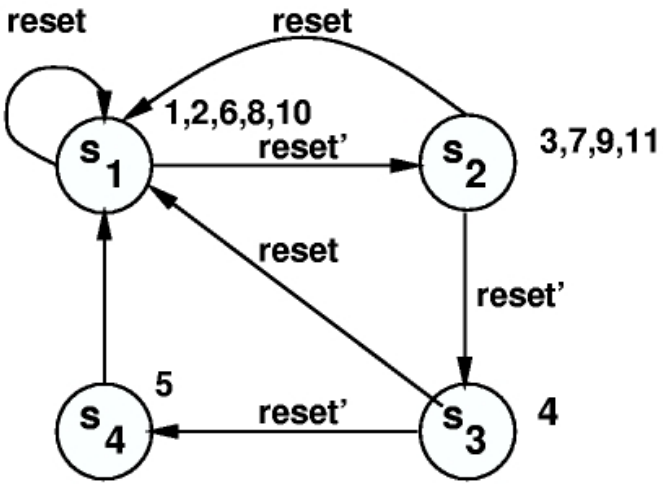
- Simple model:
  - No branching, iteration, hierarchy.
  - No data-dependent delays.
- Implementation:
  - FSM-oriented design:
    - Hardware: PLAs, gates, registers.
    - One FSM state per schedule level.
  - Microcode-oriented design:
    - Hardware: ROM, PLA, counter.

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



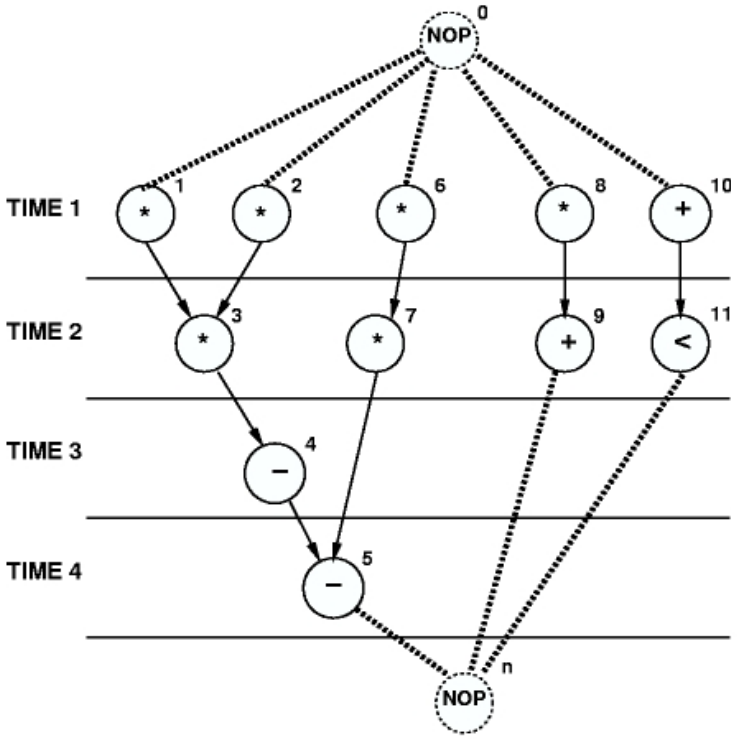


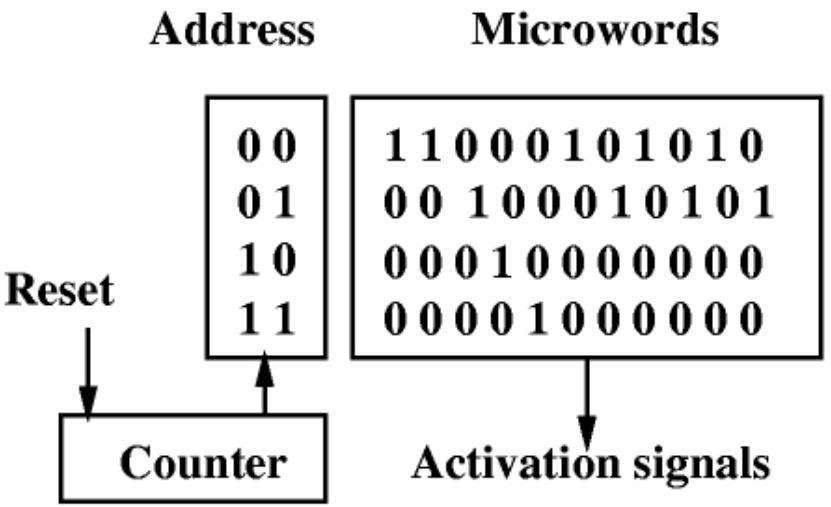
TU Delft

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.

