Testing and Debugging

- Logic Probe - very simple but enough for quick test
- Oscilloscope
  - Shows electrical details
    - Benefits: Wideband, accurate
    - Disadvantages: < 4 inputs; triggering
- Logic analyzer
  - Shows 0/1 - according to some threshold
    - Benefits: Many channels, trigger on patterns
    - Disadvantages: Idealized waveforms, insufficient access
- 3. Embedded test
  - *You* design in built-in test features
    - Benefits: Only way to test large chips
    - Disadvantages: Uses chip area, incomplete scan, difficult design

Logic Probe

- Examine one signal
  - Display 0/1/Z/changing
  - Select TTL or CMOS technology (5v)
- Catch pulses
  - Connect to signal, set pulse
  - If pulse occurs, probe triggers and catches it
- Very rudimentary, but quick to catch simple things
  - Unconnected signals (bad protoboard, broken wire)
  - Wrong connections
  - Bad chips

Oscilloscope

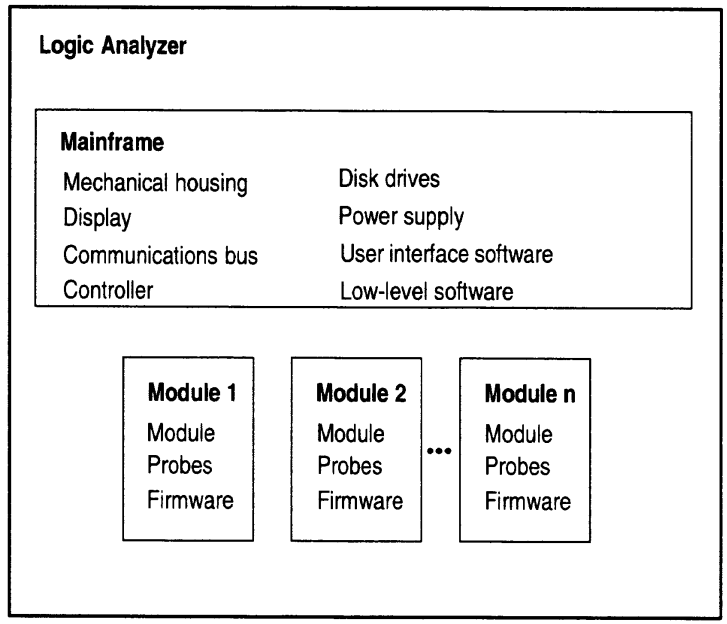
- Samples signal voltage over time
  - Displays signal as a waveform, one voltage value per time step\
- Triggering
  - Choose when to start sampling the signals
  - Slope: rising voltage/falling voltage
  - Threshold: trigger when signal reaches this value
- Repeat mode
  - Assume that signal is periodic
  - Repeated triggering captures the same signal
  - You'll never see a glitch
- Capture mode
  - Triggers only once, stores waveform in memory
  - You'll be very lucky to catch a glitch

Logic analyzers

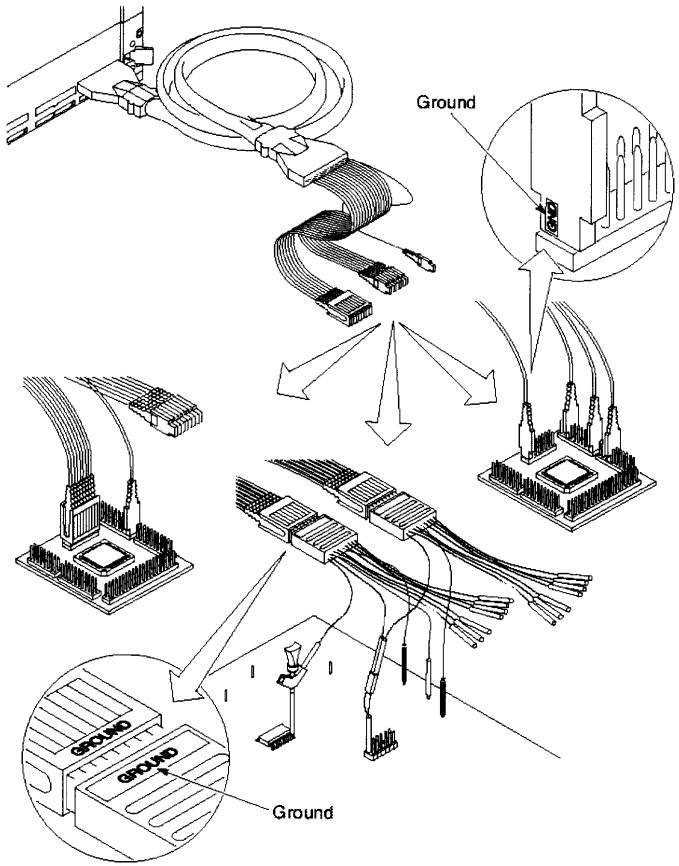
- Instruments for acquiring digital data
  - Wide data “bus” - capture many signals
  - Memory stores bus data
  - Smart triggering decides what data to store
  - Embedded computer processes the data
- We use the Tektronix TLA704
  - 128 channels
  - 32k memory per channel
  - 100MHz state
  - 2GSPS sampling
  - Win95 interface

