Course Assignments

(DTU #02917 “Advanced Topics in Embedded Systems”)

Martin Fränzle, Andreas Eggers

Idea

- Intensified work on some aspects of the course.
- Two options:
  - Application of a verification tool to a specific domain.
  - Writing a verification backend.
- You choose one of these options and complete working on it in the remaining time of the three week period.
- Working in groups of approximately 2-3 people is appreciated.

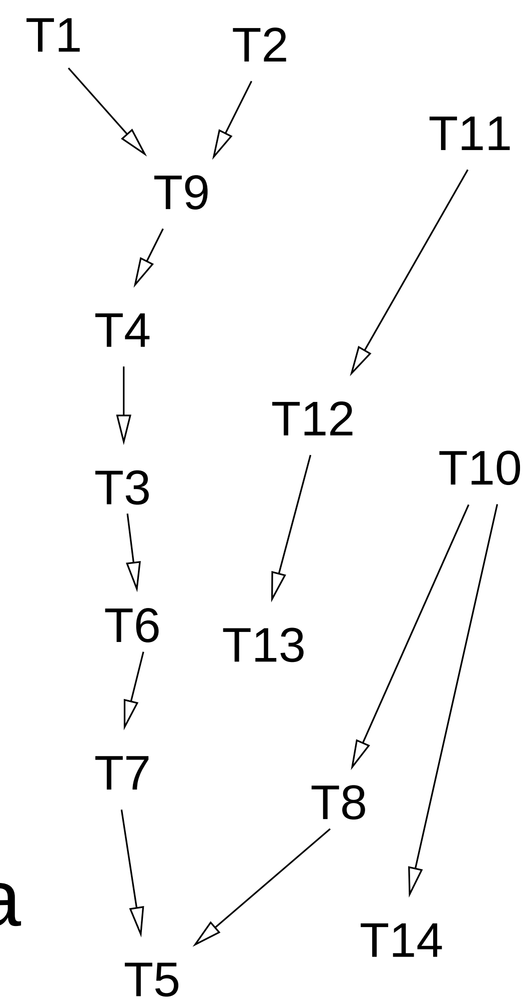
Option I

Option I: Application

- Domain: Allocation of tasks to architectures of embedded control units (ECUs).
- Task: Write a tool that converts problem descriptions into constraint systems that can be solved with HySAT.
- Expected output of the tool: Allocation mapping and static schedule of the tasks.

