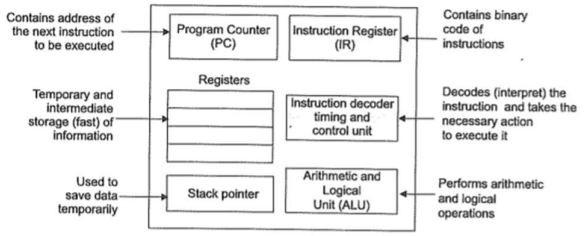
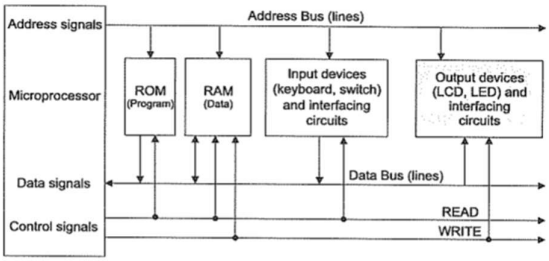
General block diagram of a microprocessor (CPU)

A microprocessor is made up of a series of interconnected functional units all controlled by a central timing and control unit that behaves much like a finite state machine.



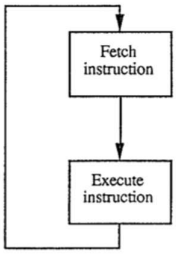
Components of a microcomputer

A microcomputer is built up from a microprocessor, some memory and several I/O devices. These all communicate via a parallel bus which itself consists of an **Address** bus, a **Data** bus and a **Control** bus.



How does a microprocessor work?

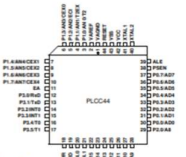
The flow chart to the right shows, in its simplest form, what a microprocessor does.

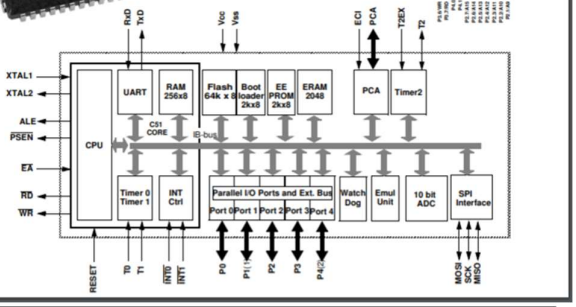


The microprocessor fetches an instruction (we will ignore how it does this for now). This instruction tells it what it must do. For example, it might be to perform an exclusive-OR operation, increment a counter, or shift-right a register. Having executed this instruction the microprocessor fetches a new instruction, which it executes, and so on.

The AT89C51AC3 micro controller

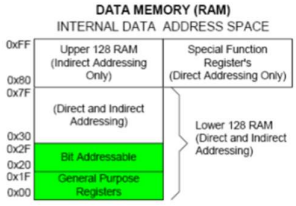
Block diagram and pin configuration





8051 Data Memory (RAM)

- Internal Data Memory space is divided into three sections
  - Lower 128
  - Upper 128
  - Special function register (SFR)
- There are 384 bytes of memory space physically, though the Upper 128 and SFRs share the same addresses from location 80H to FFH.
- Appropriate instructions should be used to access each memory block