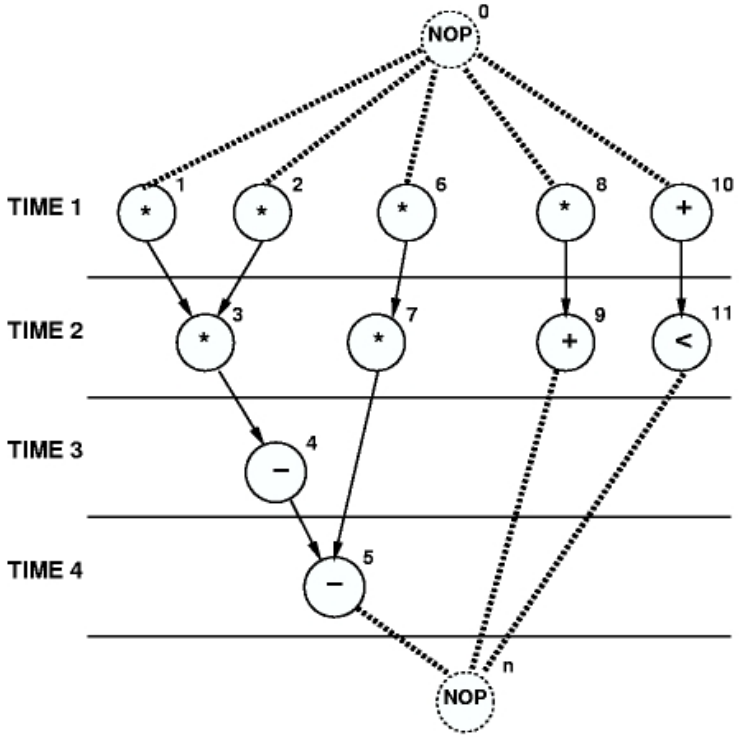
Example of horizontal microcode

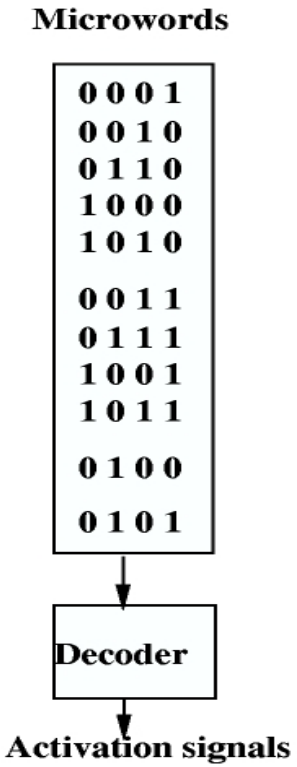




TU Delft

Example of vertical microcode





TU Delft

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.

