

# Technical Background on the Philips 80C51XA Architecture

CONTACT:

Peter Brown

Philips Semiconductors

408-991-3626

or

Kathy Keenan

Oak Ridge Public Relations, Inc.

408-253-5042

September 1994

# Technical Background on the Philips 80C51XA Architecture

Although 8-bit microcontrollers control the majority of electronic devices available today, increased microprocessor speeds, data transfer rates, and high-performance buses are fueling a gradual migration to 16-bit technology. Benefits such as dramatic increases in processing speed at a reasonable price point, the ability to use high-level languages such as C for embedded system development, and true multitasking capabilities, are causing embedded designer to evaluate 16-bit architectures.

The most popular 8-bit microcontroller architecture is arguably the 80C51. Over two decades of use have resulted in a large number of designers who are familiar with the tools and techniques of using the 80C51. These designers have large and valuable libraries of software routines developed for this architecture. An abrupt shift to a completely new architecture would result in a difficult and time-consuming transition for these designers.

Philips Semiconductor developed the 80C51XA architecture to provide embedded designers with an easy migration path to 16-bit performance. The 80C51XA is the only 16-bit microcontroller that is source code compatible with the 8-bit 80C51 architecture. As a result, designers can access their familiar routines and development libraries while expanding their expertise to 16-bit solutions.

## **The 80C51XA Architecture - An Easy Migration to 16-Bit Power**

The goals Philips set in designing the 80C51XA were twofold: The preservation of source code compatibility with the 80C51 was critical to deliver the needed migration path and the 80C51XA also had to be a clean, high-performance 16-bit microcontroller that did not require familiarity with the 80C51 architecture.

By permitting simple translation of source code, the XA allows existing 80C51 code to be re-used with a higher performance 16-bit controller meeting both objectives. The XA incorporates support for multitasking operating systems

and high-level languages such as C, while retaining the comprehensive bit-oriented operations that are the hallmark of the 80C51.

### **Subsets and Supersets - Software and Hardware Compatibility**

Many considerations were taken into account in creating the XA architecture. The primary objective was for a software translator to take 80C51 assembler source code and automatically produce XA source code. Reaching the objective required solutions in both software and hardware.

On a software basis most 80C51 instructions are subsets of more powerful XA instructions. Each 80C51 instruction translates into one XA instruction with the exception of instruction timing, where the XA provides the fastest execution feasible at a low cost.

In hardware, a major consideration in compatibility of the XA with the 80C51 is the memory map. The XA addresses this issue by having each memory area be a superset of the 80C51 (including registers, data memory, code memory, stack, special function registers. The sole exception is the processor stack where the stack growth direction is reversed to enable storage of 16-bit variables in memory, a key feature supporting the use of high-level language compilers such as C.

### **Register-Based Architecture:**

The architecture and instruction coding of the XA are optimized for register-based operations, which avoids the bottleneck problem inherent in a single accumulator register.

The XA register file is an overall superset of the 80C51 register structure. It allows access to eight words of data at any one time, which are also addressable in 16 bytes. The bottom four word registers are banked to allow the developer to minimize the time required for context switching as tasks are swapped or interrupt routines executed, and to provide more register space for complex algorithms. Some instructions (32-bit shifts, multiples and divides) allow addressing pairs of word registers as double words -- these pairs are always formed by adjacent word registers.

The upper four words of the register file are not banked. The topmost word acts as the stack pointer, while any other word register may be used as a general

purpose pointer to data memory. The entire register file is bit-addressable so that any bit in the register file, except for the 3 unselected banks of the bottom 4 words, may be operated on by bit manipulation instructions.

Special Function Registers (SFRs) provide a means for the XA to access special-purpose CPU registers, peripheral devices, and I/O ports. Any SFR may be accessed by a program any time without regard to any pointer or segment. The total SFR space is 1 Kbytes.

### **Data Memory — Upward Compatibility With the 80C51**

Sixteen megabytes of addressable memory shortens the development cycle and permits more complex real-time applications. Memory is divided into 64 kilobyte segments, improving performance and providing an intrinsic protection mechanism for multitasking. Segment registers provide the upper eight address bits for a complete 24-bit address in applications that require large data memories.

2 segment registers are used by the XA to access data memory, the Data Segment register (DS) and the Extra Segment register (ES). Each pointer register is associated with one of the segment registers via the Segment Select (SSEL) register. Pointer registers retain this association unless changed under the program control.

