An introduction to the Development, Architecture and Capability of Reflex components.

***Abstract* – This note introduces 'Reflex' - a new, flexible, relay test architecture which builds from simple low-cost relay test card modules upwards into a range of flexible parametric or life-test systems. The architecture is designed to accommodate a wide variety of end-user requirements from simple production test of a single parameter through to multi-contact life testing at very high speeds and to be quickly configured both at manufacture and by the end-user. This note describes the design evolution of the Reflex system and how some key engineering features bring superb timing accuracy and flexibility to the design.**

I. UNIFIED REFLEX ARCHITECTURE

A. Introduction.

For some years, Applied Relay Testing Ltd has made available a range of relay test products that offer extensive test capability and which are designed as integrated test instruments for the rigors of both production and laboratory use. Our original RT90 parametric test system and its successor the RT290 have enabled relay manufacturers and users to perform high quality tests at high speed coupled with traceability and extensive investigative capability [2]. To complement these dedicated test systems with their integrated 'high-end' features we are now introducing a range of lower cost relay test systems and components that combine flexibility with economy. Using these modular items it is possible to address the more cost-sensitive applications such as multi-station manual production test and simple automatic functional testing and sorting as well as more complex and variegated systems such as those for life-testing. These "Flexible Relay" or "Reflex" components and systems are the result of combining and packaging key test modules to suit the application as exactly as possible and to therefore obtain the best price to performance ratio, as well as to react to the expected increase in system configuration possibilities as the newer micro-geometry devices make multi-device, multi-contact parts possible [3][5].

B. Basic architecture.

The concept behind the Reflex architecture is to provide modular relay test systems that are built upwards from cost-effective relay-specific test functions and founded on the well-established PC plug-in card platform. Due to its aged design the PC can seem to have limitations as a hardware platform, but in recent years there have been many improvements in its application to rugged production situations with the emergence of true 'industrial' rack mountable PC chassis components complete with air filtering, redundant power supplies, stable and proven operating systems (e.g. Windows NT) all at increasingly cost-effective prices. These components are hard to ignore if one is striving for the lowest cost implementation of a test system and we decided that instead of creating a number of customised individual relay test systems that would each approximate (hopefully!) to common end-user requirements, we would create some generic low-end modular components that could be configured quickly into complete test systems such to provide quite exact end-user solutions at a cost-effective price. The basic building blocks could therefore be simple PC plug-in cards designed in such a way that any electrical and mechanical limitations of the PC platform did not intrude on relay test performance. The technical challenge would be to create not only a good mechanical and electrical test environment, but control, test and reporting software that would be able to accommodate all possible hardware flexibility, ideally configuring itself based on the hardware available. Such software should hide much of the hardware from the end-user and present itself in a simple and clear manner. Figure 1, showing a production graphical overview screen created using the Reflex software, illustrates that we have achieved this goal.

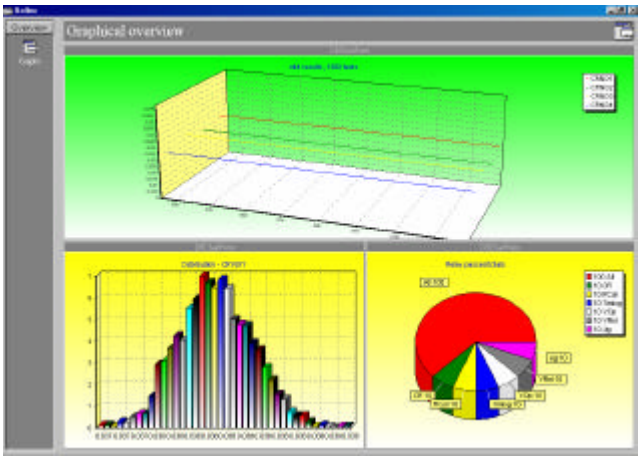


Figure 1 An example of relay batch graphics, created with the Reflex test software.

Within this chosen PC card format, we decided to partition test system resources into three basic categories:

- Contact resources (e.g. CR measurement, contact timing and loads)
- Device resources (e.g. monostable / bistable coil drive and measurement)
- System resources (e.g. test start, busy, handler control signal and global system synchronisation and timing).

The physical representation of this layout is as shown in Figure 2.

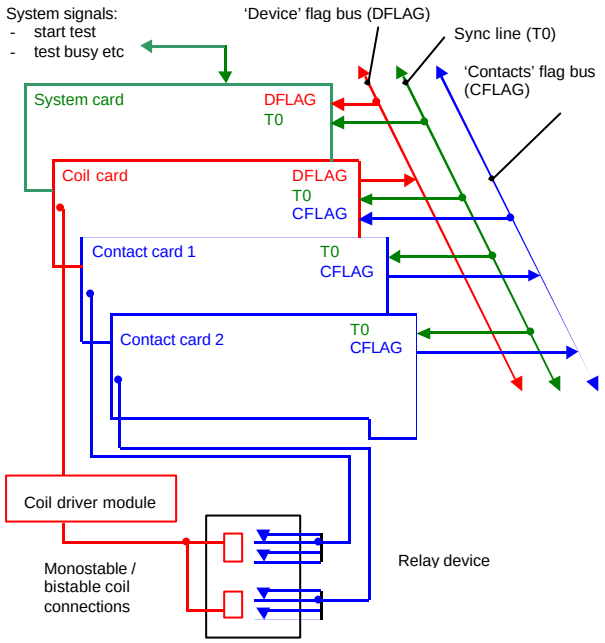


Figure 2 - Unified REFLEX architecture

Since resources for measuring the basic loaded relay contact and for driving the device coil are always required whether for production parametric tests or for contact life-testing, it turns out that by partitioning these functions to imitate the separate functionality within a relay device, systems can be built from Reflex cards that directly correspond to the device (or devices) that will be tested – almost irrespective of the actual device configuration. Furthermore, such systems can be configured into either simple parametric test systems or more complex life-test equipment simply by the number and interconnection of these basic PC card resources. The challenge is to manage the testing across all cards in such a way that system build can be as flexible as possible to target this end-user requirement, a challenge that has been met with novel hardware linked by special synchronising signals and by new more 'intelligent' software.

