Synchronous Programming Techniques for Embedded Systems

Gérard Berry

Chief Scientist

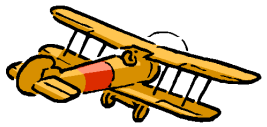
www.esterel-technologies.com
Gerard.Berry@esterel-technologies.com

- 1982-1985 : first ideas, languages, and semantics
  - Esterel (Berry – Rigault, Sophia-Antipolis)
  - Lustre (Caspi – Halbwachs, Grenoble)
  - Signal (Benveniste – Le Guernic, Rennes)
- 1985-1995 : more languages, semantics, compiling & verification
  - SyncCharts (André), Reactive C (Boussinot), TCC (Saraswat)
  - causality analysis (Gonthier, Shiple)
  - links to dataflow (Ptolemy), to hardware (Vuillemin), etc.
  - formal verification techniques (Madre & Coudert, Touati)
- 1995 –2000 : maturation, industrial experimentation
  - active international research (Edwards, Schneider, Ramesh, etc.)
  - applications: avionics, nuclear plant safety, telecom, robotics
- 2000-2006 : industrial expansion
  - major standard in avionics, expanding in rail, automotive, etc.
  - hardware circuit design

Beware of the computer!

- computers + SoCs = hardware / software mix
- complete change in device interaction
- ever-growing number of critical applications

Applications and Constraints

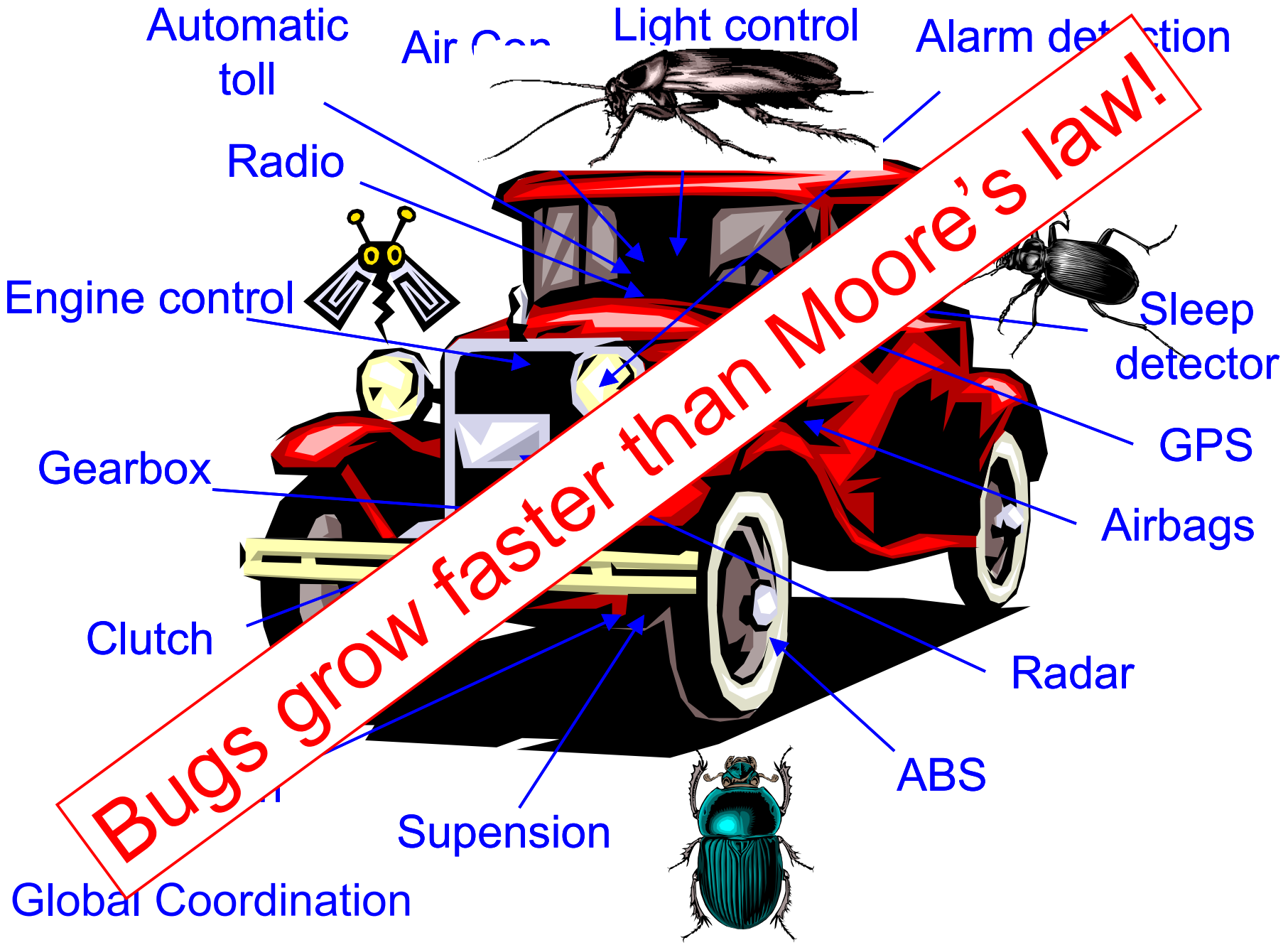


flight-control, engines, brakes, fuel, power, climate
safety-critical => certification

trajectory, attitude, image, telecom
mission-critical => very high quality

telephone, audio, TV, DVD, games
business critical => time-to market + quality

pacemakers, diabet control, robot surgeons
life-critical => TBD (!)