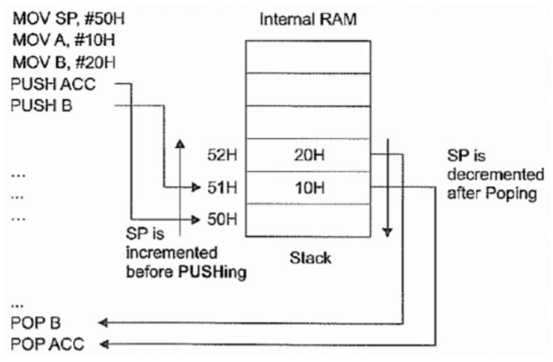


Assembler Directives—Introduction

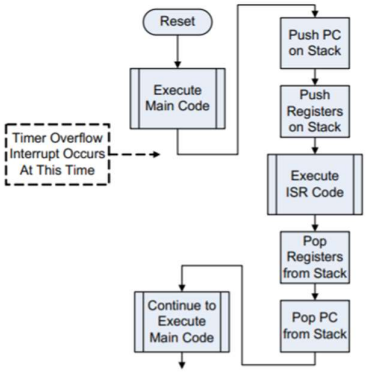
- Assembler Directives are **not** assembly language instructions as they do not generate any machine code
  - They are special codes placed in the assembly language program to instruct the assembler to perform a particular task or function
  - They can be used to define symbol values, reserve and initialize storage space for variables and control the placement of the program code
- Assembler directives are typically specific to a particular assembler. We will be using the **Keil A51 Assembler** in this course.
- The ASM directives are grouped into the following categories:
  - Segment control
  - Address control
  - Symbol definition
  - Memory initialization/reservation

The system stack

Explain how contents of Accumulator and B registers can be stored and retrieved from the stack.



Execution Flow



Microprocessors

How does it work

- The microprocessor fetches an instruction which tells it what it must do, e.g., an exclusive-OR operation, increment a counter, or shift-right a register. Having executed this instruction the microprocessor fetches a new instruction, which it executes, and so on.
- Fetching instructions
    - To fetch an instruction, the controller places the address contained in the program counter onto the address bus. A read signal is then set in the control bus and the memory responds by placing the data (instruction) onto the data bus. The microprocessor then loads this instruction into the instruction register.

Executing instructions

- To execute the instruction, the microprocessor must first decode the instruction and then perform the appropriate operations. For example, the instruction may be to increment the number contained in the accumulator.
    - For the 8051 series microprocessor this instruction, called a “machine code” has the hexadecimal code 04. This has a corresponding abbreviated description, called a Mnemonic, “INCA”.

AT89C51micro controller core

- ALU: Arithmetic Logic Unit performs all arithmetic and logical (AND, OR, NOT, EXCLUSIVE-OR and rotating) operations on 8 bit data. The ALU also updates information about the nature of the result in the flag register (PSW or Program State Word, an 8 bit register also referred as flag register).
- PC: Program Counter is a 16-bit register. It contains the memory address of the next instruction to be executed. As the CPU fetches the op- code (instruction byte) from the program memory, the PC is incremented automatically to point to the next instruction.
- Timing/Control and Instruction unit: Responsible for decoding and managing the execution of instructions.
- SFRs: The following registers are called Special Function Registers, they are all the other registers in the core and they are addressable.

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
ACC	E0h	Accumulator	–	–	–	–	–	–	–	–
B	F0h	B Register	–	–	–	–	–	–	–	–
PSW	D0h	Program Status Word	CY	AC	F0	RS1	RS0	OV	F1	P
SP	81h	Stack Pointer	–	–	–	–	–	–	–	–

- A: Accumulator; the most useful and versatile register because it is used in all arithmetic operations, majority of logical operations, and all data transfer between the 8051 and any external memory.
- B: used along with A in multiplication operations to hold one of the operands and to store higher order byte of the result. It is also used in division operation to hold divisor and to store remainder of the result.
- DPTR: Data Pointer is a 16-bit register. It is used to point to a data byte in external data (RAM) or program (ROM) memory. It can be used as a single 16 bit register or accessed as two separate 8 bit registers named DPL and DPH.
- SP: Stack Pointer. The stack is a special area of memory allocated for temporary storage and is often used to store the status of the core when the processor is switched to a different task, either due to an interrupt or the jumping to a subroutine. the Stack pointer always points to the top of the stack and is used to access data from there. It is an 8 bit register. The data is stored on to the stack using PUSH and CALL instructions and retrieved using POP and RET instructions.
- I/O port registers (latches): P0, P1, P2 and P3. The 8051 has four 8 bit ports named as P0, P1 P2 and P3, each can be used as an input or output or both. All ports are byte as well as bit addressable.
- Peripheral control and status registers: These are a series of special registers for the control and monitoring of the peripheral units such as a serial UART interface. The interrupt control and status registers are included within this group and are used to enable/disable interrupts as well as setting the interrupt priorities
- Memory: The 8051 family has separate on-chip program and data memory, this is known as a Harvard Architecture and results in faster execution as both program and data memory can be accessed at the same time. The alternative is the simpler but slower Von Neumann Architecture where the program and data are stored in the same memory.

