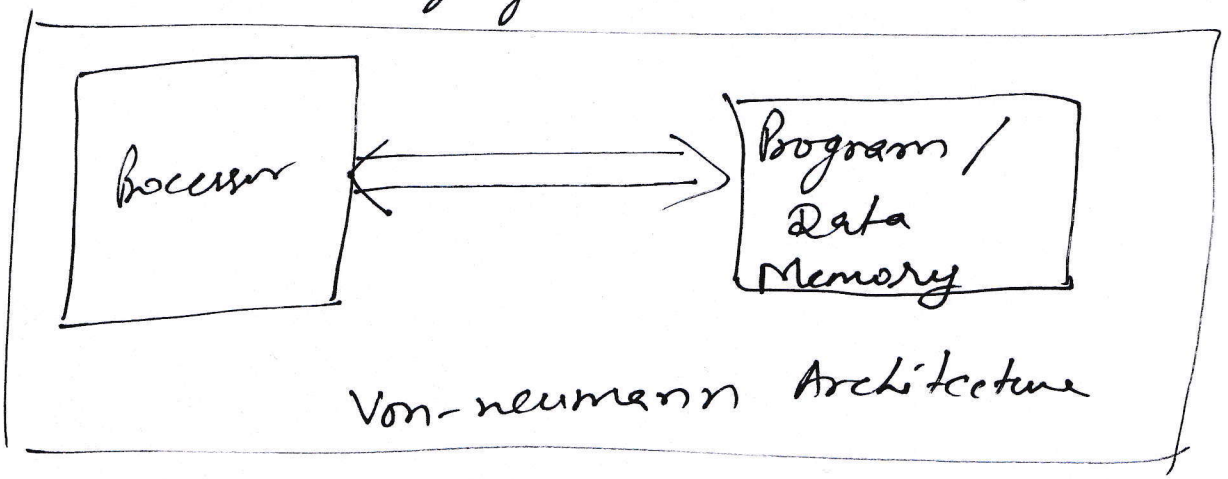
→ Software Program : A set of instructions required to solve a problem.

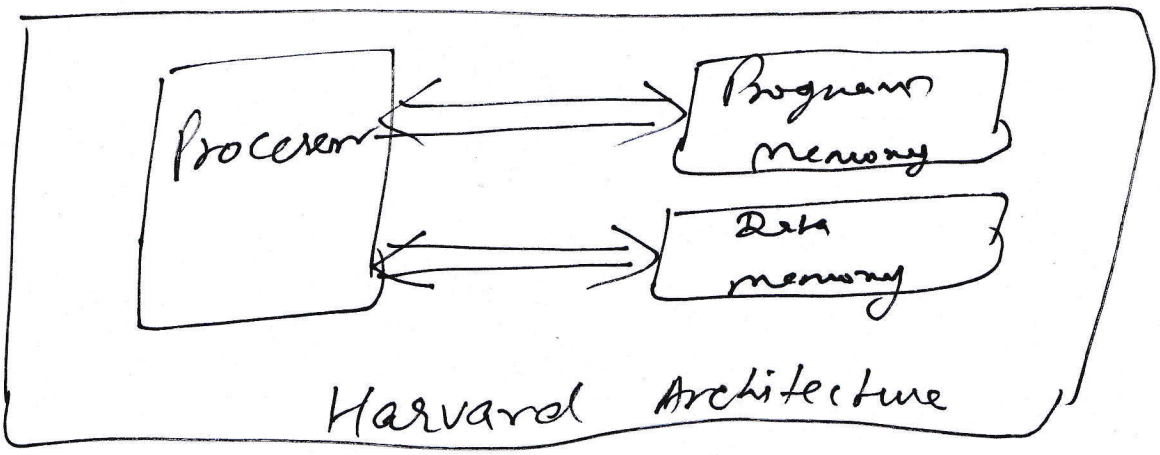
- → Application Software
- → System Software

→ Classification of Computer Architecture

- → Von-neumann architecture
- → Harvard architecture
- → Emerging architecture



Von-neumann Architecture



Harvard Architecture

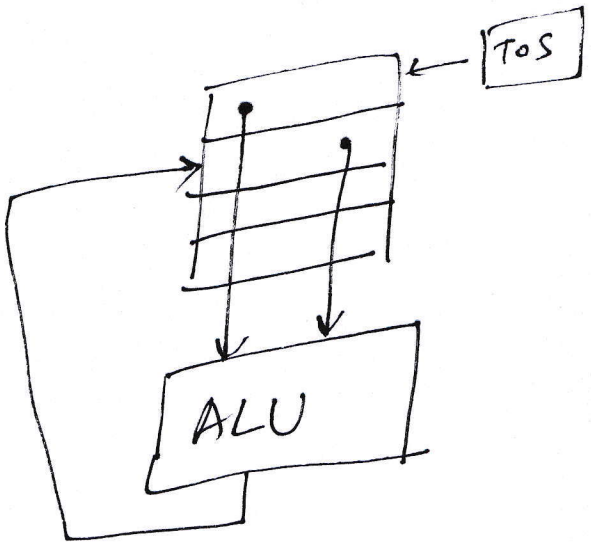
Evolution of instruction sets

1) Accumulator- based - 1960's (EDSAC, IBM 1130)
2.) Stack- based - 1960-70 (IBM 360)
3.) Memory- Memory based - 1970-80
4.) Register-Memory based - 1970- present (Intel x86)
5.) Register- Register based - 1960- present (MIPS, SPARC)