Tasks

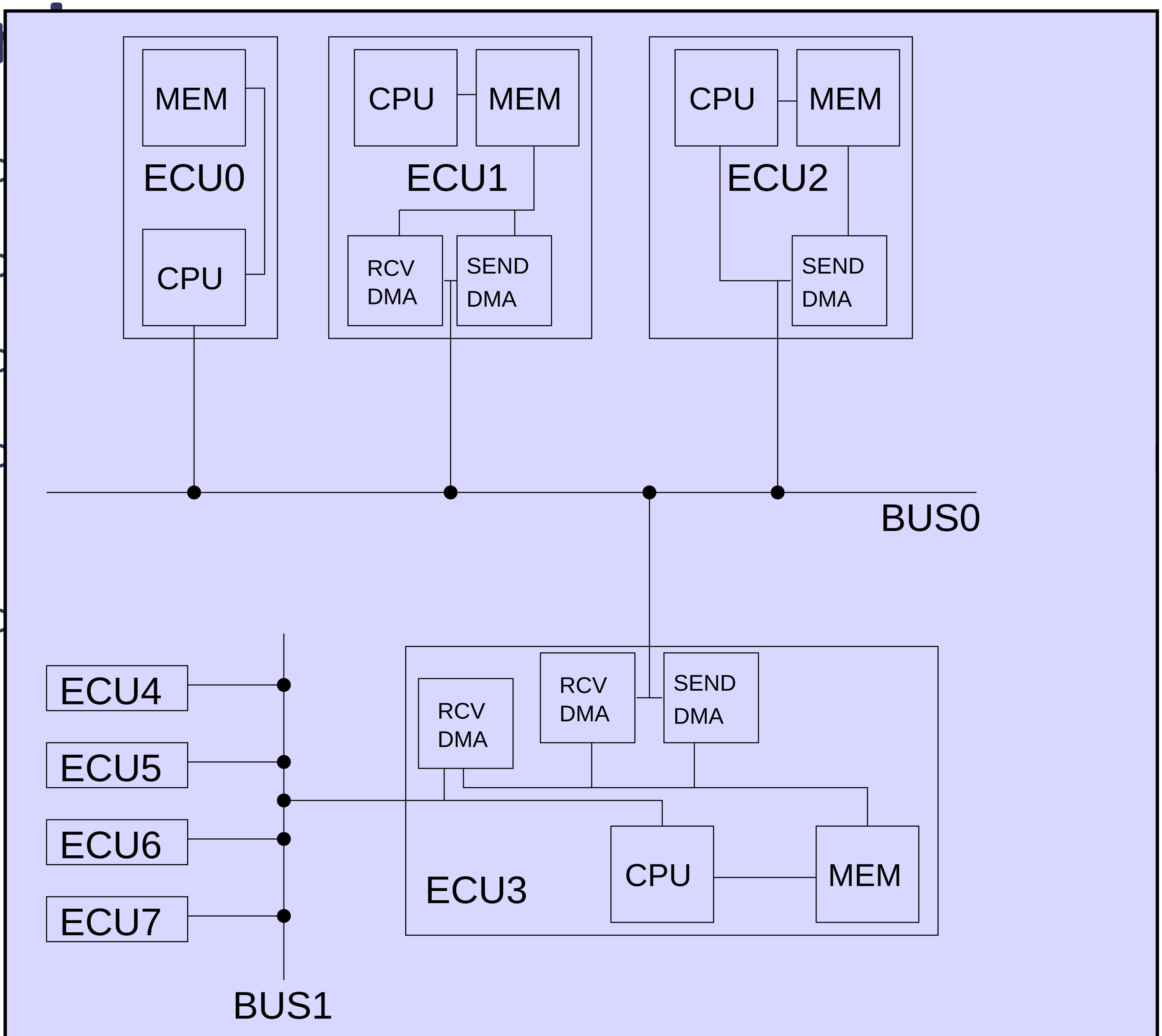
- Executed only once
- Characterized by:
  - Start time (ready)
  - Execution time (depending on ECU)
  - Deadline
- Interdependency between tasks $\Rightarrow$ Communication
- Constraints on which ECU a task can run



Hardware

- Single threaded ECUs
- Busses connecting ECUs
- Different speeds of busses and of ECUs
- Some ECUs can do bus communication in the background
- No broadcasting / multicasting

Har


MEM
ECU0
CPU
CPU
MEM
ECU1
RCV DMA
SEND DMA
CPU
MEM
ECU2
SEND DMA
BUS0
ECU4
ECU5
ECU6
ECU7
RCV DMA
RCV DMA
SEND DMA
ECU3
CPU
MEM
BUS1

Scheduling

- No preemption of tasks, i.e. ECU is blocked until task has finished
- Similar: transmission of a message blocks bus until finished
- Goal: Schedule is to be generated statically by the solver

Assignment

Parse system description in given input format

Generate constraints from system description

- allocation of tasks to ECUs
- generation of a schedule
- no violation of given side conditions (e.g. communication over busses)

Run HySAT to get allocation and schedule

Assignment

Parse system description in given input format	Generate constraints from system description − allocation of tasks to ECUs − generation of a schedule − no violation of given side conditions (e.g. communication over busses)	Run HySAT to get allocation and schedule

If you want to reduce the complexity of the assignment: Drop some of the conditions, e.g. backgrounding of bus communication, existence of several busses. Depending on the amount of work you cut off, this will cost you some points in the fi nal grade.

Option II

(by Tino Teige)

Option II: Building a solver

- Deepen your understanding of the DPLL procedure
- Domain of the variables: bounded integers
- Constraints: Disjunctions of simple bounds $x \geq y + k$ with constant k
- Does there exist a valuation that satisfies the constraint system?

Example

- $X = \{x_0, x_1, x_2\}$ and $D_0 = [0, 0], D_1 = [1, 6], D_2 = [-2, 3]$.
- C contains the following simple constraints:

$$c_1 = (x_1 \geq x_0 + 5 \ \vee \ x_2 \geq x_1 + (-1)),$$
$$c_2 = (x_0 \geq x_2 + 2 \ \vee \ x_1 \geq x_2 + 4).$$

- CSP is **satisfiable**, since $x_0 = 0, x_1 = 6, x_2 = -2$ is a solution.

Basic backtrack algorithm (example)

Try to prove satisfi ability of

$$c_1 = (x_1 \geq x_0 + 5 \ \vee \ x_2 \geq x_1 + (-1)),$$

$$c_2 = (x_0 \geq x_2 + 2 \ \vee \ x_1 \geq x_2 + 4)$$

by manipulating **interval valuations**.

