INTEGRATION AND TESTING OF MICROPROCESSOR-BASED SYSTEMS

The process of integration is the combining of the independently tested software and hardware modules to produce a prototype system. Invariably the integrated system does not work, precipitating the pointed index finger phase of the design procedure whereby the hardware and software engineers simultaneously raise their index fingers to each other and exclaim "It's your fault". - They are of course both correct.

Failure at switch on after integration is often the consequences of poor design procedure. For example, the following procedure will inevitably lead to problems: at an early stage in the development of the system the specification is divided into hardware and software work packages, two design teams are then formed, installed in different rooms if not separate floors of a building, and then produce their part of the system. The next time these teams meet is at the integration stage. To illustrate what can occur consider the design of an analogue data acquisition unit. (figure 1)

The microprocessor is required to control the sampling of the analogue signal that is converted by the ADC. The conversion is initiated by the "Start Of Conversion" (SOC) control line; when the conversion is finished the ADC informs the microprocessor by means of the "End Of Conversion" (EOC) control line.

Data and control signals are conveyed via two ports as selected by the address decoder.

Faults that can arise due to poor inter personnel communications and lack of documentation are manifold, even in such a basic unit. Two examples are given below.

1. The port addressing is not unique and conflicts with another port:

   ACTION: The hardware team change the port address without informing the software team.

   RESULT: EOC is never detected as the program is reading the wrong port, the system fails.
2. It is a critical real-time system, that must cycle once every 4 seconds. The conversion time of the ADC is 50 ms. The input data is found to have noise. 

ACTION: The software team take 10 samples per cycle and average them, without first finding out the conversion time of the ADC. 

RESULT: The system cannot cycle fast enough. 

Such faults can be prevented by ensuring proper communications between design teams; preferably there should be only one integrated design team with all members providing input on both hardware and software. Regular project meetings should be held, controlled by a single project manager, and all documentation kept up to date. 

Other problems will still arise, but they can be minimised by ensuring maximum co-operation between disciplines. The faults that do occur are often to be found in the hardware/software interfaces, or are timing problems. The former are to be expected as this is the first time that the interfaces are tested. The latter arise from the fact that the system is being tested in a real-time as opposed to simulated environment. 

It is essential that the testing of the integrated system is carried out in real-time with the microprocessor operating at full speed, executing a program held in its ROM. These demands on the test environment means that the system has to be treated as a black box, whereby the processes being performed (ie the execution of the program, and transfer of data around the system) are hidden inside the microprocessor chip and are occurring at a high speed. 

Various tools have been developed which will enable the designers to observe the functioning of the system, without interfering with the operation of that system. Two such tools are an in-circuit emulator, and a logic analyser. Both tools are used to monitor the system, keeping a record of events that have occurred. The format of the recordings determine the applications of these tools: an in-circuit emulator is used in the diagnosis of software faults, the logic analyser for hardware faults. 

To use an in-circuit emulator the microprocessor is replaced by a d.i.l. probe which fits into its base. This probe contains another microprocessor of the same type under test, but is connected simultaneously
to both the prototype (via the base) and the in-circuit emulation system via a cable. The in-circuit emulator is now used to continuously monitor the activity on the pins of the microprocessor, whilst it is running in real time, without interfering with the operation of the microprocessor. A logic analyser, thus allowing the designer to directly observe a snapshot of a system output. The emulator is usually an extension of a software development system and the information captured is accessed via the development system keyboard display. This information is the states of the monitored logic lines (ie the microprocessor pins) at specific times, usually related to the microprocessor clock or instruction fetch cycles.

As well as the microprocessor probe, it is usual to also have a number of logic probes, which are used to monitor the state of logic lines anywhere in the system. However, unlike a logic analyser, these probes sample data relatively slowly, synchronised to the emulator clock.

In-circuit emulators should employ two separate memories, one for the operating system and the other for the user programs. The operating system memory is used solely by the computer that manages the functions of the in-circuit emulator. The user memory is available for use by the microcomputer system that is being tested. The operator has the option of storing the programs to be tested in either the memory that is part of the prototype system, or the memory within the in-circuit emulator.

Logic analysers are the digital equivalent of analogue transient recorders. The analyser acquires data via probes attached to points on the circuit. The data are stored in memory and traces are displayed on a CRT. Most analysers have the capability to handle at least 8 channels and very fast logic glitches can be detected.

The analyser continuously samples the logic signals at the points to which the probes are connected. The information that is displayed is determined by preset trigger condition and a frame width. The preset trigger condition is a logic pattern, as determined by the operator, which when present on all the probes simultaneously will cause data to be stored and the CRT to start displaying traces of the information. The frame width is the amount of time for which the data is captured for display.
In-circuit emulators and logic analysers usually have the facility to generate an external trigger, and alternatively can start tracing or capturing data upon the receipt of an external trigger. A powerful development aid can be configured with an in-circuit emulator triggering a logic analyser, thus allowing the designer to directly correlate a sequence of events in the hardware with the execution of a known part of the program.
Failure at switch-on or after testing is one cause of poor design procedure. For example, the following procedure will inevitably lead to problems when the software development and hardware design are separate. In this case, the software is developed independently of the hardware design. When the hardware is completed, the software is written and tested. This process can lead to errors if the software and hardware do not work together correctly.

The process can be illustrated using a diagram. The diagram shows the relationship between the software and hardware components. The software components, such as the ADC and SOC, are connected to the hardware components, such as the LATCH and DECcoder. The relationship between the software and hardware is indicated by the arrows on the diagram.