C. Basic considerations regarding the PC as a platform.

Before committing ourselves to the PC as a platform for this Reflex architecture, we gave careful consideration to the tradeoffs between its obvious low-cost and popularity and the technical limitations that may affect our use of it in a relay test scenario. In an ideal world, one designs a test system starting with a suitable electrical back-plane that accommodates not only the processor signals to interface with each hardware module, but also any special custom test signals that are always required by the test system itself, and in this respect the PC platform appears very inflexible. In addition, we felt that as a general purpose platform for test equipment it was less than ideal unless we could come up with positive solutions to the following questions that we posed to ourselves:

How would we built a system physically to suit harsh environments?

This covers topics such as housing, cooling, ease of maintenance and upgrading, all of which have serious cost implications for commercial applications. On examination of the industrial implementations of the PC platform housed within a 19-inch racking chassis, these points are well thought out and are very different to the casual, almost disposable approach taken with the more popular desktop PC. As examples, some 'off-the-peg' industrial housings are shown in Figure 3.

Figure 3 The flexibility of the industrial PC platform

What steps would we take to facilitate the quickest build or upgrading of new or existing systems?

The Reflex architecture is based on being able to quickly configure a system to an end-user's requirements by simply inserting resource cards into a card frame, thus it is important to make this procedure as simple as possible, especially if this same procedure is expected from the customer during an upgrade at some stage. Traditional tasks when installing cards into a system require the setting of various 'personality' or address switches to inform the system of the card location and function and this is an area where mistakes are easily made and where the small switches used to implement these settings can themselves introduce unreliability over the long term. Present day PC's implement

a generic 'plug-and-play' definition for I/O cards that already alleviate these set-up problems and we decided to enhance this further to create our own fully-automatic installation procedure that provides us, and the end-user, with the following benefits:

- Our cards have no switch settings, they are simply inserted into the back plane.
- The actual card address, its exact function and capability are scanned by the software at start-up, leading to a fully dynamic and re-configurable system that is easy to upgrade.
- The path to future cards with additional functionality is clearly paved.

How would we construct systems that require a large number of cards?

This is an important requirement for systems that will test multi-pole devices or for larger life-test systems. Desktop PC's are very poor in this respect, offering only a few I/O slots as well as being very restricted and unpredictable mechanically. By contrast, a single industrial PC chassis can offer up to 20 slots in a defined and planned mechanical layout, quite adequate for most small or medium relay test systems. Smaller units are also available with reduced numbers of slots. For the largest systems we have extended this concept further with a scheme whereby we can link multiple industrial PC chasses into a 'virtual' PC containing many more I/O slots, for example to cover (say) a 200-contact life-test system. This scheme has required us to create a means of performing the 'invisible' linking of the chasses as well as the implementation of a novel shared-address scheme that allows us to add an almost unlimited number of cards to a system without having to consider address range issues.

How would we pack our contact and device resources into the limited PC card PCB area?

Due to the fact that the PC was originally designed with the objective of housing only simple I/O cards, the actual available PCB area on each card is rather limited compared with the higher cost alternatives such as VME or VXI. This space limitation poses a challenge that can only really be met by using surface mount components where possible and by using a high level of integration such as the newer >10,000 gate in-circuit programmable FPGA devices. Programmed using a high-level logic language VHDL, these logic parts have been found ideal to host the sophisticated contact timing measurement and other logic required to test relay devices.

Finally, having attended to each of these points it was clear that we were actually able to harness the low-cost of the PC platform at the expense only of implementing some inter-card connection scheme to route signals required by our relay test requirements.

II. CARD RESOURCES.

A test system constructed with Reflex cards is typically partitioned into three areas, each implemented by one or more cards:

- System control (typically one card)
- Device control (this card controls device coils, and there are as many cards as are required by the application – i.e. one card per device)
- Contact interface (this card interfaces with device contacts, and again there are as many cards as are required by the application – one card per contact for example).

This section will look at typical implementations of these card functions in more detail.

A. System card.

Figure 4 shows the system control card. Typically this card fulfils the task of interfacing the system to the controls in the outside world, allowing device tests to be synchronised with mechanical handling activity and to report the status of test results for sorting.

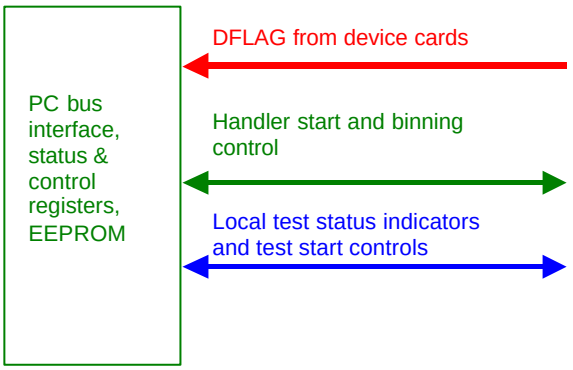


Figure 4 System card

Although such simple functions could easily be 'lost' within one of the other cards, the existence of a separate system control card is important to the Reflex architecture and results in extremely flexible systems as we shall see in later sections.

B. Contact card.

This section looks in more detail at the contact card and its resources. The block diagram of a fully populated contact card is shown in Figure 5.

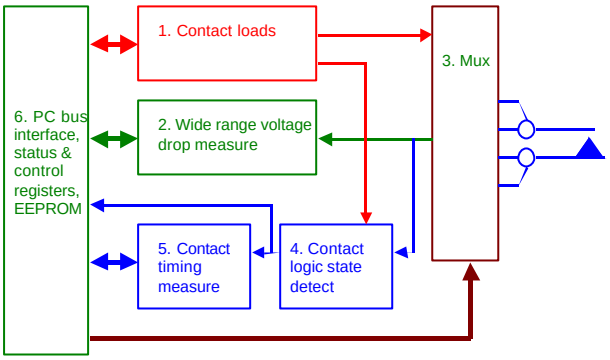


Figure 5 Contact card resources - block diagram

This circuitry incorporates established and proven technology already in use by us and with the major components being:

1. **Contact loads.** On-card resistive loads of 10,60,100,300 and 1k are provided, which together with a load voltage generator of +/-20mV to +/-10V, allow a wide range of low and medium level contact environments. The loads can be disabled for dry circuit switching or to use the system in CVD (contact voltage drop) mode, and a special option takes the on-card load capability further up to +28V, 100mA.
2. **Versatile contact voltage drop measurement.** DC or A/C voltages can be measured at the contact sense terminals in the presence of open-circuit load voltages up to +/-400vpk.
3. **A flexible multiplexer**. This routes the load and voltage measurements to the contact connections to permit measurement of contact resistance (CR), contact voltage drop (CVD) Kelvin fixture check, self-test and calibration.
4. **A contact open / closed state comparator.** This operates at a programmable voltage that can be a simple fixed value, or a percentage of the load voltage (e.g. 90%).
5. **Timing logic.** This logic provides results for operate and release time, bounce time and number of bounces. The bounce criteria is also programmable to exclude events shorter than specific limits.

