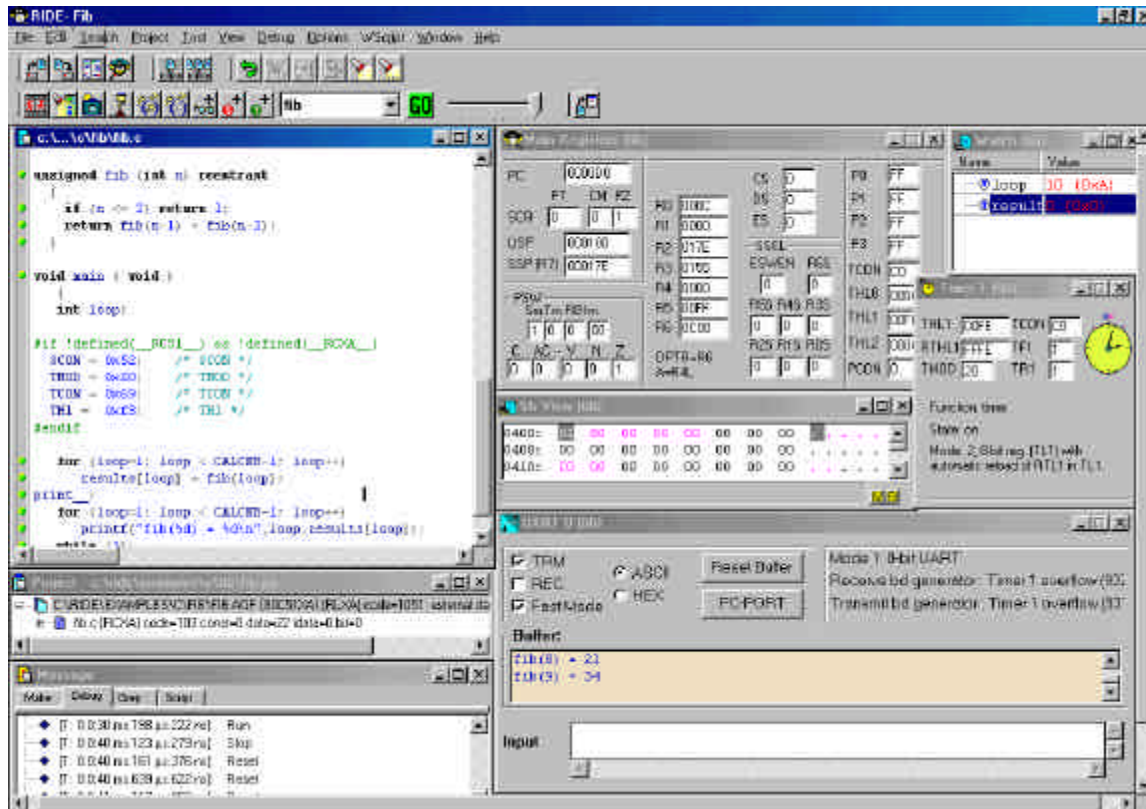


Rkit-XA is a complete toolchain for the XA family of micros. It features an optimizing ANSI C Compiler, a Macro-Assembler, a Linker, a Real Time OS, a ROM Monitor and an accurate Simulator, all integrated into the same Development Environment. Rkit-XA provides a clean and simple solution for migrating from the classic 8051 to the extended architecture of the XA.

Rkit-XA



www.raisonance.com



Integrated into RIDE

Rkit-XA is delivered with Raisonance Integrated Development Environment (RIDE), a fully integrated IDE featuring color syntax highlighting editor, project manager, on-line help and full control over all the tools of the toolchain.

From editing to compiling, linking and debugging (with the simulator, ROM monitor or real time emulator), RIDE increases productivity by putting all the tools at your fingertips into a coherent and easy to use user interface.

RIDE allows the simultaneous management of both 8051 and XA projects, by selecting the appropriate tools.

DOS 32-bit Version

RIDE also provides convenient access to 32-bit DOS version of the XA tool set. As an additional advantage, the DOS tools run in a DPML mode and are not memory limited.

Flexible Tools

All Rkit-XA tools are tightly integrated into RIDE, offering the developer a complete and cleanly integrated tool kit. In addition, each tool: assembler, compiler, linker, simulator, and so on, can be run "stand-alone" as an individual, self-supported tool.

Easy migration from the 8051

Rkit-XA provides an upward compatibility path for converting your 8051-based applications. Each tool accepts existing 8051 directives and syntax. In this way the new XA core is seen as a superset of the 8051. An 8051 based C language file is directly compiled. An 8051 based assembler file is easily translated by the integrated translator, and then assembled in the usual way.



