CONTROL UNIT

- The Control Unit Provides the necessary timing and control signals to all the operations in the Microcomputer
- It controls the flow of data between the Microprocessor and Memory and Peripherals.
- The Control unit performs 2 basic tasks
  - Sequencing
  - Execution

1. SEQUENCING

- The control unit causes the processor to step through a series of micro-operations in the proper sequence, based on the program being executed.

2. EXECUTION

- The control unit causes each micro operation to be performed.

Control Signals

- For the control unit to perform its function it must have inputs that allow it to determine the state of the system and outputs that allow it to control the behavior of the system.
- Clock
  - One micro-instruction (or set of parallel micro-instructions) per clock cycle
- Instruction register
  - Op-code for current instruction
  - Determines which micro-instructions are performed
- Flags
  - State of CPU
  - Results of previous operations
- From control bus
  - Interrupts
  - Acknowledgements
- Outputs :
  - Control signals to Memory
  - Control signals to I/O
  - Control Signals within the Processor.

Outputs of Control Signal

- Within CPU
  - Cause data movement
  - Activate specific functions
- Via control bus
  - To memory
  - To I/O modules

Types of Control

1) Hardware Control
2) Software control
3) Micro-programmed Control

Hardware Control

- Control unit inputs
- Flags and control bus
  - Each bit means something
- Instruction register
  - Op-code causes different control signals for each different instruction
  - Unique logic for each op-code

  - Decoder takes encoded input and produces single output
  - *n* binary inputs and 2*n* outputs

- Clock
  - Repetitive sequence of pulses
  - Useful for measuring duration of micro-ops
  - Must be long enough to allow signal propagation
  - Different control signals at different times within instruction cycle
  - Need a counter with different control signals for t1, t2 etc.

Problems With Hard Wired Designs

- Complex sequencing & micro-operation logic
- Difficult to design and test
- Inflexible design
- Difficult to add new instructions

Micro programmed Control

- Use sequences of instructions to control complex operations
- Called micro-programming or firmware

Implementation

- All the control unit does is generate a set of control signals
- Each control signal is on or off
- Represent each control signal by a bit
- Have a control word for each micro-operation
- Have a sequence of control words for each machine code instruction
- Add an address to specify the next micro-instruction, depending on conditions
- Today's large microprocessor
  - Many instructions and associated register-level hardware
  - Many control points to be manipulated
- This results in control memory that
  - Contains a large number of words
    - co-responding to the number of instructions to be executed
  - Has a wide word width
    - Due to the large number of control points to be manipulated
- Each micro-instruction specifies single (or few) micro-operations to be performed
  - (*vertical* micro-programming)
- Each micro-instruction specifies many different micro-operations to be performed in parallel
  - (*horizontal* micro-programming