Timing logic – block diagram.

Figure 6 shows the block diagram of the logic used to monitor contact operate / release and bounce times. This logic is simple in concept but complex in its number of registers and their interconnection, prompting us to implement it within one of the newer >10k gate FPGA devices, allowing us to incorporate it together with other card logic and to obtain other benefits such as flash-programming and ease of changes.

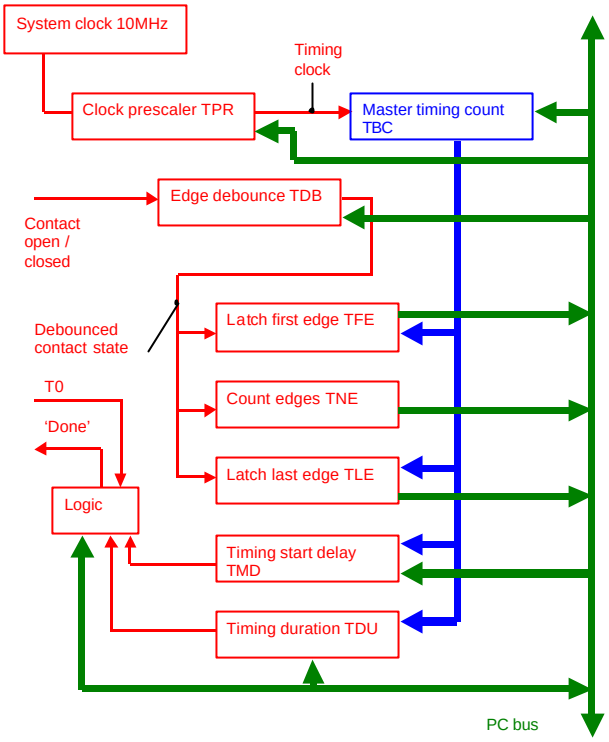


Figure 6 Block diagram, timing measurement

Timing measurement operates as follows.

1. The system clock (10MHz) runs all the time and is divided by the prescaler TPR to produce the actual timing sample clock, for example 1us (divide by 10).
2. The timing clock feeds the master counter TBC which starts counting upward from zero at T0 and counts upward.
3. When the master count reaches the value in the start delay register TMD, contact monitoring commences and the registers TFE, TNE and TLE latch values for the time of the first edge, the number of edges and the time of the last edge respectively.
4. Finally, the master count reaches the value in the timing duration register TDU – this defines the total time allocated to the timing measurement and the 'timing done' flag is asserted.
5. Following a timing measurement, software fetches these register values and processes them to produce operate and release time, bounce times and number of bounces.

A/D measurement logic – block diagram.

Figure 7 shows how an A/D is provided 'behind' each contact. This may seem an extravagance, but the performance and flexibility that results is well worth the additional expense. In fact simply providing the A/D itself is not the real problem, the real challenge being to obtain a wide range of A/D measurement modes and qualities depending on the system application. Our high-integration FPGA allows us to solve this problem again by providing control logic that handles the A/D

without any software intervention at all, permitting truly parallel test systems to maintain their timing independent of the number of contacts being measured. The logic allows us to make a general-purpose voltage measurement over any integration period, with flexible sample counts and to automatically perform phase-sensitive rectification on known AC signals whilst largely eliminating unwanted induced noise and AC pickup [4]. As a result, this voltage measurement is used for measuring contact voltage drop, contact resistance as well as for the Kelvin connection checks, self-test and calibration.

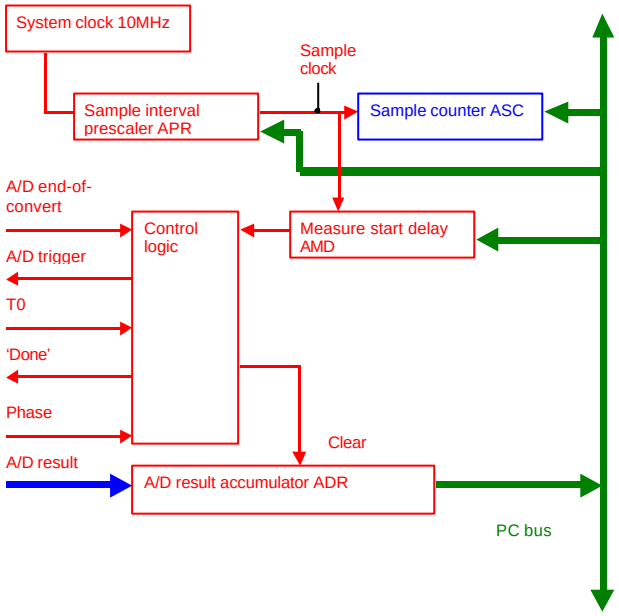


Figure 7 Block diagram, A/D intergation

A/D integration operates as follows.

1. The system clock (10MHz) runs all the time and is divided by the prescaler APR to produce the actual A/D sample timing interval, for example 30us.
2. The sample clock feeds the sample counter ASC which holds the required number of samples and it also feeds the start delay counter AMD which delays the start of the AMD counter to offset the A/D measurement if required.
3. At the start of the integration, the result accumulator ADR is cleared to zero.
4. During the integration, each sample causes the A/D to be triggered and its result added to the accumulator. This register is 32 bits, allowing for 65536 (16-bit) sample count with a 16-bit A/D result. Note the 'phase' signal which allows this logic to accommodate AC measurement by instructing the logic to either add or subtract from the accumulator.
5. Finally, after the integration, the processor reads this value and performs the necessary sample count division to normalise it back to a useful value.

This logic is very flexible because it can be programmed to cover a wide range of integration times and measurement qualities. This is particularly useful in a life test situation where short measurement times are desirable, but where some investigation is still needed for the end-user to establish the best integration time versus measurement performance trade-off.

