

ASS: The Arbadell Assembler

By Shane Avery

This is a document describing how to use the Arbadell assembler. The source code is free and open to anyone who wishes to use it. It has been compiled on a COMPAQ DESKPRO PC with an Intel PIII running at 500MHz with 320 Megs of RAM with gcc (Linux) and TurboC (Windows). Also, the source has been compiled and runs on my iBook running OS10.2. The assembler is very simple and supports the assembly instructions of the Arbadell CPU processor. The CPU document contains a more detailed list of assembly instructions and their functionality. Briefly, here is a list of the instructions.

movel	B	move the literal value B to the scratch register
movem	BB	move the value in memory BB to the sregister
moves	BB	move the s register to the memory location BB
inv		invert scratch register
inc		increment scratch register
dec		decrement scratch register
shiftr		shift scratch register to the right one bit
shiffl		shift scratch register to the left one bit
addl	B	add the literal value B to the scratch register
andl	B	and the literal value B to the scratch register
orl	B	or the literal value B to the scratch register
addm	BB	add the value in memory BB to the s register
andm	BB	and the value in memory BB to the s register
orm	BB	or the value in memory BB to the s register
call	BB	call the subroutine located in memory location BB
bra	BB	branch unconditionally to memory location BB
braz	BB	branch if zero to memory location at BB
ret		return from subroutine call
reti		return from interrupt
iorecv	B	receive data from chipset
iosend	B	send data to chipset
setie	B	set interrupt enables
ti0	BB	test interrupt 0. branch to BB if interrupt is set
ti1	BB	test interrupt 1. branch to BB if interrupt is set

The assembler takes in one to four arguments. The first argument is the file to be assembled. The second argument is the name of the file that the assembled source file is stored. If no second file is specified then the default output file is arb.txt. A -v argument is for verbose and will output the progress of assembler as it goes through the file. The assembler will make two temporary files during assembly named arb.map and arb.tmp and they will be removed before the assembler exits. For debugging purposes a -d argument will keep these files from being deleted. Comments are indicated with a semi-colon. Any new lines will be ignored so please feel free to separate code.

Labels are also supported. Labels cannot be a reserved word(instruction),

cannot begin with 0x, and cannot begin with a semi-colon. Any argument containing a label will be replaced by the address of its label.

There are three supported assembler directives. ORG tells the assembler where you wish your code to be assembled to in memory. EQU (equate) is a means of declaring constants throughout the code. DC (declare character) will tell the assembler to put the byte that follows into memory at this location.

All arguments in the assembler are in hex. All hex arguments must lead with 0x. The assembler doesn't care if you use capitals or not. Be sure there is an end statement at the end of your source code. You can add it with or without a tab.