Logic analyzer physical model

- A mainframe
  - Housing, bus, controller, UI
- Plug-in modules
  - Modules acquire data
  - Ours TLAs have 4 7L1 modules
    - 32k memory depth
    - 100MHz state
    - 32 data and 2 clock each
- Probe pods
  - Pods are wire bundles
  - Probes attach to your circuit
    - We have P6417 probes

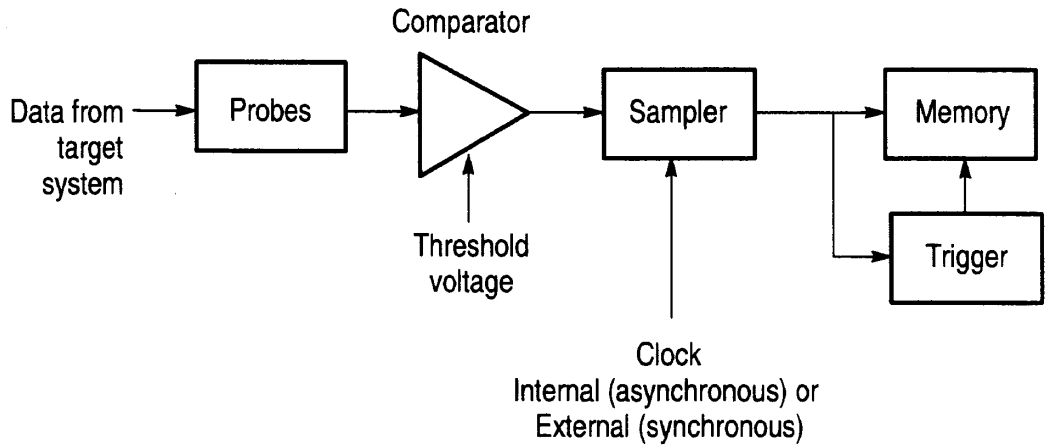


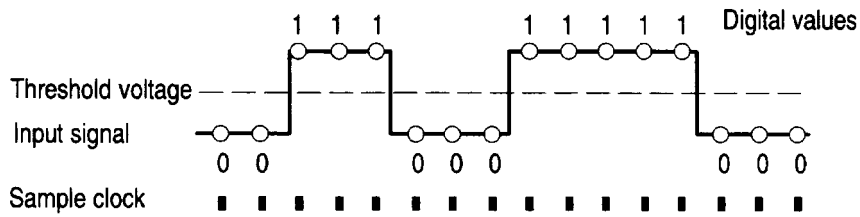
Probe pods



Logic analyzer conceptual model

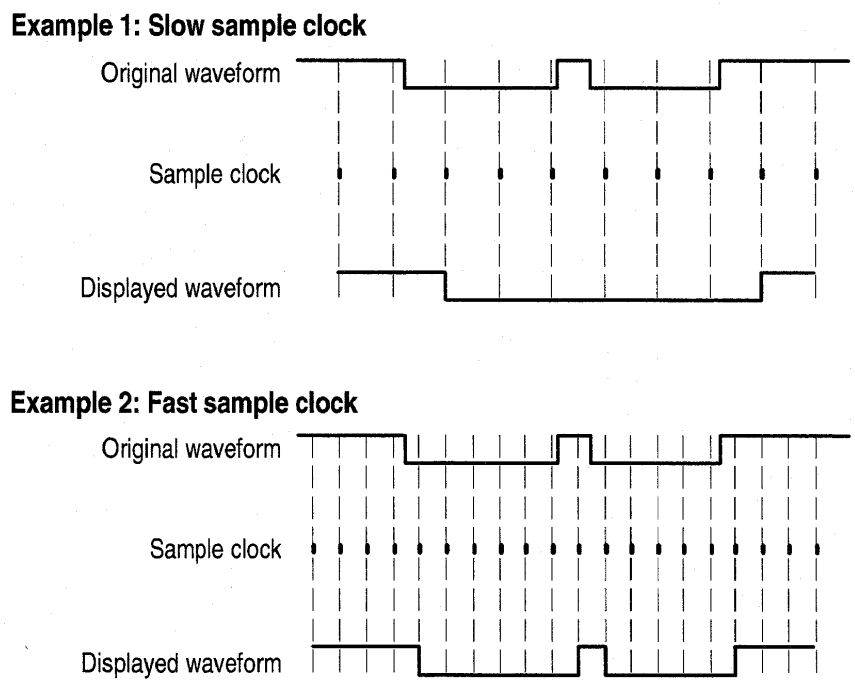
- All parameters are adjustable
  - Threshold voltage
  - Clock rate
  - Trigger conditions
  - Memory depth





Clocking a logic analyzer

- Samples data every clock cycle
- External/synchronous clocking