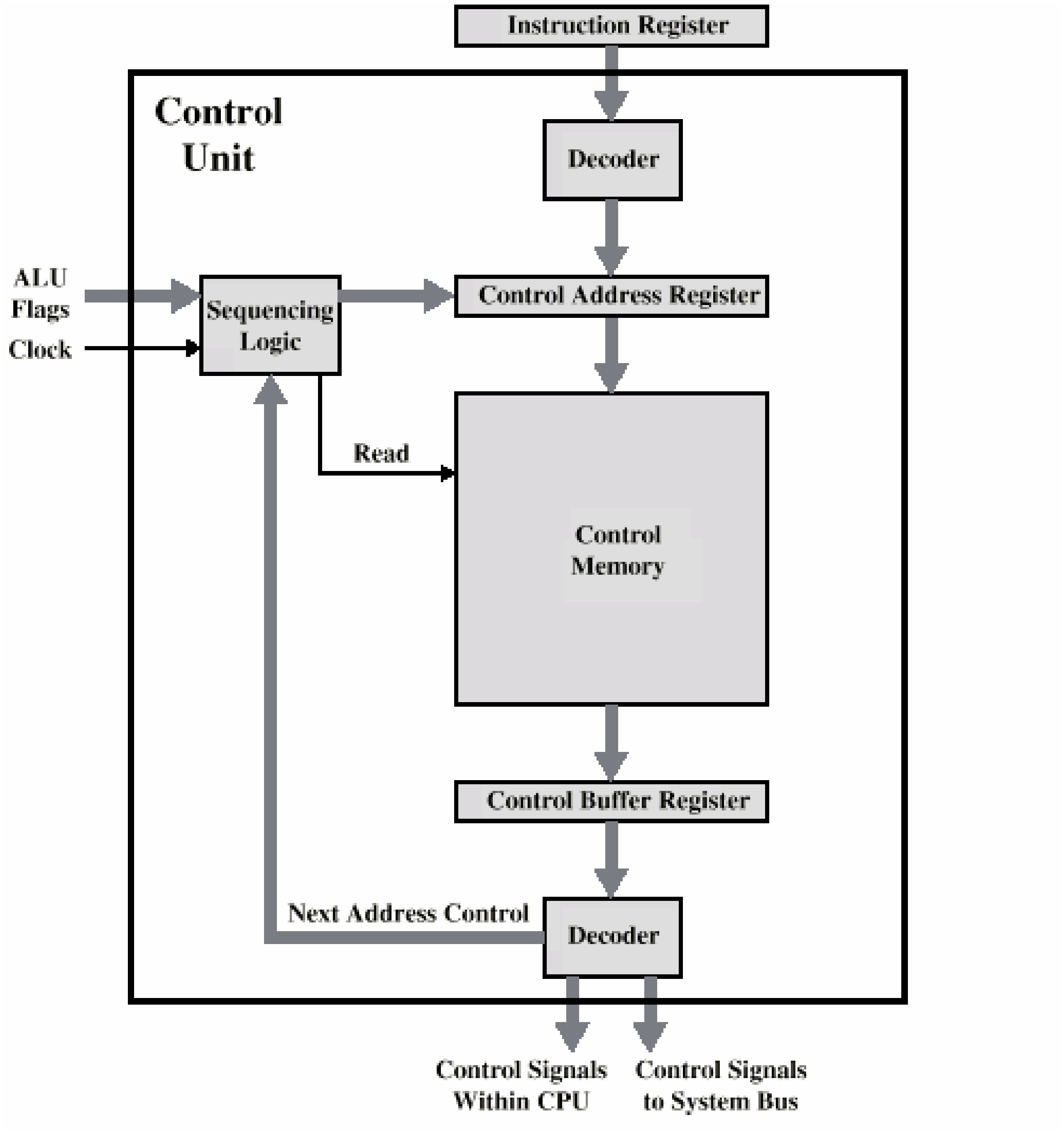
Vertical Micro-programming

- Width is narrow
- n control signals encoded into log2 n bits
- Limited ability to express parallelism
- Considerable encoding of control information requires external memory word decoder to identify the exact control line being manipulated

Advantages and Disadvantages

- Simplifies design of control unit
  - Cheaper
  - Less error-prone

- Slower



RISC processors (Reduced Instruction Set Computer)

RISC?

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.

Characteristics of RISC processor

- *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 <I.PIPELINING< I>;
- *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
- *Less number of Addressing mode*
- *Hardwired control: the operation codes are converted into control signal by using combinational circuits.*

CISC
Complex Instruction Set Computer

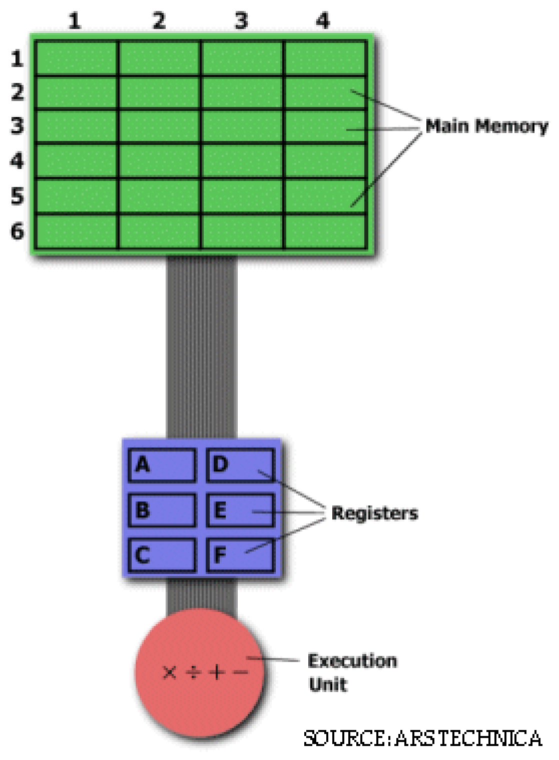
The Performance Equation

The following equation is commonly used for expressing a computer's performance ability:

$$\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$

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	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



Pipeline

Two or more instruction can be executed in parallel, these systems are known as parallel system and parallel processing.

Pipeling is an implementation technique in which two or more sub-operation overlap in execution.

There are two types of pipelining
I} Arithmetic pipeline
II) Instructional pipeline