Towards a Formal Semantics of Verilog using Duration Calculus

Gerardo Schneider
VERIMAG, France

Xu Qiwen
International Institute for Software Technology
United Nations University (UNU/IIST)

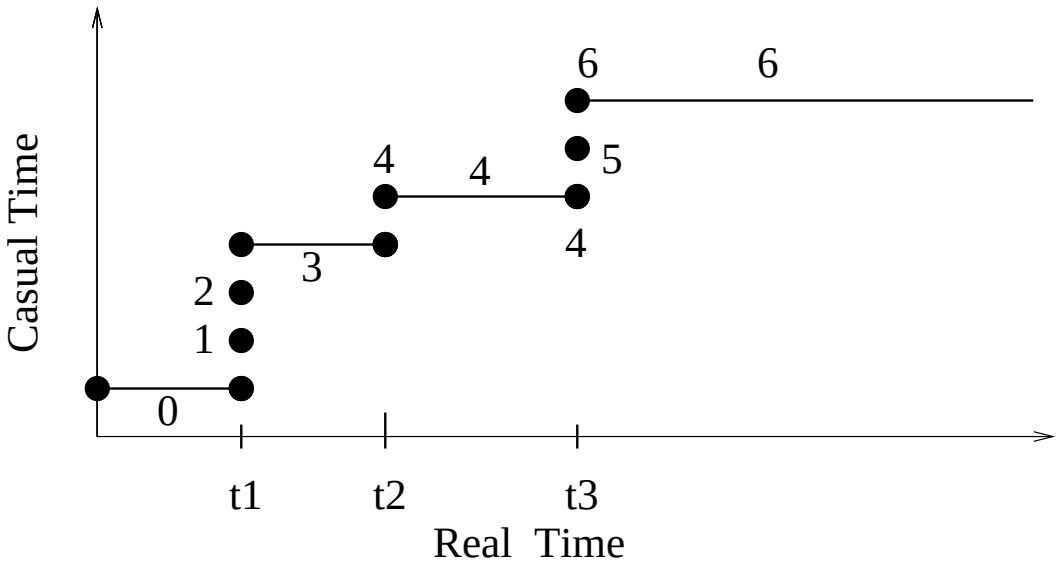
* Thanks: UNU/IIST

Overview of the presentation

- μWDCI
- Verilog
- Example
- Semantics of some commands
- Conclusions

Duration Calculus of Weakly Monotonic Time (WDC)

System evolves by Discrete steps
Several steps at the same time



Stepped Time Point (t, i)

t real time point (or macro time)

i causal time point (or micro time)

μWDCI: WDC with Infinite Intervals and Least Fixpoint Operator

Verilog

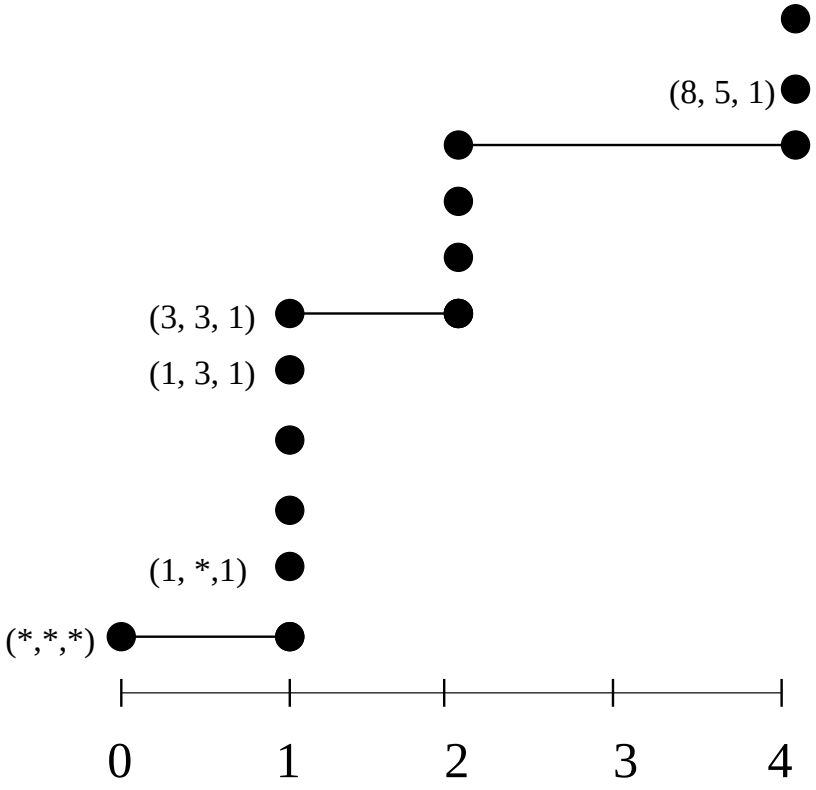
It's a HDL that has (among others):

- shared states, updated by possibly instantaneous assignments from different processes;
- zero-time computation
- continuous assignments
- delay statements;
- synchronisation by waiting for some conditions to become true;
- recursion (finite and infinite).