  - You supply the clock
  - Use when you need to see long data records
    - Analyzer stores one sample per clock period
- Internal/asynchronous clocking
  - Analyzer supplies the clock
    - 4ns to 50ms
  - Use when you need to see precise timing
    - Find glitches



Triggering and acquisition

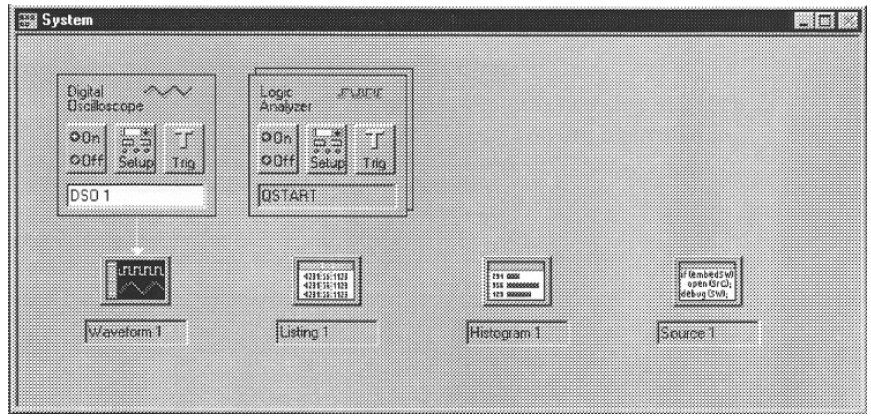
- Derive trigger from sampled data
  - Data values
  - Data ranges
  - Signals from another module
  - Internal counters
- Data acquisition is continuous
  - Memory is a circular buffer
  - New samples continually overwrite oldest samples
  - Trigger tells the acquisition to stop
- Triggers can qualify acquisition
  - Store only selected data

