System Design -
Methods and Models

SS 2009
Hw/Sw Codesign

Christian Plessl
Overview

• Abstraction layers
• System synthesis
• Graph models for control and data flow
Abstraction Layers

- **Behavior**
- **Structure**

Diagram illustrating the relationship between hardware (HW) and software (SW) at different abstraction layers:

- **System**
  - **Architecture**
    - **Logic**
  - **Module**
    - **Block**

The diagram visually represents the hierarchical nature of system design, with behavior and structure at the top, and hardware and software at the bottom.
System Design

specification

system synthesis

SW-synthesis

machine code

I/F-synthesis

net lists

HW-synthesis

estimation
Synthesis

• Synthesis transforms behavior into structure

• Synthesis tasks
  – allocation: select components
  – binding: assign functions to components
  – scheduling: determine execution order
Specification on the System Level

task graph

A

B

C

D

E

constraints

< 200 ms
Allocation on the System Level

- Processors, dedicated hardware
- Memory, I/O
- Interconnection structures
Binding on the System Level
Scheduling on the System Level

- DSP
- MIPS
- ASIC

Time

A → B → C → D → E
Optimal Design Points

1, 2, 4, 6 are Pareto points
Control/Data Flow Models

• Graph $G(V,E)$
  – set of vertices (nodes) $V$ : operations, tasks
  – set of arcs (edges) $E$ : dependencies

• Dependencies
  – data dependency
  – control dependency
  – resource conflict (caused by implementation)

• Models
  – data flow graph (DFG)
  – control flow graph (CFG)
  – combined control/data flow graph (CDFG)
    (eg. sequence graph)
Data Flow Graph (DFG)

\[ x = 3a + b*b - c; \]
\[ y = a + b*x; \]
\[ z = b - c*(a + b); \]
what_is_this {
  read (a,b);
  done = FALSE;
  repeat {
    if (a>b)
      a = a-b;
    elseif (b>a)
      b = b-a;
    else done = TRUE;
  } until done;
  write (a);
}
Sequence Graph

- **Hierarchy of chained units**
  - units model data flow
  - hierarchy models control flow

- **Special nodes**
  - start/end nodes: NOP (no operation)
  - branch nodes (BR)
  - iteration nodes (LOOP)
  - module call nodes (CALL)

- **Attributes**
  - nodes: computation times, cost, …
  - edges: conditions for branches and iterations
\[ w = a + b; \]
\[ x = w \times c; \]
\[ y = b \times b; \]
\[ z = w - c; \]
\[ c = a < b; \]
\[ \text{IF } (c) \text{ THEN} \]
\[ p = m + n; \]
\[ q = m \times n; \]
\[ \text{ENDIF} \]
\[ x = a - b; \]
d = 2*x;
WHILE (d<5) DO
  write(d);
  d = d + 1;
ENDWHILE
d = x - y;
e = d * x;
sub(x, y);
...

PROCEDURE sub (m, n)
  p = m + n;
  q = m * n;
END sub