Appendix A
PIC Microcontroller Varieties

Overview

Introduction

In this section, we will examine the wide array of PIC microcontrollers, and try to give the hobbyist some help in selecting a microcontroller for future projects.

In this section

Following is a list of topics in this section:

PIC Families

Introduction

PIC microcontrollers come in a dizzying array of models. At first, the astonishing number of parts can seem quite confusing. Actually, there is a certain amount of rhyme and reason to these choices. In this section, we will look at the top division of PICs.

Cores

Microchip divides the PICs down into families they call "cores". These different families represent somewhat different architectures. The four divisions are:

- 12 bit core
- 14 bit core
- 16 bit core
- Enhanced 16 bit core

These four families appeared chronologically as they appear in the list.

Perhaps a bit confusing, the number of bits doesn't refer to the data bus width, as it does on popular microcomputers. The data bus width is always 8 bits, meaning that a data cell can hold one of 256 different values. The "core size" refers to the width of the program memory. Each instruction can be 12, 14 or 16 bits long. More bits in the program memory means that more memory can be addressed, or more instructions can be provided. As we discussed in Lesson 1, there is a tradeoff between the number of instructions and the memory size.

The first few digits of the part number identify the core, but in a particularly confusing way:

- 12 bit core – PIC12x
- 14 bit core – PIC16x
- 16 bit core – PIC17x
- Enhanced 16 bit core – PIC18x

In all the families, the 'x' represents the memory technology for the program memory.

To further confuse the issue, there is a PIC14000, which is a 14 bit core device.

Continued on next page

PIC Families, Continued

12 bit core devices	The 12 bit core parts were the first PIC microcontrollers to be made available, and are probably the most popular. The 12 bit core can address 1K of program memory. Parts are available with up to 128 bytes of RAM. Most of the PIC12 parts are in 8 pin packages. The architecture supports 35 instructions. While the 8 pin package and small memory may seem like significant limitations, for most production uses, the PIC12 is very adequate, and its low price makes it a popular choice. Microchip shows "budgetary"[1] pricing for the PIC12 parts ranging from under a dollar to slightly over two dollars. Most of the 12 bit parts have historically been either one time programmable or UV erasable, rather than FLASH. This, along with the low I/O count, has made this series somewhat less attractive to hobbyists.
14 bit core devices	The 14 bit core parts are the most popular among hobbyists. They are also widely used in industrial applications, although they are not as pervasive as the 12 bit parts. The 14 bit core can address 8K of program memory, and are available with up to 368 bytes of RAM. The instruction set is the same as the 12 bit parts, except that some of the instructions can address more locations. The PIC16 parts are available in 14, 18, 28 and 44 pin DIP packages, as well as a variety of surface mount packages. The more pins the package has, the more pins are available for I/O. Budgetary prices range from under two dollars to under seven dollars. Quantity one pricing for almost all these parts is under ten dollars, so the price, coupled with the large I/O compliment, has made this the most popular series for hobbyists.
16 bit core devices	The 16 bit core has to be a mistake on Microchip's part. The part was introduced as yet another enhancement to the PIC line, but at the time compilers were becoming popular, and the PIC17 parts did not provide much support for compilers. The 16 bit core can address 32 K of program memory and are available with up to 902 bytes of RAM. The instruction set adds a few instructions to the instruction set shared by the 12 and 14 bit core devices. The PIC17 parts are available in 40 pin DIP packages and a few surface mount packages. The larger parts are only available in surface mount. Budgetary prices range from over six dollars to just over twelve dollars. There are only 8 parts in the PIC17 family (excluding package differences).

Continued on next page

[1] On large quantity buys, pricing is almost always negotiated. Microchip lists budgetary pricing to give designers a feeling for how these negotiations might turn out. Quantity one prices are significantly higher. For example, Microchip lists the budgetary price for the 16F84A at $3.42. This turns out to be the lowest priced package. For a plastic DIP package, the budgetary price for the 16F84A-20/P is listed at $3.65. DigiKey supplies that part in quantity one for $6.00. Similarly, the budgetary price for the 16F628-20/P is $2.04; DigiKey's quantity one price is $3.88.

