William Stallings
Computer Organization and Architecture
7th Edition

Chapter 13
Reduced Instruction Set Computers

v/s

Complex Instruction Set Computers

Introduction

- Hardware fused with software (**Intel v/s Apple**)
- Intel's hardware oriented approach is termed as CISC while that of Apple is RISC
- **Instruction Set Architecture- Interface** to allow easy communication between the programmer and the hardware.
- ISA- execution of data, copying data, deleting it, editing
- Instruction Set , Addressing Modes,

RISC-Reduced Instruction Set Computer

- RISC processor design has separate **digital circuitry**

  in the control unit

- Signals needed for the execution of each instruction

  in the instruction set of the processor.

- Examples of RISC processors:
  - IBM RS6000, MC88100
  - DEC's Alpha 21064, 21164 and 21264 processors

CISC-Complex Instruction Set Computer

- Control unit □micro-electronic circuitry

  - generates a set of control signals □activated by a micro-code

- The primary goal of CISC architecture is to complete a task in as few lines of assembly code as possible.

- **Examples of CISC processors are:**
  - Intel 386, 486, Pentium, Pentium Pro, Pentium II, Pentium III
  - Motorola's 68000, 68020, 68040, etc.

CISC processor features

- Instruction set with 120-350 instructions
- Variable instruction/data formats
- Small set of general purpose registers(8-24)
- A large number of addressing modes
- High dependency on micro program
- Complex instructions to support HLL features

CISC processor features

- Complex pipelining
- Many functional chips needed to design a computer using CISC
- Difficult to design a superscalar processor
- Rarely supports on chip cache memory.
- Difficulty in handling data dependancy,resource conflict,branching problem during pipelining.

RISC processor features

- Instruction set with limited number of instructions
- Simple instruction format
- Large set of CPU registers
- Very few addressing modes
- Easy to construct a superscalar processor

RISC processor features

- Hardwired control unit for sequencing microinstructions
- Supports on chip cache memory
- All functional units on a single chip, faster CPU.
- Simple pipelining

Example for RISC vs. CISC

Consider the the program fragments:

CISC

```
mov ax, 10
mov bx, 5
mul bx, ax
```

RISC

```
        mov ax, 0
        mov bx, 10
        mov cx, 5
Begin   add ax, bx
        loop Begin
```

The total clock cycles for the CISC version might be:

(2 movs × 1 cycle) + (1 mul × 30 cycles) = 32 cycles

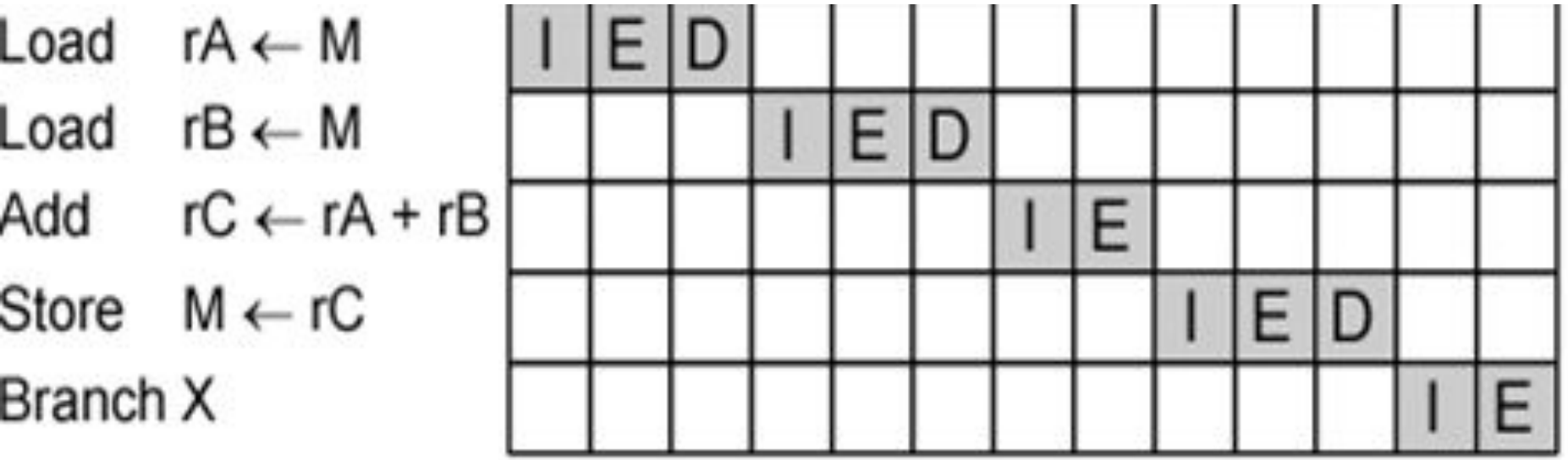
While the clock cycles for the RISC version is:

(3 movs × 1 cycle) + (5 adds × 1 cycle) + (5 loops × 1 cycle) = 13 cycles

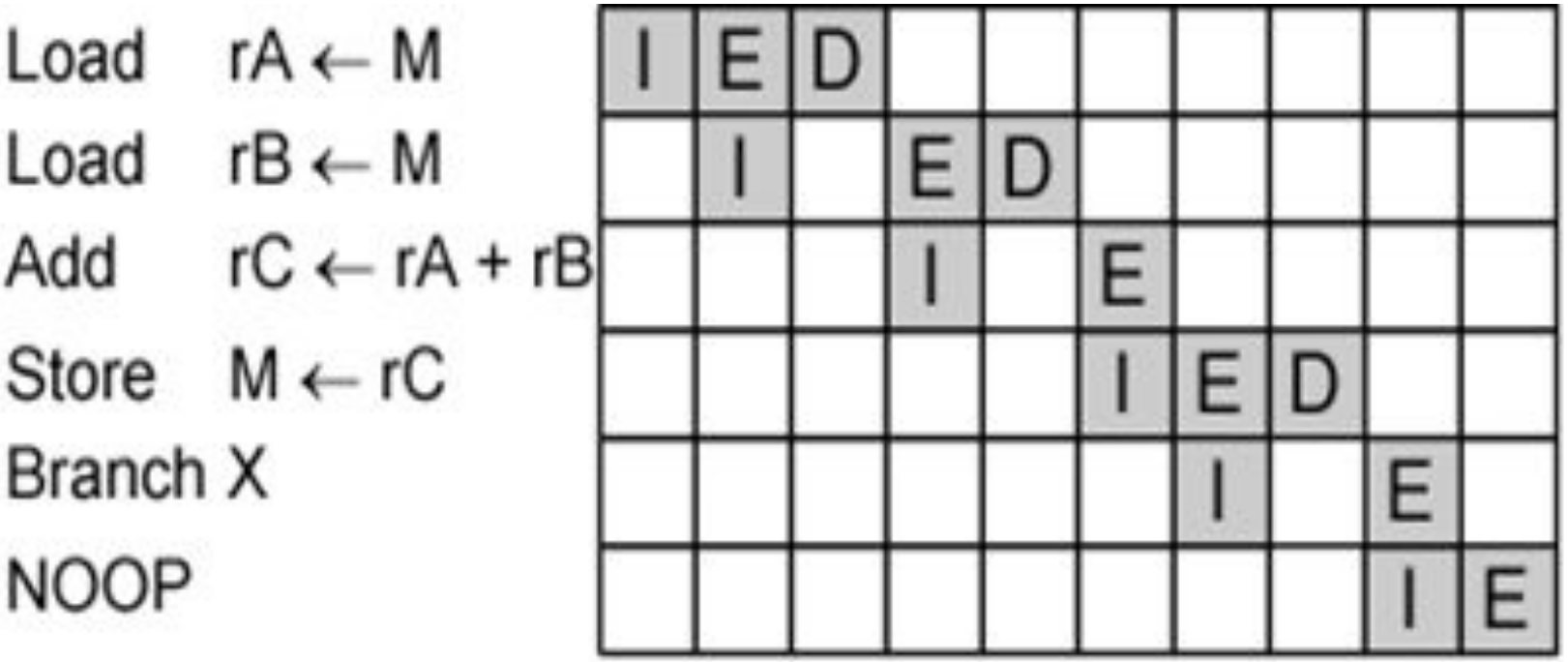
CISC	RISC
Emphasis on hardware	Emphasis on software
Multiple instruction sizes and formats	Instructions of same set with few formats
Less registers	Uses more registers
More addressing modes	Fewer addressing modes
Extensive use of microprogramming	Complexity in compiler
Instructions take a varying amount of cycle time	Instructions take one cycle time
Pipelining is difficult	Pipelining is easy

