System Design - Methods and Models

SS 2010
Hw/Sw Codesign

Christian Plessl
Overview

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

specification

system synthesis

estimation

SW-synthesis

I/F-synthesis

HW-synthesis

machine code

net lists
• Synthesis transforms behavior into structure

• Synthesis tasks
  – allocation: select components
  – binding: assign functions to components
  – scheduling: determine execution order

In the following: Illustration of these steps in system level design
Specification on the System Level

| A | B | C | D | E |

- **task graph**
- **constraints**

\[< 200 \text{ ms}\]
Allocation on the System Level

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

[Diagram showing connections between hardware components (A, B, C, D, E) and software components (MIPS, DSP, ASIC, Mem)]
Scheduling on the System Level

The diagram illustrates the scheduling of tasks on different hardware components. The tasks A, B, C, D, and E are scheduled on DSP, MIPS, and ASIC components over time.

- Task A starts first on the DSP component.
- Task B follows on the MIPS component.
- Task C is next on the same MIPS component.
- Task D follows on the ASIC component.
- Task E is the last on the ASIC component.

The diagram shows the temporal order and dependencies between tasks and hardware components.
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)
\[ x = 3a + b^2 - c; \]
\[ y = a + bx; \]
\[ z = b - c(a + b); \]
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;
IF (c) THEN
  p = m + n;
  q = m * n;
ENDIF
x = a - b;
\[
d = 2x;
\text{WHILE } (d<5) \text{DO}
\hspace{1em} \text{write}(d);
\hspace{1em} d = d + 1;
\text{ENDWHILE}
\]
\[
d = x - y; \\
e = d \times x; \\
\text{sub}(x, y); \\
\ldots
\]

PROCEDURE sub (m,n)
\[
\begin{align*}
p &= m + n; \\
q &= m \times n; \\
\end{align*}
\]
END sub