PIC Families, Continued

Enhanced 16 bit core devices

The earlier families of PICs did not support compilers for modern languages very well. While there are a large number of so-called "C" compilers available for the 16x and even 12x parts, these are fairly crude compilers. The language that is implemented may look a little like C, but the compilers are unable to support even the most common C language idioms.

Part of the reason has to do with the small memory size of the PIC, but the major reason is the extreme scarcity of a stack on the PIC. Modern nested-scope languages like C require a stack to be properly implemented. Indeed, this is a major characteristic which sets them apart from earlier languages like FORTRAN and COBOL.

The 12 and 14 bit PICs have an 8 level stack. On the 17x parts, this was increased to 16. By comparison, some early PC C compilers were limited to a 2K stack, *which was a significant limitation*. In addition to having a limited stack, the PIC instruction set has no instructions for manipulating the stack, so the compiler designer has to implement the stack in software. With the PIC's small RAM size, this is a challenge and a major impact on performance as well.

The PIC18x parts implement a dramatically different architecture, intended to provide better support for modern compilers. PIC18x parts are available to support up to 64K of program memory and have RAM sizes up to 3K. The instruction set is much broader than the earlier parts, consisting of 75 instructions, but includes the PIC17x instruction set.

The PIC18x parts come in 28 and 40 pin DIP packages, as well as a variety of surface mount packages. Like the 17x parts, the larger models are only available in 64 and 80 pin TQFP packages. Budgetary pricing ranges from under four dollars to about 11 dollars. Quantity one pricing for the PIC18F458, one of the larger models available in a DIP package, is just over ten dollars.

The availability of meaningful compilers, along with the greatly improved instruction set and reasonable price, has contributed to a rapidly increasing popularity of the PIC18x parts.

Program Memory

Introduction	The program memory for the PIC is implemented in one of five different technologies. These technologies are differentiated by only three different letters in the model number, C, E or F.
OTP	The largest number of PIC models have the program memory implemented as CMOS ROM which can be programmed only once. This **O**ne **T**ime **P**rogramming feature makes these models unappealing for hobbyists, but the low cost is attractive when a number of the same device will be produced. Most of the popular keyer chips, and the controllers for radios, when they are PICs (as in the Elecraft radios), are implemented as these one time programmable parts. The one time programmable models are identified with the letter **C** following the family number.
Erasable PROM	Development is difficult when every test of a new program requires a new chip, so to support development, Microchip introduced parts which include a UV Erasable PROM. There are relatively few of these parts, most of them in the 12 bit family. Initially, Microchip had a UV Erasable part for each configuration of a one time programmable part, but recently, these have been replaced by FLASH parts. The erasable PROM models are identified with the letter **C** following the family number. A few parts are electrically erasable and marked with an **E**.
FLASH	FLASH is a nonvolatile semiconductor memory that can be read and erased electrically. Most of the recent PIC models have included FLASH for the program memory. The FLASH parts include an **F** in their part number (like the PIC16F84). An advantage of FLASH over EEPROM is that with FLASH, the designer can trade off read time with write time. In the case of a PIC, the read time can be a limitation on the processor's clock speed, so it is an advantage to accept a slower write time to get a faster read time. As PICs get larger, though, this slower write time becomes visible when programming the PIC. Although not a major issue, programming times for 4K and 8K parts can get annoyingly long. To speed up programming, some newer PICs have been architected so that four words can be written to program memory at once. This requires a difference in the programming instructions used to program the part. As a result, the software used to drive the programmer must change. The FPP program we are using will program the older types of FLASH, which still constitute the majority of parts. Other software must be used to program the newer type of program memory.

PIC Part Numbers

Introduction

The PIC part number contains a lot of information. In this section, we will look at what the number is telling us.

Part number format

The PIC model number might look something like:

PIC16F84A-20E/P

The 2 digits after the word PIC represent the family, mentioned earlier. The letter represents the program memory technology. Sometimes the F is preceded by an L indicating a low power part. The digits following the F or C represent the model number. This may be followed by a letter which represents some sort of variant. Often, an A variant means the part uses the newer type of FLASH, but that is not always the case. In the case of the 16F84A, it is a much more subtle difference. After the dash comes a maximum clock speed, which may be lacking. Then comes on optional letter, which indicates the temperature specification. Finally, there is a slash followed by the package type.

