AC EXTEST Preliminary Specification

Agilent Technologies

05/11/2001

1 Introduction

This document describes the details of a solution for testing AC-coupled nets between integrated circuits on printed circuit boards or in systems. Traditionally, such interconnect has been DC-coupled and can thus be tested with well-known Boundary-Scan methods, specifically with the EXTEST instruction as codified in IEEE Std 1149.1. However, the use of AC-coupling capacitors disallows DC-based techniques, giving rise to the need for an extension of Boundary-Scan into the AC realm.

There are several possible technical solutions to enable AC Boundary-Scan testing. The method herein described is based on the use of frequency discrimination, which is robust and scalable. Section 3 introduces the related concepts, then the necessary elements of an AC Boundary-Scan solution are illustrated: modifications to the PC board (Section 4), additions of test circuitry to the integrated circuit chips (Section 5), and a test application procedure (Section 6). Section 7 provides a specification for AC Boundary-Scan. Section 8 provides conformance and documentation requirements. Section 9 shows an example of interconnected ICs containing this new technology.

Table of Contents

2 References

This document shall be used in conjunction with the following standards. When the following standards are superseded by an approved revision, the revision shall apply.

IEEE Std 100-1996, IEEE Standard Dictionary of Electrical and Electronic Terms.

IEEE Std 1149.1-1990 (includes IEEE Std 1149.1a-1993), IEEE Standard Test Access Port and Boundary-Scan Architecture.

IEEE Std 1149.1b-1994 Supplement to IEEE Std 1149.1-1990 IEEE Standard Test Access Port and Boundary-Scan Architecture – Annex B: Boundary-Scan Description Language.

IEEE Std 1149.4-2000 IEEE Standard for a Mixed Signal Test Bus.

3 Technology

The presence of coupling capacitors on chip interconnects, whether they are discreet devices mounted on a PC board or integrated inside an IC, prevents DC values from being driven between chips. Any AC Boundary-Scan methodology must therefore use a time-varying signal to pass through the capacitors when in test mode.

3.1 Signal Pin Types

It is expected that a chip possessing high-speed pins requiring AC coupling will also possess "normal" (i.e, DC coupled) pins as well. These DC pins would supply data and/or control to/from lower-speed portions of the chip. For test purposes, it is necessary that all these pins be tested simultaneously with an EXTEST-like capability because that is how shorts (unwanted connectivity) between these pins are reliably detected. This document will refer to DC and AC pins henceforth. DC pins are those that IEEE Std 1149.1 currently governs for testing. AC pins have been treated by 1149.1 as an "analog" problem and effectively ignored.

3.2 Test Signal Types

For AC pins, there are three time-varying signal options:

- Timing-based patterns: a known pattern is driven from one chip and received in another chip. There must be synchronization between the driver and the receiver so that the correct transfer of the pattern can be verified, implying that the two chips must share a common clock reference.
- Fixed frequencies: a known frequency (or set of frequencies) is driven from one chip and detected in another chip. The two chips must share a common reference[1] frequency.
- Single transition edges: an edge is driven from one chip and detected on another chip. No external references are required.

In the first two cases, the minimum frequency must be high enough to pass through the high-pass filter defined by the coupling capacitor and termination network. In the third case, the edge rise/fall time must be sufficiently fast and the edge-detectors must be immune to noise sources.

The need for a skew-matched synchronous clock in the timing-based pattern approach implies the need for careful system design, placing an upper limit on the frequency at which a test can be made to work reliably, while the size of the capacitors places a lower limit on the operational frequency used for test. As technologies scale and when capacitors become smaller with higher frequencies and/or integration of capacitors on-chip, these two limits can become difficult to observe with sufficient guardband.

3.3 Test Signal Implementation

High-speed signals are often transmitted as a differential pair. The insertion of test hardware can be performed in four combinations on a differential pair:

[1] It is possible to envision a "fixed frequency" approach where the global frequency is a specified constant that can be created on-chip by a local oscillator. However, this approach would not be scaleable into the future when test frequencies may need to rise to account for smaller coupling capacitors. It also would require an accuracy specification for the on-chip frequency generator that could be challenging to implement.

- Dual single-ended drive, dual single-ended receive
- Dual single-ended drive, differential receive
- Differential drive, dual single-ended receive
- Differential drive, differential receive

The testability and diagnosability is maximized in the first case as it effectively converts a differential problem back into the familiar, fully single-ended problem. Noise generation and power distribution problems are minimized in the differential drive cases. The best compromise is the third choice. Note that the last case (pure differential) has very poor testability. In some cases, for example it has been shown that *all* open solder joint faults and missing capacitors will be undetectable in a fully differential implementation.

3.4 The Frequency-based Scheme

The most scalable and robust scheme is to use a frequency-based signal sent from a differential driver and captured in dual single-ended receivers. In order to detect defects on manufactured assemblies and to maintain compatibility with existing DC Boundary-Scan, a set of two signal frequencies are used, one corresponding to a digital zero and the other corresponding to a digital one. A receiver of these frequencies discriminates the two based on a derived reference frequency. These frequencies are derived from a common global frequency that may be distributed to all the chips on a board or system that require AC Boundary-Scan. Internal to each chip are clock divider circuits that generate the two data and discrimination frequencies.

One of the most important features of this approach is that it is *phaseless*, meaning there is no need for any phase relationship between a frequency transmitter and any downstream receiver(s). This greatly relaxes practical concerns on the production and distribution of frequencies, eliminating for example any need for skew control. Indeed, two portions of a system separated by considerable distance could be tested, without concern for synchronization of test events, other than the test clock (TCK) distribution itself.

4 Board and System-level Context

The board-level implementation is shown in Figure 1. In one case, the global frequency signal f_g is supplied externally and in the other, there is a local oscillator generating f_g. Note that there are few routing restrictions on the global frequency signal, f_g (shown with a dotted line), since the phase of this signal is irrelevant. Further, the global frequency signal could be a sinusoidal wave that makes board layout much less restrictive. The nets that are tested are shown in solid lines. Interconnect nets A and D are tested with DC Boundary-Scan. Note that the tester and/or fixture will have to provide frequency generation and discrimination capability for nets such as C if AC_EXTEST is used to test on/off board signals. AC_EXTEST is used to test AC coupled signals such as B.

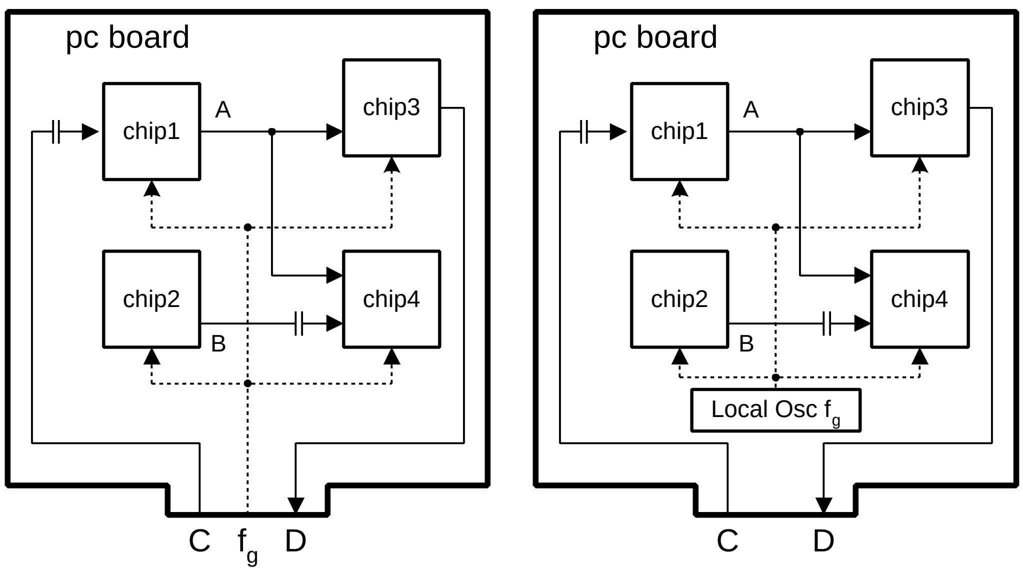

Figure 1: Printed circuit boards with SerDes interconnect and frequency reference *fg* distribution. In one instance, the frequency reference is provided by an external source. In another, the reference is generated on-board.

For system testing multiple boards connected by a backplane or cabling, there are two approaches. First, as in Figure 2, the external oscillator concept can be extended to provide the global frequency from the backplane. This has the advantage that the frequency can be chosen to match system testing requirements should those requirements change in time, for example, when there is a mix of older and more modern boards. The older boards may be testable with lower frequencies, but the newer boards may be using smaller coupling capacitors and require higher test frequencies.

However, since there are no phase relationship requirements for frequency-based testing, the scheme shown in Figure 3 will also work. Each board contains its own frequency generator. As long as the generators are all providing the same frequency (plus or minus a reasonable tolerance) then a frequency generated on one board can be discriminated on another board. There is no need for phase locking. This has the advantage of not requiring frequency distribution across the entire system but must require agreement on the test frequency among all board designers.