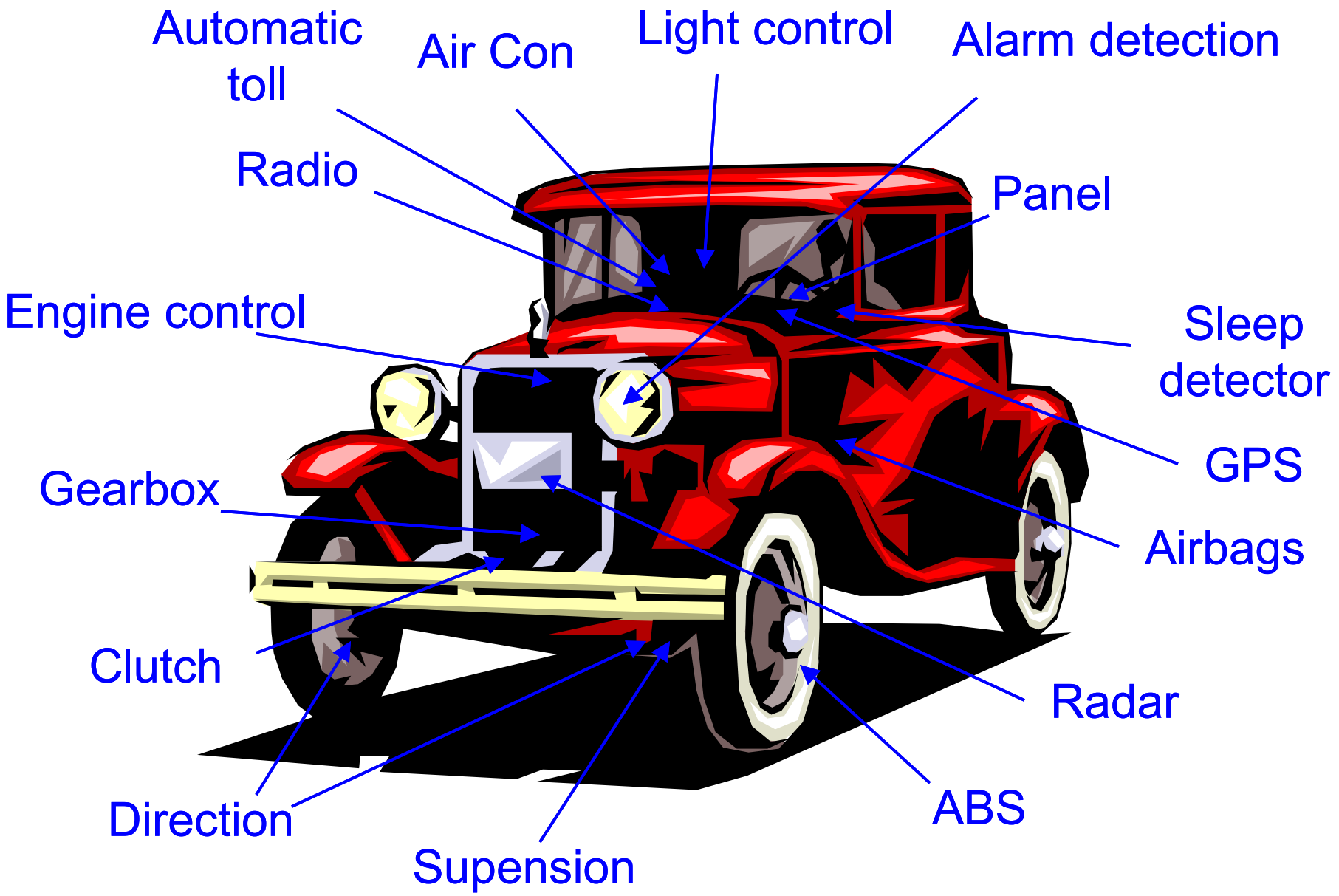


How to avoid or control bugs?

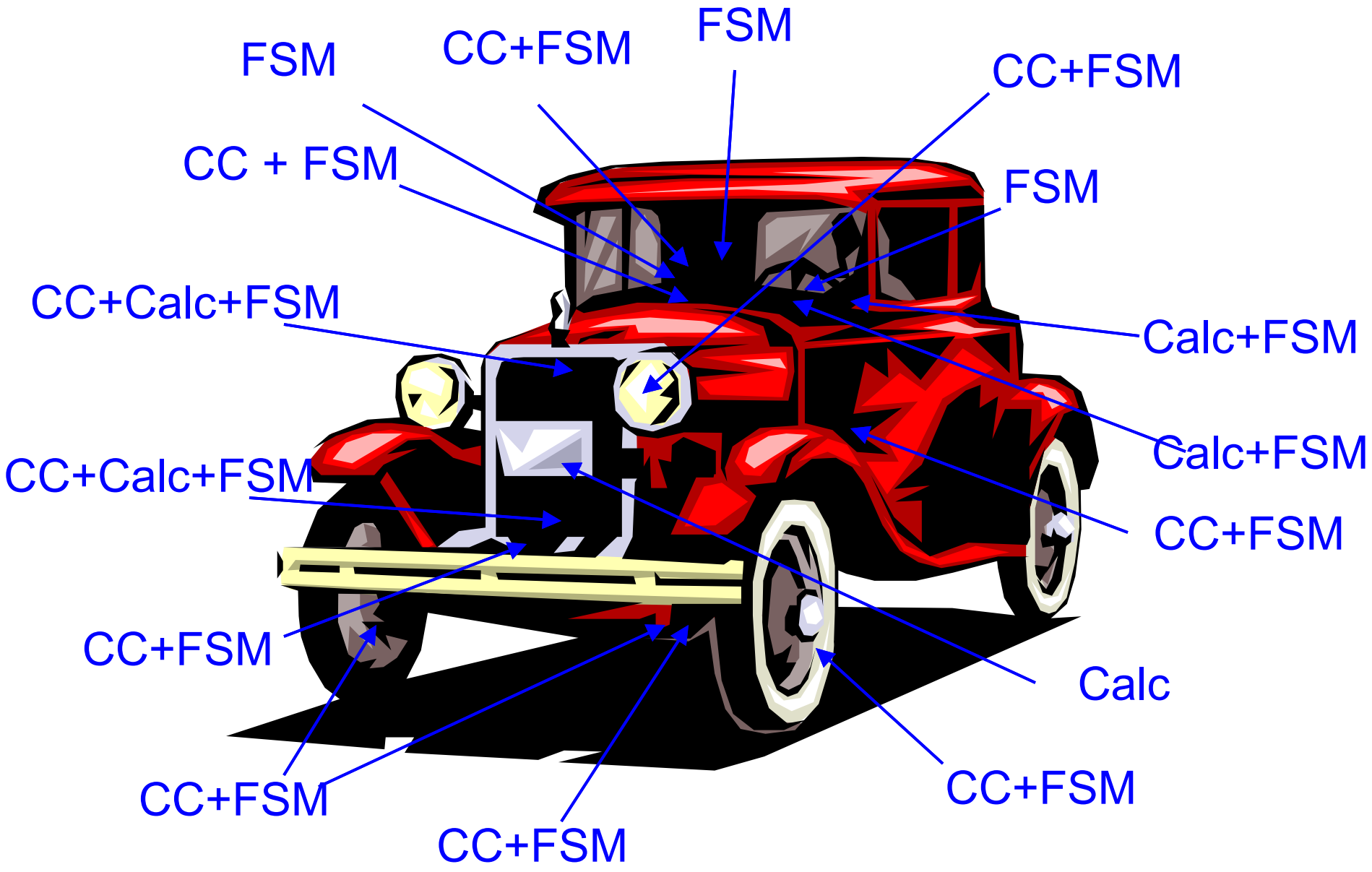
- Traditional : better verification by fancier simulation
- Next step : better design
  - better and more reusable specifications
  - simpler computation models, formalisms, semantics
  - reduce architect / designer distance
  - reduce hardware / software distance
- Mandatory: better tooling
  - synthesis from high-level descriptions
  - formal property verification / program equivalence
  - certified libraries

Embedded Modules Anatomy

- CC : continuous control, signal processing
  - differential equations, digital filtering
  - specs and simulation with Matlab / Scilab
- FSM : finite state machines (automata)
  - discrete control, protocols, security, displays, etc.
  - flat or hierarchical FSMs
- Calc : heavy calculations
  - navigation, encryption, image processing
  - C + libraries
- Web : HMI, audio / video
  - user interaction / audio / vidéo
  - data flow networks, Java



Global Coordination



Global Coordination : Calc+CC+FSM

Key Computation Principles

- Concurrency is fundamental
  - implicit in CC, audio / video, protocols, etc.
  - also mandatory for Web and Calc
- Determinism is fundamental
  - implicit for CC and FSM
  - who would drive a non-deterministic car?
  - can be relaxed for Web, infotainment, etc.
- Physical distribution becomes fundamental
  - separation of functions, links between them
  - redundancy for fault-tolerance
  - global time needed for distributed control