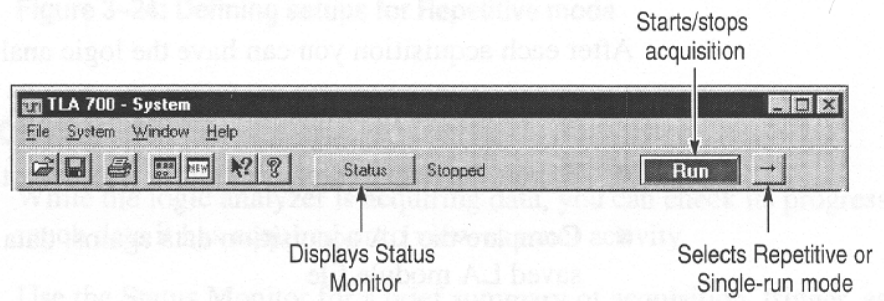
Modules are semi-autonomous

- Each module has its own setup
  - Its own clock
  - Its own trigger
  - Acquires and stores its own data
- Modules communicate via their trigger programs
  - Can trigger all modules
  - Or have one module arm another
- All data is time correlated
  - Regardless of the module

System window

- Top-level in hierarchy
  - Open other windows
    - Module
    - Data
  - Create new data, listing, and waveform windows
  - Shows which modules are associated with a data window
  - Enable/disable modules
  - Save and load files

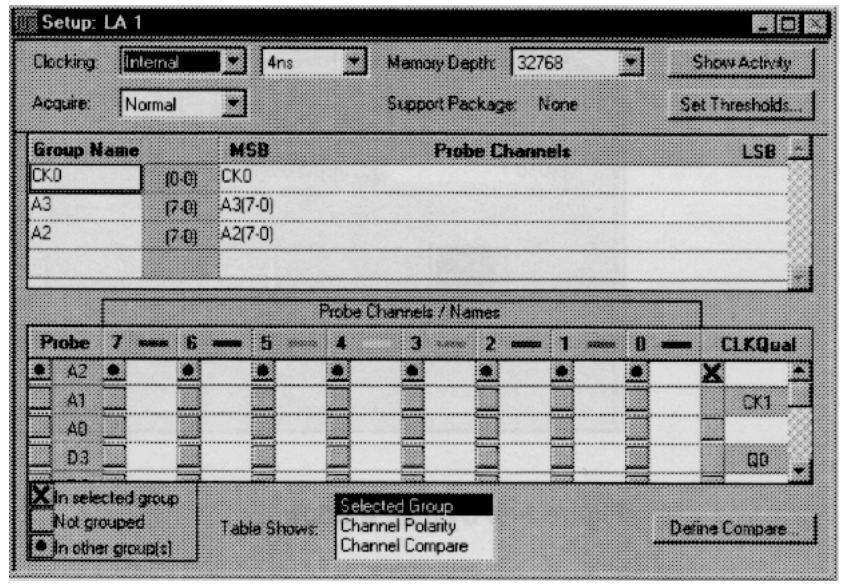




- *Note: We don't have DSO modules

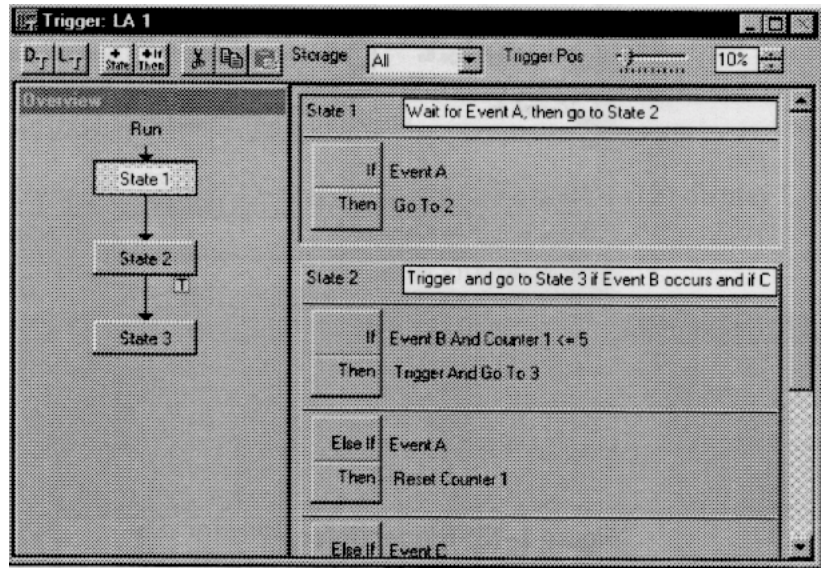
Module setup window

- Each module has its own setup and trigger windows
  - Set up each module independently
- Set all parameters
  - Assign channels to groups
  - Thresholds
  - Clock rate
  - Comparisons
- Configure setup before trigger
  - Trigger settings depend on the module settings



