CPU Organization:
ALU & CU Organization

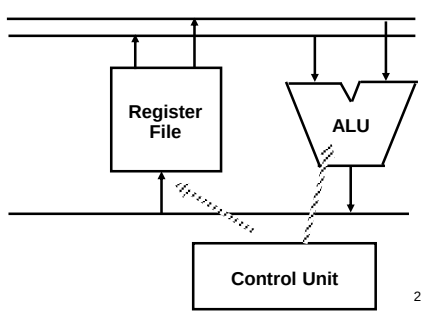
Major Components of CPU

- Storage Components:
- Registers
- Flip-flops

- Execution (Processing) Components:
- Arithmetic Logic Unit (ALU): Arithmetic calculations, Logical computations, Shifts/Rotates

- Transfer Components:
- Bus

- Control Components:
- Control Unit



How does the CPU works?

- The processor performs three basic functions;
- **Accept input**;
- This is the process by which external data is received into the computer. This could either be running a program or getting keyboard response. The computer system analyzes the input data.
- **Process data**;
- Process describes the converting input into output and is generally guided by a program.
- **Provide output**;
- Output is the process by which the CPU sends data to devices such as monitor, printer, etc. output takes the results of the processing and sends them to be stored in memory or printed or displayed.

Basic Processing Cycle

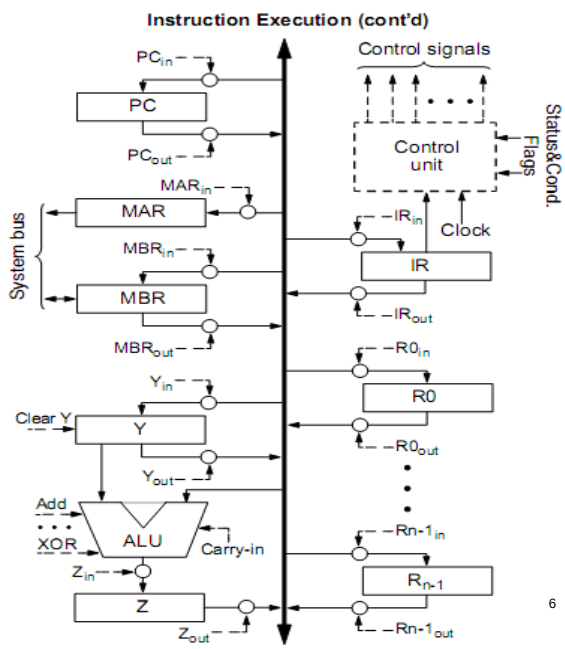
- **Fetch instruction:** The processor reads an instruction from memory (register ,cache, main memory).
- **Interpret instruction**: The instruction is decoded to determine what action is required.
- **Fetch data:** The execution of an instruction may require reading data from memory or an I/O module.
- **Process data:** The execution of an instruction may require performing some arithmetic or logical operation on data.
- **Write data:** The results of an execution may require writing data to memory on I/O module.

Instruction Execution

- The CPU executes a sequence of instructions.
- The execution of an instruction is organized as an instruction cycle: it is performed in several steps;
- Each step is executed as a set of several micro operations.
- All the micro-operation are controlled by CU by performing two basic tasks:
  - Sequencing: It causes the processor to step through the series of micro-operation in proper sequence, based on program being executed.
  - Execution: It causes each micro-operation to be performed.

Instruction Execution

- The task performed by any micro operation falls in one of the following categories:
- Transfer data from one register to another;
- Transfer data from a register to an external interface (system bus)
- Transfer data from an external interface to a register;
- Perform an arithmetic or logic operation, using registers for input and output.