RISC Pipelining

- Most instructions are **register to register**
- Two phases of execution, **I E**
  - I: Instruction fetch
  - E: Execute
    - ALU operation with register input and output
- For load and store(**memory**),**I E D**
  - I: Instruction fetch
  - E: Execute
    - Calculate memory address
  - D: Memory
    - Register to memory or memory to register operation



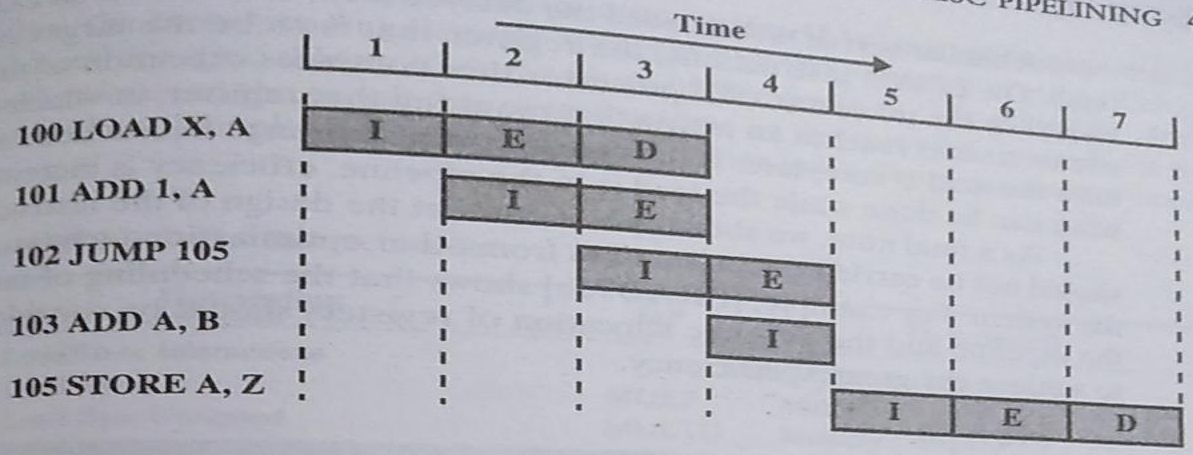
(a) Sequential execution



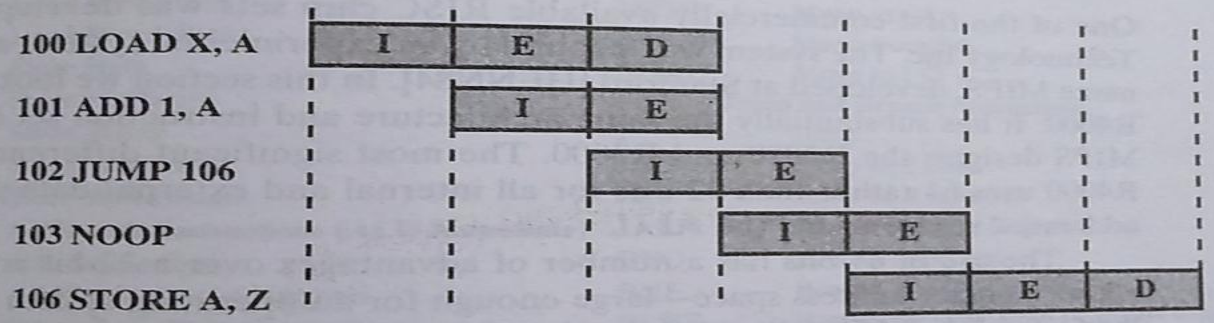
(b) Two-stage pipelined timing

Load rA ← M	I	E	D					
Load rB ← M		I	E	D				
NOOP			I	E				
Add rC ← rA + rB				I	E			
Store M ← rC					I	E	D	
Branch X						I	E	
NOOP							I	E