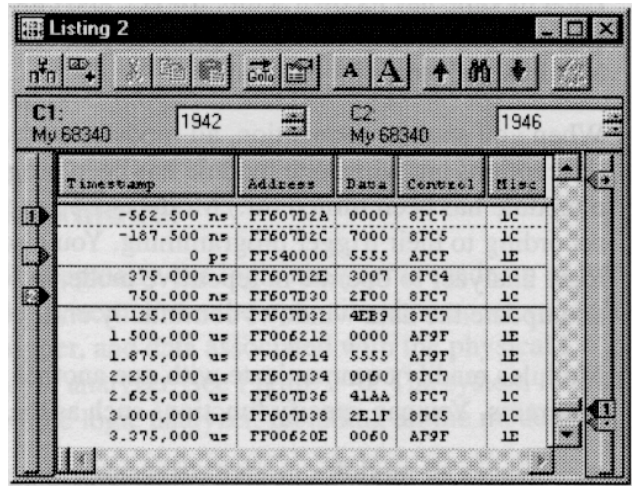
Module trigger window

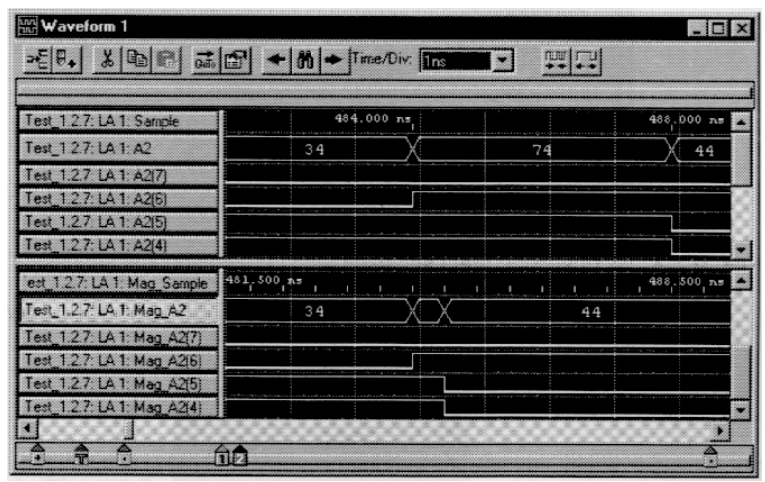
- Triggering is the ***key*** feature of a logic analyzer
  - Tells the analyzer how to find the data that you want
    - Trigger off a data pattern
    - Trigger off a data sequence
  - Multiple states
    - Multiple clauses per state
- Analyzer has a trigger library!
- Logic analyzers are designed for non-repetitive data
  - Unlike an oscilloscope



Data windows

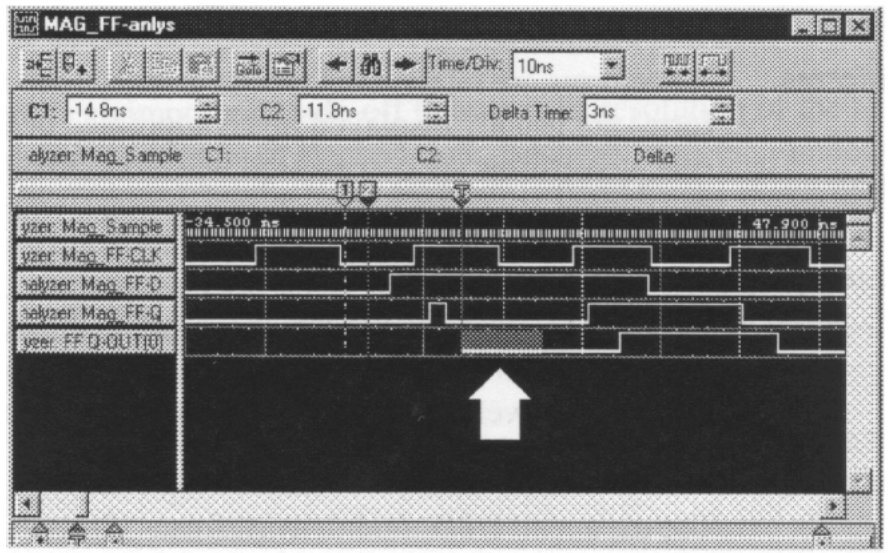
- Many types
  - Listing window
  - Waveform window
  - Histogram window
  - Source-data window
- Features common to all
  - Cursors
  - Flags
  - Scroll
  - Search