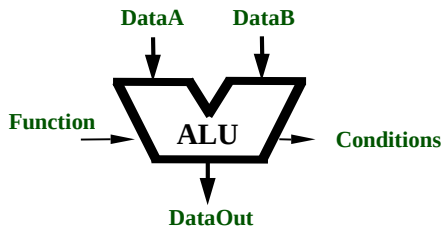
Arithmetic Logic Unit

Arithmetic Logic Unit

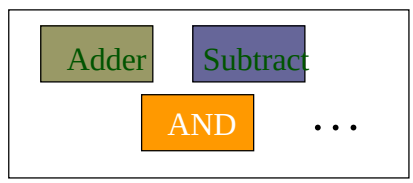
- Main computation unit in most computer systems
- ALUs perform a variety of different functions
  - Add, subtract, OR, AND…
- Example: ALU chip (74LS382)
  - Has data and control inputs
- Individual chips can be chained together to make larger ALUs
- ALUs are important parts of data paths
  - ROMs often are used in the control path
- Build a data and control path

Typical Schematic Symbol of an ALU

- Arithmetic logic unit contain :
- Two multi-bit data inputs: data A and data B
- Function indicates action (e.g. add, subtract, OR…)
- Data Out is same bit width as multi-bit inputs (Data A and Data B)
  - Conditions indicate special conditions of arithmetic activity (e.g. overflow, carry-in, carry-out,).

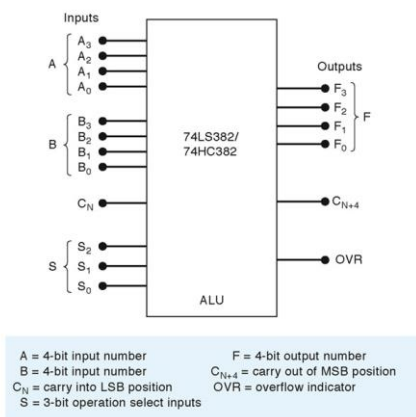


Think of ALU as a number of other arithmetic and logic blocks in a single box! Function selects the block



ALU Integrated Circuit

- Integrated circuit components: Examine the functionality of this ALU chip



A = 4-bit input number
B = 4-bit input number
C_N = carry into LSB position
S = 3-bit operation select inputs
F = 4-bit output number
C_{N+4} = carry out of MSB position
OVR = overflow indicator

(a)

Function Table

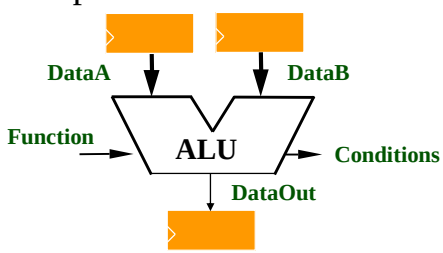
S_2	S_1	S_0	Operation	Comments
0	0	0	CLEAR	$F_3F_2F_1F_0 = 0000$
0	0	1	B minus A	Needs $C_N = 1$
0	1	0	A minus B	Needs $C_N = 1$
0	1	1	A plus B	Needs $C_N = 0$
1	0	0	A ⊕ B	Exclusive-OR
1	0	1	A + B	OR
1	1	0	AB	AND
1	1	1	PRESET	$F_3F_2F_1F_0 = 1111$

Notes: S inputs select operation.
OVR = 1 for signed-number overflow.

(b)

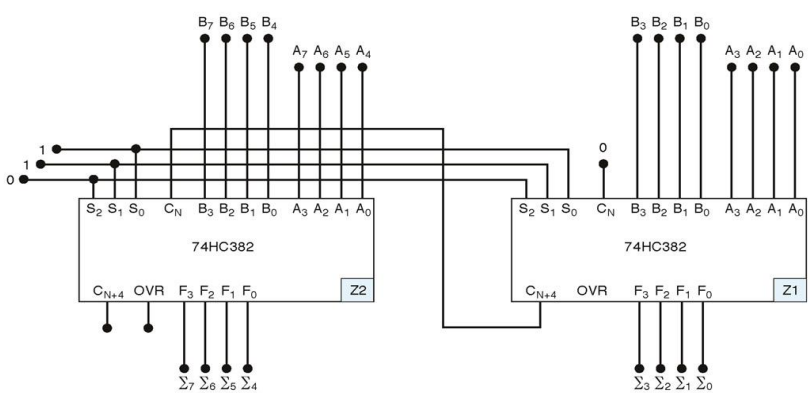
Example

- Determine the 74HC382 ALU outputs for the following inputs: $S_2S_1S_0$=010, $A_3A_2A_1A_0$=0100, $B_3B_2B_1B_0$=0001, and C_N=1.
  - Function code indicates subtract
  - 0100 – 0001 = 0011
- Change the select code to 101 and repeat.
  - Function code indicates OR
  - 0100 OR 0001 = 0101



