Computer Organization & Architecture

Chapter 2 – Addressing Modes

Zhang Yang 张杨

cszyang@scut.edu.cn

Autumn 2025

Content of this lecture

- 2.4 Addressing Modes
  - Immediate Mode
  - Absolute Mode
  - Register Mode
  - Register Indirect Mode
  - Indirect Mode (only in CISC)
  - Index Mode and Variations
  - Summary

Addressing Modes (1)

- Once we have determined the number of addresses contained in an instruction, the manner in which each address field specifies memory location must be determined.
  - How to specify where an operand for an instruction is located?
  - How the bits of an address field in an instruction are interpreted?
- Want the ability to reference a large range of address locations.

Addressing Modes (2)

- What is Addressing Modes?
  - The different ways in which the location of an operand is specified in an instruction are referred to as addressing modes.

Addressing Modes (2)

- Typical Addressing Modes

Table 2.1 RISC-type addressing modes.

Name	Assembler syntax	Addressing function
Immediate	#Value	Operand = Value
Register	R*i*	EA = R*i*
Absolute	LOC	EA = LOC
Register indirect	(R*i*)	EA = [R*i*]
Index	X(R*i*)	EA = [R*i*] + X
Base with index	(R*i*,R*j*)	EA = [R*i*] + [R*j*]

EA = effective address
Value = a signed number
X = index value

Immediate Mode (1)

- The operand is given explicitly in the instruction.
- Instruction Format

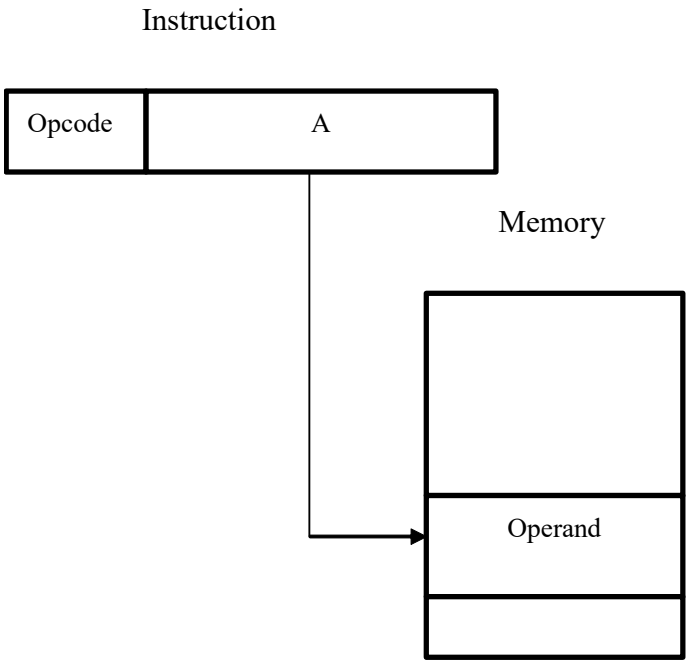
- Example
  - Add R4, R6, #200
  - [R4] ← [R6] + 200
- Usage
  - Define and use constants
  - Set initial values of variables

Immediate Mode (2)

- Advantage
  - No memory reference other than the instruction fetch is required to obtain the operand.
- Disadvantages
  - Only a constant can be supplied this way.
  - The size of the number is limited by the size of the address field.

Absolute Mode (1)

- The operand is in a memory location; the address of this location is given explicitly in the instruction.



- EA = A
  - EA : actual (effective) address of the location containing the referenced operand
  - A: contents of an address field in the instruction

Absolute Mode (2)

- Example
  - Integer NUM1, NUM2, NUM3
  - Load R2, NUM1
  - Load R3, NUM2
  - Load R3, NUM3

