MONITORS AND DEBUGGERS

Debuggers

This section describes a number of debuggers that I have developed.

D.M.S.

Dunfield MicroScope is a general purpose high-level debugger that I created as a companion to my Micro-Ctoolsets. It provides source level debugging at both 'C' and Assembler language level, as well as dissassembly level debugging where required. Other features include: Multiple watchpoints, intelligent breakpoints, multiple stepping options and much more.
DMS is completely target independant. I provide DMSC, a compiler that allows you to define a new processor/instruction set, register descriptions, memory address spaces and block types, as well as the control commands which are sent to a small resident debug kernel on the target system.

DESMO

DEbug Scripting MOnitor is a PC hosted target-independant debug monitor which interfaces to a small resident kernel to perform physical operations on the target system. It provides memory and I/O read/write operations in 8, 16 or 32 bits. It features a powerful scripting language with high-level constructs such as variables, constants, loops and conditionals. Using the scripting language, complex operations can be automated into a single command. Debug scripts can accept arguments and produce results, and can be nested to provide higher and higher level operations. Use of DESMO scripts can significantly reduce the tedious and repetative manual operations required to test and debug a complex hardware project.

ARM Monitor/Debugger

As part of my work with an ARM7/ARM9 based telephone system, I have developed an ARM debug monitor and other diagnostic tools. The debugger in incorporated into my ArmOS operating system, and operates in an isolated context from the remainder of the system. When the debugger is entered (processor exception, memory protection violation, breakpoint, software panic, user request etc.) the current context of the ARM is saved and the monitor activated. Facilities include full display/update of all registers in all ARM modes, disassembly. stack backtrace, all the ususal debugger commands and more). At any time the processor context can be restored and execution continues at the point of the original exception. The debugger works from within all ARM modes (User, Irq, Privileged).

Resident Debug Monitors

These are target resident debug monitors which feature: Memory/Register display and edit, Interrupt vector control, Disassembly, single-step, breakpoints, code download and many more features. I have implemented resident monitors for many processor families including: 68HC08, 6809, 68HC11, 68HC12, 8051/52, 8080/85, 8086, 8096, Z80, AVR and ARM. I have used these debuggers in many of the small devices and prototypes which I have constructed.

8051 In-Circuit Debugger

This is a PC hosted debugger with small-footprint resident kernel for the 8051 processor family. With a small bit of hardware support, it provides full debug capabilities. It can also use the internal SRAM and memory mappng capability of Dallas DS5000/2250 processor family to implement a reasonably complete 8051 In-Circuit emulator. The resident kernel is also compatible with my Emily51 simulator to provide in-circuit simulation capabilities.

6809 Radio Monitor/Debugger

As part of a personal robotics project, I developed a wireless monitor and debugger for the 6809. This provides complete download and debugging facilities over a radio link.