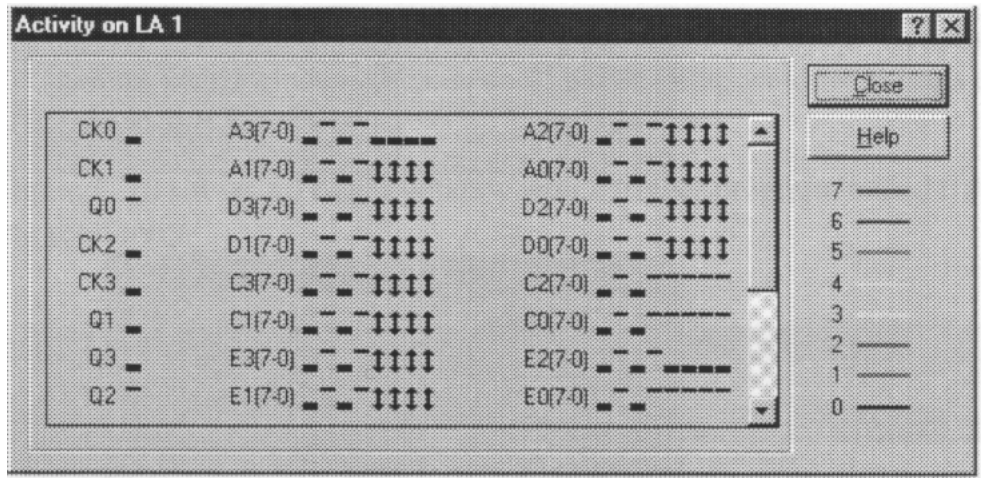
Capturing glitches

- Trigger on the glitch
  - Triggering looks for multiple transitions in a clock cycle
  - Captures dynamic hazards
- Can also trigger on setup and hold violations



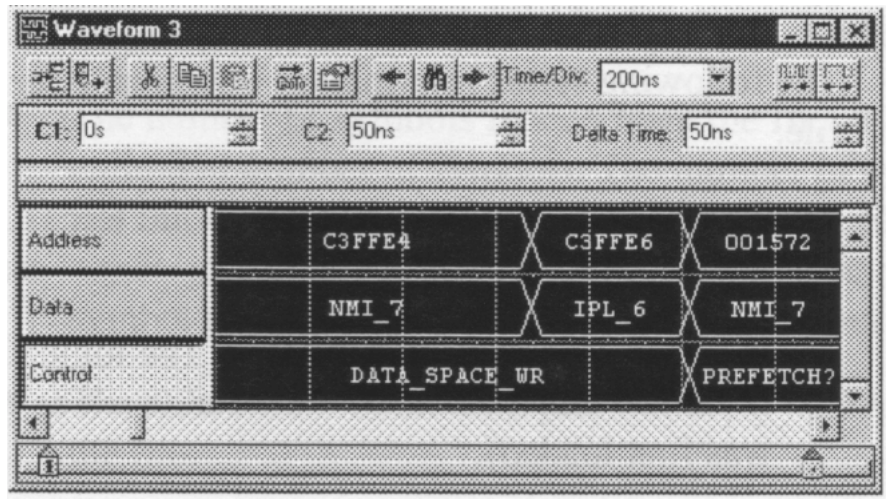
Other features: Activity indicator

- How do you know if a pod is active?
  - Hooked up properly?
  - Seeing data?