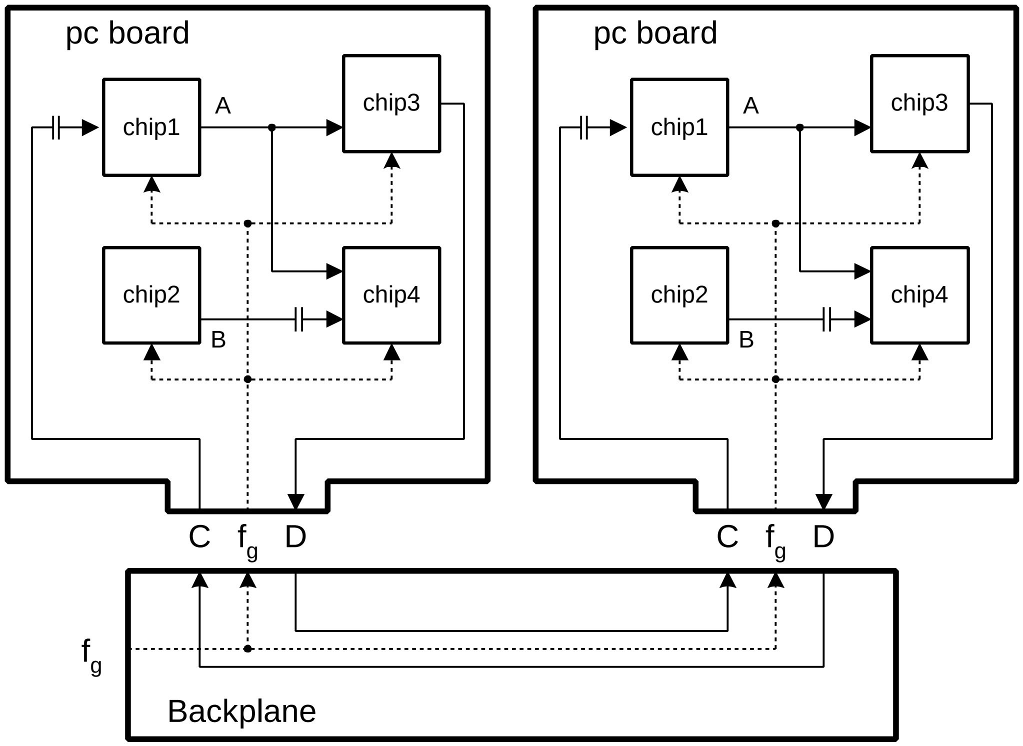

Figure 2: Printed circuit boards connected with a system backplane or cabling can be tested with a single global reference frequency.

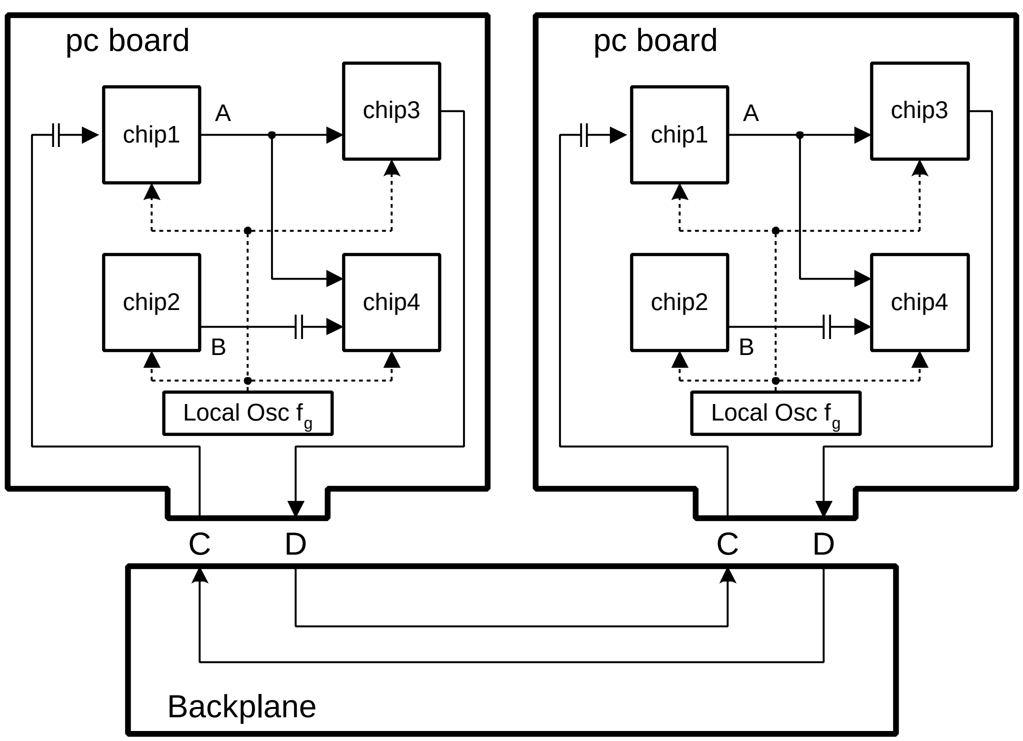

Figure 3: Printed circuit boards connected with a system backplane or cabling can be tested with local oscillators, provided they each produce the same frequency.

5 On-chip Hardware

There are three basic modifications that must be made to the silicon to support AC EXTEST: the Boundary-Scan cells themselves must be modified, a frequency generation block must be added, and the TAP must be extended to support an AC EXTEST instruction. The following sections show a possible implementation that abides with the rules presented in section 7. Other implementations are possible as well.

5.1 Boundary-Scan Cells

5.1.1 Cell Architecture

The Boundary-Scan cells for both the drivers and receivers must be modified. A composite drawing is shown in Figure 4 for a differential pathway.

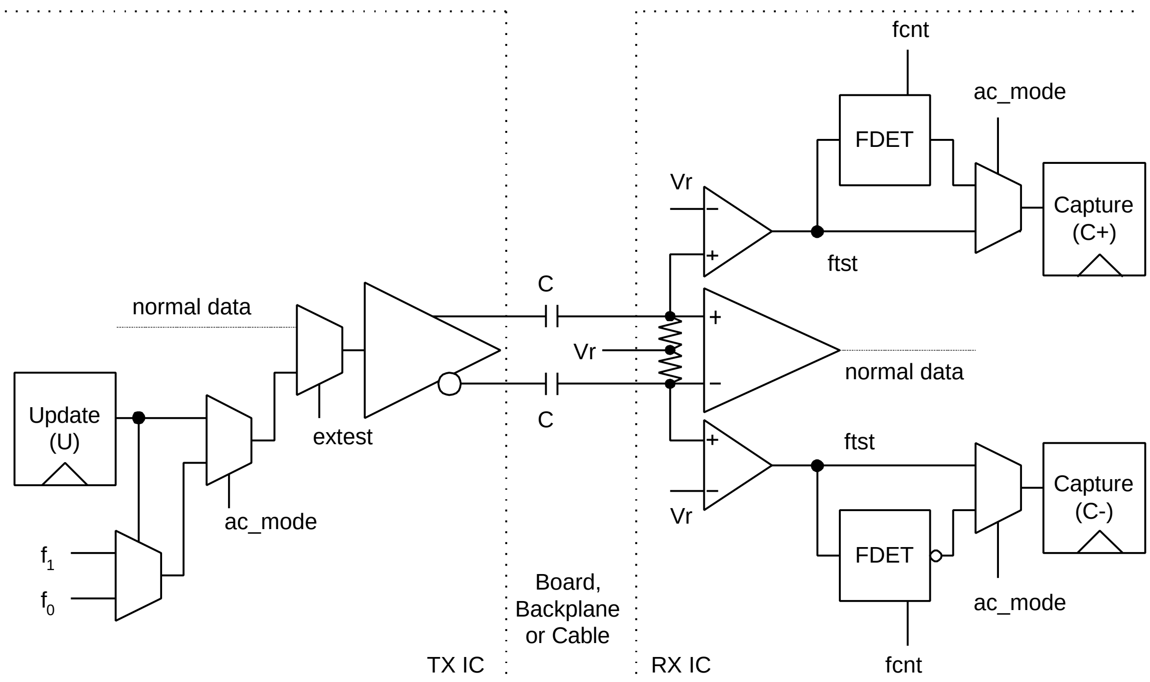

Figure 4: Frequency-based EXTEST circuitry on a differential AC path, drive and receive.

Several features of the circuit should be noted. In both ICs, both the mandatory (DC) EXTEST instruction and the new AC_EXTEST instruction are supported on AC pins. If the *AC_mode* control signal is low during the EXTEST instruction, then the value in the Boundary Register cell *U* is driven out of the chip on the differential pair. If the *AC_mode* control is high, then either f_1 or f_0 (the frequencies corresponding to a 1 and a 0) are driven out, depending on the value in the Boundary-Scan register cell *U*. On the receiving chip, two additional input buffers (similar or even identical to that used in the normal design, but with one leg tied to the bias reference voltage *Vr*) are used. In (DC) EXTEST mode, the received value is latched directly into the Boundary Register register cells (*C+* on the positive leg and *C-* on the negative leg). When the *AC_mode* control is high, the outputs of the frequency detectors (*FDET* blocks) are latched into the Boundary Register cells.

Note that for (DC) EXTEST (assume no blocking capacitors are present) a single bit in the driver side *U* register will propagate to two bits in the receive side, *C+* and *C-*, and that these received bits will have the opposite parity. Since both wires have opposing polarity, any short will interfere with signal propagation.

