Computer Concepts

- Mechanical Computers
  Electronic Computers
  Turing's Universal Machine

  Harvard Architecture-
  Block Diagram Discussion
  General Purpose Computers vs. 'Programmable Controllers'
  Typical Computer Instruction Set -Assembly Language (80xx)

  Communication with Program-
  PIO, INT, DMA

  Microcomputer (uP) Semi-Conductor Technology Development

- Computer History/Timeline—Available from Internet-Pictures etc.
  Moore's Law
  Future Developments

Mechanical Computers Concepts

- Ancient
  ABACUS (2,500 years ago)
  Greek/Roman- ANTIKYTHERA
- Renaissance Time
  Scientists used Tables to perform calculations more rapidly. Unfortunately, human error limited the usefulness of the Tables. Schickard(1592-1630), Pascal (Count 'sums of money'), Leibintz (More reliable device-Calculus) developed machines to calculate tables.
- 1800s
  Babbage-General Purpose 'Analytic Engine'-Not able to be built.
  Hollerth –Punch Cards— Engineer in Census Bureau—Started his own Company and obtained contract for 1890 US Census.
- 1900s
  IBM-Office Equipment
  Marchand-Mechanical Add/Mult/Divide
- Turing's 'Universal Machine' Concept
- Mechanical Computers –Limited Capability-Too difficult to design & fabricate for modern requirements.

Electronic Computer Concepts

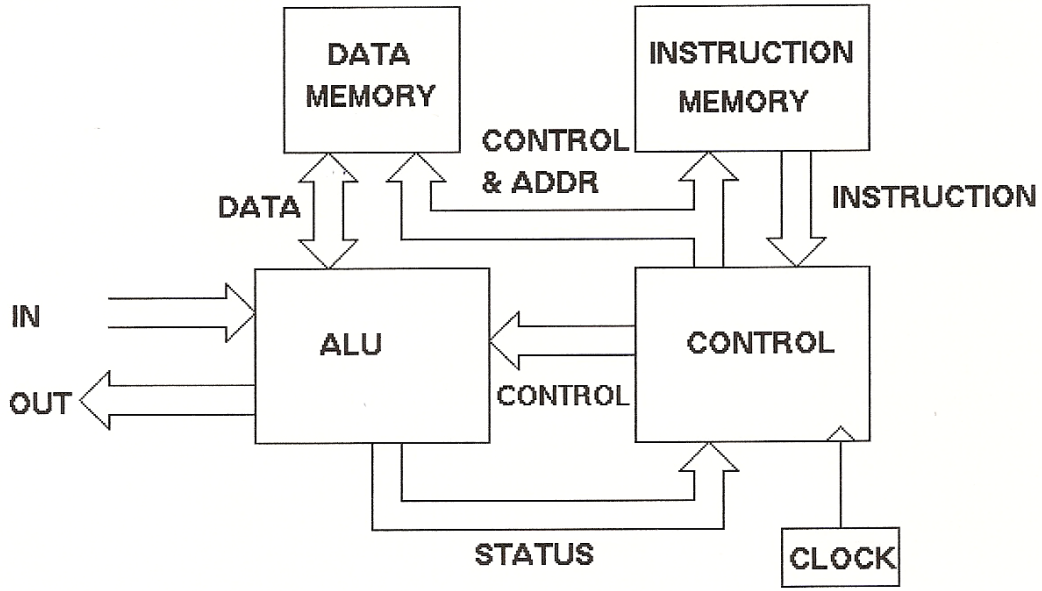
- Von Neuuman's Stored Program Concept
  He identified 'Organic' Computer sections- Arithmetic, Storage, Control, Human Interface and recognized that the Instructions could be contained in the same Memory as Data & Intermediate Results.
- Harvard Architecture
  Contains CPU, Instruction & Data Memory, Control & Timing, I/O
  Information flows In & Out of Memory via CPU, potential 'Bottleneck'.
  Criticized for 50 years, but no serious challengers- appears to be an elegant, 'natural' Configuration.

Block Diagram/ Computer Micro-program Discussion

- Computer Program Counter
  The Computer Program Counter (PC) is Initialized at ‘Power turn-on’ to Zero & ‘steps through’ the Program Instructions which are contained in the Instruction Memory (IM). Each Count forms the Address of an Instruction in the Instruction Memory. At Final (Terminal) Count, the Counter ‘Overflows’ and returns to Zero.
- Instruction Register (IR)
  Since time is required to execute Instructions, as each Instruction is ‘Read’ out to the IM, it is coupled to the IR which is provided for temporary storage of the Instructions, which are then ‘Routed’ to the Micro-Program Read Only Memory (ROM). The ‘Bit’ pattern of each instruction is different, with certain fields or sections use to form a specific address to the Micro-Program Read Only Memory (ROM). The outputs of the ROM are the Micro-Instructions employed to control the ‘Micro-Program Controller’.