Expanding the ALU

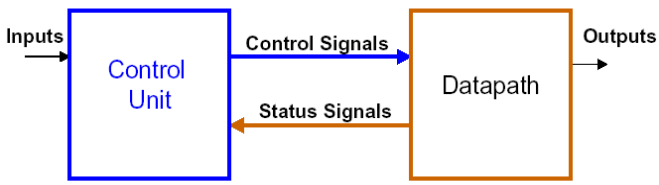
- Multi-bit ALU created by connecting carry output of low-order chip to carry in of high order



Notes: Z1 adds lower-order bits.
Z2 adds higher-order bits.
Σ_7–Σ_0 = 8-bit sum.
OVR of Z2 is 8-bit overflow indicator.

Interaction Between Data and Control Units

- Processor can be partitioned in a **Datapath** and **Control Unit**



Component of the processor that commands the datapath, memory, I/O devices according to the instructions of the memory.

Components of the processor that perform arithmetic operation and hold data, include

- **Registers**
- **MUXes, ALU, Shifters, Comb. Circuits and buses**
- **Implements microoperations (under control of the control unit)**

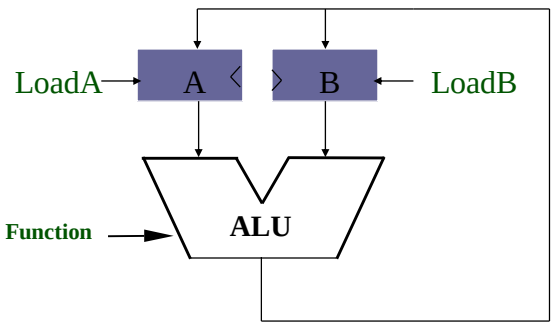
Computation in a Typical Computer

- Control logic often implemented as a finite state machine (including ROMs)
- Datapath contains blocks such as ALUs, registers, tri-state buffers, and RAMs
- In a processor chip often a 5 to 1 ratio of datapath to control logic

Using a Datapath

- **Consider the following computation steps**
  1. ADD A, B and put result in A
  2. Subtract A, B and put result in B
  3. OR A, B put result in A
  - Repeat starting from step 1

Determine values for Function, LoadA, LoadB

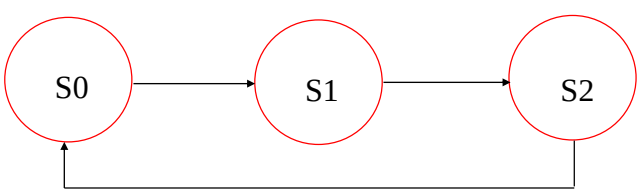


Modeling Control as a State Machine

- **Consider the following computation steps**
  1. ADD A, B and put result in A
  2. Subtract A, B and put result in B
  3. OR A, B put result in A
  - Repeat starting from step 1

Determine values for Function, LoadA, LoadB

Model control as a state machine.
Determine control outputs for each state



Modeling Control as a State Machine

- **Consider the following computation steps**
  1. **ADD A, B and put result in A**
  2. **Subtract A, B and put result in B**
  3. **OR A, B put result in A**
  - **Repeat starting from step 1**

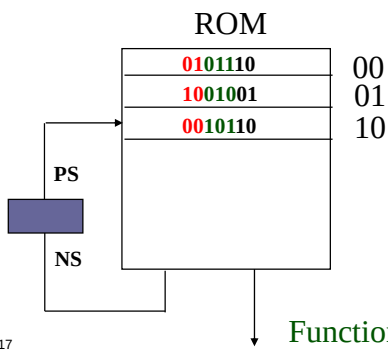
States
S0 = 00
S1 = 01
S2 = 10

Present State	Next State	Function	LoadA	LoadB
00	01	011	1	0
01	10	010	0	1
10	00	101	1	0

We know how to implement this using an SOP.
Can we use a ROM?

