The unofficial history of 8051.

by Jan Waclawek (wek at efton.sk)

Preface

In 2005, the 8051 microcontroller celebrated it's 25th anniversary. I thought Intel will grab on the opportunity and perhaps add an item to their "museum" site, or remember in other way - but they did not. When I asked for some historical recollections on those days, I received a copy of a 8051 datasheet, dated 1980 and marked "preliminary". Well, I expected something else - but thanks for that, too.

But, also in 2005, Intel notified they discontinue all automotive versions of their microcontrollers, including 8051. Car engine control units were once perhaps the most prominent application for 8051s. This means only one thing, Intel gives up the microcontrollers for good. This is confirmed by product change notification published in early 2006, announcing that Intel drops its whole microcontroller business.

So, it is not too likely Intel will ever write up the history of 8051. It seems it is up to the volunteers, then.

I came to 8051's quite late, so I can't remember most of its history. I just collected a couple of facts I found interesting, fascinating. My thanks go to the 8052.com[1] discussion forum, where the idea of this document came from, and the forum users, contributing and correcting many of the details.

I invite any correction, being it spelling, wording or factual; and of course additions and ideas to this document.

Chapter -1

8048: Prehistory.

In fact, it should have started with chapter -2, the invention of microprocessor. Intel introduced a single-chip processor, the 4004, in 1971. It was a 4-bit microprocessor, with whopping processing speed of 100 thousand operations per second, and was meant for an electronic calculator. There is a lot of 4-bit processing in calculators, especially if the software is based on BCD arithmetics. Later Intel introduced the 8-bitter 8008 and it's grown-up brother - the famous 8080 (which then was perfected by an ex-Intel employee as Zilog Z80, one of the best 8-bit microprocessors of all times).

In 1976, Intel introduced its first microcontroller, 8048. It integrated the processing core with code and data memory and certain peripherals. The code memory was a 1kB mask ROM (defined by the last metallisation mask during the chip processing) or EPROM (after all, Intel invented EPROM), the data memory was 64 bytes of RAM (including the 8-level stack and two pages of eight general purpose registers). Besides general-purpose I/O (see below), peripherals included a timer and an external interrupt (plus the necessary interrupt system).

Although the 8048 is clearly an 8-bit architecture, it is said to be an ancestor of the 4-bit 4004 rather than the 8080. Also it is said to bear remarkable similarities to Fairchild F8 microprocessor. Today, it is hard to say whether something of this is true, but one thing is sure, the 8048 has a couple of strange features. Using four of its general purpose input/output ports, and adding one or more 8243-type chip - and the I/O expand into another four 4-bit ports. This expansion has not only support in the hardware - dedicated pins on 8048 - but also in the instruction set, having

[1] In early 2018, 8052.com has moved to www.8052mcu.com

dedicated instructions for I/O operations (including AND and OR(!)) via the expander.

The 8048 already had a lot of useful features known well to 8051-users: external code memory support; external data memory support (inherently only 256 bytes addressed indirectly by R0 and R1 as there is no 16 bit pointer register such as the DPTR in 8051 - the 8051 inherited this 8-bit external data access); quasibidirectional I/O ports. Maximum clock is 11MHz, but an instruction cycle takes 15 oscillator clocks. The "A" version (advanced) introduced powerdown mode

There were multiple variations of the 8048 around, mostly with different numbering, but generally denoted as the MCS-48 family. 8048 itself denoted a mask-ROM part, 8748 an EPROM part - windowed (CERDIP - erasable) for development, and unwindowed (PDIP) OTP. The romless part was a bit surprisingly marked 8035 (probably most of the parts sold as romless were parts with unusable ROM, due to error in the "programmed" firmware). There was a low-cost version with reduced pin count and omitted some of the features as 8021, and versions with more ROM and RAM as 8049 (2kB ROM/128B RAM) and 8050 (4kB ROM/256B RAM); with ROMless versions as 8039 and 8040; and 8049 had also an EPROM version 8749 (the funny thing is, that 8749 came in 1981, one year after 8051/8751). 8048's were second sourced by a number of manufacturers, including NEC, Toshiba, and were cloned also behind the then iron curtain in Czechoslovakia (Tesla MHB8048/8035) and USSR. Application specific versions of 8048 were also built quite early, with adding of various peripherals, such as 8-bit ADC in 8022 and a parallel-bus slave interface in 8041/8042.

The MCS-48 family was used in a quite wide range of applications. One of the first applications of 8048 was in a gaming console (Magnavox Odyssey2), but there were also more "serious" applications, for example in one of the first car engine "computerized" control units. But the biggest hit came when IBM decided to use 8048 in its original PC keyboard. Although in the AT keyboard IBM used the (presumably cheaper) 6805, it used 8042 as a co-processor on the mainboard, communicating with the keyboard (and performing a few other specific tasks in memory management). The 8042 is still present in almost each and every PC even today, but don't search for a chip with "8042" on it - it is integrated in the chipset. It may come as a surprise to somebody, but thanks to this fact the 8048 with its derivatives is most probably the most widespread microcontroller at all.

