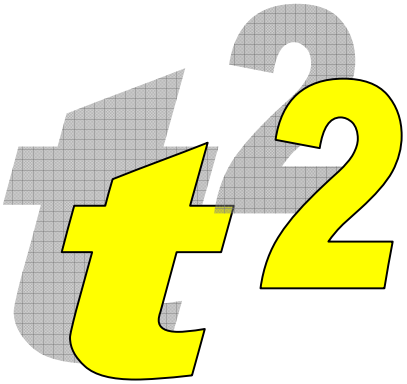


TimeSquare

… once upon a Time

C. André
AOSTE project I3S/INRIA
University of Nice, CNRS, INRIA
December, 2008

TimeSquare Environment

- Purpose: Modeling & Analysis of timed systems
- Full support of the MARTE time profile and CCSL (Clock Constraint Specification Language)
- Multiform times
- Relies on a formal semantics of CCSL
- Timed behaviors displayed as waveforms (VCD format)
- Integrated in the OpenEmbeDD platform

Outline

1. Time Model in MARTE
2. CCSL
3. TimeSquare
4. Example

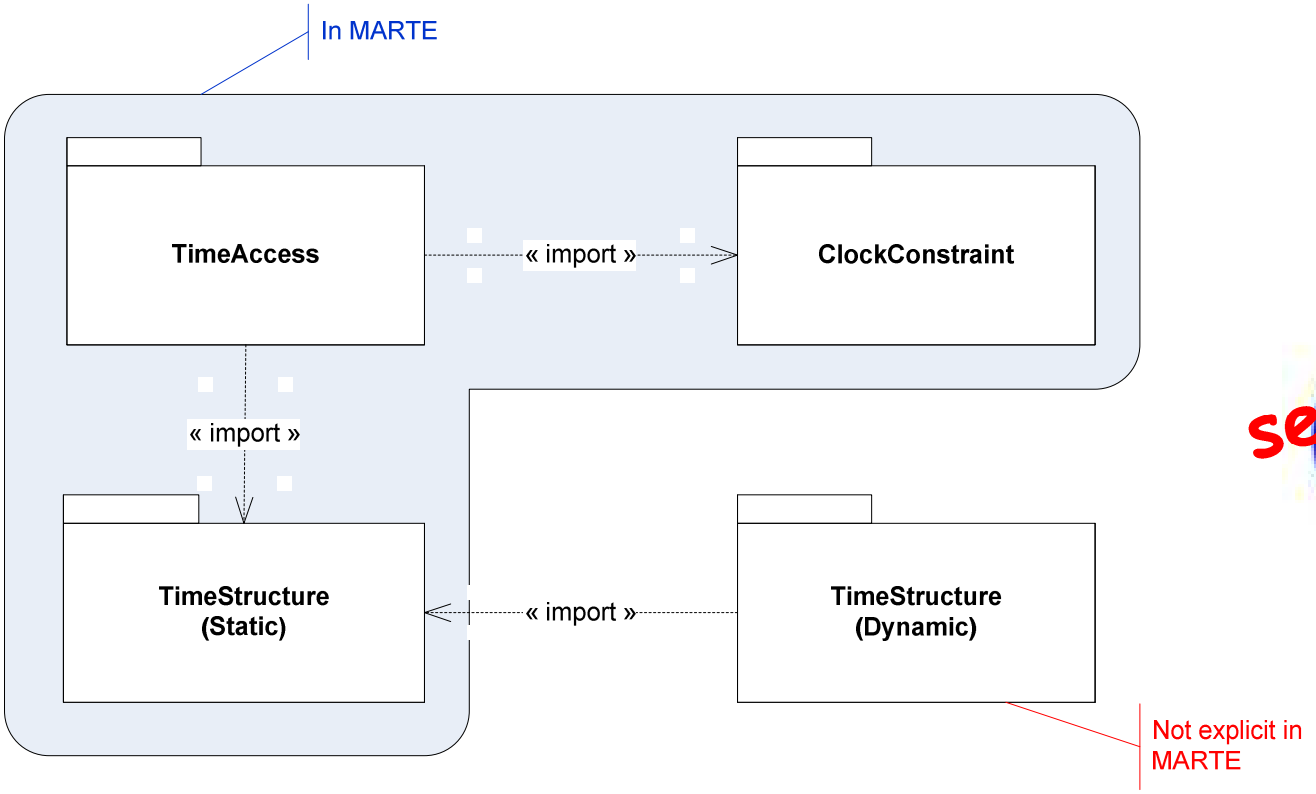
Time in MARTE

… The Times they are a'changing (B. Dylan)