HARVARD ARCHITECTURE

MICROPROCESSOR

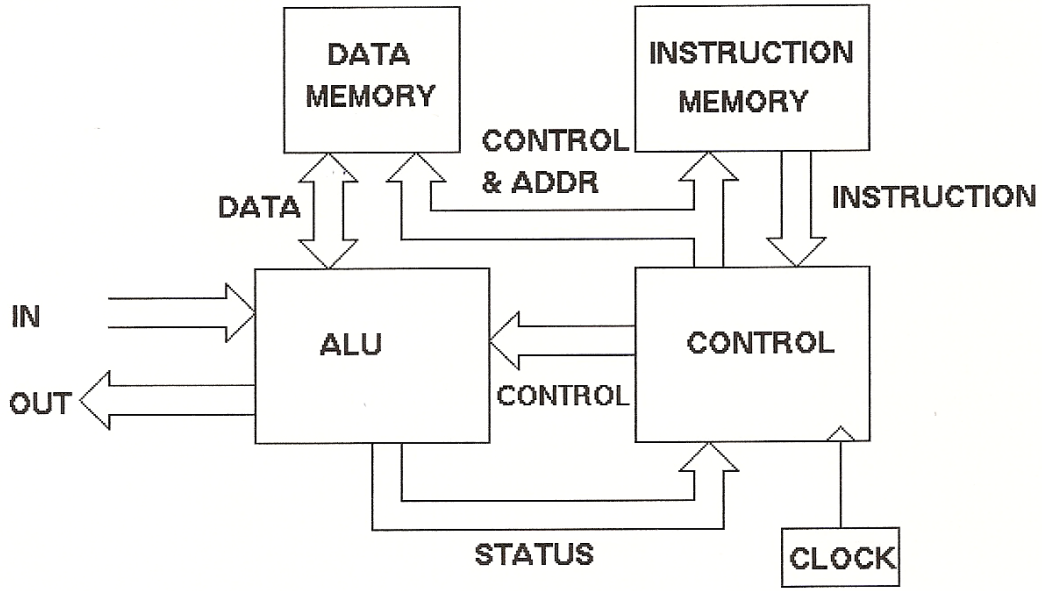


Block Diagram/ Computer Micro-program Discussion

- Computer Program Counter
  The Computer Program Counter (PC) is Initialized at ‘Power turn-on’ to Zero & ‘steps through’ the Program Instructions which are contained in the Instruction Memory (IM). Each Count forms the Address of an Instruction in the Instruction Memory. At Final (Terminal) Count, the Counter ‘Overflows’ and returns to Zero.
- Instruction Register (IR)
  Since time is required to execute Instructions, as each Instruction is ‘Read’ out to the IM, it is coupled to the IR which is provided for temporary storage of the Instructions, which are then ‘Routed’ to the Micro-Program Read Only Memory (ROM). The ‘Bit’ pattern of each instruction is different, with certain fields or sections use to form a specific address to the Micro-Program Read Only Memory (ROM). The outputs of the ROM are the Micro-Instructions employed to control the ‘Micro-Program Controller’.

HARVARD ARCHITECTURE

MICROPROCESSOR



Block Diagram/ Computer Micro-program Discussion (Cont'd)

- The Micro-Program Controller
  The Micro-Program Controller is essentially a 'Processor-within-a-Processor', whose functions are 'tailored' to the specific internal Processing Functions of the CPU etc. Thus, in the same sense that the main Computer Program is 'stepped through' by the Computer Program Counter (PC), the Micro-Program Controller 'steps through' the Micro-Instructions.
- Since there 'multiple sub-steps' in the Micro-Instructions which take place sequentially, a Computer Cycle Counter (CC Counter) is provided to 'step through' these various portions of the Micro-Program. For example, if the Instruction is Read Data from the Data Memory, and transmit it to the CPU for processing, the first part of the Instruction is to 'Fetch' the data, with the next part to 'Write' the Data to the ALU. Finally, the ALU performs the required operations. As indicated, these "steps' of the Micro-Instruction are performed sequentially.