As in the 70s there were no pdf-s and no world-wide web, datasheets and other documentation is hardly available over the internet. I believe Intel will give out a copy if one really wants it (there is a "literature request" form at their "museum" pages). However, there seems to be a couple of enthusiastic people, one of the maintaining a wonderful document called “Grokking the MCS-48 System” at http://home.mnet-online.de/al/mcs-48/mcs-48.pdf .

Chapter 0

8051: The Classics.

In 1980, Intel introduced the successor to 8048, the 8051.

Intel made sure that the transition from the already successful model will be as smooth as possible. Architecturally, the 8051 is an extension to 8048. Almost every feature and resource of 8048 is present in 8051 in same or superior form. 4kB ROM and 128B RAM on chip. Pin compatibility was not maintained, but it was not a real issue. Software compatibility is not binary-wise but source-wise, but that is also acceptable. The preliminary datasheet read: "Enhanced MCS-48 Architecture".

The extensions included code and data memory extended to 64kB with appropriate support in instruction set and registers (DPTR), relative conditional and unconditional jumps (conditionals and DJNZ were constrained within a 256-byte page in 8048), four register banks instead of two, "unlimited" stack (8048 had stack limited to 16 bytes), multiple and divide instructions. As for peripherals, second timer was added and both were extended to 16 bits with multiple modes (including 8-bit autoreload mode), and an UART (which was a luxury that many lower-end microcontrollers didn't have even a couple of years ago). The raw clock frequency did not increase considerably, being 12MHz, but an instruction cycle is 12 clocks now.

Similarly to 8048, also the 8051 had variants, but there was no cut-down "low-cost" version (presumably because of the cost of ROM/RAM and the DIP40 package went low enough). The romless version was 8031 and the EPROM version was 8751. The "extended" version - 8052 (with 8032 and 8752) came 3 years later and featured besides 8kB ROM and 256B RAM also an extra 16-bit timer. An unusual chip was the 8052AH-BASIC, which according to Intel was "software-on-silicon version of the 8052 microcontroller with a BASIC interpreter on-chip in 8K ROM". The whole family was eventually called MCS-51 and was manufactured in NMOS, since 1986 in CMOS.

Intel provided all the needed initial tools and support with the 8051 - assembler, application notes, example software, in-circuit emulator. Some of the appnotes and software still can be found on Intel's webpages and are of excellent quality. The basic datasheet set - dubbed in the community as "the bible" - is still THE reference source of information on 8051 and its derivatives, even today.

So, Intel did its job, providing everything needed to make 8051 successful, and the rest is... history.

Chapter 1

The Birds Are Out Of The Nest.

Similar to 8048, also the 8051 has been licensed to various manufacturers worldwide. Some of the early adopters include Philips, Signetics, MHS (Matra) and Siemens. Most of these companies don't exist any more, some have been taken over, others have been renamed; but most of them still manufacture some derivative of 8051.

The licensees started to make fully compatible models. Naturally, they took over also the datasheets, for example the "bible" is better used in the Philips version, which is a verbatim copy of the Intel version, except that it is a true searchable pdf, while the Intel is a scanned copy of paper document, unsearchable. More than that, the manufacturers took over the annoying practice of Intel to include in datasheets only the specific differences to the "bible", very confusing for the newbies (but there are opinions on this, some of the users consider this arrangement better than having huge datasheets containing all the "common" details). The manufacturers published their own appnotes, which all together form a huge knowledge base and code library, but... due to competition it is scattered across the manufacturers' sites, an another confusing fact for the newbies.

Later, the manufacturers rolled out their own derivatives and variants with varying marking - there is no real standard in it (although there are some idiosyncrasies present in the marking of most manufacturers). All types of modifications described in the following chapters were applied; but the compatibility to the original 8051 was usually maintained. This, together with the availability of second-, third-,...,35th-,...-source of 8051 is the true source of its immortality.

Chapter 2

Embedded In Embedded.

Intel and the licensees soon realized that 8051 is a nice core that can be embedded in various ASIC chips to perform setup and control tasks. Typically, the resources of the ASIC are mapped as external data memory, as if the ASIC would be connected to a conventional 8051 chip. This approach allows to use an unmodified core, which speeds up the chip development and decreases the chance for error; also the ASIC could be breadboard-prototyped in this form easily.

As an example, Intel produced 80C51SL, a descendant of 8042. Philips has a line of 8051-based teletext controllers. In a particular USB webcamera, the chip interfacing the CCD and USB was controlled by an embedded 8051. There are probably much more examples around, but most of them never get public. In spite of this, the 8051 in this form is produced probably in much higher volumes than as general-purpose microcontrollers.

