IFT 307

Computer Organization and Architecture

Mr. Ibrahim Lawal

ADDRESSING MODES

- Information involved in any operation performed by the CPU needs to be addressed.
- In computer terminology, such information is called the operand.
- Therefore, any instruction issued by the processor must carry at least two types of information.

These are the operation to be performed, encoded in what is called the:

- ✔ Op-code field, and the
- ✔ Address information of the operand on which the operation is to be performed, encoded in what is called the address field.

.

Instructions can be classified based on the number of operands as:

- ✔ three-address,
- ✔ two-address,
- ✔ one-and-half-address,
- ✔ one-address, and
- ✔ zero-address

- We would use operation, source, destination to express any instruction.

- **Operation** represents the operation to be performed, for example, add, subtract, write, or read.

- **The Source Field** represents the source operand(s). The source operand can be a constant, a value stored in a register, or a value stored in the memory.
- **The Destination Field** represents the place where the result of the operation is to be stored, for example, a register or a memory location.

TABLE 2.1 Instruction Classification

Instruction class	Example
Three-address	*ADD* R_1,R_2,R_3
	ADD A,B,C
Two-address	*ADD* R_1,R_2
	ADD A,B
One-and-half-address	*ADD B*,R_1
One-address	*ADD* R_1
Zero-address	*ADD* (*SP*)+, (*SP*)

Immediate Mode

According to this addressing mode, the value of the operand is (immediately) available in the instruction itself.

Consider, for example, the case of loading the decimal value 1000 into a register Ri.

- This operation can be performed using an instruction such as the following:
  LOAD #1000, Ri.
- In this instruction, the operation to be performed is to load a value into a register.
- The source operand is (immediately) given as 1000, and the destination is the register Ri.

- It should be noted that in order to indicate that the value 1000 mentioned in the instruction is the operand itself and not its address (immediate mode),
- it is customary to prefix the operand by the special character (#). As can be seen the use of the immediate addressing mode is simple.

- The use of immediate addressing leads to poor programming practice.
- This is because a change in the value of an operand requires a change in every instruction that uses the immediate value of such an operand.
- A more flexible addressing mode is explained below.

Direct (Absolute) Mode

According to this addressing mode, the address of the memory location that holds the operand is included in the instruction.

- Consider, for example, the case of loading the value of the operand stored in memory location 1000 into register Ri.
- This operation can be performed using an instruction such as
  LOAD 1000, Ri.

In this instruction, the source operand is the value stored in the memory location whose address is 1000, and the destination is the register Ri.