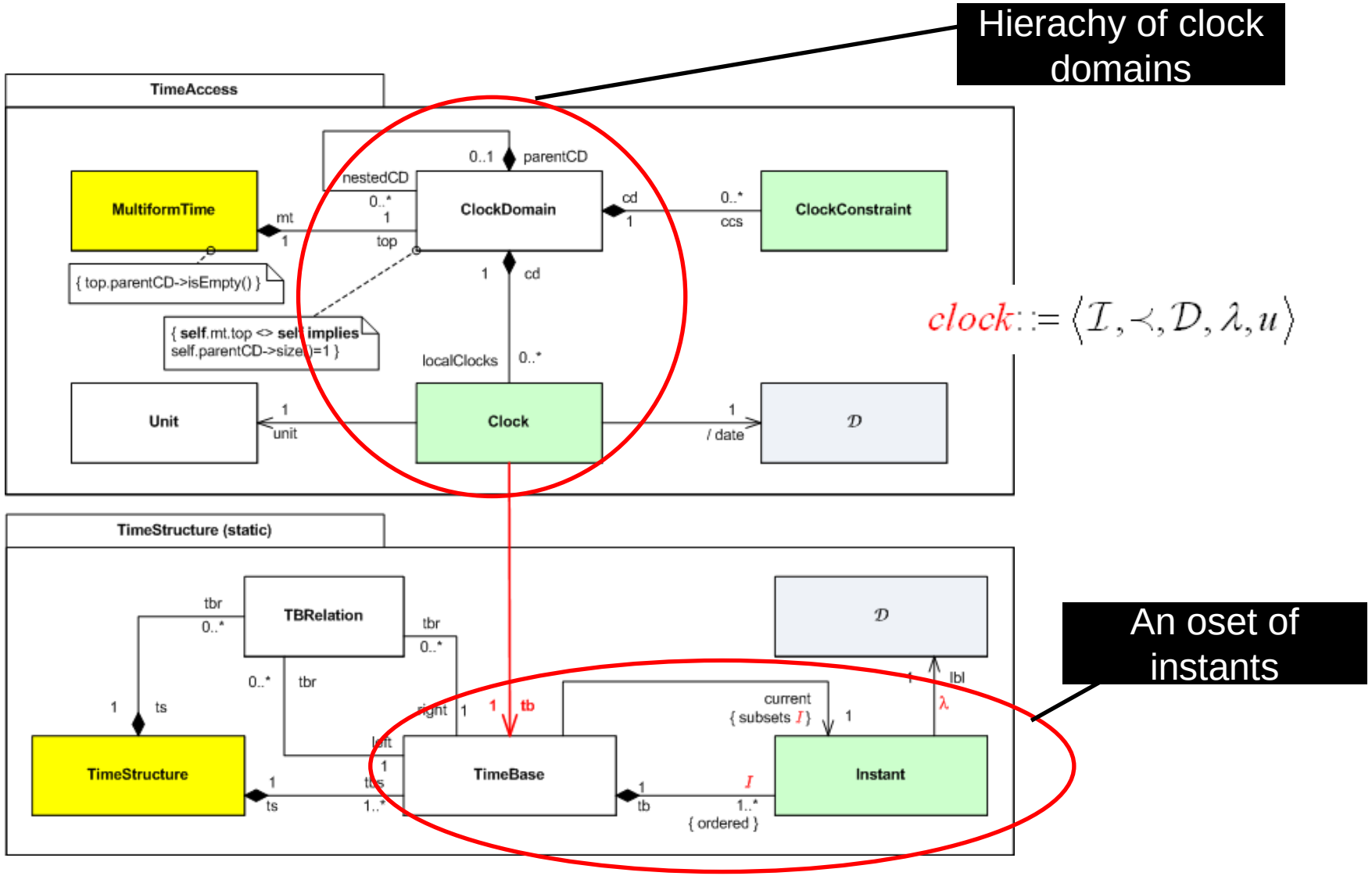
Or … Time is money (OMG)

Time in MARTE (1)

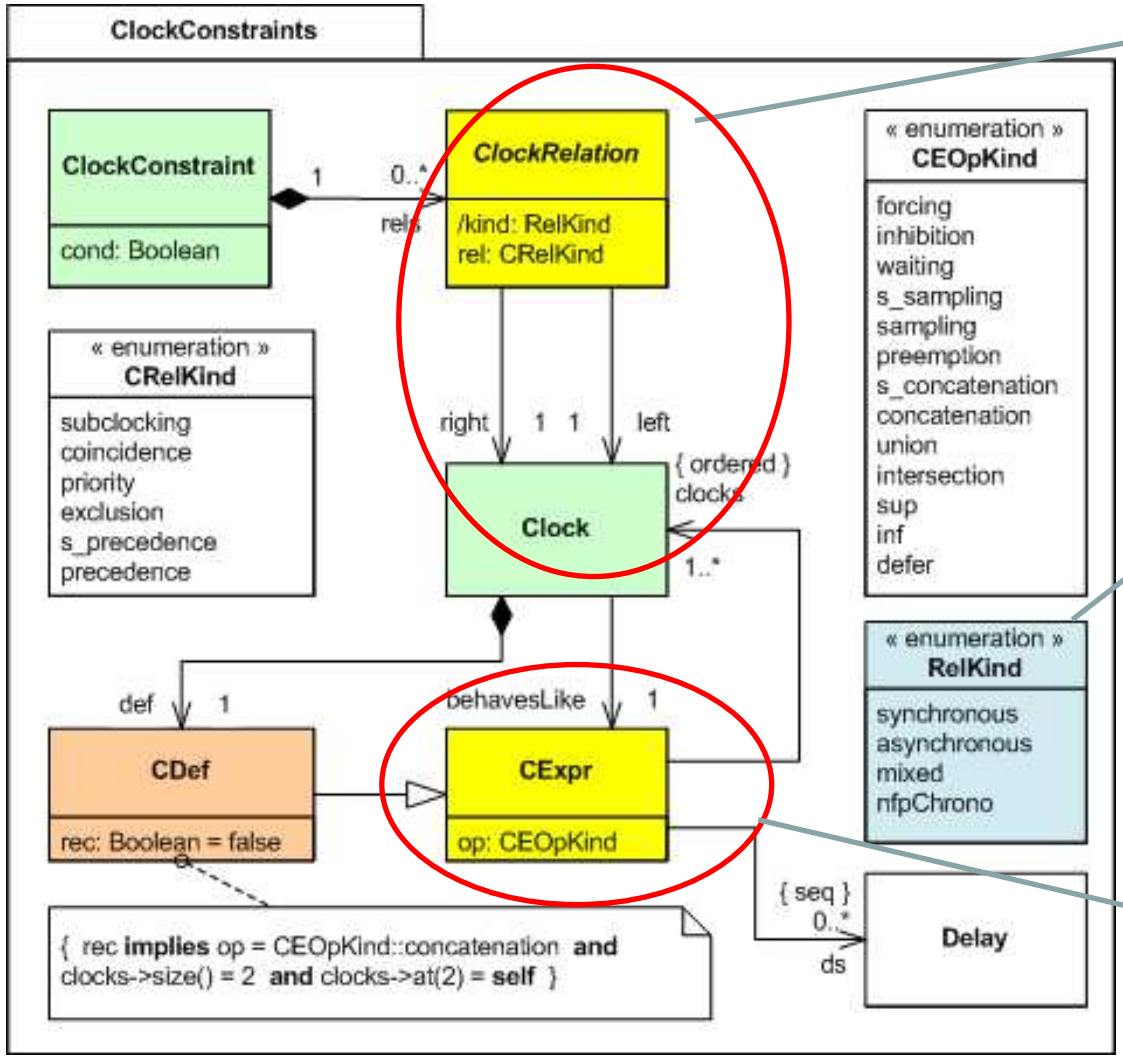
UML models are mostly untimed
Goal: Equip UML with rich time concepts: Timed model elements
Hidden intent: capability to deal with synchronous concepts