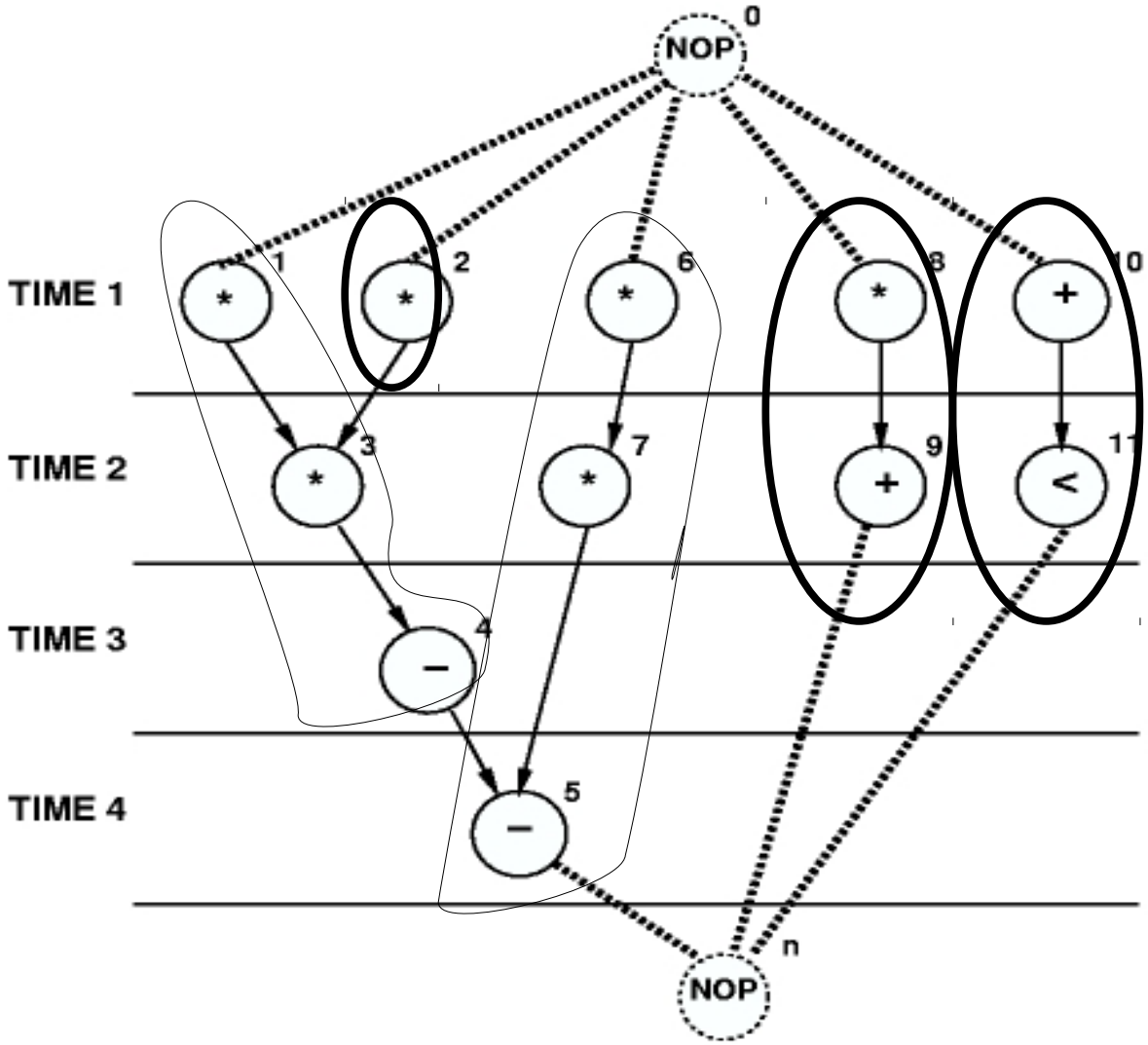
Microcode optimization

- 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

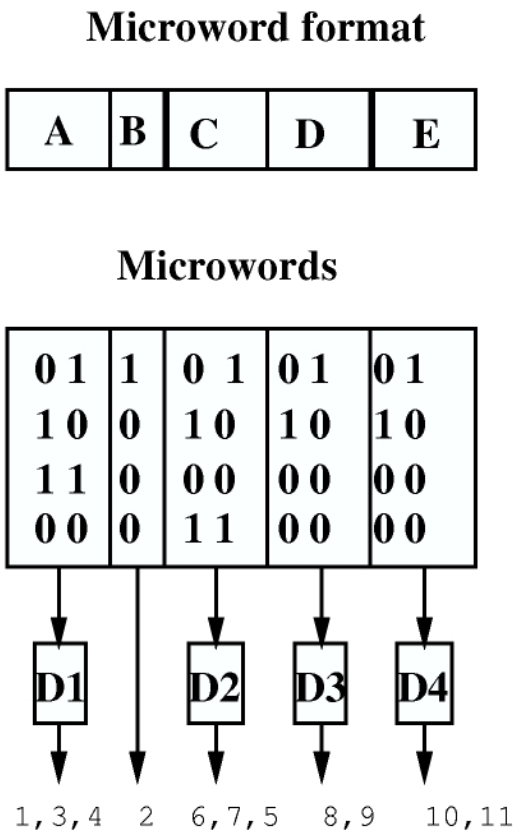
    yields minimum number of *bits*.



TU Delft

Example

field	op	code
A	1	01
A	3	10
A	4	11
B	2	1
C	6	01
C	7	10
C	5	11
D	8	01
D	9	10
E	10	01
E	11	10

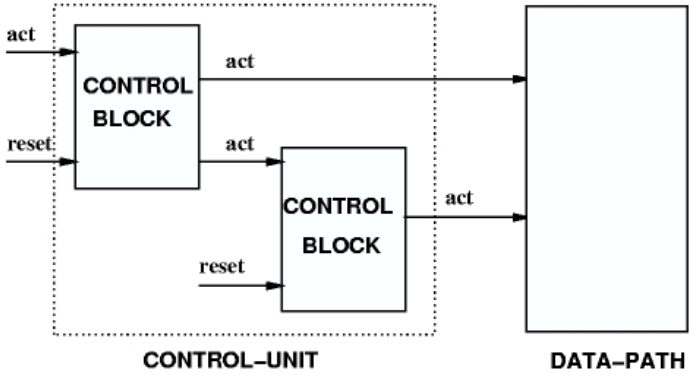


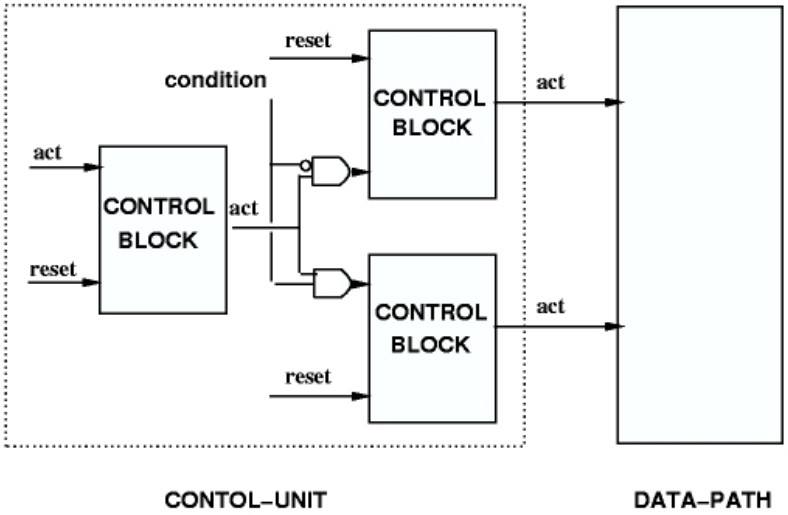
TU Delft

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





TU Delft

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.