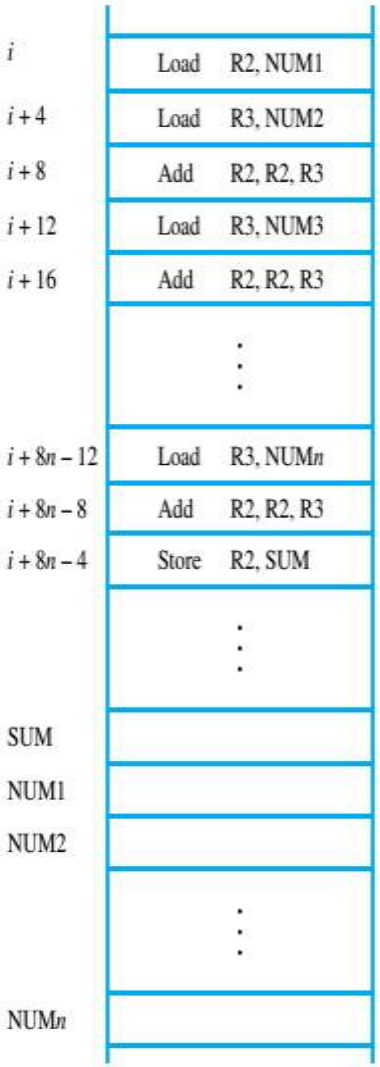


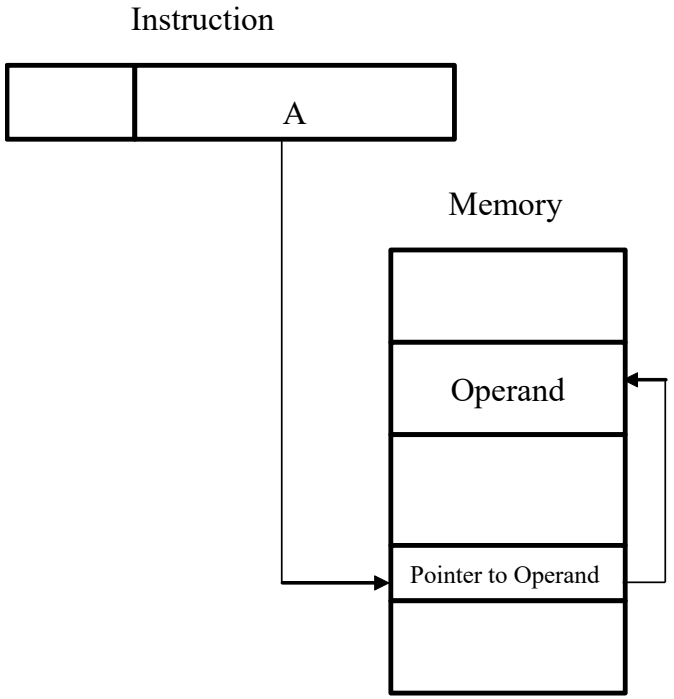
Figure 2.5 A program for adding *n* numbers.

Absolute Mode (3)

- In a system without virtual memory, the effective address will be either a main memory address or a register.
- In a virtual memory system, the effective address is a virtual address or a register address
- Usage
  - Access global variables whose address is known at compile time.
- Advantage
  - Only one memory reference and no special calculation.
- Disadvantages
  - The instruction will always access exactly the same memory location.
  - It provides only a limited address space.

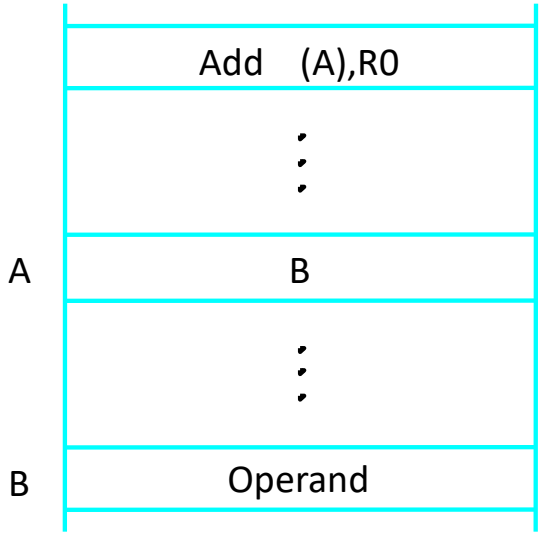
Indirect Mode (1)

- Indirect addressing through a memory location is also possible, but it is found only in CISC-style processors.
- The effective address of the operand is the contents of a memory location whose address appears in the instruction.
- EA = [A]



Indirect Mode (2)

