Realize Your Product Promise®

Revisiting coverage criteria for SCADE models

Jean-Louis Colaço

7 December 2016

Context

- *Code coverage* is a measure that characterises how much a given *test suite* exercises a *code*,
- lots of criteria exist, avioncs standard (DO-178) requires MC/DC for the most critical application,
- in DO-178C (2011), suplement DO-331 about *Model Based Design* now requires model coverage.
- SCADE proposes model coverage for about 10 years:
  - was based on ad'hoc criteria defined by the user per operator,
  - recent solution is inspired by work of Parissis et al.

A. Lakehal and I. Parissis,
Structural coverage criteria for LUSTRE/SCADE programs,
in *Software Testing, Verification and Reliablity*, Wiley Interscience, 2009

J-L. Camus, C. Haudebourg and M. Schlickling
Data Flow Model Coverage Analysis: Principles and Practice
in *Embedded Real Time Software and Systems*, 2016

 © ANSYS, Inc.

Why revisiting?

- current solution is based on *Paths* in the dataflow: quite complex objects;
- to study the relationship between model coverage and generated code coverage: paths are not well suited;
- to overcome some limitation of current implementation.

© ANSYS, Inc.

ANSYS

Why revisiting?

ESTEREL Technologies

- current solution is based on *Paths* in the dataflow: quite complex objects;
- to study the relationship between model coverage and generated code coverage: paths are not well suited;
- to overcome some limitation of current implementation.

The idea we had for the rework was actually nicely presented in:

M. Whalen, G. Gay, Y. Dongjiang, M. P.E. Heimdahl and M. Staats
Observable modified condition/decision coverage
in *Proceedings of the 35th International Conference on Software Engineering*, 2013

 © ANSYS, Inc.

Why revisiting?

- current solution is based on *Paths* in the dataflow: quite complex objects;
- to study the relationship between model coverage and generated code coverage: paths are not well suited;
- to overcome some limitation of current implementation.

The idea we had for the rework was actually nicely presented in:

M. Whalen, G. Gay, Y. Dongjiang, M. P.E. Heimdahl and M. Staats
Observable modified condition/decision coverage
in *Proceedings of the 35th International Conference on Software Engineering*, 2013

present work continues and extends it to full SCADE 6 language.

© ANSYS, Inc.

Agenda

Intuition

Ideal definition of coverage

SCADE tagged semantics

Tag based definition of coverage

Static tag reduction

Conclusion

© ANSYS, Inc.

Intuition

Ideal definition of coverage

SCADE tagged semantics

Tag based definition of coverage

Static tag reduction

Conclusion

 © ANSYS, Inc.

Glossary

flow *or* **stream**: infinite sequence of values.

model: a SCADE program and a *root node*.

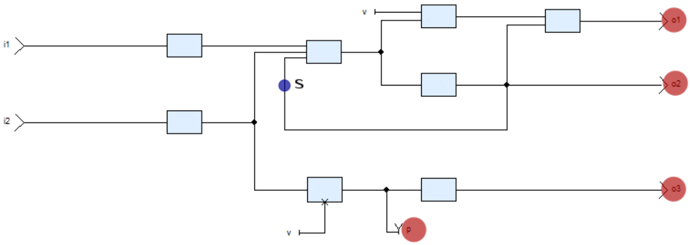
monitor: any construction that allows to observe a flow out of the model: (root node) outputs, probes, . . .

outcome (of a test) values taken by all the monitors of the model when running a test.

source designates any construction that introduces flow that that does not result from the combination of other flows. (root node) inputs, sensors, literal values, reference to constants.

© ANSYS, Inc.

The intuition



- Covering a *stream occurrence s* requires exhibiting a test that shows its ability to *influence* a monitor (red bubles);
- Covering a model requires covering all its *streams occurrences*.

 © ANSYS, Inc.

Criterion 1: Influence

A test T shows the influence of stream x of a model $\mathcal{M}$ if:

- T is such that x is in situation to influence an output of $\mathcal{M}$
- i.e. T is such that modifying stream x in the execution of the test changes the outcome.

A test suite $\mathcal{T}_S$ covers a model $\mathcal{M}$ if for all stream x of $\mathcal{M}$, $\mathcal{T}_S$ contains a test T that covers stream x.

© ANSYS, Inc.