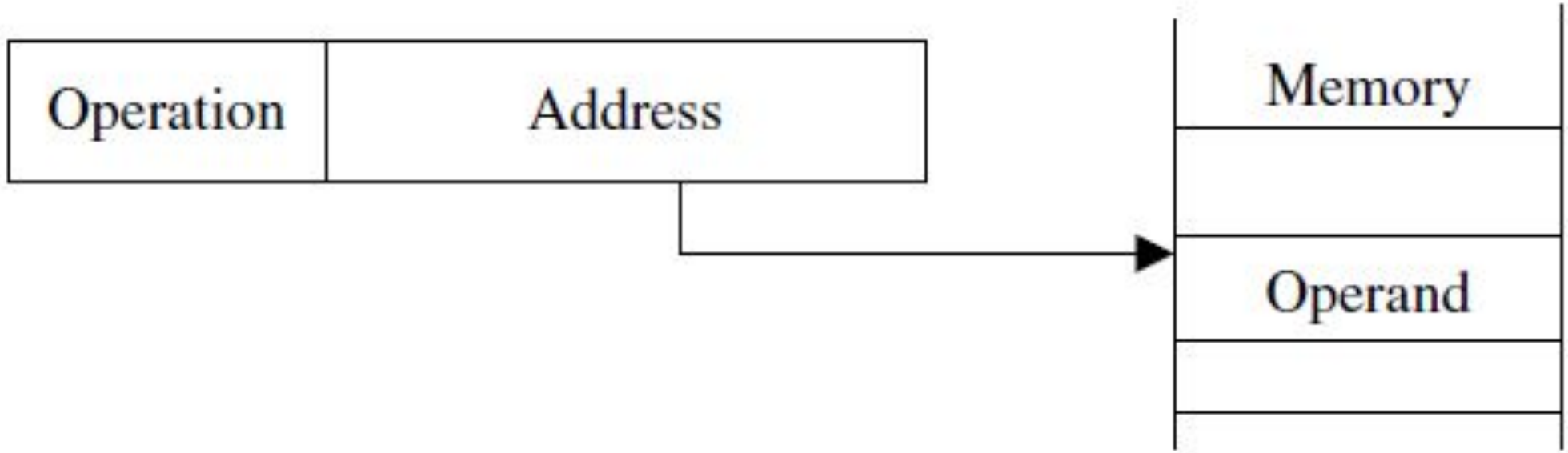


Figure 2.6 Illustration of the direct addressing mode

- Note that the value 1000 is not prefixed with any special characters, indicating that it is the (direct or absolute) address of the source operand.
- Figure 2.6 shows an illustration of the direct addressing mode.

For example, if the content of the memory location whose address is 1000 was (2345) at the time when the instruction LOAD 1000, Ri is executed, then the result of executing such instruction is to load the value (2345) into register Ri.

- Direct (absolute) addressing mode provides more flexibility compared to the immediate mode.
- However, it requires the explicit inclusion of the operand address in the instruction.
- A more flexible addressing mechanism is provided through the use of the indirect addressing mode. This is explained below.

Indirect Mode

In the indirect mode, what is included in the instruction is not the address of the operand, but rather a name of a register or a memory location that holds the (effective) address of the operand.

In order to indicate the use of indirection in the instruction, it is customary to include the name of the register or the memory location in parentheses.

- Consider, for example, the instruction
  LOAD (1000), Ri.

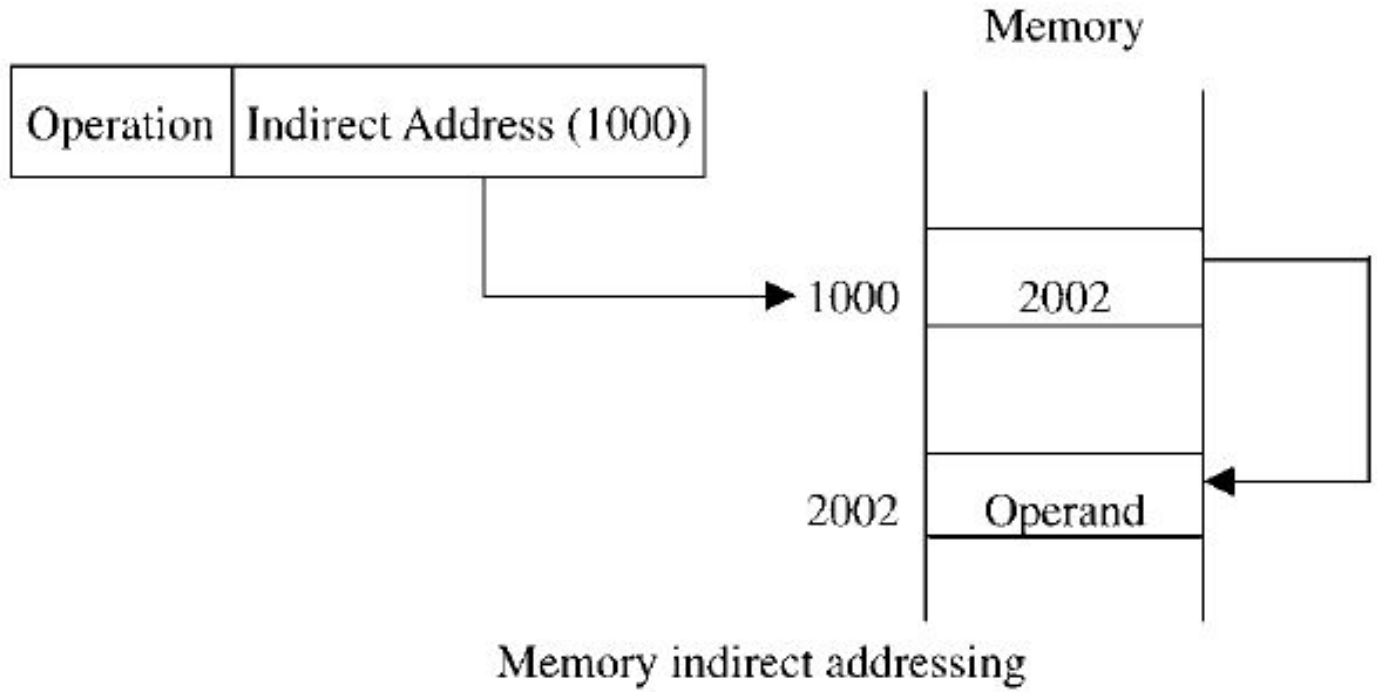
- This instruction has the memory location 1000 enclosed in parentheses, thus indicating indirection.