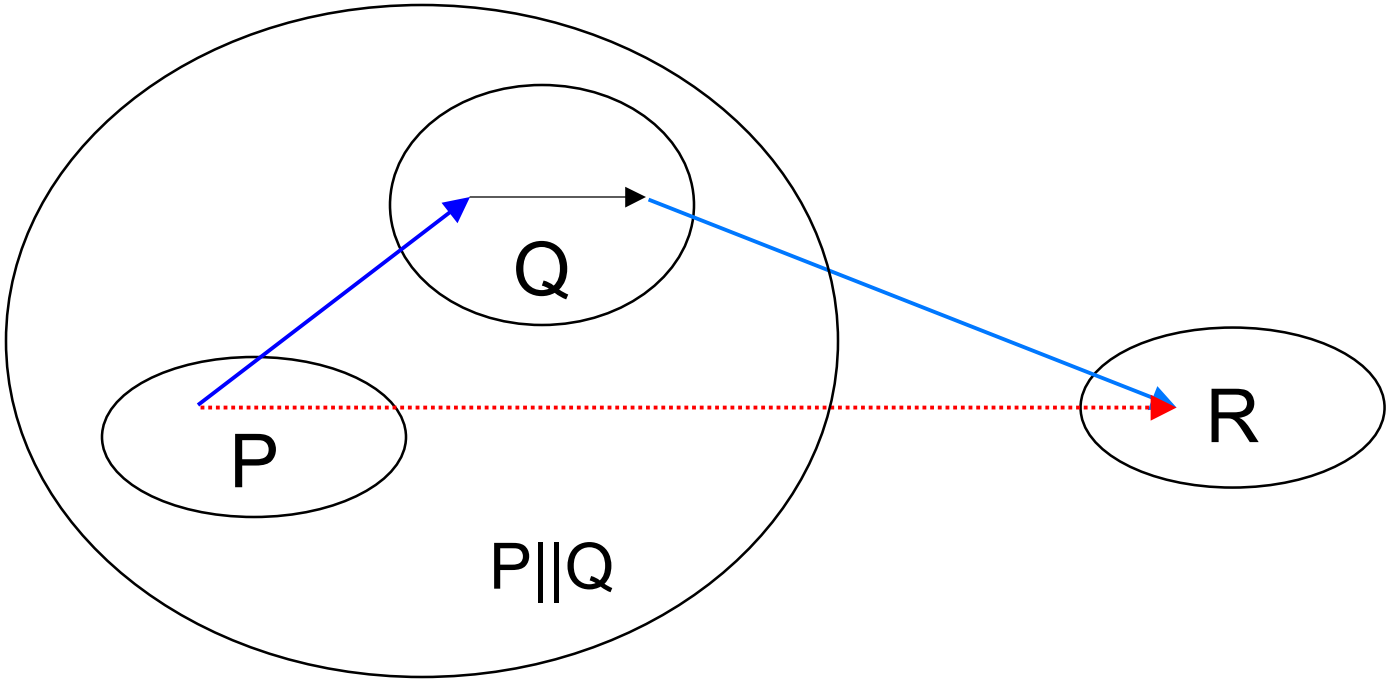
The Classical Software Development Model is Inadequate

- Turing complete => too rich, too hard to check
- OS- or thread-based concurrency => too hard too check
  interference, non-determinism
- CC implementation too indirect (manual action scheduling)
- Inadapted to circuit design (except for filters)

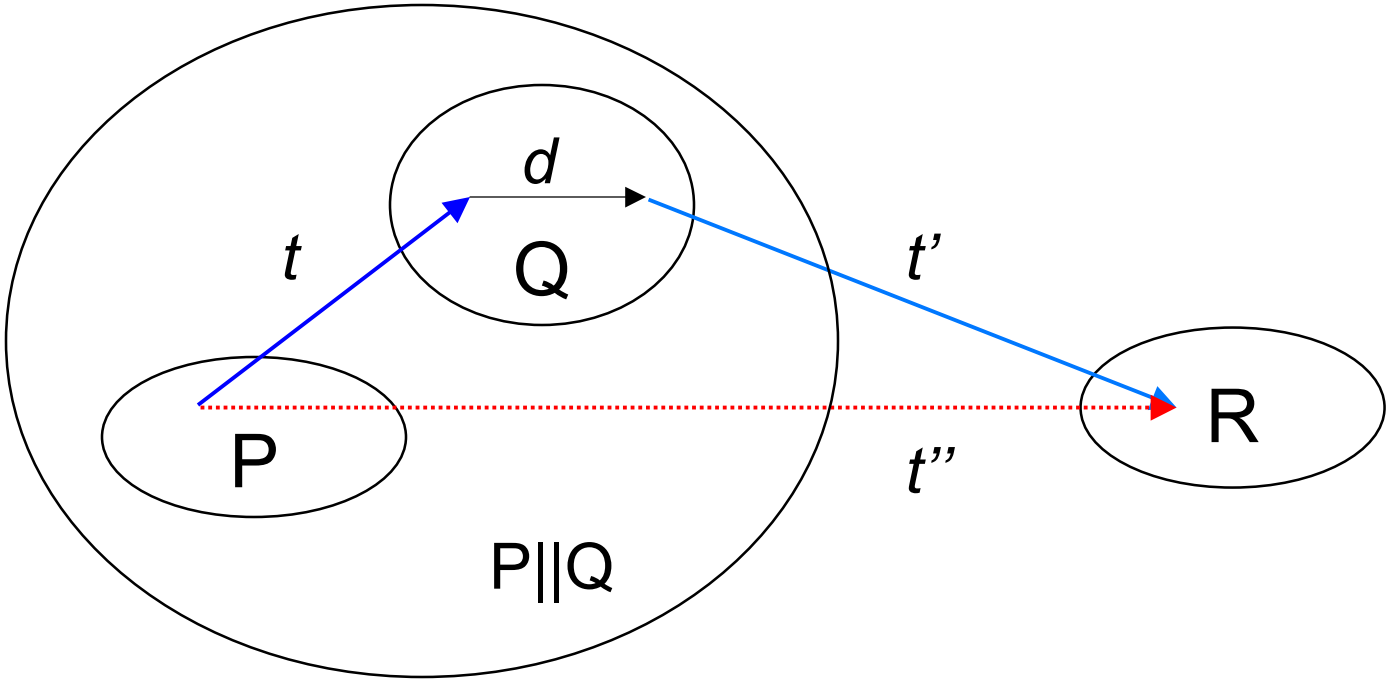
The Classical Hardware Development Model becomes Inadequate

- Structural RTL descriptions hide behavior dynamics
- HDLs inadequate for software
- Concurrency OK, but sequencing very indirect
- Quite old language basis, semantics too vague

$\Rightarrow$ much simpler models are needed
that reconcile sequencing and concurrency