ROM Implementation of State Machine

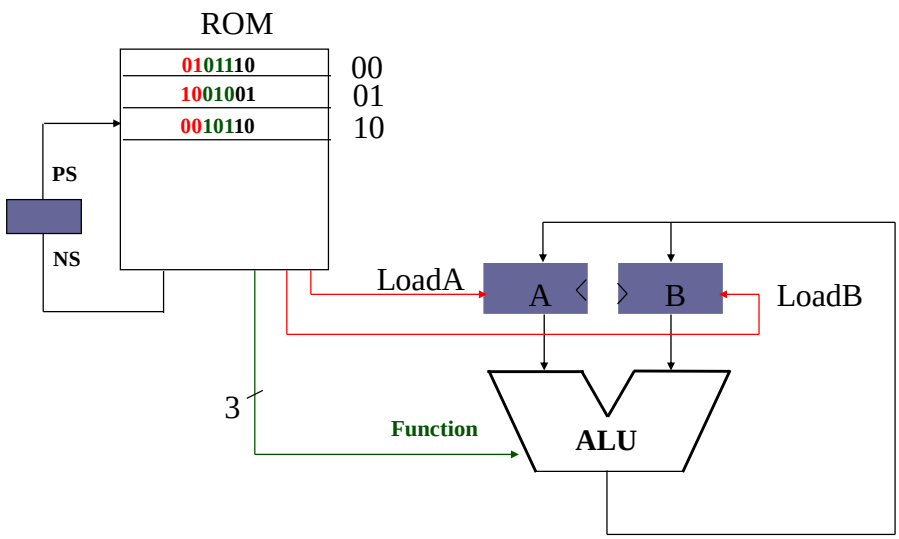
Present State	Next State	Function	LoadA	LoadB
00	01	011	1	0
01	10	010	0	1
10	00	101	1	0



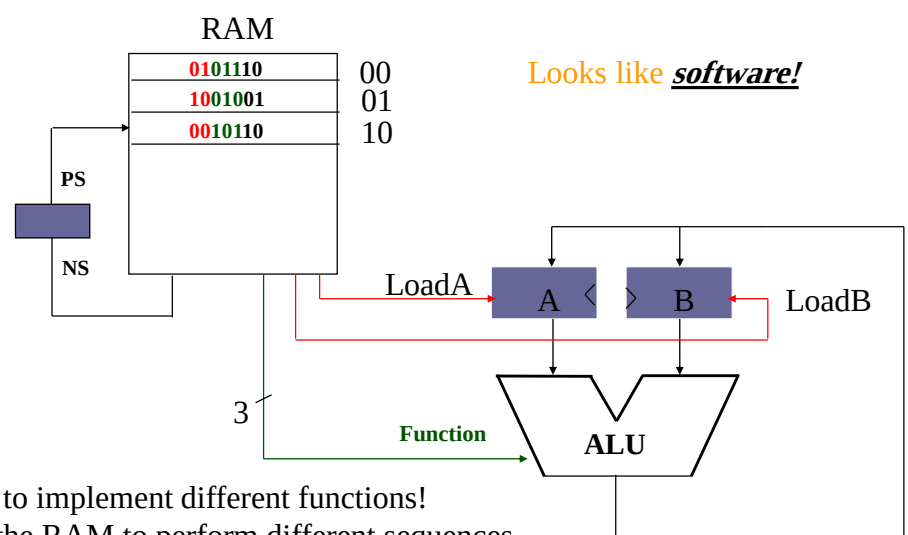
States
S0 = 00
S1 = 01
S2 = 10

Note: No minimization!
One line in ROM for each state

Putting the Control and Datapath Together



What if we replaced the ROM with RAM?



Looks like ***software!***

Possible to implement different functions!