Time in MARTE (2)



Clock Constraints

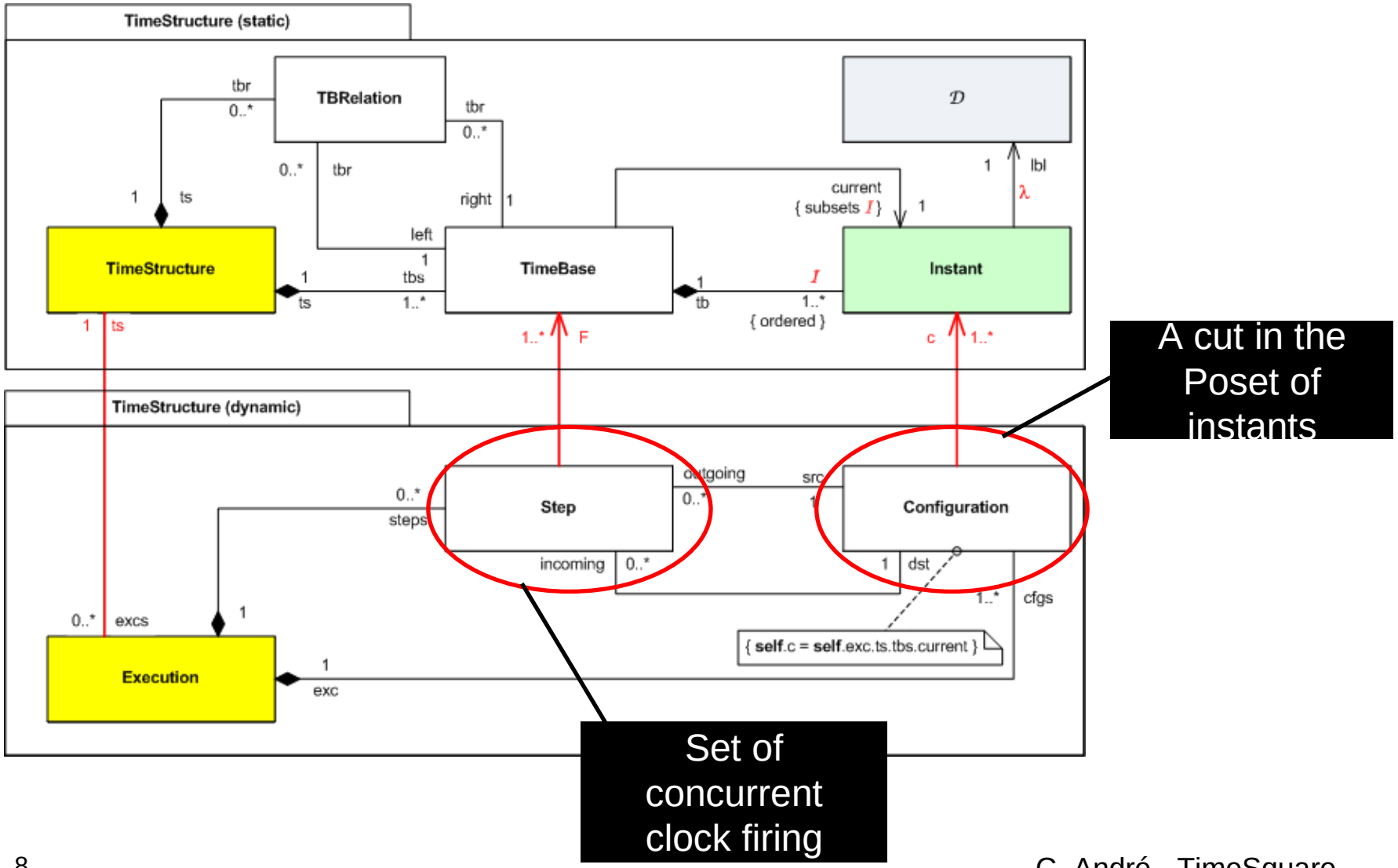


Relation between clocks

Kinds of relations

Expressions on clocks

Executable model



Informal introduction to Clock Constraints

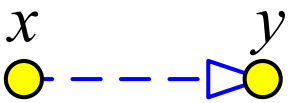
… from Time to Time

Relations on Instants

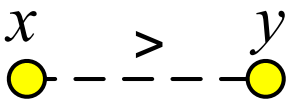
coincidence $x \equiv y$

strict precedence $x \prec y$

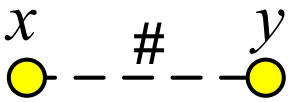
precedence $x \preccurlyeq y$



priority $x \multimap y$

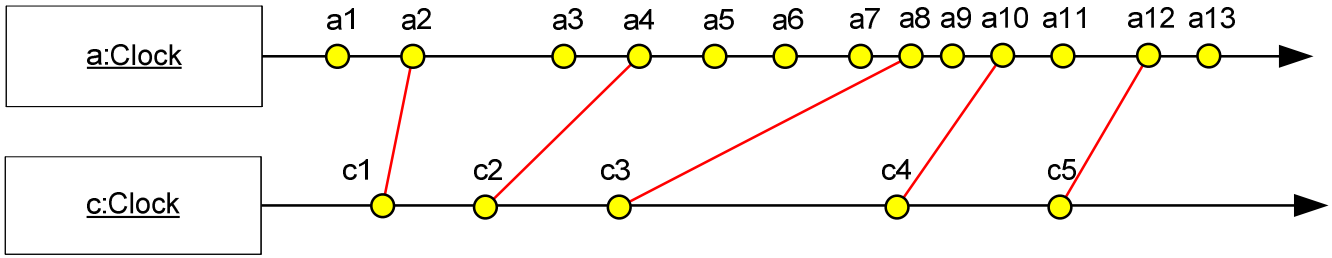


exclusion $x \# y$



Relations on TimeBases (1)