Concurrency : the compositionality principle



$$t'' = t + d + t'$$

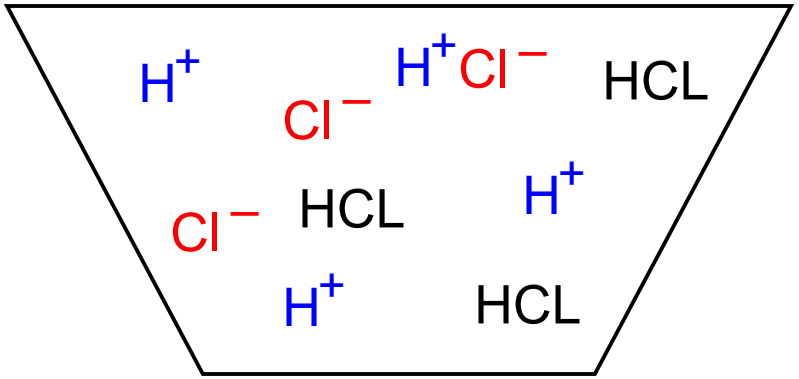
$$t'' \sim t \sim d \sim t'$$

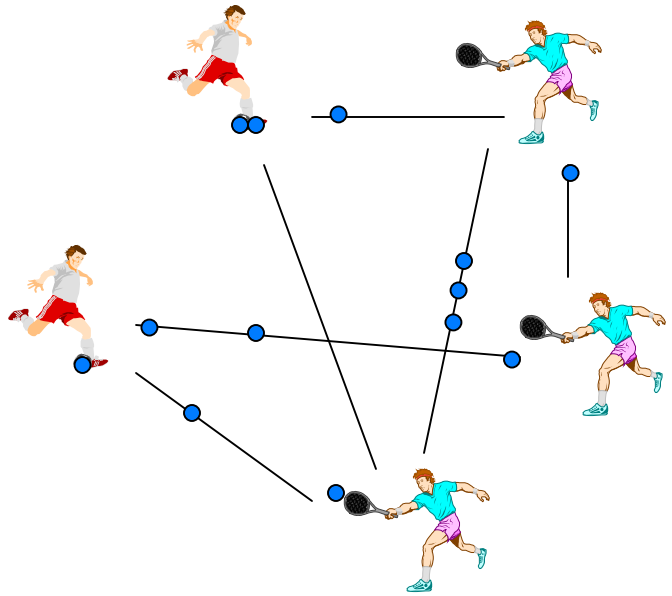
$$t \sim t + t$$

Only 3 solutions :

- *t* arbitrary asynchrony
- *t* = 0 synchrony
- t predictable vibration

Arbitrary Delay : Brownian Motion





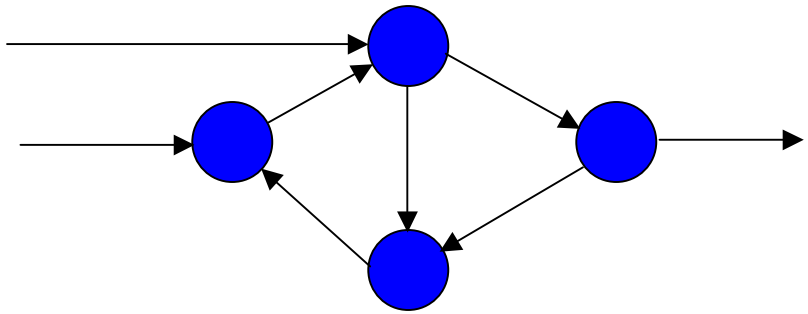
Chemical reaction

$H^+ + Cl^- \longleftrightarrow HCL$

Internet routing

Models : Kahn networks, π-calculus, CHAM, Join-Calculus, Ambients, etc...

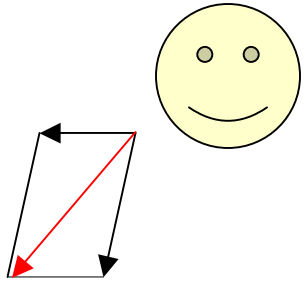
Kahn Networks



nodes = deterministic programs
arrows = infinite fifos

- result-dterministic (independent of computation order)
- easy semantics by flow equations
- heavily used in streaming applications (audio, TV)

Zero delay example: Newtonian Mechanics



Concurrency + Determinism
Calculations are feasible

The most difficult real-time manoeuver ever

Refer to a fabulous drawing of Hergé's
"On a Marché sur la Lune", in English
"Explorers on the Moon". French edition, page 10,
first drawing.

Drunk Captain Haddock has become a satellite of the Adonis asteroid. To catch him, Tintin, courageously standing on the rocket's side, asked Pr. Calculus to start the rocket's atomic engine. At precisely the right time, he shouts "STOP"!

This is the trickiest real-time manoeuver ever performed by man. It required a perfect understanding of Newtonian Mechanics and absolute synchrony.

Refer to the 3rd drawing of page 10 of "Explorers on the Moon". Tintin's manoeuver was a perfect success, and he now catches Haddock with a lasso (highly non-trivial in deep space!)

Digital synchronous circuits

the RTL zero-delay model

Synchronous languages

Esterel, Lustre, Signal, ...

Excellent abstract model

```
trap HeartAttack in
    every Morning do
        abort
            loop
                abort  run Slowly  when  100 Meter ;
                abort
                    every Step do
                        run Jump || run Breathe || run CheckHeart
                    end every
                when 15 Second ;                    exit HeartAttack
                run FullSpeed
            each  Lap
        when  2 Lap
    end every
handle HeartAttack fo
   run RushToHospital
end trap
```

t predictable : vibration

Nothing can illustrate vibration better than Bianca Castafiore, Hergé's famous prima donna. See [1] for details. The power of her voice forcibly shakes the microphone and the ears of the poor spectators.

[1] King's Ottokar Sceptre, Hergé, page 29, last drawing.

propagation of light, electrons, program counter...

Full Abstraction

Bianca Castafiore singing for the King Muskar XII in Klow, Syldavia. King's Ottokar Sceptre, page 38, first drawing.

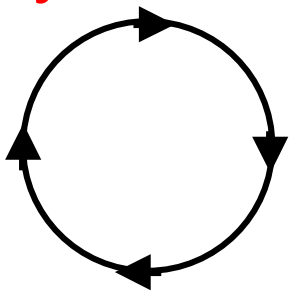
Although the speed of sounds is finite, it is fast enough to look infinite. Full abstraction!

If room is small enough,
predictable delay implements zero-delay

Specify with zero-delay
Implement with predictable delay
Control room size

Software Synchronous Systems

Cycle based



read inputs

compute reaction

produce outputs

Synchronous = 0-delay = within the same cycle

propagate control

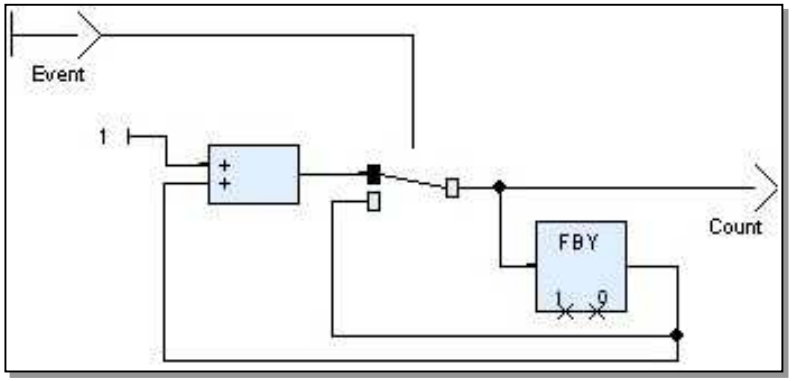
propagate signals

No interference between I/O and computation
Room size control = Worst Case Execution Time (AbsInt)

