Lecture 11
Examples of Z Specifications

David Garlan
Carnegie Mellon University

Models of Software Systems
15-671
Fall 1995

This Lecture

<u>An Industrial Case Study</u>

A Formal Specification of an Oscilloscope

Delisle and Garlan, *IEEE Software* (Sept 1990)

<u>New concepts</u>

- **Bridging the gap between theory and practice**
- **Axiomatic specifications**
- **Higher-order functional specifications**

But First ...
Two Useful Identities

1. $\forall x: T \mid P(x) \bullet Q(x)$

$\equiv$

$\forall x: T \bullet P(x) \Rightarrow Q(x)$

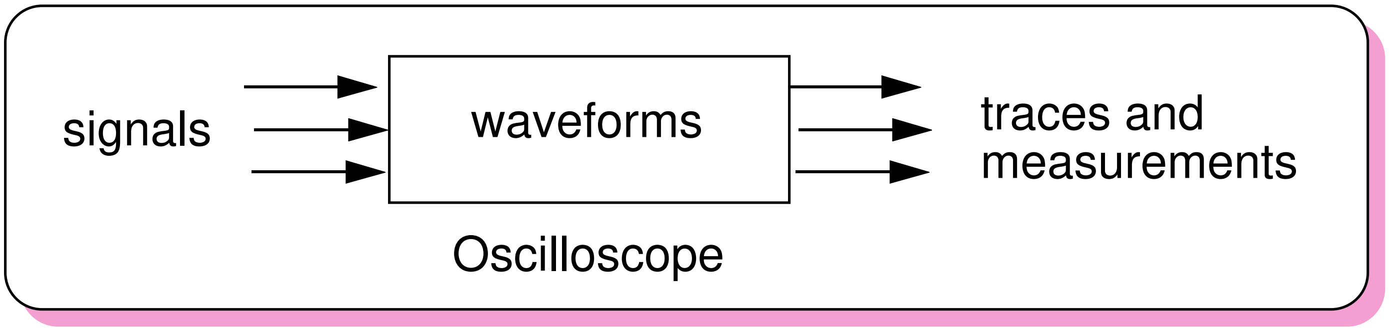
2. $\exists x: T \mid P(x) \bullet Q(x)$

$\equiv$

$\exists x: T \bullet P(x) \wedge Q(x)$

Context

Oscilloscopes are simple in theory



But complex in practice

- Complex software
- Multi-processing environment
- Special hardware
- Sophisticated user interface
- Many specialized modes

The Problem

<u>At Tektronix</u>

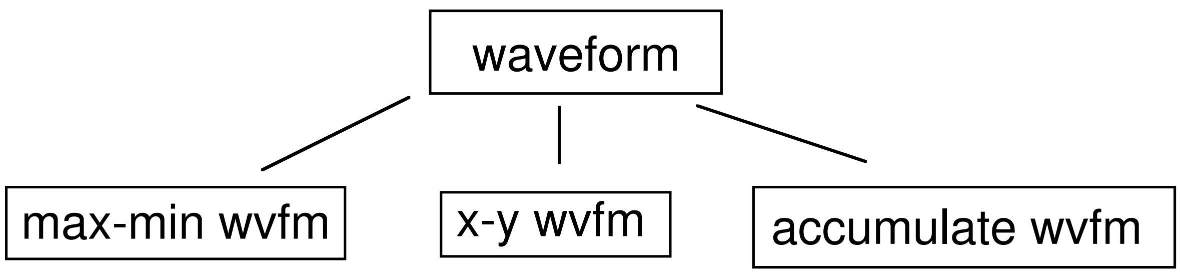
- Many divisions building similar product
- No reuse among product efforts
- Long time-to-market (5 years)
- Each product typically built from scratch
- User interface increasingly important
- Software effort increasingly larger proportion of development costs

Goals

- Build next generation instrumentation systems
- Allow reuse between product divisions
- Support better interactive response to user changes
- Multiple hardware platforms for same user interface
- Multiple user interfaces for same platform
  (vertical markets)