C. Device card.

This section looks in more detail at the device card and its resources. A typical device card is mainly responsible for controlling device coils with a block diagram typically as show in Figure 8.

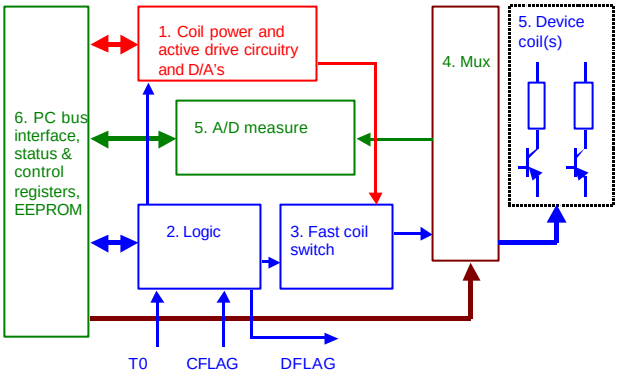


Figure 8 - Device card resource block diagram

Basic device card capability.

The major components of the device card capability are:

1. **Coil power supply**. This is an interface to a local power module that provides a programmable voltage and current bipolar supply. In addition to this coil supply there is optional active drive circuitry to accommodate transistor and FET-based relays where required.
2. **Control logic** manages the coil on/off switching and response to the bus signals T0, CFLAG and generates a 'device done' flag DLFAG. In addition, this logic contains the ability to 'step' a nominated D/A, auto-incrementing or auto-decrementing it to make processor independent measurements of the device operate or release characteristics.
3. **A fast coil switch** is provided between the coil power and the device to enable accurate timing tests.

4. **A coil terminal multiplexer** allows various modes of coil power connection to the device, including fixture tests and combining or isolating coils for bistable testing.
5. **Coil drivers** suitable for monostable or bistable single or dual coil drive, with or without active device drive.
6. **The PC bus interface** to access various card registers and to configure the card prior to, and after, a hardware test phase.

Stepping logic – block diagram.

For true parallel device testing, the device card is capable of auto-incrementing or auto-decrementing coil or active drive D/A values without processor intervention – this permits hardware determination of operate and release voltages or currents. Figure 9 shows the block diagram of the logic used.

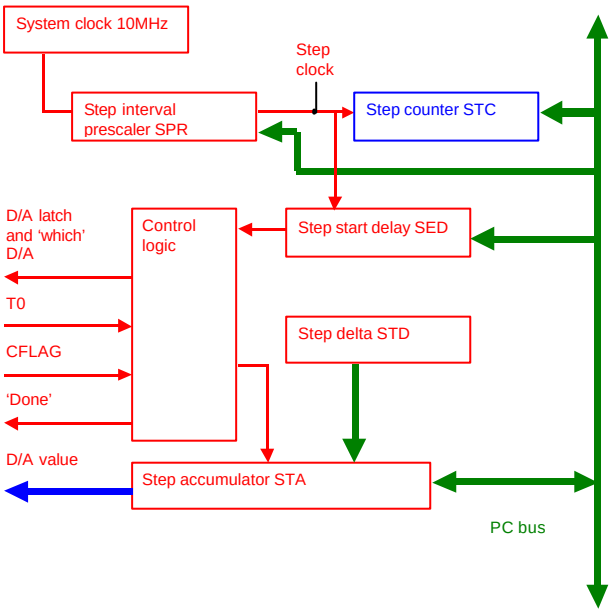


Figure 9 Block diagram, step logic

The step logic operates as follows.

1. The system clock (10MHz) runs all the time and is divided by the prescaler SPR to produce the step timing clock, for example 1us (divide by 10).
2. The step clock feeds the step counter STC, which starts counting downward from an allowed step count after the step delay SED elapses. Step activity is completed when this counter reaches zero or on CFLAG, depending on the mode of operation.
3. On every step, the value in the step delta register STD is either added or subtracted from the step accumulator STA.
4. When the value in the accumulator STA changes, the control logic issues a load pulse to the appropriate D/A, transferring the STA value to it.
5. Finally, the step activity is complete and the 'done' flag is set. The processor can read the step count to determine the exact point at which device activity occurred.

III. CARDS BECOME SYSTEMS – THE INTER-CARD CONTROL SIGNALS.

When a relay test system is configured using these Reflex cards, its effectiveness is due in no small part to the design of some key global control signals that inter-link the cards. These signals actually have nothing to do with the PC architecture and have to implemented as custom wiring, yet they enable the cards to co-exist and to form a true relay test system. It would actually be possible to implement a relay test system without these signals and with the card resources controlled totally by software alone but the measurement timing of such a system would be very uncontrolled and there would be a significant degradation of performance as the number of cards increased, exactly the opposite of that which we wish to achieve with a fast multi-contact life-test system. These inter-card signals synchronise and flag key device activity and release the controlling software from having to meet unrealistic timing obligations.

There are three main control signals, typically bussed across all cards and with functions as follows.

System timing (T0).
This signal is asserted at the start of a hardware test (possibly to initiate a measurement of CR or timing) and its assertion marks the start of all synchronised activity across all of the required resources, for example if the test system is testing a 4-pole changeover device for contact resistance, all 4 closed contacts are measured at exactly the same time with no software delay or uncertainty. This master signal is issued by the system card and is received by all other cards.

'Contacts' flag bus (CFLAG).
This signal is asserted when all local tasks relating to contact resources are complete, or when a required contact target state is reached. This signal is issued by contact cards and is received by a device card. As such, events such as the completion of a parallel CR measurement can be monitored on a device basis. A feature of the exact electrical definition of this signal is that it can be programmed to indicate either a group event (the usual case where for example all CR measurements have finished) or individual contact events (for example to measure the operate voltage of each contact during an operate coil voltage ramp). More about this 'group' and 'individual' capability later.

'Device' flag bus (DFLAG).
This signal is asserted when all tasks relating to the device are complete. This signal is received by the system card and is normally an output from all device cards. As with the

CFLAG line, it can be programmed to be asserted on 'all' devices, or on 'any' device depending on the nature of the test.

A. *An example of how these key signals measure operate voltage.*

The action of these key relay test signals can be seen in the timing diagram of Figure 10 which shows how the system measures the operate voltage of a relay without software intervention during the applied test ramp and despite the fact that device contacts may be arbitrarily distributed across separate contact cards. Without these control signals this measurement would only be practical using significant software querying intervention and with the attendant timing uncertainty.