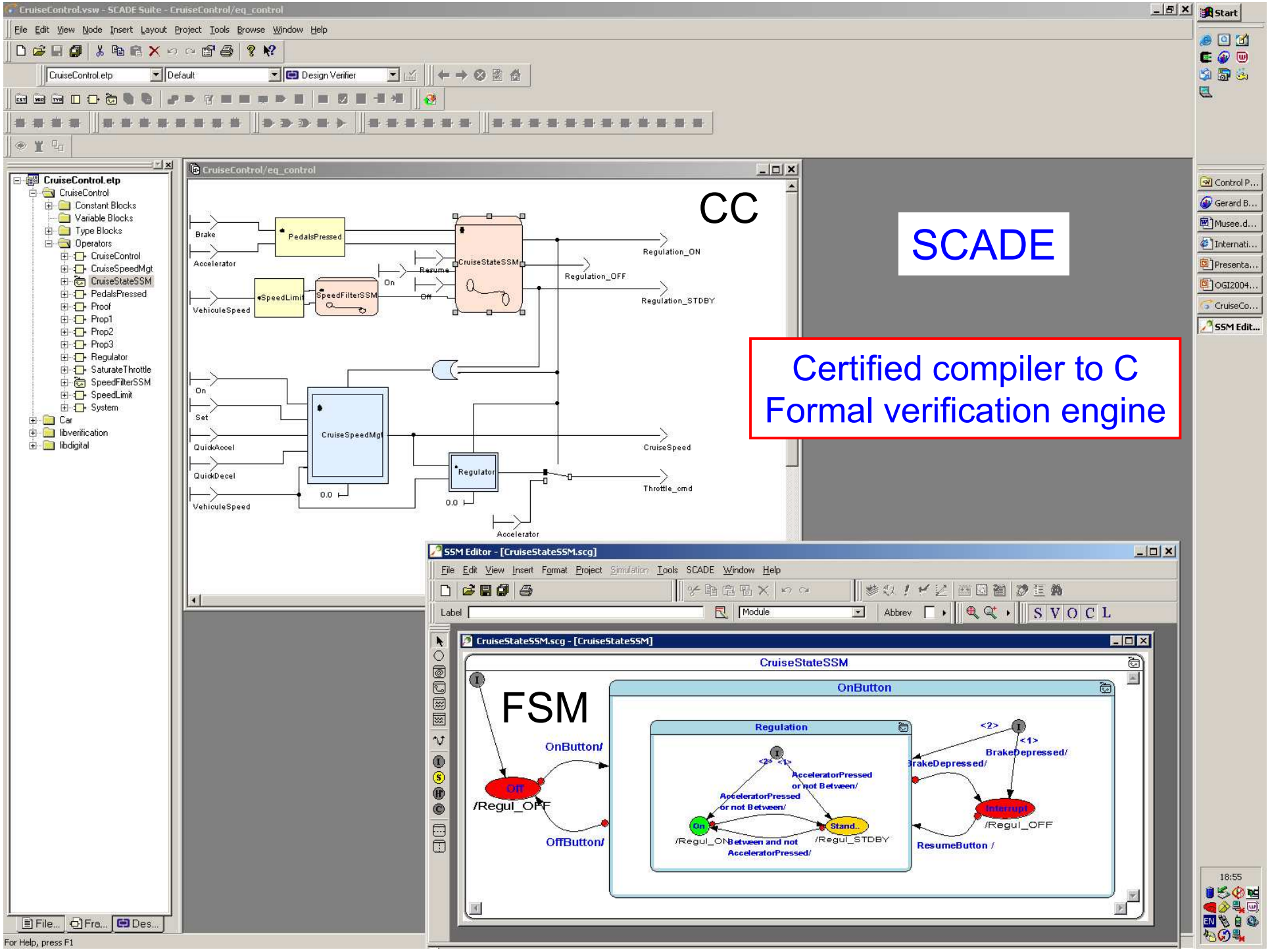
Lustre = Synchronous Kahn Networks

A simple counter

$$\begin{cases} Count(0) = 0 \\ \forall t > 0,\ Count(t) = \begin{cases} Count(t-1) + 1, & if\ Event(t) = true \\ Count(t-1), & otherwise \end{cases} \end{cases}$$



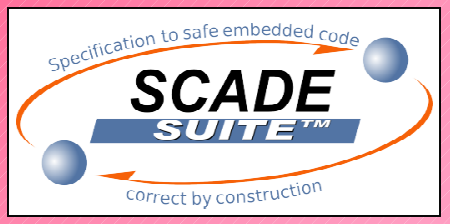

```
Count = 0 ->
    (if Event
     then pre(Count)+1
     else pre(Count)
```


CruiseControl.vsw - SCADE Suite - CruiseControl/eq_control
CruiseControl/eq_control
CC
SCADE
Certified compiler to C
Formal verification engine
SSM Editor - [CruiseStateSSM.scg]
CruiseStateSSM.scg - [CruiseStateSSM]
FSM
CruiseStateSSM
OnButton
Regulation
Off
/Regul_OFF
OnButton/
OffButton/
On
/Regul_ON
Stand..
/Regul_STDBY
Interrupt
/Regul_OFF
BrakeDepressed/
ResumeButton /
AcceleratorPressed
or not Between/
Between and not
AcceleratorPressed/
Brake
Accelerator
VehiculeSpeed
PedalsPressed
SpeedLimit
SpeedFilterSSM
CruiseStateSSM
Resume
On
Off
Regulation_ON
Regulation_OFF
Regulation_STDBY
Set
QuickAccel
QuickDecel
CruiseSpeedMgt
Regulator
CruiseSpeed
Throttle_cmd
0.0

SCADE Suite™ Customers Base

Civilian Avionics

- Aircraft Braking Systems
- Airbus
- Chengdu Aircraft Development & research Institute
- Chinese Aeronautical Radio Electronics Research Institute
- CMC Electronics Inc.
- Dassault Aviation
- Diehl Avionik Systeme GmH
- Elbit Systems
- Eurocopter
- Honeywell
- Flight Automatic Control Research Institute
- Liebherr-Aerospace
- Messier-Bugatti
- Nanjing University of Aeronautics and Aerospace
- Pratt & Whitney
- Rockwell Collins
- SAAB Aerospace
- SAFRAN
- Seditec
- Smiths Aerospace
- Snecma Aerospace
- Thales Avionics
- Transiciel
- Turbomeca