Criterion 2: OMC/DC

A pair of tests (T_1, T_2) satisfies OMC/DC criterion for a Boolean stream b of a model $\mathcal{M}$ if T_1 and T_2 are such that:

- b takes different values in each test case and
- toggling b in both test cases changes the outcome.

A test suite $\mathcal{T}_S$ covers a model $\mathcal{M}$ in the sense of OMC/DC if for all Boolean stream b of $\mathcal{M}$, $\mathcal{T}_S$ contains two tests T_1 and T_2 such that satisfy the condition above.

© ANSYS, Inc.

Intuition

Ideal definition of coverage

SCADE tagged semantics

Tag based definition of coverage

Static tag reduction

Conclusion

 © ANSYS, Inc.

Notations

- $\mathcal{D}^n$ represent the set of stream prefix of size smaller or equal to n.
- If x is a stream prefix, $|x|$ represents its size.
- If x is a stream prefix, $(x)_i$ where $i \leq |x|$ represents i^{th} value.
- Let $\mathcal{M}$ be a SCADE model and n_{in} its number of inputs.
- A test case T of length n cycle is a tuple of n_{in} components of $\mathcal{D}^n$.
- $\mathcal{M}(T)$ represents the execution of test case T; the outcome of this execution is itself a tuple of values in $\mathcal{D}^n$ (one per monitor).
- If v is a stream prefix of a Boolean stream, $\neg_i(v)$ represents the prefix with same length built from v by negating its i^{th} value.
- A stream occurrence is represented as $\lfloor e \rfloor_{\mathbf{k}}$ where k is an integer and e is a stream expression.

© ANSYS, Inc.

Occurrences identification

Defined by function *Streams* (.):

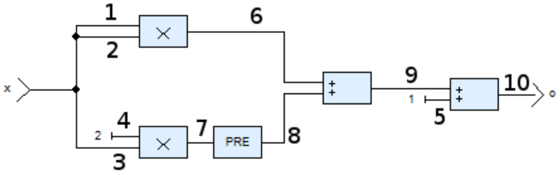
$$
\begin{array}{ll}
\mathit{Streams}\,(\texttt{x1},\ \texttt{...},\ \texttt{xn}\ \texttt{=}\ e\texttt{;}) & \stackrel{\text{def}}{=} \mathit{Streams}\,(e) \\
\ldots & \stackrel{\text{def}}{=} \ldots \\
\mathit{Streams}\,(\texttt{x}) & \stackrel{\text{def}}{=} \{\lfloor \texttt{x} \rfloor_{\mathbf{k}}\} \\
\mathit{Streams}\,(\texttt{1}) & \stackrel{\text{def}}{=} \{\lfloor \texttt{1} \rfloor_{\mathbf{k}}\} \\
\mathit{Streams}\,(\texttt{'s;}) & \stackrel{\text{def}}{=} \{\lfloor \texttt{'s} \rfloor_{\mathbf{k}}\} \\
\mathit{Streams}\,(\textbf{last}\ \ \texttt{'s;}) & \stackrel{\text{def}}{=} \{\lfloor \textbf{last}\ \ \texttt{'s} \rfloor_{\mathbf{k}}\} \\
\mathit{Streams}\,(\mathrm{op}(e_1, \ldots, e_n)) & \stackrel{\text{def}}{=} \{\lfloor \mathrm{op}(e_1, \ldots, e_n) \rfloor_{\mathbf{k}}\} \ \bigcup\ \mathit{Streams}\,(e_1)\ \bigcup\ \ldots \\
\ldots & \stackrel{\text{def}}{=} \ldots
\end{array}
$$

 © ANSYS, Inc.