Next note that for AC_EXTEST, an analogous event occurs. A given frequency is propagated from *both* sides of the driver, but are 180 degrees out of phase (which could detect a short). On the receive side the two detectors, which are phase insensitive, will both perceive the same transmitted frequency. However, on the negative leg the output of the frequency detector is inverted before being captured. Thus for AC or DC EXTEST functions, data on the transmit side is propagated to the receive side with opposite polarities being captured in the *C+* and *C-* cells and in both cases, a short between the legs will cause errors in the received data.

5.1.2 Frequency detector

A typical digital frequency detector circuit (*FDET* block) is shown in Figure 5.

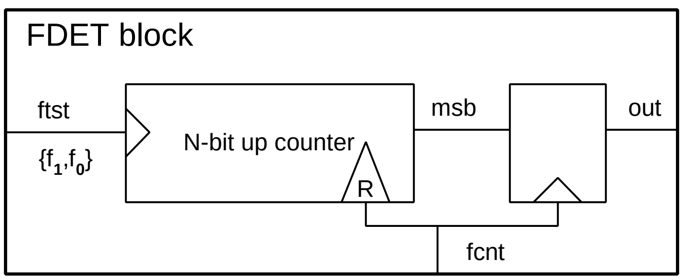

Figure 5: Digital frequency detector based on a digital counter.

The input to this circuit is the output of the test input buffer *ftst*, which will be (in a defect-free board) either the frequency f_1 or f_0. A derived clock *fcnt* (see the next section) is used to reset a counter and sample the most significant bit (*msb*) of that counter into a flip-flop. (Note the reset is edge-triggered, and there is zero hold time on the sample flip-flop. This circuit is easily described in RTL and is synthesizable.) The example shown uses a N-bit counter[2] where *fcnt* is derived according to *fcnt = fref / 2^ (N – 1*) (see section 5.2). Smaller values of N offer a shorter discrimination cycle time. Larger values of N offer more noise immunity.

In the presence of board manufacturing defects, a test signal *ftst* may arrive in some distorted condition at the input of a frequency detector. The detector is capable of distinguishing f_1 from f_0, but the distorted *ftst* may be neither of these. In fact, many defects (such as opens or stuck-at faults and some shorts) will manifest themselves as no signal at all. In these cases, the frequency detector must produce a deterministic and consistent output, such that a sequence of tests that expect both 0s and 1s will detect a failure in one of the two states.

5.2 Frequency Generation

The frequency-based method uses one global, free-running clock frequency *fg* to derive four frequencies: f_1, f_0, *fref*, and *fcnt*. The frequency generation circuit is shown is Figure 4.

[2] This counter may be implemented as a simple ripple counter since the frequency detection process is phase insensitive. The delay caused by carry ripple will not affect the result.

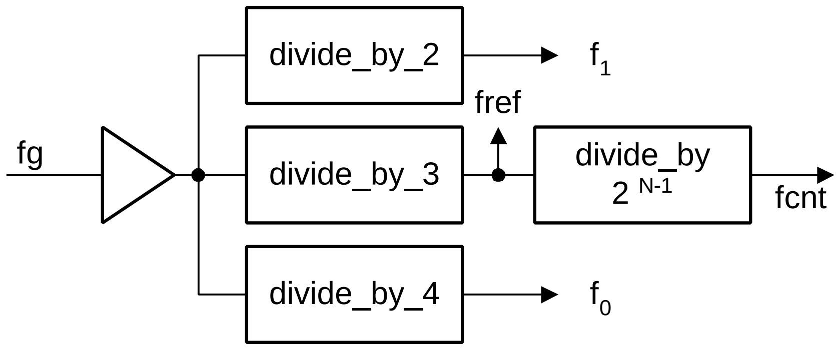

Figure 6: Frequency generator, shared by all AC EXTEST I/O circuits.

A salient feature of this scheme is that f_1 and f_0 are relatively prime with respect to *fref*, and the fact that $f_0 < fref < f_1$, which allows the frequency detectors to clearly differentiate a frequency higher or lower than the reference frequency and thus unambiguously map them into a 0 or a 1. Note also that f_0 and f_1 are both the result of dividing by a power of 2, so any non-square duty cycle on *fg* will be eliminated, resulting in nearly perfect square waves[3] for f_0 and f_1. Another important feature is that *fcnt* is generated by dividing[4] *fref* by a power of 2 (with *fref* and *fcnt* also relatively prime) where the power of 2 is ½ that of the up-counter in the FDET block. Note that this clock generation circuit can be shared for all the AC Boundary-Scan pads on a chip.

The *fcnt* signal is used to periodically reset the frequency detector counter (to define the starting point of a discrimination cycle) and to capture the most significant bit of the counter to indicate whether f_1 or f_0 has been detected in a discrimination cycle. After the first discrimination cycle completes, the *out* signal from the frequency detector is valid. During testing, a given frequency detector will (at times) be connected to f_1 or f_0.

Figure 7 shows the performance of two detectors that are discriminating each frequency. The regions marked “zero hold time” show where the detector must capture the msb of the up-counter as the counter is being reset. Figure 7 is shown for analysis only. In reality, the two data frequencies would be generated in a first IC and the discriminators and a second *fcnt* generator would exist in a second IC as seen later (in Figure 12). The performance characteristics would be the same.

5.3 TAP

The TAP modifications are straightforward. An AC_EXTEST instruction must be added. Since this instruction involves operation for a number of *fg* clock cycles, it must remain active during the Run-Test/Idle state. This instruction asserts both the *EXTEST* and *AC_Mode* signals seen in Figure 4.

[3] This squareness means that there will be no DC offsets on the far side of the blocking capacitors filtering these waves.
[4] This divider can also be implemented as a ripple counter since the injected delay is unimportant in this phase-insensitive scheme.

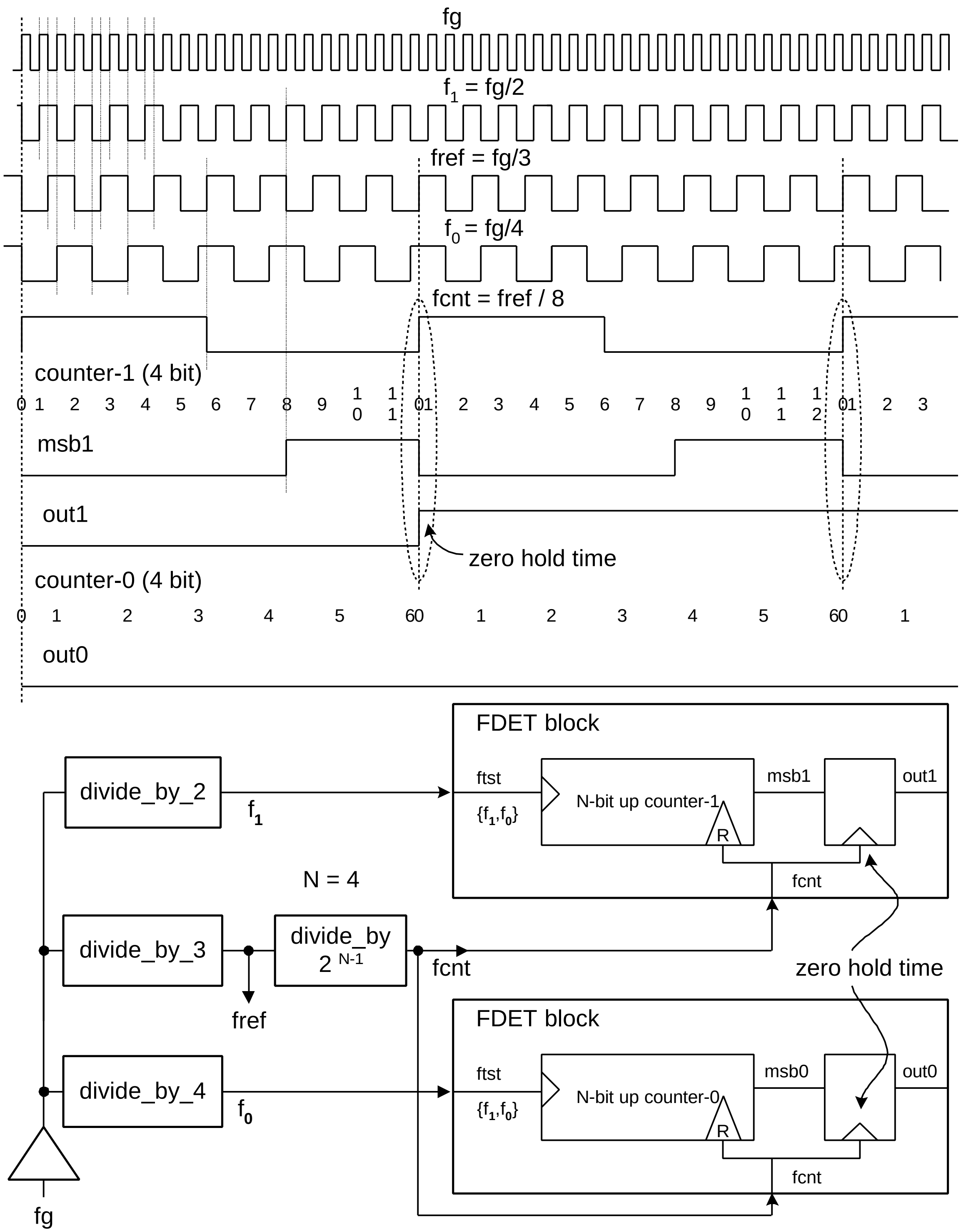

Figure 7: Frequency detectors receiving both f_1 and f_0. The states of the up-counters and signals are shown. The value of *out1* becomes valid after the first discrimination cycle completes. Note N = 4.