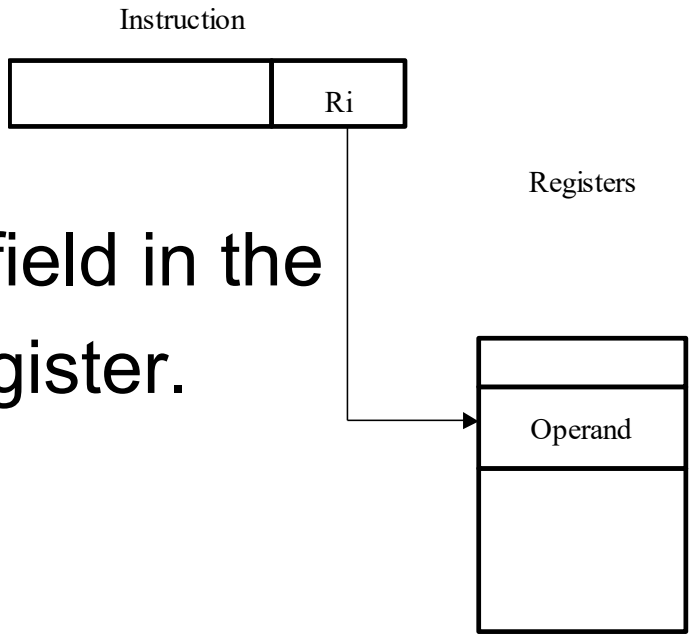
- Example
  - Add (A), R0



- Advantage
  - The address space is very large.
    - E.g. Memory word length N→2^N address space
- Disadvantage
  - Instruction execution requires two memory references to fetch the operand : one to get its address and a second to get its value.

Register Mode (1)

- The operand is the contents of a processor register; the address of the register is given in the instruction.
- EA = Ri
  - Ri : contents of an address field in the instruction that refers to a register.
- Example
  - Add R4, R2, R3
  - [R4]←[R2]+[R3]

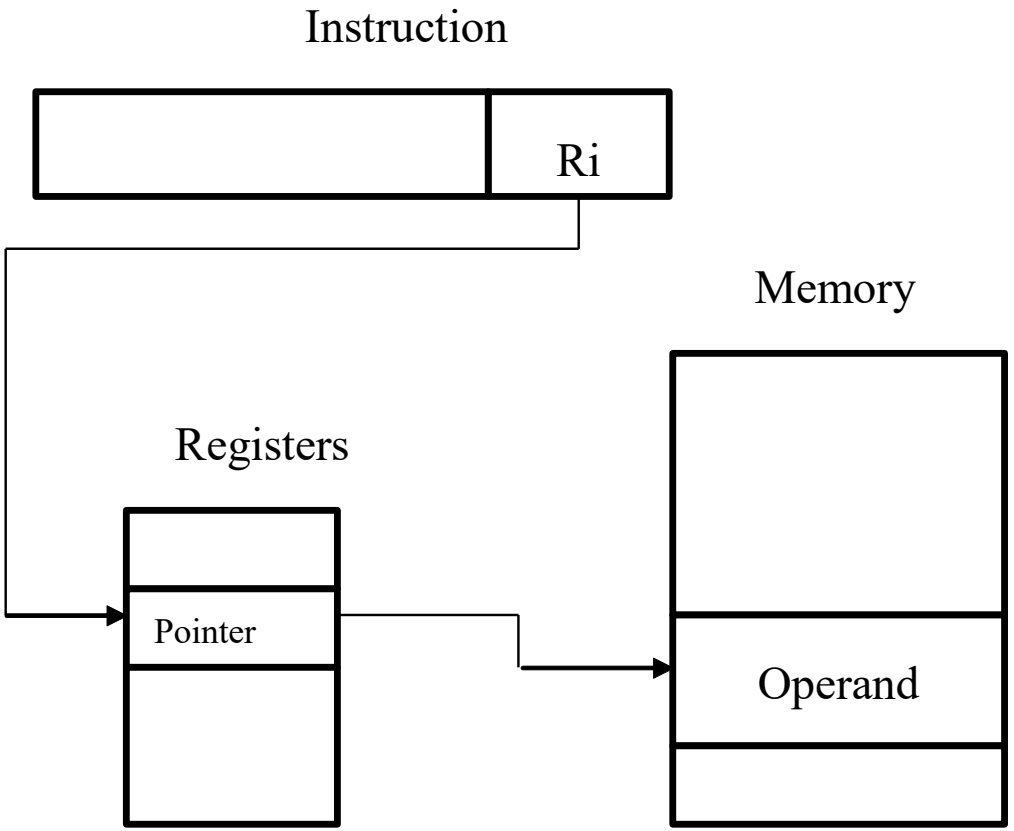


Register Mode (2)

- Usage
  - Access variables which are accessed most often
- Advantages
  - Only a small address field is needed in an instruction.
  - No memory reference references are required.
- Disadvantage
  - The address space is very limited.