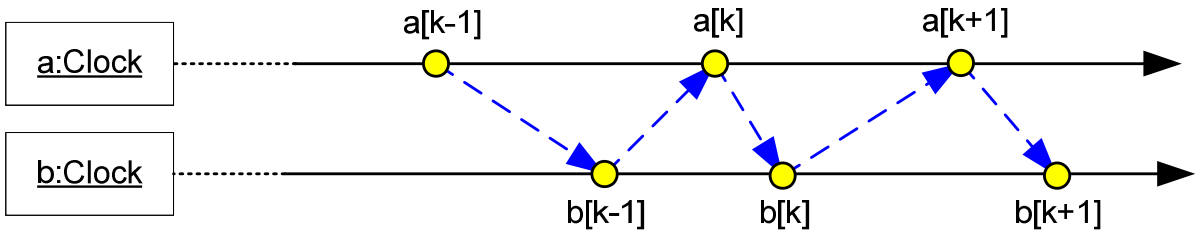
An example of Synchronous relation



c = a filteredBy 0b0101000(10)

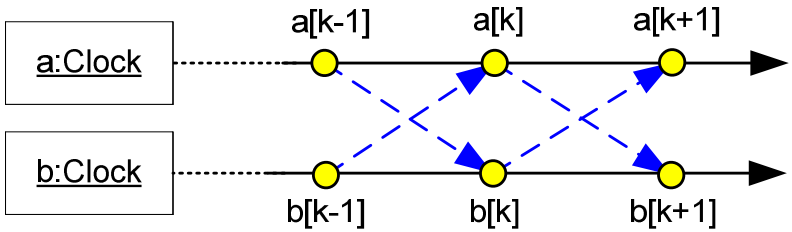
Relations on TimeBases (2)

2 examples of Asynchronous relations



a alternatesWith b

$$a \sim b \triangleq \left(\forall k \in \mathbb{N}^*\right) a[k] \prec b[k] \prec a[k+1]$$

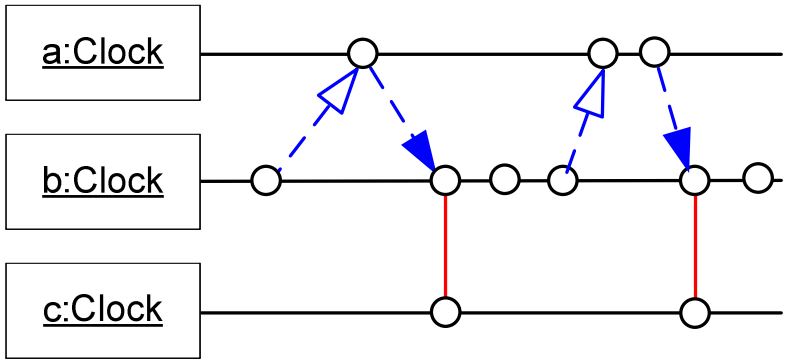


a synchronizesWith b

$$a \triangleright\!\triangleleft b \triangleq \left(\forall k \in \mathbb{N}^*\right)\left(a[k] \prec b[k+1]\right) \wedge \left(b[k] \prec a[k+1]\right)$$

Relations on TimeBases (3)

An example of Mixed relation



c = a sampledOn b

$$c = a \searrow b \triangleq \left(\exists i, j, k \in \mathbb{N}^*\right)\left(b[i-1] \preccurlyeq a[k] \prec b[i]\right) \Rightarrow \left(b[i] \equiv c[j]\right)$$

Semantics of CCSL

… hard time

Kernel CCSL (1)

Clock **Expressions**

Terminating CExpr

$\tau^0 ::= !1$ // forcing

$| !0$ // inhibition

$| t \wedge n$ // waiting next nth t

$| t \searrow_{\bullet} t$ // (strict) sampling

$| t \searrow_{\circ} t$ // non-strict sampling

$| t \lightning t$ // t upto t

Non-terminating CExpr, c-independent

$\tau^1 ::= t$ // simple time base reference

$| t \bullet t$ // concatenation

$| t + t$ // union = coarsest finer $(\mathrm{Sup}_{\subseteq})$

$| t * t$ // intersection

$| t, \sigma \rightsquigarrow t$ // defer

Non-terminating CExpr, c-dependent

$\tau^2 ::= t \vee t$ // sup = fastest slower $(\mathrm{Sup}_{\preccurlyeq})$

$| t \wedge t$ // inf = slowest faster $(\mathrm{Inf}_{\preccurlyeq})$

CExpr

$\tau ::= \tau^0 | \tau^1 | \tau^2$ // simple clock expressions

$|$ if b then τ else τ // conditional clock expression

Kernel CCSL (2)

Clock & Instant **Relations**

Definition CRel

$\rho^d ::= t \triangleq \tau$ // clock standard definition

$| \; t \triangleq t \bullet t$ // clock recursive definition

Basic CExpr, c-independent $\rho^1 ::= t \left(\boxed{\subset} | \boxed{\multimap} | \boxed{\#} \right) t$ // basic constraints on clocks, cnt-independent

Basic CExpr, c-dependent $\rho^2 ::= t \left(\boxed{=} | \boxed{\prec} | \boxed{\preccurlyeq} \right) t$ // basic constraints on clocks, cnt-dependent