Occurrences identification example

$$\mathit{Streams}\,(\texttt{o = x*x + }\mathbf{pre}\texttt{ (2*x) + 1;}) =$$

$$\left\{ \begin{array}{l}
\lfloor x \rfloor_{1},\quad \lfloor x \rfloor_{2},\quad \lfloor x \rfloor_{3},\quad \lfloor 2 \rfloor_{4},\quad \lfloor 1 \rfloor_{5},\\
\lfloor \lfloor x \rfloor_{1} * \lfloor x \rfloor_{2} \rfloor_{6},\quad \lfloor \lfloor 2 \rfloor_{4} * \lfloor x \rfloor_{3} \rfloor_{7},\quad \lfloor \lfloor \mathbf{pre}(\lfloor 2 \rfloor_{4} * \lfloor x \rfloor_{3}) \rfloor_{7} \rfloor_{8},\\
\lfloor \lfloor \lfloor x \rfloor_{1} * \lfloor x \rfloor_{2} \rfloor_{6} + \lfloor (\mathbf{pre}\, \lfloor \lfloor 2 \rfloor_{4} * \lfloor x \rfloor_{3} \rfloor_{7} \rfloor_{8} \rfloor_{9},\\
\left\lfloor \lfloor \lfloor \lfloor x \rfloor_{1} * \lfloor x \rfloor_{2} \rfloor_{6} + \lfloor (\mathbf{pre}\, \lfloor \lfloor 2 \rfloor_{4} * \lfloor x \rfloor_{3} \rfloor_{7} \rfloor_{8} \rfloor_{9} + \lfloor 1 \rfloor_{5} \right\rfloor_{10}
\end{array} \right\}$$



© ANSYS, Inc.

Stream occurrence mutation

Let $\mathcal{M}$ be a model where:

- $\lfloor e \rfloor_{\mathbf{k}}$ one of its stream occurrences: $\lfloor e \rfloor_{\mathbf{k}} \in \mathit{Streams}\,(\mathcal{M})$,
- v is a finite stream prefixe: $v \in \mathcal{D}^n$,
- e and v are of same type,
- e' is a stream expression with same clock as e:

e	e_0	$\cdots$	e_n	e_{n+1}	e_{n+2}	$\cdots$
v	v_0	$\cdots$	v_n			
e'	v_0	$\cdots$	v_n	e_{n+1}	e_{n+2}	$\cdots$

$\mathcal{M}^{(v \blacktriangleright \lfloor e \rfloor_{\mathbf{k}})}$ represents the model obtained by substituting $\lfloor e \rfloor_{\mathbf{k}}$ in $\mathcal{M}$ by a e'; we called it a *mutant* of $\mathcal{M}$ for the occurrence $\lfloor e \rfloor_{\mathbf{k}}$.

 © ANSYS, Inc.

Influence ideal definition

Coverage of stream x by T:

$$\mathit{Influence}(T, x, \mathcal{M}) \stackrel{\text{def}}{=} \exists n > 0.\ \exists v \in \mathcal{D}^n.\ \ \mathcal{M}(T) \neq \mathcal{M}^{(v \blacktriangleright x)}(T)$$

Coverage of model $\mathcal{M}$ by a test suite $\mathcal{T}_S$:

$$\forall x \in \mathit{Streams}\,(\mathcal{M})\,.\ \exists T \in \mathcal{T}_S.\ \mathit{Influence}(T, x, \mathcal{M})$$

© ANSYS, Inc.

OMC/DC Ideal definition

Coverage of stream x by (T_1, T_2):

$$Omcdc(T_1, T_2, b, \mathcal{M}) \stackrel{\text{def}}{=}$$

$$\exists (i, j) \in \mathbb{N} \times \mathbb{N}. \left(\begin{array}{ll} & (b_{T_1})_i \neq (b_{T_2})_j \\ \bigwedge & \mathcal{M}(T_1) \neq \mathcal{M}^{(\neg_i (b_{T_1}) \blacktriangleright b)}(T_1) \\ \bigwedge & \mathcal{M}(T_2) \neq \mathcal{M}^{(\neg_j (b_{T_2}) \blacktriangleright b)}(T_2) \end{array} \right)$$

Coverage of model $\mathcal{M}$ by a test suite $\mathcal{T}_S$:

$$\forall b \in Streams(\mathcal{M}) .$$
$$\exists (T_1, T_2) \in \mathcal{T}_S \times \mathcal{T}_S. \big((b : \text{bool}) \Rightarrow Omcdc(T_1, T_2, b, \mathcal{M})\big)$$

© ANSYS, Inc.

Limit of the ideal definition

Not really implementable:

- based on the exitence of mutants without giving a way to build them (it is a guess);
- requires both executions on original model and on the mutant;
- needs one mutant per stream occurrence.

© ANSYS, Inc.