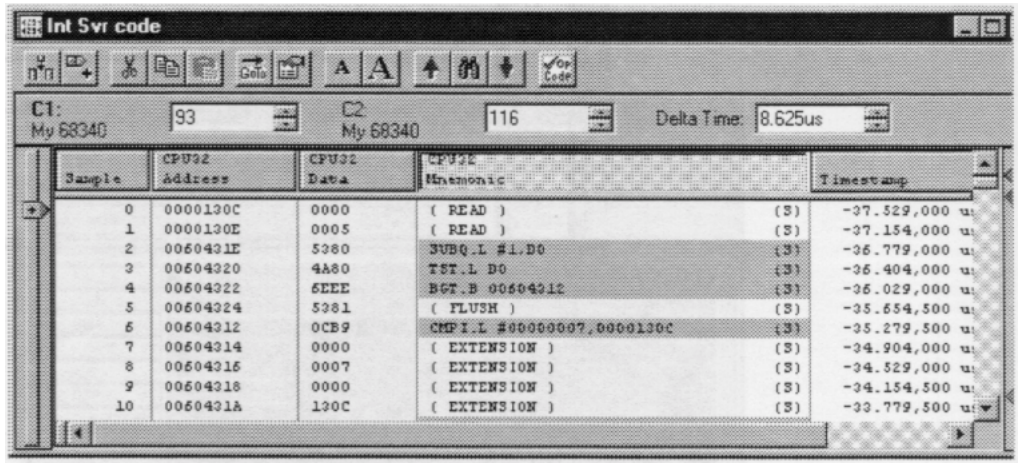
Other features: Programmability

- Symbols
  - You define in a LUT
  - Analyzer assigns symbols to data patterns
- User programs
  - e.g. export to file and continue



Other features: µP support

- Analyzer disassembles data to µP mnemonics
- Requires special module podsets

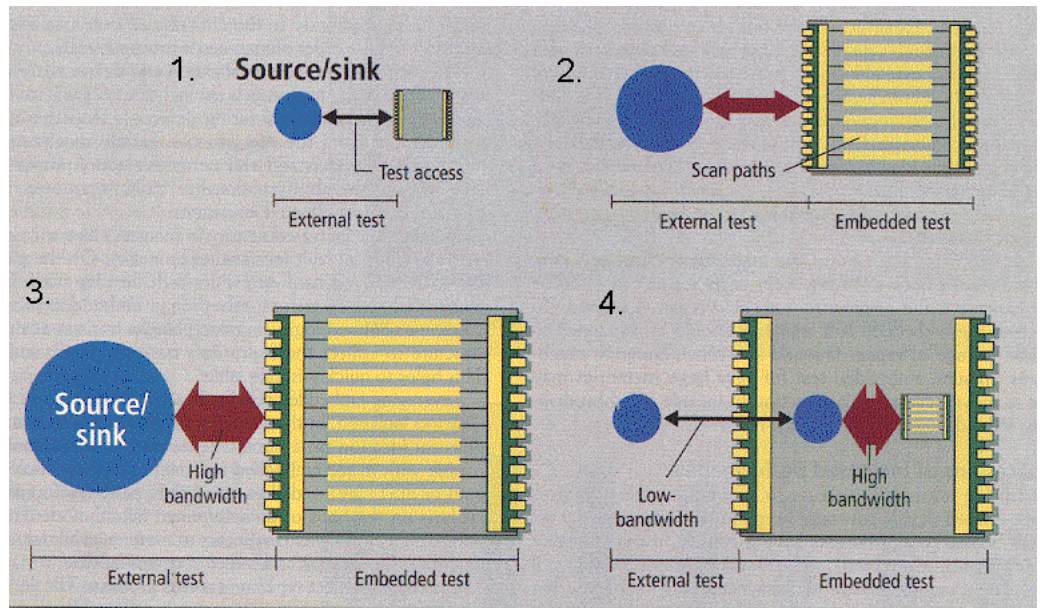


Testing: The big picture

- The difference between internal and external BW has driven test technology
  1. External test
  2. Embedded scan path
  3. High-BW embedded
  4. Embedded source
  5. ???