Program the RAM to perform different sequences

Control Unit

Control unit

- The function of control unit is to generate timing and control signals to all operations in the computer.
- Control Unit is “the brain within the brain”.
- It controls the flow of data between the processor and memory and peripherals.
- The control unit must communicate with both the arithmetic logic unit (ALU) and main memory.
- The control unit instructs the arithmetic logic unit that which logical or arithmetic operation is to be performed.
- The control unit coordinates the activities of the arithmetic/logic units as well as all peripherals and auxiliary storage devices linked to the computer.

Control Unit Organization

- The basic task of the control unit : For each instruction the control unit causes the CPU to go through a sequence of control steps; in each control step the control unit issues a set of signals
- The control unit is driven by the processor clock.
- The signals to be generated at a certain moment depend on:
  - the actual step to be executed;
  - the condition and status flags of the processor;
  - the actual instruction executed;
  - external signals received on the system bus (e.g. interrupt signal)

Control Unit Organization

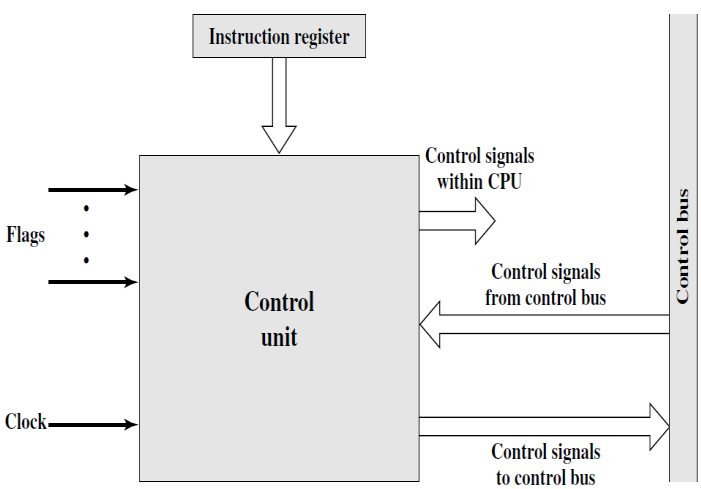


Figure Block Diagram of the Control Unit

Control Signal Sources

- Clock
  - It helps to synchronize the operation. It causes one micro-operation to be performed for each clock pulse
- 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 / Bus Requests
  - Acknowledgements

Control Signal Outputs

- Within Processor
  - Cause data movement
  - Activate specific functions
- Via Main Bus
  - To memory
  - To I/O modules

Types

- There are two design approach for CU:
  - Hardwired approach
  - Micro-programming approach

Hardwired Approach

- The control signals are generated by the help of the hardware.
- Hardwired control units are implemented through use of sequential logic units or circuits like gates , fliflops , decoders multiplexers and other logic buildings blocks.
- Hardwired control units are generally faster than micro-programmed designs.
- This architecture is preferred in reduced instruction set computers (RISC) as they use a simpler instruction set.

Advantages

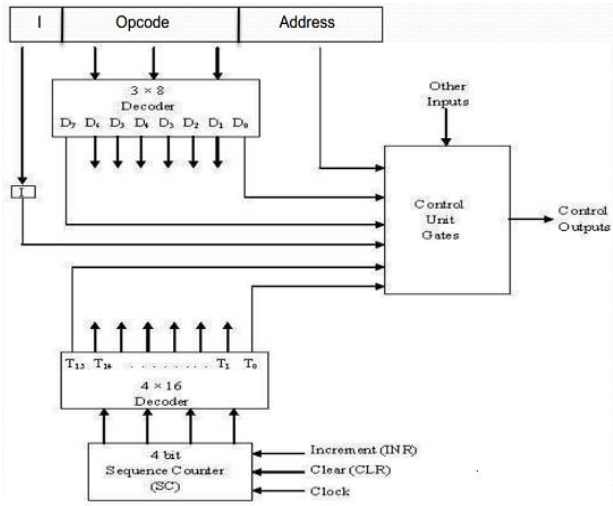
- Hardwired Control Unit is fast because control signals are generated by combinational circuits.
- The delay in generation of control signals depends upon the number of gates.
- The performances is high as compared to micro-programmed control unit.