First Attempt

First attempt was an Object-Oriented Architecture

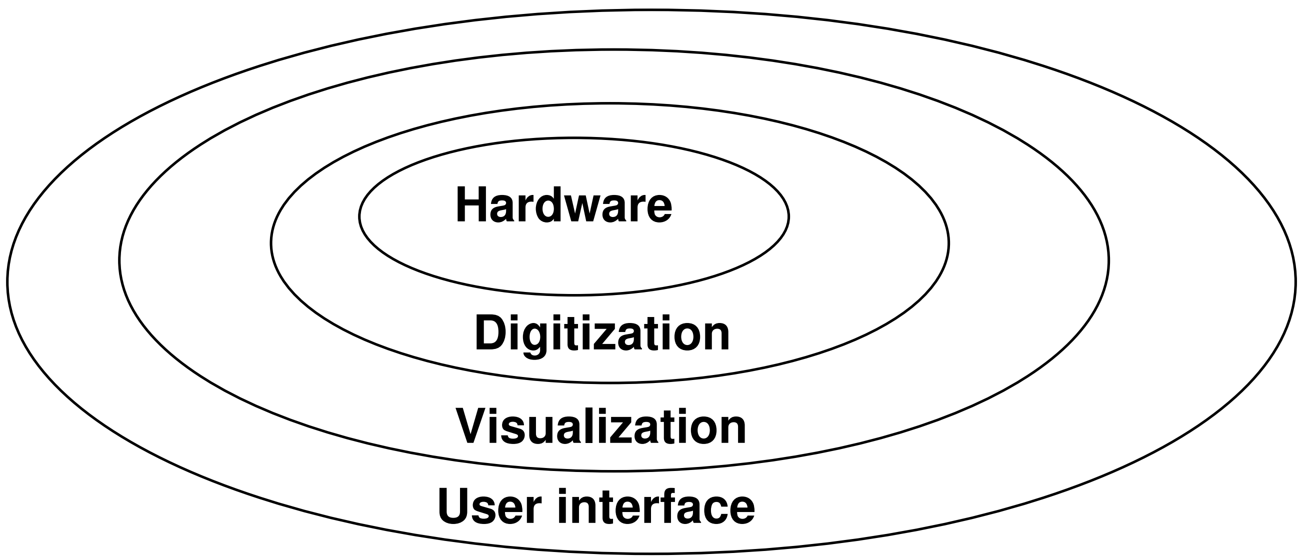


Result:

Hundreds of classes, little structure, no overall pattern

Layered View

Second attempt was Layered Architecture



Boundaries of abstraction not realistic

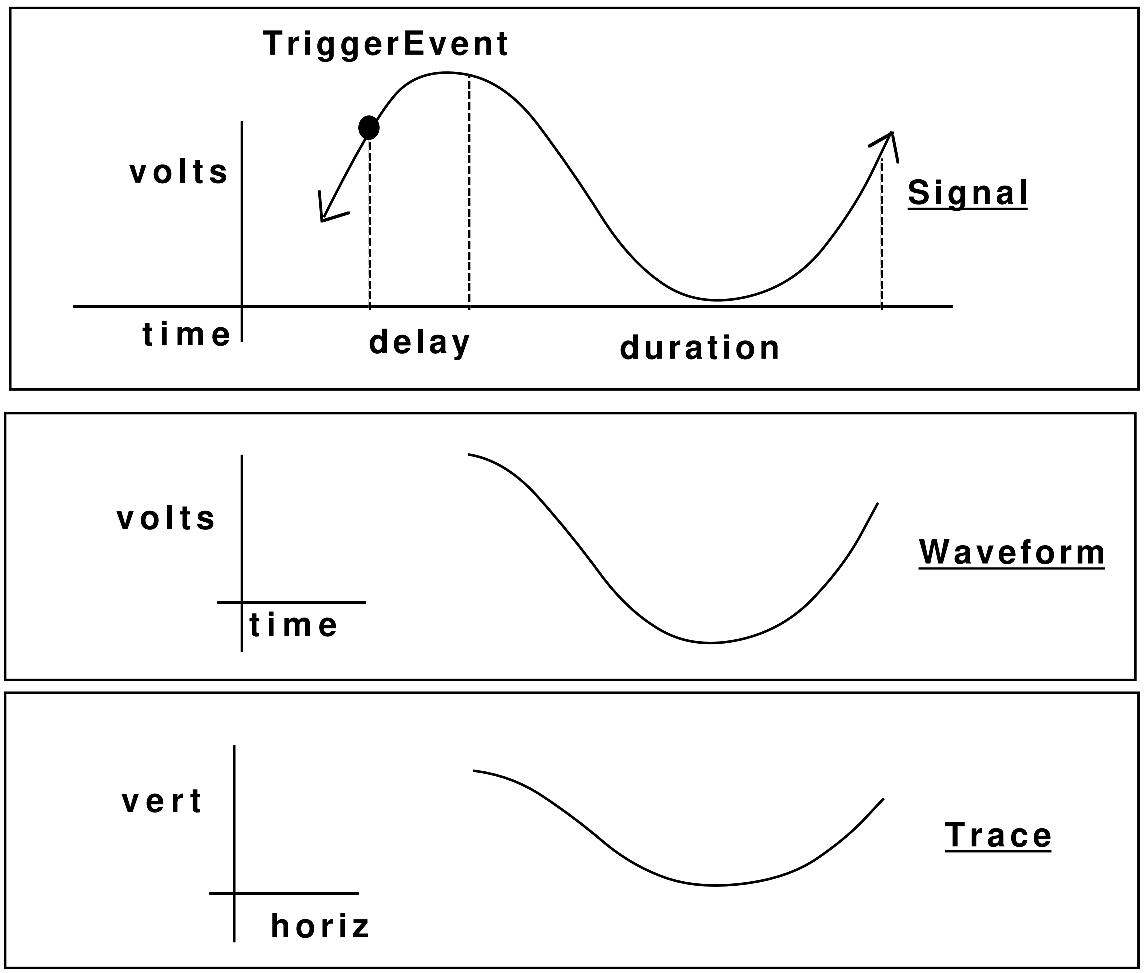
Functional View

Traces = Oscilloscope (Signals)

But:

1. How to handle user input?
2. How can you decompose it into manageable pieces?

Signals, Waveforms, Traces



Basic Types

AbsTime $==$ $\mathbb{N}$

RelTime $==$ $\mathbb{N}$

Volts $==$ $\mathbb{Z}$

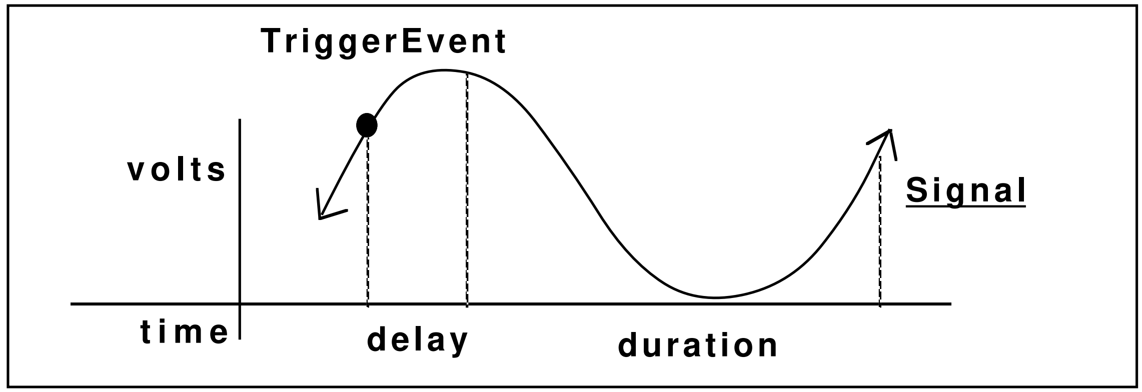
Horiz $==$ $\mathbb{Z}$

Vert $==$ $\mathbb{Z}$

Signals, Waveforms, Traces

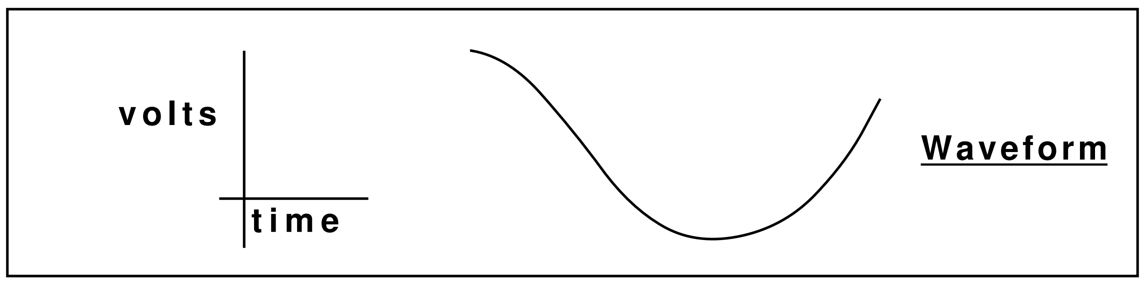
Signal ==

AbsTime → Volts



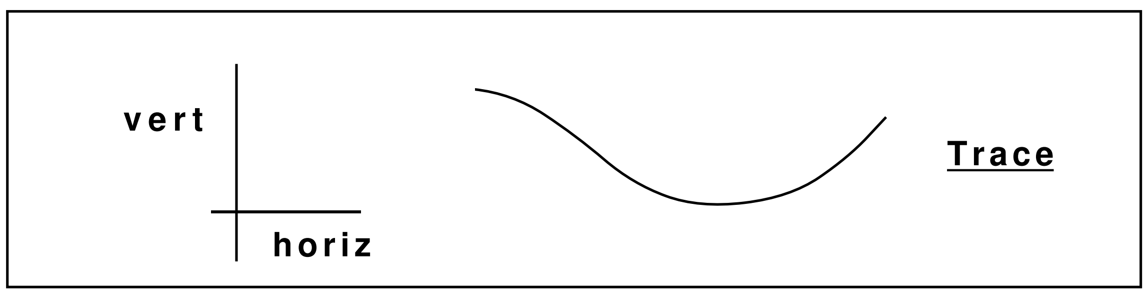
Waveform ==

AbsTime ⇸ Volts

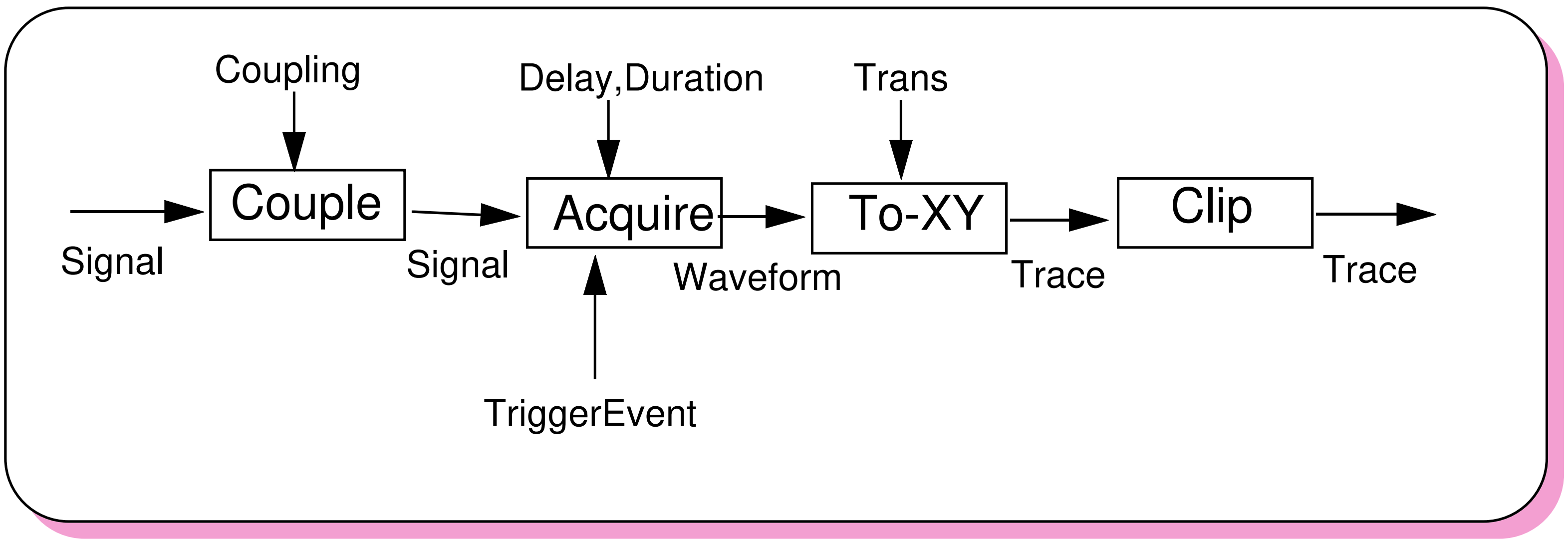


Trace ==

Horiz ⇸ Vert



Channel Subsystem



Coupling

Coupling ::= DC | AC | GND

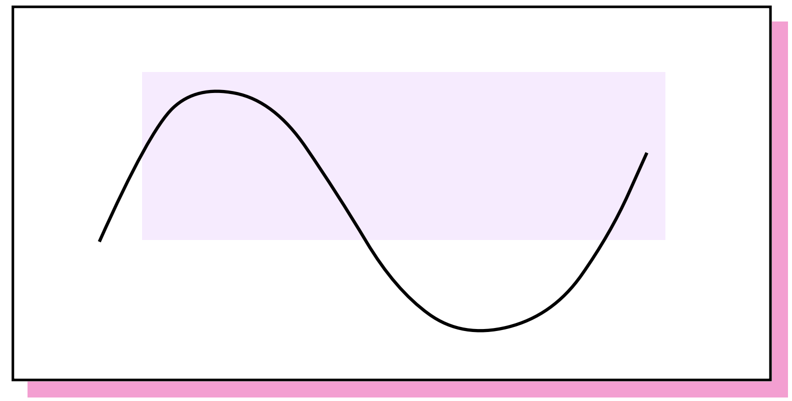
$$
\begin{array}{|l}
\text{Couple: Coupling} \rightarrow \text{Signal} \rightarrow \text{Signal} \\
\text{dc: Signal x AbsTime} \rightarrow \text{Volts} \\
\hline
\text{Couple DC s} = \text{s} \\
\text{Couple AC s} = \\
\qquad \lambda\ \text{t: AbsTime} \bullet \text{s(t)} - \text{dc (s, t)} \\
\text{Couple GND s} = \\
\qquad \lambda\ \text{t: AbsTime} \bullet 0
\end{array}
$$

Acquisition

TriggerEvent == AbsTime

$$\begin{array}{|l} \textbf{Acquire: RelTime x RelTime} \rightarrow \\ \qquad\qquad \textbf{TriggerEvent} \rightarrow \textbf{Signal} \rightarrow \textbf{Waveform} \\ \hline \textbf{Acquire (delay, dur) trig s =} \\ \quad \{\ t: \textbf{AbsTime} \mid \textbf{trig} + \textbf{delay} \leq t \leq \textbf{trig} + \textbf{delay} + \textbf{dur}\ \} \lhd s \end{array}$$

