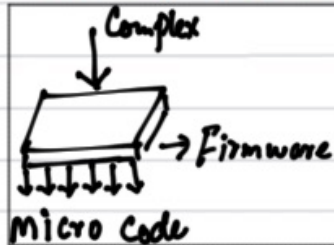


## RISC

MVL 23, 29, (#63 + #64), R4



- Flexible inst. size
- Short code
- Compiler does less.
- Favourable for High level languages
- Less effort for programmers.
- processor does more
- High heat generation
- Requires cooling system
- Desktops
- Bigger instruction set
- Memory to memory  
↳ Load & store is built in.
- Complex production
- Expensive

## CISC

C = Complex  
I = Instruction  
S = Set  
C = Computer

## RISC

R = Reduced  
I = Instruction  
S = Set  
C = Computer

LDM #63

ADD #64

PROD 29

PROD R4

STO 23

- fixed inst. size
- Large code size
- Compiler does more
- More effort for programmer.
- processor has separate circuits for every instruction.
- processor does less.
- less heat generation
- Casings are enough to pacify the heat.
- Portable devices.
- less commands
- register to register
- Easy production methods
- Cheaper



**Zak**  
ZAFAR ALI KHAN

CISC	RISC
C = Complex	R = Reduced
I = Instruction	I = Instruction
S = Set	S = Set
C = Computer	C = Computer
<ul style="list-style-type: none"> <li>- Single instruction is executed in multiple cycles.</li> <li>- Uses control unit.               <ul style="list-style-type: none"> <li>↳ Multiclock cycles per instruction.</li> </ul> </li> <li>- Many type of instructions are available for memory addressing.</li> <li>- Microprogrammed Control Unit</li> <li>- Pipelining is difficult</li> </ul>	<ul style="list-style-type: none"> <li>- Single instruction is executed in single cycle.</li> <li>- Separate unit for every instruction.               <ul style="list-style-type: none"> <li>↳ Single clock cycle per instruction.</li> </ul> </li> <li>- Only load and store instructions are available.</li> <li>- Hard-wired control unit.</li> <li>- Pipelining is easier.</li> </ul>



**Zak**  
ZAFAR ALI KHAN