Clock Constraint

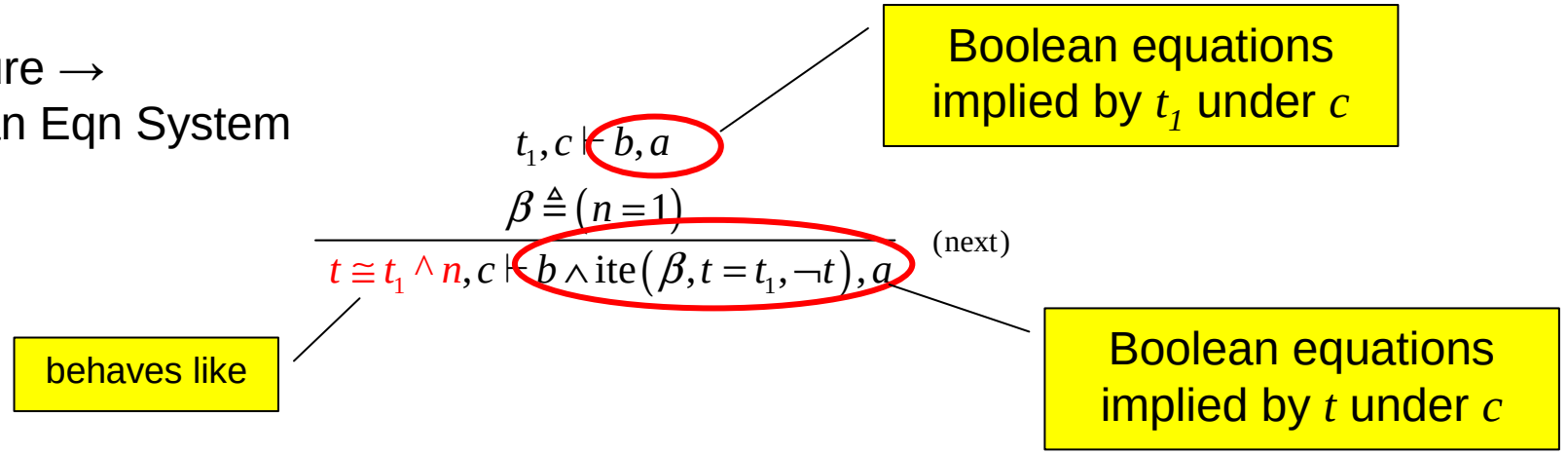
$\gamma ::= \gamma | \gamma$ // constraint conjunction

$| \; \rho^d | \rho^1 | \rho^2$ // simple relation

$| \; \rho \text{ if } b$ // conditional relation

Example of Semantic Rules

Structure →
Boolean Eqn System
⇒ E



F ⇒ Rewritten Structure

$$t \cong t_1 \wedge 1 \xrightarrow{t_1 \in F} t \cong 0 \quad \text{(RWnext 1)}$$

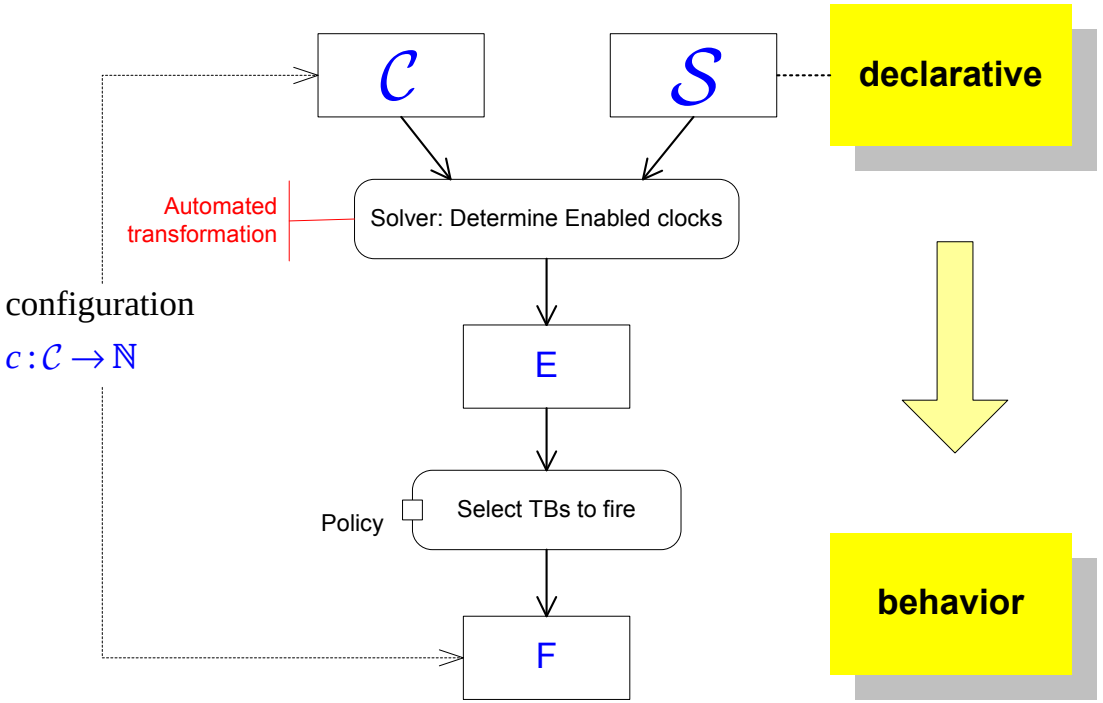
$$\frac{n > 1 \quad t_1 \xrightarrow{t_1 \in F} t_2}{t \cong t_1 \wedge n \xrightarrow{t_1 \in F} t \cong t_2 \wedge (n-1)} \quad \text{(RWnext 2)}$$

Conditional rewriting rules

From Constraints to Behavior

… Time after Time

Overview



Objective: build sequences of steps that respect $\mathcal{S}$

Solution: SOS for a Kernel of CCSL. $\mathcal{S}, c \xrightarrow{F} \mathcal{S}', c'$

User's viewpoint: standard CCS library provided

\+ facility for user's defined constraints

\+ stochastic parameters

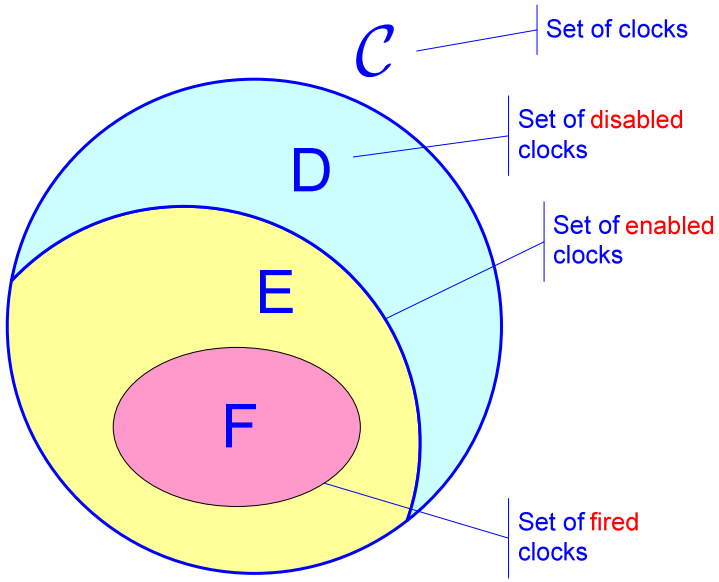
Three phases

For each step:

1. For each statement determine the set of implied “Boolean constraints”
2. When all constraints are analyzed, determine the set of enabled clocks (E)
3. Select the set of clocks to fire (F)

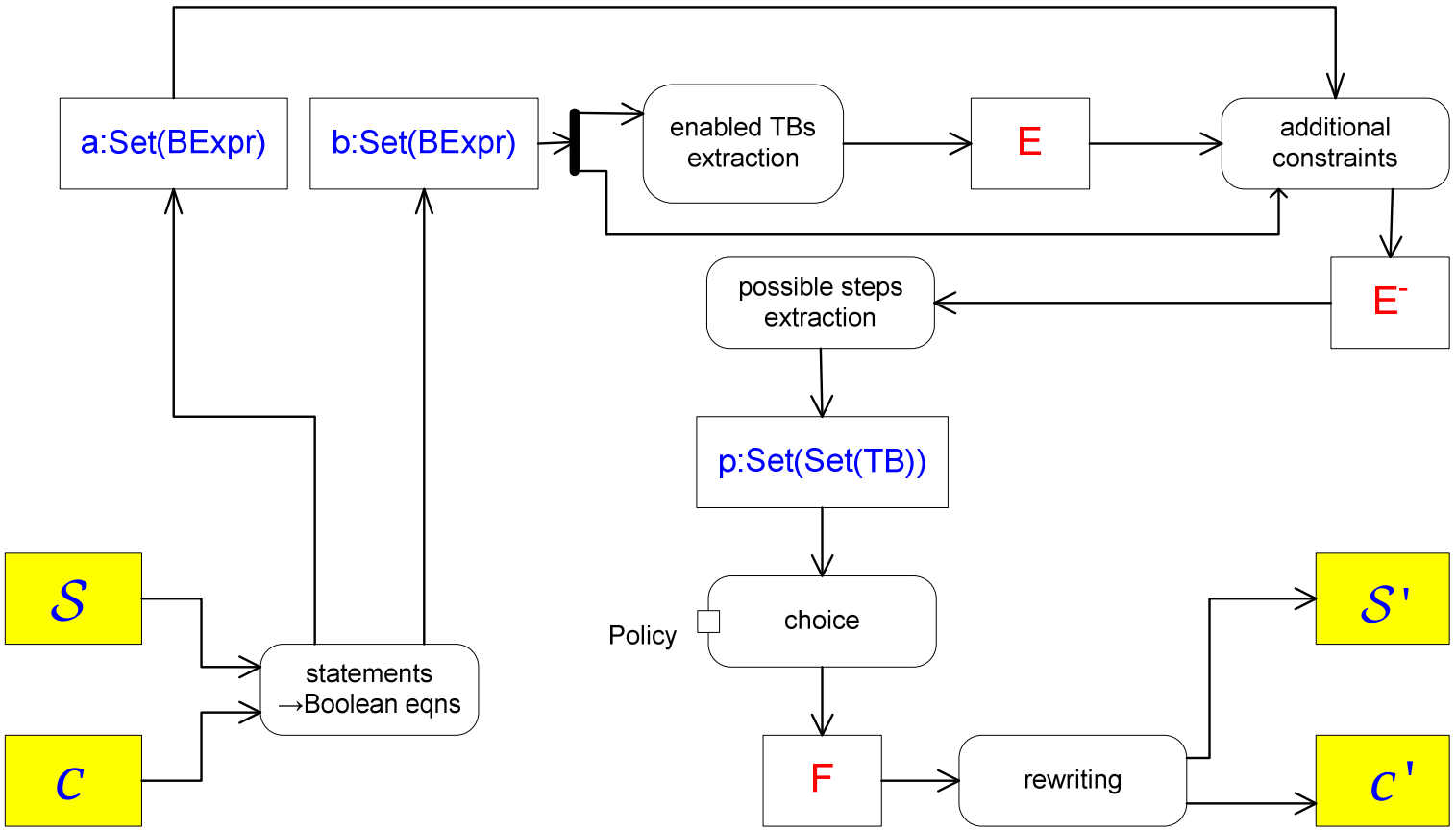
Sets of clocks

Sets:



For a given configuration c

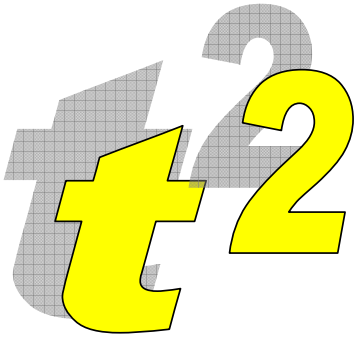
Computation of a Step



TimeSquare

… have a good Time

TimeSquare



Four main functionalities

1. Interactive clock-related specifications
2. Clock constraint checking
3. Generation of sequences of steps
4. Displaying & exploring waveforms

