aiT WCET Analyzer computes **tight bounds** for the worst-case execution time of tasks in safety-critical systems. These bounds are **safe**, i.e. they are valid for any input scenario and each task execution.

aiT is based on statically analyzing a task’s intrinsic **cache and pipeline behavior**, thus enabling the development of complex hard real-time systems on state-of-the-art hardware.

### Application code
```c
void Task (void) {
  variable++;
  function();
  next++;
  if (next)
    do_this();
  terminate();
}
```

### Specifications (*.ais)
- `clock kHz; 10200` (loop + loop exactly end; )
- `_codebook` 1
- `_fac` 6
- `snippet` is not analyzed and takes max cycles;
- `printf` 333
- `flow + bytes / + bytes` is max;
- `U_MOD` "U_MOD" 0xAC 0xC4 4
- `area from` to `is readonly;` 0x20 0x497

### The Challenge:
- **Measuring** the execution time of a task is typically **not safe**. It is often impossible to prove that all the conditions determining maximum execution time are taken into account. Instrumentation and debug information change the timing behavior.

- Analysis methods that do not consider **cache and pipeline behavior** typically seriously overestimate the WCET.

- Switching off instruction and data caches to simplify WCET prediction can lead to severe performance degradation (by a factor of up to 30 for the PowerPC 604, according to a study by EADS).

### This is where aiT steps in:
- aiT-computed bounds are **valid for all inputs** and each execution of a task. Extensive timing testing is now a thing of the past.

- aiT directly analyzes binary executables. This means that **no modification of your tool chain** or the program's operational behavior and performance is required.

- aiT-computed bounds are tight and reflect the **real performance** of your system. Cache and pipeline effects are fully taken into account. Ensuring deadline adherence is no longer done at the expense of hardware resources.
Why do you need aIT?

The worst-case execution time of each task in a real-time system has to be known prior to its execution. In event-triggered or periodic systems (e.g. RMA) the WCET is required for schedulability analysis; in time-triggered systems (e.g. TTA, FlexRay, ...) it is required for determining a static schedule.

The increasing performance of microcontrollers enables more and more functionality to be implemented by a single embedded control unit. The software is complex and the timing behavior of the interacting software components rarely known. Typically it is not practical – or even possible – to test the system with all potential inputs.

aiT Features:

- **Visualization** of the call and control flow graph of the application. The illustration shows the critical path and the contribution of each function to the overall worst-case execution time. Developers can quickly identify those program parts relevant for optimizing worst-case timing behavior.

- Visualization of the machine states at different program points. Developers get an in-depth analysis of the reason of performance effects which provides valuable hints for **timing optimization**.

- Qualification Support Kits are available providing support for automatic **tool qualification** up the highest criticality levels (e.g. wrt. DO-178B, ISO-26262, IEC-61508, EN-50128).

- Support for **sophisticated hardware components**, like superscalar, out-of-order execution pipelines, branch prediction units, instruction and data caches, etc.

- **Flexible annotation mechanism.** Developers can provide programmer-specific knowledge to aIT to further improve the analysis precision.

- aIT can be coupled with **model-based code generators** and **system-level scheduling tools** via an open XML-based interface to provide timing information in the development phase.

- **Graphical comparison of different analysis runs.** Developers can quickly understand the effect of program modifications on worst-case timing.

- aIT can replace error-prone methods based on tests and measuring. Thus, **enhancing safety**.

- aIT has been qualified as a validation tool according to DO-178B, up to Level A. Thus, allowing to **certify safety-critical real-time software**.

- aIT provides automatic tool support for calculating the WCET of your applications. Thus, **saving development time**.

- aIT determines the timing behavior of interacting software components. Thus, **enabling software integration**.