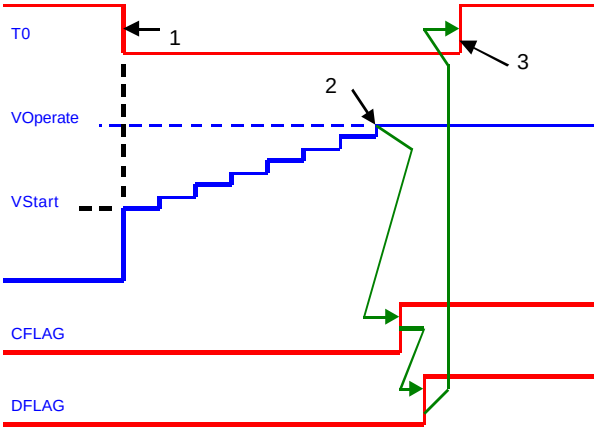


Figure 10 - Operate Voltage measurement

The measurement is a simple 3 step process.

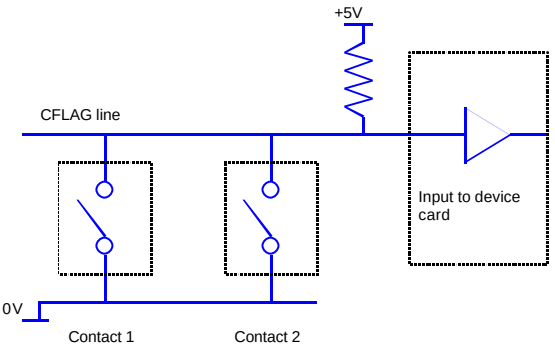
- At '1', the synchronising signal T0 is asserted to indicate the start of the test. This causes the device card coil resources to commence ramping the coil voltage from VSTART upwards and using the 'stepping' logic discussed earlier.

- At '2', the device is completely operated causing the CFLAG line to become asserted, and causing the coil voltage ramp to stop at the device operate voltage. Note that in this mode, the CFLAG line will only become asserted when ALL contacts have reached their target state, i.e. when the device is operated. (As will be seen later, another mode is possible where CFLAG is asserted and stops the ramp on each contact state change, should individual contact operate voltages be required).

- At '3', the device flag line DFLAG becomes asserted to indicate that the device activity is finished and this signals the system card to note the end of the test and to return T0 to its inactive state. The software can now interrogate the device card to read the actual final coil voltage and to reset the various hardware ready for another test.

B. *Electrical wiring of the key signals solves the monitoring of contact events*

Most of the power of these timing signals is in their flexibility, being able to indicate single or group events (for example either a single contact change or a complete device operate condition). This is achieved by linking the actual signal meaning to its polarity.



Mode	CFLAG asserted when	Contact activity to assert CFLAG	Typical application
Group (all contacts)	High (+5v)	All contacts must open their 'switch' to release CFLAG to go high.	Monitoring for device operated / released.
Single (any contact)	Low (ground)	Any contact may close its switch to pull CFLAG line low.	Monitoring for change / fail in any contact.

Figure 11 Combining contact information with CFLAG

Figure 11 shows the wiring of the CFLAG and DFLAG lines – a scheme termed a 'wire-or', so called because the signal calling in at various cards (contact #1, contact #2 etc) actually creates a logical OR function where the line 'wants' to float high to +5v by the action of the resistor, but can be pulled low by one or more contact cards simply by each one closing its CFLAG output 'switch' (shown as 'Contact 1', 'Contact 2' etc). Thus contact #1 OR contact #2 OR contact #n can control the line – a situation analogous to an emergency brake cord requesting a stop action. By simply defining the meaning attached to a contact switch closure we can use the line to indicate device contact activity across many contacts, and the table shows how the line can be used to indicate the state of a group of contacts, or simply that of a single contact.

IV. HOW THE REFLEX ARCHITECTURE BUILDS VARIOUS SYSTEMS.

These control signals are actually very powerful and be used to link contact and device cards in either a 'flat' or 'hierarchical' manner to construct test systems with various parallel test characteristics. This section looks at how to configure a number of different relay test systems from Reflex resources and depending on the end application of the system.

A. A simple parametric test system.

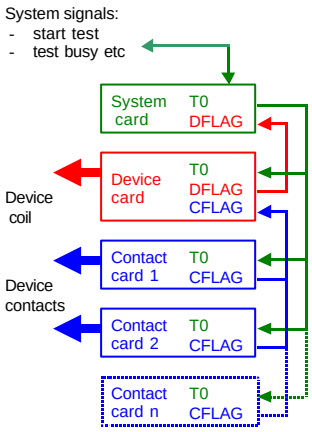


Figure 12 - Signal wiring - basic system.

Figure 12 shows a basic system for testing one relay device. A system card manages system timing and external control signals, while a device card controls the device coil and two contact cards interface with the contacts. These cards are linked with the T0, DFLAG and CFLAG signals as shown, allowing additional contact cards if required. A system configured in this way would be similar to the RT290 parametric test system in its 'single device' functionality.

B. Testing two devices synchronously.

In many production situation more than one device is tested at a time, for example as part of an automatic handling system where the indexing time is comparable to the device test time and where this is then the only way to achieve increased throughput. The Reflex architecture has been designed to accommodate this by simply configuring the required cards as shown in Figure 13.

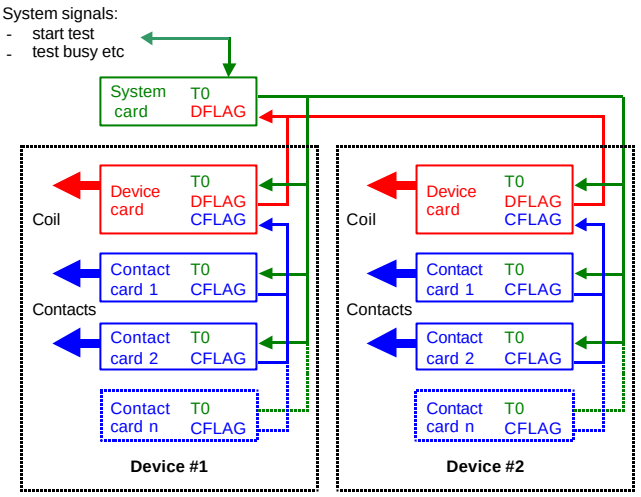


Figure 13 Signal wiring - dual system.

