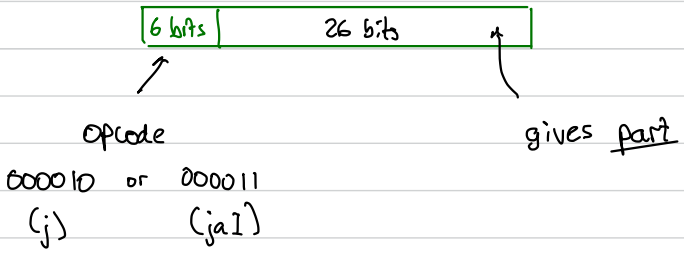
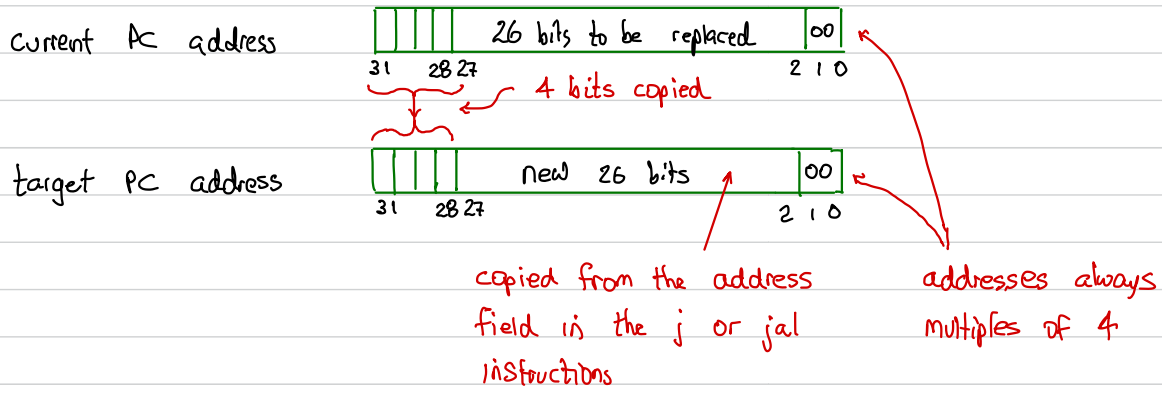
j and jal encoding

j, jal :



j and jal update the program counter (PC) in the following way



Compiling and Building C programs

We now briefly discuss programming environments on "real" systems, not simulated ones like MARS.

e.g., Linux / Windows / Mac

The compiler

Here, we interpret "compiler" to mean:

- A program to translate a high-level language to assembly.

Running gcc on Cygwin 64 - the gcc toolchain

Example: foo.c, bar.c : gcc foo.c bar.c

Lots happen: 1. Preprocessor creates translation unit from foo.c and its header files.

2. Compiler program converts translation unit to assembly language.
3. Assembler converts assembly language file to binary object file

4-6. Repeat steps 1-3 for bar.c

7. Linker creates executable file a.exe by combining the two object files and system library code.

Many kinds of files involved

C source file: programmer's .c and .h files

Library header files: .h files to define library types and functions (e.g., stdio.h, math.h, etc.)

Translation unit: .i files produced by the preprocessor. Preprocessors handle all preprocessor directives (e.g., #include, #define, #ifdef, etc.)

Assembly-language file: Output from the compiler, equivalent to .asm files (.text, .data, instructions, etc.)

- Note the above files are all text files
- All others are binary (e.g., object files, machine-code libraries, executables).

Typical object file layout (output of assembler)

object file header	← gives sizes, and layout of object file
text segment	← binary from instructions in text segment
data segment	← code from .data segments (.word, .byte, ...)
relocation information	← (explained shortly)
symbol table	← list of global symbols (.globl) with their relative address information

Example symbol table

Suppose a procedure defines global symbols:

procedures: main, foo, bar
static data: xx, yy

Typical symbol table layout:

symbol	segment type	byte offset from start of segment
main	text	0
foo	text	96
bar	text	224
xx	data	0
yy	data	4

The executable file

Created by the linker program. Combines

- one or more object files
- system library procedures