8051 Instruction Set

- A computer instruction is made up of an operation code (opcode) followed by either zero, one or two bytes of operands. The op-code identifies the type of operation to be performed while the operands identify the source and destination of the data
- If the instruction is associated with more than one operand, the format is always:
    - Instruction Destination, Source

Special function registers

- SFRs provide control and data exchange with the microcontroller's resources and peripheral (not all are shown here, only the ones common to the core)
- Registers which have their byte addresses ending with 0H or 8H are byte (or bit) addressable. Some registers are not bit addressable – these include the stack pointer (SP) and data pointer register (DPTR)

Addressing Modes	Instruction
Register	MOV A, B
Direct	MOV 30H,A
Indirect	ADD A,@R0
Immediate Constant	ADD A,#80H
Relative*	SJMP AHEAD
Absolute*	AJMP BACK
Long*	LJMP FAR_AHEAD
Indexed	MOVC A,@A+PC

Addressing Modes

- Eight modes of addressing are available with the 8051. The different addressing modes determine how the operand byte is selected

Indirect addressing

- Uses a pointer to hold the effective address of the operand; only registers R0, R1 and DPTR can be used as the pointer registers. "@" is used as the pointer dereference operator
- R0 and R1 registers can hold an 8-bit address, whereas DPTR can hold a 16-bit address

Relative Addressing

- This mode of addressing is used with some type of jump instructions, like SJMP (short jump) and conditional jumps like JNZ. These instructions transfer control from one part of a program to another
- The destination address must be within -128 and +127 bytes from the current instruction address because an 8-bit offset is used

Indexed Addressing

- A 16-bit register (data pointer) holds the base address and the accumulator holds an 8-bit displacement or index value. The sum of these two registers forms the effective address for a JMP or MOVC instruction

8051 Operations tables

Mnemonic	Description
MOV @Ri, direct	[@Ri] = [direct]
MOV @Ri, #data	[@Ri] = immediate data
MOV DPTR, #data 16	[DPTR] = immediate data
MOVC A,@A+DPTR	A = Code byte from [@A+DPTR]
MOVC A,@A+PC	A = Code byte from [@A+PC]
MOVX A,@Ri	A = Data byte from external ram [@Ri]
MOVX A,@DPTR	A = Data byte from external ram [@DPTR]
MOVX @Ri, A	External[@Ri] = A
MOVX @DPTR,A	External[@DPTR] = A
PUSH direct	Push into stack
POP direct	Pop from stack
XCH A,Rn	A = [Rn], [Rn] = A
XCH A, direct	A = [direct], [direct] = A
XCH A, @Ri	A = [@Rn], [@Rn] = A
XCHD A,@Ri	Exchange low order digits

Mnemonic	Description
CLR C	Clear C
CLR bit	Clear direct bit
SETB C	Set C
SETB bit	Set direct bit
CPL C	Complement c
CPL bit	Complement direct bit
ANL C,bit	AND bit with C
ANL C,/bit	AND NOT bit with C
ORL C,bit	OR bit with C
ORL C,/bit	OR NOT bit with C
MOV C,bit	MOV bit to C
MOV bit,C	MOV C to bit
JC rel	Jump if C set
JNC rel	Jump if C not set
JB bit,rel	Jump if specified bit set
JNB bit,rel	Jump if specified bit not set
JBC bit,rel	if specified bit set then clear it and jump

