PRESENTATION ON

RISE and CISE Architecture

Subject :- Computer Organization

Submitted by :-DIPTA BERA

College roll no :-CSE/23/021

University roll no :-10700123052

Department Of Computer Science and Engineering

College of Engineering and Management,Kolaghat

What is RISC?

- RISE ?

RISC, or *Reduced Instruction Set Computer*. is a type of microprocessor architecture that utilizes a small, highly-optimized set of instructions, rather than a more specialized set of instructions often found in other types of architectures..

- HISTORY

The first RISC projects came from IBM, Stanford, and UC-Berkeley in the late 70s and early 80s. The IBM 801, Stanford MIPS, and Berkeley RISC 1 and 2 were all designed with a similar philosophy which has become known as RISC. Certain design features have been characteristic of most RISC processors:

- *one cycle execution time*: RISC processors have a CPI (clock per instruction) of one cycle. This is due to the optimization of each instruction on the CPU and a technique called PIPELINING
- *pipelining*: a techique that allows for simultaneous execution of parts, or stages, of instructions to more efficiently process instructions;
- *large number of registers*: the RISC design philosophy generally incorporates a larger number of registers to prevent in large amounts of interactions with memory

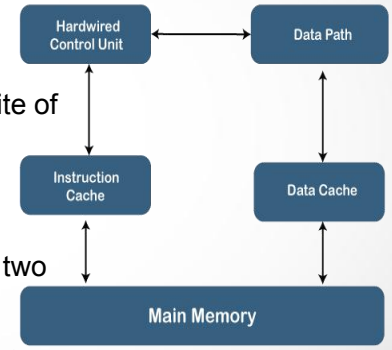
RISC Attributes

The main characteristics of CISC microprocessors are:

- Extensive instructions.
- Complex and efficient machine instructions.
- Microencoding of the machine instructions.
- Extensive addressing capabilities for memory operations.
- Relatively few registers.

In comparison, RISC processors are more or less the opposite of the above:

- Reduced instruction set.
- Less complex, simple instructions.
- Hardwired control unit and machine instructions.
- Few addressing schemes for memory operands with only two basic instructions, LOAD and
- STORE
- Many symmetric registers which are organised into a register file.



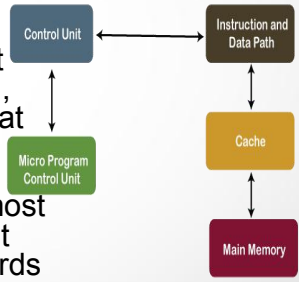
RISC Architecture

What is CISC?

- CISC is an acronym for Complex Instruction Set Computer and are chips that are easy to program and which make efficient use of memory. Since the earliest machines were programmed in assembly language and memory was slow and expensive, the CISC philosophy made sense
- Most common microprocessor designs such as the Intel 80x86 and Motorola 68K series followed the CISC philosophy.
- But recent changes in software and hardware technology have forced a re-examination of CISC and many modern CISC processors are hybrids, implementing many RISC principles.
- CISC was developed to make compiler development simpler. It shifts most of the burden of generating machine instructions to the processor. For example, instead of having to make a compiler write long machine instructions to calculate a square-root, a CISC processor would have a built-in ability to do this.

Most CISC hardware architectures have several characteristics in common:

- Complex instruction-decoding logic, driven by the need for a single instruction to support multiple addressing modes.
- A small number of general purpose registers. This is the direct result of having instructions which can operate directly on memory and the limited amount of chip space not dedicated to instruction decoding, execution, and microcode storage.
- Several special purpose registers. Many CTSC designs set aside special registers for the stack pointer, interrupt handling, and so on. This can simplify the hardware design somewhat, at the expense of making the instruction set more complex.
- A 'Condition code" register which is set as a side-effect of most instructions. This register reflects whether the result of the last operation is less than, equal to, or greater than zero and records if certain error conditions occur.