This shows two devices – device #1 and device #2, each with their own resources much as for the simple single system in Figure 12 but with a common system card. The difference is that:

- The T0 system timing line is taken to all cards so that all cards act simultaneously. Note that this imposes the restriction that the test for each device must start synchronously (although the two devices could be different and be tested to different test programs).
- The DFLAG (device ready) line is joined between the two device cards, allowing the system card to respond to devices as a group, or singly depending on the software programming of its assertion state.

Conceptually, this paralleling of devices can be extended indefinitely to test many devices in parallel – note though, that the controlling software must be able to understand the wiring configuration and manage the resources for configuring a test program and collecting result data. As we will see shortly, this is not too difficult given today's processing power, object programming and multi-threading.

C. A life test configuration.

To create a life-test system, the architecture of the dual system can be extended to provide as many 'coils' and 'contacts' as required, and the T0, CFLAG and DFLAG signals connected as shown in Figure 14.

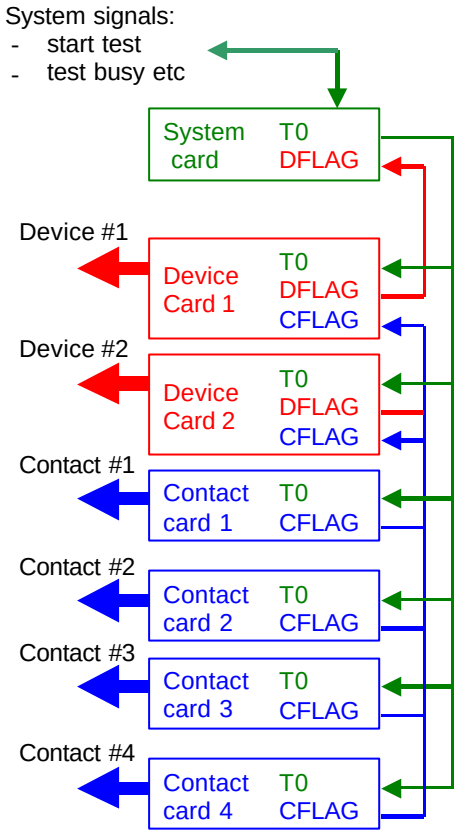


Figure 14 Life test system architecture

This shows a '2 device', '4 contact' life test system, synchronously testing from the common timing signal T0 and with a common contact flag CFLAG and a common device flag DFLAG. The fact that these signals are common across all devices and contacts breaks the architecture somewhat but only restricts the system to testing all devices at the same time and whilst it would be possible to run different devices on this system, the system would cycle at the speed of the slowest device. (By adopting the architecture of the dual system where system control and CFLAG routing accurately reflected the device construction, it would be possible to run life tests on independent devices at different rates, although this would bring increased software reporting complexity).

Clearly this represents a larger system that can be built with Reflex components and requires not only a larger number of cards, but power supplies and other ancillaries as well as the industrial PC rack. To illustrate this, a typical physical implementation of the Reflex 50 life-test system is shown in Figure 15.

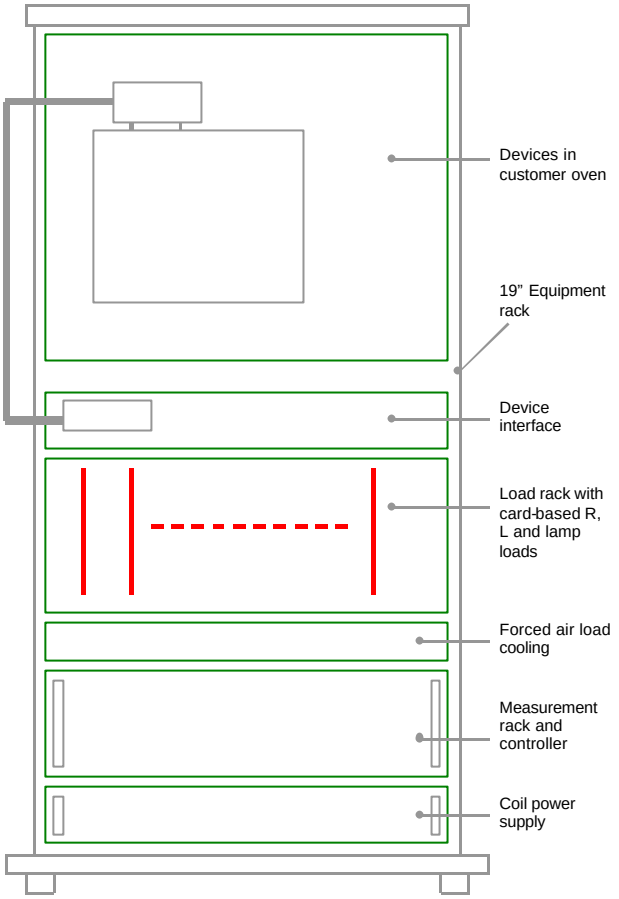


Figure 15 Rack mounted items – Reflex 50 Life Test System.

D. Testing two devices asynchronously.

As an extension of the synchronous dual system, very occasionally it is required to test multiple devices but with each device handled completely independently, for example where each device might be located within a separate mechanical handling system and cannot be guaranteed to be ready for testing at exactly the same time.

The Reflex architecture can handle this by simply adding more system cards to create multiples of a basic single system and as shown in Figure 16.

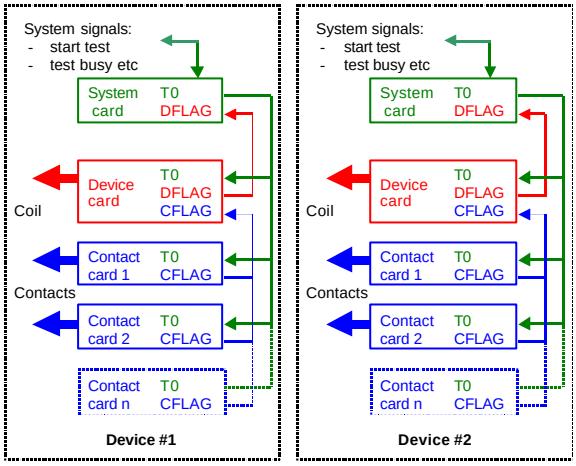


Figure 16 Multiple asynchronous system

As can be seen, devices #1 and #2 know nothing about each other at all, and although the control is by common software that sets up and collects data from all cards in both systems, the two systems actually operate independently. Again, it is instructive to note that the Reflex architecture can easily accommodate this requirement.