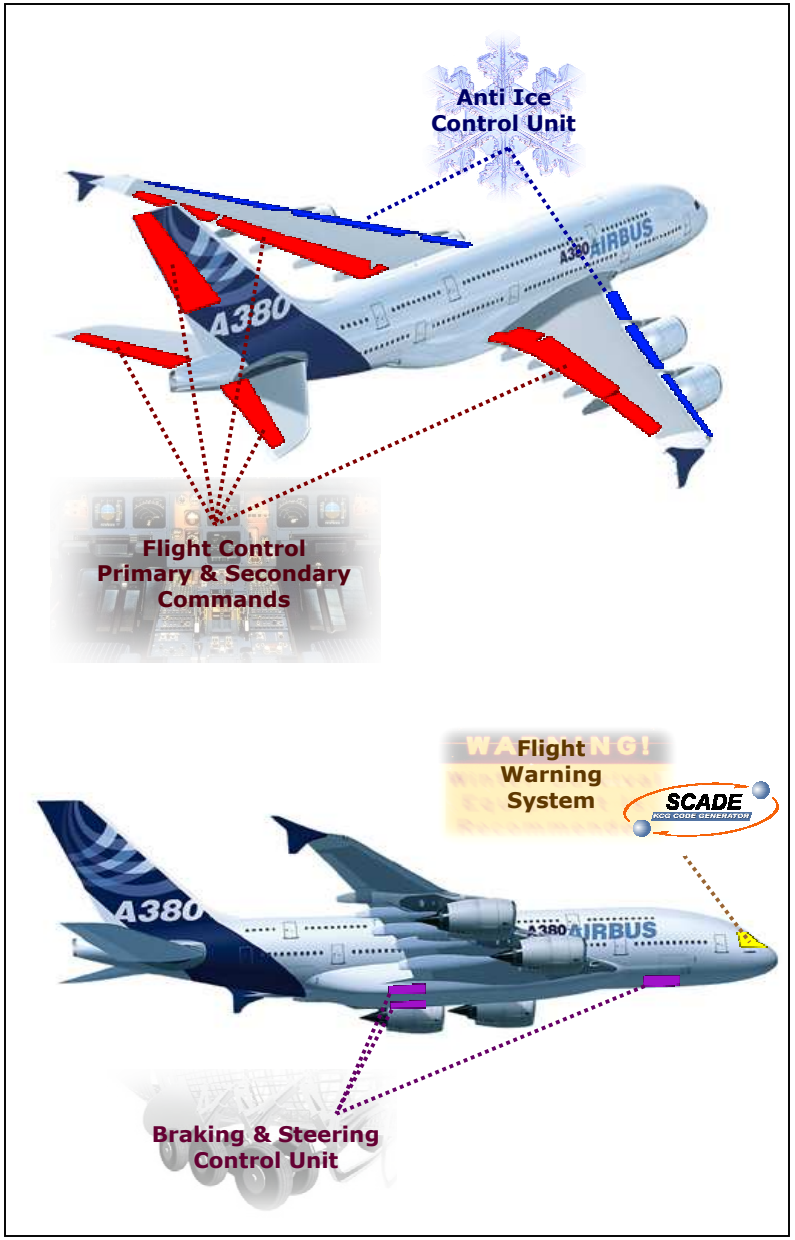
Energy & Transportation

- Ansaldo Signal
- DS & S
- Framatome
- Schneider Electric
- Siemens Transp.

Defense & Space

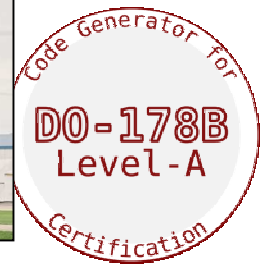
- CAST 504th Institute
- CRIL Technology
- Dassault Aviation
- EADS Military
- EADS SD Electronics
- EADS Space Transportation
- Elbit Systems Ltd.
- ESA
- Eurocopter
- Hills US Air Force Base
- Hispano-Suiza
- Intertechnique
- Lockheed Martin
- MBDA
- NASA
- Rockwell Collins
- Rockwell Collins Flight Dynamics
- SAGEM
- Thales Airborne Systems
- Thales Communication

SCADE Suite in the A380



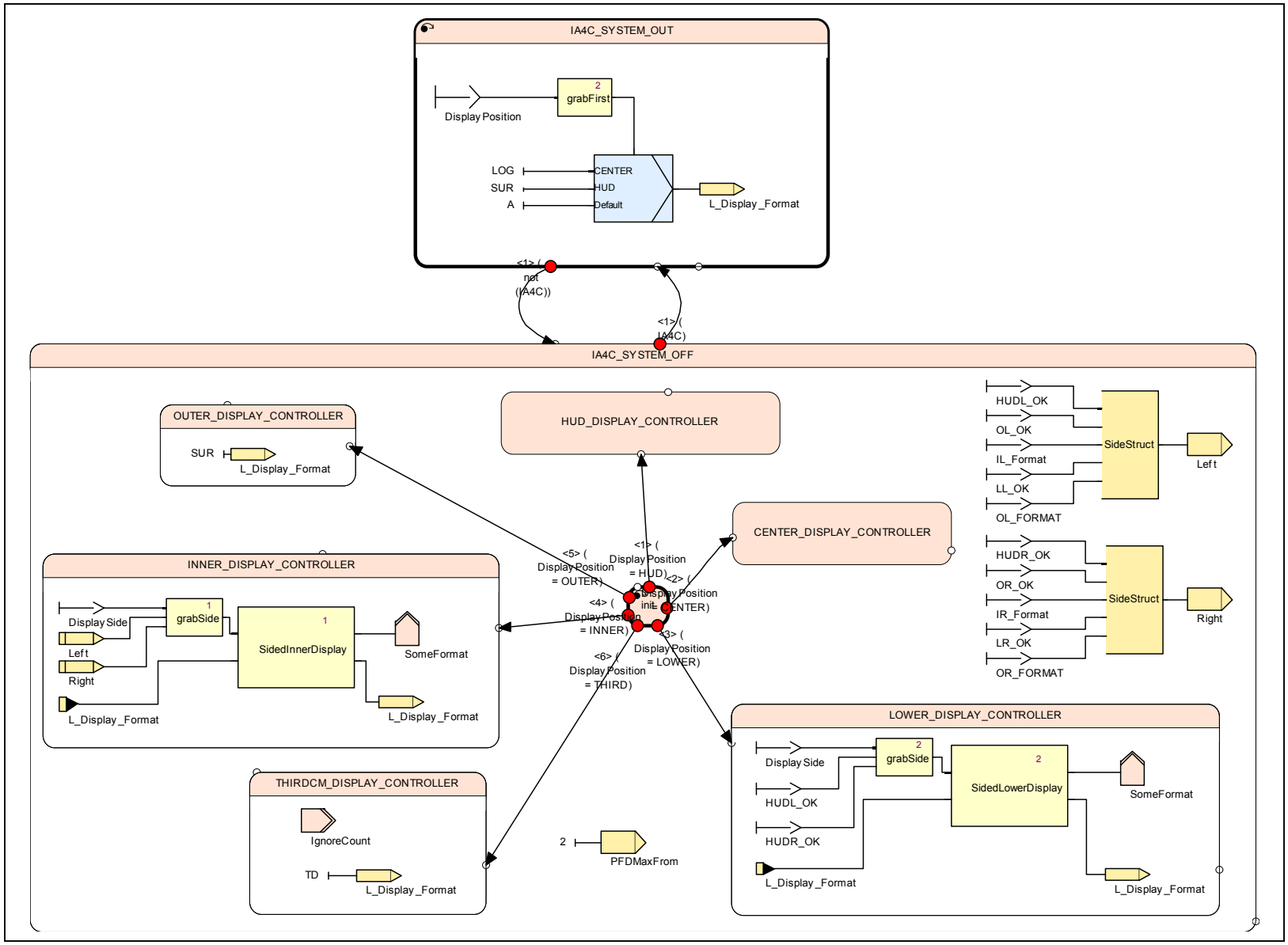
- SCADE = Airbus corporate standard for all new airplanes developments
  - Flight Control system
  - Flight Warning system
  - Electrical Load Management system
  - Anti Icing system
  - Braking and Steering system
  - Cockpit Display system
  - Part of ATSU (Board / Ground comms)
  - FADEC (Engine Control)
  - EIS2 : Specification GUI Cockpit:
    - PFD : Primary Flight Display
    - ND : Navigation Display
    - EWD : Engine Warning Display
    - SD : System Display