A simple bound, e.g., $x_1 \geq x_0 + 5$ can be

- **true**, e.g. $x_0 \in [0, 10], x_1 \in [20, 40]$

  $x_1 \in [\mathbf{20}, 40], x_0 + 5 \in [0, 10] + 5 = [5, \mathbf{15}]$

- **false**, e.g. $x_0 \in [0, 10], x_1 \in [-10, 0]$

  $x_1 \in [-10, \mathbf{0}], x_0 + 5 \in [0, 10] + 5 = [\mathbf{5}, 15]$

- **inconclusive**, e.g. $x_0 \in [0, 10], x_1 \in [-10, 10]$

  $x_1 \in [-10, \mathbf{10}], x_0 + 5 \in [0, 10] + 5 = [\mathbf{5}, \mathbf{15}]$

  Neither true nor false!

Basic backtrack algorithm (contd.)

$$c_1 = (x_1 \geq x_0 + 5 \;\vee\; x_2 \geq x_1 + (-1)),$$
$$c_2 = (x_0 \geq x_2 + 2 \;\vee\; x_1 \geq x_2 + 4)$$

Initially, $x_0 \in [0,0], x_1 \in [1,6], x_2 \in [-2,3]$, and all bounds are **inconclusive**.

Decision step: Split the search space by splitting an interval of a variable, e.g. interval $[1,6]$ of x_1 into $[1,3]$ and $[4,6]$.
Decide for $[1,3]$ and store $[4,6]$ as alternative for backtracking.

Basic backtrack algorithm (contd.)

$$c_1 = (x_1 \geq x_0 + 5 \;\vee\; x_2 \geq x_1 + (-1)),$$
$$c_2 = (x_0 \geq x_2 + 2 \;\vee\; x_1 \geq x_2 + 4)$$

Under new interval valuation
$x_0 \in [0, 0], x_1 \in [1, 3], x_2 \in [-2, 3]$
the bound $x_1 \geq x_0 + 5$ becomes **false**, since
3 $\geq x_1 \geq x_0 + 5 \geq$ **5**
does not hold.

Decision step: Split the interval $[-2, 3]$ of x_2 into $[-2, 0]$ and $[1, 3]$. Decide for $[1, 3]$ and store $[-2, 0]$ as alternative for backtracking.

Basic backtrack algorithm (contd.)

$$c_1 = (x_1 \geq x_0 + 5 \ \vee \ x_2 \geq x_1 + (-1)),$$
$$c_2 = (x_0 \geq x_2 + 2 \ \vee \ x_1 \geq x_2 + 4)$$

Under new interval valuation $x_0 \in [0, 0], x_1 \in [1, 3], x_2 \in [1, 3]$ the second constraint becomes **false**, since

$$\mathbf{0} \geq x_0 \geq x_2 + 2 \geq \mathbf{3},$$
$$\mathbf{3} \geq x_1 \geq x_2 + 4 \geq \mathbf{5}$$

do not hold. **Conflict!**

Basic backtrack algorithm (contd.)

Backtrack: Go back to the last decision (split on x_2), undo all changes up to there, and assert the alternative interval ($[-2, 0]$).

Proceed until

- all constraints are satisfied: CSP **satisfiable**, or
- no backtrack point exists: CSP **unsatisfiable**.

Extension

Enhance the basic version by **deduction**.

$$c_1 = (x_1 \geq x_0 + 5 \ \vee \ x_2 \geq x_1 + (-1))$$

Under interval valuation $x_0 \in [0, 0], x_1 \in [1, 3], x_2 \in [-2, 3]$ the bound $x_1 \geq x_0 + 5$ is false, while $x_2 \geq x_1 + (-1)$ is inconclusive.

$x_2 \geq x_1 + (-1)$ is the last satisfiable bound in c_1.
To make c_1 true, $x_2 \geq x_1 + (-1)$ has to be true.
Therefore we can potentially deduce new intervals from $x \geq x_1 + (-1)$.

Assignment

- Solving *Constraint Satisfaction Problems* (CSPs) by an extended DPLL algorithm
- Theoretical part:
  - soundness/ completeness,
  - a generalization, and
  - *(optionally)* possible acceleration techniques
- Implementation part:
  - a (simple) parser and
  - the basic and enriched algorithm
  - testing and comparison of both tools on benchmarks,