TimeSquare implementation

- Plug-in developed with Ganymede Eclipse Modeling Tools
  - Includes EMF, GMF, MDT XSD/OCL/UML2, M2M, and EMFT
- CCSL parser: ANTLR 2.7
- Solver: JavaBDD
- Waveforms: VCD compliant (IEEE Std1364)
- Available at:

http://www-sop.inria.fr/aoste/dev/time_square/

Example: Digital Filter

Specification

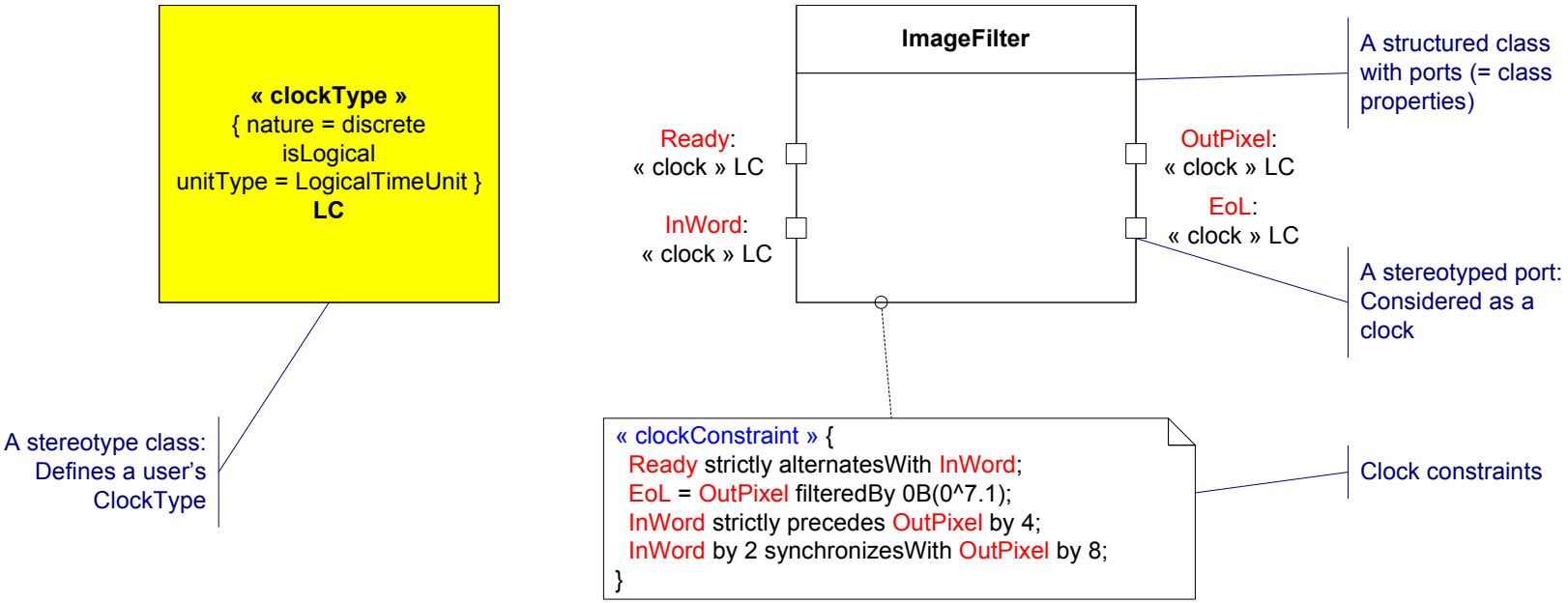
① $Ready \sim InWord$

② $EndOfLine = OutPixel \blacktriangledown \left(0^{LINE\_LENGTH-1}.1\right)^{\omega}$

③ $InWord \prec OutPixel \,/\, PIXELS\_PER\_WORD$

④ $InWord \,/\, WORDS\_PER\_LINE \rhd\lhd_{\equiv} OutPixel \,/\, LINE\_LENGTH$

UML user's view



Digital Filter (2)

