Application of ESL Synthesis on GSM Edge Algorithm for Base Station

Alan Su
Global Unichip Corp.
Multi-Core Designing

Data → AP1 → AP2 → Display

Task Graph

AP1

ARM Core

OS

Interrupt

AP2

DSP Core

ISR

AHB0

USB

ICM

RAM

AHB1

APB

LCD Display
ESL Synthesis Compartments

High Level Synthesis

Architecture Synthesis (AS)

Embedded Software Compiler

SW/SW Partitioning

SW/HW Partitioning

Two Level ESL Synthesis

Multi-Sampling Rate

Implementation I: down and up sampling

Implementation II: Loop Unrolling
Architecture Synthesis (AS)
**Task Graph Modeling in Petri Net**

*Petri Net* \( C = (P, T, I, O) \)

- **\( P \):** The set of places where each task is a place
- **\( T \):** The set of transitions. Each data flow edge between two tasks is denoted as a transition. Data rates at both sides of a data flow edge are represented by input and output functions of the transition.
- **\( I \):** The set of input functions
- **\( O \):** The set of output functions

**Example Task Graph**

\[
C = (P, T, I, O) \]

\[
P = \{t_1, t_2\} \]

\[
T = \{e_1\} \]

\[
I(e_1) = \{t_1, t_1, t_1\} \]

\[
O(e_1) = \{t_2, t_2\} \]
AS Resource Allocation

- Genetic Algorithm
  - Gene: an implementation of a task
    - Gene pool: the set of implementations of a given task, e.g. SW and HW implementation of FIR
  - DNA: the assemblage of genes (tasks)
    - DNA assemblage must obey design rules, e.g. a SW code must go with at least a processor, memory and bus.
  - Chromosome: the complete architecture assembled by DNA’s
    - Chromosome assemblage must again obey design rules

- Spatial Design Space Exploration
AS Scheduler

- Use various scheduling algorithms
  - ASAP
  - ALAP
  - ASAP list scheduling with increasing order
  - ASAP list scheduling with decreasing order
  - ALAP list scheduling with increasing order
  - ALAP list scheduling with decreasing order
- More scheduling algorithms can be used
- Temporal Design Space Exploration
User Defined Cost Function

- Hard to find universal cost function
- User defined cost function

\[ \text{Cost} (S) = m ( x_1 \cdot x_2 \cdot x_3 \cdot \ldots \cdot x_{n-1} \cdot x_n ) \]

where:

- \( S \) is the target system
- \( m \in \{ \text{Max, min} \} \), where Max is the maximum function and min the minimum function.
- \( x_i \in \{ p, 1/p \} \), 1 \( \leq i \leq n \), where \( p \) is a numerical cost property in the component property list
- \( \circ \in \{ +, -, \times, \div \} \), the set of addition, subtraction, multiplication and division operations
Convergence Condition

- Provide a universal and warranted convergence condition for all cost functions
- Observe the cost standard deviation of the architecture pool of each generation. GA converges if a given epsilon is maintained for a given number of generations.
- Experiment shows convergence with epsilon $\geq 0.0001$ in 5 minutes
High Level Synthesis

- It is difficult to characterize the hardware implementation of a new function unit.
- HLS is used as a characterization tool providing early, fast and accurate characterizations.
Synthesizing GSM Edge Algorithm

- Input task graph contains 100+ multi-rate tasks, in 10+ sub-graphs
- Reference implementation employs dual TI 54xx
- Synthesized implementations used single TI 54xx with 50% loading only
- By putting processor loading as a constraint, in an experiment AS can synthesized dual TI 54XX with 25% and 14% loading, respectively. The synthesized implementation closely resembled the reference implementation
- With an epsilon of 0.0001 GA converged in 5 minutes on a Pentium III 800MHz PC in ~30 generations
Future Work

- Software Synthesis
  - SW/SW Partitioning
  - Embedded software compilation
- Buffer Synthesis
- Cache Synthesis
- DMA Synthesis
- Language Representation for Multi Sampling Rate
Q & A