Addressing Modes$_{\text{(contd.)}}$

Indexing and arrays: Lists and arrays are represented using index mode.

- **Index mode:** The effective address of the operand is generated by adding a constant value to the contents of a register. This register may be a special purpose register or a general purpose register of the processor. This is referred to as **index register**.

- Index mode is presented as

  X(Ri) X is the constant value

- The effective address of the operand is calculated as

  EA=X+[Ri] ⟵ index register

- The constant X may be an explicit number given in the instruction or a symbolic name representing a numerical value.

Addressing Modes$_{\text{(contd.)}}$

- When the instruction is converted into machine code, the constant X is considered as a part of the instruction and requires fewer bits than the word length of the computer.
- **Offset/Displacement address:** Indicates the distance or location of a specific data, in an array or other data structure, from the base address. This is known as **relative addressing**.

See Figure 2.13, 2.14 and 2.15

- What is dimension of the array (Figure 2.14)?
- How many rows and columns are there in the array (Figure 2.14)?

Addressing Modes (contd.)

- **Relative mode:** The effective address is determined by the Index mode using the PC instead of a general purpose register.

- X(PC), because PC always identifies the current execution point in a program.

Additional Addressing Modes: Many computers provide additional modes to aid in programming tasks. The following two modes are used to access data in successive memory locations:

- **Autoincrement mode:** The effective address of an operand is the contents of a register given in the instruction. After accessing this operand, the contents of the register are incremented automatically to point to the next item in the list.

- (Ri)+ The contents of the register would be incremented by (in a byte addressable memory with 32 bit word length, the increment would be by 4)

- **Autodecrement mode:** The contents of the register are first decremented automatically and then used as the effective address of the operand.

  -(Ri)

See Figure 2.16

RISC

Reduced Instruction SET Computer (RISC) is a type of microprocessor architecture that utilizes a small and highly optimized set of instructions.

- The first RISC projects came from IBM, Stanford, and University of California, Berkeley in the late 70s and early 80s.
- Beginning in the mid-1990s, RISC technology was integrated into PCs and, in the early 21st century, into mobile devices such as smartphones and tabs.
- RISC architecture reduces the cycles per instruction at the cost of the number of instructions per program; thus increases the CPU performances.
- Example of RISC processor: MIPS, SPARC, RISC-V, ARM.

Characteristic of RISC

- Simpler instruction, hence simple instruction decoding.
- Instruction comes undersize of one word.
- Code size is large.
- Instruction takes a single clock cycle to get executed.
- More general-purpose registers.
- Simple Addressing Modes.
- Less Data types.
- Performs only Register to Register Arithmetic operations.
- Pipeline can be achieved.
- Focuses on software.

Pipelining

- To improve the performance of the CPU, one way is to arrange the hardware in a way that more than one operation can be performed at the same time.
- Pipelining is a technique where multiple instructions are overlapped during execution. Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end.
- Simultaneous execution of more than one instruction takes place in a pipelined processor.
- RISC processor has 5 stage instruction pipeline to execute all the instructions in the RISC instruction set.
- **Stage 1 (Instruction fetch):**The CPU reads instructions from the address in the memory whose value is present in the program counter.
- **Stage 2 (Instruction decode):** Instruction is decoded and get the values from the registers used in the instruction.
- **Stage 3 (Instruction execute):** ALU operations are performed.
- **Stage 4 (Memory access):** In this stage, memory operands are read and written from/to the memory that is present in the instruction.
- **Stage 5 (Write back):** Result is written back to the register present in the instruction.

Pipelining (contd.)

Instr. No.	Pipeline Stage						
1	IF	ID	EX	MEM	WB		
2		IF	ID	EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX
Clock Cycle	1	2	3	4	5	6	7

RISC Instruction Set and Addressing Mode

- Hardwired control unit is designed for RISC instruction set.
- RISC instructions operate on processor’s registers only.
- For ALU operation, the operands are either on registers or given direct values in the instruction.
- Load, Store, Add, Sub,…...
- Immediate, Register, Absolute, Indirect, Index addressing modes.
- Unconditional conditional branch, procedure call, system call.

CISC

Complex Instruction SET Computer (CISC) attempts to minimize the number of instructions per program but at the cost of increase in number of clock cycles per instruction.

- CISC processors were evolved in 1970s before the evolution of RISC.
- The original goal was to produce fewer lines of assembly codes.
- A single instruction will perform loading, evaluating, and storing operations, hence it is complex.
- Example of CISC processors: Intel 80486, IBM 370/168, AMD, Motorola 68000.
- A vast majority of laptops and desktops use CISC processors.
- Used in security system, home automation, etc.

Characteristic of CISC

- Complex instruction, hence complex instruction decoding.
- Instructions are larger than one-word size.
- Code size is small.
- Instruction may take more than a single clock cycle to get executed.
- Less number of general-purpose registers as operation get performed in memory itself.
- Complex Addressing Modes.
- More Data types.
- Performs register to register, register to memory, or memory to memory arithmetic operations.
- Focuses on hardware.

CISC Instruction Set and Addressing Mode

- A single instruction can execute several operations.
- A single instruction has several low level instructions.
- Uses Move instruction, instead of Load/Store, to access memory operands.
- Instructions directly access memory locations.
- Requires several clock cycles to execute a single instruction.
- Immediate, direct, register, index and indirect addressing modes.
- Autoincrement, Autodecrement and relative mode also.
- Complex addressing mode makes the memory access flexible.

Execution of A=B+C in RISC and CISC

- RISC
  - Load B,R1
  - Load C,R2
  - Add R1,R2
  - Store R2,A
- CISC
  - Move B,A
  - Add C,A

A summary of RISC vs. CISC

RISC	CISC
Focuses on hardware	Focuses on software
Uses hardwired control unit only	Uses both hardwired and micro programmed control unit
Fixed sized instruction	Variable sized instructions
Performs only register to register arithmetic operation	Performs register to register/register to memory/memory to memory operations also
Requires more number of registers	Requires less number of registers
Code is large size	Code is small size
An instruction is within one word	Larger than one word

MIPS

- Million Instructions Per Second (MIPS) is a RISC architecture developed by MIPS Computer Systems.
- Initially it has 32 bits to 64 bits general purpose registers with word size 32 bits.
- In total 111 instructions:
    - 21 arithmetic instructions (Add, Subtract, Multiply, Divide, ….)
    - 8 logic instructions (And, Or, …..)
    - 8 bit manipulation instructions (Shift left, Shift right, …..)
    - 8 comparison instructions ()
    - 25 branch/jump instructions (conditional branch, unconditional branch, Jump, ….)
    - 15 load instructions (Load word, Load immediate, …..)
    - 10 store instructions (Store word, Store immediate, …..)
    - 8 move instructions ()
    - 4 miscellaneous instructions ()