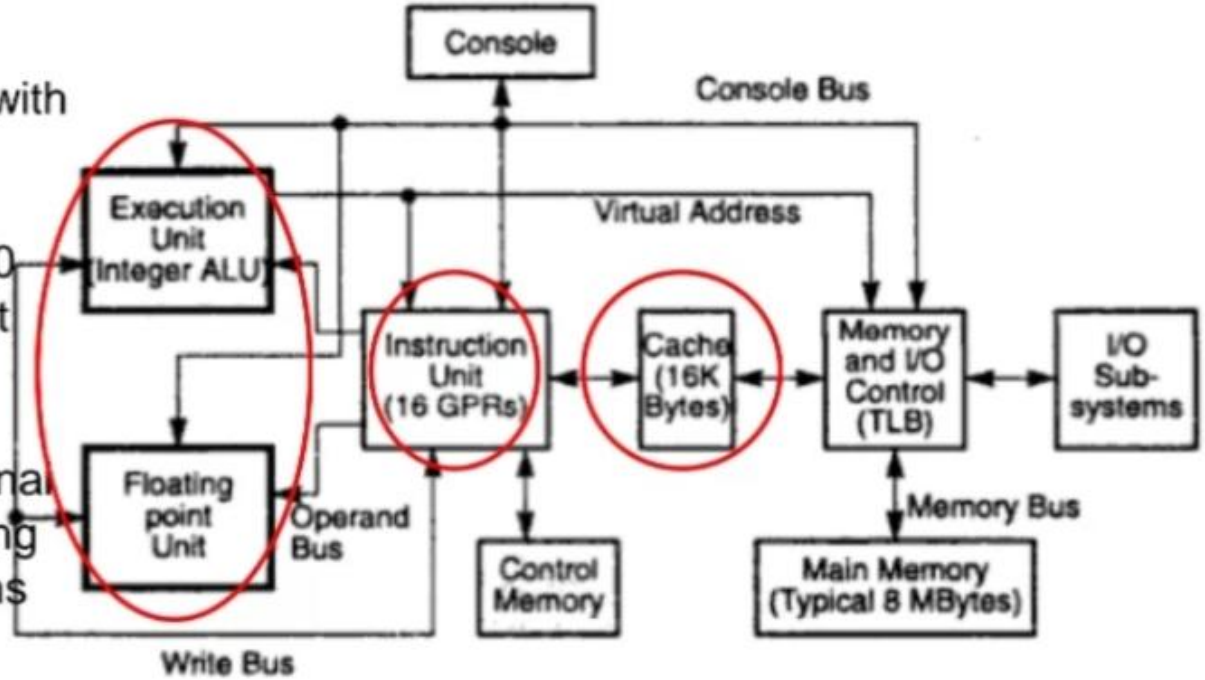
# Example 1

- Typical CISC architecture with Microprogrammed control
- Instruction set contains 300 instructions with 20 different addressing modes
- CPU consist of two functional units for execution of floating point and integer instructions
- Unified cache holds both instructions and data




Captions:

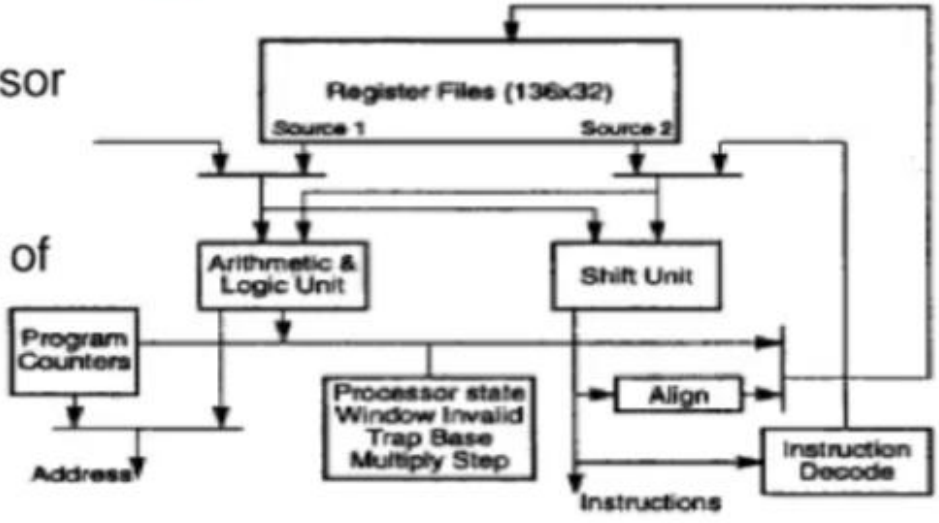
CPU = Central Processor Unit

# RISC Scalar Processor

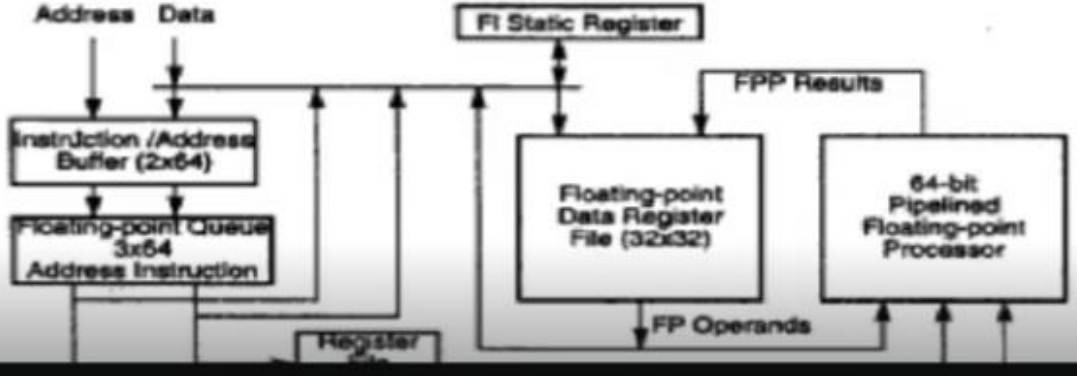
- Generic RISC processors are called *scalar RISC* because they are designed to issue one instruction per cycle
- RISC processors push some of the less frequently used operations into software
- RISC processors depend heavily on a good compiler because complex HLL instructions are to be converted into primitive low level instructions, which are few in number

# Example 1

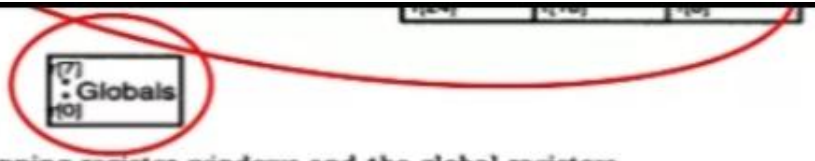
- SPARC stands for scalable processor architecture
- Scalability is due to use of number of *register windows* (explained on next slide)
- Floating point unit (FPU) is implemented on a separate chip




(a) The Cypress CY7C601 SPARC processor



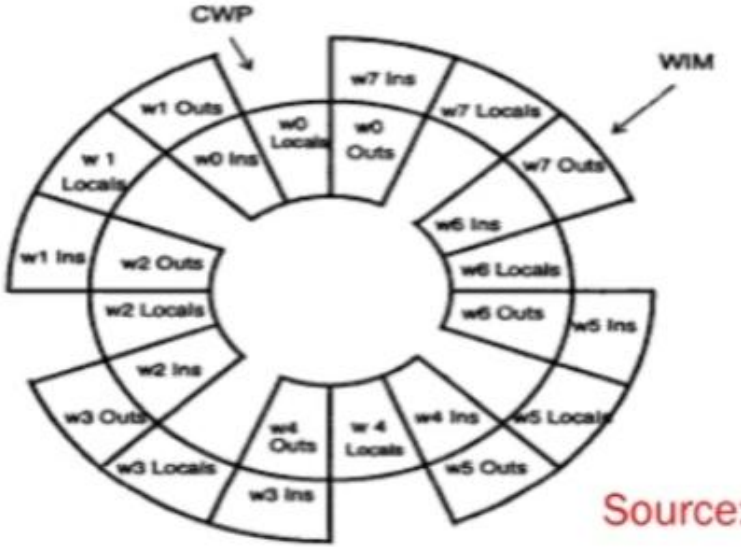
- Eight of these registers are *global registers* shared by all procedures




(a) Three overlapping register windows and the global registers

- Remaining twenty four registers are ***window registers*** associated with only one procedure

- Concept of using overlapped registers is the most important feature introduced




(b) Eight register windows forming a circular stack

- Each register window is divided into three sections – *Ins, Locals* and *Outs*

Source: Kai Hwang

- *Locals* are addressable by each procedure and *Ins & Outs* are shared among procedures