University of Toronto, Faculty of Applied Science and Engineering
Department of Electrical and Computer Engineering

ECE 243S – Computer Organization – 2012

Microprocessor Assembler

An assembler is provided with the processor to aid students with the process of creating simple programs. It generates a Memory Initialization File (MIF) that can be used to initialize the memory of the processor design. The assembler is invoked by running *assembler.exe.*

The assembler will ask for one mandatory argument: the name of the file containing the assembly program. The file can have any file extension, but we suggest using ".s". If compilation was done successfully, the assembler will produce the "mem.mif" file.

The assembler is based on the concept of columns, similar to many other assemblers. The first column begins at the first character of a line, the second column is separated from the first by any amount of whitespace, and the third column is separated from the second in the same way. If more columns are present, the assembler will issue an error, unless the text immediately after the third column is a comment.

Comments are indicated using a semicolon. The assembler will ignore anything following a semicolon, unless the semicolon interrupts an instruction or directive, in which case an error will be issued.

The first column must either be blank, or contain a unique label. Please note that this assembler only allows labels on the same line as the instruction/directive to which they refer. A label cannot appear on a line by itself. The second column must contain a valid keyword identifying the instruction or directive. Finally, the third column must contain the parameters/operands. Please note that in this assembler no spaces are allowed in the third column. If two operands are required, they are separated using a comma without spaces in between.

Numerical constants can be specified in binary, octal, decimal, or hexadecimal format. Binary values are preceded with a '%' sign, octal values are preceded by an initial zero (0), and hexadecimal values are preceded with a '$' sign. If none of these characters precedes the number, it is assumed to be in decimal format. Please note that negative sign ('-') is only supported when using the decimal format.

Overall, 14 keywords are supported:

- add r1,r2
- sub r1,r2
- nand r1,r2
- shift r1,imm3
- shiftl r1,imm2
- shiftr r1,imm2
- ori imm5
- load r1,(r2)
- store r1,(r2)
- bpz ADDR
- bz ADDR
- bnz ADDR
- org imm8
- db imm8

The shift instruction, as described in the course notes, accepts a 3-bit value. Bit #2 determines the direction (0 – right, 1 – left), and bits #1 and #0 determine the number of shift positions. For convenience, “shiftl” and “shiftr” are provided to shift left and right, respectively. They accept a number between 0 and 3, specifying the number of shift positions. They are not true instructions, because they both get converted to an equivalent “shift” instruction.

Please note that imm2, imm3, and imm5 must be unsigned positive numbers (i.e. negative signs are not allowed). The brackets around “r2” for “load” and “store” are also required. ADDR can be either a previously defined label, or a 4-bit signed number. Please note that the effective branch address is calculated as follows: current_address + imm4 + 1. Do not forget about the 1 that is added, especially when using backward branching.

“org” can be used to set the current address. It is specified here to use imm8, but the actual width of the number depends on the memory size. For this lab, a number between 0 and 255 can be used.

“db” is used to place a single byte of at the current address. The byte can be either an 8-bit number, or a character in single quotes. Only one byte can be specified per db” directive. While “org” and “db” are not instructions, they can be associated with a label, identifying the new address that they define (in the case of “org”), or the address at which they place their data (in the case of “db”). Please note that a label associated with an “org” statemen t and with an instruction or directive immediately following it identify the same address, unless another“ org” statement is the following directive.