6 Test Application

An AC Boundary-Scan test consists of a sequence of "test frames" applied by serially shifting commands and data into the TAPs and Boundary-Scan registers of the chips on a board or system. A given test frame for AC EXTEST will look identical to one for (DC) EXTEST, with the only exception being that the AC EXTEST instruction is executed:

On the board being tested:

1. Start a free-running global frequency *fg* (which may be a sinusoidal oscillator). The value of *fg* is chosen to assure that for the smallest coupling capacitors, there will be adequate signal passage through the high-pass filter they create. This reference may be on on-board resource, or be provided by a test system. It only needs to run during the time that AC_EXTEST is in operation.

On all chips:

2. Load SAMPLE/PRELOAD into all chips that will participate in EXTEST (either AC or DC) testing.
3. UPDATE_IR: make SAMPLE/PRELOAD operational.
4. SHIFT_DR: load all Boundary Registers with driver data for first frame of testing (both AC and DC).
5. UPDATE_DR: transfer test data to update latches of Boundary Register.
6. SHIFT_IR: load AC EXTEST instruction into all chips that possess it, and ordinary (DC) EXTEST into all other chips participating in the test.
7. UPDATE_IR: make EXTEST instructions (AC or DC) operational: sets extest = 1 and ac_mode = 1 or 0 depending on if chip is loaded with AC or DC EXTEST. This applies the pattern set up by SAMPLE/PRELOAD. In chips performing EXTEST, the normal DC pattern is driven. In chips performing AC EXTEST, those pins with the AC capability begin transmitting (on both positive and negative terminal) the frequency selected by the data register bit, while those pins that do not have the AC capability will behave exactly as for normal (DC) EXTEST. (Note the AC frequency transmission begins immediately when the AC EXTEST instruction becomes effective.)
8. RUN-TEST/IDLE: Wait for a number of cycles of *fg* to allow the frequency discriminators to discern a frequency. Note that *fg* is independent of TCK and free-running, so the time it takes the *fg* cycles to pass may equal a small number of TCK cycles, perhaps even small enough to skip passage through RTI and proceed directly to the next step.
9. CAPTURE_DR: Those input pins that possess the AC capability have now discriminated the existence (or non-existence) of f_1 or f_0 frequencies. This bit is captured. (Note, if no frequency is present, then the discriminator will digitize a 0 since a frequency of zero Hertz is less than the *fref*.) Those pins that are performing normal (DC) EXTEST will capture the DC value on their inputs.
10. SHIFT_DR: scan new frame data into the Boundary Registers while scanning out the captured data from the current frame for analysis.

11. UPDATE_DR: for the new frame, drive *ftst* (either f_1 or f_0, depending on data in *U*}) from transmitter on the AC coupled drivers and normal (DC) EXTEST data on the rest of the drivers.
12. Until there is no more frame data, go to step 8.
13. If the last frame has been executed, the last capture step will capture its results that must be shifted out. The data shifted in is "don't care" data and can be chosen from the "safe" bits specified in BSDL.

When a defect causes no frequency to be detected (as is the case for many defects), the frequency detector circuit must output a deterministic and consistent value. Upon execution of multiple test frames expecting different values from the receiving Boundary-Scan registers, the defect will result in one or more of the frames failing in certain bit locations.

Once all the frame result data is collected, failure analysis using well-known Boundary-Scan diagnostic routines can be done. One item must be considered however. Since a single data cell is used to initiate a data transfer over a differential pair but two cells (that capture positive and inverted copies of this data) receive this data, diagnostic routines that are unfamiliar with differential signaling may need to be improved to recognize this.

7 Implementation Specifications

7.1 General Implementation

7.1.1 Rules

a) Components conforming with this standard shall first conform to the rules in IEEE Std 1149.1.

NOTE – In particular, all system pins shall implement (DC) EXTEST. Differential drivers will have a single Boundary Register data cell for drive data. A differential receiver will have a single Boundary Register cell that monitors the output of the differential receiver. This document shows how to add new resources for supporting AC testing.

b) All system pins of a component shall be categorized either as (normal) DC pins or as AC pins.

NOTE 1 – This categorization is performed by the component designer based on the expectation that such pins will use AC coupling in their intended application. If AC coupling is a board designer's choice at board design, then the component designer may wish to designate them as AC so the option will exist to use AC_EXTEST in those instances when AC coupling is used.

NOTE 2 – If pins that are designated as AC pins are actually wired together without AC coupling, it is assumed they are DC compatible and perform adequately. In this event, the AC_EXTEST instruction is also expected to perform correctly.

c) All AC pins shall be implemented with the provisions given herein for an AC_EXTEST instruction.

d) All DC pins shall perform (DC) EXTEST when the AC_EXTEST instruction is loaded in the TAP instruction register.

7.1.2 Description

[TBA]

7.2 Frequency References and Detectors

7.2.1 Rules

a) A global frequency reference *fg* shall be provided as an input pin for components that possess AC pins.

NOTE – This pin provides a frequency reference and as such is a clock pin by the definition of IEEE Std 1149.1. Permission 7.2.3(a) allows the reuse of a system clock pin for *fg*.

b) The *fg* input pin shall accept as valid input, a waveform with a frequency less than or equal to a stated maximum that may be a sinusoidal waveform, or a rectangular waveform with a duty cycle anywhere between 30% and 70%.

NOTE 1 – The range on duty cycle and the allowed input waveforms assures system designers that the specifications for the signal driving *fg* are not stringent and thus do not require sophisticated clock distribution techniques at the board or system levels.

NOTE 2 – Voltage levels for *fg* are not specified by this standard. It may happen that several copies of the signal driving *fg* are needed in a board or system where voltage levels for dissimilar components must be provided. Because this frequency-based scheme is insensitive to the phase of *fg* at any device, the buffering and distribution of *fg* has no strict requirements.

c) The *fg* input pin shall be provided with an *Observe_Only* input cell in the Boundary Register.

NOTE – An Observe_Only cell cannot block the *fg* pin from the internal on-chip circuitry while either EXTEST or AC_EXTEXT is the effective instruction. This allows the *fg* pin to be tested using Boundary-Scan test techniques using ordinary (DC) EXTEST algorithms.

d) Internal to any component that has AC output pins, two frequency signals $f_1 = fg / 2$ and $f_0 = fg / 4$ shall be generated, with a duty cycle of 50% plus or minus 10%.

NOTE – It is expected that the f_0 and f_1 waveforms emanating from drivers will be have no appreciable DC offset seen after passing through AC coupling capacitors. A waveform with 50% duty cycle assures this.

e) Internal to any component that has AC input pins, a frequency reference signal $fref = fg / 3$ shall be used for frequency discrimination.

NOTE – Thus, frequencies $f_1 > fref > f_0$.

f) When no frequency reference is applied from an external source to *fg*, signals *fref,* f_1 and f_0 shall be quiescent, in no way interfering with component operation.

NOTE – Power consumption by these circuits can be zero when *fg* is removed. When the AC_EXTEST instruction is not loaded, these circuits can also be turned off to eliminate power consumption even when a signal on the *fg* pin is present.

g) When permission 7.2.3(a) is implemented, system signals applied to this frequency reference pin while the AC_EXTEST instruction is not in effect shall be ignored.

h) The frequency detector circuit shall respond to "no-frequency-detected" conditions by producing a deterministic output.

i) The frequency detector circuit shall respond to an insufficient discrimination interval condition (e.g. less than 2^{N-1} cycles of *fref*) by producing a deterministic output.

NOTE – This can be done by clearing the value in the output flip-flop of the frequency detector (see Figure 5) upon initiation of AC_EXTEST.

j) The input amplifier used to drive the frequency detector shall have a threshold between 1/8 and ½ of the input voltage swing measured from the termination voltage.

NOTE – This rule would apply to simple buffer amplifiers that do not incorporate hysteresis in the design.

k) If the input amplifier used to drive the frequency detector is implemented with hysteresis, then the rising edge threshold shall be between 1/8 and ½ of the input voltage swing measured from the termination voltage, and the falling edge threshold shall be between -1/8 and -½ of the input voltage swing measured from the termination voltage.

NOTE – A buffer amplifier with hysteresis will produce output waveforms with a squareness approaching that of the waveform originated at the driver. See the description in section 7.2.4.

l) The frequency detector circuit shall have a maximum input frequency documented by the designer.

NOTE – This frequency will imply an upper limit on *fg* since the higher test frequency (f_1) is ½ of *fg*.

7.2.2 Recommendations

a) The upper limit for frequency of *fg* should be sufficiently high to allow $f_0 = fg / 4$ to pass through the expected coupling capacitors with acceptable attenuation.

NOTE – It is recommended that an upper limit be selected that anticipates reduced capacitor values that may be observed during the lifetime of the component.

b) The length of the discrimination cycle time should be chosen to enhance immunity to spurious noise that could be anticipated in a system.

NOTE – Longer discrimination times will allow the rejection of more noise impulses.

7.2.3 Permissions

a) The *fg* pin may be used for system functions.

NOTE – However, when AC_EXTEST is in operation, this pin shall transmit the reference frequency to the AC_EXTEST function. When the device is in system operation, this pin may be used for other clocking purposes.

b) A single clock generation circuit can be used to generate f_1, *fref*, and f_0; these signals can be distributed as needed to all pad circuits requiring AC_EXTEST.

