Microcontroller and Embedded Systems

CISC VS RISC

- **Complex Instruction Set Computer (CISC)**
- Memory in those days was expensive
- bigger program->more storage->more money
- Hence needed to ***reduce the number of instructions per program***
- Number of instructions are reduced by having ***multiple operations*** within a single instruction
- Multiple operations lead to many different kinds of instructions that access memory
- In turn making instruction length variable and fetch-decode-execute time unpredictable – making it more complex
- Thus hardware handles the complexity

- Example: x86 ISA

- Examples of CISC processors are the
  - System/360(excluding the 'scientific' Model 44),
  - VAX,
  - PDP-11,
  - Motorola 68000 family
  - Intel x86 architecture based processors.

RISC

- Original idea to reduce the ISA(inst. Set archit)

- Provide ***minimal set of instructions that could carry out all essential*** operations
- Instruction complexity is reduced by
- 1. Having ***few simple instructions*** *that are the same length*
- 2. Allowed memory access only ***with explicit load and store instructions***
- Hence each instruction performs less work but instruction execution time among different instructions is consistent
- The complexity that is removed from ISA is moved into the domain of the assembly programmer/compiler

- Examples: LC3, MIPS, PowerPC (IBM), SPARC (Sun)

- Apple iPods (custom ARM7TDMI SoC)
- Apple iPhone (Samsung ARM1176JZF)
- Palm and PocketPC PDAs and smartphones (Intel XScale family, Samsung SC32442 - ARM9)
- Nintendo Game Boy Advance (ARM7)
- Nintendo DS (ARM7, ARM9)
- Sony Network Walkman (Sony in-house ARM based chip)
- Some Nokia and Sony Ericsson mobile phones

- Consider the the program fragments:
- CISC:
```
mov ax, 10
mov bx, 5
mul bx, ax
```

- RISC:
```
mov ax, 0
mov bx, 10
mov cx, 5
Begin add ax, bx
loop Begin
```

- The total clock cycles for the CISC version might be:
- **(2 movs × 1 cycle) + (1 mul × 30 cycles) = 32 cycles**
- While the clock cycles for the RISC version is:
- **(3 movs × 1 cycle) + (5 adds × 1 cycle) + (5 loops × 1 cycle) = 13**

Comparison

RISC

- • Simple instructions, few in number (128 or less)
- • Fixed length instructions
- All operations are performed on registers, no memory op
- Few addressing modes
- Complexity in compiler
- Only **LOAD/STORE** instructions access memory
- Pipelined inst. Execution
- One inst. Per clock cycle.
- Hardwired control unit design rather than microprogram

Performance

- The CISC approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction. RISC does the opposite, reducing the cycles per instruction at the cost of the number of instructions per program.

CISC OR RISC

CISC

- Complex Instruct Set Computers.
- There are a large no. of instructions, each carrying out a different permutation of the same operation with instructions perceived to be useful by the processor's designer
- A conditional jump is usually based on status register bit.
- 8051

RISC

- Reduced Instruct Set Computers.
- The instructions are at as bare a minimum as possible to allow the user to design their own operations.
- It allows the use of simple instructions to be used for different operations.
- They have orthogonal Register set.
- A conditional jump can be based on a bit anywhere in memory
- PIC