Intuition

Ideal definition of coverage

SCADE tagged semantics

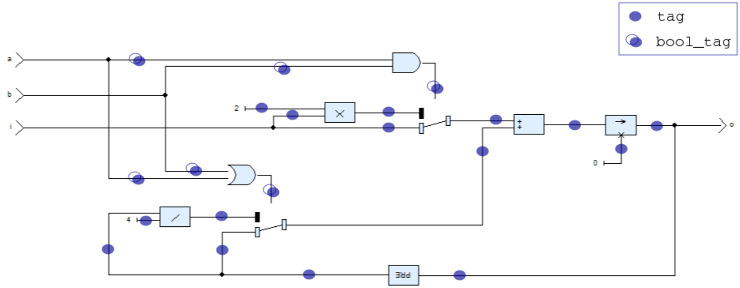
Tag based definition of coverage

Static tag reduction

Conclusion

 © ANSYS, Inc.

Tagged semantics



Tagged semantics:

- based on tagged values;
- defines tag propagation rules.
- provides primitives for tag introduction;

 © ANSYS, Inc.

Tagged values

The values used in a tagged SCADE model $\mathcal{M}^{\#}$ are in $\mathcal{V}^{\#}_{n,m}$ defined by:

$$
\begin{aligned}
\mathcal{V}^{\#}_{0,m} &\stackrel{\text{def}}{=} (\mathbf{bool} \bigcup \mathbf{numeric} \bigcup \{\text{declared enum values}\}) \times \mathcal{P}(\mathit{Tags}) \\
\mathcal{V}^{\#}_{n+1,m} &\stackrel{\text{def}}{=} \mathcal{V}^{\#}_{n,m} \\
&\bigcup \left\{ [v^{\#}_1, \ldots, v^{\#}_p] \;\middle|\; 1 \leq i \leq p \leq m,\;\; v^{\#}_i \in \mathcal{V}^{\#}_{n,m} \right\} \times \mathcal{P}(\mathit{Tags}) \\
&\bigcup \left\{ \{l_1 : v^{\#}_1, \ldots, l_p : v^{\#}_p\} \;\middle|\; 1 \leq i \leq p \leq m,\;\; v^{\#}_i \in \mathcal{V}^{\#}_{n,m} \right\} \times \mathcal{P}(\mathit{Tags})
\end{aligned}
$$

where *Tags* is a finite set of tags

© ANSYS, Inc.

Tag propagation of combinatorial operators

For most operators input tags propagate to the outputs:

$$\mathtt{op}^{\#}((v_1, \tau_1), \ldots, (v_n, \tau_n)) = (\mathtt{op}(v_1, \ldots, v_n), \bigcup_{i \in [1..n]} \tau_i)$$

Tag propagation of temporal operators

Behave as usual but on tagged streams:

(a, τ^a)	(a_0, τ^a_0)	(a_1, τ^a_1)	(a_2, τ^a_2)	(a_3, τ^a_3)	$\cdots$
(b, τ^b)	(b_0, τ^b_0)	(b_1, τ^b_1)	(b_2, τ^b_2)	(b_3, τ^b_3)	$\cdots$
pre$^\#$ (a, τ^a)	$(nil, \emptyset)$	(a_0, τ^a_0)	(a_1, τ^a_1)	(a_2, τ^a_2)	$\cdots$
(a, τ^a) ->$^\#$ (b, τ^b)	(a_0, τ^a_0)	(b_1, τ^b_1)	(b_2, τ^b_2)	(b_3, τ^b_3)	$\cdots$

© ANSYS, Inc.

Specific propagation rules

$\textbf{and}^{\#}$ (also exists for $\textbf{or}^{\#}$):

a	b	$a\ \textbf{and}^{\#}\ b$
$\textbf{false}, \tau_a$	$\textbf{false}, \tau_b$	$\textbf{false}, \tau_a \cap \tau_b$
$\textbf{false}, \tau_a$	$\textbf{true}, \tau_b$	$\textbf{false}, \tau_a$
$\textbf{true}, \tau_a$	$\textbf{false}, \tau_b$	$\textbf{false}, \tau_b$
$\textbf{true}, \tau_a$	$\textbf{true}, \tau_b$	$\textbf{true}, \tau_a \cup \tau_b$

flow selection:

$$\textbf{if}^{\#}\ (\textbf{true}, \tau_c)\ \textbf{then}^{\#}\ (v_1, \tau_1)\ \textbf{else}^{\#}\ (v_2, \tau_2) = (v_1, \tau_c \cup \tau_1)$$

$$\textbf{if}^{\#}\ (\textbf{false}, \tau_c)\ \textbf{then}^{\#}\ (v_1, \tau_1)\ \textbf{else}^{\#}\ (v_2, \tau_2) = (v_2, \tau_c \cup \tau_2)$$

© ANSYS, Inc.