Typical Computer Instruction Set/ CISC/RISC uPs

- Computers employ Instructions of various types to perform specific functions as related to the computer's application.
  For 'General Purpose (GP) Computers', the Instructions must be well formulated to satisfy multiple applications, and the GPs are generally programmed utilizing Higher Order Languages (HOLs).
  In addition to GPs, many applications can be satisfied by 'Programmable Controllers' which are Hardware type Electronic devices requiring very specific Instructions which were contained in a Memory Chip, as a separate part of the Programmable Controller.
  These instructions employed Mnemonics to represent the processing operations that related to Hardware commands (Turn-on/Off Switch etc.) and are written at the Assembly Language Level.
  The development of Microprocessors containing the Instruction Memory within the Silicon Chip, required formulation of Instruction types which would provide 'Programming Flexibility', but not compromise Computer performance.
  INTEL developed the 80XX family of 147 instructions called the Complex Instruction Set Computer (CISC) Instructions. These were well received by Programmers, but were considered to be 'Performance Limited', since many of the Instructions contained 'sub-instructions' each requiring Machine Cycles (time) to be executed. This led to the development of Reduced Instruction Set Computer (RISC) Computers, which completed each instruction in a single Machine Cycle, but required multiple cycles to perform the equivalent CISC Instruction.

Typical Computer Instruction Set/ CISC/RISC uPs (Cont'd)

- While some RISC uPs were made, INTEL resisted this development because of its 'backward/forward' software compatibility concerns. Improved Semi-conductor technology eventually led to the PENTIUM family which 'transparently' combined RISC instructions into CISC, while maintaining performance and S/W compatibility.
  Improved Software Development Tools (Compilers etc.) eventually permitted programmers to readily 'Code' for both type Microprocessors.
  Note that the Assembly Level 80XX Instructions are extremely tedious and difficult to use, and may require Engineering Oriented Programmers.
  Complex Electronic Systems (Navigation, Control, Communication etc.) programs are usually required to be written in HOL, but can contain Assembly Language Instructions for critical timing operations.
- Typical UP Instruction Set.
  8086 Family Instructions:
  CLR- Clear, Carry
  CMP- Compare
  SAR-Shift Arithmetic Right.

Communication With the Computer Program

- Basic computer operation involves repetitive execution of a specific program.
  In order to provide controls which facilitate operation with the outside world various functions are provided:
- Programmed Input Output Control (PIO)--
  Periodically data from the External World can flow In or Out of the Computer, but the action does not change the repetitive program operation.
- Interrupt/Acknowledge (Int/Ack)--An Interrupt Service Routine (ISR), which is not part of the normal program execution, allows data from the External World to flow In or Out of the Computer Program.
- Direct Memory Access (DMA)– The Computer Program is Halted, and the I/O can send Data Directly in/out of the Computer Memory (requires additional circuitry).

Micro-Computer Semi-conductor Technology Development/ Genealogy

- Transistor Feature Size has been reduced by a Factor of approximately 300,000 in 35 years !! Follows Moore's Law closely.
- Date Mnf'r Model # Transistors
  1971 INTEL 8008 3,300
  1979 INTEL 8088 29,020
  1981 MOT 68000 68,000
  1985 INTEL 80386 275,000
  1987 MOT 68030 273,000 1993 INTEL Pentium 3.1 Million
  2000 INTEL Pentium 4 42 Million
  2005 INTEL Dual Proc 1 Billion

Computer History Museum---Timeline

- This Museum is available on the Internet, and contains pictures and descriptions of Computer developments, including:
  Companies
  Components
  Computers
  Graphics and Games
  Networking
  Robots and Artificial Intelligence
  Storage

Future Developments

- Moore's Law postulates that Circuit Density doubles every 2 years, or in 45 years 2^22.5 or 5.9 Million Times!!
- Fundamental Physical Limitations on shrinking Planar transistor size are being approached--presently 45 Nanometers (Human Hair is 80,000 Nanometers).
  This is equivalent to packing 1 Million Transistors in the area of a PERIOD!
- 'FINFETS' are being developed--so named because they appear to be similar to 'Fish-fins' protruding from the Planar Surface. Smaller area and improved 'leakage' characteristics are keys to this technology under development by both IBM and INTEL.
- In order to further miniaturize and achieve improvements in speed and performance of Microprocessors, approaches are being considered beyond Semi-Conductor technology. These involve:
  Magnetic, Quantum, and Nanomechanical Switching principles.
- Note that Improved Performance is being achieved by use of Multiple Processors (Dual/Quad etc.).
  While Multiple Processors are effective, Programming Synchronization remains a major consideration and is an area requiring significant development.