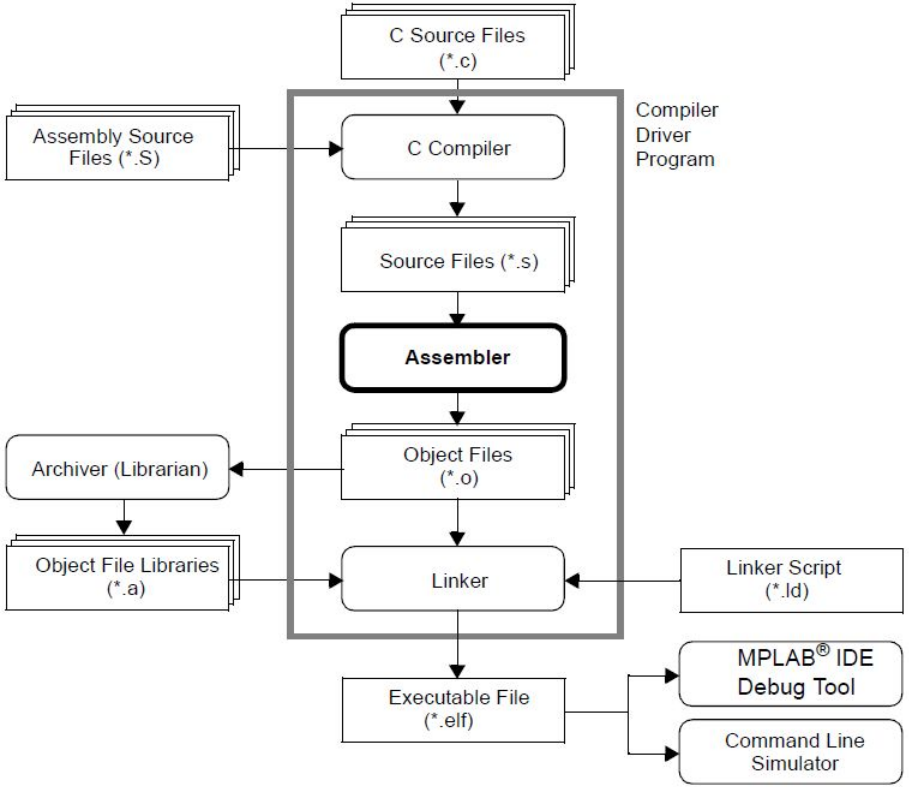
Lecture 3. Assembly

Syntax and selected directives

Yuri Panchul, 2014

Microchip MPLAB X tools flow



From Microchip Technology MPLAB-ASMLINK32-User-Guide.pdf

Symbols, numbers and expressions

- Symbols use letters, digits, underscore ‘_’ and period ‘.’
- Symbols may not begin with a digit
- Numbers - like in C, but also has binary numbers - 0b01010101
- Expressions - like in C
- Special symbol “.” for program counter

Local labels for branches

- Special case - local labels “0:”, “1:”, … “9:”
- Used in branch instructions with suffix “f” (“forward”) and “b” (“backward”)

```
1: b  1f; nop
2: b  1b; nop
1: b  2b; nop
```

Directives: Sections

- .text - program code section
- .data - initialized data section
- .rodata - initialized read-only data section
  - Used to place C const variables: const int a = 3;
- .bss - uninitialized data section
  - Initialized with zeroes by boot code
- .sdata - “small data” for use with gp register
- .sbss - also for use with gp register

Directives: Initialization

- Characters: .ascii *“string$_1$” [, …, “string$_n$”]*
- Zero-terminated: .asciz *“string$_1$” [, …,“string$_n$”]*
- Bytes: .byte *expr$_1$ [, …,expr$_n$]*
- 2-byte halfword: .hword *expr$_1$ [, …,expr$_n$]*
- 4-byte word: .word *expr$_1$ [, …,expr$_n$]*
- 8-byte doubleword: .dword *expr$_1$ [, …,expr$_n$]*
- double float: .double *value$_1$ [, …,value$_n$]*

Other data-related directives

- .global *symbol*
- .extern *symbol [, size]*
- .align *[align [, fill]]*
- .space *size [, fill]*

Directives to repeat code sequences

- .rept *count* … .endr
- .irp *symbol value*$_1$ *[, …,value*$_n$*]* … .endr
- .irpc *symbol value* … .endr

```
.irp reg, 0, 1, 2, 3
lw $\reg, 1024 + \reg * 4 (sp)
.endr
```

Controlling code generation

- .set noat - assembler must not use *at* ($1) register
- .set noreorder - assembler must not move instructions inside branch delay slots
- .set nomacro - generate warnings for so-called synthesized instructions that are expanded into multiple machine instructions

Compile-time error directive

- .err
- .error *“string”*
- They are useful for conditional compilation when preprocessor is used

Special: “Small” memory support

- gp - “global pointer”, register $28
- gp-relative addressing
  - A convention to quickly access “global” 64K memory
  - In addition to accessing memory in “normal” way
  - Saves 1 instruction to load upper part of the address
- Assembly support
  - .sdata section for grouping “small” memory variables
  - %gp_rel macro: lw t0, %gp_rel (my_variable) (gp)
  - .extern should be with size: .extern my_variable, 4

Thank you!