Processor Speed

All PIC processors are capable of supporting clock speeds down to DC. The part number may indicate a maximum. Typical values are 4, 10 and 20. On newer parts, a missing clock speed almost always means 20 MHz.

The letter L used to mean that the part was a 100 kHz maximum speed part, which draws considerably less current than a 20 MHz part. All PICs draw less current as their clocks are slowed, however, and newer models draw no more current at 100 kHz than the older, “low power” models. On those newer models that are available in an “L” version, the L implies that the sleep current is slightly less. This is typically not much of an issue for amateur applications because the sleep current is extremely low in any case – typically a few microamps.

Temperature Specification

The default PIC operating range is 0 to 70 degrees C, which Microchip calls “Commercial”. An **I** before the slash indicates an “Industrial” part, with a temperature range of -40 to 85 degrees C. An **E** indicates an “Automotive” part with a range of -40 to 125 degrees C.

Packages

PICs come in a variety of packages, indicated by the suffix. Packages which may be of interest to hobbyists are:

Suffix	Package
P	Plastic DIP
SP	Plastic skinny DIP
SO	Small Outline SOIC
SS	Shrink Small Outline SSOP

PIC Model Differences

Introduction

That 2 or 3 digit number after the C or F can mean a lot. Unfortunately, the number itself cannot be easily dissected. Here we will talk about the specification differences that give rise to all those numbers.

Memory Sizes

The most obvious difference between the various PICs is the memory size. There are three memories to consider; Program memory, File Register (RAM), and EEPROM. Here are the values for a few of the more popular PICs:

Part	Program	RAM	EEPROM
PIC16F83	512	36	64
PIC16F84	1K	68	64
PIC16F88	4K	368	256
PIC16F628	2K	224	128
PIC16F877	8K	368	256

I/O Compliment

The various PIC models have an amazing variety of I/O. All PICs have a number of bidirectional I/O pins. In general, the number of digital I/Os is determined by the number of pins in the package. For almost all PICs, any pin that's not needed for a major function (like ground) is available as an I/O. On some newer PICs, even the connections for the crystal and reset line can be retargeted as I/O (with obvious implications elsewhere!)

Many PICs also incorporate analog inputs which allow the PIC to sense a voltage. Many include comparators. Some include special ports which can automatically generate Pulse Width Modulation output. Some even have UART or USART ports. All these possible combinations lead to a huge number of possible devices.

The following table shows some of the specialized I/O for the parts shown above:

Part	Analog	Comparator	PWM	Timer
PIC16F83	0	0	0	1
PIC16F84	0	0	0	1
PIC16F88	7	2	1	3
PIC16F628	0	2	1	3
PIC16F877	8	0	2	3

Hobbyist Parts

Introduction	Hobbyists, of course, may use any of the various PIC models. However, there are a few parts which are commonly used, and the avid PIC builder is likely to stock only a small number of different models. Here we make some suggestions for parts to include in the junk box.
16F84A	This part has, by far, the most projects available on the web. It has been the hobbyist workhorse for years, and is a very solid part. However, this writer would not recommend using this part for new projects. The 16F628 is a drop-in replacement electrically, has more capability, and costs half as much.
16F628	This is the ideal hobbyist workhorse. It comes in an 18 pin package, not so large as the ‘877, and has a huge complement of capabilities. The 628 has twice the program memory of the 84, twice the EEPROM, three times the RAM, and adds comparators, a PWM module, two more timers, and a USART. Unfortunately, the part is not a software drop in for the ’84. As a minimum, small changes need to be made to the program in order to replace an ’84 in an “off the web” project.
16F877	If you need analog I/O, this is the part. Unfortunately, the 40 pin package is large, so it might not be the best choice for that backpack rig. The ‘877 has a lot of memory, and a wide complement of additional I/O. All that I/O demands pins to connect to, though, and thus the large package. This part might just make you take up surface mount to get more pins in a smaller space! The ‘877A adds a couple of comparators to the 877’s I/O complement, but the ‘877A uses the new FLASH technology, and thus might not work with your current programming software.
16F88	The 16F88 is a relatively new part that looks to be quite promising. It offers analog I/O in an 18 pin package, one of the few FLASH parts to do so. And it offers 4K of program memory.