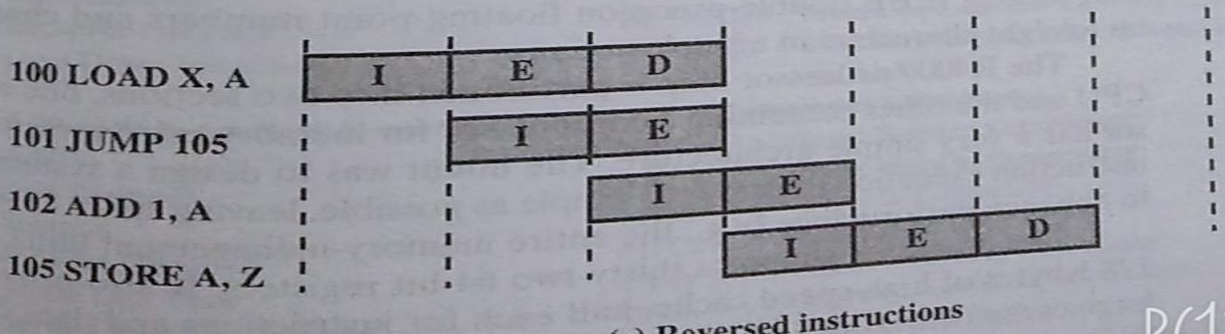
(c) Three-stage pipelined timing



(a) Traditional pipeline



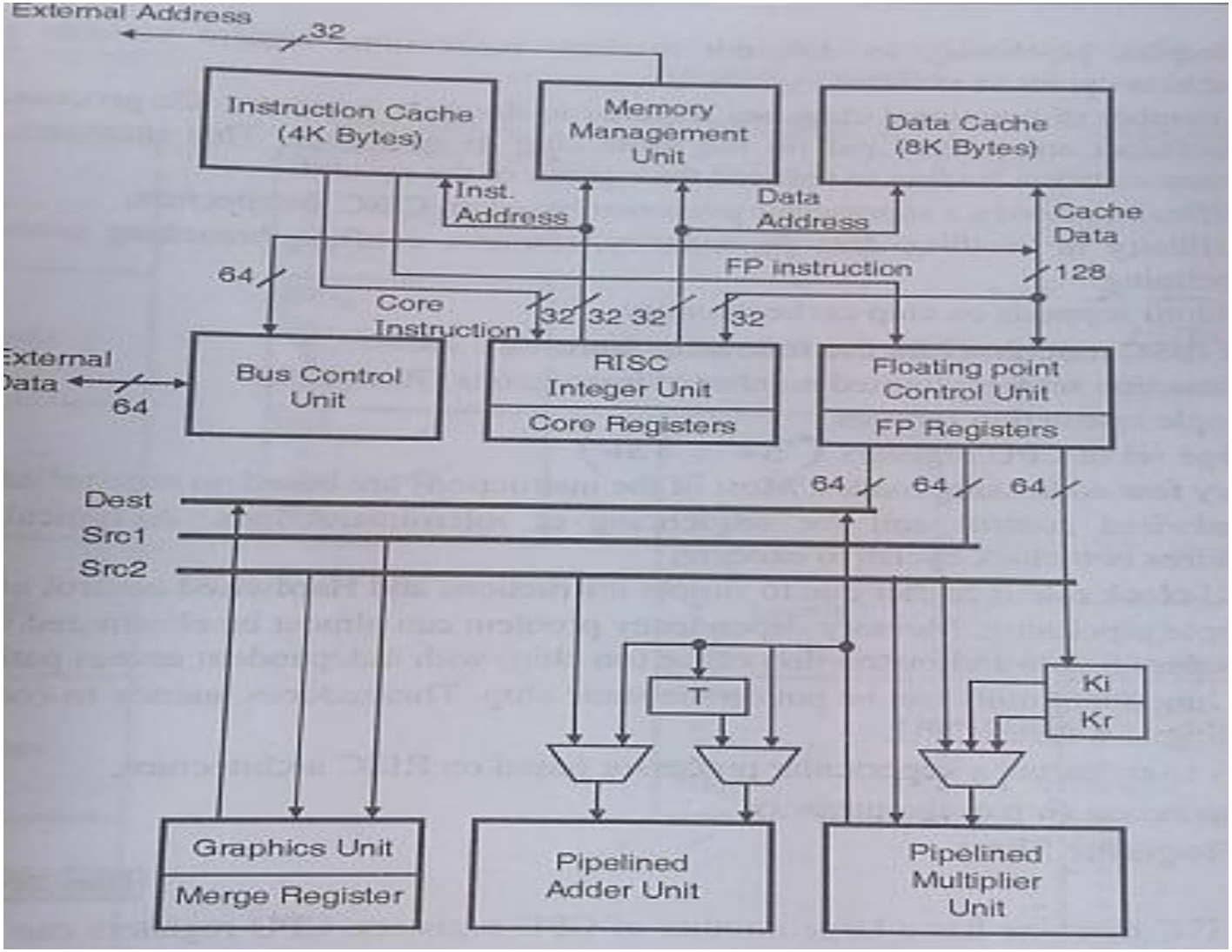
(b) RISC pipeline with inserted NOOP



(c) Reversed instructions

Figure 13.7 ... of the Delayed Branch

RISC Architecture



RISC Architecture

- 9 **functional units** interconnected by multiple data paths with width ranging from 32-128 bits
- All internal- external buses are 32 bit wide
- Separate instruction (4KB)and data cache(8KB)
- **MMU-** implements paged virtual memory structure
- **RISC integer unit** executes load,store,fetch etc
- 2 floating point units , multiplier unit and adder unit
- **Graphics unit** to support 3D drawing

