1. Instruction Format

- **R Type instructions**
  Register type instructions require two registers and sometimes shift amount sh.

opcode(6)	rs(5)	rt(5)	Unused	sh(5)	fn(6)

- **I Type instructions**
  Immediate instructions require only one register and immediate value. One among rs and rt is used.

opcode(6)	rs(5)	rt(5)	Immediate(16)

- **Jump Instructions**
  Jump or branch instructions do not require any registers. Only 26 bit address is required.

opcode(6)	Jump target address(jta)(26)

2. Fetch instruction:

We do not need to store the instruction in a separate register since every instruction is executed in a single cycle. So when the new instruction is executed, the old instruction is already executed.

3. Load/Store Instructions

- Need of separate data and instruction cache
  The data cache and instruction cache are required to be accessed in the same clock cycle in a single cycle execution unit. More over after data access in load instruction registers need to be updated. If we make single cache for both instructions and data, there won't be enough edges to access both instruction and data. That is why two cache are required. One can be positive edge triggered while other negative ede triggered.

4. CPU-memory interface

A unidirection address bus has been made from CPU to memory using input port in verilog and a birectional data bus between CPU and memory using inout port and tristate circuit as shown below.