RAISONANCE

Optimizing ANSI-C Compiler

A super-set of ANSI-C

RC-XA implements the ANSI standards for the C language, expanded with XA specific keywords and powerful new directives. 8051 specific keywords are included.

bit	Generic	Code	sbit	sfr
asm	At	_at_	near	far
interrupt	Exception	Trap	system	user
task	Priority	Group	swinterrupt	using
reentrant	Stack	Istack		

Memory Models

The memory model specifies the configuration and the size of the application:

Model	Memory type	Notes
TINY	Code+constant < 64 KB data < 32 KB	Mode: PAGE 0 Compact generic pointers
SMALL	Code+constant < 64 KB data < 64 KB	Mode: PAGE 0
COMPACT	Code < 16 MB Constant < 64 KB data < 64 KB (16MB far)	Code < 16 MB Constant < 64 KB data < 64 KB (16MB far)
MEDIUM	Code < 16 MB Constant < 64 KB idata < 64 KB (16MB far)	Code < 16 MB Constant < 64 KB data < 64 KB (16MB far)
LARGE	Code+constant < 16 MB data < 64 KB (16MB far)	Mode: standard far pointers by default

Base Types

- Integer Types: 8 bits : "signed char" and "unsigned char",
16 bits : "signed int" and "unsigned int",
32 bits : "signed long" and "unsigned long",
- Real Types: 32 bits : "float" (IEEE754 single precision),

Pointers Types

Two types of pointers are always available, generic pointers and space qualified pointers. Moreover, the distinction between near and far pointers allows to better tailor the code to the application constraints.

Code Optimizations

RC-XA optimizes the code to be both compact and fast. Many global optimizations have been inherited from our highly optimizing RC-51 compiler, others have been introduced specifically to suit the XA core. RC-XA allows to reach the following Benchmark results:

Program (model)	Code Size* (module / total)	Speed** (@ 12 MHz)
Sieve (Small)	156 / 1098	2.41 s
Dhrystone (Small)	1628 / 3304	194 ms
Whetstone (Small)	2624 / 7682	1169 ms

* Optimized Size

** Optimized Speed

Parameter Passing

One of the benefits of the XA core is the availability of several registers. RC-XA places the first parameters (up to 8 bytes) into registers, then pushes the others onto the processor's stack. This method permits recursion and makes the code naturally compact.

Libraries

RC-XA is supplied with ANSI C standard libraries as described in: math.h, stdio.h, string.h, ctype.h, and stdlib.h standard header files.

MA-XA Macro-Assembler

The MA-XA syntax is compliant with MA-51 (or ASM-51) for both directives and macros.

The instruction syntax implemented is compliant with the rules specified by Philips. In addition the old 8051 notations are accepted too. For example, both "@Ri" and "[Ri]" can be used to specify indirection.

ASM-51 to XA Translator

To ease the migration of existing 8051 assembly code to the XA, RKitXA includes an assembler instructions translator. This tool emits an AXA source file from an A51 source file by replacing 8051 instructions by their equivalent XA instructions.

RL-XA Linker/Relocator

RL-XA is based on existing RL-51 concepts, and is expanded by new directives. New concepts, like segment grouping, permit better CPU control and more efficient mapping of user memory. RL-XA produces a symbolic output format (OMF-XA), a HEX output file, and a complete listing file.

KR-XA Real Time OS

KR-XA offers a variety of services, and is fully compliant with KR-51 providing time and memory management, semaphores, synchronizing events, inter-task communications, and more sophisticated functions such as a perpetual calendar. The Kernel uses the multitasking features of the XA core to permit fast task switching, full preemption, and a time-slicing mechanism.

KR-XA is fully integrated into Rkit-XA. Task declaration is done at the C-source level, while configuration of the kernel is made at the project level. The SIMICE-XA simulator provides a variety of information on the status of the Kernel and each task.

SIMICE-XA Simulator/Debugger

The simulator is tightly integrated into RIDE and uses the information generated by RC-XA or MA-XA tools to provide fully symbolic high level debugging.

SIMICE-XA takes into account the characteristics of the selected device and simulates—in considerable detail—all internal peripherals. Associated with RIDE, SIMICE-XA permits simulation of multi-processor applications, and offers various solutions to simulate external inputs and outputs. SIMICE-XA provides various emulator-like functions such as trace management or complex breakpoint control. SIMICE-XA can also be run as a stand alone simulator.

MONITOR-XA ROM Monitor

RKitXA is supplied with a ROM Monitor, directly driven by RIDE. The ROM Monitor allows simple commands like "step by step execution" and variable inspection among others. If the target provides a "write CODE" capability, advanced features such as "execution line by line" or dynamic code breakpoint are possible too.

RAISONANCE

755, av. A. Croizat
38920 CROLLES

France

Tel. : (+33) 4 76 08 18 16

Fax : (+33) 4 76 08 09 97

Email: info@raisonance.com

4851 Keller Springs Rd. Ste. 205
Addison, TX 75001

USA

Tel : +1 877-315-0792

Fax : +1 972-818-0996

Email: info@amrai.com

Local distributor :