Chapter 3

Extras.

Besides application-specific, also general purpose derivatives have been introduced by Intel and the licensees, with enhanced features and increased code and data memories. In contrast with the ASICs mentioned above, these chips tend to implement the extra features in the core itself, accessed usually via extra SFRs. This allows faster code as SFRs are accessed by all the instructions using direct addressing (mov, logic), and some of them by the bit-manipulation instructions, too.

One of the first such derivative by Intel was the 80C51FA, which introduced the programmable counter array (PCA) (and was a 8052 otherwise). It was intended for automotive applications (brake control). Soon, FB and FC continued, with more and more code memory. 80C51RA/RB/RC followed, with added "internal external" data memory. These were the basis for the today's 89C51RD2 "sub-family", produced by Philips, Atmel (as ex-Temic), SST and Winbond.

Chapter 4

Fat Boys: 16-bit Extensions.

When the 8051 was accepted widely enough, some of the applications started to grow and soon required more power than the 8051 even with enhancements could provide. There were 16-bit microcontrollers around (e.g. Intel had it's 80C196 line), but it seemed a good idea to provide a more natural migration path by creating a 16-bit version of 8051.

Intel addressed the problem by introducing 80C251. It went all the way to achieve compatibility - it was able to run 8051 binary code (being able to switch to native 16-bit 251-mode) and had a package pin-compatible with 8051. It was not a big success, most probably for bad market timing (although it is second sourced by Temic/Atmel).

Philips on the other hand employed source-compatibility for its XA family, which seems to be adequate for most of the applications, where legacy code has to be maintained or parallel development with 8051 is needed; and poses little constraint on the chip design itself.

All in all, the 16-bit versions of 8051 gained far less popularity than the 8051 and are less widespread.

Chapter 5

Flash For The Masses.

In the 90s, Atmel introduced a derivative of 8051 with Flash code memory, enabling fast erasure and reprogramming. It enabled to use the production-grade chip in development, and enabled the chips used in the product to be reprogrammed when upgrade or a bugfix was needed, cutting down costs. It brought down the 8051 to the masses - the small "garage" companies and hobbyists. Besides that, Atmel introduced also 89C2051 with decreased pin count (and price). This was a smart move, the chip proved to be extremely popular in many small applications.

Today, virtually all manufacturers produce 8051 derivatives with Flash, most of them able to be programmed via some few-pin serial interface (called in-situ programming (ISP), SPI-style or UART-style) and the higher-end versions also able to reprogram themselves (in-application programming, IAP). MaskROM and EPROM - windowed or OTP - seems to become extinct, at least in the mainstream applications.

Chapter 6

Need For Speed.

The need for higher processing power, addressed unsuccessfully by the 16-bit versions, has been solved by introducing the high speed derivatives of 8051. The original 12-clock instruction cycle scheme is obviously inefficient and also the technology progressed enough to achieve higher clock rates than the original 12MHz.

The first derivative addressing this in a radical way is the now legendary Dallas DS80C320. It featured a 4-clocker core with incompatible timing, and could be clocked as high as 33MHz. Unfortunately, it was produced as ROMless only.

The following step was taken by Cygnal, where a single-clock core has been developed. In the top-range models, the clocking is as high as 100MHz, being the fastest 8051s around.

Today, there are many 8051 derivatives with sped-up cores available. They can be divided into two groups: the 6-clockers (e.g. the 8xC51RD2) and 2-clockers (Philips LPC9xx) have the same number of instruction cycle per instruction as the original; while the 4-clockers and single-clockers are incompatible in this way, requiring recalculation of timing loops if used.

Chapter 7

Where Is It Going?

The 8051 is a sound mcu core with rich history. However, it seems that it is already over its peak, although it might take quite a lot of time until it will be completely replaced by most modern microcontrollers.

So we now have superfast 8051 derivatives with loads of internal FLASH and RAM. ISP and IAP seems to be the standard these days. There are the 8051s built around advanced analog circuits, mainly high resolution ADC. There are derivatives suitable for extreme applications - high temperature, radiation hardened. There are softcores around, tuned up, and even open source.

There is a wealth of knowledge and experience, however, it is scattered around and the newbies tend to get the easier path - competing 8-bit microcontrollers usually do have a single-stop information resource site, so this knowledge and experience seems to die out as the "old boys" retire gradually. The price difference between the high-end 8-bitters and the much more powerful low-end 32-bit RISCs (such as the ARMs) seems to decrease rapidly and will change eventually, as the 32-bitters are becoming the standard in all but the least demanding applications.

So there is perhaps still a need for the 8051s, but this need is decreasing and 8051s life cycle is slowly approaching its end.

But I might be wrong, of course.