V. HOW THE REFLEX ARCHITECTURE GUARANTEES MEASUREMENT TIMING

Although not always essential, it is desirable to make device measurements in a repeatable way between tests, for example a specified time should elapse between applying operate voltage to a device coil and the closed contact being measured for resistance. In a software-driven system this often results in guaranteeing that at least a minimum time will elapse for this – enough to ensure that the contact is indeed closed and has become stable – but this still causes uncertainty about the actual point of measurement and may lead to inefficiency where the testing of small, fast devices becomes limited by software response times such as in reed relay contact or micro-relay life testing.

The Reflex architecture with its inter-card signals avoids this pitfall and results in accurate measurement timing across large numbers of devices, releasing the software to simply gather result data after a measurement has completed. In this section we will look at how this timing accuracy is achieved.

A. Timing of events during a CR measurement.

Let's look more closely at the interaction of these CFLAG, DFLAG and T0 lines in a typical system, perhaps as shown in the simple single relay test system of Figure 12, where there is one device card, two contact cards and a system card.

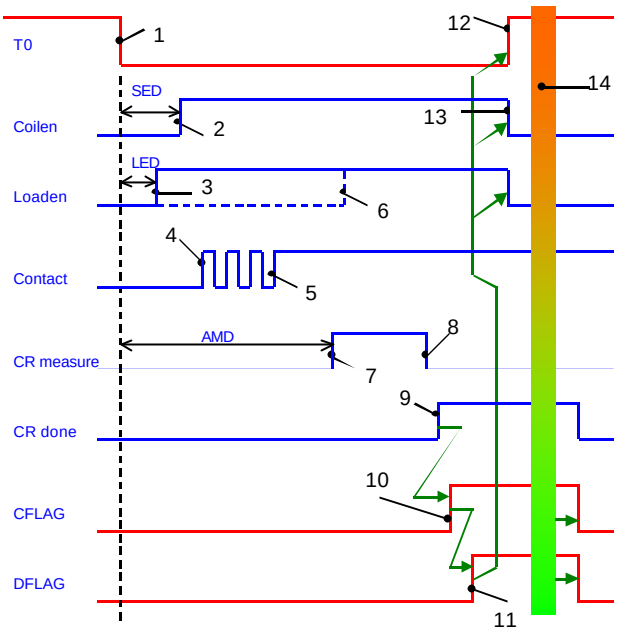


Figure 17 Example system signals – CR measurement.

Figure 17 shows how these signals interact to complete a contact resistance (CR) measurement on all contacts. An important point to note is that all measurement timing is controlled by hardware, not by the controlling PC. This is important for two reasons – firstly because although processor speed has increased dramatically over the last few years, so have the number of 'background' tasks such as printing, networking etc, all of which 'steal' time away from an application such as relay testing and – worse – can be very unpredictable. Secondly, hardware-derived timing does not degrade as further hardware resources are added, making it possible to implement potentially large life-test systems without significant impact on measurement time.

The actions during a CR measurement are as follows.

- At '1', the T0 line is asserted by the system controller to indicate the start of a test phase (in this case a CR measurement on all contacts in parallel).
- At '2', the coil drive is enabled after a programmable delay SED.
- At '3', the contact load is enabled after a programmable delay LED – this delay and its relation with the coil delay SED allows a variety of dry and wet contact switching to be obtained.
- At '4' an example contact changes and starts to bounce until '5'. In this example, the load is already enabled so the contact switches 'wet'. Alternatively, if the load were delayed until '6', the contact would switch dry.
- At '7', the contact resistance measurement starts after a programmed delay TMD.
- At '8', the measurement is complete and asserts an internal CRDONE flag at '9'.

- The CRDONE flag for each contact releases its CFLAG output and may remove the load (depending on load mode), but CFLAG will not go high until all contacts have released it (at ‘10’), allowing the system to observe CFLAG as relating to all contacts completing their measurement.
- At ‘11’, CFLAG is seen asserted by the device card, may clear the coil drive (at ‘13’, and depending on coil mode) and causing it to release its DFLAG line, indicating that device activity is complete.
- At ‘12’, the system card has observed that DFLAG is asserted and ends the test phase by clearing T0.
- At ‘14’, the system responds to the end of the test phase by the processor interacting with the hardware to clear down the various hardware flags and then to read the CR result information from each card.

For completeness, the timing of events during a timing measurement and during life-testing are reproduced in Appendix I and II respectively, and it will be seen that there is a great deal of similarity in these to the CR measurement just illustrated.

VI. OTHER DEVELOPMENTS OF THIS REFLEX TECHNOLOGY

We are already building on the foundation of the Reflex architecture with components that permit tests other than the traditional contact and coil measurements. For example, we have now implemented the requirements of CECC and MIL-spec [1] to create a fully digital 8-channel chatter detector card for device vibration monitoring, and we plan to extend the range further.

APPENDIX 1 – TIMING OF EVENTS DURING A CONTACT TIMING MEASUREMENT.

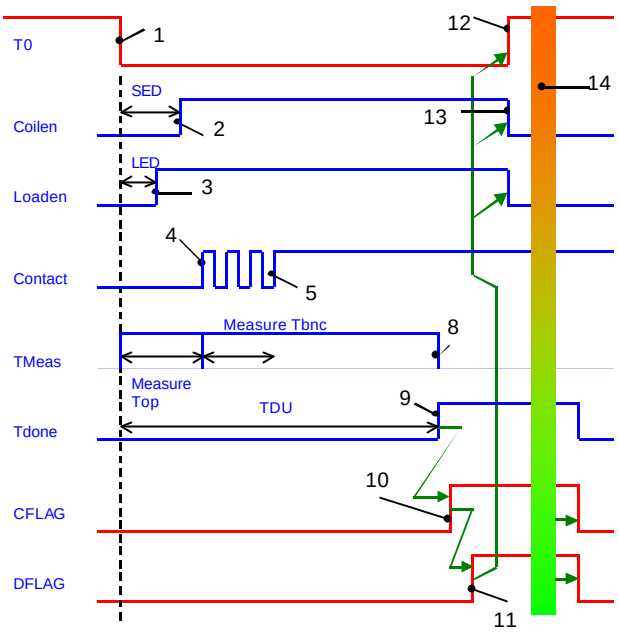


Figure 18 Example timing signals - Timing

The actions in Fig 18 during a timing measurement are as follows. These signals are identical to the contact resistance diagram of Figure 17 with the exception of those indicated below.