Example

P_1 : #1;
$x = 1$;
$y = 3$;
$x = x + 2$;
#3;
$x = y + x$;

P_2 : #2;
$y = 5$;



Semantics of a VERILOG Program

The semantics of a program P (considered as a closed system) is:

$$\mathcal{M}(P) \wedge \mathbf{Ax_1} \wedge \mathbf{Ax_2}$$

where

$$\mathbf{Ax_1} : \forall x \in \mathit{Var}^{+} \; . \; \Box(\mathsf{cint} \Rightarrow \mathbf{b}.x = \mathbf{e}.x)$$

$$\mathbf{Ax_2} : \; \exists i \in I_{Proc} \; . \; \Box(\mathsf{dint} \Rightarrow \mathbf{e}.\partial = i)$$

Semantics of some Commands

- Assignment
  $\mathcal{M}(v = e) \stackrel{\text{def}}{=}$

  $$\ell = 0 \wedge (\mathsf{idle}_{\mathsf{i}} \frown (\mathsf{dint} \wedge \mathbf{e}.v = \mathbf{b}.e \wedge$$

  $$unchanged_{\mathit{Var}-\{v\}} \wedge \mathbf{e}.\partial = i))$$

- Sequential Composition
  $\mathcal{M}(\texttt{begin } S_1; S_2; \ldots; S_n \texttt{ end}) \stackrel{\text{def}}{=}$

  $$\mathcal{M}(S_1) \frown \mathcal{M}(\texttt{begin } S_2; \ldots; S_n \texttt{ end})$$

- Iteration
  $\mathcal{M}(\texttt{while } (eb)\, S) \stackrel{\text{def}}{=}$

  $$\nu X.((\mathcal{M}(eb) \frown \mathcal{M}(S)) \frown X) \vee \mathcal{M}(\neg eb))$$

- Parallel Composition

  $\mathcal{M}(P_1 \parallel \ldots \parallel P_n) \stackrel{\text{def}}{=}$

  $\bigvee_{i=1}^{n}((\mathcal{M}(P_1)\frown \mathsf{idle}_1) \wedge \ldots \wedge \mathcal{M}(P_i)$

  $\wedge \ldots \wedge (\mathcal{M}(P_n)\frown \mathsf{idle}_n))$

- Boolean Expressions

  $\mathcal{M}(eb) \stackrel{\text{def}}{=}$

  $\ell = 0 \wedge (\mathsf{idle}_i \frown (\mathbf{b}.eb \wedge \mathsf{dint} \wedge unchanged_{Var} \wedge$
  $\mathbf{e}.\partial = i))$

Semantics of other commands

- Delay
  $\mathcal{M}(\#n) \stackrel{\text{def}}{=} \mathsf{idle}_{\mathsf{i}} \wedge \ell \leq n \wedge (\widehat{\mathsf{fin}} \Rightarrow \ell = n)$

- Conditionals
  $\mathcal{M}(\texttt{if } (eb)\, S_1 \texttt{ else } S_2) \stackrel{\text{def}}{=}$

  $(\mathcal{M}(eb) \frown \mathcal{M}(S_1)) \vee (\mathcal{M}(\neg eb) \frown \mathcal{M}(S_2))$

- Continuous assignment
  $\mathcal{M}(\texttt{assign } w = e) \stackrel{\text{def}}{=} [\![w = \mathcal{M}(e)]\!] \wedge \mathsf{inf}$

Conclusions

- We have given a formal semantics of a subset of Verilog using μWDCI

- Some features:
  - It is compositional
  - It is maximal (vs prefix closed)
  - It has infinite intervals (infinite zero-time computations)
  - It allows to represent intermediate transitions

Conclusions

- Future works:
  - Extend the language (i.e. non-blocking assignments)
  - Change the semantics for Hybrid Systems
  - Axiomatisation of μWDCI