- The meaning of this instruction is to load register Ri with the contents of the memory location whose address is stored at memory address 1000.
- Because indirection can be made through either a register or a memory location,

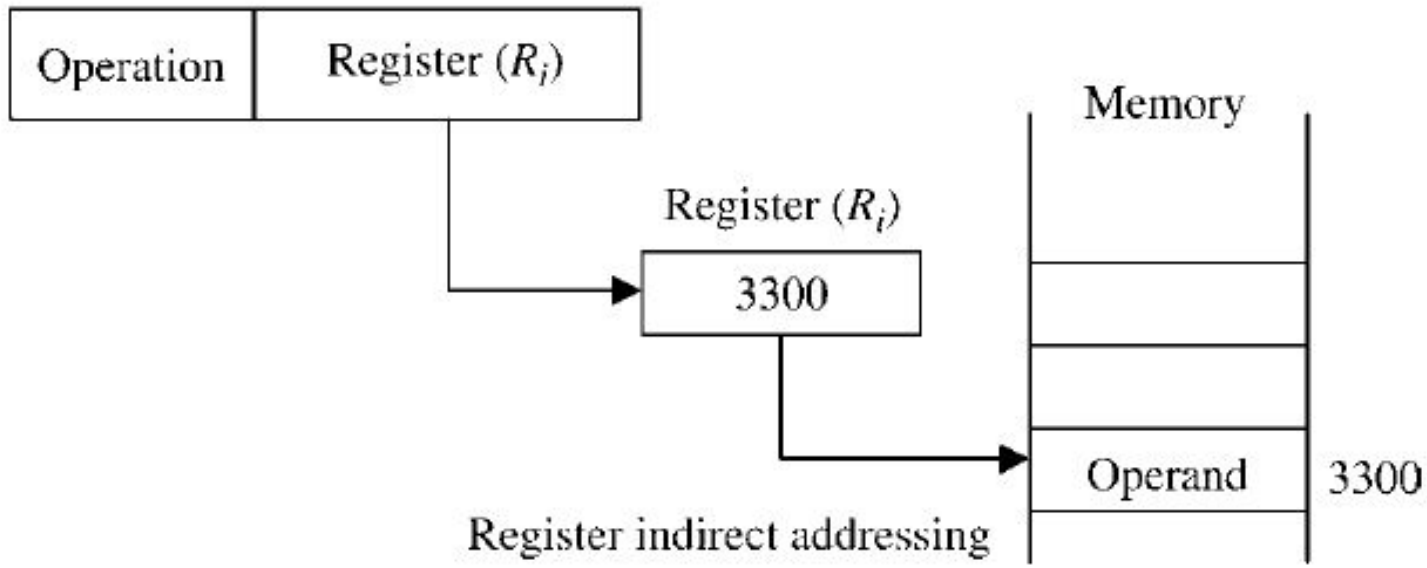
we can identify two types of indirect addressing. These are:

- register indirect addressing, if a register is used to hold the address of the operand,
- memory indirect addressing, if a memory location is used to hold the address of the operand.