Disadvantage

- The control signals required by the CPU will be more complex
- Modifications in control signal are very difficult. That means it requires rearranging of wires in the hardware circuit.
- It is difficult to correct mistake in original design or adding new features in existing design of control unit.

Hardwired Architecture

- Control unit consist of a:
  - Instruction Register
  - Number of Control Logic Gates,
  - Two Decoders
  - 4-bit Sequence Counter



- An instruction read from memory is placed in the instruction register (IR).
- The instruction register is divided into three parts: the I bit, operation code, and address part.
- First 12-bits (0-11) to specify an address, next 3-bits specify the operation code (opcode) field of the instruction and last left most bit specify the addressing mode I.
  - I = 0 for direct address
  - I = 1 for indirect address

- First 12-bits (0-11) are applied to the control logic gates.
- The operation code bits (12 – 14) are decoded with a 3 x 8 decoder.
- The eight outputs (D0 through D7) from a decoder goes to the control logic gates to perform specific operation.
- Last bit 15 is transferred to a I flip-flop designated by symbol I.
- The 4-bit sequence counter SC can count in binary from 0 through 15.
- The counter output is decoded into 16 timing pulses T0 through T15.
- The sequence counter can be incremented by INR input or clear by CLR input synchronously.

Micro programmed Approach

- Micro-programs were organized as a sequence of *microinstructions* and stored in special control memory.
- A micro-programmed control unit is implemented using programming approach. A sequence of micro operations are carried out by executing a program consisting of micro-instructions.
- Micro-program, consisting of micro-instructions is stored in the control memory of the control unit.
- Execution of a micro-instruction is responsible for generation of a set of control signals.

Definitions

- Micro-Programs: Microprogramming is the concept for generating control signals using programs. These programs are called micro - programs.
- Micro-Instructions: The instructions that make micro-program are called micro-instructions.
- Micro-Code: Micro-program is a group of microinstructions. The micro-program can also be termed as micro-code.
- Control Memory: Micro-programs are stored in the read only memory (ROM). That memory is called control memory.

Advantage

- The design of micro-program control unit is less complex because micro-programs are implemented using software routines.
- The micro-programmed control unit is more flexible because design modifications, correction and enhancement is easily possible.
- The fault can be easily diagnosed in the micro-program control unit using diagnostics tools by maintaining the contents of flags, registers and counters.

Disadvantage

- The micro-program control unit is slower than hardwired control unit. That means to execute an instruction in micro-program control unit requires more time.
- The design duration of micro-program control unit is more than hardwired control unit for smaller CPU.

Micro programmed Architecture

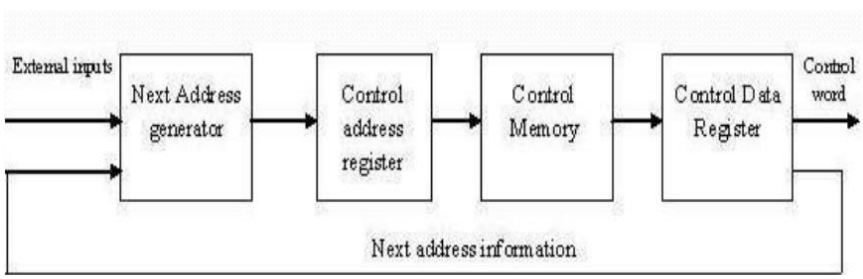
Control unit consist of a :

Next address generator

Control address register

Control memory

Control data register



Micro programmed Architecture

- The address of micro-instruction that is to be executed is stored in the control address register (CAR).
- Micro-instruction corresponding to the address stored in CAR is fetched from control memory and is stored in the control data register (CDR).
- This micro-instruction contains control word to execute one or more micro-operations.
- After the execution of all micro-operations of micro-instruction, the address of next micro-instruction is located.

Comparison

Attributes	Hardwired Control	Microprogramming Control
Speed	Fast	Slow
Cost of Implementation	More	Cheaper
Flexibility	Difficult to modify	Flexible
Ability to handle complex instruction	Difficult	Easier
Decoding	Complex	Easy
Application	RISC	CISC
Instruction Set Size	Small	Large
Control Memory	Absent	Present