EUROCOPTER

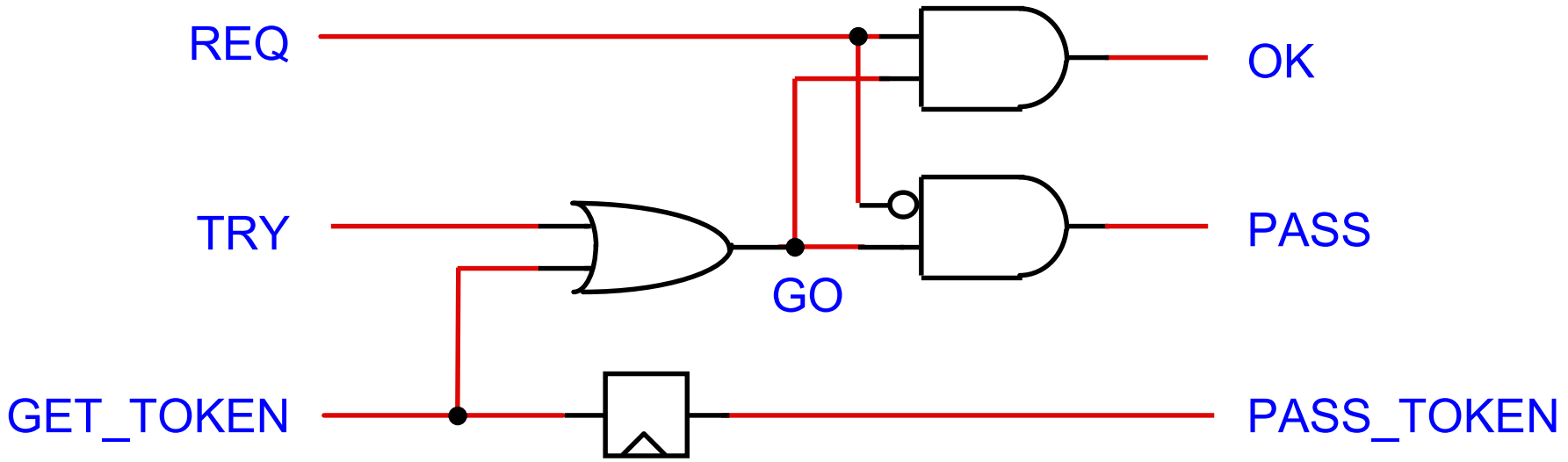


- World leader in civilian helicopters
- Introduced SCADE Suite™ for **EC135 and EC155 autopilots**
- Results
  - 90% of the code with SCADE
  - Development time divided by 2
  - (8 level A certifications by JAA : EC155, EC135, EC145; EC225 on-going)
  - The entire modification cycle can be performed in < 48 h !

SCADE 6 : full data-flow / control-flow integration



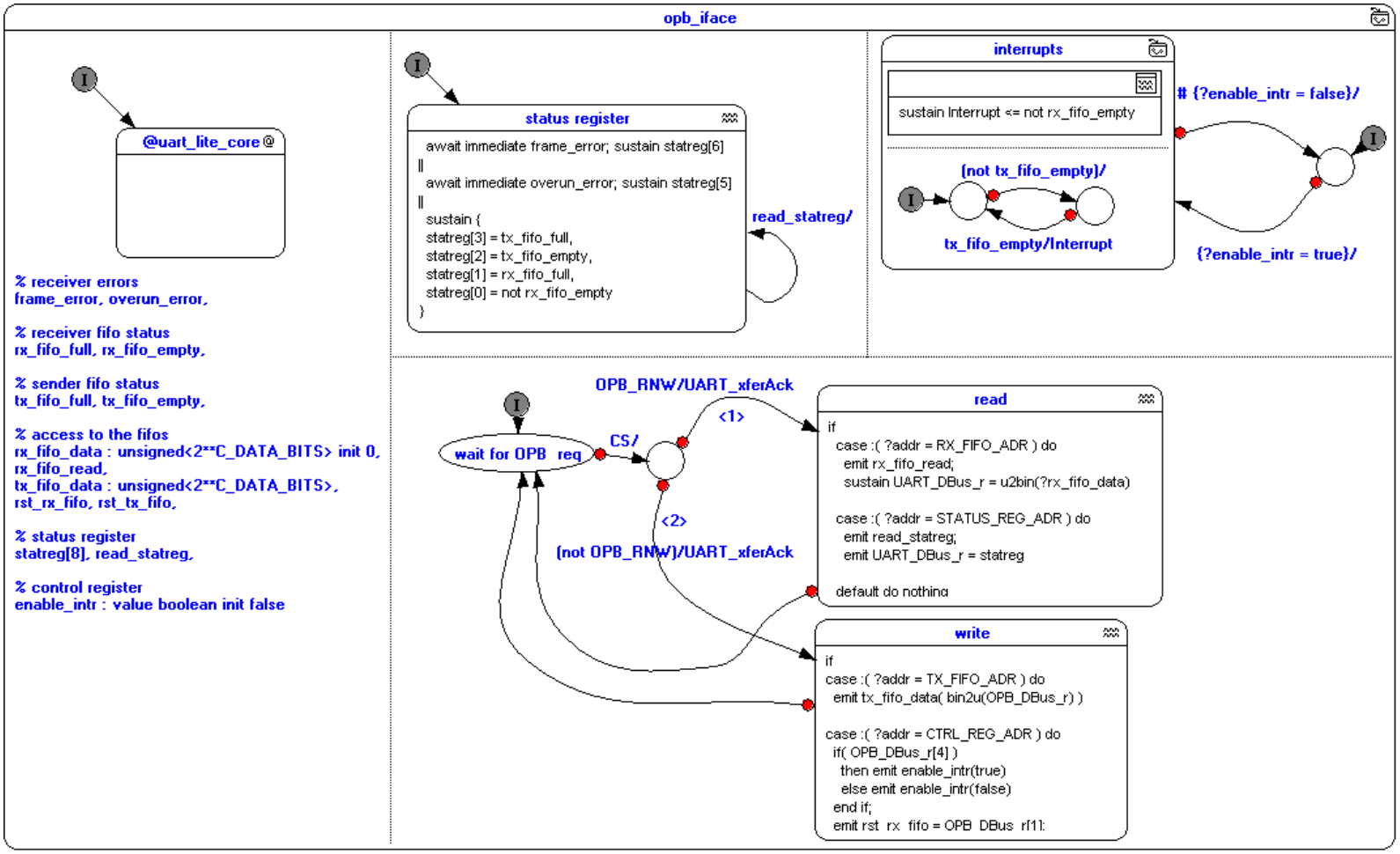
Hardware Synchrony: the RTL model



OK = REQ and GO
PASS = not REQ and GO
GO = TRY or GET_TOKEN
PASS_TOKEN = reg(GET_TOKEN)

Room size control = timing closure

Esterel v7 (Berry – Kishinevsky)



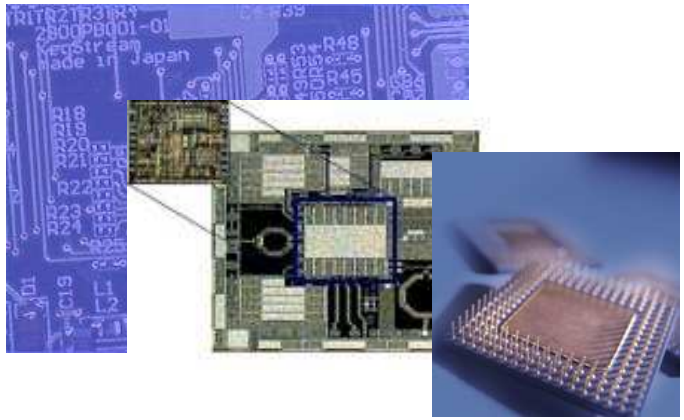
text + graphics, concurrency + sequencing
clear semantics