Code → Z = X+Y;

Stack- based

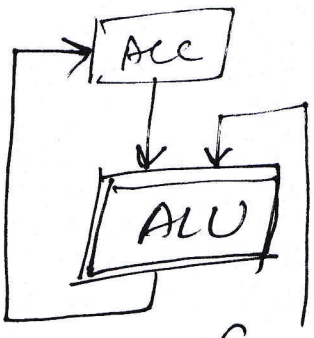
```
PUSH X
PUSH Y
ADD
POP  Z
```



→ The add instruction pops two elements from stack, adds them and pushes back result

- Accumulator-based

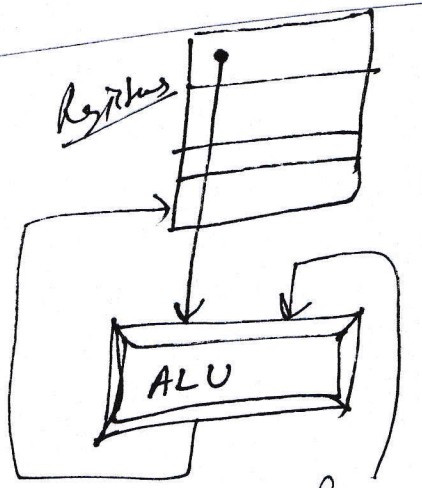
```
LOAD   X    // Acc = Mem[x]
ADD    Y    // Acc = Acc + Mem[Y]
STORE  Z    // Mem[z] = Acc
```



- All instructions assume that one of the operands (and result also) is in a special purpose register called accumulator.

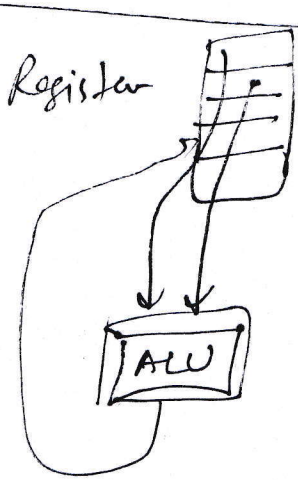
- Register-Memory machine

```
LOAD  R2, X   // R2 = Mem[X]
ADD   R2, Y   // R2 = R2 + Mem[Y]
STORE Z, R2   // Mem[z] = R2
```



- Register-Register m/c

```
LOAD  R2, X
LOAD  R2, Y
ADD   R3, R1, R2
```



→ Also called as load-store architecture, as only LOAD and store instructions can access memory

CISC & RISC Architecture

CISC – Complex Instruction Set Computer

- IBM 360/370 (1960-70)
- VAX-11/780 (1970-80)
- Intel x86/Pentium (1985- present)

↓

Only CISC instructions set that survived over generations

→ Desktop PC's / Laptops use these

→ Sufficient hardware resources available today to translate from CISC to RISC internally

→ The volume of chips manufactured is so high.

RISC – Reduced instruction set Computer (RISC)

→ Also referred to as Load-store Architecture.

→ Only LOAD & STORE instructions access memory.

→ All other instructions operate on processor registers.

- CDC 6600
- MIPS family
- SPARC
- ARM micro-controller family

→ Almost all the computers today use a RISC based pipeline for efficient implementation.

→ RISC-based computers use compilers to translate into RISC instructions.

→ CISC-based computers (Intel x86) use hardware to translate into RISC instructions.

CISC	RISC
The original microprocessor ISA	Redesigned ISA that emerged in the early 1980s
Instructions can take several clock cycles	Single-cycle instructions
Hardware-centric design - the ISA does as much as possible using hardware circuitry	Software-centric design - High-level compilers take on most of the burden on coding many software steps from the programmer
More efficient use of RAM than RISC	Heavy use of RAM (can cause bottlenecks if RAM is limited)
Complex & Variable length instructions	Simple, standardized instructions
Large number of instructions	small number of fixed-length instructions
Compound addressing modes	Limited addressing mode

Pipeline in Executing Instructions

→ Instruction execution is divided into 5 stages:
- Instruction fetch (IF)
- Instruction Decode (ID)
- ALU operation (EX)
- Memory Access (MEM)
- Write Back results (WB)

→ These five stages can be executed in an overlapped fashion in a pipeline architecture.

	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	WB
5					IF	ID	EX	MEM ..
Clock cycles	1	2	3	4	5	6	7	8 ..

↓ will result in pipeline slowdown as both are accessing memory.

this can be done in Harvard Archi.