Clip



maxH: Horiz

maxV: Vert

Clip : Trace $\rightarrow$ Trace

Clip tr = (0 .. maxH) $\lhd$ tr $\rhd$ (0 .. maxV)

Channel Parameters

Channel Parameters

c: Coupling

delay, duration: RelTime

scaleH: RelTime

scaleV: Volts

posnV: Horiz

Channel Configuration

ChannelConfiguration: ChannelParameters $\rightarrow$
  TriggerEvent $\rightarrow$ Signal $\rightarrow$ Trace

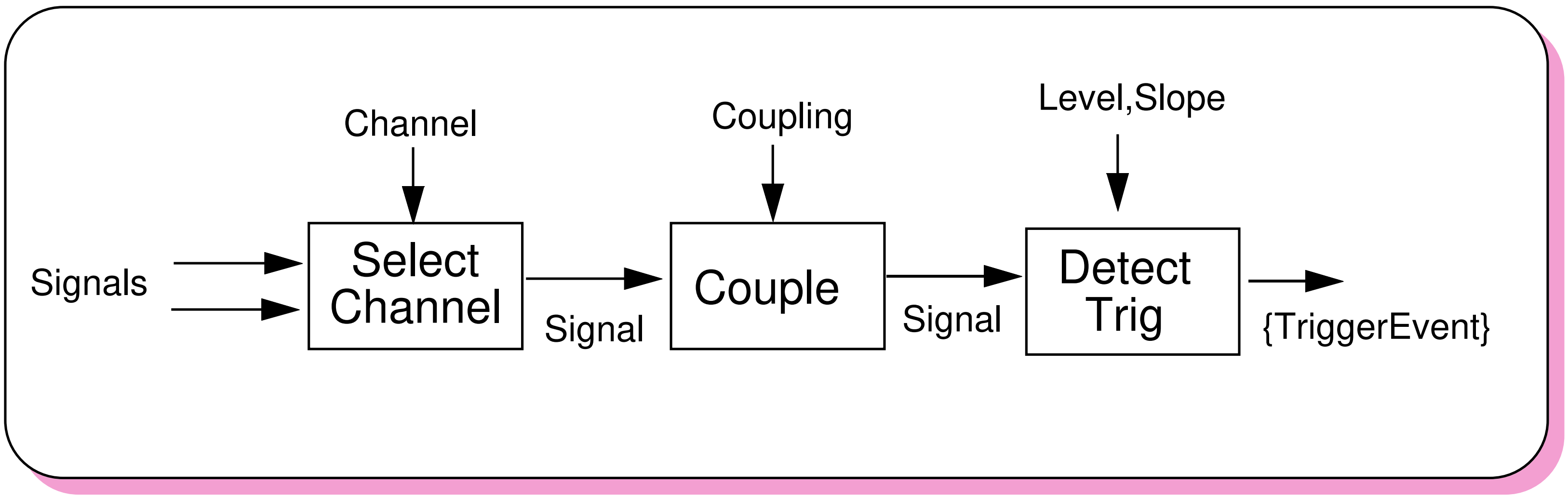
ChannelConfiguration p =

  λ trig: TriggerEvent •

    Clip ¤ WaveformToTrace (...) ¤

    Acquire (p.delay, p.dur) trig ¤ Couple p.c

Trigger Subsystem



Channel Selection

Channel ::= CH1 | CH 2

$$
\begin{array}{|l}
\text{SelectChannel: Channel} \rightarrow \\
\quad \text{Signal x Signal} \rightarrow \text{Signal} \\
\hline
\text{SelectChannel Ch1 (s1, s2) = s1} \\
\text{SelectChannel Ch2 (s1, s2) = s2}
\end{array}
$$