The XA provides flexible addressing modes. Most arithmetic, logic, and data movement instructions support the following modes: direct, indirect, indirect with offset, and indirect with auto-increment.

### **Code Memory - On or off-chip flexibility**

The XA is a Harvard architecture controller: code and data spaces are separate, giving the designer greater flexibility in handling code either on-or off-chip. The XA provides a continuous, unsegmented code space that may be as large as 16 megabytes. Most XA derivatives will also have an external data bus for off-chip data and code access, and may be used in ROM-less mode with no code memory used on-chip.

## **Multitasking Facilitated**

Multitasking allows several programs to run at once on the same processor, with a supervisory program determining when each program or task runs, and for how long. The XA's processor stacks were designed to optimize multitasking. Since each task in a multitasking operation shares the same CPU, the system resources required by each must be kept separate and the CPU state restored when switching execution from one task to another. By dividing into two separate modes of operation, the System Stack which supports the supervisory program in system mode, and the User Stack to support basic tasks in user mode, the XA is able to run several programs at once under the direction of the supervisory program, while protecting user mode tasks from accidentally destroying data on the system stack.

Code running in system and user mode use different stack pointers. The system stack is always located in the first 64K data memory segment to take advantage of the fast on-chip RAM. The user stack is located within each task's local data segment, identified by the Data Segment register.

The four register banks are a feature that is useful in small multitasking systems by using each bank for a different task, including one for system code. This means less CPU state that must be saved during task switching.

## **Instruction Syntax Optimized for Control Applications**

Users of the 80C51 will find the instruction syntax for the XA familiar. The XA instruction set is a diverse set designed to support most common control applications. Instruction encoding is optimized for commonly used instructions: register-to-register or register with indirect arithmetic and logic operations, and short conditional and unconditional branches.

Instruction execution within the XA has no fixed cycles; each instruction uses only as many CPU clocks to execute as are needed to perform the operation. The execution of instructions normally overlaps instruction fetch, and sometimes write-back operations to further speed processing.

## User-Configurable External Bus

The XA's standard external bus is designed to provide flexibility, simplicity of connection, and optimization for external code fetches. The external bus is user-configurable in several ways.

- The bus size may be configured to either 8 or 16 bits. This may be configured by the logic level on a pin at reset or under firmware control (if code is initially executed from on-chip code memory) prior to any actual external bus operations.
- The number of address lines may also be configured in order to make optimal use of I/O ports up to 24 address lines. Since external bus functions are typically shared with I/O ports and/or peripheral I/O functions, it is advantageous to set the number of address lines to only what is needed for a particular application, freeing I/O pins for other uses.
- The standard XA bus also provides a high degree of bus timing configurability. Timing is programmable in a range that will support most RAMs, ROMs, EPROMs, and peripheral devices over a wide range of oscillator frequencies without the need for additional external latches, buffers, or WAIT state generators.

## Power and Form Factor

The XA is based on low-power (3-volt) Combined Metal Oxide Silicon (CMOS). The designer can utilize both Power Down and Standby or Idle modes. Either mode is activated by setting a bit in the Power Control register. The Idle mode shuts down all processor functions except on-chip peripherals and external interrupts. An interrupt from any operating source will cause the XA to resume operations where it left off.

Power Down shuts down everything including the on-chip oscillator. Resuming operation from the power down mode requires the oscillator to be restarted - which takes about 10 milliseconds.

Features embedded in the architecture simplify and shorten design time by allowing a smaller form factor and reducing parts count. Designers have a choice of 40-pin DIP, 44-pin PLCC and 44-pin QFP packages.

## **Conclusion**

For new users, the XA meets all the criteria of today's 16-bit designs with multitasking, high level language support, low power consumption and an extensive and flexible register-based architecture. It is the only 16-bit microprocessor that is backward-compatible with the 8-bit 80C51. The XA provides an easy migration path to 16-bit power that satisfies the need for dramatically increased performance, multitasking and high level language support -- without forcing the designer to use new tools, techniques and software.

The XA offers 16-bit performance at 8-bit prices. The first derivative is targeted at less than \$10. The XA offers a performance ratio per dollar six to eight times higher than comparable microcontrollers positioned at the low end of the 16-bit market. When high-level programming languages are used with the XA, the yield will increase speeds up to ten times faster than the fastest 80C51 derivative.

**[Block Diagram of the 80C51XA Architecture]**