- At ‘3’, the load is enabled before any contact activity is expected, since timing requires that a load be available to determine the contact open / closed status.
- Synchronised to T0, timing counters monitor for each contact ‘first edge’, ‘number of bounces’ and ‘last edge’, with results available at ‘8’(TDONE) after the allowed timing duration TDU.

APPENDIX II - TIMING OF EVENTS DURING LIFE-TESTING.

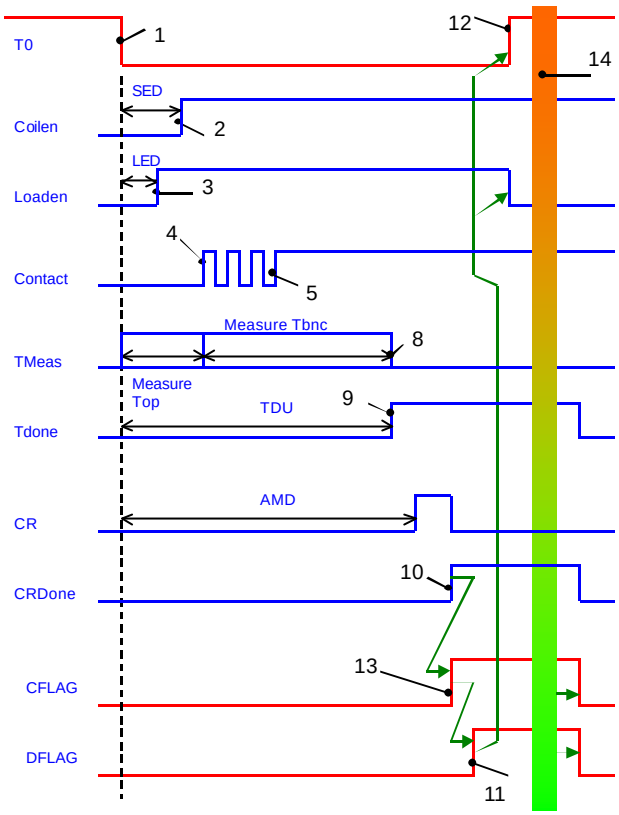


Figure 19 Life-test timing

For a life-test system, the Reflex architecture shows its true worth. Figure 19 shows the timing signals to make CR and timing measurements on every contact and on every operate and release cycle during a life-test. This next description goes through the signals and how they relate to a life test. Note that this only shows the 'operate' phase of a life-test cycle - operate and release phases are actually identical in hardware and only the software data read actions and specific contact activity changes.

- At '1', the T0 line is asserted to define the test phase start for the operate series of measurements. All hardware timers start running.
- At '2', after the programmed delay SED, the device coil is enabled. (In this life-test mode, software has configured it to remain active until the next T0 at the start of the release phase).
- At '3', the load is enabled ready for the contact switching.
- At '4', an example first contact edge is seen and its time is recorded in the 'first edge timer'. This will produce the timing result 'operate time' OT.
- Between '4' and '5', the contact bounces and at the end of this time, counters have values that will become 'operate bounces' (ONB) and 'bounce time' (OBT) taken from the time of the last edge.
- At '8' the allowed timing measurement duration ends, producing a 'timing complete' signal TDONE.
- After the CR measure delay timer AMD expires, the CR measurement (or contact voltage drop) starts, completing its measurement at '10' and setting the CRDONE flag.
- At '13', with timing complete (TDONE) and CR complete (CR DONE), the contact can assert its CFLAG, and the bus CFLAG line will assert when all contacts are complete (typically at the same time).
- At '11', the device card observes CFLAG and responds by setting its DFLAG output.
- At '12', the system card observes DFLAG and uses this to de-assert T0 at E23, signalling the end of the test phase and that the processor should now intervene to collect data and prepare the system for the release phase.
- At '14' there is processor activity, to collect timing data and to clear CFLAG, DFLAG and to configure the system ready for the release phase. Typically, in a life test situation, the load is required to remain ON if a contact fails – so after pass / fail classification at '14' the load can either be removed (the normal case) or remain on with a halted system. (Note that it is also possible that the same load is present and ON for the entire life-test cycle – load modes permit this to be selected in software).
- When the processor has configured the system ready for the release phase, another T0 occurs and the cycle repeats as shown for operate. There is no difference between the release and operate timing, only that CR (or contact voltage drop) is measured across normally closed contacts instead of normally open contacts.

CONCLUSION

The 'Reflex' relay test components developed by Applied Relay Testing implement a flexible solution that permits a wide range of test systems from simple single-seat operator test stations up through automatic handler-based test through to large life-test systems. The flexibility and ease of configuration achieved with the described technology translates into a good match between the device configuration and the test system and achieves an excellent price-performance trade-off.

ACKNOWLEDGEMENT.

Applied Relay Testing Ltd thanks the following companies for their support: Axicom AG (Switzerland), Teledyne Relays (CA, USA), CII Technologies Inc (NC, USA) Pickering Electronics (UK).

REFERENCES.

[1] Specification CECC 16000 Section 5.28 (Vibration), MIL-R-39016E section 4.8.10 etc. (Shock).

[2] B.J.Frost "A new generation of test equipment" *Proc. of the 45th relay conference, NARM, April 1997*

[3] W.P.Taylor et al, "Batch Fabricated Electromagnetic Microrelays" *Proc. of the 45th relay conference, NARM, April 1997*

[4] B.J.Frost "Real-time analysis of the Contact Electrical Environment During Switching" *Proc. of the 47th relay conference, NARM, April 1999*

[5] Helmut F.Schlaak et al, "Silicon Microrelay – A Small Signal Relay With Electrostatic Actuator" *Proc. of the 45th relay conference, NARM, April 1997.*