Mnemonic	Description
ANL A, Rn	A = A & [Rn]
ANL A, direct	A = A & [direct memory]
ANL A,@Ri	A = A & [memory pointed to by Ri]
ANL A,#data	A= A & immediate data
ANL direct,A	[direct] = [direct] & A
ANL direct,#data	[direct] = [direct] & immediate data
ORL A, Rn	A = A OR [Rn]
ORL A, direct	A = A OR [direct]
ORL A,@Ri	A = A OR [@RI]
ORL A,#data	A = A OR immediate data
ORL direct,A	[direct] = [direct] OR A
ORL direct,#data	[direct] = [direct] OR immediate data
XRL A, Rn	A = A XOR [Rn]
XRL A, direct	A = A XOR [direct memory]
XRL A,@Ri	A = A XOR [@Ri]
XRL A,#data	A = A XOR immediate data
XRL direct,A	[direct] = [direct] XOR A
XRL direct,#data	[direct] = [direct] XOR immediate data
CLR A	Clear A
CPL A	Complement A
RL A	Rotate A left
RLC A	Rotate A left (through C)
RR A	Rotate A right
RRC A	Rotate A right (through C)
SWAP A	Swap nibbles

Timer Operations

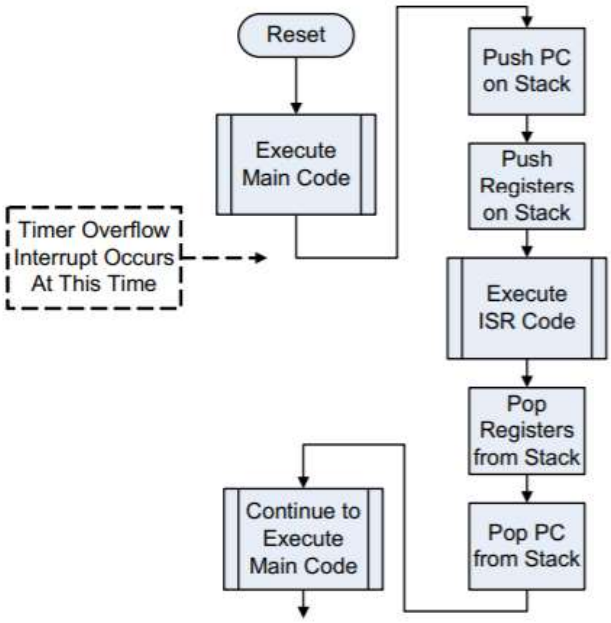
- ¨ In interval timing applications, a timer is programmed to overflow at a regular interval and the following:
  - Set the timer overflow flag or generate an interrupt
- The flag or interrupt is used to synchronise the program to perform an action such as checking the state of inputs and updating the DAC output or ADC input
- AT89C51AC3 has multiple timer/counters. We will only consider Timers 0 and 1 and Timer Mode 1
  - The timer counts up from the preloaded value THx, TLx and when it reached the max count value it sets the flag TFx and generates an interrupt (if enabled). We can "poll" the TFx flag to determine when the max count has been reached.

System stack

- When calling a subroutine or servicing an interrupt it is necessary to preserve the return address and the contents of other registers. The stack is a special area of data memory for temporary storage. It is a LIFO (Last In First Out) structure with a special Stack Pointer (SP) register is used to store the address of the top of the stack. The reset value of the SP is 07H, which is just after the first register bank.
- The maximum available memory available to 8051 stack is 128 bytes, so we must be careful of overflows
- With a call to a subroutine, for example "ACALL". The operation will cause the PC to increase by 2, then it pushes the 16- bit PC value onto the stack (low order byte first) and increments the stack pointer twice.
- At the end of the subroutine the RET instruction pops the high byte and low byte address of PC from the stack and decrements the SP by 2. The execution of the instruction will result in the program to resume from the location just after the "call" instruction