CISC Architecture

Complex Instruction Set Computer (CISC) Characteristics

- F Major characteristics of a CISC architecture
- »1) A large number of instructions - typically from 100 to 250
- instruction
- »2) Some instructions that perform specialized tasks and are
- used infrequently
- »3) A large variety of addressing modes - typically from 5 to 20
- different modes
- »4) Variable-length instruction formats
- »5) Instructions that manipulate operands in memory (RISC in
- register)

Reduced Instruction Set Computer (RISC)

• RISC Instruction
• Only use LOAD and STORE instruction when communicating between memory and CPU
• All other instructions are executed within the registers of the CPU without referring to memory

Program to evaluate X = (A + B) * (C + D)

```
LOAD R1, A
LOAD R2, B
LOAD R3, C
LOAD R4, D
ADD R1, R1, R2
ADD R3, R3, R4
MUL R1, R1, R3
STORE X, R1
```

- Load instruction transfers the operand from memory to CPU Register.
• Add and Multiply operations are executed with data in the registers without accessing the memory.
• Result is then stored in the memory with store information.

OVERLAPPED REGISTER WINDOWS

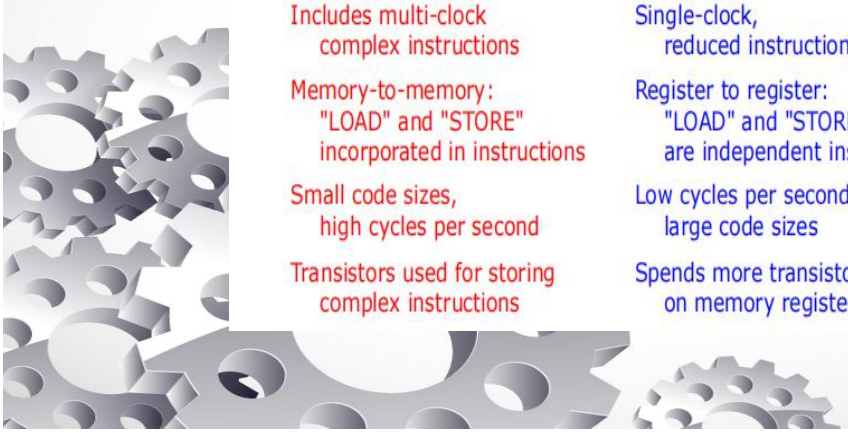
• There are three classes of registers:
– Global Registers
• Available to all functions
– Window local registers
• Variables local to the function
– Window shared registers
• Permit data to be shared without actually needing to copy it
• Only one register window is active at a time
– The active register window is indicated by a pointer
• When a function is called, a new register window is activated
– This is done by incrementing the pointer
• When a function calls a new function, the high numbered registers of the calling function window are shared with the called function as the low numbered registers in its register window
• This way the caller's high and the called function's low registers overlap and can be used to pass parameters and results

OVERLAPPED REGISTER WINDOWS

- In addition to the overlapped register windows, the processor has some number of registers, G, that are global registers
  – This is, all functions can access the global registers.
- The advantage of overlapped register windows is that the processor does not have to push registers on a stack to save values and to pass parameters when there is a function call
  – Conversely, pop the stack on a function return
- This saves
  – Accesses to memory to access the stack.
  – The cost of copying the register contents at all
- And, since function calls and returns are so common, this results in a significant savings relative to a stack-based approach

CISC versus RISC

CISC	RISC
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes
Transistors used for storing complex instructions	Spends more transistors on memory registers



CONCLUSION

- Summary :RISC is the way to make hardware simpler whereas CISC is the single instruction that handles multiple works.
- Importance :RISC architecture can be used with high-end applications like telecommunication, image processing, video processing, etc. CISC architecture can be used with low-end applications like home automation, security system, etc. It has fixed format instruction. It has variable format instruction.

THANK YOU