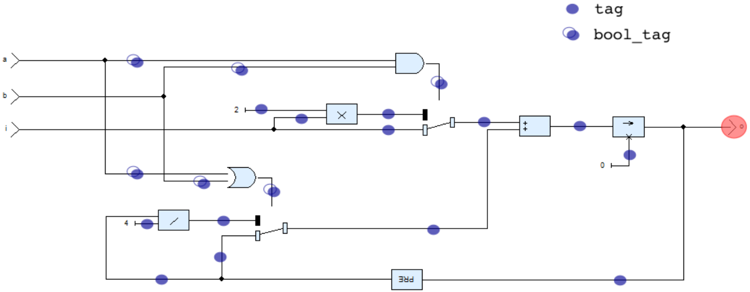
Tags introduction

- sources are extended with an empty set of tags,
- memories are initialy extended with an empty set of tags,
- new primitives $\texttt{tag}(e, t)$ and $\texttt{bool\_tag}(e, t_1, t_2)$ introduce tags:

$$
\begin{aligned}
\texttt{tag}((v, \tau), t) &= (v, \{t\} \cup \tau) \\
\texttt{bool\_tag}((\textbf{true}, \tau), t_1, t_2) &= (\textbf{true}, \{t_1\} \cup \tau) \\
\texttt{bool\_tag}((\textbf{false}, \tau), t_1, t_2) &= (\textbf{false}, \{t_2\} \cup \tau)
\end{aligned}
$$

© ANSYS, Inc.

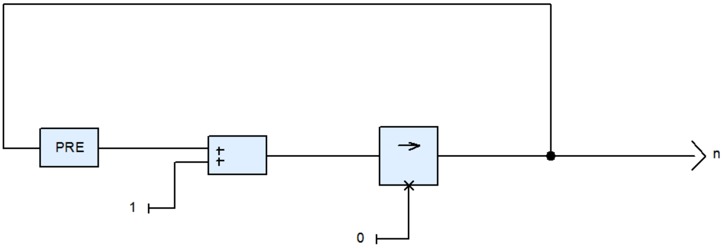
Tagged semantics for coverage purpose



- introduce a tag for each stream occurrence and
- register tags when reaching a monitor.