NOTE – The distribution of these signals is not constrained by any timing or phase relationships.

7.2.4 Description

The global frequency source pin transmits *fg* into a device and this is used to generate all the needed test frequencies. While the *fg* signal itself may be sinusoidal or a non-square rectangular waveform, the derived frequencies f_1, f_0 and *fcnt* are all square waves. However after either f_1, or f_0 have been transmitted from a driver and passed through a coupling capacitor, the signal seen by an AC input pin may be decidedly non-square.

Consider a pathway for a driven signal X, through a coupling capacitor (signal Y) and then through a simple input receiver producing signal Z that serves as input to a frequency detector, as shown in Figure 8. The selection of thresholds follows rule 7.2.1(j).

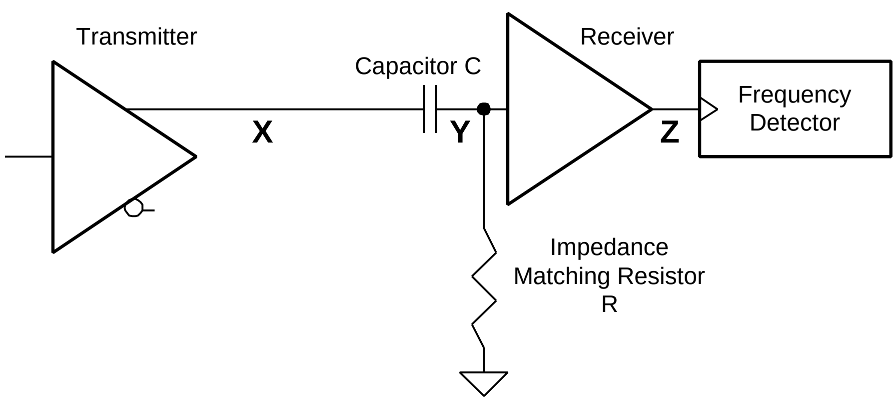

Figure 8: A pathway for a test frequency where the receiver is a simple amplifier with one threshold.

Figure 9 shows a set of waveforms possibly seen as a function of the test frequency and the RC time constant of the coupling capacitor and the termination resistor. The simple input buffer will have a threshold somewhere between an eighth and a half of the positive input waveform[5] which

[5] Note that the voltage swing at point Y is +/- 2 volts, but this is because the signal period is much larger than the time constant of the coupling network. If the time constant were much larger, then the swing would approach +/- 1 volt. The threshold range (between 1/8 and ½ the swing) is related to the latter case.

restores the rectangular look of the waveform. Depending on the actual threshold, the restored waveform will have some amount of time where it is a one. The best and worst cases are shown. The design of the input buffer should pass legitimate test pulses, but reject small noise (runt) pulses. This presents a design tradeoff, since having a shorter time constant will allow the use of smaller coupling capacitors, but makes noise rejection more difficult.

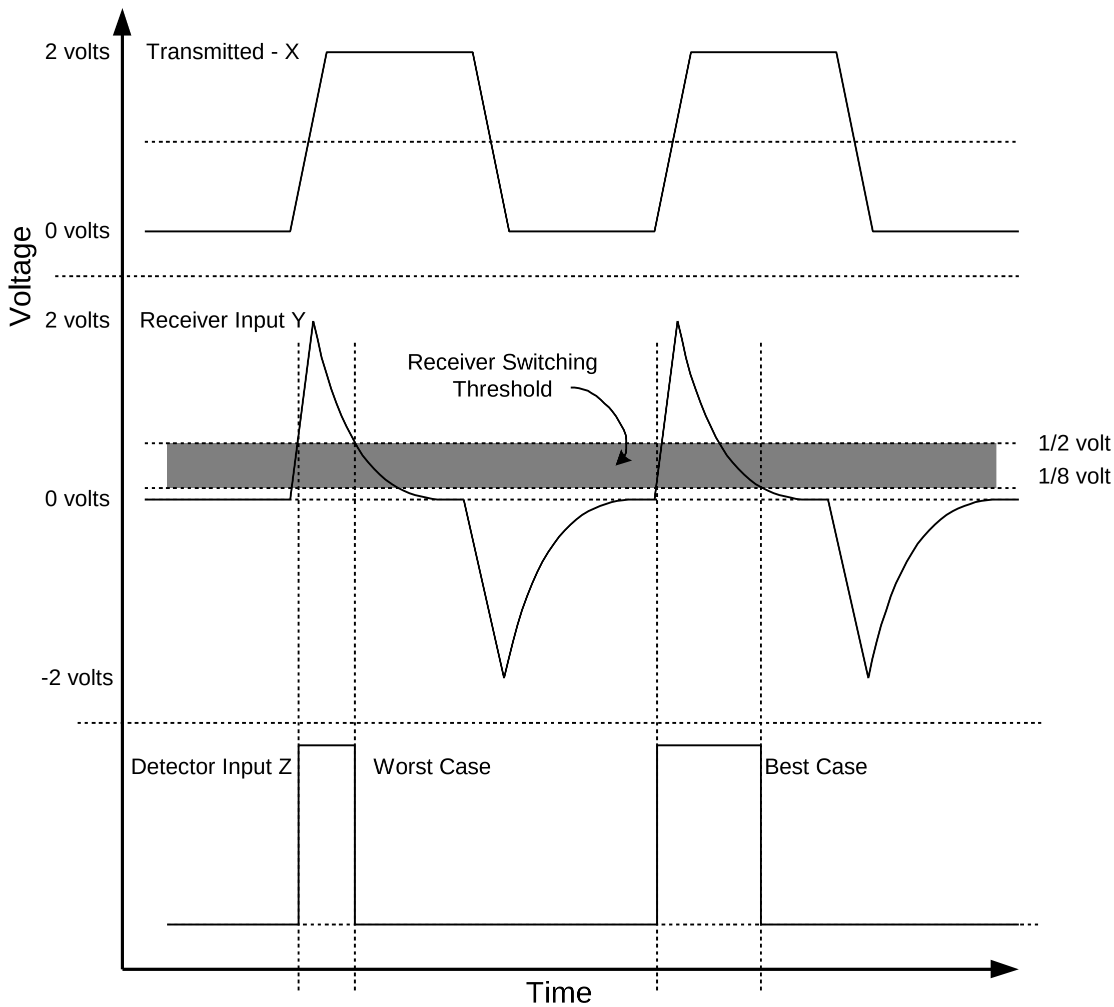

Figure 9: Waveforms seen at positions X, Y and Z in Figure 8.

The circuit in Figure 10 uses a more sophisticated input buffer amplifier which uses hysteresis and both the positive and negative halves of the waveform seen at point Y. The output of the receiver is seen in Figure 11. The selection of thresholds follows rule 7.2.1(k).

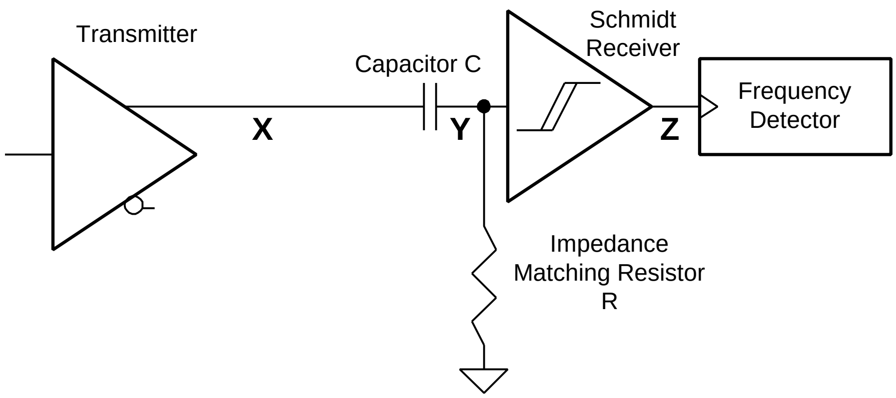

Figure 10: Input receiver with hysteresis, utilizing the negative-going signal information as well.