16F84 to 16F628 Conversion

Introduction	The 16F628 is fast becoming the most popular PIC for hobbyists. But most of the programs out there are for the 16F84. In this section, we will look at the issues around converting code from the 'F84 to the 'F628.
File Register	The first thing one notices about the 628 is that there is a lot more of all kinds of memory. However, there are also a lot more registers. These registers force the file register RAM to start at a higher address. RAM in the 16F84 starts at address H'0c', in the 628, at H'20'. This means that programs need to have the locations they use for RAM converted. On a few programs this is simply a matter of changing the address in a `cblock` statement. More often, though, older programs use a string of `equ` statements to define their memory use. To do the conversion, you need to understand how the memory was used, and the move it up above H'20'.
CMCON	In the 16F84, all of the I/O pins are set to be I/O on power up. The only pin that can do anything else is RA4, and it needs to be programmed to be different. In the 16F628, RA0, 1, 2, and 3 power up as part of comparators; the normal I/O behavior doesn't work. In order to make these pins behave like a 16F84, the CMCON register needs to be initialized. Specifically, the low order 3 bits of CMCON need to be set.
File Register Banks	This is not *normally* a problem, but when it is, it can be messy. Most of the special purpose registers in the 16F84A are mapped into both banks. On the 628, this is not the case. When a program plays with the bank bits in the 16F84, one needs to examine the code carefully to be sure that the program isn't writing to a register that won't be there on the 628. This takes understanding what the program is doing, and having both datasheets right by your side!
Register Locations	Unfortunately, many programs you find on the web use their own constants for special purpose registers, or worse yet, access these registers by their addresses directly. Not all the registers in the '628 are at the same addresses as those in the '84. While this isn't conceptually difficult to fix, it can be very difficult to locate the places that need to be changed. The registers for addressing the EEPROM have even changed banks!
EEIE/EEIF	The EEIE bit has moved from the INTCON register to the PIE1 register. EEIF has moved from EECON1 to PIR1. These bits are used when writing to the EEPROM.
Obvious changes	Of course, you need to change the processor directive and the name of the include file.

Why did we choose the 16F84

Introduction	In a previous section, we mentioned that we would not recommend the PIC16F84 for new projects, yet we chose it for the PIC-EL. We have been asked this question before we started the course, and it continues to come up. Most often it is asked in the form of "Why didn't you use the 16F628?" Here is some of the rationale.
Simplicity	The major reason is simplicity. Some have argued that the 628 is no more difficult to use than the 84. This is close to true, if you are already familiar with the PIC. For many of the folks taking the course, this is their first foray into microcontrollers. Certainly, after the course, they should be able to apply a 628 as easily as an 84, but starting out, the 628 is pretty intimidating.
Multi-use pins	On the PIC16F84, 12 of the 13 I/O pins are nothing more than I/O pins. The user just learning the PIC can use most of the capability while more or less ignoring the one oddball pin, expecting to learn about it later. Even so, if the new programmer treats that 13th pin just like the other 12, it will behave as expected. The user only needs to learn more when he is ready to do more. On the PIC16F628, every pin has three or four uses. The new developer is faced with an alphabet soup around every pin of confusing acronyms. Not only that, but several of the I/O pins require circuitry changes to make them useable. These changes require giving up something, and the designer has to consider these tradeoffs. To further add to the confusion, some of the "old style" I/O pins are no longer plain I/O on power up. Special initialization needs to be performed to make them accessible.
Memory Banks	On the PIC16F84A, the only registers that require bank switching to access are the TRIS registers and some registers dealing with EEPROM. Since most programs don't use EEPROM, the only time a developer needs to consider memory banking is when he is setting the direction of the I/O lines. This is conceptually relatively simple … switch banks to access the TRIS registers, do your business, switch back. Nothing mysterious happens. On the 628, however, most of the user's data space is also banked. The switch to bank 1 causes most of his program variables to disappear. Now he needs to carefully consider where he places the variables so that they are available in bank 1. Not hard, but another source of confusion.
Summary	There is little doubt that the PIC16F628 is a more capable chip at a much lower price. However, that capability comes at a cost in complexity that can be intimidating for someone who is just learning. Once one is familiar with the PIC, using any new feature is simply a matter of looking it up in the datasheet. From that perspective, one part is no more complex than another. But for someone who is just learning, the PIC16F84 is simple, with far less complexity than the newer parts.