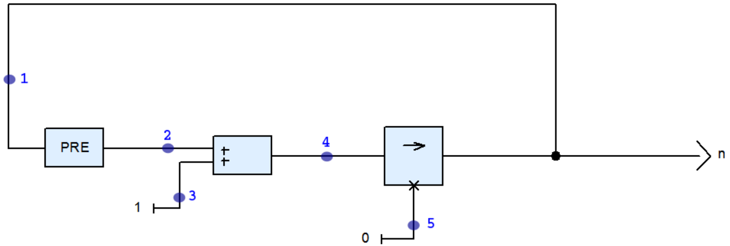
A simple example of propagation

model



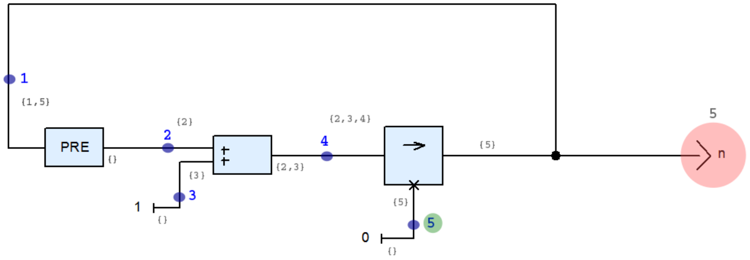
A simple example of propagation

tagged model



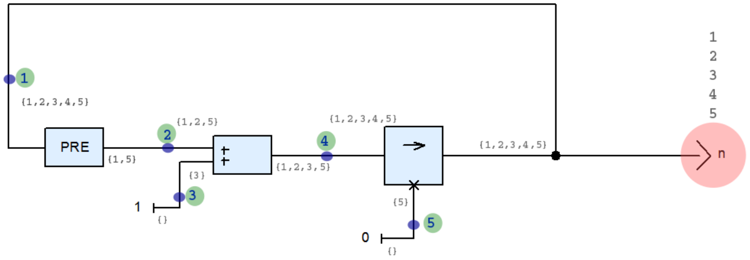
A simple example of propagation

first cycle



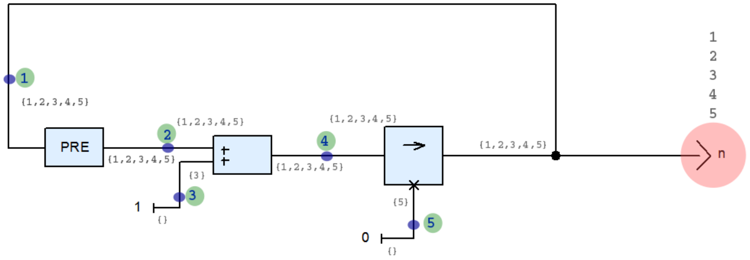
A simple example of propagation

second cycle



A simple example of propagation

other cycles



 © ANSYS, Inc.

Intuition

Ideal definition of coverage

SCADE tagged semantics

Tag based definition of coverage

Static tag reduction

Conclusion

© ANSYS, Inc.

Influence tagged definition

Coverage of stream x by T:

$$Influence^{\#}(T, x, \mathcal{M}) \stackrel{\text{def}}{=} t_x \in Otags(\mathcal{M}^{\#}(T))$$

Coverage of model $\mathcal{M}$ by $\mathcal{T}_S$:

$$\forall x \in Streams\,(\mathcal{M})\,.\,\exists T \in \mathcal{T}_S.\ Influence^{\#}(T, x, \mathcal{M})$$

© ANSYS, Inc.

OMC/DC tagged definition

Coverage of stream x by (T_1, T_2):

$$Omcdc^{\#}(T_1, T_2, b, \mathcal{M}) \stackrel{\text{def}}{=} t_b^{\circ} \in Otags(\mathcal{M}^{\#}_{Bool}(T_1)) \bigwedge t_b^{\bullet} \in Otags(\mathcal{M}^{\#}_{Bool}(T_2))$$

Coverage of model $\mathcal{M}$ by $\mathcal{T}_S$:

$$\forall b \in Streams\,(\mathcal{M})\,.\ \exists (T_1, T_2) \in \mathcal{T}_S \times \mathcal{T}_S.\ \big((b : \mathrm{bool}) \Rightarrow Omcdc^{\#}(T_1, T_2, b, \mathcal{M})\big)$$

 © ANSYS, Inc.

Gap with ideal definition

There are situations where tags are propagated while no contribution can be observed:

- absorption: `x * 0`
- unobservable selection: **if** `c` **then** `x` **else** `x`

© ANSYS, Inc.

Gap with ideal definition

There are situations where tags are propagated while no contribution can be observed:

- absorption: `x * 0`
- unobservable selection: **if** `c` **then** `x` **else** `x`

Gaps exist but it still be a good compromise.

© ANSYS, Inc.

Intuition

Ideal definition of coverage

SCADE tagged semantics

Tag based definition of coverage

Static tag reduction

Conclusion

 © ANSYS, Inc.

Reduction

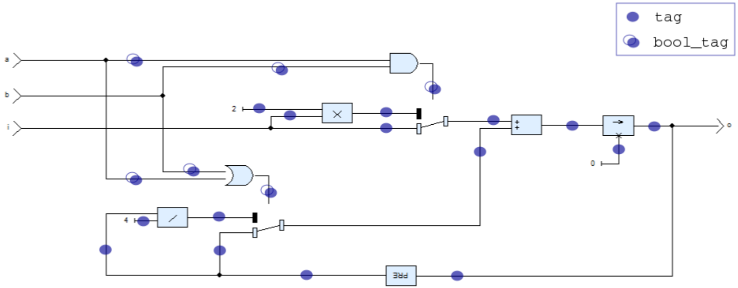
- Criteria are based on tags on all the expressions and sub-expressions $\Rightarrow$ **big number of tags**.
- Many tags are related: *each time* t_1 *is observed* t_2 *is also observed*.
- Reduction concists in removing tags whose observation can be deduced from other tags observation.
- Reduction is used in the model instrumentation phase.

© ANSYS, Inc.