Interrupt

- An interrupt is the occurrence of a condition that causes a temporary suspension of a program while the condition is serviced by another (sub) program
- Interrupts are important because they allow a system to respond asynchronously to an event and deal with the event while in the middle of performing another task
- When an interrupt occurs, the main program temporarily suspends execution and branches to the ISR
- The ISR executes, performs the desired operation, and terminates with a "return from interrupt" (RETI) instruction – different from normal RET instructions



Priority

- Each interrupt source can be individually programmed to one of two priority levels, low or high
- The two priority levels allow an ISR to be interrupted by an interrupt of higher priority than the current one being serviced

Assembly Directives

- Special codes placed in the assembly language program to instruct the assembler to perform a particular task or function. They can be used to define symbol values, reserve and initialize storage space for variables and control the placement of the program code

Absolute segment

- Absolute segment means a fixed memory segment – they are created by CSEG and DSEG directives.

Address control – ORG and END

- ORG directive is used to set the location counter
- END directive indicates the end of the source file; informs the assembler where program ends

Symbol definitions – EQU and SET

- Like #define, assigns a symbolic name to an expression or a register

Analogue to Digital and Digital to Analog Conversion.

Digital to Analog Conversion Process

- Digital code is converted to a voltage or current proportional to the digital code; bipolar DAC's use 2's compliment to represent negative voltages
- Voltage reference used to determine the full scale O/P - "Pseudo analogue" as O/P cannot take on continuous values.
- Analog O/P = K x digital I/P where K is resolution of scale

Resolution

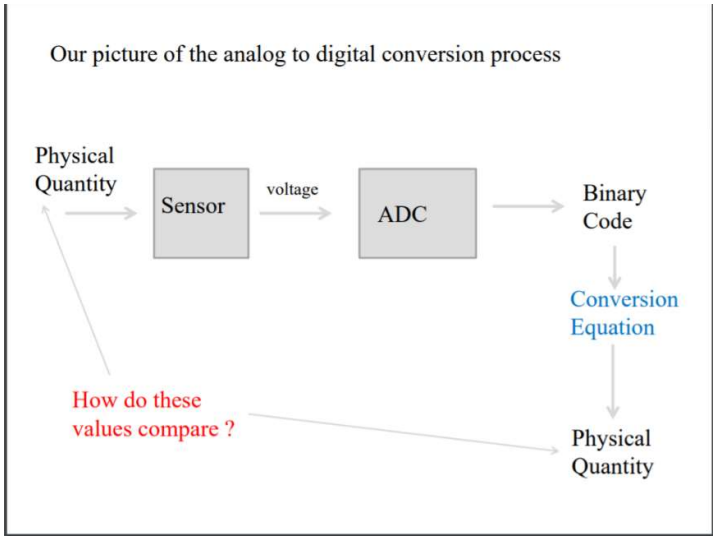
- Difference in O/P voltage caused by single bit change in I/P
  - $Resolution = \frac{Full\ scale}{2^n - 1} = K$, where n is the number of bits
- Often useful to express as % of full scale (% Resolution = Step Size/Full Scale x 100%)

Applications

- Used when a digital circuit output must provide an analogue voltage or current
  - Control: a digital computer output to adjust motor speed or furnace temperature
  - Automatic testing: computer generated signals to test analogue circuitry
  - Signal reconstruction: restoring an analogue signal after it has been converted to digital.
  - A/D conversion
  - Direct Digital Synthesis

Sampling and converting an analogue waveform

- Analog signals vary continuously in both amplitude and time. In order to sample such a signal, we must take a "snapshot" at a certain instant in time. Taking this "snapshot" will have errors and uncertainties in both amplitude and time



To convert a waveform into a digital format:

- Sample the waveform at certain time intervals
- Convert the amplitude of the signal at that time into a digital (binary) signal.
- Provide a time stamp for the time of sampling
- Convert the digital signal into a decimal number to make it comprehensible to the human operator

Practical embodiment of an A to D.