External BW ≡ (# of I/O) * (external clock rate)

Internal BW ≡ (# of transistors)* (internal clock rate)



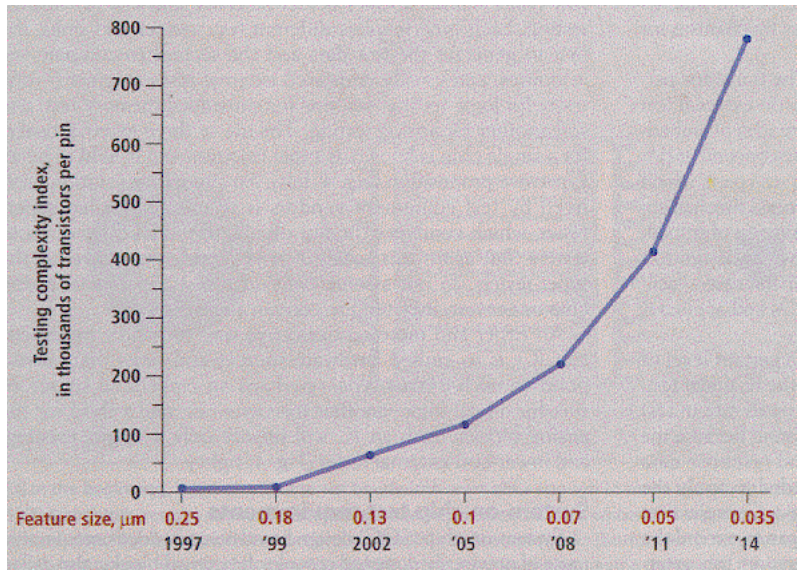
Bandwidth:								
	Internal	1	3.165	19.52	85.26	375.35	1621.62	7132.76
	External	1	1.97	3.59	6.14	10.5	17.24	28.89
		1997	1999	2002	'05	'08	'11	'14

Source: 1998 International Technology Roadmap for Semiconductors

From IEEE Spectrum, 7/99, pgs. 55 / 57

Semiconductor scaling confounds testing

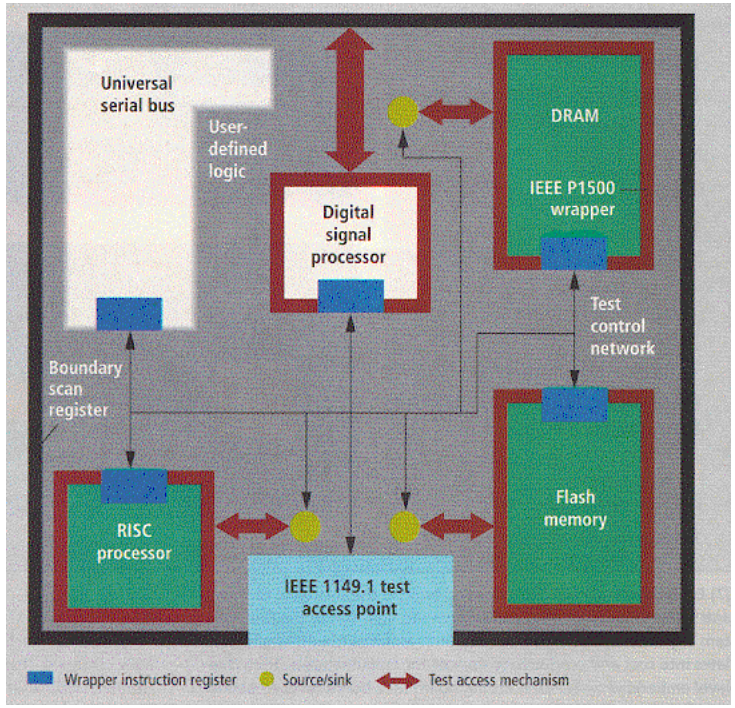
- Testing is a key obstacle to future advancement in digital technology
- Need to ensure logic functionality
  - Despite ever-more-limited access to internal logic
- Testing at the board, subsystem, and system level becomes ever harder



From IEEE Spectrum, 7/99, p. 55

System-on-a-chip testing

- IEEE P1500 standard
  - For embedded core test
  - In development
- Standardized core test language
- Standardized core test wrapper
  - Configurable
  - Scalable



From IEEE Spectrum, 7/99, p. 59