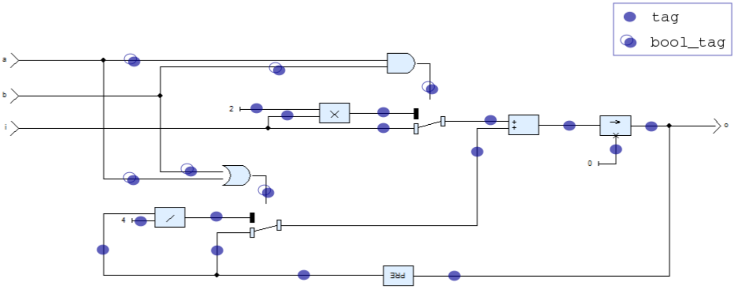
Example

```
node N(a, b : bool; i : int16)
  returns (o : int16)
var m : int16;
let
  m = pre o;
  o = 0 -> (if a and b then 2 * i else i)
           + (if a or b then m / 4 else m);
tel
```

Example: initial tagging

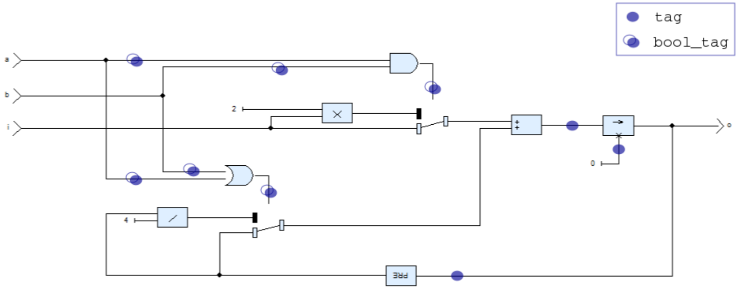


Example: initial tagging

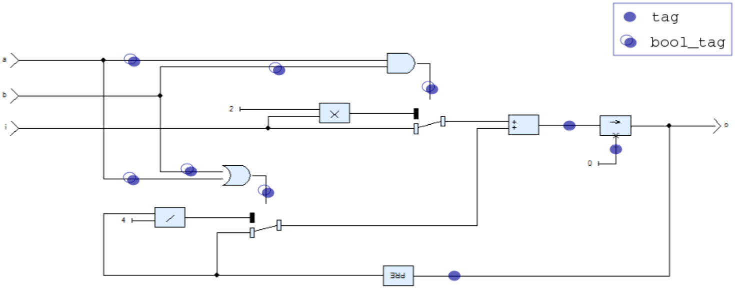


27 tags

Example: simple tag reduction

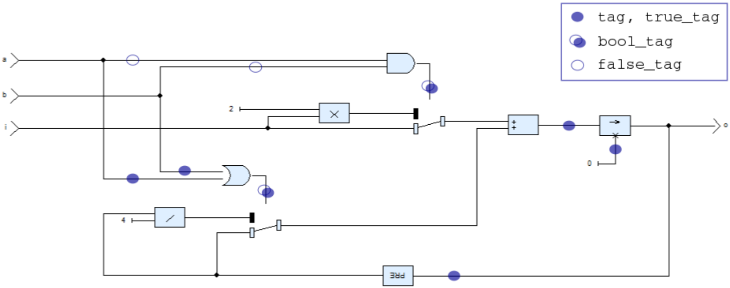


Example: simple tag reduction



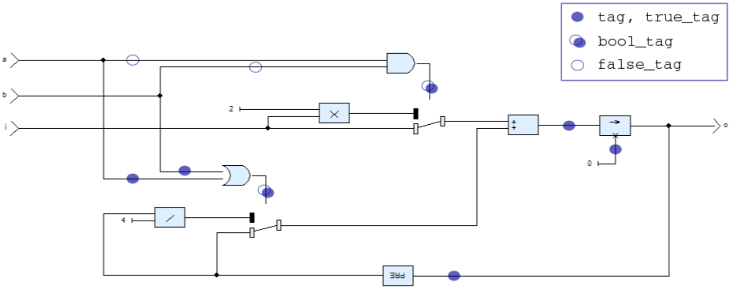
15 tags

Example: + Boolean reduction



 © ANSYS, Inc.

Example: + Boolean reduction



11 tags

Intuition

Ideal definition of coverage

SCADE tagged semantics

Tag based definition of coverage

Static tag reduction

Conclusion

© ANSYS, Inc.

Conclusion

- extends to all SCADE 6 language, including automata;
- implementation:
  - **instrumentation** of the model (addition of `tag(...)`) and
  - **code generation** for the tagged semantics;
- static reduction is important, divides by 2 to 3 the number of tags;
- good scale up (tested on big industrial models).

© ANSYS, Inc.