Register Indirect Mode (1)

- The effective address of the operand is the contents of a register that is specified in the instruction.
- EA = [Ri]



Register Indirect Mode (2)

- Example
  - Load R2, (R5)

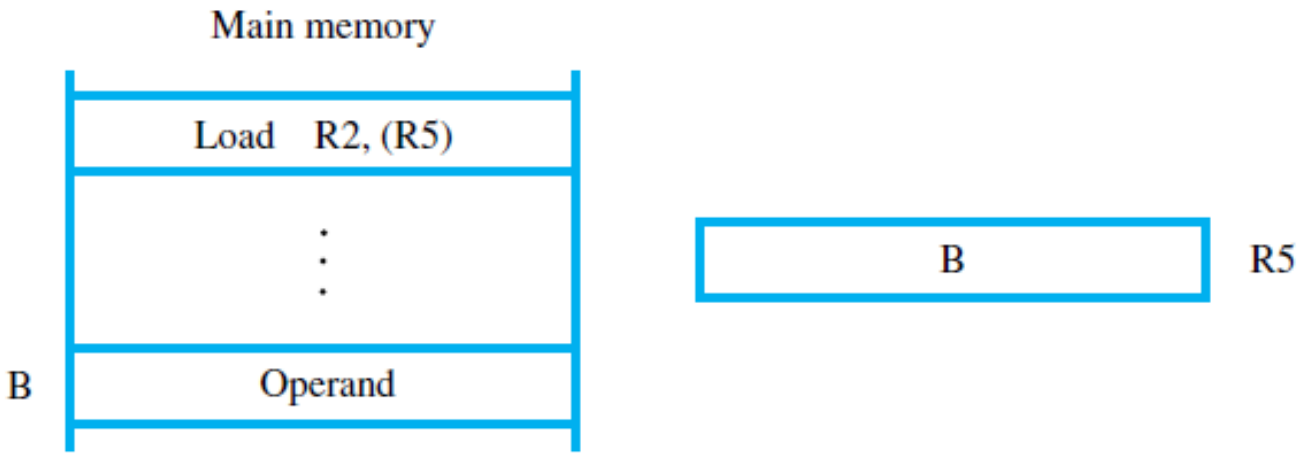


Figure 2.7 Register indirect addressing.

Register Indirect Mode (3)

- Example: Using indirect addressing to access a list of n numbers

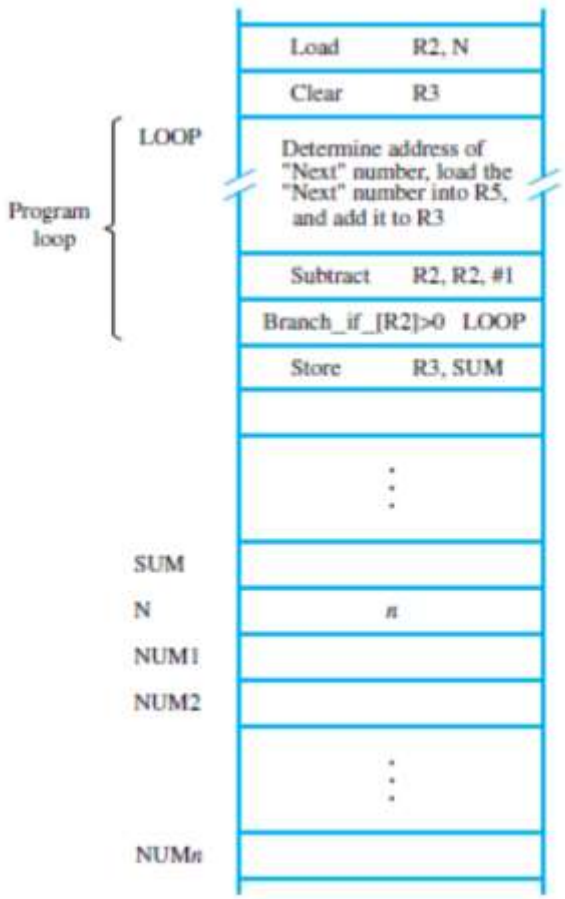


Figure 2.6 Using a loop to add n numbers.