- Compare the incoming voltage to certain, set reference levels
- Set the binary output state according to this comparison

The Comparator: based on an operational amplifier

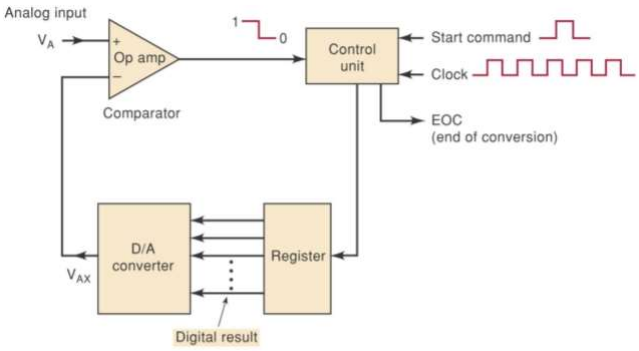
- It takes an analogue signal at its input and compares it to a reference analogue signal.
- Depending on the input, the output of the comparator is either LO or HI

Vin < Vref	LO
Vin > Vref	HI

- Operation of many types of A/D use comparators

Feedback ADCs

- Uses a DAC and a comparator; a guess is applied and the resulting output is compared with the input.



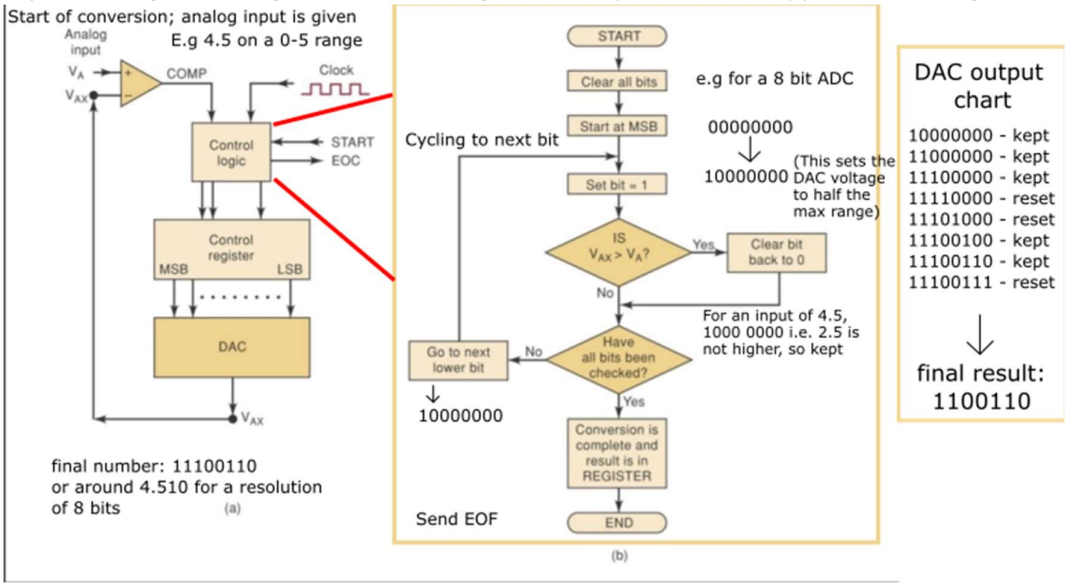
Analogue to Digital and Digital to Analog Conversion.