The two types are illustrated in Figure 2.7.



Memory indirect addressing



Register indirect addressing

Indexed Mode

- In this addressing mode, the address of the operand is obtained by adding a constant to the content of a register, called the index register.
- Consider, for example, the instruction

  LOAD X(Rind), Ri.

This instruction loads register Ri with the contents of the memory location whose address is the sum of the contents of register Rind and the value X.

- This instruction loads register Ri with the contents of the memory location whose address is the sum of the contents of register Rind and the value X.
- Figure 2.8 illustrates the index addressing mode.

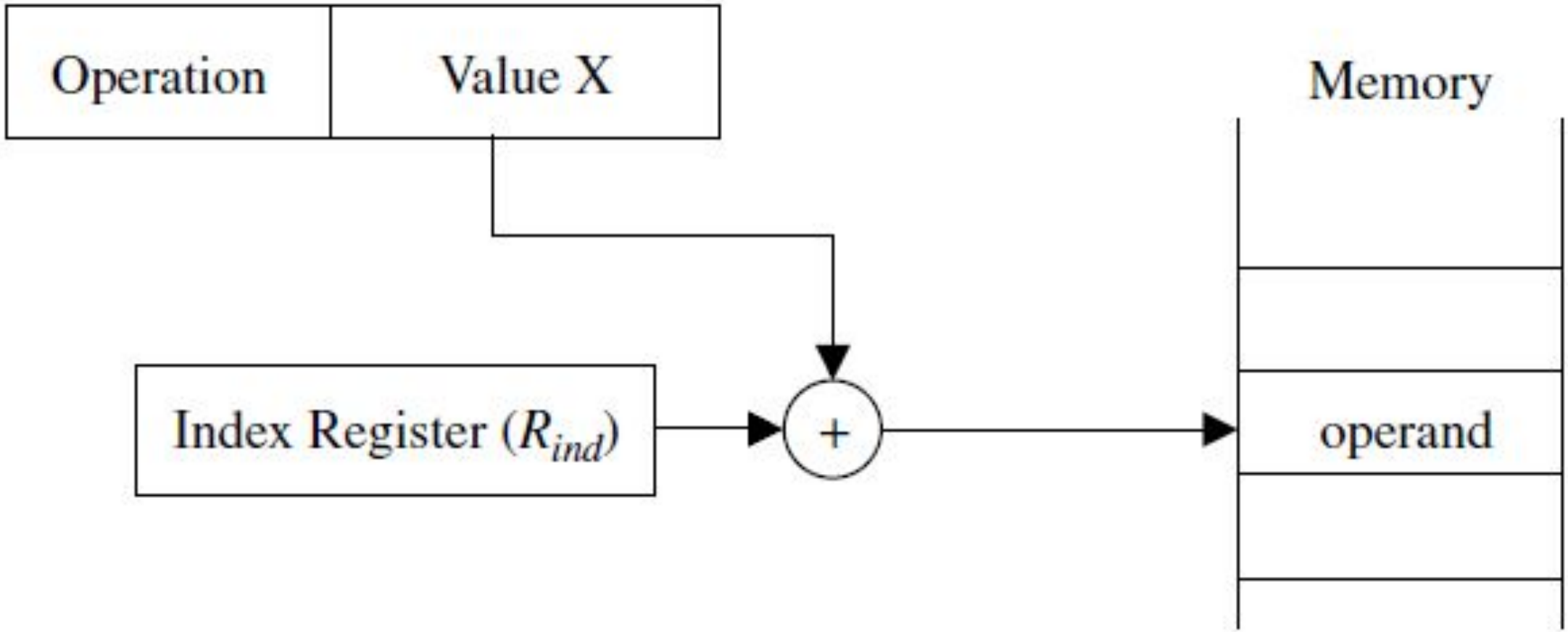


Figure 2.8 Illustration of the indexed addressing mode

Other Modes

- The addressing modes presented above represent the most commonly used modes in most processors.
- They provide the programmer with sufficient means to handle most general programming tasks.