Register Indirect Mode (4)

- Example: Using indirect addressing to access a list of n numbers

```
        Load            R2, N           Load the size of the list.
        Clear           R3              Initialize sum to 0.
        Move            R4, #NUM1       Get address of the first number.
LOOP:   Load            R5, (R4)        Get the next number.
        Add             R3, R3, R5      Add this number to sum.
        Add             R4, R4, #4      Increment the pointer to the list.
        Subtract        R2, R2, #1      Decrement the counter.
        Branch_if_[R2]>0 LOOP           Branch back if not finished.
        Store           R3, SUM         Store the final sum.
```

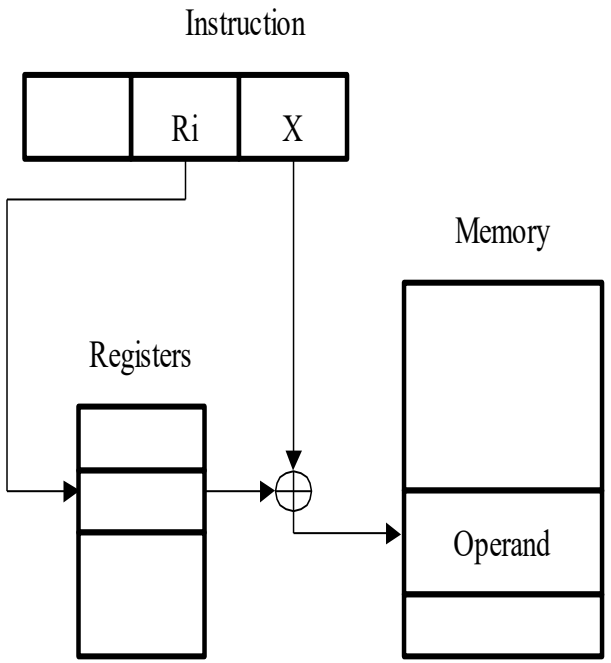
Figure 2.8 Use of indirect addressing in the program of Figure 2.6.

Register Indirect Mode (5)

- Advantages
  - It can reference memory without paying the price of having a full memory address in the instruction.
  - Reduce the memory access times.
- Disadvantage
  - Extra memory cycles

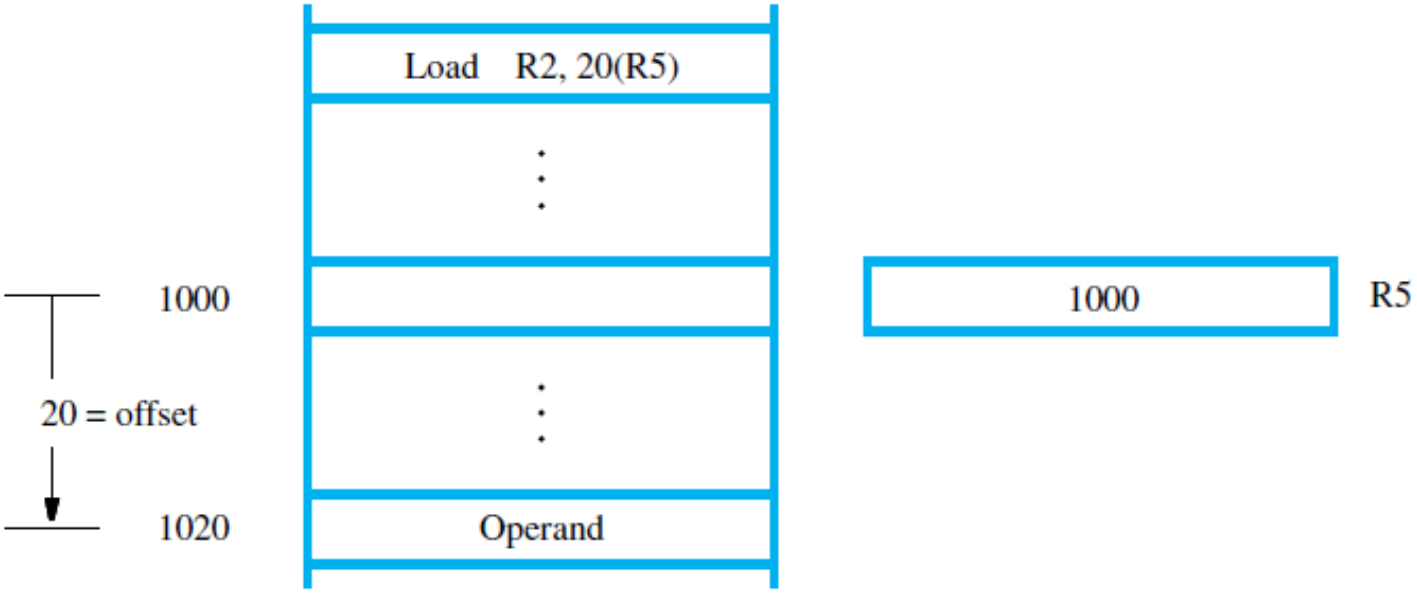
Index Mode and Variations (1)