CISC Architecture

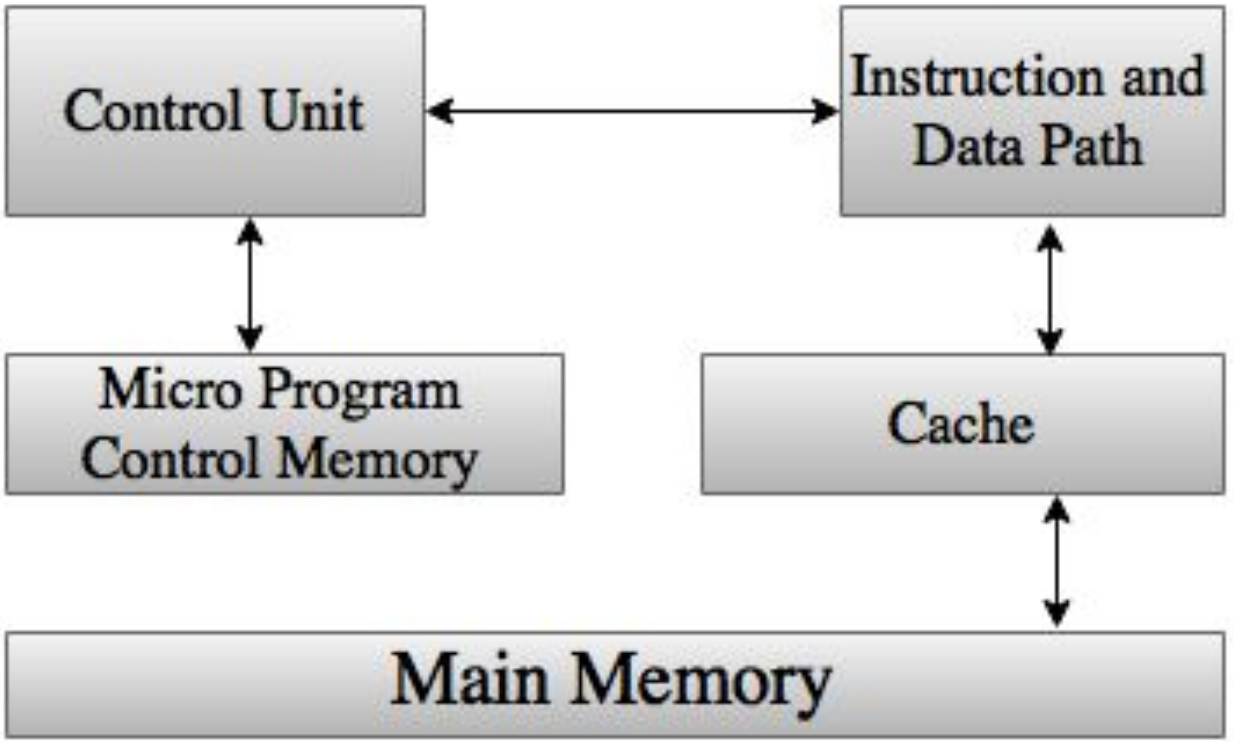
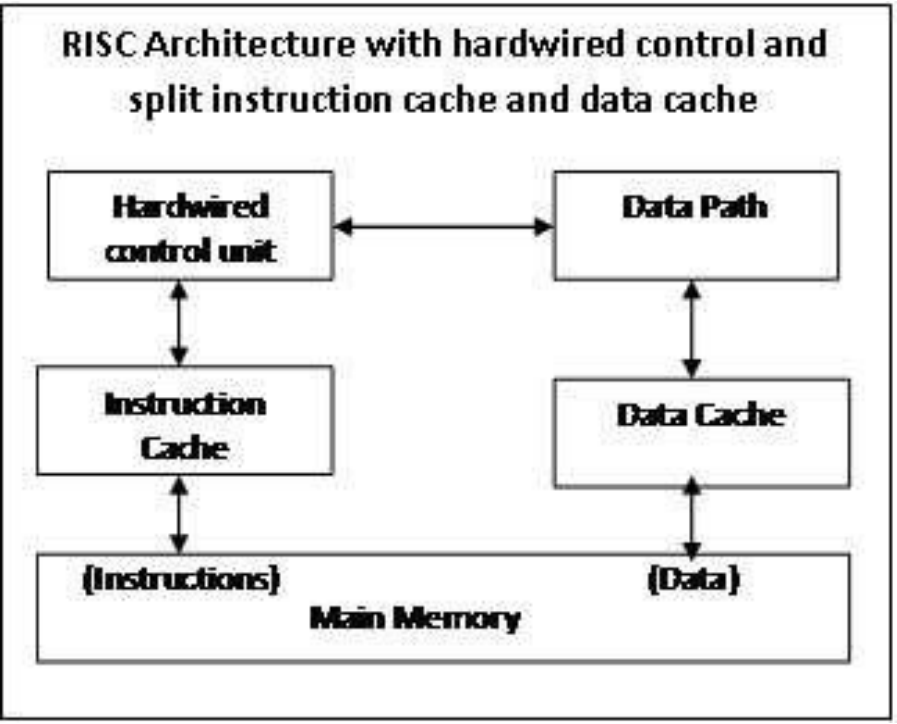
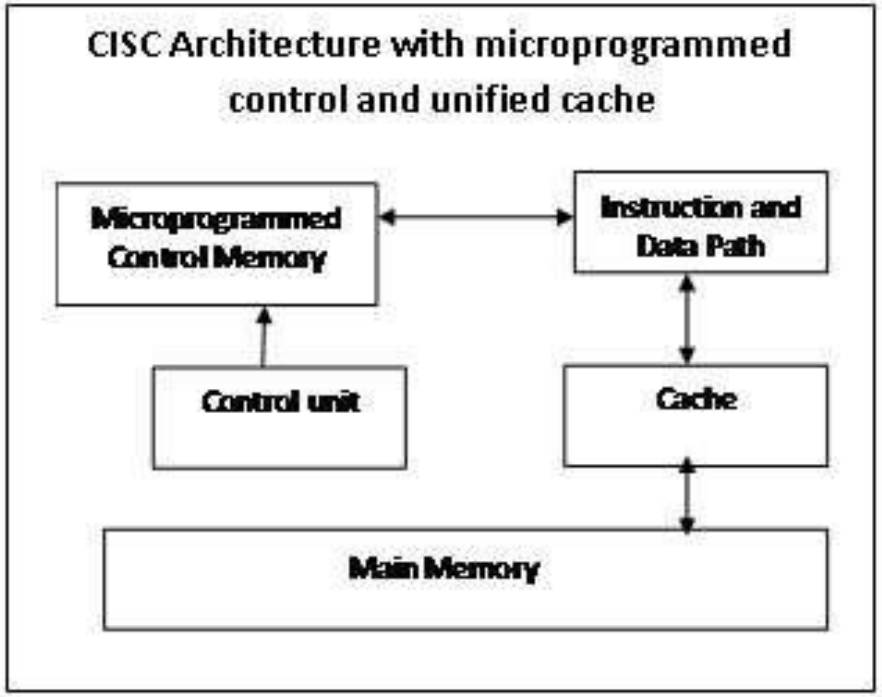


Fig. CISC Architecture


RISC Architecture with hardwired control and split instruction cache and data cache
Hardwired control unit
Data Path
Instruction Cache
Data Cache
(Instructions)
(Data)
Main Memory


CISC Architecture with microprogrammed control and unified cache
Microprogrammed Control Memory
Instruction and Data Path
Control unit
Cache
Main Memory