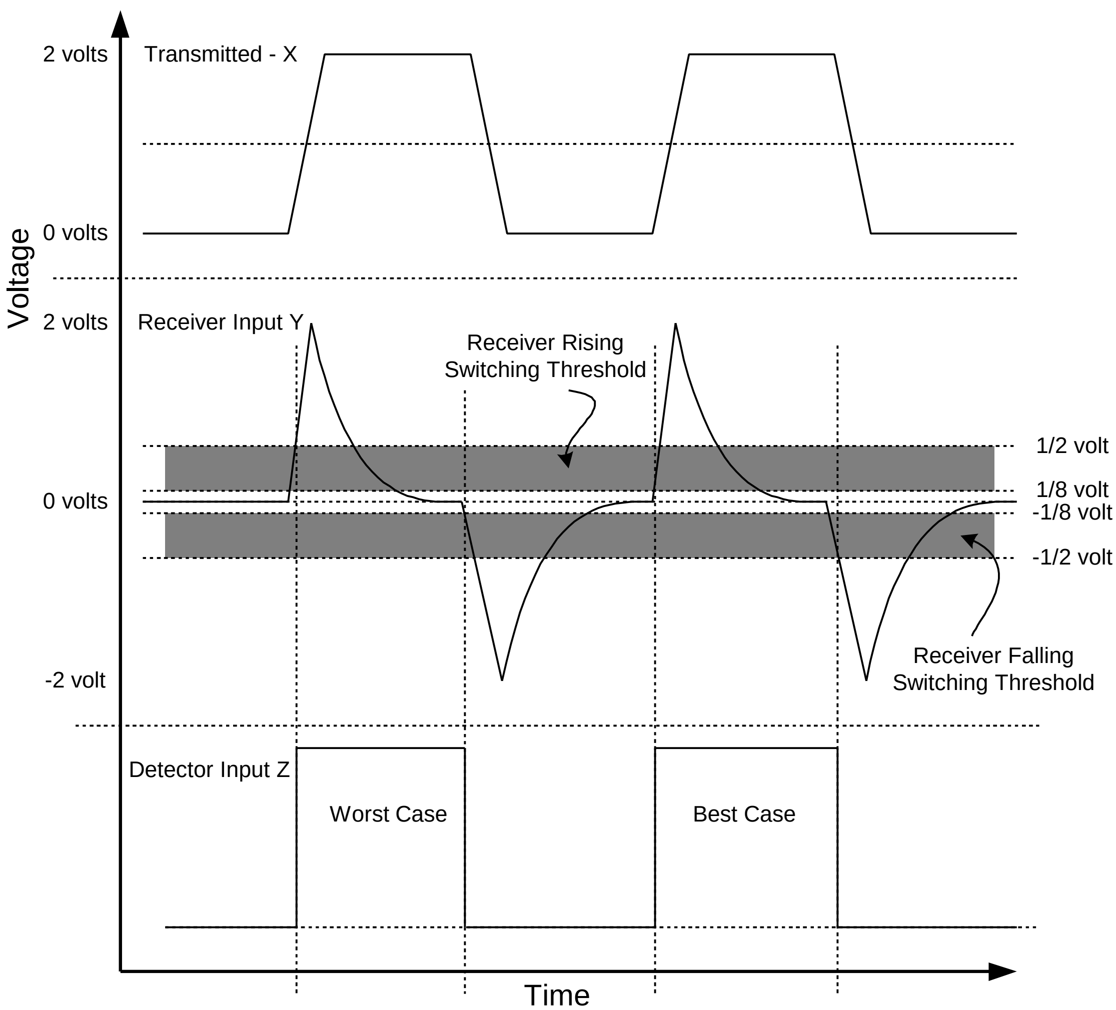

Figure 11: Waveforms seen at X, Y and Z in Figure 10.

In Figure 11 the waveforms produced by the input buffer amplifier are much nearer to being square and there is much less range between best and worst cases. This will allow more triggering energy to be supplied to the frequency detector input, which should allow the use of smaller coupling capacitors.

Editor's note: An example should be supplied here about how the time constant of the coupling network will determine the test frequency which then determines *fg*.

7.3 TAP Instruction AC_EXTEST

7.3.1 Rules

a) An AC_EXTEST instruction shall be provided for components that possess AC pins and the opcode(s) for this instruction shall be selected by the component designer.

b) The AC_EXTEST instruction shall become effective at the falling edge of TCK in the *Update-IR* TAP Controller state.

NOTE – By "effective" it is meant that (enabled) AC drive pins shall begin to produce either f_1 or f_0 frequencies at this time. AC receive pins shall begin to discriminate these signals at this time.

c) The AC_EXTEST instruction shall remain effective in the *Run-Test/Idle* TAP Controller state.

NOTE – This rule assures that if extra discrimination time is required by AC receiving pins, this time can be provided by cycling in the *Run-Test/Idle* TAP controller state for as many TCK cycles as it takes to satisfy the discrimination requirements.

d) The time required for discrimination shall be specified in terms of the number of cycles *fg* required, or as an absolute time in seconds.

NOTE 1 – In a digital implementation such as discussed in this document, the discrimination time will be a cycle count of *fg*. If an analog implementation is used, it is possible that absolute time is required. However this circuitry must use *fref* as the discrimination frequency reference per rule 7.2.1e) above.

NOTE 2 – It is possible that this discrimination time is very short compared to the 2.5 cycles of TCK it takes to travel from *Update-DR* (or *Update-IR*) to *Capture-DR* thus eliminating the need to spend any time cycling in the *Run-Test/Idle* TAP Controller state.

NOTE 3 – Tools that develop tests using AC_EXTEST may deliberately lengthen the discrimination time to include extra time for reducing DC offsets on coupling capacitors that may have built up before testing began.

e) There shall be no other restrictions imposed on discrimination time.

NOTE – For example, the specification for the amount of time spent waiting for discrimination shall not contain maximums (e.g, not to exceed) or be specified as exact multiples of the discrimination interval.

7.3.2 Description

[TBA]

7.4 Boundary Cell Provisions for Single-Ended AC Pins

7.4.1 Rules

a) When the AC_EXTEST instruction is selected and a single-ended AC driver is enabled for driving, either f_1 or f_0 shall be driven from the output pin determined by the state of the Boundary Register data cell with f_1 = 1 and f_0 = 0, for each pin, beginning at the falling edge of TCK in the *Update_DR* or *Update-IR* TAP controller state.

NOTE – A single-ended 2-state driver is always enabled for driving. This rule also applies to drivers that have output enables controlled by Boundary Register control cells.

b) When the AC_EXTEST instruction is selected and a single-ended AC driver is *not* enabled for driving, the pin shall not be driven.

c) When the AC_EXTEST instruction is selected, the frequency value (f_1 = 1 and f_0 = 0) on a single-ended AC system input pin, as determined by a frequency detector using *fref* (= *fg / 3*) as the basis for discrimination, shall be loaded into the Boundary Register data cell for that pin on the rising edge of TCK in the *Capture_DR* state.

d) When the AC_EXTEST instruction is selected, if no frequency is present on a single-ended AC system input pin, a deterministic value shall be loaded into the Boundary Register data cell for that pin on the rising edge of TCK in the *Capture_DR* state.

7.4.2 Description

[TBA]

7.5 Boundary Cell Provisions for Differential AC Pins

7.5.1 Rules

a) A differential AC output pin pair shall have a single Boundary Register data cell used to provide test data for that pin pair.

NOTE – This means the insertion point for Boundary-Scan data for AC_EXTEST will likely be accomplished ahead of the output driver, which is desirable in many differential implementations.

b) When the AC_EXTEST instruction is selected and a differential AC driver is enabled for driving, either f_1 or f_0 shall be driven from the AC system differential output pins determined by the state of the Boundary-Scan data cell with f_1 = 1 and f_0 = 0, for each pin, beginning at the falling edge of TCK in the *Update_DR* or *Update-IR* TAP controller state.

NOTE 1 – A differential 2-state driver is always enabled for driving. This rule also applies to drivers that have output enables controlled by Boundary Register control cells.

NOTE 2 – Both legs of an enabled differential pair shall thus each be transmitting frequency f_1 or f_0. While logically the same value (1 or 0), these signals are 180 degrees out of phase with each other and as such will interfere with each other if there is a short between them.

e) When the AC_EXTEST instruction is selected and a differential AC driver is *not* enabled for driving, the pin shall not be driven.

f) When the (DC) EXTEST instruction is selected, either a static 1 or a static 0 shall be driven from the AC system positive output pins determined by the state (1 or 0) of the Boundary-Scan data cell, for each pin, beginning at the falling edge of TCK in the *Update_DR* TAP controller state.

NOTE 1 – This rule states that the (DC) EXTEST instruction from IEEE Std 1149.1 defines the DC behavior of an AC positive output pin.

NOTE 2 – This rule also ensures that there is matching parity between the Boundary Register cell and the positive leg of the signal pair, and a parity inversion on the negative leg.

g) A differential AC input pair shall have a single boundary register data cell *per pin* to receive data from that pair.

NOTE – This rule assures that while there is a single Boundary Register data cell for a differential driver pair per rule a) above, there must be two data cells (one for each leg) of a differential receiver pair.

h) When the AC_EXTEST instruction is selected, the frequency value (f_1 = 1 and f_0 = 0) on the *positive* leg of a differential AC system input pin pair, as determined by a frequency detector using *fref* (= *fg / 3*) as the basis for discrimination, shall be loaded into the Boundary Register data cell for that pin on the rising edge of TCK in the *Capture_DR* TAP Controller state.

i) When the AC_EXTEST instruction is selected, the frequency value (f_1 = 1 and f_0 = 0) on the *negative* leg of a differential AC system input pin pair, as determined by a frequency detector using *fref* (= *fg / 3*) as the basis for discrimination, shall be *inverted* and then loaded into the Boundary Register data cell for that pin on the rising edge of TCK in the *Capture_DR* TAP Controller state.

NOTE – This rule assures that the negative leg of a differential receiver pair will receive the complement of the data that was transmitted by the driver. This occurs even though the *same frequency* was present (though phase shifted by 180 degrees) on both legs. This means the behavior of differential pins for both AC and DC EXTEST is logically identical.

j) When the AC_EXTEST instruction is selected, if no frequency is present on a differential AC system input pin, a deterministic value shall be loaded into the Boundary Register data cell for that pin on the rising edge of TCK in the *Capture_DR* TAP Controller state.

NOTE 1 – The component designer is free to choose the value that is loaded, but required to assure it is deterministic.

NOTE 2 – There is no requirement that this default value be the same or different on the negative leg versus the positive leg.

k) When the (DC) EXTEST instruction is selected, the static value (1 or 0) on a *positive or negative leg* of a differential AC system input pin pair shall be loaded into the Boundary Register data cell for that pin on the rising edge of TCK in the *Capture_DR* TAP Controller state.