- However, a number of other addressing modes have been used in a number of processors to facilitate execution of specific programming tasks.
- These additional addressing modes are more involved as compared to those presented above.

- Among these addressing modes the relative, autoincrement, and the autodecrement modes represent the most well-known ones. These are explained below.

- Relative Mode Recall that in indexed addressing, an index register, Rind , is used.
- Relative addressing is the same as indexed addressing except that the program counter (PC) replaces the index register.

- For example, the instruction LOAD X(PC), Ri loads register Ri with the contents of the memory location whose address is the sum of the contents of the program counter (PC) and the value X.
- Figure 2.9 illustrates the relative addressing mode.

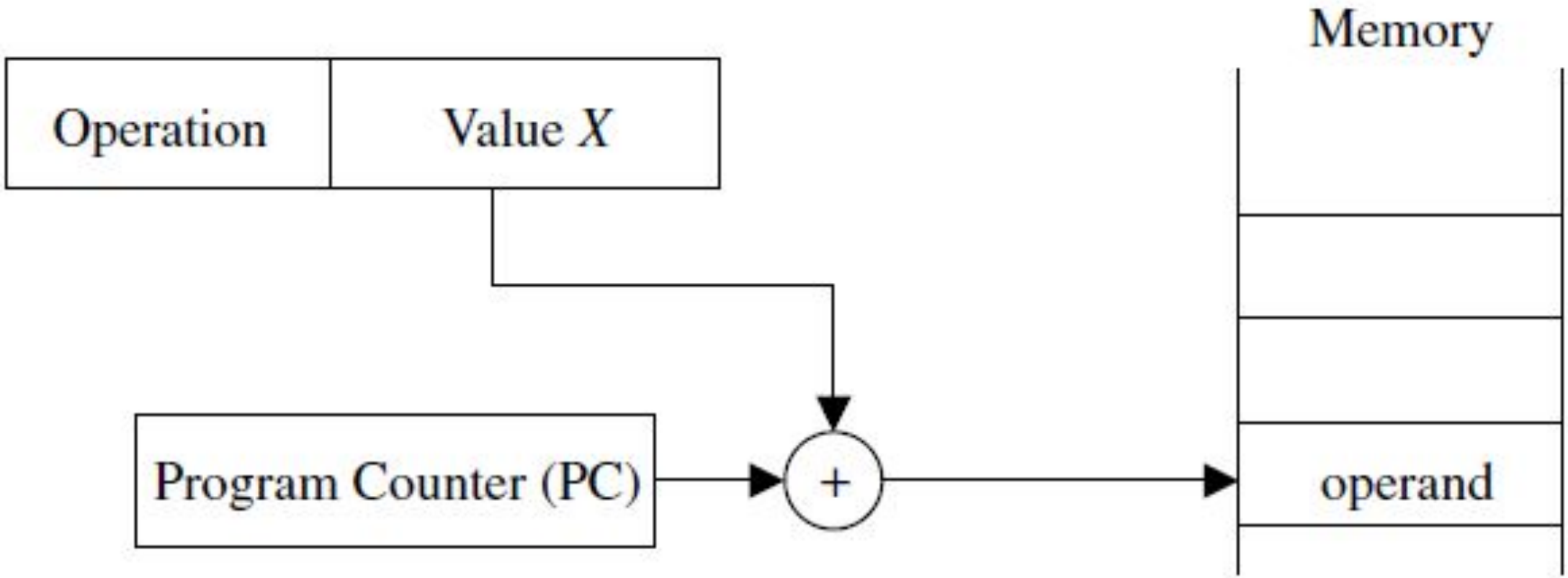


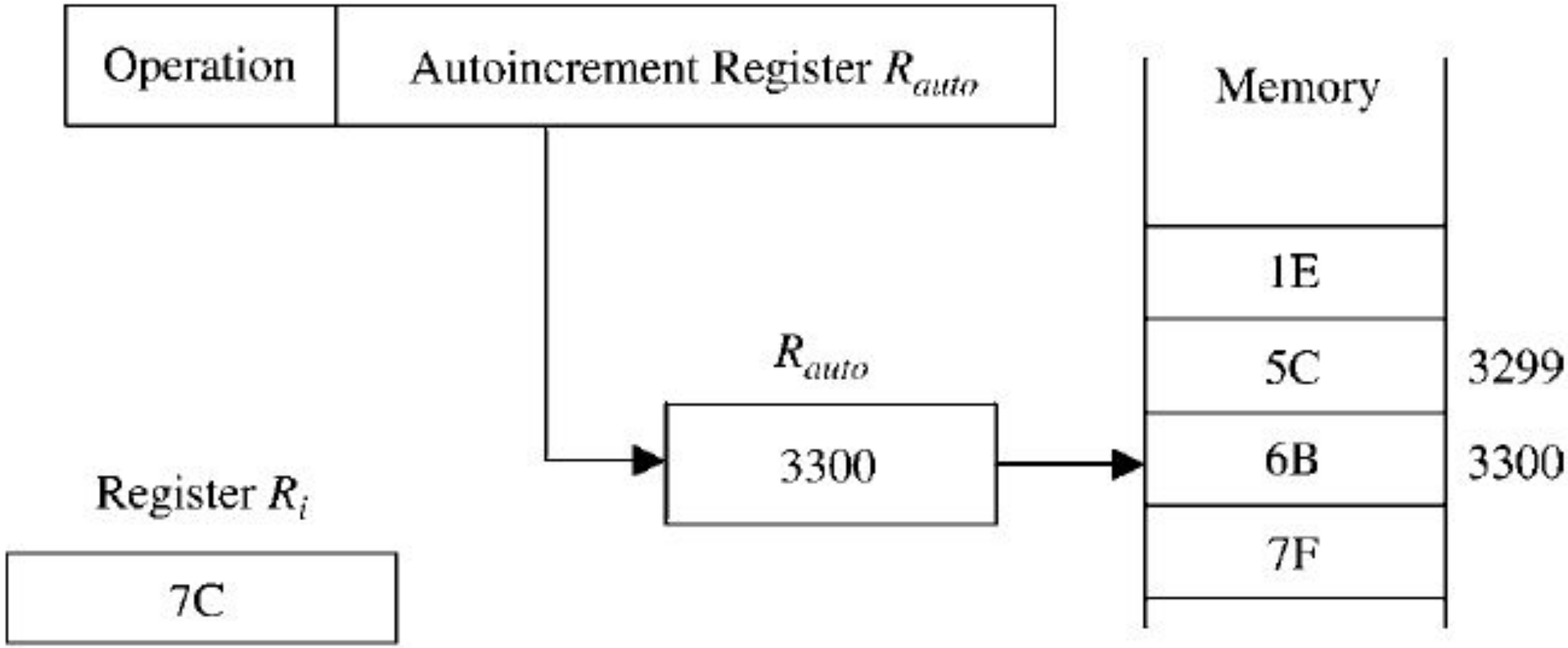
Figure 2.9 Illustration of relative addressing mode

Autoincrement Mode This addressing mode is similar to the register indirect addressing mode in the sense that the effective address of the operand is the content of a register, call it the autoincrement register, that is included in the instruction.