Trigger Detection

Slope ::= POS | NEG

Level == Volts

$$\begin{array}{|l} \textbf{DetectTrig: Level x Slope} \rightarrow \textbf{Signal} \rightarrow \mathbb{P}\ \textbf{TriggerEvent} \\ \hline \forall\ t: \textbf{DetectTrig (l, sl) s} \bullet \\ \quad (sl = POS) \Rightarrow s\,(t - 1) \leq l \leq s\,(t) \\ \quad (sl = NEG) \Rightarrow s\,(t) \leq l \leq s\,(t - 1) \end{array}$$

Trigger Parameters

TriggerParameters

l: Level

sl: Slope

ts: Channel

tc: Coupling

---

tc $\in$ **{AC, DC}**

Trigger Configuration

TriggerConfiguration: TriggerParameters $\rightarrow$ Signal x Signal$\rightarrow$ $\mathbb{P}$ TriggerEvent

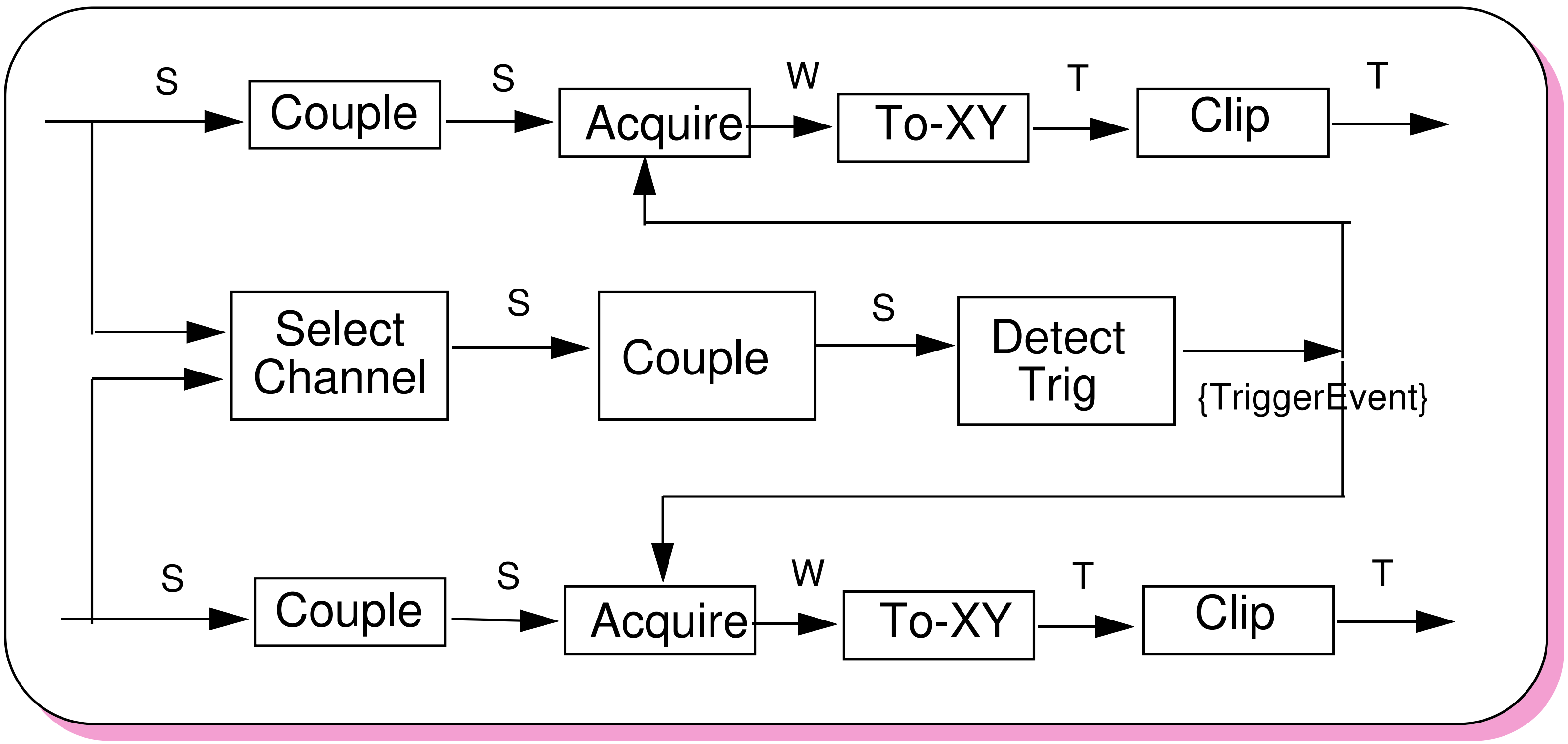
---

TriggerConfiguration p =

DetectTrig (p.l, p.sl) ¤ Couple p.tc ¤

SelectChannel p.ts

The Whole System



Oscilloscope

```
┌─Oscilloscope ─────────────────────────────
│ s1, s2: Signal
│ cp1, cp2: ChannelParameters
│ tp: TriggerParameters
│ ts1, ts2: seq Trace
├────────────────────
│ ∀ t: ran ts1 •
│     ∃ trig: TriggerConfiguration tp (s1, s2) •