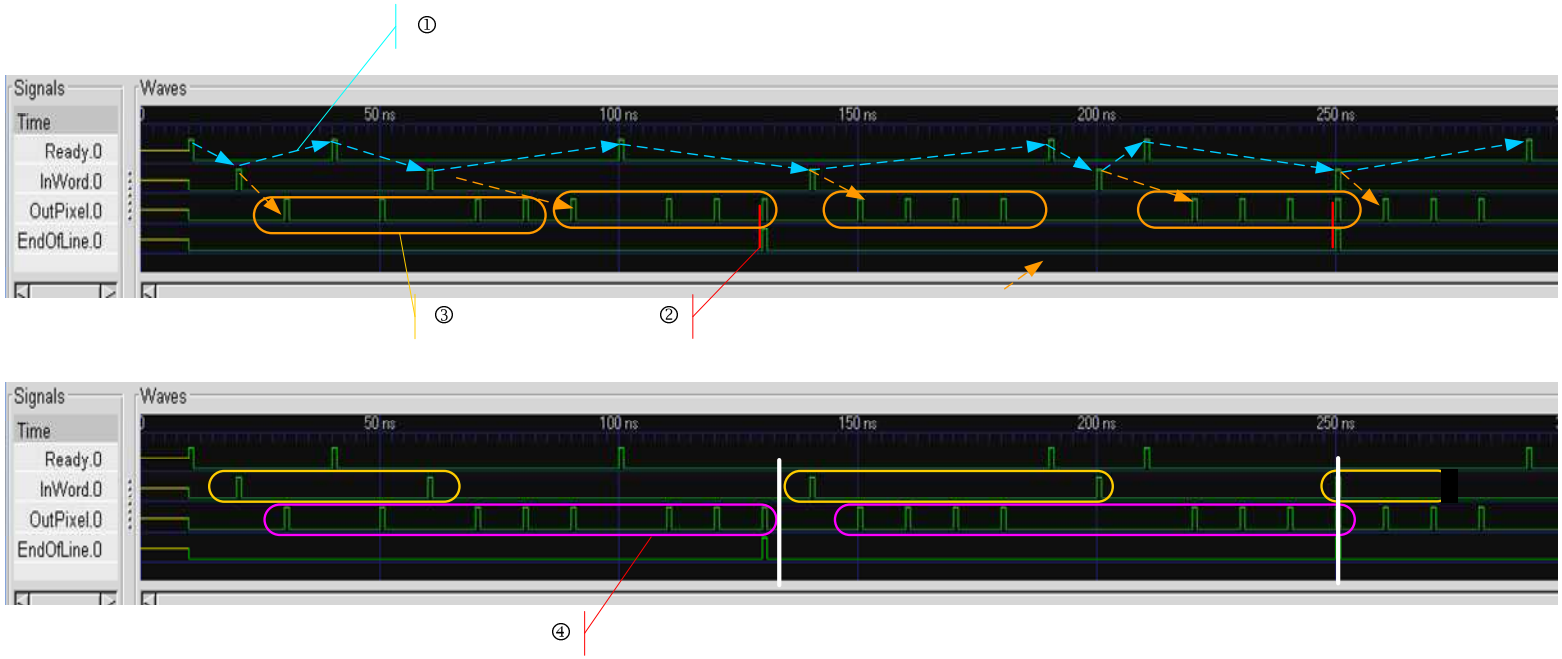
Specification

① $Ready \sim InWord$

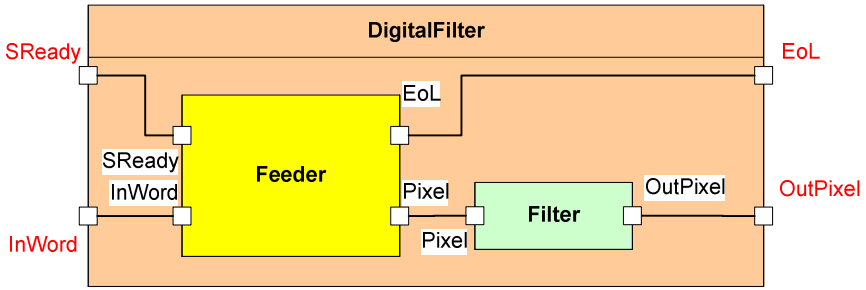
② $EndOfLine = OutPixel \blacktriangledown \left(0^{LINE\_LENGTH-1}.1\right)^{\omega}$

③ $InWord \prec OutPixel \,/\, PIXELS\_PER\_WORD$

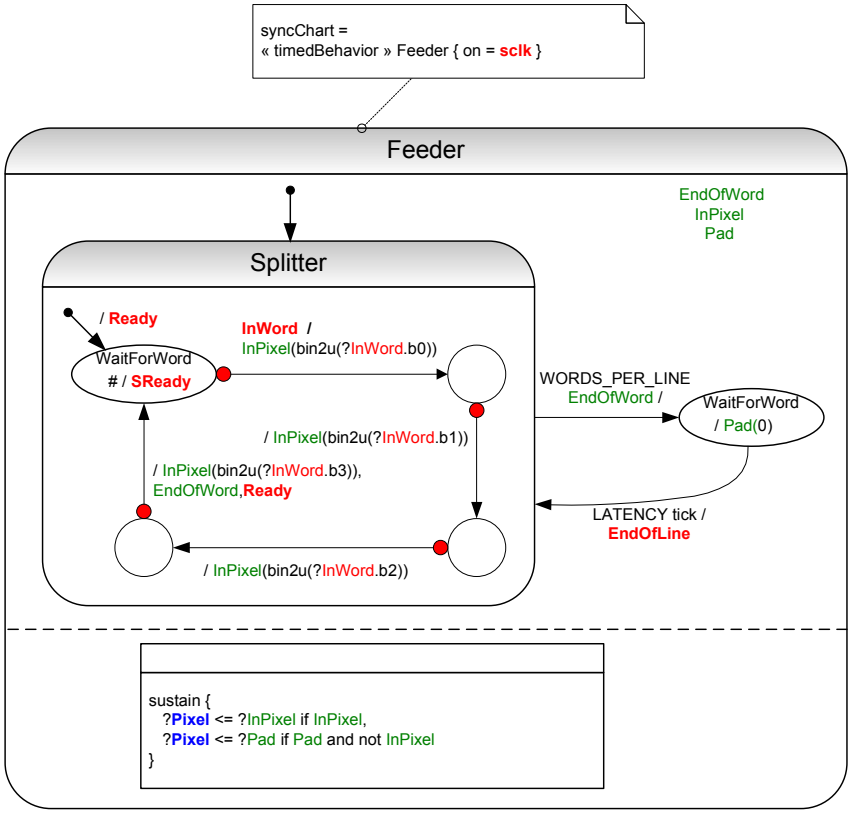
④ $InWord \,/\, WORDS\_PER\_LINE \rhd\lhd_{\equiv} OutPixel \,/\, LINE\_LENGTH$

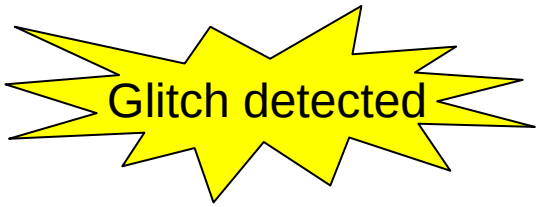


Synchronous Implementation



$$
\begin{aligned}
&\{\ \text{Clock } \mathit{InPixel}, \mathit{Pad}, \mathit{EndOfWord}, \mathit{FirstInstant}, \mathit{Pixel}\\
&\quad \mathit{FirstInstant} = \mathit{sclk} \blacktriangledown 1\\
&\quad \mathit{InWord} = \mathit{InPixel} \blacktriangledown \left(1.0^3\right)^{\omega}\\
&\quad \mathit{EndOfWord} = \mathit{InPixel} \blacktriangledown \left(0^3.1\right)^{\omega}\\
&\quad \mathit{InPixel} = \mathit{Pixel} \blacktriangledown \left(1^8.0^2\right)^{\omega}\\
&\quad \mathit{Pad} = \mathit{Pixel} \blacktriangledown \left(0^8.1^2\right)^{\omega}\\
&\quad \mathit{EndOfLine} = \mathit{Pixel} \blacktriangledown \left(0^9.1\right)^{\omega}\\
&\quad \mathit{OutPixel} = \mathit{InPixel} \text{ delayedFor 2 on } \mathit{Pixel}\\
&\quad \mathit{Ready} = \mathit{FirstInstant} + \mathit{EndOfWord}\\
&\quad \mathit{SReady} = \text{sustain } \mathit{Ready} \text{ upto } \mathit{InWord}\\
&\}
\end{aligned}
$$





The end

… Time is up!