- Index Mode
  - The effective address of the operand is generated by adding a constant value to the contents of a register (index register).
  - EA = X + [Ri]
    - X: offset, the constant value contained in the instruction
    - The contents of the index register are not changed in the process of generating the effective address.



Index Mode and Variations (2)

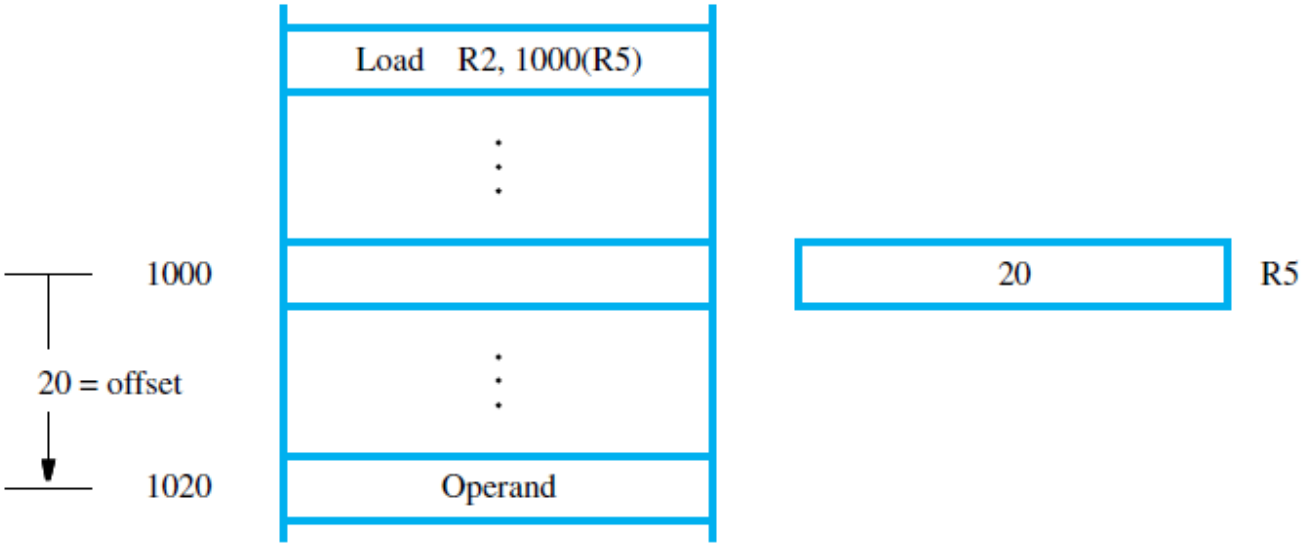
- Two Ways of Using the Index Mode
  - Offset is given as a constant.



(a) Offset is given as a constant

Index Mode and Variations (3)

- Two Ways of Using the Index Mode (ctd.)
  - Offset is in the index register.
    - This form requires an offset field in the instruction large enough to hold an address.



(b) Offset is in the index register

Index Mode and Variations (4)

- Index Mode Usage
  - Facilitate access to an operand whose location is defined relative to a reference point within the data structure in which the operand appears.

Index Mode and Variations (5)

- Index Mode Usage (ctd.)
  - Example
    - N rows and four columns array
    - The memory is byte addressable and the word length is 32 bits.