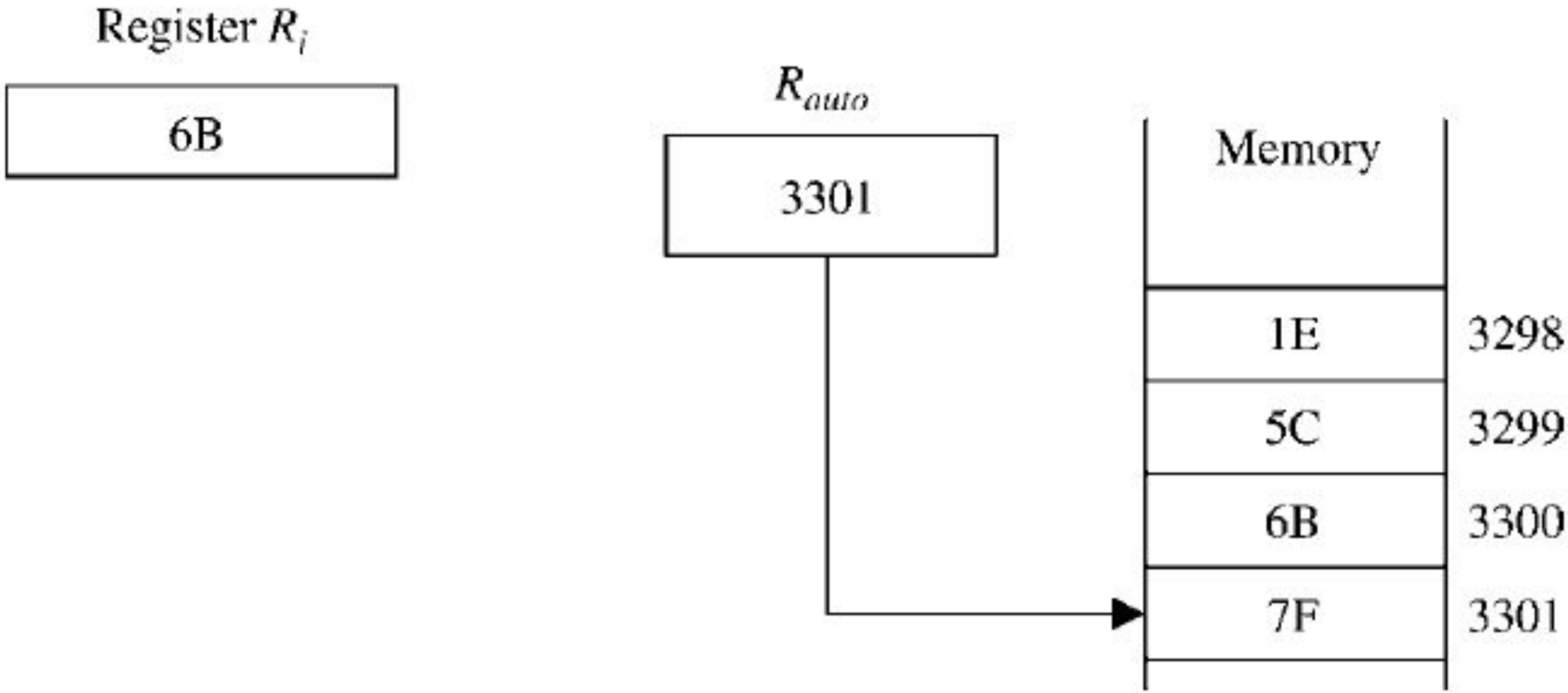
- However, with autoincrement, the content of the autoincrement register is automatically incremented after accessing the operand.
- As before, indirection is indicated by including the autoincrement register in parentheses.

- The automatic increment of the register's content after accessing the operand is indicated by including a (þ) after the parentheses.
- Consider, for example, the instruction LOAD (Rauto)þ, Ri.

- This instruction loads register Ri with the operand whose address is the content of register Rauto.
- After loading the operand into register Ri, the content of register Rauto is incremented, pointing for example to the next item in a list of items.



(*a*) Before execution



(*b*) After execution

Figure 2.10 Illustration of the autoincrement addressing mode

- Autodecrement Mode Similar to the autoincrement, the autodecrement mode uses a register to hold the address of the operand.
- However, in this case the content of the autodecrement register is first decremented and the new content is used as the effective address of the operand.

- In order to reflect the fact that the content of the autodecrement register is decremented before accessing the operand, a (2) is included before the indirection parentheses.
- Consider, for example, the instruction LOAD _ (Rauto), Ri.