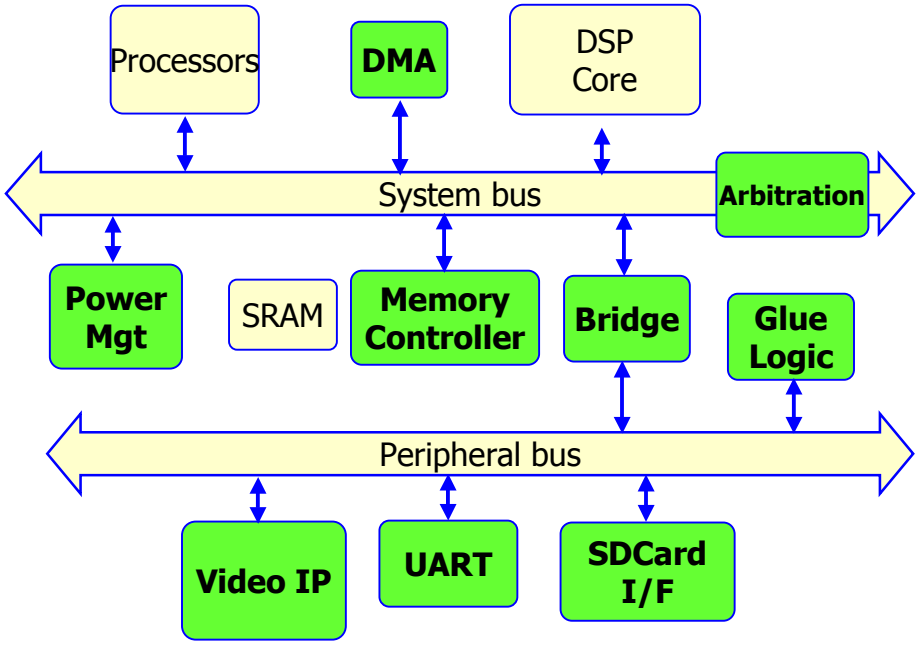
Esterel Customer Consortium

- In 2001 Esterel Technologies formed a consortium of leading Semiconductor companies
  - Strategic involvement
  - Collaborative specification of the requirements
  - Strong influence on roadmap
  - Early adopters of Esterel Studio™
  - Working with Esterel Technologies for IEEE standardization of the Esterel language
- 2005 observers: ARM, EVE, Synopsys

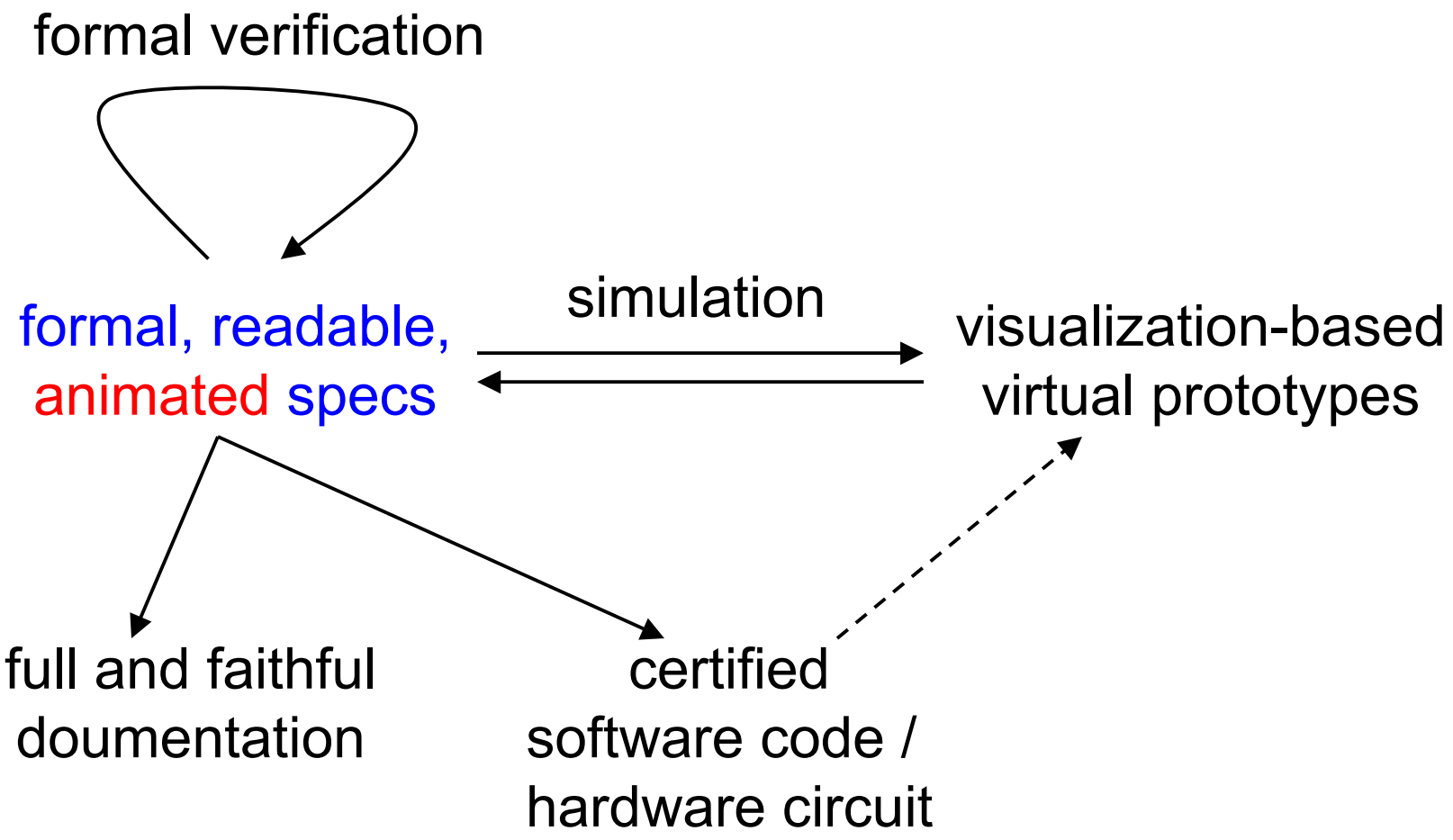
Application Targets

- Bus interfaces and peripheral controllers
  - Bus Bridge
  - Serial ATA
  - Secure Memory Card
  - Video Controller
- Processor core peripherals
  - Complex Instruction and Data Cache
  - Arbiters
  - Complex Power Management
  - DMA
  - Interrupt Controller
- Communication IPs
  - Serial Controller
  - HDLC
  - Fast serial links (UART, Aurora)
  - Bluetooth Call Control
  - Ethernet MAC Controller





The Usage Model



Computer Science at Work

1. Language design & mathematical semantics
   - Esterel: imperative, SOS semantics (residuals)
     constructive logic, proof networks
   - Lustre/SCADE: declarative, functional, denotational semantics
     clock calculus = static type-check of dynamics

2. Compiling
   - Esterel: translation to circuits (hardware)
     translation to concurrent flow graphs (software)
   - Lustre/SCADE: clock calculus to drive expression execution
   - All: static scheduling of elementary actions

3. Formal Verification: properties and equivalence
   - forward / backward reachable state space analysis (BDDs)
   - SAT + numerical solving
   - Abstract Interpretation (Astrée, Cousot)

The Pure Esterel Kernel

nothing	0
pause	1
emit S	! s
present S then p else q end	s ? p, q
suspend p when S	s ⊃ p
p; q	p; q
loop p end	p*
p || q	p | q
trap T in p end	{p} ↑ p
exit T	k, k > 1
signal S in p end	p \ s