NOTE 1 – This rule states that the (DC) EXTEST instruction from IEEE Std 1149.1 defines the DC behavior of an AC positive input pin. Further there is no parity inversion between either pin of a pin pair and the associated Boundary Register data cell.

NOTE 2 – Rule i) above that states there must be a parity inversion on the negative leg input pin for AC_EXTEST. However, this matches the *lack* of parity inversion that the transmitted frequency has on the negative leg of a driver during AC_EXTEST. Thus, the logical result of AC and DC EXTEST are identical on differential pairs.

l) When a differential pin pair serves as both a driver and receiver (that is, it supports bidirectional data flow), then the rules for differential drivers *and* receivers shall be followed for this pin pair.

NOTE 1 – This means that there would be a single control-and-observe data cell for the driver, and two frequency detectors and data cells for receiving signals on the two legs.

NOTE 2 – This structure allows on-chip monitoring of a pin driver's ability to drive a channel. If for example there is a short in the channel ahead of the capacitors, the driver is unlikely to perform correctly. However if the short occurs *after* the capacitors, then the driver may appear to be unaffected, but the downstream receiver will perceive a failure.

7.5.2 Description

[TBA]

8 Conformance and Documentation Requirements

8.1 Conformance

8.1.1 Specification

Rules

a) A component conforming to this standard shall comply with all rules set out herein.

NOTE – Due to rule 7.1.1(a), this also implies conformance with the rules set out in IEEE Std 1149.1.

8.1.2 Description

Conformance to the rules set out herein and in IEEE Std 1149.1 are essential for testing boards and other assemblies containing both DC and AC coupled interconnections, allowing manufacturing defects such as shorted or open solder joints to be found and repaired before shipment. Conformance allows:

- IC vendors to provide testability features in a standardized way, so that each new IC design does not need new engineering investment to provide testability.
- makers of Automatic Test Equipment to develop and continually refine standardized tools for the automation of test development, test execution and diagnosis of failures.
- end-users to strategize and develop test methodologies in a standardized way, making full use of the automation provided by tools, to allow them to produce very large and complex boards and system more rapidly and efficiently.

8.2 Documentation

Because adherence to this standard implies adherence to IEEE Std 1149.1, all ISC devices shall have a description supplied in Boundary-Scan Description Language (BSDL) provided with IEEE Std 1149.1b-1994.

8.2.1 Specification

Rules

a) A component conforming to this standard shall be documented with a Boundary-Scan Description Language (BSDL) description.

NOTE – See IEEE Std 1149.1b-1994 for a description of IEEE BSDL. The precursor to IEEE BSDL developed in 1990 cannot be used for documentation because it does not support the concept of “BSDL Extensions”.

b) A component shall have the AC_EXTEST instruction and register relationship and optional provisions documented via the syntax provided by BSDL and the BSDL extension provided by this standard.

NOTE – New BSDL syntax (contained within a “BSDL extension”) for describing concepts and structures introduced by this standard are given in 8.4. A BSDL extension is a mechanism provided by IEEE Std

1149.1b-1994 (see "User extensions to BSDL") which allows proprietary syntax to be provided that will allow tools to work that are unaware of this syntax.

c) Other properties of a device not described in BSDL shall be documented by the manufacturer.

NOTE – For example, this includes voltage levels for frequency reference *fg*, TAP signal voltage level requirements, and any additional power requirements of the device.

8.2.2 Description

Consider a device with a full set of 1149.1 instructions and registers as well as the new AC_EXTEST instruction. Here are fragments of BSDL needed to document this device.

First, for devices that have AC differential outputs, a "Port_Grouping" attribute must be given that identifies the positive and negative legs. Notice that because the implementation of AC_EXTEST requires single-ended receivers for differential input pin pairs, the test behavior of these pins is single-ended and thus should not be included in a port grouping description. (An error would result during compilation.) For example, for a device with four pairs of differential data drivers, the Port_Grouping could be:

```
Attribute PORT_GROUPING of ACDEV:entity is
  "Differential_Voltage ( "
        "  (D(1), Dbar(1)), "
        "  (D(2), Dbar(2)), "
        "  (D(3), Dbar(3)), "
        "  (D(4), Dbar(4)) )";
```

NOTE 1 – Refer to IEEE Std 1149.1b-1994, section B.8.8, titled "Grouped Port Identification", for precise information regarding the Port_Grouping attribute.

NOTE 2 – The entity name in these examples is shown as "ACDEV". This would be replaced with a name unique to the device.

Since AC coupled circuits cannot transmit current, the "Differential_Voltage" BSDL keyword is used rather than "Differential_Current".

At a board or system level, differential pairs are interconnected with pair-wise wiring or AC blocking capacitors. With the information from the port grouping attribute, software can trace the pathways of the positive and negative legs of each pair starting at the drivers. Thus, though the receivers are not identified as differential, their differential relationship can be discovered via this tracing exercise.

Next, an "Instruction_Opcode" attribute used to define instruction names and binary code assignments should be given:

```
Attribute INSTRUCTION_OPCODE of ACDEV:entity is
  "BYPASS              (111111), " &
  "EXTEST              (000000), " &
  "SAMPLE              (000001), " &
  "IDCODE              (010001), " &
  "USERCODE            (010010), " &
  "AC_EXTEST           (000010)  ";
```

NOTE – Refer to IEEE Std 1149.1b-1994, section B.8.11, titled "Instruction Register Description", for precise information regarding the Instruction_Opcode attribute.

Later, a Register_Access attribute is given:

```
Attribute REGISTER_ACCESS of ACDEV:entity is
  "Boundary      (AC_EXTEST)";
```

The Register_Access attribute defines the existence of optional registers, their length, the instruction(s) that target them, and any consistent capture data they may be loaded with in the *Capture-DR* TAP Controller state. The above attribute documents the fact that AC_EXTEST targets the Boundary Register.

NOTE 1 – Refer to IEEE Std 1149.1b-1994, section B.8.13, titled "Register Access Description", for precise information regarding the Register_Access attribute.

NOTE 2 – This attribute may also (though redundantly) document the associations of EXTEST, SAMPLE, BYPASS, USERCODE and IDCODE as well. This was omitted here.

8.3 BSDL Extention for AC_EXTEST Description (STD_xxxx_yyyy)

The information provided above in 8.2 showed how to document features of a device implementing AC_EXTEST using the existing syntax of BSDL. This information is incomplete. BSDL has been defined (see IEEE Std 1149.1b-1994) to be extensible using a mechanism called a "BSDL Extension". A BSDL Extension for describing additional features of AC_EXTEST devices is given here.

NOTE – Refer to IEEE Std 1149.1b-1994, section B.8.17, titled "User extensions to BSDL", for precise information regarding BSDL extensions.

The extension mechanism chosen for describing devices is based on the definition of a VHDL package with the name "STD_xxxx_yyyy" which contains the definitions of attributes that will be used to supply relevant data. Therefore, an compliant device BSDL will contain an additional "use" statement appearing just after the "standard use statement" as in this excerpt of a BSDL file:

```
  …
use STD_1149_1_1994.all;     -- Standard 'use' statement
use STD_xxxx_yyyy.all;       -- BSDL Extension for AC_EXTEST
  …
```

NOTE – Refer to IEEE Std 1149.1b-1994, section B.8.5, titled "Use Statement", for precise information regarding references to additional packages.

Utilization of the extension mechanism of BSDL guarantees that AC_EXTEST information can be supplied to applications that are cognizant of this functionality without hindering other applications that may not be aware of this functionality. Non-cognizant applications will simply ignore the extension.

The VHDL package STD_xxxx_yyyy contains the definition of additional attributes used to complete the description of the ISC features of a device. The content of this VHDL package is given here.

Syntax

```
Package STD_xxxx_yyyy is   -- Attribute definitions for AC_EXTEST
description
  use STD_1149_1_1994.all;  -- Refer to BSDL definitions

  attribute AC_EXTEST_Pin_Behavior:   BSDL_Extension;
  attribute AC_EXTEST_Frequency_Ref:  BSDL_Extension;
end STD_xxxx_yyyy;
```

```
Package Body STD_xxxx_yyyy is
  -- No content, this package body is required by BSDL syntax
end STD_xxxx_yyyy;
```

This VHDL package is “read-only” and may be maintained within a given system in the same location as the standard package STD_1149_1_1994.

8.4 BSDL Extension Structure

The following sections define the syntax and applicable semantics for the BSDL attributes defined by the AC_EXTEST extension. The form of the syntax used is defined in IEEE Std 1149.1b-1994.

NOTE 1 – See IEEE Std 1149.1b-1994, “Lexical elements of BSDL” and “Notes on syntax definitions” for the conventions used herein to describe the parsing requirements for BSDL.

NOTE 2 – Syntactic items are shown surrounded in “< >” brackets when referenced in this text. Many of these items will be defined here, but some will be adopted from IEEE Std 1149.1b-1994. Those that are adopted will be <u>underlined</u> to indicate their source.

When an AC_EXTEST extension exists in the extension area of a BSDL description, it must have the structure shown here. The various attributes, both mandatory and optional must appear in a prescribed order and not be intermixed with other statements.

NOTE – These other statements include BSDL attributes, attributes for other BSDL extensions and general VHDL constructs.