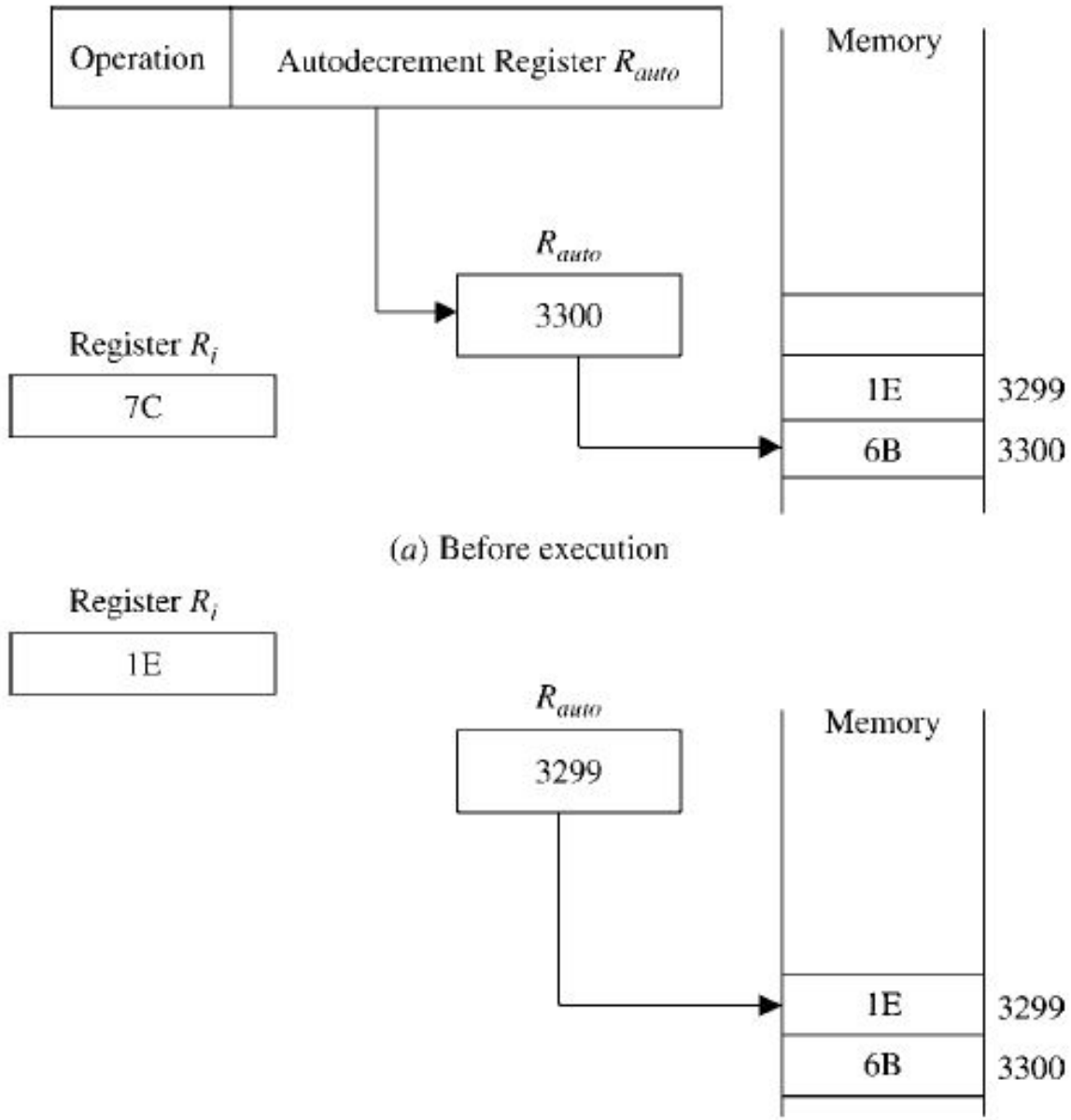


Figure 2.11 Illustration of the autodecrement addressing mode

THANK YOU