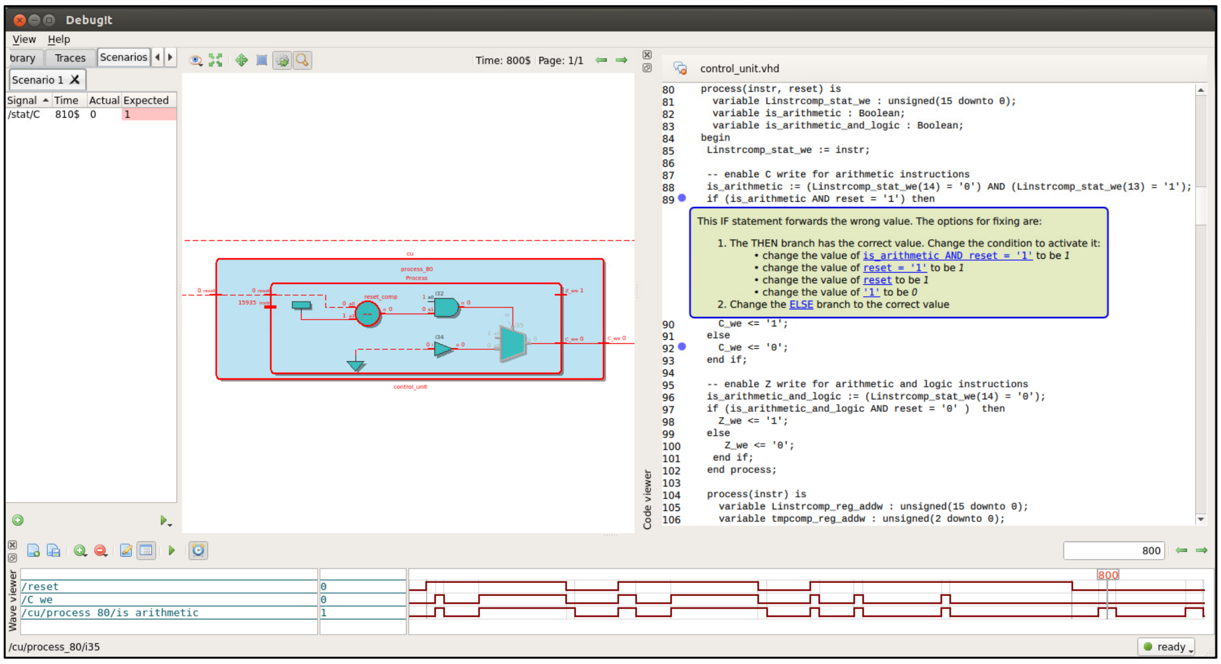
Debug!t - Automated Debugging for Chip Design

solvertec enables Register Transfer Level (RTL) designers to accelerate the process of functional verification of complex chip designs by orders of magnitude through automatically detecting the sources of design errors and providing a seamless approach of fixing them, thus improving the predictability of the design schedule.

Debug!t picks up where verification tools leave off. *Debug!t* performs full automated root cause analysis of the HDL code and a failing trace. Just load a design into *Debug!t* and import a log file that contains the failing and expected behavior to connect the verification output with *Debug!t*. By simply clicking the *Debug!t* button, *Debug!t* computes the sources of a design error (bug locations) and thereby reduces the input-cone to only those HDL statements that are relevant for the bug fix. At the same time, for each bug location an explanation is given together with hints how to fix the bug.

Highlights:

- Pinpoints sources of design errors in RTL
- Gives textual and visual explanations
- Provides hints to fix
- Integrates seamlessly into existing flows
- Increases productivity
- Eliminates hard-to-fix bug uncertainties
- Secures schedule predictability
- Accelerates time-to-market



Signal	Time	Actual	Expected
/stat/C	810$	0	1

```
process(instr, reset) is
  variable Linstrcomp_stat_we : unsigned(15 downto 0);
  variable is_arithmetic : Boolean;
  variable is_arithmetic_and_logic : Boolean;
begin
 Linstrcomp_stat_we := instr;

 -- enable C write for arithmetic instructions
 is_arithmetic := (Linstrcomp_stat_we(14) = '0') AND (Linstrcomp_stat_we(13) = '1');
 if (is_arithmetic AND reset = '1') then
```

This IF statement forwards the wrong value. The options for fixing are:

1. The THEN branch has the correct value. Change the condition to activate it:
   - change the value of is_arithmetic AND reset = '1' to be *1*
   - change the value of reset = '1' to be *1*
   - change the value of reset to be *1*
   - change the value of '1' to be *0*
2. Change the ELSE branch to the correct value

```
   C_we <= '1';
 else
   C_we <= '0';
 end if;

 -- enable Z write for arithmetic and logic instructions
 is_arithmetic_and_logic := (Linstrcomp_stat_we(14) = '0');
 if (is_arithmetic_and_logic AND reset = '0' )  then
   Z_we <= '1';
 else
    Z_we <= '0';
  end if;
 end process;

 process(instr) is
   variable Linstrcomp_reg_addw : unsigned(15 downto 0);
   variable tmpcomp_reg_addw : unsigned(2 downto 0);
```

```
/reset                      0
/C we                       0
/cu/process 80/is arithmetic 1
```

Debug!t: Automated RTL design error root cause analysis that seamlessly fits into existing verification flows

Contact:
Embedded World 2014: Hall 4, Booth 4-614
Daniel Große | solvertec GmbH | Anne-Conway-Str. 1 | 28359 Bremen
Phone: +49 421 40 89 84 51 | E-mail: grosse@solvertec.de | Web: www.solvertec.de | Twitter: @solvertecEDA