8.4.1 Specification

Syntax

```
<AC_EXTEST Extension> ::=
  <AC_EXTEST pin behavior description>                    (see 8.5.1)
  <AC_EXTEST Frequency Ref description>                   (see 8.5.2)
```

Rules

a) The syntax for an <AC_Extest Extension> as it appears in the extension area of a BSDL description shall be that shown above.

NOTE – The attributes must appear in the order shown above.

Permissions

b) An <AC EXTEST Extension> may appear anywhere in the extension area of a BSDL description, subject to any rules defined by other extensions as to ordering.

NOTE – If more than one extension exists, an <AC EXTEST Extension> may appear before or after any of them within the limits of their definitions.

8.4.2 Description

The two mandatory attributes of an <AC_EXTEST Extension> must appear in the order shown. No other statements may be intermixed within the extension, as the syntax above specifies. This extension may coexist with a number of other extensions.

8.4.3 Keywords for AC_EXTEST BSDL

This subsection lists the keywords of AC_EXTEST BSDL. These keywords are in addition to the reserved words of BSDL and VHDL. (See "BSDL reserved words" and "VHDL reserved and predefined words" in IEEE Std 1149.1b-1994.)

```
AC_EXTEST_Pin_Behavior
AC_EXTEST_Frequency_Ref
cycles
seconds
```

8.5 BSDL Attribute Definitions

The mandatory and optional attributes needed to describe the ISC properties of a device are given in the remainder of this section. They must appear in the order given in 8.4.1.

8.5.1 Attribute AC_EXTEST_Pin_Behavior

The mandatory AC_EXTEST_Pin_Behavior attribute is used to enumerate those system pins of a component that are "AC" pins, per rule 7.1.1(b), and have been provided with AC_EXTEST capability.

8.5.1.1 Specification

Syntax

<AC EXTEST pin behavior description> :: = **attribute AC_EXTEST_Pin_Behavior**
of <component name> **: entity is** <AC pin string> **;**
<AC pin string> ::= **"** <AC pin list> **"**
<AC pin list> ::= <AC pin> { **,** <AC pin> }
<AC pin> ::= <port ID>

Rules

a) The syntax for the mandatory AC_EXTEST_Pin_Behavior attribute shall be that shown above.
b) The value of <port ID> shall contain a <port name> declared in the logical port description for the device.

NOTE – A <port ID> may consist of a <port name> or a <subscripted port name> as described in "Commonly used syntactic elements" in IEEE Std 1149.1b-1994. See also "Logical port description statement" in that standard.

c) The value of <port ID> shall refer to a system pin only and not to any pin that is a scan port signal or a compliance port pin; further, the <pin type> of the <port name> shall not be "linkage".

8.5.1.2 Description

The AC_EXTEST_Pin_Behavior attribute can refer to individual system pins as <subscripted port name> elements or as <port name> elements that refer to individual system pins or collections of system pins.

Example

```
attribute AC_EXTEST_Pin_Behavior of ACDEV : entity is
        "Enable, Data_bus, Cntl(2), Cntl(1), Addr_bus" ;
```

8.5.2 Attribute AC_EXTEST_Frequency_Ref

The mandatory AC_EXTEST_Frequency_Ref attribute is used to identify the *fg* system input pin of a component. When the frequency discrimination time for the device (that has AC input pins) is measured in cycles of *fg* or an absolute time, this attribute also documents this requirement. If the device has no AC input pins, this attribute need not document a discrimination time.

8.5.2.1 Specification

Syntax

< AC_EXTEST Frequency Ref description> :: = **attribute AC_EXTEST_Frequency_Ref**
of <component name> **: entity is** <ref pin string> **;**
<ref pin string> ::= **"** <ref pin spec> **"**
<ref pin spec> ::= <port ID> [<discrimination time>]
<discrimination time> ::= <cycle count> | <time>
<cycle count> :: = <integer> **cycles**
<time> ::= <real number> **seconds**

Rules

a) The syntax for the mandatory AC_EXTEST_Frequency_Ref attribute shall be that shown above.

b) The value of <port ID> shall contain a <port name> declared in the logical port description for the device.

NOTE – A <port ID> may consist of a <port name> or a <subscripted port name> as described in "Commonly used syntactic elements" in IEEE Std 1149.1b-1994. See also "Logical port description statement" in that standard.

c) The value of <port ID> shall refer to a system pin only and not to any pin that is a scan port signal or a compliance port pin; further, the <pin type> of the <port name> shall not be "linkage".

d) When specified, the value of <cycle count> shall be greater than 0 cycles.

e) When specified, the value of <time> shall be greater than 0.0 seconds.

8.5.2.2 Description

The AC_EXTEST_Frequency_Ref attribute identifies an individual system pin as a <subscripted port name> element or as a <port name> element that refers to an individual system pin.

Examples

```
attribute AC_EXTEST_Frequency_Ref of ACDEV : entity is
        "Sys_Clk  16 cycles" ;

attribute AC_EXTEST_Frequency_Ref of ACDEV : entity is
        "M_Clk  12.0e-6 seconds" ;
```

9 Example

Figure 12 shows an example of a sub-section of the PC board from Figure 1, with the relevant pieces of Figure 4 and Figure 6 connected as intended. As an illustration, these frequencies could be employed:

fg = 300 MHz
f_1 = 150 MHz
$fref$ = 100 MHz
f_0 = 75 MHz (this determines the lower limit on values for AC coupling capacitors)
$fcnt$ (for N = 4) is $fref / 2^{N-1}$ = 12.5 MHz

The discrimination time is $3 * 2^{N-1}$ = 24 cycles of fg, or in this example, 80 nanoseconds.

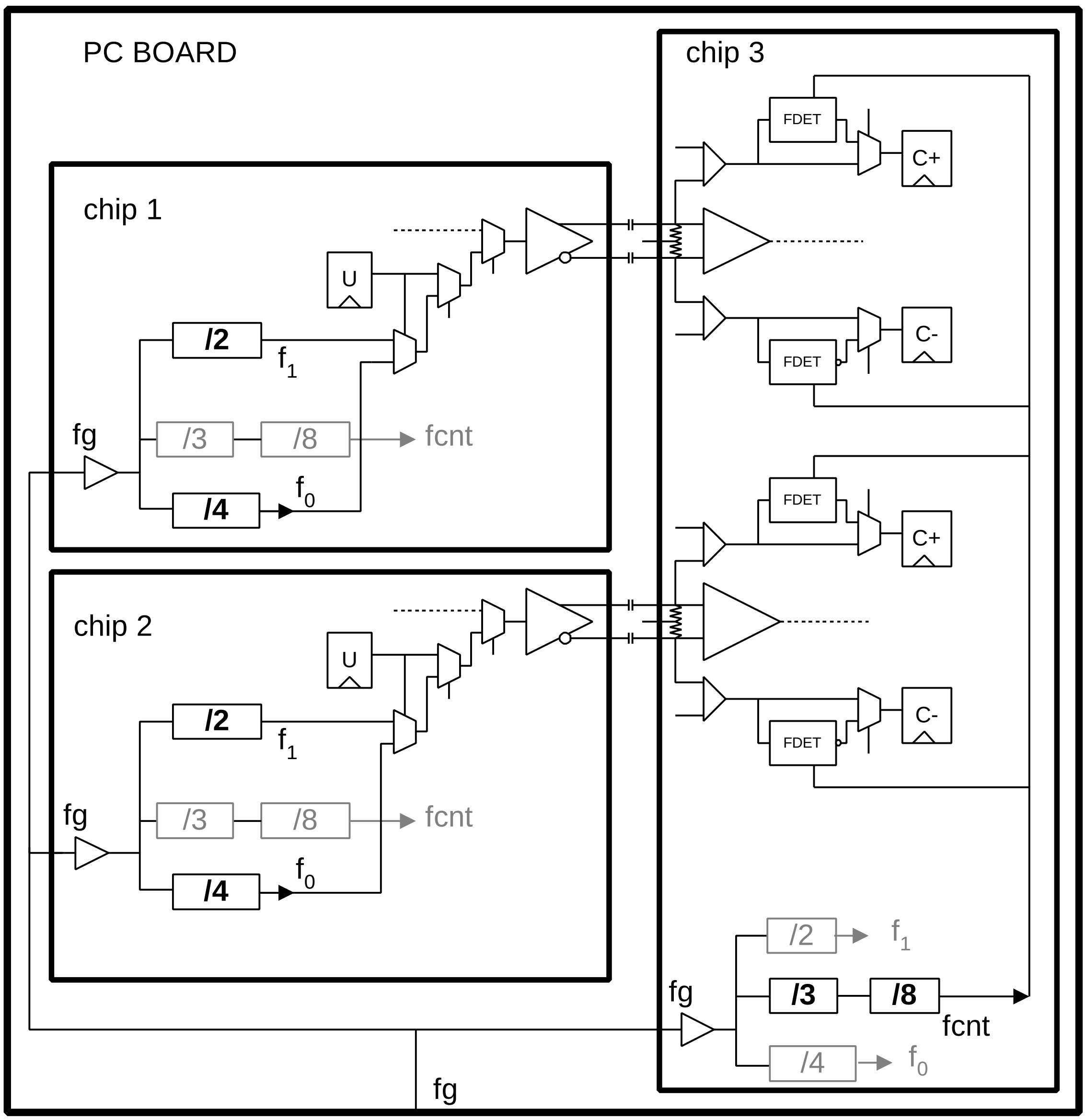

Figure 12: AC EXTEST resources among interconnected ICs.