Ramp/staircase/counter ADC

- Uses a binary up counter into digital to analogue converter (DAC). DAC generates an analogue voltage representing the binary count of the counter
- Increasing analogue voltage is compared by comparator to the analogue voltage to be measured (Vin) and the comparator will change state when VDAC > Vin
- Comparator transition stops counter and also signal “End Of Conversion” The counter value representing the binary conversion of the analogue voltage.
- Easy and simple but
    - Time to perform a conversion is not constant – depends on the magnitude of Vin – large voltage will take more clock cycles to convert.
    - Start counter at zero every time – makes it a relatively slow conversion process.
- Tracking ADC is simple improvement; instead of starting from 0, start from last recorded value

Successive approximation A/D

- Binary Search algorithm: digital conversion is generated by a successive approximation register.



- Fixed conversion time – number of comparison steps that need to be done is always equal to the number of output bits in the converter. More bits will thus lead to better accuracy but also a slower conversion process.
- Conversion speed is relatively fast compared to some other types of converters (eg. ramp converters) but not as fast as the high-priced flash converters. Moderately priced.

Multiplexing

- Several analogue inputs can be digitised together with one ADC by multiplexing the signals. The multiplexer inputs are set to the analogue inputs and a clock with a MOD4 counter, which cycles through each input, sampling the signals sequentially. This however reduces the sampling rate for each channel regardless of how fast the clock can switch the mux

Sample and hold circuit

- Sampling rate theorem: In order to construct the original waveform correctly, the sampling rate must be greater than twice the highest frequency. If the Nyquist-Shannon theorem is not satisfied there will be alias frequencies present in our sampled data i.e. aliasing
- The sample-and-hold circuit essentially takes a “snapshot” of the current voltage of a time varying signal,
- Useful for a very high frequency input signal, especially when a suitably fast ADC is not available/feasible.
- Typical circuit consists of two Op-Amps and a capacitor with a control line for sampling and holding. When the circuit is sample mode, the switch is closed and the capacitor is connected to the buffer amplifier, charging to the output voltage – in hold mode, the switch is disconnected, and the signal voltage is stored in the capacitor, which can be read through the second Op-amp. This enables the signal duration to be increased, allowing the ADC conversion to occur without significant errors from the signal switching too fast.

Memory

Memory Terminology

- Memory cell: a device or circuit used to store a single memory bit – e.g. : a FF, a capacitor (charged/uncharged), spit on magnetic or optical disc.
- Memory word: group of bits (cells) that represents data or an instruction of some type. Word sizes typically ranges from 8 bits to 64 bits.
- Byte: group of 8 bits
- Capacity: Measure of the number of bits that can be stored in a memory device. – Also termed the density
- Density calculations
    - A memory consists of 4096 words of 20 bits each. Total capacity thus 4096 x 20 = 81 920 bits
    - Number of words typically represented as a multiple of 1024 (2^{10} = 1K)
- Address: A unique number (binary/hex code that identifies the location of a word in memory.
- Read operation: The operation whereby the binary word stored in a specific memory location is sensed and transferred to another device. Also called a fetch operation.
- Write operation – Operation whereby a new word is placed at a memory location, replacing the word previous stored there. Also referred to as a store operation.
- Access time (tACC) – Amount of time required to perform a read operation.

Memory types

- Volatile memory: Memory that requires the application of electrical power to retain information – typical semiconductor memory
- Non-volatile memory: retains memory when no electrical power is supplied e.g., magnetic or optical memory
- Random access memory (RAM): memory in which the physical location of a memory word has no effect on how long it takes to access – same for all words in system.
- Sequential access memory (SAM): access time is not constant but depends on address location. Used when words must be accessed in the same order every time- e.g. video memory, DVD and even DRAM.
- Read write memory (RWM): memory that can be written to or read from with equal ease.

Memory Hierarchy

- Cache (SRAM): high speed block of memory that operates between the main memory and the CPU to optimise speed
- Main memory (DRAM): the computer's working memory; stores the instructions and data that the CPU is currently working on.
- Auxiliary memory (mass storage): stores large amount of data external to the main memory. Slower than main memory and always non-volatile.