│         t = ChannelConfiguration cp1 trig s1
│ ∀ t: ran ts2 • ...
└───────────────────────────────────────────
```

Role of Formalism

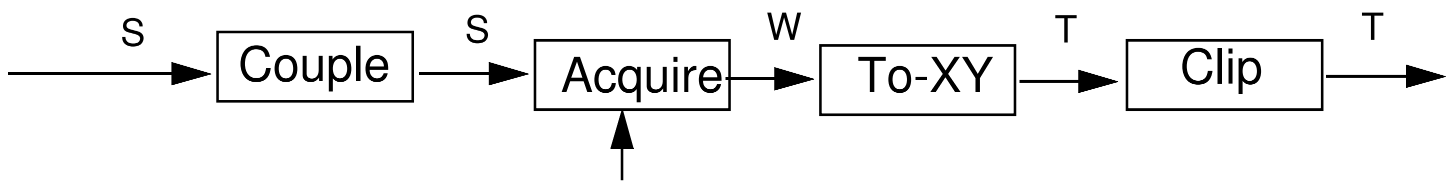
- What was formalized
  - User level model (higher-order P/F)
  - Connection framework (colored P/F)
  - User interface architecture
- Benefits of formalism
  - Motivated and constrained architectures
  - Conceptual prototyping
  - Communication medium
- Non-benefits
  - Correctness
  - Completeness
  - Adoption of formal methods within Tektronix

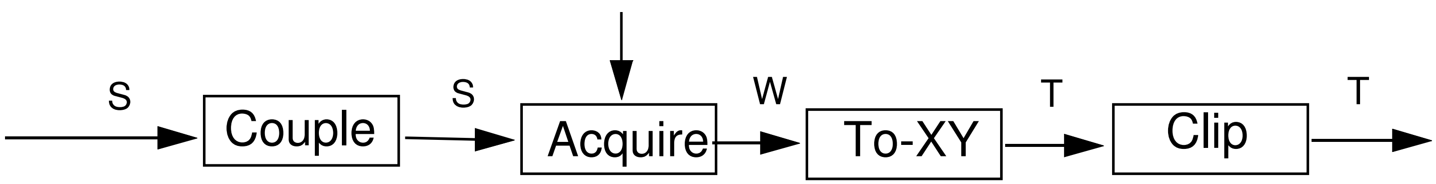
Some Morals

- Success requires
  - Domain experts and system builders
  - Expertise in abstraction
  - Skill in manipulating formal models
  - Willingness to abandon old implementation patterns
  - Willingness to reject inappropriate models
- Tools
  - Not needed during conceptualization
  - Essential during development
- Management
  - Must keep hands off the process initially
  - Must help inforce standards later

Last Slide

The Whole System




S
Couple
S
Acquire
W
To-XY
T
Clip
T