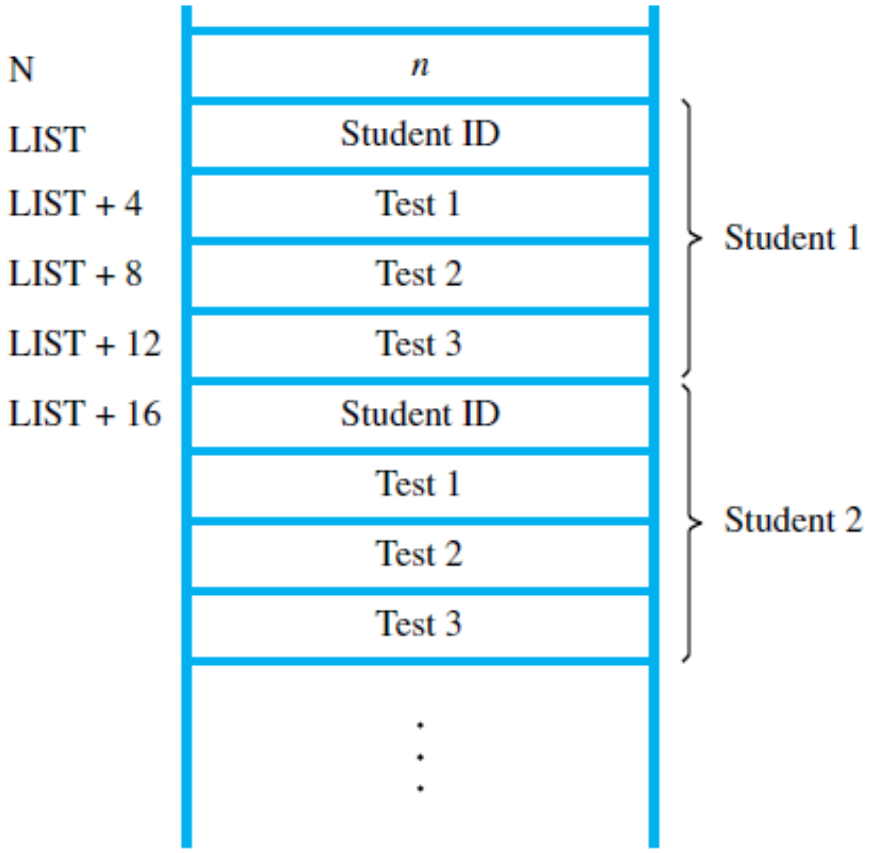


Figure 2.10 A list of students' marks.

Index Mode and Variations (6)

- Index Mode Usage (ctd.)
  - Example (ctd.)

```
        Move        R2, #LIST     Get the address LIST.
        Clear       R3
        Clear       R4
        Clear       R5
        Load        R6, N         Load the value n.
LOOP:   Load        R7, 4(R2)     Add the mark for next student's
        Add         R3, R3, R7     Test 1 to the partial sum.
        Load        R7, 8(R2)     Add the mark for that student's
        Add         R4, R4, R7     Test 2 to the partial sum.
        Load        R7, 12(R2)    Add the mark for that student's
        Add         R5, R5, R7     Test 3 to the partial sum.
        Add         R2, R2, #16   Increment the pointer.
        Subtract    R6, R6, #1    Decrement the counter.
        Branch_if_[R6]>0 LOOP     Branch back if not finished.
        Store       R3, SUM1      Store the total for Test 1.
        Store       R4, SUM2      Store the total for Test 2.
        Store       R5, SUM3      Store the total for Test 3.
```

Figure 2.11 Indexed addressing used in accessing test scores in the list in Figure 2.10.

Index Mode and Variations

- Variations of index addressing
  - Base with index
    - (Ri, Rj)
    - A second register (base register) is used to contain the offset X.
    - EA = [Ri] + [Rj]
  - Base with index and offset
    - X(Ri, Rj)
    - Use index register, base register and a constant.
    - EA = [Ri] + [Rj] + X

Summary

- 知识点: Addressing Modes
  - What is addressing modes?
  - Typical Addressing Modes
    - Immediate
    - Absolute
    - Indirect
    - Register
    - Register indirect
    - Index mode and variation
- 掌握程度
  - 针对每种寻址方式，掌握该种方式的操作数存放位置，有效地址如何计算，优缺点。

Exercise (1)

- 1. In register indirect addressing mode, where is the operand in?
  - A. general-purpose register
  - B. memory location
  - C. program counter
  - D. stack

Exercise (2)

- 2. In the following addressing modes, which does not belong to RISC style computers?
  - A. immediate mode
  - B. absolute mode
  - C. direct mode
  - D. indirect mode

Exercise (3)

- 3. 3. Which addressing mode adopt the following ways to get operand address? Addressing memory by giving a register (explicit or implicit) plus a constant offset.
  - A. direct mode
  - B. indirect mode
  - C. register indirect mode
  - D. index mode