Write process

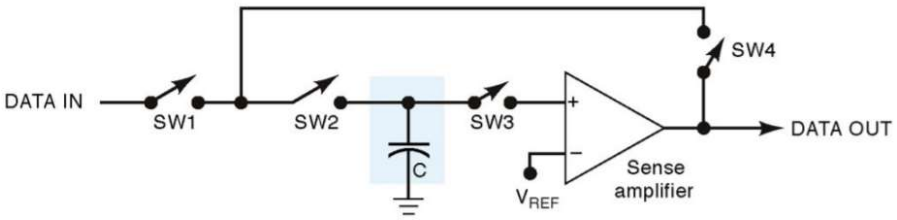
1. CPU supplies the binary address of the memory location where the data will be stored. It places the memory location address on the address bus.
2. Address decoder activates the required memory device (IC)
3. The CPU places data to be stored on the data bus.
4. The CPU activates the control signal for the W operation (high to low)
5. The memory IC internally decodes the address to determine the location for the store operation.
6. The CPU signals to the memory when it should latch the data. This is normally done by changing the state of the write control signal. (low to high) 7. Data on the data bus is transferred to the memory location.

Read process

1. CPU places address of memory location for data retrieval on the address bus
2. An address decoder activates the enable input of the required memory device.
3. CPU activates the control signal lines for the R operation (high to low)
4. Memory ICs determine location of data being retrieved
5. Memory IC places data from the memory location onto the data bus
6. CPU latches data and signals read cycle complete by changing the state of either the R or Enable control lines. (low to high)

Memory Types

- Read only memory (ROM): Designed for applications with many read cycles, typically only one write cycle.
    - Holds data that does not change, or changes only infrequently; non-volatile
    - Typical application: Program or permanent data in PC. Boot program, Look up table.
    - Different types of ROM:
        - Mask programmed ROM (MROM)
            - Done during IC manufacturing process; photolithographic "mask" establishes electrical interconnections
            - Not user programmable and economical only in high volume applications
        - Electrically erasable PROM (EEPROM)
            - Uses a special transistor where a permanent charge is optionally stored that alters the behaviour of the transistor. Voltage is used to charge or discharge memory element. Individual bytes (words) can be erased, but often done a "sector" at a time.
        - Flash Memory – type of EEPROM
            - Allow rapid in-circuit reprogramming of individual blocks. Combines best features of EEPROM – All Flash memory is electrical variants of EEPROM.
        - Optical ROM: Light is used to stores binary data
    - Applications
        - Embedded microcontroller program memory, data transfer and portability, data tables, data converter, function generator, auxiliary storage
            - Can store samples of a waveform for a function generator, or measured values for comparison in a data converter
- Semiconductor RAM: random access memory; all locations are equally accessible. Used for temporary storage; volatile.
    - Fast read/write times are necessary
    - Static RAM (SRAM): semiconductor memory in which the stored data will remain permanently so long as power is applied without the need for refreshing (re-writing the data)
        - Flip-flop type storage element; usually very fast, used as cache
    - Dynamic RAM (DRAM): semiconductor memory in which the stored data needs to be constantly refreshed
        - High capacity; moderate speed. Capacitors used to store data are periodically refreshed by DRAM controller. Used for main internal memory in PCs
        - DRAM structure and operation



Symbolic representation of a dynamic memory cell. During a WRITE operation, semiconductor switches SW1 and SW2 are closed. During a read operation, all switches are closed except SW1.

    - When a read operation is performed on a cell, all the cells in the row will be refreshed. Refresh control logic is used to make sure that each cell is refreshed within the time limit
    - DRAM Technology
        - •DRAM: standard DRAM with fixed random access time.
        - SDRAM: Synchronous Dynamic RAM. Access time is reduced for a burst of sequential reads
        - DDR SDRAM: Double Data Rate SDRAM. One word is transferred on each system clock edge. Using both clock edges meaning we can use a lower clock frequency.